User Guide

AWS Deadline Cloud

Version latest

AWS Deadline Cloud: User Guide

Table of Contents

What is AWS Deadline Cloud?

Deadline Cloud is an AWS service you can use to create and manage rendering projects and jobs on Amazon Elastic Compute Cloud (Amazon EC2) instances directly from digital content creation pipelines and workstations.

Deadline Cloud provides console interfaces, local applications, command line tools, and an API. With Deadline Cloud, you can create, manage, and monitor farms, fleets, jobs, user groups, and storage. You can also specify hardware capabilities, create environments for specific workloads, and integrate the content creation tools that your production requires into your Deadline Cloud pipeline.

Deadline Cloud provides a unified interface to manage all of your rendering projects in one place. You can manage users, assign projects to them, and grant permissions for job roles.

Topics

Features of Deadline Cloud

Here are some of the key ways Deadline Cloud can help you run and manage visual compute workloads:

- Quickly create your farms, queues, and fleets. Monitor their status, and gain insights into the operation of your farm and jobs.
- Centrally manage Deadline Cloud users and groups, and assign permissions.
- Manage sign-in security for project users and external identity providers with AWS IAM Identity Center.

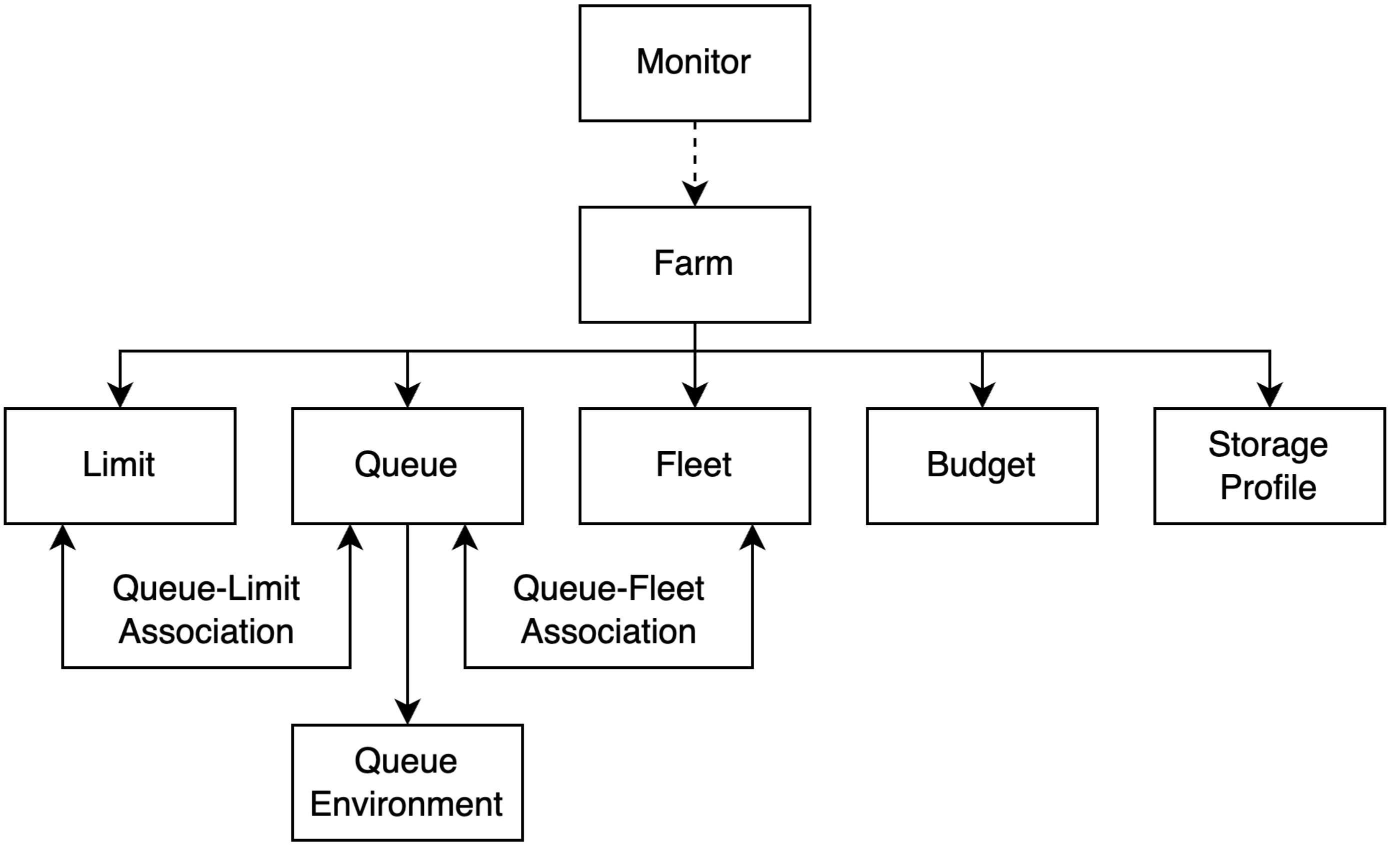
- Securely manage access to project resources with AWS Identity and Access Management (IAM) policies and roles.
- Use tags to organize and quickly find project resources.
- Manage project resource usage and estimated costs for your project.
- Provide a wide range of compute management options to support rendering in the cloud or in person.

Concepts and terminology for Deadline Cloud

To help you get started with AWS Deadline Cloud, this topic explains some of its key concepts and terminology.

Farm resources

This diagram shows how Deadline Cloud farm resources work together.



Farm

A farm contains all other resources related to submitting and running jobs. Farms are independent from each other making them useful for separating production environments.

Queue

A queue holds jobs for scheduling on associated fleets. Users can submit jobs to a queue and manage their priority and status inside the queue. A queue must be associated with a fleet with a queue-fleet association for its jobs to be run, and queues can be associated with multiple fleets.

Fleet

A fleet contains compute capacity for running jobs. Fleets can be service-managed or customer-managed. Service-managed fleets run in Deadline Cloud and include built-in functionality like autoscaling, licensing, and software access. Customer-managed fleets run on your own compute resources like Amazon EC2 instances or on-premises servers.

Budget

A budget sets spending thresholds for your job activity and allows you to take actions when thresholds are reached, such as stopping job scheduling.

Queue environment

A queue environment defines scripts that run on each worker to set up or tear down the workload environment. They are useful for setting environment variables, installing software, and configuring asset storage.

Storage profile

A storage profile is a configuration for a group of hosts and workstations, that tells where data is located on the file system. Deadline Cloud uses storage profiles to map paths when running jobs on differently configured hosts, such as a job submitted from Windows and running on Linux.

Limit

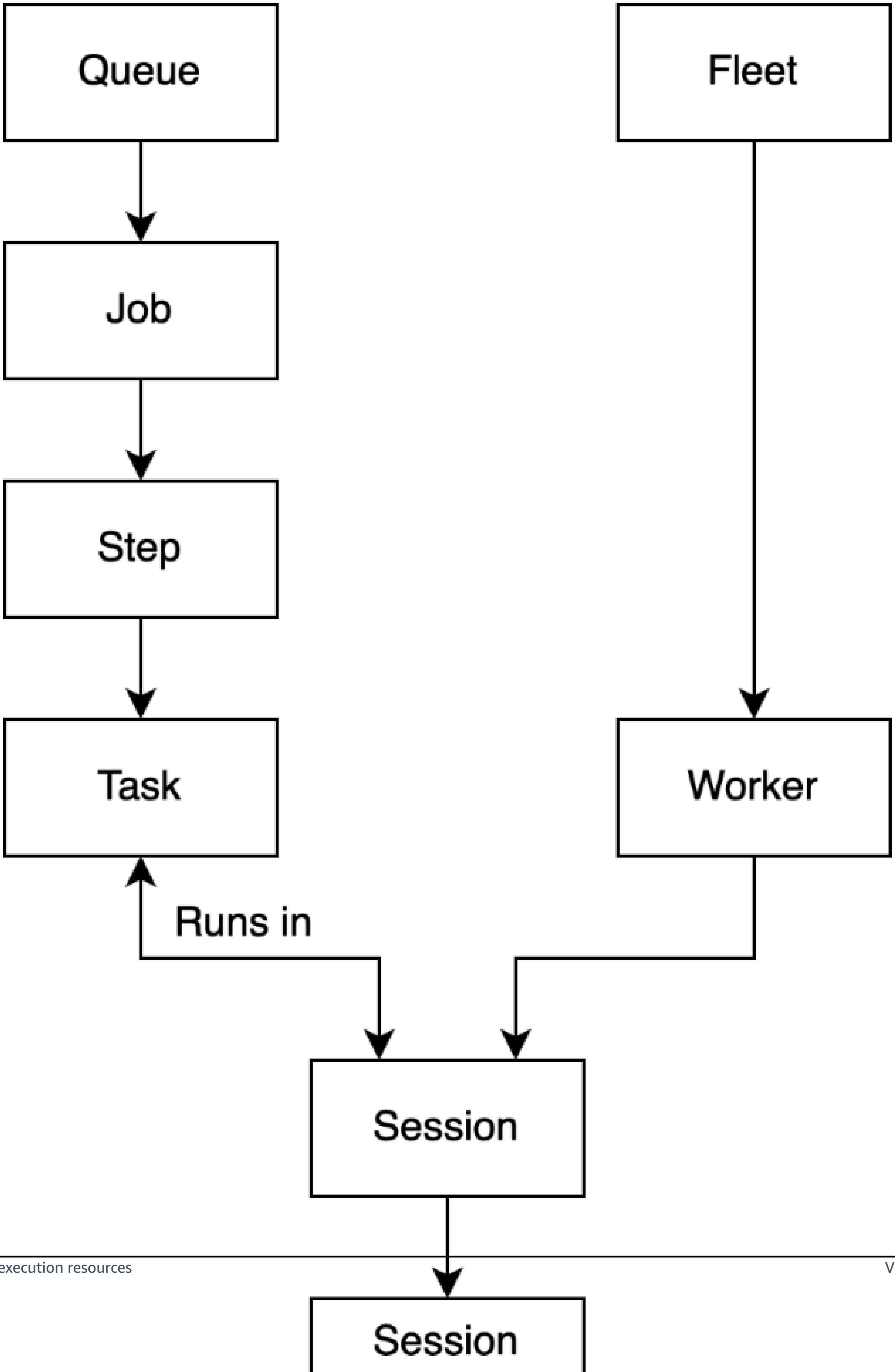
A limit allows you to track usage of shared resources such as floating licenses and control how they are allocated between jobs. Limits are associated with queues with queue-limit associations.

Monitor

The monitor configures the URL for the Deadline Cloud monitor web application, allowing end users to monitor and manage jobs. It can be accessed in a browser or through the Deadline Cloud monitor desktop application.

Job execution resources

This diagram shows how Deadline Cloud job resources work together.


Queue
Fleet
Job
Step
Task
Worker
Runs in
Session
Session

Job

A job is a set of work that a user submits to Deadline Cloud to be scheduled and run on available workers. A job may render a 3D scene or run a simulation. Jobs are created from reusable job templates, which define the runtime environment and processes, and job-specific parameters. Jobs contain steps and tasks that define the work to be performed, and they can be configured with priorities, maximum worker counts, and retry settings.

Job priority

Job priority is the approximate order that Deadline Cloud processes a job in a queue. You can set the job priority between 1 and 100, jobs with a higher number priority are generally processed first. Jobs with the same priority are processed in the order received.

Job properties

Job properties are settings that you define when submitting a render job. Some examples include frame range, output path, job attachments, renderable camera, and more. The properties vary based on the DCC that the render is submitted from.

Step

A step is part of a job that provides a template for running many tasks that are identical except for the task parameter values. Steps can have dependencies on other steps, allowing you to create complex workflows with sequential or parallel execution paths. In rendering jobs, a step often defines the command for rendering a frame and uses the frame number as the task parameter.

Task

A task is the smallest unit of work in Deadline Cloud. Tasks are part of steps and are executed by workers, representing individual operations that need to be performed as part of a job. Tasks can be configured with specific parameters and are assigned to workers based on their capabilities and availability. In rendering jobs, a task often renders a single frame.

Worker

Workers are part of a fleet and execute tasks from jobs. Workers can be configured with specific capabilities such as GPU accelerators, CPU architecture, and operating system. In service-managed fleets, workers are created automatically as the fleet scales out and in.

Instance

Fleets use instances for CPU resources. An instance is an Amazon EC2 performance instance. Deadline Cloud uses On-Demand and Spot instances.

On-Demand instance

On-Demand instances are priced by the second, have no long-term commitment, and will not be interrupted.

Spot instance

Spot instances are unreserved capacity that you can use at a discounted price, but may be interrupted by On-Demand requests.

Wait and Save

The Wait and Save feature provides delayed job scheduling for lower cost and can be interrupted by On-Demand and Spot requests. Wait and Save is only available within Deadline Cloud service-managed fleets.

Wait and Save is for managing the execution of visual computing workloads in AWS Deadline Cloud. See AWS service terms for details.

Session

A session represents a worker's sequence of work on a job. During a single session, a worker may be assigned multiple tasks which it runs one after another. Sessions often have setup actions which configure environments and load assets before running the task actions.

Session action

A session action represents specific operations performed during a session such as setting up the environment, running a task, and syncing assets.

Other important concepts and terminology

Usage explorer

Usage explorer is a feature of Deadline Cloud monitor. It provides an approximate estimate of your costs and usage.

Budget manager

Budget manager is part of the Deadline Cloud monitor. Use the budget manager to create and manage budgets. You can also use it to limit activities to stay within budget.

Deadline Cloud client library

The open-source client library includes a command line interface and library for managing Deadline Cloud. Functionality includes submitting job bundles based on the Open Job Description specification to Deadline Cloud, downloading job attachment outputs, and monitoring your farm using the command line interface (CLI).

Digital content creation application (DCC)

Digital content creation applications (DCCs) are third-party products where you create digital content. Deadline Cloud has built-in integrations with many DCCs such as Autodesk Maya, Blender, and Maxon Cinema 4D allowing you to submit jobs from within the DCC and render on service-managed fleets with pre-configured software and licensing.

Job attachments

Job attachments are a Deadline Cloud feature that you upload and download assets as part of a job such as textures, 3D models, and lighting rigs. Job attachments are stored in Amazon S3 and avoid the need for shared network storage.

Job template

A job template defines the runtime environment and all processes that run as part of a Deadline Cloud job.

Deadline Cloud submitter

A Deadline Cloud submitter is a plugin for a DCC that allows users to easily submit jobs from within the DCC.

License endpoint

A license endpoint makes Deadline Cloud's usage-based licensing for third-party products available inside your VPC. This model is pay as you go, and you are charged for the number of hours and minutes that you use. License endpoints are not connected to farms and can be used independently.

Tags

A tag is a label that you can assign to an AWS resource. Each tag consists of a key and an optional value that you define. With tags, you can categorize your AWS resources in different ways, such as by purpose, owner, or environment.

Usage-based licensing (UBL)

> Usage-based licensing (UBL) is an on-demand licensing model that is available for select third-party products. This model is pay as your go, and you are charged for the number of hours and minutes that you use.

Getting started with Deadline Cloud

Use Deadline Cloud to quickly create a render farm with default settings and resources, such as Amazon EC2 instance configuration and Amazon Simple Storage Service (Amazon S3) buckets.

You can also define the settings and resources when you create a render farm. This method takes more time than using the default settings and resources but gives you more control.

After you're familiar with Deadline Cloud Concepts and terminology, see Getting started for step-by-step instructions for creating your farm, adding users, and links to helpful information.

Accessing Deadline Cloud

You can access Deadline Cloud in any of the following ways:

- **Deadline Cloud console**– Access the console in a browser to create a farm and its resources, and manage user access. For more information, see Getting started.
- **Deadline Cloud monitor**– Manage your render jobs, including updating priorities and job statuses. Monitor your farm and view logs and job status. For users with Owner permissions, the Deadline Cloud monitor also provides access to explore usage and create budgets. The Deadline Cloud monitor is available as both a web browser and a desktop application.
- **AWS SDK and AWS CLI**– Use the AWS Command Line Interface (AWS CLI) to call the Deadline Cloud API operations from the command line on your local system. For more information, see Set up a developer workstation.

Related services

Deadline Cloud works with the following AWS services:

- **Amazon CloudWatch**– With CloudWatch, you can monitor your projects and associated AWS resources. For more information, see Monitoring with CloudWatch in the *Deadline Cloud Developer Guide*.

- **Amazon EC2**–This AWS service provides virtual servers that run your applications in the cloud. You can configure your projects to use Amazon EC2 instances for your workloads. For more information, see Amazon EC2 instances.
- **Amazon EC2 Auto Scaling**– With Auto Scaling, you can automatically increase or decrease the number of instances as the demand on your instances changes. Auto Scaling helps to make sure that you're running your desired number of instances, even if an instance fails. If you enable Auto Scaling with Deadline Cloud, instances that are launched by Auto Scaling are automatically registered with the workload. Likewise, instances that are terminated by Auto Scaling are automatically de-registered from the workload. For more information, see the Amazon EC2 Auto Scaling User Guide.
- **AWS PrivateLink**– AWS PrivateLink provides private connectivity between virtual private clouds (VPCs), AWS services, and your on-premises networks, without exposing your traffic to the public internet. AWS PrivateLink makes it easy to connect services across different accounts and VPCs. For more information, see AWS PrivateLink.
- **Amazon S3**– Amazon S3 is an object storage service. Deadline Cloud uses Amazon S3 buckets to store job attachments. For more information, see the Amazon S3 User Guide.
- **IAM Identity Center**– IAM Identity Center is an AWS service where you can provide users with single sign-on access to all their assigned accounts and applications from one place. You can also centrally manage multi-account access and user permissions to all of your accounts in AWS Organizations. For more information, see AWS IAM Identity Center FAQs.

How Deadline Cloud works

With Deadline Cloud, you can create and manage rendering projects and jobs directly from digital content creation (DCC) pipelines and workstations.

You submit jobs to Deadline Cloud using the AWS SDK, AWS Command Line Interface (AWS CLI), or Deadline Cloud job submitters. Deadline Cloud supports the Open Job Description (OpenJD) for job template specification. For more information, see Open Job Description on the GitHub website.

Deadline Cloud provides job submitters. A *job submitter* is a DCC plugin for submitting render jobs from a third-party DCC interface, such as Maya or Nuke. With a submitter, artists can submit rendering jobs from a third-party interface to Deadline Cloud where project resources are managed and jobs are monitored, all in one location.

With a Deadline Cloud farm, you can create queues and fleets, manage users, and manage project resource usage and costs. A *farm* consists of queues and fleets. A *queue* is where submitted jobs are located and scheduled to be rendered. A *fleet* is a group of worker nodes that run tasks to complete jobs. A queue must be associated with a fleet so that the jobs can render. A single fleet can support multiple queues and a queue can be supported by multiple fleets.

Jobs consist of steps, and each step consists of specific tasks. With the Deadline Cloud monitor, you can access statuses, logs, and other troubleshooting metrics for jobs, steps, and tasks.

Permissions in Deadline Cloud

Deadline Cloud supports the following:

- Managing access to its API operations using AWS Identity and Access Management (IAM)
- Managing access of workforce users using an integration with AWS IAM Identity Center

Before anyone can work on a project, they must have access to that project and the associated farm. Deadline Cloud is integrated with IAM Identity Center to manage workforce authentication and authorization. Users can be added directly to IAM Identity Center, or permission can be connected to your existing identity provider (IdP) such as Okta or Active Directory. IT administrators can grant access permissions to users and groups at different levels. Each subsequent level includes the permissions for the previous levels. The following list describes the four access levels from the lowest level to the highest level:

- **Viewer**– Permission to see resources in the farms, queues, fleets, and jobs they have access to. A viewer can't submit or make changes to jobs.
- **Contributor**– Same as a viewer, but with permission to submit jobs to a queue or farm.
- **Manager**– Same as contributor, but with permission to edit jobs in queues they have access to, and grant permissions on resources that they have access to.
- **Owner**– Same as manager, but can view and create budgets and see usage.

Note

These permissions don't give users access to the AWS Management Console or permission to modify Deadline Cloud infrastructure.

Users must have access to a farm before they can access the associated queues and fleets. User access is assigned to queues and fleets separately within a farm.

You can add users as individuals or as part of a group. Adding groups to a farm, fleet, or queue can make it easier to manage access permissions for large groups of people. For example, if you have a team that is working on a specific project, you can add each of the team members to a group. Then, you can grant access permissions to the entire group for the corresponding farm, fleet, or queue.

Software support with Deadline Cloud

Deadline Cloud works with any software application that can be run from a command line interface and controlled by using parameter values. Deadline Cloud supports the OpenJD specification for describing work as **jobs** with software script **steps** that are parameterized (such as across a frame range) into **tasks**. Assemble OpenJD job instructions into job bundles with Deadline Cloud tools and features to create, run, and license the steps from a third-party software application.

Jobs need licensing to render. Deadline Cloud offers usage-based-licensing (UBL) for a selection of software application licenses that is billed by the hour in minute increments based on usage. With Deadline Cloud, you can also use your own software licenses if you like. If a job can't access a license, it doesn't render and produces an error that displays in the task log in the Deadline Cloud monitor.

Integrate Deadline Cloud into your pipeline

You can integrate your existing rendering pipelines with AWS Deadline Cloud to streamline your workflow management and job submission processes.

What is pipeline integration?

A pipeline integration of Deadline Cloud refers to how a Deadline Cloud farm provides batch processing for your interactive and automated workflows. This example uses a visual effects pipeline that you can adapt to the applications and processes your operators use in their workflows.

A visual effects pipeline consists of the stages of post-production to process input footage, 3D models, animation, textures, lighting, rendered images, and more. It prescribes how different departments exchange assets to perform the tasks they are responsible for. A well-designed pipeline facilitates efficient creation of final images for a television show or similar.

By integrating a Deadline Cloud farm into your pipeline, you can offload long-running jobs onto a queue, and prioritize how Deadline Cloud schedules them on fleets of worker hosts. You can use fleets managed by the service, and you can create your own fleets on-premises or on AWS.

For creating your pipeline integration, consider the following factors:

- Where is your asset data stored, and how will you provide them to worker hosts in the farm?
- Which applications and plugins do your jobs need, and how will you provision them onto worker hosts in the farm?
- When artists or other operators have jobs to run, how will they submit them to the farm?
- Who will monitor the progress and status of jobs, and how will you control costs and optimize the utilization of worker hosts?

Example of an on-premises studio with a farm on AWS

This example focuses on a pipeline where artists work together on-premises and submit jobs to a farm on AWS for rendering. The approach presented here is quick to onboard onto Deadline Cloud and provides a flexible starting point for customization.

Here are the factors for pipeline integration of this example studio:

- Asset data is stored on a NAS shared file system in their on-premises office.
  - On Windows, projects are mounted to the P: drive and utilities are mounted to X:.
  - On macOS, projects are mounted to /Volumes/Projects and utilities are mounted to /Volumes/Utilities.
- They use Maya for 3D modeling, Arnold for rendering, and Nuke for compositing. No custom plugins are installed in these applications.
- They want to use the default submission experience.
- Artists will monitor their own jobs and producers will monitor costs and adjust priorities when needed.

The pipeline integration for this studio uses job attachments to transfer data from the studio premises to and from AWS, as it can be easy to get started with and can scale to large fleet sizes. The job attachments S3 bucket configured on the queue acts as a cache tier between the on-premises NAS and worker hosts on AWS.

When artists submit jobs from Maya or Nuke, the Deadline Cloud integrated submitter scans the scene to identify the files needed for the job to run, and then attaches them to the job by uploading them to S3. A high performance hash is used to identify files that were previously uploaded by any artist in the studio. This way, when an artist is iteratively submitting new versions of the same shot, or one artist hands a shot off to another, only new or modified files need to be uploaded in the process of submitting the job.

The studio uses both Windows and macOS workstations, so they configure storage profiles with file system locations of local type for both their projects and utilities drive. See the Storage profiles for job attachments topic for more details about how this supports the path mapping necessary when jobs run on a different operating system than they are submitted from. They also configure a Linux host on their network to automatically download the output of all tasks of jobs in the queue when they complete. To learn how to set it up, see Automatic downloads for job attachments.

The farm contains two Linux service-managed fleets with vCPUs and RAM requirements set to ranges starting from a minimum specification the studio needs for their jobs. One of the fleets is configured to provide a small number of spot instances to provide consistent render capacity during work hours, and the other fleet is configured as wait and save to render more jobs during off-peak hours at a lower cost. All of Maya, the Maya for Arnold plugin, and Nuke are provided for Linux service-managed fleets from the deadline-cloud conda channel, alongside usage-based licensing. In order to save overhead from application installation, they replace the default conda environment configured for the queue in the Deadline Cloud console with the github sample conda queue environment with improved caching.

To support job submission, they set up Deadline Cloud submitters on each workstation, selecting the Maya and Nuke integrations. With Deadline Cloud monitor, they can log into the farm, monitor progress of jobs, and view log outputs for diagnosing issues. Both the Maya and Nuke submitters feature integrated dialogs to submit jobs from within the application interface.

When configuring user access levels in the farm, they give Contributor access to artists so they can submit jobs, view all jobs, and modify properties of their own jobs. They give Manager access to render wranglers so they can modify properties of all the jobs. They give Owner access to producers, so they can track spending and usage by creating budgets and exploring usage costs.

Getting started with Deadline Cloud

To create a farm in AWS Deadline Cloud, you can use either the Deadline Cloud console or the AWS Command Line Interface (AWS CLI). Use the console for a guided experience creating the farm, including queues and fleets. Use the AWS CLI to work directly with the service, or for developing your own tools that work with Deadline Cloud.

To create a farm and use the Deadline Cloud monitor, set up your account for Deadline Cloud. You only need to set up the Deadline Cloud monitor infrastructure once per account. From your farm, you can manage your project, including user access to your farm and its resources.

To create a farm with minimal resources to accept jobs, select **Quickstart** in the console home page. **Set up the Deadline Cloud monitor** walks you through those steps. These farms start with a queue and a fleet that are automatically associated. This approach is a convenient way to create sandbox style farms to experiment in.

Topics

Set up your AWS account

Set up your AWS account to use AWS Deadline Cloud.

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

When you first create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account.

> ⚠ **Important**
>
> We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Set up the Deadline Cloud monitor

To get started, you'll need to create your Deadline Cloud farm infrastructure, including a monitor, queue, and fleet. You can also perform additional, optional steps including adding groups and users, choosing a service role, and adding tags to your resources.

Step 1: Create your monitor

The Deadline Cloud monitor uses AWS IAM Identity Center to authorize users. By default, the IAM Identity Center instance that you use for Deadline Cloud must be in the same AWS Region as the monitor. However, if you have Multi-Region support enabled in IAM Identity Center, you can create a monitor in a different Region. For more information, see What is AWS IAM Identity Center. If your console is using a different Region when you create the monitor, you'll get a reminder to change to the IAM Identity Center Region.

Your monitor's infrastructure consists of the following components:

- **Monitor name**: The **Monitor name** is how you can identify your monitor — for example *AnyCompany monitor*. Your monitor's name also determines your **monitor URL**.
- **Monitor URL**: You can access your monitor by using the **Monitor URL**. The URL is based on the **Monitor name** — for example *https://anycompanymonitor.awsapps.com*.

- **AWS Region**: The **AWS Region** is the physical location for a collection of AWS data centers. When you set up your monitor, the Region defaults to the closest location to you. We recommend changing the Region so it is located closest to your users. This reduces lag and improves data transfer speeds. By default, AWS IAM Identity Center must be enabled in the same AWS Region as Deadline Cloud, unless you have Multi-Region support enabled in IAM Identity Center. For more information, see What is AWS IAM Identity Center.

  > ⚠ **Important**
  >
  > You can't change your Region after you finish setting up Deadline Cloud.

Complete the tasks in this section to configure your monitor's infrastructure.

To configure your monitor's infrastructure

1. Sign in to the **AWS Management Console** to start the Welcome to Deadline Cloud setup, then choose **Next**.
2. Enter the **Monitor name** — for example **`AnyCompany Monitor`**.
3. (Optional) To change the **Monitor URL**, choose **Edit URL**.
4. (Optional) To change the **AWS Region** so it's closest to your users, choose **Change Region**.
   a. Select the Region closest to your users.
   b. Choose **Apply Region**.
5. (Optional) To further customize your monitor setup, select **Additional settings**.
6. If you are ready for Step 2: Define farm details, choose **Next**.

Additional settings

Deadline Cloud setup includes additional settings. With these settings, you can view all the changes Deadline Cloud setup makes to your AWS account, configure your monitor user role, and change your encryption key type.

AWS IAM Identity Center

AWS IAM Identity Center is a cloud-based single sign-on service for managing users and groups. IAM Identity Center can also be integrated with your enterprise single sign-on (SSO) provider so that users can sign in with their company account.

Deadline Cloud enables IAM Identity Center by default, and it is required to set up and use Deadline Cloud. By default, the IAM Identity Center instance that you use for Deadline Cloud must be in the same AWS Region as the monitor. However, if you have Multi-Region support enabled in IAM Identity Center, you can create a monitor in a different Region. For more information, see What is AWS IAM Identity Center.

Configure service access role

An AWS service can assume a service role to perform actions on your behalf. Deadline Cloud requires a monitor user role for it to give users access to resources in your monitor.

You can attach AWS Identity and Access Management (IAM) managed policies to the monitor user role. The policies give users permissions to perform certain actions, such as creating jobs in a specific Deadline Cloud application. Because applications depend on specific conditions in the managed policy, if you don't use the managed policies, the application might not perform as expected.

You can change the monitor user role after you complete setup, at any time. For more information about user roles, see IAM Roles.

The following tabs contain instructions for two different use cases. To create and use a new service role, choose the **New service role** tab. To use an existing service role, choose the **Existing service role** tab.

New service role

To create and use a new service role

1. Select **Create and use a new service role**.
2. (Optional) Enter a **Service user role** name.
3. Choose **View permission details** for more information about the role.

Existing service role

To use an existing service role

1. Select **Use an existing service role**.
2. Open the dropdown list to choose an existing service role.
3. (Optional) Choose **View in IAM console** for more information about the role.

Step 2: Define farm details

Back on the Deadline Cloud console, complete the following steps to define the farm details.

1. In **Farm details**, add a **Name** for the farm.
2. For **Description**, enter the farm description. A description can help you identify your farm's purpose.
3. Create a group and add uses for your farm. After you set up your farm, you can use the Deadline Cloud management console to add or change groups and users.
4. (Optional) Choose **Additional farm settings**.

    a. (Optional) By default, your data is encrypted with a key that AWS owns and manages for your security. You can choose **Customize encryption settings (advanced)** to use an existing key or to create a new one that you manage.

    If you choose to customize encryption settings using the checkbox, enter a AWS KMS ARN, or create a new AWS KMS by choosing **Create new KMS key**.

    b. (Optional) Choose **Add new tag** to add one or more tags to your farm.
5. Choose one of the following options:

    - Select **Skip to Review and Create** to review and create your farm.
    - Select **Next** to proceed to additional, optional steps.

(Optional) Step 3: Define queue details

The queue is responsible for tracking progress and scheduling work for your jobs.

1. Starting in **Queue details,** provide a **Name** for the queue.
2. For **Description**, enter the queue description. A clear description can help you quickly identify your queue's purpose.
3. For **Job attachments**, you can either create a new Amazon S3 bucket or choose an existing Amazon S3 bucket. If you don't have an existing Amazon S3 bucket, you'll need to create one.

    a. To create a new Amazon S3 bucket, select **Create new job bucket**. You can define the name of the job bucket in the **Root prefix** field. We recommend calling the bucket **`deadlinecloud-job-attachments-[QUEUENAME]`**.

You can only use lowercase letters and dashes. No spaces or special characters.

   b. To search for and select an existing Amazon S3 bucket, select **Choose from existing Amazon S3 bucket**. Then, search for an existing bucket by choosing **Browse S3**. When the list of your available Amazon S3 buckets display, select the Amazon S3 bucket you want to use for your queue.

4. (Optional) Choose **Additional farm settings**.

   a. If you are using customer-managed fleets, select **Enable association with customer-managed fleets**.

      i. For customer-managed fleets, add a **Queue-configured user**, and then set the POSIX and/or Windows credentials. Alternatively, you can bypass the run-as functionality by selecting the checkbox.

      ii. If you want to set a budget for a queue, choose **Require a budget for this queue**. If you require a budget, you must create the budget using the Deadline Cloud console to schedule jobs in the queue.

   b. Your queue requires permission to access Amazon S3 on your behalf. We recommend you create a new service role for every queue.

      i. For a new role, complete the following steps.

         A. Select **Create and use a new service role**.

         B. Enter a **Role name** for your queue role or use the provided role name.

         C. (Optional) Add a queue role **Description**.

         D. You can view the IAM permissions for the queue role by choosing **View permission details**.

      ii. Alternatively, you can select an existing service role.

   c. (Optional) Add environment variables for the queue environment using name and value pairs.

   d. (Optional) Add tags for the queue using key and value pairs.

Choose one of the following options:

- Select **Skip to Review and Create** to review and create your farm.
- Select **Next** to proceed to additional, optional steps.

(Optional) Step 4: Define fleet details

A fleet allocates workers to execute your rendering tasks. If you need a fleet for your rendering tasks, check the box for **Create fleet**.

1. **Fleet details**

    a. Provide both a **Name** and optional **Description** for your fleet.

    b. Review the fleet type and operating system for awareness.

2. In the **Instance market type** section, choose either **Spot**, **On-demand**, or **Wait and Save Instance**. Amazon EC2 On-demand instances provide faster availability and Amazon EC2 Spot and Wait and Save instances are better for cost saving efforts.

3. For **Auto scaling** the number of instances in your fleet, choose both a **Minimum** number of instances and a **Maximum** number of instances.

    We strongly recommend to always set the minimum number of instances to `0` to avoid incurring extra costs.

4. Review the worker capabilities for awareness.

5. (optional) Choose **Additional fleet settings**

    a. Your fleet requires permission to write to CloudWatch on your behalf. We recommend you create a new service role for every fleet.

        i. For a new role, complete the following steps.

            A. Select **Create and use a new service role**.

            B. Enter a **Role name** for your fleet role or use the provided role name.

            C. (Optional) Add a fleet role **Description**.

            D. To view the IAM permissions for the fleet role, choose **View permission details**.

        ii. Alternatively, you can use an existing service role.

    b. (Optional) Add tags for the fleet using key and value pairs.

After you enter all the fleet details, choose **Next**.

Step 5: Review and create

Review the information entered to create your farm. When you're ready, choose **Create farm**.

The progress of your farm's creation is displayed on the **Farms** page. A success message displays when your farm is ready for use.

Set up your workstation

This process is for administrators and artists who want to install, set up, and launch the AWS Deadline Cloud submitter. A Deadline Cloud *submitter* is a digital content creation (DCC) plugin. Artists use it to submit jobs from a third-party DCC interface that they're familiar with.

> **Note**
>
> This process must be completed on all workstations that artists will use for submitting renders.

Each workstation must have the DCC installed before installing the corresponding submitter. For example, if you want to download the Deadline Cloud submitter for Blender, you need to have Blender already installed on your workstation.

We provide reasonable defaults for keeping workstations secure. For more information about securing your workstation, see Security best practices - workstations.

Topics

Step 1: Install the Deadline Cloud submitter

The following sections guide you through the steps to install the Deadline Cloud submitter.

> **Note**
>
> **Unreal Engine:** The Unreal Engine submitter is not included in the standard installer and requires a separate setup process. For installation instructions, see the Unreal Engine Submitter Setup Guide.

Download the submitter installer

Before you can install the Deadline Cloud submitter, you must download the submitter installer.

1. Download the submitter installer for your operating system:

Download for Windows	Download for Linux	Download for MacOS (arm64)

2. (Optional) Verify the authenticity of downloaded software.

Install the Deadline Cloud submitter

With the installer, you can install the following submitters:

Software	Supported versions	Windows installer	Linux installer	MacOS (arm64) installer
Adobe After Effects	2024 - 2025	Included	Not included	Included
Autodesk 3ds Max	2024 - 2026	Included	Not included	Not included
Autodesk Arnold for Cinema 4D	4.8.4.1	Included	Not included	Included
Autodesk Arnold for Maya	7.1 - 7.4	Included	Included	Included
Autodesk Maya	2023 - 2026	Included	Included	Included
Autodesk VRED	2025 - 2026	Included	Not included	Not included
Blender	3.6 - 5.0	Included	Included	Included
Chaos V-Ray for Maya	6 - 7	Included	Included	Included

Software	Supported versions	Windows installer	Linux installer	MacOS (arm64) installer
Foundry Nuke	15 - 16	Included	Included	Included
KeyShot Studio	2023 - 2025	Included	Not included	Included
Maxon Cinema 4D	2024 - 2026	Included	Not included	Included
Maxon Redshift for Maya	2025-2026	Included	Included	Included
SideFX Houdini	19.5 - 21.0	Included	Included	Included

> **Note**
>
> **Unreal Engine:** The Unreal Engine submitter is not included in the standard installer and requires a separate setup process. For installation instructions, see the Unreal Engine Submitter Setup Guide.

Windows

1. In a file browser, navigate to the folder where the installer downloaded, and then select `DeadlineCloudSubmitter-windows-x64-installer.exe`.

    a. If a **Windows protected your PC** pop-up displays, choose **More info**.

    b. Choose **Run anyway**.

2. After the AWS Deadline Cloud Submitter Setup Wizard opens, choose **Next**.
3. Choose the installation scope by completing one of the following steps:
    - To install for only the current user, choose **User**.
    - To install for all users, choose **System**.

      If you choose **System**, you must exit the installer and re-run it as an administrator by completing the following steps:

   a. Right-click on **`DeadlineCloudSubmitter-windows-x64-installer.exe`**, and then choose **Run as administrator**.

   b. Enter your administrator credentials, and then choose **Yes**.

   c. Choose **System** for the installation scope.

4. After selecting the installation scope, choose **Next**.
5. Choose **Next** again to accept the installation directory.
6. Select **Integrated submitter for Nuke**, or whichever submitter you want to install.
7. Choose **Next**.
8. Review the installation, and choose **Next**.
9. Choose **Next** again, and then choose **Finish**.

Linux

> **Note**
>
> The Deadline Cloud integrated Nuke installer for Linux and Deadline Cloud monitor can only be installed on Linux distributions with at least GLIBC 2.31.

1. Open a terminal window.
2. To do a system install of the installer, enter the command **`sudo -i`** and press **Enter** to become root.
3. Navigate to the location where you downloaded the installer.

   For example, **`cd /home/`*`USER`*`/Downloads`**.
4. To make the installer executable, enter **`chmod +x DeadlineCloudSubmitter-linux-x64-installer.run`**.
5. To run the Deadline Cloud submitter installer, enter **`./DeadlineCloudSubmitter-linux-x64-installer.run`**.
6. When the installer opens, follow the prompts on your screen to complete the Setup Wizard.

macOS (arm64)

1. In a file browser, navigate to the folder where the installer downloaded, and then select the file.
2. After the AWS Deadline Cloud Submitter Setup Wizard opens, choose **Next**.
3. Choose **Next** again to accept the installation directory.
4. Select **Integrated submitter for Maya**, or whichever submitter you want to install.
5. Choose **Next**.
6. Review the installation, and choose **Next**.
7. Choose **Next** again, and then choose **Finish**.

Step 2: Install and set up Deadline Cloud monitor

You can install the Deadline Cloud monitor desktop application with Windows, Linux, or macOS.

Windows

1. Download the Deadline Cloud monitor installer for Windows:

   Download Deadline Cloud monitor for Windows
2. Run the downloaded installer and follow the prompts to complete the installation.

   To perform a silent install, use the following command:

   ```
   DeadlineCloudMonitor_x64-setup.exe /S
   ```

   By default the monitor is installed in `C:\Users{username}\AppData\Local\DeadlineCloudMonitor`. To change the installation directory, use this command instead:

   ```
   DeadlineCloudMonitor_x64-setup.exe /S /D={InstallDirectory}
   ```

Linux (AppImage)

To install Deadline Cloud monitor AppImage on Debian distros

1. Download the Deadline Cloud monitor AppImage:

[Download Deadline Cloud monitor (AppImage)]

2. **Note**

   This step is for Ubuntu 22 and up. For other versions of Ubuntu, skip this step.

   To install libfuse2, enter:

   ```
   sudo apt update
   sudo apt install libfuse2
   ```

3. To make the AppImage executable, enter:

   ```
   chmod a+x deadline-cloud-monitor_amd64.AppImage
   ```

Linux (Debian)

To install Deadline Cloud monitor Debian package on Debian distros

1. Download the Deadline Cloud monitor Debian package:

   [Download Deadline Cloud monitor (.deb)]

2. **Note**

   This step is for Ubuntu 22 and up. For other versions of Ubuntu, skip this step.

   To install libssl1.1, enter:

   ```
   wget https://archive.ubuntu.com/ubuntu/pool/main/o/openssl/
   libssl1.1_1.1.1f-1ubuntu2_amd64.deb
   sudo apt install ./libssl1.1_1.1.1f-1ubuntu2_amd64.deb
   ```

3. To install the Deadline Cloud monitor Debian package, enter:

   ```
   sudo apt update
   sudo apt install ./deadline-cloud-monitor_amd64.deb
   ```

4. If the install fails on packages that have unmet dependencies, fix the broken packages and then run the following commands.

   ```
   sudo apt --fix-missing update
   sudo apt update
   sudo apt install -f
   ```

Linux (RPM)

To install Deadline Cloud monitor RPM on Rocky Linux 9 or Alma Linux 9

> **Note**
>
> Rocky Linux 9 and Alma Linux 9 use OpenSSL 3.0 by default and don't include the `libssl.so.1.1` library. You must install the `compat-openssl11` package for Deadline Cloud monitor to run.

1. Download the Deadline Cloud monitor RPM:

   Download Deadline Cloud monitor (.rpm)

2. Add the extra packages for the Enterprise Linux 9 repository:

   ```
   sudo dnf install epel-release
   ```

3. Install `compat-openssl11` for the `libssl.so.1.1` dependency:

   ```
   sudo dnf install compat-openssl11 deadline-cloud-monitor.x86_64.rpm
   ```

To install Deadline Cloud monitor RPM on Red Hat Linux 9

> **Note**
>
> Red Hat Linux 9 uses OpenSSL 3.0 by default and doesn't include the `libssl.so.1.1` library. You must install the `compat-openssl11` package for Deadline Cloud monitor to run.

1. Download the Deadline Cloud monitor RPM:

   Download Deadline Cloud monitor (.rpm)

2. Enable the CodeReady Linux Builder repository:

   ```
   subscription-manager repos --enable codeready-builder-for-rhel-9-x86_64-rpms
   ```

3. Install the extra packages for Enterprise RPM:

   ```
   sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-
   latest-9.noarch.rpm
   ```

4. Install `compat-openssl11` for the `libssl.so.1.1` dependency:

   ```
   sudo dnf install compat-openssl11 deadline-cloud-monitor.x86_64.rpm
   ```

To install Deadline Cloud monitor RPM on Rocky Linux 8, Alma Linux 8, or Red Hat Linux 8

1. Download the Deadline Cloud monitor RPM:

   Download Deadline Cloud monitor (.rpm)

2. Install the Deadline Cloud monitor:

   ```
   sudo dnf install deadline-cloud-monitor.x86_64.rpm
   ```

macOS (arm64)

1. Download the Deadline Cloud monitor installer for macOS:

   Download Deadline Cloud monitor for macOS (arm64)

2. Open the downloaded file. When the window displays, select and drag the Deadline Cloud monitor icon into the **Applications** folder.

After you complete the download, you can verify the authenticity of the downloaded software. You might want to do this to ensure no one has tampered with the files during or after the download process. See Verify the authenticity of downloaded software in Step 1.

After downloading Deadline Cloud monitor and verifying the authenticity, use the following procedure to set up the Deadline Cloud monitor.

To set up Deadline Cloud monitor

1. Open **Deadline Cloud monitor**.
2. When prompted to create a new profile, complete the following steps.

   a. Enter your monitor URL into the URL input, which looks like **`https://MY-MONITOR.deadlinecloud.amazonaws.com/`**

   b. Enter a **Profile** name.

   c. Choose **Create Profile**.

   Your profile is created and your credentials are now shared with any software that uses the profile name that you created.
3. After you create the Deadline Cloud monitor profile, you can't change the profile name or the studio URL. If you need to make changes, do the following instead:

   a. Delete the profile. In the left navigation pane, choose **Deadline Cloud monitor** > **Settings** > **Delete**.

   b. Create a new profile with the changes that you want.
4. From the left navigation pane, use the **>Deadline Cloud monitor** option to do the following:
   - Change the Deadline Cloud monitor profile to log in to a different monitor.
   - Enable **Autologin** so you don’t have to enter your monitor URL on subsequent opens of Deadline Cloud monitor.
5. Close the Deadline Cloud monitor window. It continues to run in the background and enable other Deadline Cloud tools to access your render farm.
6. For each digital content creation (DCC) application that you plan to use for your rendering projects, complete the following steps:

   a. From your Deadline Cloud submitter, open the Deadline Cloud workstation configuration.

   b. In the workstation configuration, select the profile that you created in the Deadline Cloud monitor. Your Deadline Cloud credentials are now shared with this DCC and your tools should work as expected.

Step 3: Launch the Deadline Cloud submitter

The following example shows how to install the Blender submitter. You can install other submitters using similar steps.

To launch the Deadline Cloud submitter in Blender

> **Note**
>
> Support for Blender is provided using the conda environment for service-managed fleets. For more information, see Default conda queue environment.

1. Open **Blender**.
2. In the **Render** menu, choose **Submit to AWS Deadline Cloud**.
   1. If you are prompted to install GUI dependencies, choose **OK** and the Deadline Cloud submitter dialog will appear shortly.
   2. If you are not already authenticated in the Deadline Cloud submitter, the **Credentials Status** shows as **NEEDS_LOGIN**.
   3. Choose **Login**. You will be prompted to log in with your user credentials in a browser.
   4. You are now logged in and the **Credentials Status** shows as **AUTHENTICATED**.
3. Choose **Submit**.

Now your job is submitted to your Deadline Cloud farm and will be processed by a compatible fleet. For information on how to view job progress in the monitor, see Using the Monitor.

Using the Deadline Cloud monitor

The AWS Deadline Cloud monitor provides you with an overall view of your visual compute jobs. You can use it to monitor and manage jobs, view worker activity on fleets, track budgets and usage, and to download a job's results.

Each queue has a job monitor that shows you the status of jobs, steps, and tasks. The monitor provides ways to manage jobs directly from the monitor. You can make prioritization changes, cancel jobs, requeue jobs, and resubmit jobs.

The Deadline Cloud monitor has a table that shows summary status for a job, or you can select a job to see detailed task logs that help troubleshoot issues with a job.

You can use the Deadline Cloud monitor to download the results to the location on your workstation that was specified when the job was created.

The Deadline Cloud monitor also helps you monitor usage and manage costs. For more information, see Track spending and usage for Deadline Cloud farms.

Topics

Share the Deadline Cloud monitor URL

When you set up the Deadline Cloud service, by default you create a URL that opens the Deadline Cloud monitor for your account. Use this URL to open the monitor in your browser or on your desktop. Share the URL with other users so that they can access the Deadline Cloud monitor.

Before a user can open the Deadline Cloud monitor, you must grant the user access. To grant access, either add the user to the list of authorized users for the monitor or add them to a group with access to the monitor. For more information, see Managing users in Deadline Cloud.

To share the monitor URL

1. Open the Deadline Cloud console.
2. From **Get started**, choose **Go to Deadline Cloud dashboard**.
3. On the navigation pane, choose **Dashboard**.
4. In the **Account overview** section, choose **Account details**.
5. Copy and then securely send the **URL** to anyone who needs to access the Deadline Cloud monitor.

Open the Deadline Cloud monitor

You can open the Deadline Cloud monitor by any of the following ways:

- **Console** – Sign in to the AWS Management Console and open the Deadline Cloud console.
- **Web** – Go to the monitor URL that you created when you set up Deadline Cloud.
- **Monitor** – Use the desktop Deadline Cloud monitor.

When you use the console, you must be able to sign in to AWS using an AWS Identity and Access Management identity, and then sign in to the monitor with AWS IAM Identity Center credentials. If you only have IAM Identity Center credentials, you must sign in using the monitor URL or the desktop application.

To open the Deadline Cloud monitor (web)

1. Using a browser, open the monitor URL that you created when you set up Deadline Cloud.
2. Sign in with your user credentials.

To open the Deadline Cloud monitor (console)

1. Open the Deadline Cloud console.
2. In the navigation pane, select **Farms**.
3. Select a farm, then choose **Manage jobs** to open the **Deadline Cloud monitor** page.
4. Sign in with your user credentials.

To open the Deadline Cloud monitor (desktop)

1. Open the Deadline Cloud console.

   -or-

   Open the Deadline Cloud monitor - web from the monitor URL.
2. - On the Deadline Cloud console, do the following:
     1. In the monitor, choose **Go to Deadline Cloud dashboard**, and then choose **Downloads** from the left menu.
     2. From **Deadline Cloud monitor**, choose the monitor version for your desktop.
     3. Choose **Download**.
   - On the Deadline Cloud monitor - web, do the following:
     - From the left menu, choose **Workstation setup**. If the **Workstation setup** item isn't visible, use the arrow to open the left menu.
     - Choose **Download**.
     - From **Select an OS**, choose your operating system.
3. Download the Deadline Cloud monitor - desktop.
4. After you download and install the monitor, open it on your computer.

   - If this is your first time opening the Deadline Cloud monitor, you must provide the monitor URL and create a profile name. Next you sign in to the monitor with your Deadline Cloud credentials.
   - After you create a profile, you open the monitor by selecting a profile. You might need to enter your Deadline Cloud credentials.

Change your language settings

After you create and open your Deadline Cloud monitor, you can change your language settings. By default, the monitor language is set to your system's language settings.

To change your language settings from Deadline Cloud monitor (desktop)

1. From your user profile, select **Settings**, then choose **Language**.
2. From the dropdown menu, select one of the available languages.
3. Confirm that your chosen language is the listed option, then choose **Confirm and apply** to apply the change.

   After the monitor refreshes, it displays in the chosen language.

   After you change the language setting, it is the default upon opening and remains the default until you change it again or uninstall the desktop application.

To change the Deadline Cloud monitor language on the web, change the preferred language in your browser settings.

> **Note**
>
> If your browser or operating system is set to a language that is not supported by Deadline Cloud, English becomes the default language for Deadline Cloud monitor.

Submit a job bundle

You can submit a job bundle directly from the AWS Deadline Cloud monitor desktop application. A job bundle is a directory that contains the files and information needed to submit a job to Deadline Cloud. For sample job bundles, see the deadline-cloud-samples repository on GitHub.

To submit a job bundle

- In the Deadline Cloud monitor desktop application, choose **File**, **Submit Job Bundle**. This feature is not available in the Linux AppImage or MacOS x64 builds.

View queue and fleet details in Deadline Cloud

You can use the Deadline Cloud monitor to view the configuration of the queues and fleets in your farm. You can also use the monitor to see a list of the jobs in a queue or the workers in a fleet.

You must have `VIEWING` permission to view queue and fleet details. If the details don't display, contact your administrator to get the correct permissions.

To view queue details

1. Open the Deadline Cloud monitor.
2. From the list of farms, choose the farm that contains the queue that you're interested in.
3. In the list of queues, choose a queue to display its details. To compare the configuration of two or more queues, select more than one check box.
4. To see a list of jobs in the queue, choose the queue name from the list of queues or from the details panel.

If the monitor is already open, you can select the queue from the **Queues** list in the left navigation pane.

To view fleet details

1. Open the Deadline Cloud monitor.
2. From the list of farms, choose the farm that contains the fleet that you're interested in.
3. In **Farm resources**, choose **Fleets**.
4. In the list of fleets, choose a fleet to display its details. To compare the configuration of two or more fleets, select more than one check box.
5. To see a list of workers in the fleet, choose the fleet name from the list of fleets or from the details panel.

If the monitor is already open, you can select the fleet from the **Fleets** list in the left navigation pane.

Manage jobs, steps, and tasks in Deadline Cloud

When you select a queue, the job monitor section of the Deadline Cloud monitor shows you the jobs in that queue, the steps in the job, and the tasks in each step. When you select a job, step, or task, you can use the **Actions** menu to manage each.

To open the job monitor, follow the steps to view a queue in View queue and fleet details in Deadline Cloud, then select the job, step, or task to work with.

For jobs, steps, and tasks, you can do the following:

- Change the status to **Requeued**, **Succeeded**, **Failed**, or **Canceled**.
- Download the processed output from the job, step, or task.
- Copy the ID of the job, step, or task.

For the selected job, you can:

- Archive the job.
- Modify the job properties, including name, description, priority, or max worker count.
- View step to step dependencies.
- View additional details using the job's parameters.
- Resubmit the job.

For for more information, see View and manage job details in Deadline Cloud.

For each step, you can:

- View the dependencies for the step. The dependencies for a step must be completed before the step runs.

For details, see View a step in Deadline Cloud.

For each task, you can:

- View logs for the task.
- View task parameters.

For more information, see View a task in Deadline Cloud.

View and manage job details in Deadline Cloud

The **Job monitor** page in the Deadline Cloud monitor provides you with the following:

- An overall view of the progress of a job.
- A view of the steps and tasks that make up the job.

Choose a job from the list to view a list of steps for the job, and then choose a step from the list of steps to view the tasks for the job. After you choose an item, you can use the **Actions** menu for that item to view details.

To view job details

1. Follow the steps to view a queue in View queue and fleet details in Deadline Cloud.
2. In the navigation pane, select the queue where you submitted your job.
3. Select a job using one of the following methods:

    a. From the **Jobs** list, select a job to view its details.

    b. From the **search** field, enter any text associated with the job, such as the job name or user that created the job. From the results that display, select the job you want to view.

The details of a job include the steps in the job and the tasks in each step. You can use the **Actions** menu to do the following:

- Change the status of the job.
- View and modify the properties of a job.
    - You can view the dependencies between steps in the job.
    - You can change the priority of the job in a queue. Jobs with higher number priority are processed before jobs with lower number priority. Jobs can have a priority between 1 and 100. When two jobs have the same priority, the oldest job is scheduled first.
- View the parameters for the job that were set when the job was submitted.
- Download the output of a job. When you download the output of a job, it contains all of the output generated by the steps and tasks in the job.

Archive a job

To archive a job, it must be in a terminal state, `FAILED`, `SUCCEEDED`, `SUSPENDED`, or `CANCELED`. The `ARCHIVED` state is final. After a job is archived, it can't be requeued or modified.

The job's data is not affected by archiving the job. The data is deleted when the inactivity timeout is reached, or when the queue containing the job is deleted.

Other things that happen to archived jobs:

- Archived jobs are hidden in the Deadline Cloud monitor.
- Archived jobs are visible in a read-only state form the Deadline Cloud CLI for 120 days before deletion.

Requeue a job

When you requeue a job, all of the tasks without step dependencies switch to `READY`. The status of steps with dependencies switch to `READY` or `PENDING` as they are restored.

- All jobs, steps, and tasks switch to `PENDING`.
- If a step doesn't have a dependency, it switches to `READY`.

Resubmit a job

There might be times when you want to run a job again, but with different properties and settings. For example, you might submit a job to render a subset of testing frames, verify the output, then run the job again with the full frame range. To do this, resubmit the job.

When you resubmit a job, new tasks without dependencies become `READY`. New tasks with dependencies become `PENDING`.

- All new jobs, steps, and tasks become `PENDING`.
- If a new step doesn't have a dependency, it becomes `READY`.

When you resubmit a job, you can only change properties that were defined as configurable when the job was first created. For example, if the name of a job is not defined as a configurable property of the job when first submitted, then the name cannot be edited on resubmission.

View a step in Deadline Cloud

Use the AWS Deadline Cloud monitor to view the steps in your processing jobs. In the **Job monitor**, the **Steps** list shows the list of steps that make up the selected job. When you select a step, the **Tasks** list shows the tasks in the step.

To view a step

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select a job from the **Jobs** list.
3. Select a step from the **Steps** list.

You can use the **Actions** menu to do the following:

- Change the status of the step.
- Download the output of the step. When you download the output of a step, it contains all of the output generated by the tasks in the step.
- View the dependencies of a step. The dependencies table shows a list of steps that must be complete before the selected step starts, and a list of steps that are waiting for this step to complete.

View a task in Deadline Cloud

Use the AWS Deadline Cloud monitor to view the tasks in your processing jobs. In the **Job monitor**, the **Tasks** list shows the tasks that make up the step selected in the **Steps** list.

To view a task

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select a job from the **Jobs** list.
3. Select a step from the **Steps** list.
4. Select a task from the **Tasks** list.

You can use the **Actions** menu to do the following:

- Change the status of the task.

- View task logs. For more information, see View session and worker logs in Deadline Cloud.
- View that parameters that were set when the task was created.
- Download the output of the task. When you download the output of a task, it only contains the output generated by the selected task.

View session and worker logs in Deadline Cloud

Logs provide you with detailed information about the status and processing of tasks. In the AWS Deadline Cloud monitor, you can see the following two types of logs:

- *Session logs* detail the timeline of actions, including:
  - Setup actions, such as attachment syncing and loading the software environment
  - Running a task or set of tasks
  - Closure actions, such as shutting down the environment on a worker

  A session includes processing of at least one task, and can include multiple tasks. Session logs also show information about Amazon Elastic Compute Cloud (Amazon EC2) instance type, vCPU, and memory. Session logs also include a link to the log for the worker used in the session.
- *Worker logs* provide details for the timeline of actions that a worker processes during its lifecycle. Worker logs can contain information about multiple sessions.

You can download session and worker logs so that you can examine them offline.

To view session logs

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select a job from the **Jobs** list.
3. Select a step from the **Steps** list.
4. Select a task from the **Tasks** list.
5. From the **Actions** menu, choose **View logs**.

The **Timelines** section shows a summary of the actions for the task. To see more tasks run in the session and to see the shutdown actions for the session, choose **View logs for all tasks**.

To view worker logs from a task

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select a job from the **Jobs** list.
3. Select a step from the **Steps** list.
4. Select a task from the **Tasks** list.
5. From the **Actions** menu, choose **View logs**.
6. Choose **Session info**.
7. Choose **View worker log**.

To view worker logs from fleet details

1. Follow the steps in View queue and fleet details in Deadline Cloud to view a fleet.
2. Select a **Worker ID** from the **Workers** list.
3. From the **Actions** menu, choose **View worker logs**.

View worker details in the worker dashboard

The *worker dashboard* provides details for the worker that processes a task. You can see:

- Metadata, such as the instance type, for the worker
- The session actions that the worker performed
- Worker performance, including CPU, memory and disk usage
- A graph of the CPU, memory, and disk usage over time
- A graph of the disk speed over time
- The worker log for the task

To view the worker dashboard from a task

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select a job from the **Jobs** list.
3. Select a step from the **Steps** list.

4. Select a task from the **Tasks** list.
5. In the task table, from the **Actions** menu, choose **View worker dashboard**.

To view the worker dashboard from fleet details

1. Follow the steps in View queue and fleet details in Deadline Cloud to view a fleet.
2. Select a **Worker** from the **Workers** list.
3. From the **Actions** menu, choose **View worker dashboard**.

Use cases

Detecting under-provisioned instances

When renders take longer than expected, the worker dashboard can help determine if your instances are adequately sized for your workloads. While 100% vCPU utilization is normal for many renderers, consistently high memory usage near maximum capacity and elevated disk space utilization may indicate that your instances are under-provisioned. In such cases, upgrading your fleet's instance configuration can reduce render errors and significantly improve render times. However, it's important to continue monitoring worker performance after upgrading to ensure you've found the optimal balance - upgrading too aggressively can lead to unnecessary costs through over-provisioning.

Detecting over-provisioned instances

Even when tasks are completing successfully, there may be opportunities to optimize your costs. The worker dashboard can reveal if you're paying for more compute power than your workloads require. If you see that the worker has low average vCPU usage, minimal memory utilization, and excess unused disk space, you can downsize the instance configuration of your fleet.

Troubleshooting failed tasks

When investigating failed tasks, the worker dashboard serves as a valuable diagnostic tool. Pay particular attention to peak memory usage and disk space utilization - if these metrics approach or reach 100%, they're likely the root cause of your task failures. Such resource exhaustion indicates that your current instances lack the capacity to handle your workloads effectively. In these cases, provisioning instances with increased memory or disk space will help ensure successful task completion.

Optimal instance utilization rate

vCPU Utilization

Target range: 70–90%

- **Below 70%**: Likely underutilizing compute resources, meaning you're paying for more CPU than your workload needs
- **70–90%**: Optimal range where you're efficiently using resources without hitting bottlenecks
- **Consistently at 100%**: Could indicate CPU bottlenecks that might slow down renders

Keep in mind that some render tasks will naturally be more CPU-intensive than others, and 100% vCPU usage may not be an issue. Real-time visualization tasks might show more consistent CPU utilization, while tasks with changing computational requirements might have varying patterns.

Memory Utilization

Target range: 70–85%

- **Below 50%**: Potentially oversized instances for your workload
- **70–85%**: Optimal utilization with enough headroom for spikes
- **Above 90%**: Risk of performance degradation or out-of-memory errors

Memory requirements can vary significantly depending on scene complexity, texture resolution, and simulation data. Monitoring memory trends over time is important to identify if your workloads are growing in memory requirements.

Disk Space Utilization

Target range: 60–80%

- **Below 40%**: Likely over-provisioned storage
- **60–85%**: Good utilization with room for temporary files and caches
- **Above 85%**: Risk of running out of space during large renders

Remember that disk I/O performance can be just as important as capacity, especially for workloads that read/write large texture or cache files during rendering.

Download finished output in Deadline Cloud

After a job is finished, you can use the AWS Deadline Cloud monitor to download the results to your workstation. The output file is stored with the name and location that you specified when you created the job.

Output files are stored indefinitely. To reduce storage costs, consider creating an S3 Lifecycle configuration for your queue's Amazon S3 bucket. For more information, see Managing your storage lifecycle in the *Amazon Simple Storage Service User Guide*.

To download the finished output of a job, step, or task

1. Follow the steps in View and manage job details in Deadline Cloud to view a list of jobs.
2. Select the job, step, or task that you want to download the output for.
   - If you select a job, you can download all of the output for all of the tasks in all of the steps for that job.
   - If you select a step, you can download all of the output for all of the tasks in that step.
   - If you select a task, you can download the output for that individual task.
3. From the **Actions** menu, choose **Download output**.
4. The output will be downloaded to the location set when the job was submitted.

> **Note**
>
> Downloading output using the menu is currently only supported for Windows and Linux. If you have a Mac and you choose the **Download output** menu item, a window shows the AWS CLI command that you can use to download the rendered output.

Automate Deadline Cloud monitor desktop deployment and workflows

The AWS Deadline Cloud monitor desktop application includes a command-line interface (CLI) that administrators can use to set up profiles for users and that artists and developers can use to integrate the monitor into automated workflows on their workstations.

Finding the Deadline Cloud monitor executable

To use the CLI commands, run the Deadline Cloud monitor executable from a terminal. The default installation location depends on your operating system and installation method.

Windows

```
%LOCALAPPDATA%\DeadlineCloudMonitor\DeadlineCloudMonitor.exe
```

macOS

```
/Applications/DeadlineCloudMonitor.app/Contents/MacOS/DeadlineCloudMonitor
```

Linux (deb or RPM package)

```
/usr/bin/deadline-cloud-monitor
```

Linux (AppImage)

Run the AppImage file directly from the location where you downloaded it.

In the following examples, replace `DeadlineCloudMonitor` with the full path to the executable for your operating system.

Setting up a profile for streamlined user access

Administrators use the `create-profile` command to create Deadline Cloud monitor profiles for users. This command configures a profile so that users can open the monitor, log in, and start working without additional configuration or profile selection.

The `create-profile` command accepts the following flags:

- `--enable-auto-login` – Configures the monitor to automatically log in with the most recently used profile when the application starts.
- `--set-as-deadline-default` – Sets the profile as the default for Deadline Cloud tools, including the Deadline Cloud submitter, the Deadline CLI, and the Deadline Cloud GUI applications. This flag does not affect the AWS Command Line Interface (AWS CLI).

When both flags are enabled, users open the monitor and are logged in automatically with no other configuration or profile selection required.

To create a profile

Run the following command, replacing the placeholder values with your monitor details.

```
DeadlineCloudMonitor create-profile \
    --profile profile-name \
    --monitor-id monitor-id \
    --monitor-url https://monitorName.region.deadlinecloud.amazonaws.com \
    --enable-auto-login \
    --set-as-deadline-default
```

The command creates the profile and writes the configuration to the Deadline Cloud configuration files on the user's workstation. The monitor URL must be in the format `https://monitorName.region.deadlinecloud.amazonaws.com`.

> **Note**
>
> The `create-profile` command exits after creating the profile. To open the monitor with the new profile, run the `login` command or open the Deadline Cloud monitor desktop application.

Integrating the Deadline Cloud monitor into your workflows

Use the `login`, `logout`, and `handle-url` commands to integrate the Deadline Cloud monitor into scripts and automated workflows on your workstation.

Logging in and logging out

Use the `login` and `logout` commands to control authentication as part of a workflow. For example, a script that submits jobs can use the `login` command to ensure the user is authenticated before submission begins.

When you use the `login` command, the monitor opens directly to the specified profile, skipping the profile selection screen. After authentication completes, the monitor minimizes to the system tray so that your workflow can continue. If the monitor is already running for the specified profile, the existing window comes to the foreground instead of starting a new instance.

To log in to a profile

Run the following command, replacing *profile-name* with the name of your Deadline Cloud monitor profile.

```
DeadlineCloudMonitor login --profile profile-name
```

To log out of a profile

Run the following command to clear the credentials for a profile and signal any running monitor instance for that profile to exit.

```
DeadlineCloudMonitor logout --profile profile-name
```

Opening the monitor to a specific page

Use the `handle-url` command to open the Deadline Cloud monitor to a specific page. This command is useful when a script performs an action, such as creating a job, and you want to automatically open the monitor to show the result. For example, after a script submits a job, the script can call `handle-url` to open the monitor directly to the job details page.

You can also use the `deadline-cloud-monitor://` URL as a link on company websites, wikis, or task trackers to let users open the monitor directly to a specific page.

The URL uses the `deadline-cloud-monitor://` protocol scheme with a `launch` command. The URL includes the profile name and the monitor page URL to open.

To open the monitor to a specific page

Run the following command, replacing *monitor-page-url* with the URL-encoded monitor page URL and *profile-name* with your profile name.

```
DeadlineCloudMonitor handle-url --url "deadline-cloud-monitor://launch?url=monitor-
page-url&profile=profile-name"
```

Deadline Cloud farms

With a Deadline Cloud farm, you can manage users and project resources. A *farm* is a where your project resources are located. Your farm consists of queues and fleets. A *queue* is where submitted jobs are located and scheduled to be rendered. A *fleet* is a group of worker nodes that run tasks to complete jobs. After you create a farm, you can create queues and fleets to meet your project's needs.

Create a farm

1. From the Deadline Cloud console, choose **Go to Dashboard**.
2. In the Farms section of the Deadline Cloud dashboard, choose **Actions** → **Create farm**.
   - Alternatively, in the left side panel choose **Farms and other resources**, then choose **Create Farm**.
3. Add a **Name** for your farm.
4. For **Description**, enter the farm description. A clear description can help you quickly identify your farm's purpose.
5. (Optional) By default, your data is encrypted with a key that AWS owns and manages for your security. You can choose **Customize encryption settings (advanced)** to use an existing key or to create a new one that you manage.

   If you choose to customize encryption settings using the checkbox, enter a AWS KMS ARN, or create a new AWS KMS by choosing **Create new KMS key**.
6. (Optional) Choose **Add new tag** to add one or more tags to your farm.
7. Choose **Create farm**. After creation, your farm displays.

Deadline Cloud queues

A queue is a farm resource that manages and processes jobs.

To work with queues, you should already have a monitor and farm set up.

Topics

- Create a queue
- Create a queue environment
- Associate a queue and fleet

Create a queue

1. From the Deadline Cloud console dashboard, select the farm that you want to create a queue for.
    - Alternatively, in the left side panel choose **Farms and other resources**, then select the farm you want to create a queue for.
2. In the **Queues** tab, choose **Create queue**.
3. Enter a name for your queue.
4. For **Description**, enter the queue description. A description helps you identify your queue's purpose.
5. For **Job attachments**, you can either create a new Amazon S3 bucket or choose an existing Amazon S3 bucket.
    a. To create a new Amazon S3 bucket
        i. Select **Create new job bucket**.
        ii. Enter a name for the bucket. We recommend naming the bucket `deadlinecloud-job-attachments-[MONITORNAME]`.
        iii. Enter a **Root prefix** to define or change your queue's root location.
    b. To choose an existing Amazon S3 bucket
        i. Select **Choose an existing S3 bucket** > **Browse S3**.
        ii. Select the S3 bucket for your queue from the list of available buckets.

6. (Optional) To associate your queue with a customer-managed fleet, select **Enable association with customer-managed fleets**.
7. If you enable association with customer-managed fleets, you must complete the following steps.

   > ⚠ **Important**
   >
   > We strongly recommend specifying users and groups for run-as functionality. If you don't, it will degrade your farm’s security posture because the jobs can then do everything the worker's agent can do. For more information about the potential security risks, see Run jobs as users and groups.

   a. For Run as user:

      To provide credentials for the queue's jobs, select **Queue-configured user**.

      Or, to opt out of setting your own credentials and run jobs as the worker agent user, select **Worker agent user**.

   b. (Optional) For Run as user credentials, enter a user name and group name to provide credentials for the queue's jobs.

      If you are using a Windows fleet, you must create an AWS Secrets Manager secret that contains the password for the Run as user. If you don't have an existing secret with the password, choose **Create secret** to open the Secrets Manager console to create a secret. For more information, see Manage access to Windows job user secrets in the *Deadline Cloud Developer Guide*.

8. Requiring a budget helps manage costs for your queue. Select either **Don't require a budget** or **Require a budget**.
9. Your queue requires permission to access Amazon S3 on your behalf. You can create a new service role or use an existing service role. If you don't have an existing service role, create and use a new service role.

   a. To use an existing service role, select **Choose a service role**, and then select a role from the dropdown.

   b. To create a new service role, select **Create and use a new service role**, and then enter a role name and description.

10. (Optional) To add environment variables for the queue environment, choose **Add new environment variable**, and then enter a name and value for each variable you add.
11. (Optional) Choose **Add new tag** to add one or more tags to your queue.
12. To create a default conda queue environment, keep the checkbox selected. To learn more about queue environments, see Create a queue environment. If you are creating a queue for a customer-managed fleet, clear the checkbox.
13. Choose **Create queue**.

Create a queue environment

A queue environment is a set of environment variables and commands that set up fleet workers. You can use queue environments to provide software applications, environment variables, and other resources to jobs in the queue.

When you create a queue, you have the option of creating a default conda queue environment. This environment provides service-managed fleets access to packages for partner DCC applications and renderers. The default environment For more information, see Default conda queue environment.

You can add queue environments using the console, or by editing the json or YAML template directly. This procedure describes how to create an environment with the console.

1. To add a queue environment to a queue, navigate to the queue and select the **Queue environments tab**.
2. Choose **Actions**, then **Create new with form**.
3. Enter a name and description for the queue environment.
4. Choose **Add new environment variable**, and then enter a name and value for each variable you add.
5. (Optional) Enter a priority for the queue environment. The priority indicates the order that this queue environment will run on the worker. Higher priority queue environments will run first.
6. Choose **Create queue environment**.

Default conda queue environment

When you create a queue associated with a service-managed fleet, you have the option of adding a default queue environment that supports conda to download and install packages in a virtual environment for your jobs.

If you add a default queue environment with the Deadline Cloud console, the environment is created for you. If you add a queue another way, such as the AWS CLI or with CloudFormation, you'll need to create the queue environment yourself. To ensure you have the correct contents for the environment, you can refer to queue environment template YAML files on GitHub. For the contents of the default queue environment, see the default queue environment YAML file on GitHub.

There are other queue environment templates available on GitHub that you can use as a starting point for your own needs.

Conda provides packages from *channels*. A channel is a location where packages are stored. Deadline Cloud provides a channel, `deadline-cloud`, that hosts conda packages that support partner DCC applications and renderers. Select each tab below to view the available packages for Linux or Windows.

Linux

- Autodesk Arnold for Cinema 4D
  - `cinema4d-c4dtoa=2025`
- Autodesk Arnold for Maya
  - `maya-mtoa=2024.5.3`
  - `maya-mtoa=2025.5.4`
  - `maya-mtoa=2026.5.5`
- Autodesk Maya
  - `maya=2024`
  - `maya=2025`
  - `maya=2026`
  - `maya-openjd`
- Autodesk VRED
  - `vredcore=2025`

  - `vredcore=2026`
- Blender
  - `blender=3.6`
  - `blender=4.2`
  - `blender=4.5`
  - `blender=5.0`
  - `blender-openjd`
- Chaos V-Ray for Maya
  - `maya-vray=2025.7`
  - `maya-vray=2026.7`
- Foundry Nuke
  - `nuke=15`
  - `nuke=16`
  - `nuke-openjd`
- Maxon Cinema 4D
  - `cinema4d=2025`
  - `cinema4d=2026`
  - `cinema4d-openjd`
- Maxon Redshift for Maya
  - `maya-redshift=2025.4`
  - `maya-redshift=2026.2`
- SideFX Houdini
  - `houdini=19.5`
  - `houdini=20.0`
  - `houdini=20.5`
  - `houdini=21.0`
  - `houdini-openjd`

Windows

- Adobe After Effects

  - `aftereffects=24.6`
  - `aftereffects=25.1`
  - `aftereffects=25.2`
- Autodesk Arnold for Cinema 4D
  - `cinema4d-c4dtoa=2025`
  - `cinema4d-c4dtoa=2026`
- KeyShot Studio
  - `keyshot=2024`
  - `keyshot=2025`
  - `keyshot-openjd`
- Maxon Cinema 4D
  - `cinema4d=2024`
  - `cinema4d=2025`
  - `cinema4d=2026`
  - `cinema4d-openjd`
- Unreal Engine
  - `unrealengine=5.4`
  - `unrealengine=5.5`
  - `unrealengine=5.6`
  - `unrealengine-openjd`

Note

For **Cinema 4D**, the Linux conda package does not support substance 3D materials. Jobs with this material fail with one of the following errors:

```
Commandline: ./modules/io_substance/source/substance_framework/src/details/
detailsengine.cpp:794:
 SubstanceAir::Details::Engine::Context::Context(SubstanceAir::Details::Engine&,
 SubstanceAir::RenderCallbacks*): Assertion `res==0' failed.
```

```
/home/job-user/.conda/envs/<hash>/Lib/deadline/cinema4d_adaptor/Cinema4DAdaptor/
adaptor.sh: line 44: 10832 Segmentation fault      (core dumped) $C4DEXE
 ${ARGS[*]}
```

We recommend that you submit jobs with substance materials to Windows instead.
In Cinema 4D 2025.3.3 on Linux, globalized asset paths can cause segmentation faults. Therefore, the Linux conda package contains Cinema 4D 2025.3.1 with Redshift 2025.6.0 instead. If you need features or bug fixes from Cinema 4D 2025.3.3, we recommend two options: upgrade to Cinema 4D 2026 or submit those jobs to Windows instead.
For **Cinema 4D OpenJD,** to prevent any timeout issues, we recommend you set task run timeouts to double their expected render time, instead of using the default 2 day timeout.

When you submit a job to a queue with the default conda environment, the environment adds two parameters to the job. These parameters specify the conda packages and channels to use to configure the job's environment before tasks are processed. The parameters are:

- `CondaPackages` – a space-separated list of [package match specifications](), such as `blender=3.6` or `numpy>1.22`. The default is empty to skip creating a virtual environment.
- `CondaChannels` – a space separated list of [conda channels]() such as `deadline-cloud`, `conda-forge`, or `s3://amzn-s3-demo-bucket/conda/channel`. The default is `deadline-cloud`, a channel available to service-managed fleets that provides partner DCC applications and renderers.

When you use an integrated submitter to send a job to Deadline Cloud from your DCC, the submitter populates the value of the `CondaPackages` parameter based on the DCC application and submitter. For example, if you are using Blender the `CondaPackage` parameter is set to `blender=3.6.* blender-openjd=0.4.*`.

We recommend you pin any submissions to only the versions listed in the table above, for example blender=3.6. Pinning to the major.minor version is recommended because patch releases affect the available packages. For example, when we release Blender 3.6.17, we will no longer distribute Blender 3.6.16. Any submissions pinned to blender=3.6.16 will fail. If you pin to blender=3.6, then you will get the latest distributed patch version and jobs will not be impacted. By default, the DCC submitters pin to the current versions listed in the table above, excluding the patch number, such as blender=3.6.

Associate a queue and fleet

To process jobs, you must associate a queue with a fleet. You can associate a single fleet with multiple queues and a single queue with multiple fleets. When you associate a fleet with multiple queues, it divides its workers evenly among them. Similarly, when you associate a queue with multiple fleets, it distributes jobs evenly across those fleets.

> **Note**
>
> To use wait and save, we recommend you associate your queue only with a fleet that uses wait and save instance types. If you associate your queue with more than one fleet, and any of those fleets use spot or on-demand instance types, your fleet might not process your jobs with wait and save instances.

To associate an existing queue with an existing fleet, complete the following steps:

1. From your Deadline Cloud farm, select the **Queue** you want to associate with a fleet. The queue displays.
2. To select a fleet to associate with your queue, choose **Associate fleets**.
3. Choose the **Select fleets** dropdown. A list of available fleets displays.
4. From the list of available fleets, select the **checkbox** next to the fleet or fleets you want to associate with your queue.
5. Choose **Associate**. The fleet association status should now be **Active**.

Stop a queue fleet association

To stop a queue fleet association, complete the following steps:

1. From your queue, select the **Associated fleets** tab.
2. Select the checkbox for the fleet you want to stop associating with the queue.
3. From the Actions dropdown, select **Eventual stop** or **Immediate stop**.

   To finish processing jobs before the association stops, select Eventual stop. To immediately stop processing jobs, select Immediate stop.
4. In the confirmation window, enter **`confirm`** and then choose **Stop**.

5. (Optional) To disassociate the fleet from the queue, complete the following steps:

    a. Wait for the association status to change to **Stopped**.

    b. After the association has stopped, if you haven't already, select the checkbox for the fleet.

    c. From the Actions dropdown, select **Disassociate fleet**.

    d. In the confirmation window, choose **Disassociate**.

Reactivate a queue fleet association

To reactivate a queue fleet association, complete the following steps:

1. From your queue, select the **Associated fleets** tab.
2. Select the checkbox for the fleet you want to reactivate the queue fleet association.
3. From the Actions dropdown, choose **Start**. The association status changes to Active.

Deadline Cloud fleets

This section explains how to manage service-managed fleets and customer-managed fleets (CMF) for Deadline Cloud.

You can set up two types of Deadline Cloud fleets:

- Service-managed fleets are fleets of workers that have default settings provided by Deadline Cloud. These default settings are designed to be efficient and cost effective.
- Customer-managed fleets (CMFs) provide you with full control over your processing pipeline. A CMF can reside within AWS infrastructure, on premises, or in a co-located data center. CMFs include provisioning, operations, management, and decommissioning workers in the fleet.

When you associate a fleet with multiple queues, it divides its workers evenly among those queues.

Topics

Service-managed fleets

A service-managed fleet (SMF) is a fleet of workers that have default settings provided by Deadline Cloud. These default settings are designed to be efficient and cost-effective.

Some of the default settings limit the amount of time that workers and tasks can run. A worker can only run for seven days and a task can only run for five days. When the limit is reached, the task or worker stops. If this happens, you might lose work that worker or task was running. To avoid this, monitor your workers and tasks to ensure they don't exceed the maximum duration limits. To learn more about monitoring your workers, see Using the Deadline Cloud monitor.

Create a service-managed fleet

There are 3 types of instance options you can choose for your service-managed fleet; spot, on-demand, and wait-and-save. Spot instances are unreserved capacity that you can use at a discounted price, but might be interrupted by on-demand requests. On-demand instances are priced by the second, have no long-term commitment, and will not be interrupted. Wait-and-save provides delayed job scheduling for reduced cost and can be interrupted by on-demand and spot requests.

1. From the Deadline Cloud console, navigate to the farm you want to create the fleet in.
2. Select the **Fleets** tab, and then choose **Create fleet**.
3. Enter a **Name** for your fleet.
4. (Optional) Enter a **Description**. A clear description can help you quickly identify your fleet's purpose.
5. Select **Service-managed** fleet type.
6. Choose the **Spot**, **On-demand**, or **Wait and Save** instance market option for your fleet. By default, fleets use the Spot option.
7. For service access for your fleet, select an existing role or create a new role. A service role provides credentials to instances in the fleet, granting them permission to process jobs, and to users in the monitor so that they can read log information.
8. Choose **Next**.
9. Choose between CPU only instances or GPU accelerated instances. GPU accelerated instances may be able to process your jobs faster, but can be more expensive.
10. Select the operating system for your workers. You can leave the default, **Linux** or choose **Windows**.
11. (Optional) If you selected GPU accelerated instances, set the maximum and minimum number of GPUs in each instance. For testing purposes you are limited to one GPU. To request more for your production workloads, see Requesting a quota increase in the *Service Quotas User Guide*.
12. Enter the minimum and maximum **vCPUs** that you require for you fleet.
13. Enter the minimum and maximum **memory** that you require for you fleet.
14. (Optional) You can choose to allow or exclude specific instance types from your fleet to ensure only those instance types are used for this fleet.
15. (Optional) Set the maximum number of instances to scale the fleet so that capacity is available for the jobs in the queue. We recommend that you leave the minimum number of instances at `0` to ensure the fleet releases all instances when no jobs are queued.
16. (Optional) You can specify the size of the Amazon Elastic Block Store (Amazon EBS) gp3 volume that will be attached to the workers in this fleet. For more information, see the EBS user guide.
17. Choose **Next**.

18. (Optional) Define custom worker capabilities that define features of this fleet that can be combined with custom host capabilities specified on job submissions. One example is a particular license type if you plan to connect your fleet to your own license server.
19. Choose **Next**.
20. (Optional) To associate your fleet with a queue, select a **queue** from the dropdown. If the queue is set up with the default conda queue environment, your fleet is automatically provided with packages that support partner DCC applications and renderers. For a list of provided packages, see Default conda queue environment.
21. Choose **Next**.
22. (Optional) To add a tag to your fleet, choose **Add new tag**, and then enter the **key** and **value** for that tag.
23. Choose **Next**.
24. Review your fleet settings, and then choose **Create fleet**.

Use a GPU accelerator

You can configure worker hosts in your service-managed fleets to use one or more GPUs to accelerate processing your jobs. Using an accelerator can reduce the time that it takes to process a job, but can increase the cost of each worker instance. You should test your workloads to understand the trade offs between a fleet using GPU accelerators and fleets that don't.

GPUs are not available for fleets with wait-and-save intances.

> **Note**
>
> For testing purposes you are limited to one GPU. To request more for your production workloads, see Requesting a quota increase in the *Service Quotas User Guide*.

You decide whether your fleet will use GPU accelerators when you specify the worker instance capabilities. If you decide to use GPUs, you can specify the minimum and maximum number of GPUs for each instance, the types of GPU chips to use, and the runtime driver for the GPUs.

The available GPU accelerators are:

- T4 - NVIDIA T4 Tensor Core GPU

- `A10G` - NVIDIA A10G Tensor Core GPU
- `L4` - NVIDIA L4 Tensor Core GPU
- `L40s` - NVIDIA L40S Tensor Core GPU

You can choose from the following runtime drivers:

- `Latest` - Use the latest runtime available for the chip. If you specify `latest` and a new version of the runtime is released, the new version of the runtime is used.
- `grid:r570` - NVIDIA vGPU software 18
- `grid:r550` (deprecated) - NVIDIA vGPU software 17

If you don't specify a runtime, Deadline Cloud uses `latest` as the default. However, if you have multiple accelerators and specify `latest` for some and leave others blank, Deadline Cloud raises an exception.

Software licensing for service-managed fleets

Deadline Cloud provides usage-based licensing (UBL) for commonly used software packages. Supported software packages are automatically licensed when they run on a service-managed fleet. You don't need to configure or maintain a software license server. Licenses scale so you won't run out for larger jobs.

You can install software packages that support UBL using the built-in Deadline Cloud conda channel, or you can use your own packages. For more information about the conda channel, see Create a queue environment.

For a list of supported software packages and information about pricing for UBL, see AWS Deadline Cloud pricing.

Bring your own license with service-managed fleets

With Deadline Cloud usage-based licensing (UBL) you don't need to manage separate licence agreements with software vendors. However, if you have existing licenses or need to use software that isn't available through UBL, you can use your own software licenses with your Deadline Cloud service-managed fleets. You connect your SMF to the software license server via the internet to check out a license for each worker in the fleet.

For an example of connecting to a license server using a proxy, see Connect service-managed fleets to a custom license server in the *Deadline Cloud Developer Guide*.

VFX Reference Platform compatibility

The VFX Reference Platform is a common target platform for the VFX industry. To use the standard service-managed fleet Amazon EC2 instance running Amazon Linux 2023 with software that supports the VFX Reference Platform, you should keep in mind the following considerations when using a service-managed fleet.

The VFX Reference Platform is updated annually. These considerations for using an AL2023 including Deadline Cloud service-managed fleets are based on the calendar year (CY) 2022 through 2024 Reference Platforms. For more information, see VFX Reference Platform.

> **Note**
>
> If you are creating a custom Amazon Machine Image (AMI) for a customer-managed fleet, you can add these requirements when you prepare the Amazon EC2 instance.

To use VFX Reference Platform supported software on an AL2023 Amazon EC2 instance, consider the following:

- The glibc version installed with AL2023 is compatible for runtime use, but not for building software compatible with the VFX Reference Platform CY2024 or earlier.
- Python 3.9 and 3.11 are provided with the service-managed fleet making it compatible with VFX Reference Platform CY2022 and CY2024. Python 3.7 and 3.10 are not provided in the service-managed fleet. Software requiring them must provide the Python installation in the queue or job environment.
- Some Boost library components provided in the service-managed fleet are version 1.75, which is not compatible with the VFX Reference Platform. If your application uses Boost, you must provide your own version of the library for compatibility.
- Intel TBB update 3 is provided in the service-managed fleet. This version is compatible with VFX Reference Platform CY2022, CY2023, and CY2024.
- Other libraries with versions specified by the VFX Reference Platform are not provided by the service-managed fleet. You must provide the library with any application used on a service-managed fleet. For a list of libraries, see the reference platform.

Worker AMI software contents

This section provides information on software installed on AWS Deadline Cloud service-managed worker Amazon Machine Images (AMIs).

AWS Deadline Cloud service-managed worker AMIs are based on both Windows Server 2022 and Amazon Linux 2023, and include additional software specifically installed to support rendering workloads. These AMIs are continuously updated to maintain functionality.

The software on these AMIs is organized into one of the following support categories:

Service-provided software packages

> Software specifically installed and maintained for rendering workloads

Additional system software

> All other software that might change without notice

Service-provided software packages

These software packages are installed to support rendering workloads and are maintained for compatibility. You can safely take dependencies on these packages.

Development Tools & Languages

Linux (AL2023):

- Python 3.11
- Git

Windows (Server 2022):

- Python 3.11
- Git for Windows

AWS tools

Both platforms:

- AWS Command Line Interface v2 (AWS CLI v2)

System libraries & utilities

Linux:

- FUSE and FUSE3 libraries for filesystem operations
- Image Libraries
  - libpng
  - libjpeg
  - libtiff
- OpenGL Libraries
  - mesa-libGLU
  - mesa-libGL
  - mesa-libEGL
  - libglvnd-opengl
- Development Libraries:
  - json-c (JSON parsing)
  - libnsl (network services library)
  - libxcrypt-compat (encryption compatibility)
- X Window Libraries
  - libXmu
  - libXpm
  - libXinerama
  - libXcomposite
  - libXrender
  - libXrandr
  - libXcursor
  - libXi
  - libxdamage
  - libXtst
  - libxkbcommon
  - libSM
- Network and system utilities

  - tcsh

GPU accelerated fleets

- Nvidia grid drivers

Package managers

Linux:

- conda/Mamba package manager (installed in `/opt/conda`)
- DNF package manager (system packages)
- pip (Python package installer)

Windows:

- conda/Mamba package manager (installed in `C:\ProgramData\conda`)
- pip (Python package installer)

Additional system software

All other software on the AMI can be updated, removed, or changed without notice. Do not take dependencies on any software not explicitly listed in the *Supported Software Packages* section above. This restriction includes but is not limited to:

- Operating system packages and libraries
- Service management components
- Base AMI software and drivers
- Software dependencies and runtime libraries
- System configuration tools and utilities

Additional system software examples

Linux: System packages such as systemd, kernel modules, hardware drivers, networking components, and the supporting libraries installed as part of the base AL2023 distribution.

Windows: Windows system components, Microsoft Edge, Amazon EC2 service software, hardware drivers, and Windows runtime components.

Best practices

Dependency management: Only take dependencies on software listed in the *Supported software packages* section.

Package Versions: For specific software versions, install specific packages using package managers (such as pip, conda, and more.) rather than relying on AMI-provided versions.

Environment Isolation: Use virtual environments (such as Python venv, and conda environments) to isolate your specific dependencies.

AMI update model

Note the following information about how the worker AMI updates.

- Worker AMIs are continuously updated with no versioning system.
- Updates occur automatically as part of the service operation.
- No advance notification system is provided for AMI updates.

Customer-managed fleets

When you want to use a fleet of workers that you manage, you can create a customer-managed fleet (CMF) that Deadline Cloud uses to process your jobs. Use a CMF when:

- You have existing on-premises workers to integrate with Deadline Cloud.
- You have workers in a co-located data center.
- You want direct control of Amazon Elastic Compute Cloud (Amazon EC2) workers.

When you use a CMF, you have full control over and responsibility for the fleet. This includes provisioning, operations, management, and decommissioning workers in the fleet.

For more information, see Create and use Deadline Cloud customer-managed fleets in the *Deadline Cloud Developer Guide*.

Managing users in Deadline Cloud

AWS Deadline Cloud uses AWS IAM Identity Center to manage users and groups. IAM Identity Center is a cloud-based single sign-on service that can be integrated with your enterprise single-sign on (SSO) provider. With integration, users can sign in with their company account.

Deadline Cloud enables IAM Identity Center by default, and it is required to set up and use Deadline Cloud. An organization owner for your AWS Organizations is responsible for managing the users and groups that have access to your Deadline Cloud monitor. For more information, see What is AWS Organizations.

How you manage users depends on your IAM Identity Center identity source configuration. The identity source defines where IAM Identity Center gets user information.

Topics

Understanding your identity source

IAM Identity Center uses an identity source to define where users are managed. There are two types of identity sources:

IAM Identity Center directory

> This is the default identity source. Users are created and managed directly within IAM Identity Center. You can create users through the Deadline Cloud console or the IAM Identity Center console. Users receive email invitations to join your organization, and passwords are managed within IAM Identity Center.

External identity provider (IdP)

> Users are federated from an external system such as Okta, Microsoft Entra ID, or other SAML 2.0 identity providers. Users must be created in the external system first. The Deadline Cloud console cannot create users when an external IdP is configured, but you can assign permissions to existing users. Passwords are managed by the external IdP.

To check your identity source configuration or change it, see Manage your identity source in the IAM Identity Center User Guide.

Create and manage users with IAM Identity Center directory

If your identity source is set to IAM Identity Center directory, you can create and manage users and groups directly through the Deadline Cloud console. Users created in the console will receive email invitations from IAM Identity Center. After accepting the invitation, users can access the Deadline Cloud monitor.

> **Note**
>
> If your IAM Identity Center is connected to an external identity provider, you cannot create users through the Deadline Cloud console. See the section called “Manage users with external IdP” for information about managing users with an external IdP.

1. Sign in to the AWS Management Console and open the Deadline Cloud console. From the main page, in the **Get started** section, choose **Set up Deadline Cloud** or **Go to dashboard**.
2. In the left navigation pane, choose **User management**. By default, the **Groups** tab is selected.

Depending on the action to take, choose either the **Groups** tab or **Users** tab.

Groups

To create a group

1. Choose **Create group**.
2. Enter a group name. The name must be unique among groups in your IAM Identity Center organization.

To remove a group

1. Select the group to remove.
2. Choose **Remove**.
3. In the confirmation dialog, choose **Remove group**.

> **Note**
>
> You are removing the group from IAM Identity Center. Group members can no longer sign in to the Deadline Cloud or access farm resources.

Users

To add users

1. Choose the **Users** tab.
2. Choose **Add users**.
3. Enter the name, email address, and username for the new user.
4. (Optional) Choose one or more IAM Identity Center groups to add the new user to.
5. Choose **Send invite** to send the new user an email with instructions for joining your IAM Identity Center organization.

To remove a user

1. Select the user you to remove.
2. Choose **Remove**.
3. In the confirmation dialog, choose **Remove user**.

> **Note**
>
> You are removing the user from IAM Identity Center. The user can no longer sign in to the Deadline Cloud monitor or access farm resources.

Manage users with an external identity provider

If your IAM Identity Center is connected to an external identity provider (IdP) such as Okta or Microsoft Entra ID, users must be created and managed in that external system. The Deadline Cloud console cannot create new users when an external IdP is configured.

After users are created in your external IdP and synchronized to IAM Identity Center, you can assign them permissions to Deadline Cloud resources. See the section called “Understanding access levels” for information about assigning permissions at the farm, queue, and fleet level.

For information about managing your external identity provider configuration, see Manage your identity source in the IAM Identity Center User Guide.

Understanding access levels

Regardless of your identity source, you assign permissions to users and groups at the farm, queue, and fleet level through the Deadline Cloud console. You can grant access permissions at different levels. Each subsequent level includes the permissions for the previous levels. The following list describes the four access levels from the lowest level to the highest level:

- **Viewer** – Permission to see resources in the farms, queues, fleets, and jobs they have access to. A viewer can't submit or make changes to jobs.
- **Contributor** – Same as a viewer, but with permission to submit jobs to a queue or farm.
- **Manager** – Same as contributor, but with permission to edit jobs in queues they have access to, and grant permissions on resources that they have access to.
- **Owner** – Same as manager, but can view and create budgets and see usage.

For information about customizing these access levels, see Monitor role in the *Deadline Cloud Developer Guide*.

Topics

Access level permissions matrix

The following tables show the specific permissions available at each access level for farms, queues, and fleets when using the default AWS managed policies. Managing user access is currently only available through the Deadline Cloud console and not available in the Deadline Cloud monitor. For information about customizing these access levels, see Monitor role in the *Deadline Cloud Developer Guide*.

Farm permissions by access level

Permission	Viewer	Contributor	Manager	Owner
View farm details	Yes	Yes	Yes	Yes
View queues and fleets	Yes	Yes	Yes	Yes
Submit jobs	No	Yes	Yes	Yes
Manage user access	No	No	Yes	Yes
View and create budgets	No	No	No	Yes
View usage data	No	No	No	Yes

Queue permissions by access level

Permission	Viewer	Contributor	Manager	Owner
View queue details	Yes	Yes	Yes	Yes
View jobs in queue	Yes	Yes	Yes	Yes
Submit jobs to queue	No	Yes	Yes	Yes
Edit and cancel jobs	No	No	Yes	Yes
Manage queue user access	No	No	Yes	Yes
View queue budget allocation	No	No	No	Yes

Fleet permissions by access level

Permission	Viewer	Contributor	Manager	Owner
View fleet details	Yes	Yes	Yes	Yes
View workers in fleet	Yes	Yes	Yes	Yes
Manage fleet user access	No	No	Yes	Yes

Permission	Viewer	Contributor	Manager	Owner
View fleet cost data	No	No	No	Yes

Membership inheritance

Deadline Cloud uses a hierarchical membership model where permissions can be assigned at the farm, queue, or fleet level. Understanding how membership inheritance works helps you configure access control effectively.

Farm-level membership

When you assign a user or group membership at the farm level, that membership applies to all queues and fleets within the farm. Farm-level membership provides broad access and is useful for users who need to work across multiple queues or fleets.

For example, if you assign a user as a Contributor at the farm level, that user can submit jobs to any queue in the farm.

Queue and fleet-level membership

You can also assign membership at the queue or fleet level for more granular access control. Queue-level and fleet-level membership only applies to that specific resource.

For example, if you assign a user as a Manager on a specific queue, that user can edit jobs and manage access only for that queue, not for other queues in the farm.

Users can have access to only a queue or fleet without having farm-level membership. In this case, the user cannot see the farm in their farm list, but can submit jobs to and view only the queues or fleets they have access to.

Effective permissions

When a user has membership at multiple levels, Deadline Cloud uses the highest access level. For example:

- A user with Viewer access at the farm level and Manager access on a specific queue has Manager permissions on that queue and Viewer permissions on all other queues.
- A user with Contributor access at the farm level and Owner access on a specific fleet has Owner permissions on that fleet and Contributor permissions elsewhere.

> **Note**
>
> Users without any membership at the farm, queue, or fleet level cannot access those resources, even if they are authenticated through IAM Identity Center.

For instructions on assigning membership to users and groups, see the section called “Assign permissions”.

Assign permissions to users and groups

Use the Deadline Cloud console to assign access levels to users and groups at the farm, queue, or fleet level.

> **Note**
>
> Changes to access permissions might take up to 10 minutes to reflect in the system.

To navigate to access management

1. Sign in to the AWS Management Console and open the Deadline Cloud console.
2. In the left navigation pane, choose **Farms and other resources**.
3. Select the farm to manage. Choose the farm name to open the details page. You can search for the farm using the search bar.
4. (Optional) To manage a queue or fleet instead of the farm, choose the **Queues** or **Fleets** tab, and then choose the queue or fleet to manage.
5. Choose the **Access management** tab.

Depending on the action to take, choose either the **Groups** tab or **Users** tab.

Groups

To add groups

1. Select the **Groups** toggle.
2. Choose **Add group**.

3. From the dropdown, select the groups to add.
4. For the group access level, choose one of the following options:
    - **Viewer**
    - **Contributor**
    - **Manager**
    - **Owner**
5. Choose **Add**.

To remove groups

1. Select the groups to remove.
2. Choose **Remove**.
3. In the confirmation dialog, choose **Remove group**.

Users

To add users

1. To add a user, choose **Add user**.
2. From the dropdown, select the users to add.
3. For the user access level, choose one of the following options:
    - **Viewer**
    - **Contributor**
    - **Manager**
    - **Owner**
4. Choose **Add**.

To remove users

1. Select the user to remove.
2. Choose **Remove**.
3. In the confirmation dialog, choose **Remove user**.

Deadline Cloud jobs

A *job* is a set of instructions that AWS Deadline Cloud uses to schedule and run work on available workers. When you create a job, you choose the farm and queue to send the job to.

A *submitter* is a plugin for your digital content creation (DCC) application that manages creating a job in the interface of your DCC application. After you create the job, you use the submitter send it to Deadline Cloud for processing.

The submitter creates an Open Job Specification (OpenJD) template that describes the job. At the same time it uploads your asset files to an Amazon Simple Storage Service (Amazon S3) bucket. To reduce upload time, the submitter only sends files that have changed since the last upload to Amazon S3

You can also create a job in the following ways.

- From a terminal – for users submitting a job that are comfortable using the command line.
- From a script – for customizing and automating workloads.
- From an application – for when the user's work is in an application, or when an application's context is important.

For more information, see How to submit a job to Deadline Cloud in the *Deadline Cloud Developer Guide*.

A job consists of:

- *Priority* – The approximate order that Deadline Cloud processes a job in a queue. You can set the job priority between 0 and 100, jobs with a higher number priority are generally processed first. Jobs with the same priority are processed in the order received.
- *Steps* – Defines the script to run on workers. Steps can have requirements such as minimum worker memory or other steps that need to complete first. Each step has one or more tasks.
- *Tasks* – A unit of work sent to a worker to perform. A task is a combination of a step's script and parameters, such as a frame number, that are used in the script. The job is complete when all tasks are complete for all steps.
- *Environment* – Set up and tear down instructions shared by multiple steps or tasks.

Using a Deadline Cloud submitter

A *submitter* is a tool that integrates with your digital content creation so that you can send render jobs directly to Deadline Cloud. This integration streamlines your workflow by eliminating the need to switch between applications or manually transfer files. This saves time and reduces the potential for errors.

Submitters are available for many popular DCC applications. Installing a submitter, adds Deadline Cloud specific options to your application's interface, typically in the render settings or export menu.

With a Deadline Cloud submitter you can:

- Configure render job parameters in your familiar DCC environment
- Submit jobs to Deadline Cloud without leaving your application
- Reduce the potential for errors associated with manual file transfers
- Save time because you don't need to switch between applications

To find a submitter for your DCC application, check the Set up your workstation page. Then follow the instructions in Set up your workstation to install the submitter.

If your application doesn't have a supported submitter, you can still run jobs for your application. There may be a sample job bundle available for it, or you can construct a simple submitter for the application's render CLI command. For more information, see Open Job Description (OpenJD) templates for Deadline Cloud in the *Deadline Cloud Developor Guide*.

The examples in this topic use the Blender submitter, but the steps for using other submitters are similar.

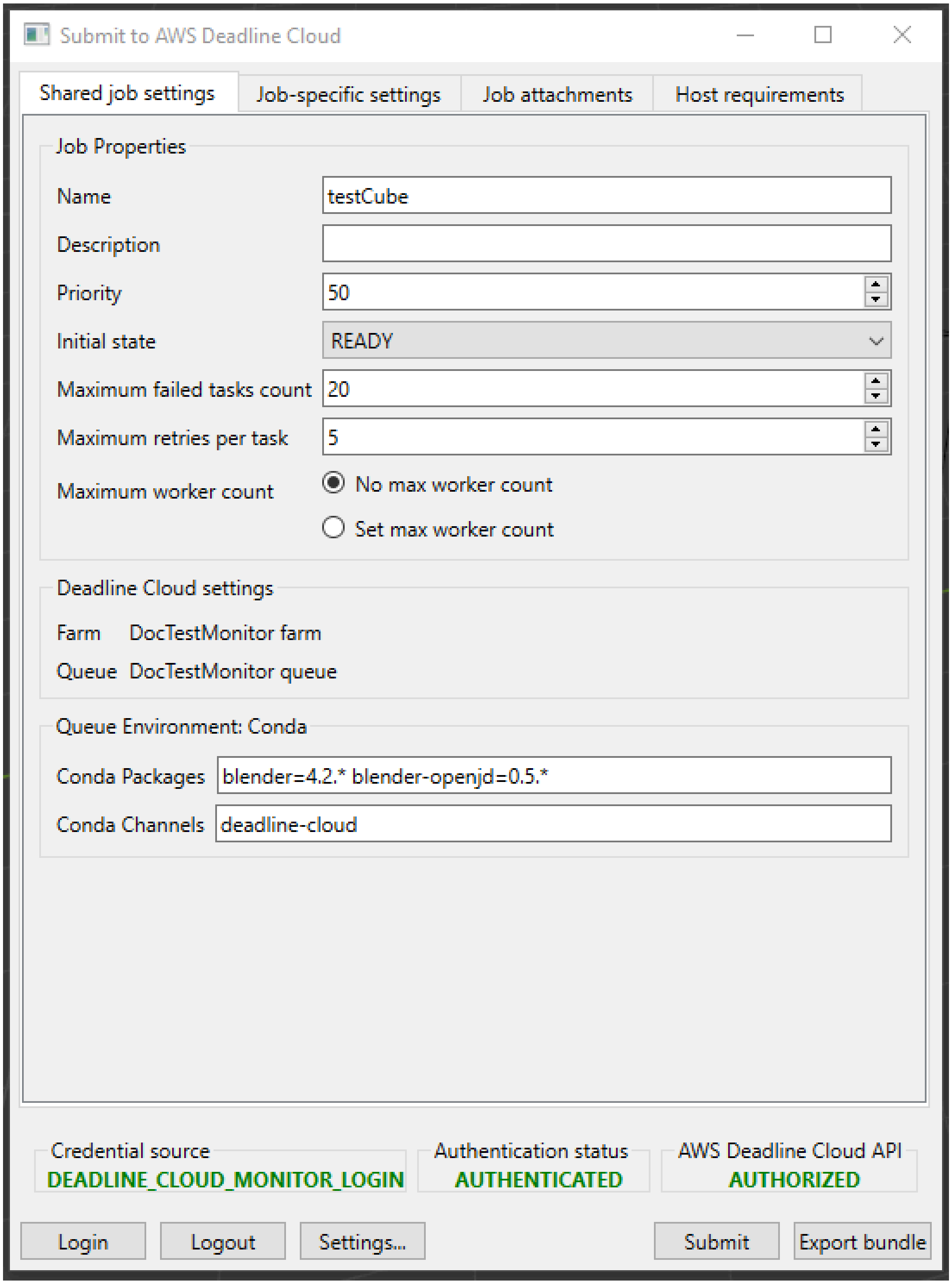
> **Note**
>
> To use a submitter, you must be signed in to the Deadline Cloud monitor.

The submitter has four tabs:

Topics

- Shared job settings tab
- Job-specific settings tab
- Job attachments tab
- Host requirements tab

Shared job settings tab



Job Properties

Name: testCube

Description:

Priority: 50

Initial state: READY

Maximum failed tasks count: 20

Maximum retries per task: 5

Maximum worker count: ● No max worker count ○ Set max worker count

Deadline Cloud settings

Farm: DocTestMonitor farm

Queue: DocTestMonitor queue

Queue Environment: Conda

Conda Packages: blender=4.2.* blender-openjd=0.5.*

Conda Channels: deadline-cloud

Credential source: DEADLINE_CLOUD_MONITOR_LOGIN

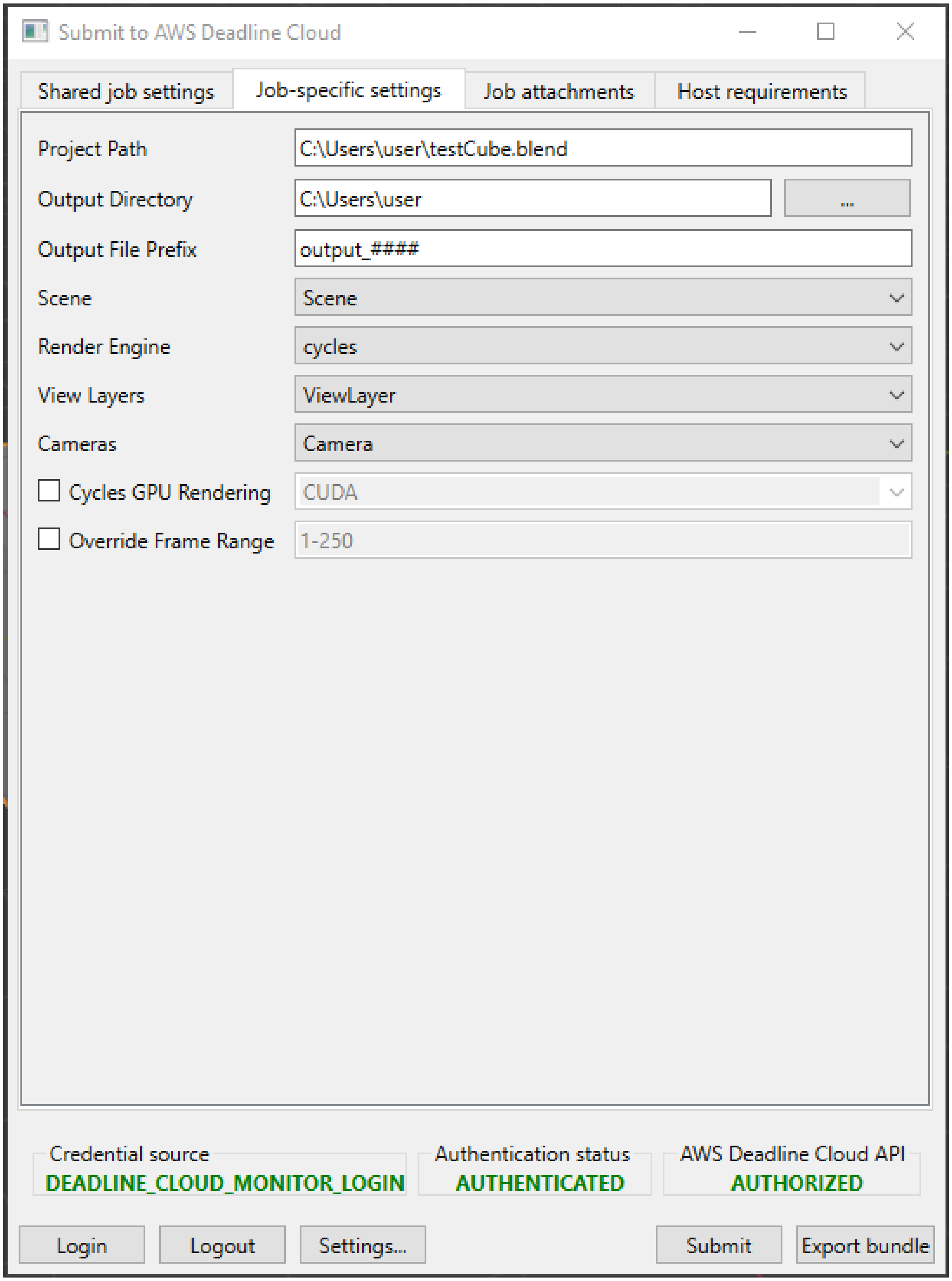
Authentication status: AUTHENTICATED

AWS Deadline Cloud API: AUTHORIZED

The shared job settings tab contains the settings that are common to all jobs sent to Deadline Cloud using the submitter. The three sections are:

- *Job properties* – Sets the overall properties of the job. These properties are present in submitters for all DCC applications.
- *Deadline Cloud settings* – Shows the farm and queue that the job is sent to. To change the farm and queue, use the **Settings...** button at the bottom of the submitter.
- *Queue environment* – Sets the parameter values defined in the queue environment. Deadline Cloud adds the default parameter values for your DCC application, you can add additional values if necessary.

Job-specific settings tab



Project Path	C:\Users\user\testCube.blend
Output Directory	C:\Users\user
Output File Prefix	output_####
Scene	Scene
Render Engine	cycles
View Layers	ViewLayer
Cameras	Camera
□ Cycles GPU Rendering	CUDA
□ Override Frame Range	1-250

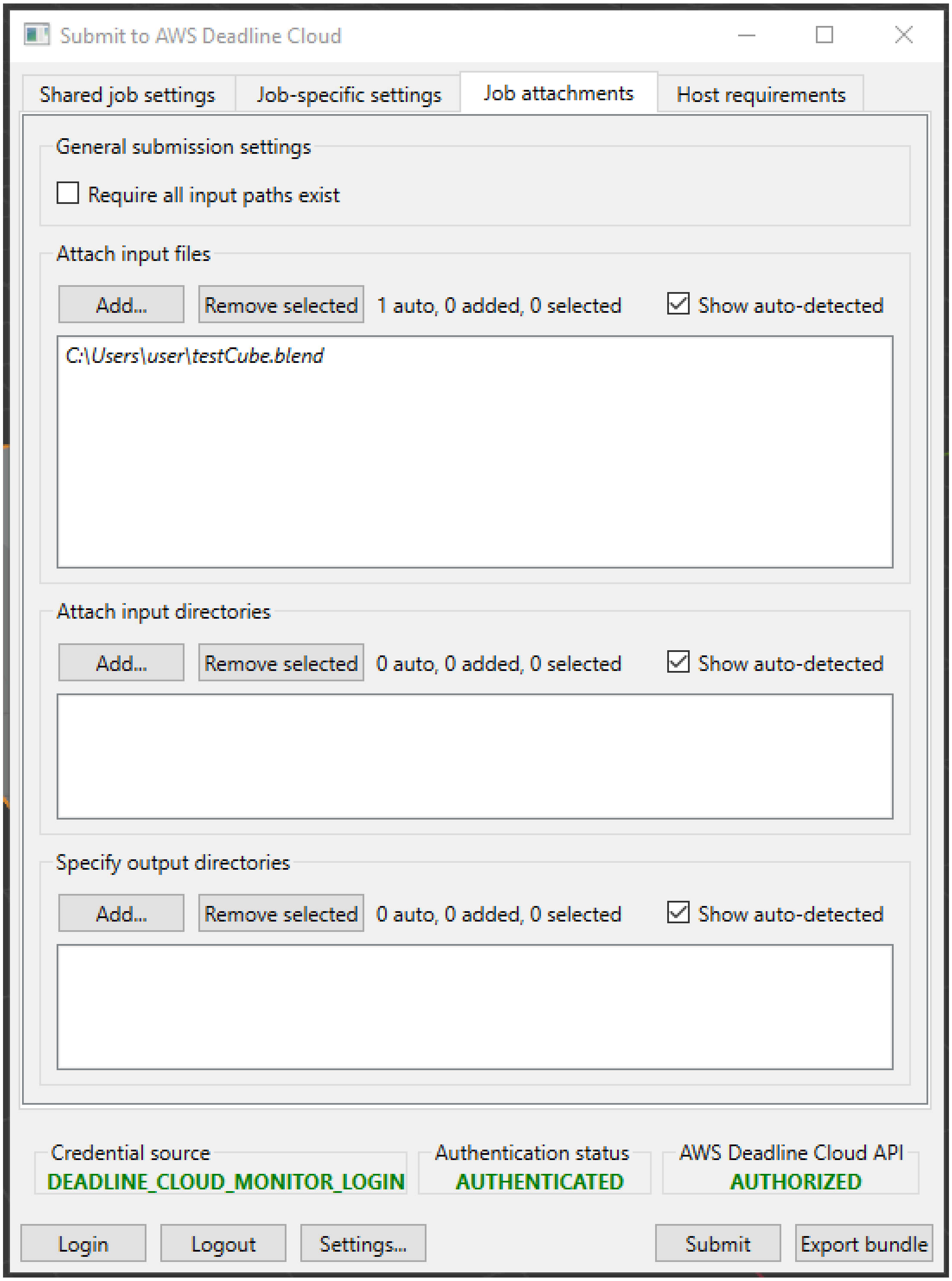
Credential source
DEADLINE_CLOUD_MONITOR_LOGIN

Authentication status
AUTHENTICATED

AWS Deadline Cloud API
AUTHORIZED

The job-specific settings tab contains the setting specific to your DCC application. Specify these settings based on the options available in your application.

Job attachments tab



General submission settings

☐ Require all input paths exist

Attach input files

1 auto, 0 added, 0 selected ☑ Show auto-detected

C:\Users\user\testCube.blend

Attach input directories

0 auto, 0 added, 0 selected ☑ Show auto-detected

Specify output directories

0 auto, 0 added, 0 selected ☑ Show auto-detected

Credential source: DEADLINE_CLOUD_MONITOR_LOGIN

Authentication status: AUTHENTICATED

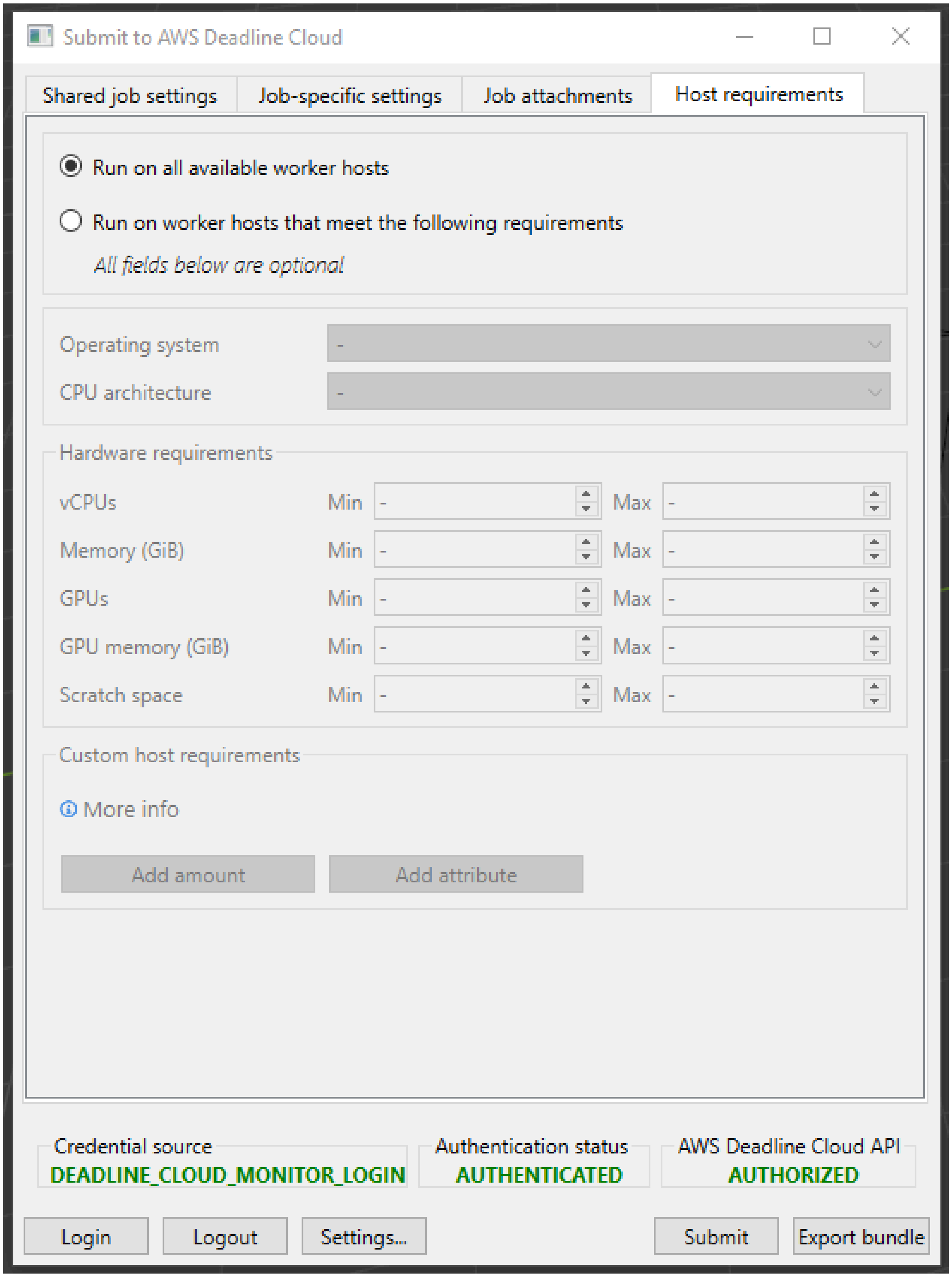
AWS Deadline Cloud API: AUTHORIZED

The job attachments tab shows all of the files needed to complete a render. The submitter tries to find all of the files required for the render. The files that it identifies appear in the lists in italics.

You can add additional input files and directories that contain other assets required for the render that were not automatically detected.

If your job writes files to multiple output directories, you must specify the directories here so that the are part of the job download.

Host requirements tab



● Run on all available worker hosts

○ Run on worker hosts that meet the following requirements

All fields below are optional

Operating system: -

CPU architecture: -

Hardware requirements

vCPUs	Min	-	Max	-
Memory (GiB)	Min	-	Max	-
GPUs	Min	-	Max	-
GPU memory (GiB)	Min	-	Max	-
Scratch space	Min	-	Max	-

Custom host requirements

Credential source: DEADLINE_CLOUD_MONITOR_LOGIN

Authentication status: AUTHENTICATED

AWS Deadline Cloud API: AUTHORIZED

The host requirements tabs sets the fleet capabilities required to process the job. Capabilities are specified for the entire fleet, not individual workers in the fleet.

If your queue has associated resource limits, use the **Add amount** button to specify the limit. For more information, see Create resource limits for jobs

Processing Deadline Cloud jobs

When a job enters a queue, Deadline Cloud schedules it on one or more fleets associated with the queues. The fleet is chosen based on the capabilities configured for the fleet and the host requirements of a specific step. If a job has a requirement that can't be met by a any of the fleets associated with the queue, the job's status is set to "Not compatible" and the rest of the steps in the job are canceled.

Next, Deadline Cloud sends instructions to the workers to set up a session for the step. The software required for the step must be available on the worker instance for the job to run. The service opens sessions on multiple workers if the fleets scaling settings allow.

You can set up the software in an Amazon Machine Image (AMI), or your worker can load the software at runtime from a repository or package manager. You can use queue, job, or step environments to deploy the software that you prefer.

The Deadline Cloud service uses the OpenJD template to identify the steps required for the job, and the tasks required for each step. Some steps have dependencies on other steps, so Deadline Cloud determines the order to complete the steps. Then, Deadline Cloud sends the tasks for each step to workers to process. When a task is finished, the service sends another task in the same session, or the worker can start a new session.

After all tasks in each step are finished, the job is complete and the output is ready to download to your workstation. Even if the job didn't finish, the output from each step and task that finished is available to download.

> **Note**
>
> Deadline Cloud removes jobs 120 days after they were submitted. When a job is removed, all of the steps and tasks associated with the job are also removed. If you need to re-run the job, submit the OpenJD template for the job again.

Monitoring Deadline Cloud jobs

The AWS Deadline Cloud monitor provides you with an overall view of your jobs. Use it to:

- Monitor and manage jobs
- View worker activity on fleets
- Track budgets and usage
- Download a job's results.

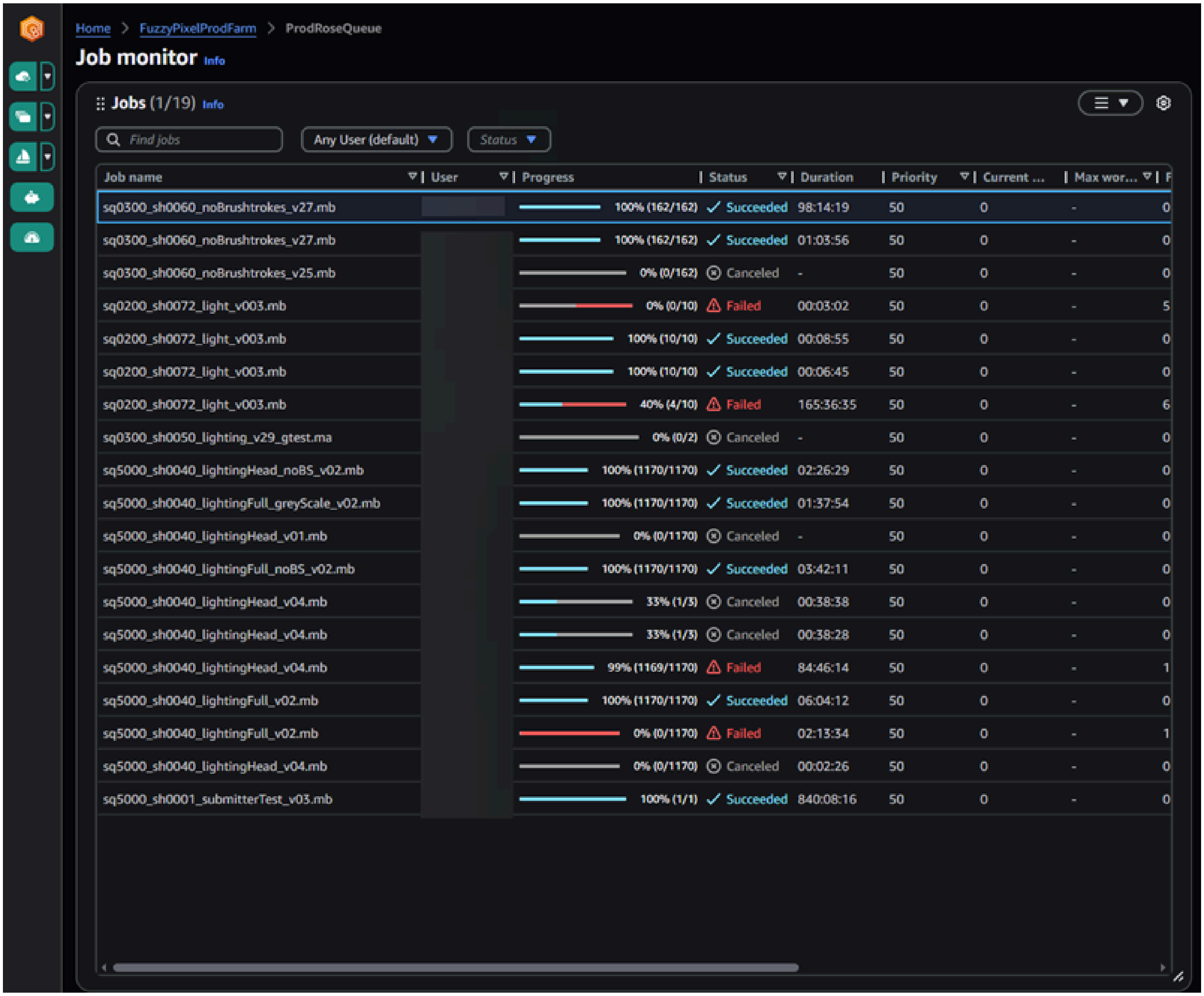
To monitor a specific job, select the farm and queue containing the job, then select the job from the list. You can use the search box to locate a specific job or jobs in the queue.

Right click on a job, step, or task to see the options for the item. You can:

- Change the status
- Suspend and resume the item
- Requeue the item
- Download the output
- For jobs: Modify job properties like the name, description, priority, or max worker count.
- For tasks: View task and worker logs.

For more information, see Using the Deadline Cloud monitor.

Each task in a job or step has a status. The status of a job or step depends on the status of its tasks. The status is determined by tasks that have these statuses, in order. Step statuses are determined the same as the job status.



Job monitor Info

Jobs (1/19) Info

Job name	User	Progress	Status	Duration	Priority	Current ...	Max wor...	F
sq0300_sh0060_noBrushtrokes_v27.mb	[redacted]	100% (162/162)	✓ Succeeded	98:14:19	50	0	-	0
sq0300_sh0060_noBrushtrokes_v27.mb	[redacted]	100% (162/162)	✓ Succeeded	01:03:56	50	0	-	0
sq0300_sh0060_noBrushtrokes_v25.mb	[redacted]	0% (0/162)	⊗ Canceled	-	50	0	-	0
sq0200_sh0072_light_v003.mb	[redacted]	0% (0/10)	⚠ Failed	00:03:02	50	0	-	5
sq0200_sh0072_light_v003.mb	[redacted]	100% (10/10)	✓ Succeeded	00:08:55	50	0	-	0
sq0200_sh0072_light_v003.mb	[redacted]	100% (10/10)	✓ Succeeded	00:06:45	50	0	-	0
sq0200_sh0072_light_v003.mb	[redacted]	40% (4/10)	⚠ Failed	165:36:35	50	0	-	6
sq0300_sh0050_lighting_v29_gtest.ma	[redacted]	0% (0/2)	⊗ Canceled	-	50	0	-	0
sq5000_sh0040_lightingHead_noBS_v02.mb	[redacted]	100% (1170/1170)	✓ Succeeded	02:26:29	50	0	-	0
sq5000_sh0040_lightingFull_greyScale_v02.mb	[redacted]	100% (1170/1170)	✓ Succeeded	01:37:54	50	0	-	0
sq5000_sh0040_lightingHead_v01.mb	[redacted]	0% (0/1170)	⊗ Canceled	-	50	0	-	0
sq5000_sh0040_lightingFull_noBS_v02.mb	[redacted]	100% (1170/1170)	✓ Succeeded	03:42:11	50	0	-	0
sq5000_sh0040_lightingHead_v04.mb	[redacted]	33% (1/3)	⊗ Canceled	00:38:38	50	0	-	0
sq5000_sh0040_lightingHead_v04.mb	[redacted]	33% (1/3)	⊗ Canceled	00:38:28	50	0	-	0
sq5000_sh0040_lightingHead_v04.mb	[redacted]	99% (1169/1170)	⚠ Failed	84:46:14	50	0	-	1
sq5000_sh0040_lightingFull_v02.mb	[redacted]	100% (1170/1170)	✓ Succeeded	06:04:12	50	0	-	0
sq5000_sh0040_lightingFull_v02.mb	[redacted]	0% (0/1170)	⚠ Failed	02:13:34	50	0	-	1
sq5000_sh0040_lightingHead_v04.mb	[redacted]	0% (0/1170)	⊗ Canceled	00:02:26	50	0	-	0
sq5000_sh0001_submitterTest_v03.mb	[redacted]	100% (1/1)	✓ Succeeded	840:08:16	50	0	-	0

The following list describes the statuses:

`NOT_COMPATIBLE`

The job is not compatible with the farm because there are no fleets that can complete one of the tasks in the job.

`RUNNING`

One or more workers are running tasks from the job. As long as there is at least one running task, the job is marked `RUNNING`.

`ASSIGNED`

One or more workers are assigned tasks in the job as their next action. The environment, if any, is set up.

`STARTING`

One or more workers is setting up the environment for running tasks.

`SCHEDULED`

Tasks for the job are scheduled on one or more workers as the worker's next action.

`READY`

At least one task for the job is ready to be processed.

`INTERRUPTING`

At least one task in the job is being interrupted. Interruptions can happen when you manually update the job's status. It can also happen in response to an interruption due to Amazon Elastic Compute Cloud (Amazon EC2) Spot price changes.

`FAILED`

One or more tasks in the job didn't complete successfully.

`CANCELED`

One or more tasks in the job have been canceled.

`SUSPENDED`

At least one task in the job has been suspended.

`PENDING`

A task in the job is waiting on the availability of another resource.

`SUCCEEDED`

All tasks in the job were successfully processed.

Supported Software

Deadline Cloud supports a wide range of digital content creation applications for 3D rendering, animation, visual effects, and compositing. Supported applications always include integrated submitters but may also support conda packages, host configuration scripts, Usage-based licensing and more. The applications listed below receive official support from Deadline Cloud. For customization options beyond the officially supported configurations, see Provide applications for your jobs and Create a conda package for an application or plugin in the *Deadline Cloud Developer Guide*.

The following DCC applications are supported by Deadline Cloud:

Topics

Adobe After Effects

> **Note**
>
> For more information about installing, configuring, and using this integration on your workstation, see the After Effects integration user guide on GitHub.

Adobe After Effects is a professional digital visual effects, motion graphics, and compositing application. After Effects is fully supported by Deadline Cloud with comprehensive integration including submitters and conda packages for increased rendering performance.

Support overview

After Effects is supported by the following components:

- **Submitter**: Integrated submitter for direct job submission from After Effects with automatic scene and asset detection.
- **Conda packages**: Deadline Cloud for automatic installation on service-managed fleets.
- **Cross-platform compatibility**: Submitter support for Windows and macOS with worker support for Windows.

After Effects version compatibility

The following table shows current support levels for After Effects versions:

Major Version	Submitter Support	Conda Support
2024	Windows, macOS	Windows
2025	Windows, macOS	Windows

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to After Effects available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version
Windows	aftereffects	24.6
Windows	aftereffects	25.1
Windows	aftereffects	25.2

Getting started

Complete the following steps to set up After Effects with Deadline Cloud. You will install the required submitter and monitor on your workstation and begin submitting render jobs to your queue.

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor on your artist workstation using the Deadline Cloud monitor Installers. For more information, see Set up your workstation.
3. Install the Deadline Cloud After Effects submitter on your artist workstation using the Deadline Cloud Submitter Installers. When you install the submitter, you can choose between User Install (no admin required) or System Install (Windows only, requires admin). macOS users must use User Install.
   - **User Install**: Installs to user directory without admin privileges. The submitter will be a standalone window rather than a dockable panel.
     - Windows: `C:\Users\<user>\DeadlineCloudSubmitter\Submitters\AfterEffects\AE<version>`
     - macOS: `/Users/<user>/DeadlineCloudSubmitter/Submitters/AfterEffects/AE<version>`
   - **System Install** (Windows only): Installs to Adobe After Effects installation directory as a dockable panel.
     - Windows: `C:\Program Files\Adobe\Adobe After Effects <version>\Support Files\Scripts\Script UI Panels`

Using the After Effects submitter

Launching the submitter

To launch the After Effects submitter

1. Launch Adobe After Effects.
2. Update the following settings within After Effects to allow scripts to write files and send communication over a network:

- For Windows, choose **Edit** > **Preferences** > **Scripting & Expressions**, and then choose **Allow scripts to write files and access networks**.
- For macOS, choose **After Effects** > **Settings** > **Scripting & Expressions**, and then choose **Allow scripts to write files and access networks**.

3. Restart After Effects.
4. Open the Deadline Cloud submitter based on your install type:

   - For a system install, select **Window**, then choose **DeadlineCloudSubmitter.jsx**.
   - For a user install, choose **File** > **Scripts** > **Run Script File**, and then locate and select **DeadlineCloudSubmitter.jsx** .

5. (Optional) If the submitter is closed and you used a user install, reopen it by choosing **File** > **Scripts** > **Recent Script Files** and selecting **DeadlineCloudSubmitter.jsx**.

Submitting a render job

To submit a render job from After Effects

1. Choose **Open Render Queue** on the submitter.
2. Add a composition to your render queue and set up your render settings, output module, and output path.
3. Choose **Refresh** on the submitter to see your composition in the composition list.
4. Select the composition to render and choose **Submit** to submit a render job.
5. If you see a warning about running a script file, suppress the warning messages by following the instructions in the popup.
6. Install any Python libraries if prompted.
7. Choose **Submit** to send your job to Deadline Cloud.
8. Monitor the job and download the output using the Deadline Cloud monitor.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of After Effects does not appear in the version table above.

If you require an unsupported version of After Effects, you have the following options:

- When submitting the job from After Effects, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `aftereffects=2025`). This may or may not work, depending on the features used by your scene and how After Effects works with scenes from your workstation version.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked below as a starting point, and package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

Open source resources

The submitter is open source and available on GitHub:

- Deadline Cloud for After Effects
- Standalone After Effects job bundle is available on GitHub.
- Comprehensive user guide is available.

Autodesk 3ds Max

> **Note**
>
> For more information about installing, configuring, and using this integration on your workstation, see the Autodesk 3ds Max integration user guide on GitHub.

> **Note**
>
> When using Autodesk 3ds Max with AWS Deadline Cloud, you can use Autodesk cloud rights included with your subscription. For more information about cloud rights and subscription benefits, see Subscription Benefits FAQ: Cloud Rights on the Autodesk website.

Autodesk 3ds Max is a professional 3D computer graphics program for creating 3D animations, models, games, and images. Deadline Cloud provides comprehensive support for 3ds Max with integrated submitters, host configuration scripts, usage-based licensing, and adaptors for increased rendering performance.

Support overview

3ds Max is supported by the following components:

- **Submitter**: Integrated submitter for direct job submission from 3ds Max with automatic scene and asset detection.
- **Host Configuration Script**: Example host configuration script to install 3ds Max.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows with worker support for Windows and automatic path mapping.
- **Usage-based Licensing**: Pay-as-you-go licensing for 3ds Max and Corona.

3ds Max version compatibility

The following table shows current support levels for 3ds Max versions:

Major Version	Submitter Support	Host Configuration Support
2024	Windows	Windows
2025	Windows	Windows
2026	Windows	Windows

3ds Max differences from other digital content creation tools

In Deadline Cloud, 3ds Max is installed using host configuration scripts instead of conda packages. This differs from most other DCCs in Deadline Cloud due to unique requirements of the 3ds Max installation process, as the application must be installed by a system administrator.

Getting started

To use 3ds Max with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Configure the fleet with GPU support if you intend to use GPU-accelerated rendering features. The fleet must be configured with a host configuration script that installs 3ds Max. For more information, see 3ds Max Host Configuration script setup and the 3ds Max Host Config example on GitHub.
2. Install the Deadline Cloud monitor and 3ds Max submitter on your artist workstation using the Deadline Cloud Submitter and monitor Installers. For more information, see Set up your workstation.
3. Submit your job directly from 3ds Max using the integrated submitter to the queue.
4. Monitor the job and download the output using the Deadline Cloud monitor.

For more information about using the 3ds Max integrated submitter, see the 3ds Max integration user guide on GitHub.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. You must ensure the version of 3ds Max used by the artist is compatible with the version of 3ds Max configured in your fleet's host configuration.

Support for older 3ds Max versions is possible via host configuration scripts. However, the integrated submitter may not function due to older Python versions. In such cases, custom job bundles can still be submitted as Deadline Cloud jobs.

3ds Max renderers

Deadline Cloud supports rendering 3ds Max jobs using the following renderers when using a host configuration script that includes them:

Renderer	Renderer Version	Host Configuration Script Provided	Usage-based Licensing Support
Autodesk Scanline	Built-in	N/A	N/A

Renderer	Renderer Version	Host Configuration Script Provided	Usage-based Licensing Support
Autodesk Raytracer (ART)	Built-in	N/A	N/A
Chaos V-Ray 6	6.x	Yes	Yes
Chaos V-Ray 7	7.x	Yes	Yes
Corona	Latest	Yes	No

Open source resources

The submitter and adaptor are open source and available on GitHub:

- 3ds Max Submitter and Adaptor
- Deadline Cloud Samples (for 3ds Max workflow examples)
- 3ds Max Host Config example

Autodesk Maya

Note

For more information about installing, configuring, and using this integration on your workstation, see the Maya integration user guide on GitHub.

Autodesk Maya is a 3D computer animation, modeling, simulation, and rendering software used for creating interactive 3D applications, including video games, animated films, TV series, and visual effects. Maya is fully supported by Deadline Cloud with comprehensive integration including submitters, conda packages, usage-based licensing, and an adaptor for increased rendering performance.

Support overview

Maya is supported by the following components:

- **Submitter**: Integrated plug-in for direct job submission from Maya.
- **Conda packages**: Automatic installation on service-managed fleets when using the submitter.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows, macOS, and Linux with worker support for Windows and Linux.
- **Usage-based Licensing**: Pay-as-you-go for Maya and renderer licensing.

Maya version compatibility

The following table shows current support levels for Maya versions:

Major Version	Submitter Support	Conda Support	Render Engines	Usage-Based Licensing
2024	Windows, macOS, Linux	Linux	Maya Software, Arnold (MtoA)	Usage-based licensing available
2025	Windows, macOS, Linux	Linux	Maya Software, Arnold (MtoA), V-Ray, Redshift	Usage-based licensing available
2026	Windows, macOS, Linux	Linux	Maya Software, Arnold (MtoA), V-Ray, Redshift	Usage-based licensing available

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Maya available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Linux	maya	2024	Includes Maya Software renderer

OS	Package	Version	Notes
Linux	maya	2025	Includes Maya Software renderer
Linux	maya	2026	Includes Maya Software renderer
Linux	maya-mtoa	2024.5.3	Arnold for Maya 2024
Linux	maya-mtoa	2025.5.4	Arnold for Maya 2025
Linux	maya-mtoa	2026.5.5	Arnold for Maya 2026
Linux	maya-openjd		Includes the Maya Adaptor
Linux	maya-redshift	2025.4	Redshift for Maya 2025
Linux	maya-redshift	2026.2.1	Redshift for Maya 2026
Linux	maya-vray	2025.7	V-Ray for Maya 2025
Linux	maya-vray	2026.7	V-Ray for Maya 2026

Getting started

To use Maya with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor and Maya submitter on your artist workstation using the Deadline Cloud Submitter and monitor Installers. For more information, see Set up your workstation.
3. Submit your job directly from Maya using the integrated submitter to the queue.

4. Monitor the job and download the output using the Deadline Cloud monitor.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of Maya does not appear in the version table above.

If you require an unsupported version of Maya, you have the following options:

- When submitting the job from Maya, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `maya=2026, maya-openjd=*`). This may or may not work, depending on the features used by your scene and how Maya works with scenes from your workstation version.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipes for supported versions as a starting point:
  - Maya conda recipe
  - Maya OpenJD adaptor conda recipe

  For more information about creating custom conda channels, see Creating custom conda channels.

Maya render engines

Maya supports multiple render engines that are fully compatible with Deadline Cloud:

Render Engine	Description	GPU Support	Notes	Usage-Based Licensing
Maya Software	Built-in CPU renderer	CPU-based	Legacy renderer with basic features	Included with Maya

Render Engine	Description	GPU Support	Notes	Usage-Based Licensing
Arnold (MtoA)	Monte Carlo ray tracer	GPU/CPU hybrid	Production quality rendering, MtoA 5.3.5+ required	Available for 2024-2026
V-Ray	Third-party photorealistic renderer	GPU/CPU hybrid	Requires separate licensing	Available for 2025-2026
Redshift	GPU-accelerated renderer	GPU optimized	Requires separate licensing	Available for 2025-2026

All render engines are automatically detected and configured by the Maya integrated submitter. The submitter maintains proper dependency handling and scene file management.

Maya plugins

Plugin	Plugin Versions	Conda Recipe Provided	SMF Conda Package Provided	Usage-based Licensing Support
Arnold (MtoA)	2024.5.3, 2025.5.4, 2026.5.5	Yes	Yes	Yes
V-Ray	2025.7, 2026.7	Yes	Yes	Yes
Redshift	2025.4, 2026.2.1	Yes	Yes	Yes

Arnold for Maya (MtoA)

Arnold is supported using the maya-mtoa conda package and is automatically installed when using the Maya integrated submitter. An additional licensing cost applies when using Arnold for rendering.

Conda recipe: maya-mtoa conda recipe

V-Ray Plugin

V-Ray is supported using the maya-vray conda package and is automatically installed when using the Maya integrated submitter. An additional licensing cost applies when using V-Ray for rendering.

Conda recipe: maya-vray conda recipe

Redshift Plugin

Redshift is supported using the maya-redshift conda package and is automatically installed using the Maya integrated submitter. An additional licensing cost applies when using Redshift for rendering.

Conda recipe: maya-redshift conda recipe

Open source resources

The submitter and adaptor are open source and available on GitHub:

- Maya submitter source code
- Maya conda recipes

Autodesk VRED

Note

For more information about installing, configuring, and using this integration on your workstation, see the VRED integration user guide on GitHub.

Autodesk VRED is a professional 3D visualization and virtual prototyping software that brings complex 3D data to life in a realistic virtual environment. This software is widely used by designers and engineers to create product presentations, design reviews, and virtual prototypes, particularly in the automotive industry.

Support overview

VRED is partially supported by Deadline Cloud with the following components:

- **Submitters**: Integrated submitters for direct job submission from VRED Pro with automatic scene and asset detection.
- **Conda packages**: Automatic installation on service-managed fleets for Linux workers using the vredcore package.
- **Cross-platform compatibility**: Submitter support for Windows with worker support for Linux with automatic path mapping. (VRED Conda packages are available for Linux only; Windows workers require manual installation.)
- **BYOL Licensing**: VRED requires Bring Your Own License (BYOL). Unlike some other DCC applications in Deadline Cloud, usage-based licensing is not available for VRED. You must have valid VRED licenses available for your render farm fleet and configure your license server to be accessible from your workers.

VRED version compatibility

The following table shows current support levels for VRED versions:

Major Version	Submitter Support	Conda Support	Usage-Based Licensing
2026	Windows	Linux	BYOL required
2025	Windows	Linux	BYOL required

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to VRED available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Linux	vredcore	2025	VRED Core for Linux
Linux	vredcore	2026	VRED Core for Linux

Requirements

To use VRED with Deadline Cloud, you need:

- VRED Pro or VRED Core 2025/2026 with valid licensing
- Python 3.11 or higher
- NVIDIA GPU driver 553.xx (recommended for optimal performance)
- Valid VRED licenses accessible from your render farm fleet
- Optionally: ImageMagick static binary for tile assembly when using region rendering with raytracing

⚠ **Important**

VRED integration requires **bring your own licensing (BYOL)**. You must have valid VRED licenses available for your render farm fleet and configure your license server to be accessible from worker nodes. For more information, see Connect service-managed fleets to a custom license server.

Getting started

To use VRED with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Ensure your fleet has access to your VRED license server.
2. Install the Deadline Cloud monitor and VRED submitter on your artist workstation using the Deadline Cloud Submitter and monitor installers. For more information, see Set up your workstation.
3. Open VRED and load your scene file.

4. Submit your job directly from VRED using the integrated submitter by selecting **Deadline Cloud** > **Submit to Deadline Cloud** from the menu.
5. Monitor the job and download the output using the Deadline Cloud monitor.

Advanced configuration

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of VRED does not appear in the version table above.

If you require an unsupported version of VRED, you may build a custom Conda recipe and channel for your desired version to be installed on the worker. Use the Conda recipe for a supported version linked below as a starting point and package your desired version in a custom conda channel. For more information about creating custom Conda channels, see Creating custom conda channels.

Open source resources

The submitter and adaptor are open source and available on GitHub:

- VRED Submitter and Adaptor
- VRED Conda recipes are available on GitHub for supported versions.

Blender

Note

For more information about installing, configuring, and using this integration on your workstation, see the Blender integration user guide on GitHub.

Blender is a free and open-source 3D computer graphics software toolset used for creating animated films, visual effects, art, 3D printed models, motion graphics, interactive 3D applications, virtual reality, and computer games. Blender is supported by Deadline Cloud with comprehensive integration including submitters, conda packages, and an adaptor for increased rendering performance.

Support overview

Blender is supported by the following components:

- **Submitter**: Integrated submitter for direct job submission from Blender with automatic scene and asset detection.
- **Conda packages**: Deadline Cloud for automatic installation on service-managed fleets.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows, macOS, and Linux with worker support for Windows and Linux with automatic path mapping.

Blender version compatibility

The following table shows current support levels for Blender versions:

Major Version	Submitter Support	Conda Support	Render Engines
3.6	Windows, macOS, Linux	Linux	Cycles, Eevee, Workbench
4.2	Windows, macOS, Linux	Linux	Cycles, Eevee, Workbench
4.5	Windows, macOS, Linux	Linux	Cycles, Eevee, Workbench

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Blender available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Linux	blender	3.6	Includes all built-in render engines
Linux	blender	4.2	Includes all built-in render engines
Linux	blender	4.5	Includes all built-in render engines
Linux	blender-openjd		Includes the Blender Adaptor

Getting started

To use Blender with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor and Blender submitter on your artist workstation using the Deadline Cloud monitor and submitter installers. For more information, see Set up your workstation.
3. Submit your job directly from Blender using the integrated submitter to the queue.
4. Monitor the job and download the output using the Deadline Cloud monitor.

For more information about using the Blender integrated submitter, see the Blender integration user guide on GitHub.

Using the Blender submitter

To submit a render job from Blender:

1. Open Blender and load your scene file.
2. Configure your render settings including output path, frame range, and render engine (Cycles, Eevee, or Workbench).

3. From the top menu, select **Render** > **Deadline Cloud**.
4. In the Deadline Cloud submission dialog:
    - Enter a job name and description.
    - Select your target farm and queue.
    - Configure job attachments to include your scene file and any external assets.
    - Review render settings and frame range.
5. Choose **Submit** to send your job to the queue.

The Deadline Cloud submission will automatically detect your scene dependencies, configure the appropriate render engine, and submit the job with the correct conda packages for your Blender version.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of Blender does not appear in the version table above.

If you require an unsupported version of Blender, you have the following options:

- When submitting the job from Blender, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `blender=4.5, blender-openjd=*`). This may or may not work, depending on the features used by your scene and how Blender works with scenes from your workstation version.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked below as a starting point, and package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

Blender render engines

Blender includes several built-in render engines that are supported:

Render Engine	Description	GPU Support	Notes
Cycles	Physically-based path tracer	GPU/CPU hybrid	Production quality rendering with GPU acceleration
Eevee	Real-time render engine	GPU optimized	Fast viewport and final rendering
Workbench	Solid shading engine	GPU optimized	For modeling and sculpting workflows

All render engines are automatically detected and configured by the Blender integrated submitter. GPU acceleration is available when using service-managed fleets with GPU-enabled instances.

Open source resources

The submitter and adaptor are open source and available on GitHub:

- Deadline Cloud for Blender
- Blender Conda recipes are available on GitHub for supported versions.

Epic Unreal Engine

> **Note**
>
> For more information about installing, configuring, and using this integration on your workstation, see the Unreal Engine integration user guide on GitHub.

Unreal Engine is a real-time 3D creation tool for photoreal visuals and immersive experiences. Unreal Engine is supported by Deadline Cloud with submitters, conda packages, and an adaptor for increased rendering performance.

Support overview

Unreal Engine is supported by the following components:

- **Submitter**: Integrated submitter plugin for direct job submission from Unreal Engine with automatic scene and asset detection.
- **Conda packages**: Deadline Cloud for automatic installation on service-managed fleets.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter and worker support for Windows only.
- **Movie Render Queue Integration**: Support for Unreal's Movie Render Queue system.

Unreal Engine version compatibility

The following table shows current support levels for Unreal Engine versions:

Major Version	Submitter Support	Conda Support
5.4	Windows	Windows
5.5	Windows	Windows
5.6	Windows	Windows

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Unreal Engine available to Service-managed fleets in the `deadline-cloud` conda channel:

OS	Package	Version
Windows	unreal-engine	5.4
Windows	unreal-engine	5.5
Windows	unreal-engine	5.6
Windows	unreal-engine-openjd	

Getting started

Prerequisites

Before installing the Unreal Engine submitter, ensure you have the following:

- Windows workstation (Windows 10 or later)
- Supported version of Unreal Engine installed
- Deadline Cloud monitor installed (download here)
- Access to an Deadline Cloud farm with either a GPU-enabled Windows service-managed fleet or a customer-managed fleet with Unreal Engine, the Unreal Engine adaptor, and licensing set up

Installing the Unreal Engine Submitter

The Unreal Engine submitter adds Deadline Cloud functionality as a plugin to Unreal Engine, allowing you to submit your Movie Render Queue jobs directly to Deadline Cloud for rendering.

For detailed installation instructions, see the Unreal Submitter Setup Guide.

Updating the Submitter

Refresh your git repository and re-run the installation script as mentioned in Unreal Submitter Setup Guide.

Using the Unreal Engine submitter

To use the Unreal Engine submitter:

1. Open Unreal Engine with your project.
2. Set up your Movie Render Queue with the desired shots and render settings.
3. Access the Deadline Cloud submitter plugin from the Unreal Engine interface.
4. Configure your job settings including:
   - Movie Render Queue configuration
   - Output paths and formats
   - Render parameters
5. Choose **Submit** to send your job to Deadline Cloud.

The submitter automatically detects Movie Render Queue configurations and handles asset dependencies, including project plugins and content files.

Advanced configurations

Service-Managed Fleets vs Customer-Managed Fleets

Service-Managed Fleets (SMF)

On Service-Managed Fleets, the Unreal Engine and adaptor are automatically available via the `deadline-cloud` Conda channel with the default Queue Environment. This provides the easiest setup experience.

Customer-Managed Fleets (CMF)

For Customer-Managed Fleets, Unreal Engine and the adaptor must be manually installed on worker hosts. This setup provides more control and supports additional features like Perforce integration.

For detailed instructions, see the CMF Worker Setup Guide.

Perforce integration

Unreal Engine integration includes support for Perforce version control systems. The integration provides utilities for syncing dependent files and managing Perforce workspaces during rendering.

For more information on submitting perforce integrated jobs to deadline-cloud see Perforce Guide
.

Unreal Engine rendering features

Unreal Engine's rendering system provides comprehensive support for:

Feature	Description	Notes
Movie Render Queue	High-quality offline rendering	Integration with job submission
Sequencer	Timeline-based animation system	Automatic shot detection and processing

Feature	Description	Notes
Project Plugins	Custom plugin support	Automatic detection and inclusion
Asset Dependencies	Content file management	Comprehensive asset tracking
Sticky Rendering	Application persistence between shots	Improved performance for multi-shot sequences

All rendering features are automatically detected and configured by the Unreal Engine integrated submitter. The adaptor maintains proper dependency handling and supports efficient multi-shot rendering without restarting Unreal Engine.

Open source resources

The submitter and adaptor are open source and available on GitHub:

- Deadline Cloud for Unreal Engine

Foundry Nuke

Note

For more information about installing, configuring, and using this integration on your workstation, see the Nuke integration user guide on GitHub.

Foundry Nuke is a node-based digital compositing and visual effects application used for television and film post-production. Nuke is supported by Deadline Cloud with submitters, conda packages, and an adaptor for increased rendering performance.

Support overview

Nuke is supported by the following components:

- **Submitter**: Integrated submitter plugin for direct job submission from Nuke with automatic scene and asset detection.

- **Conda packages**: Packages to install nuke versions 15 and 16 are available on the Deadline Cloud conda channel for service-managed fleets.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows, macOS, and Linux with worker support for Linux only with automatic path mapping.

Nuke version compatibility

The following table shows current support levels for Nuke versions:

Major Version	Submitter Support	Conda Support
15	Windows, macOS, Linux	Linux
16	Windows, macOS, Linux	Linux

Deadline Cloud Conda Channel

The following table lists conda packages applicable to Nuke available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Linux	nuke	15	Includes built-in compositing engine
Linux	nuke	16	Includes built-in compositing engine
Linux	nuke-openjd		Includes the Nuke Adaptor

Getting started

To use Nuke with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor and Nuke submitter on your artist workstation using the Deadline Cloud Submitter and monitor Installers. For more information, see Set up your workstation.
3. Submit your job directly from Nuke using the integrated submitter to the queue.
4. Monitor the job and download the output using the Deadline Cloud monitor.

Launch the submitter

To launch the Deadline Cloud submitter in Nuke

> **Note**
>
> Support for Nuke is provided using the Conda environment for service-managed fleets. For more information, see Default conda queue environment.

1. Install the Deadline Cloud monitor and Nuke submitter on your artist workstation using the Deadline Cloud Submitter and monitor Installers. For more information, see Set up your workstation.
2. Open **Nuke**.
3. Open a Nuke script with dependencies that exist within the asset root directory.
4. Choose **AWS Deadline**, and then choose **Submit to Deadline Cloud** to launch the submitter.
    a. If you are not already authenticated in the Deadline Cloud submitter, the **Credentials Status** shows as **NEEDS_LOGIN**.
    b. Choose **Login**.
    c. In the login browser window, log in with your user credentials.
    d. Choose **Allow**. You are now logged in and the **Credentials Status** shows as **AUTHENTICATED**.
5. Choose **Submit**.

Using the Nuke submitter

To use the Nuke submitter:

1. Open Nuke.
2. Load your composition with the required Write nodes configured.
3. From the menu, choose **Deadline Cloud** to launch the submitter.
4. If you are not already authenticated, choose **Login** and authenticate with your credentials.
5. Configure your job settings in the submitter interface, including:
   - Frame range settings
   - Write node selection
   - Output paths and formats
6. Choose **Submit** to send your job to Deadline Cloud.

The submitter automatically detects Write nodes in your composition and allows you to select which ones to render. It also handles automatic input/output path detection and supports multiple views rendering.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of Nuke does not appear in the version table above.

If you require an unsupported version of Nuke, you have the following options:

- When submitting the job from Nuke, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `nuke=16, nuke-openjd=*`). This may or may not work, depending on the features used by your composition and how Nuke works with compositions from your workstation version.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked below as a starting point, and

package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

Custom Nuke executable

You can set the NUKE_EXECUTABLE environment variable to point to a specific Nuke executable if it's not available on the PATH.

OpenColorIO support

The Nuke integration includes full support for OpenColorIO (OCIO) color management workflows. Color configurations are automatically detected and included with job submissions to ensure consistent color handling across the render farm.

Nuke compositing features

Nuke's compositing engine provides comprehensive support for:

Feature	Description	Notes
Write Nodes	Multiple output formats and codecs	Automatically detected by submitter
Frame Ranges	Custom frame range specification	Supports override and default ranges
Multiple Views	Stereo and multi-view rendering	Proper handling of view-specific outputs
Color Management	OpenColorIO integration	Automatic OCIO configuration detection
Path Mapping	Cross-platform path translation	Seamless Windows/Linux compatibility

Compositing features are automatically detected and configured by the Nuke integrated submitter. The submitter maintains proper dependency handling and asset management for complex compositions.

Open source resources

The submitter and adaptor are open source and available on GitHub:

- Deadline Cloud for Nuke
- Nuke Conda recipes are available on GitHub for supported versions.

KeyShot Studio

Note

For more information about installing, configuring, and using this integration on your workstation, see the KeyShot integration user guide on GitHub.

KeyShot Studio is a real-time ray tracing and global illumination program developed by Luxion for rendering 3D models and animations.

Support overview

KeyShot Studio is supported by the following components:

- **Submitter**: Integrated submitter extension for direct job submission from KeyShot with automatic scene and asset detection.
- **Conda package**: Pre-packaged software for automatic installation on service-managed fleets.
- **Cross-platform compatibility**: Submitter support for Windows and macOS with worker support for Windows.
- **Usage-based Licensing**: Pay-as-you-go for KeyShot licensing.

KeyShot version compatibility

The following table shows current support levels for Keyshot versions:

Major Version	Submitter Support	Conda Support	Render Engines	Usage-Based Licensing
2024	Windows, macOS	Windows	Built-in ray tracer	Usage-based licensing available
2025	Windows, macOS	Windows	Built-in ray tracer	Usage-based licensing available

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Keyshot available to Service-managed fleets in the `deadline-cloud` conda channel:

OS	Package	Version	Notes
Windows	keyshot	2024	Includes built-in ray tracer
Windows	keyshot	2025	Includes built-in ray tracer
Linux	keyshot-openjd		Includes the KeyShot Adaptor

Getting started

To use KeyShot with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the `deadline-cloud` conda channel. For more information, see Creating a queue environment.

2. Install the Deadline Cloud monitor and KeyShot submitter on your artist workstation using the Deadline Cloud Submitter and monitor installers. For more information, see Set up your workstation.

Using the KeyShot submitter

To use the KeyShot submitter:

1. Open KeyShot.
2. Choose **Windows** > **Scripting console** > **Submit to Deadline Cloud** and **Run**.
3. Select your preferred submission mode from the dialog that appears.
4. Configure your job settings in the submitter interface.
5. Choose **Submit** to send your job to Deadline Cloud.
6. Monitor the job and download the output using the Deadline Cloud monitor.

For more information about using the KeyShot submitter for Deadline Cloud, see KeyShot submitter guide.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of KeyShot does not appear in the version table above.

If you require an unsupported version of KeyShot, you have the following options:

- When submitting the job from KeyShot, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `keyshot=2024`). The job may run successfully depending on the features used by your scene and how KeyShot works with scenes from the version on your workstation.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked below as a starting point, and package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

Open source resources

The submitter is open source and available on GitHub:

- Deadline Cloud for KeyShot
- Standalone KeyShot job bundle is available on GitHub.
- Comprehensive user guide is available.

Maxon Cinema 4D

Note

For more information about installing, configuring, and using this integration on your workstation, see the Cinema 4D integration user guide on GitHub.

Cinema 4D is a professional 3D animation, modeling, simulation and rendering software solution from Maxon. Cinema 4D is supported by Deadline Cloud including a submitter, conda packages, usage-based licensing and an adaptor for improved performance.

Support overview

Cinema 4D is supported by the following components:

- **Submitter**: Integrated submitter for direct job submission from Cinema 4D with automatic scene and asset detection.
- **Conda packages**: Automatic installation on service-managed fleets when using the submitter.
- **Adaptor**: Middleware for more efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows and macOS with worker support for Windows and Linux with automatic path mapping.
- **Usage-based Licensing**: Pay-as-you-go licensing for Cinema 4D, Redshift, and Red Giant licensing.

Cinema 4D version compatibility

The following table shows current support levels for Cinema 4D versions:

Major Version	Submitter Support	Conda Support	Usage-Based Licensing
2024	Windows, macOS	Windows	Usage-based licensing available
2025	Windows, macOS	Windows, Linux	Usage-based licensing available
2026	Windows, macOS	Windows, Linux	Usage-based licensing available

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Cinema 4D available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Windows	cinema4d	2024	Includes Standard, Physical and Redshift renderers
Windows, Linux	cinema4d	2025	Includes Standard, Physical and Redshift renderers
Windows, Linux	cinema4d	2026	Includes Standard, Physical and Redshift renderers
Windows, Linux	cinema4d-c4dtoa	2025	Cinema4D to Arnold
Windows	cinema4d-c4dtoa	2026	Cinema4D to Arnold
Windows, Linux	cinema4d-openjd		Includes the Cinema 4D Adaptor

Note

For **Cinema 4D**, the Linux conda package does not support substance 3D materials. Jobs with this material fail with one of the following errors:

```
Commandline: ./modules/io_substance/source/substance_framework/src/details/
detailsengine.cpp:794:
 SubstanceAir::Details::Engine::Context::Context(SubstanceAir::Details::Engine&,
 SubstanceAir::RenderCallbacks*): Assertion `res==0' failed.
```

```
/home/job-user/.conda/envs/<hash>/Lib/deadline/cinema4d_adaptor/Cinema4DAdaptor/
adaptor.sh: line 44: 10832 Segmentation fault      (core dumped) $C4DEXE
 ${ARGS[*]}
```

We recommend that you submit jobs with substance materials to Windows instead.
In Cinema 4D 2025.3.3 on Linux, globalized asset paths can cause segmentation faults. Therefore, the Linux conda package contains Cinema 4D 2025.3.1 with Redshift 2025.6.0 instead. If you need features or bug fixes from Cinema 4D 2025.3.3, we recommend two options: upgrade to Cinema 4D 2026 or submit those jobs to Windows instead.
For **Cinema 4D OpenJD,** to prevent any timeout issues, we recommend you set task run timeouts to double their expected render time, instead of using the default 2 day timeout.

Getting started

To use Cinema 4D fully-managed on Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Configure the fleet with GPU support if you intend to use Redshift or Red Giant features that require a GPU. Your queue should be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor and Cinema 4D submitter on your artist workstation using the Deadline Cloud Submitter and monitor Installers. For more information, see Set up your workstation.
3. Submit your job directly from Cinema 4D using the integrated submitter to the queue.
4. Monitor the job and download the output using the Deadline Cloud monitor.

For more information about using the Cinema 4D integrated submitter, see the Cinema 4D integration user guide on GitHub.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This will fail if the workstation version of Cinema 4D does not appear in the version table above.

If you require an unsupported version of Cinema 4D, you may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked in the Open Source Resources section below as a starting point, and package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

If you create a conda package for a different version of Cinema 4D, you should ensure it will acquire a license correctly. If the version is compatible with licensing for a supported version in the table above, then usage-based licensing will work automatically. You may also bring your own license to a service-managed fleet by following Connect service-managed fleets to a custom license server.

Cinema 4D plugins

Plugin	Plugin Version	Conda Recipe Provided	SMF Conda Package Provided	Usage-based Licensing Support
Redshift	2026.3.0	Bundled*	Yes	Yes
Redshift	2025.6.0	Bundled*	Yes	Yes
Red Giant	2025.x	No	No	Yes
V-Ray	7.x	Yes	No	Yes
Insydium X-Particles	2024.x	Yes	No	N/A

Plugin	Plugin Version	Conda Recipe Provided	SMF Conda Package Provided	Usage-based Licensing Support
C4DtoArnold	4.8.4.1	Yes	Yes	Yes

*Included in the base Cinema 4D package recipe

Maxon Redshift

The Redshift renderer is included with all Cinema 4D conda packages and is automatically used when appropriate when using the Cinema 4D integrated submitter. An additional licensing cost applies when using Redshift for rendering. For more information about Deadline Cloud pricing, see Deadline Cloud pricing.

Maxon Red Giant

Red Giant is a comprehensive toolkit designed for video post-production, motion graphics, and visual effects. It offers rich color grading, smooth transitions, realistic visual effects, motion design templates and tools to create and edit your visuals. For more information, see Red Giant.

Red Giant requires custom setup on service-managed fleets. A host configuration script is provided which you can use in your Deadline Cloud fleet. Once configured, Red Giant is supported by Deadline Cloud Usage-based Licensing and requires no further configuration to operate.

V-Ray Plugin

V-Ray is a 3D photorealistic ray-traced rendering plug-in. V-Ray for Cinema 4D is not currently fully supported in Service-managed fleets. A conda recipe is provided which you can use to create your own Conda channel for use in your Deadline Cloud farm. For more information about creating custom conda channels, see Creating custom conda channels. Once installed, V-Ray is supported by Deadline Cloud Usage-based Licensing and requires no further configuration to operate.

C4DToArnold

Autodesk Arnold software is an advanced Monte Carlo ray tracing renderer. For more information, see Arnold. C4DToArnold is fully supported in Service-managed fleets.

Insydium X-Particles

X-Particles is a fully-featured advanced particle and VFX system for Maxon's Cinema 4D. For more information, see X-Particles. Insydium X-Particles is not currently fully supported in Service-managed fleets. A conda recipe is provided which you can use to create your own Conda channel for use in your Deadline Cloud farm. For more information about creating custom conda channels, see Creating custom conda channels. When you create the conda package from your X-Particles package, it will include your purchased license. No additional configuration is necessary to operate on service-managed fleets.

Open source resources

The submitter and adaptor are open source and available on GitHub:

- Deadline Cloud for Cinema 4D
- Cinema 4D Conda recipes are available on GitHub for C4D 2024, C4D 2025, the INSYDIUM X-PARTICLES plugin, the C4DtoA plugin, and the V-Ray Plugin.
- Host Configuration script is included to support Red Giant plugins.

SideFX Houdini

Note

For more information about installing, configuring, and using this integration on your workstation, see the Houdini integration user guide on GitHub.

SideFX Houdini is a 3D procedural software for modeling, rigging, animation, VFX, look development, lighting and rendering in film, TV, advertising and video game pipelines. Houdini is fully supported by Deadline Cloud with comprehensive integration including submitters, conda packages, and an adaptor for increased rendering performance.

Support overview

Houdini is supported by the following components:

- **Submitter**: Integrated render output node (ROP) for direct job submission from Houdini with automatic scene and asset detection.

- **Conda packages**: Deadline Cloud for automatic installation on service-managed fleets.
- **Adaptor**: Middleware for efficient rendering with sticky sessions and additional monitoring.
- **Cross-platform compatibility**: Submitter support for Windows, macOS, and Linux with worker support for Windows and Linux with automatic path mapping.

Houdini version compatibility

The following table shows current support levels for Houdini versions:

Major Version	Submitter Support	Conda Support	Render Engines	Usage-Based Licensing
19.0	Windows, macOS, Linux	Linux	Mantra, Karma CPU, Karma XPU	Usage-based licensing available
19.5	Windows, macOS, Linux	Linux	Mantra, Karma CPU, Karma XPU	Usage-based licensing available
20.0	Windows, macOS, Linux	Linux	Mantra, Karma CPU, Karma XPU	Usage-based licensing available
20.5	Windows, macOS, Linux	Linux	Mantra, Karma CPU, Karma XPU	Usage-based licensing available
21.0	Windows, macOS, Linux	Linux	Mantra, Karma CPU, Karma XPU	Usage-based licensing available

Deadline Cloud Conda Channel

The following table lists all conda packages applicable to Houdini available to Service-managed fleets in the deadline-cloud conda channel:

OS	Package	Version	Notes
Linux	houdini	19.0	Includes Mantra and Karma renderers
Linux	houdini	19.5	Includes Mantra and Karma renderers
Linux	houdini	20.0	Includes Mantra and Karma renderers
Linux	houdini	20.5	Includes Mantra and Karma renderers
Linux	houdini	21.0	Includes Mantra and Karma renderers
Linux	houdini-openjd		Includes the Houdini Adaptor

Getting started

To use Houdini with Deadline Cloud:

1. Create a service-managed fleet and associate it with a queue. Your queue must be set up with a queue environment that supports the deadline-cloud conda channel. For more information, see Creating a queue environment.
2. Install the Deadline Cloud monitor and Houdini submitter on your artist workstation using the Deadline Cloud Submitter and monitor installers. For more information, see Set up your workstation.
3. Submit your job directly from Houdini using the integrated submitter to the queue.
4. Monitor the job and download the output using the Deadline Cloud monitor.

Using the Houdini submitter

To use the Houdini submitter:

1. Open Houdini.
2. In the Network Editor, usually in the lower right side of Houdini, select the `/out` network.
3. Press **Tab**, and enter `deadline`.
4. Select the **Deadline Cloud** option and place it within the `/out` network to create the node.
5. Connect the output of the last render output node (ROP) (for example, Karma, Mantra, or compositing) in your existing `/out` network to the input of the Deadline Cloud node.
6. Choose the Deadline Cloud node.
7. Enter any job settings in the node editor, usually in the upper right side of Houdini.
8. In the bottom right of the node editor, choose **Submit**.

The Deadline Cloud submission will automatically parse the connected `/out` network tree and submit each node as a step in the job maintaining the dependency tree. Using non-default render networks other than `/out` is also supported.

Advanced configurations

Using unsupported versions

Deadline Cloud only supports and tests the workstation and worker software versions in the table above. When using the submitter, the worker will attempt to install the same version as used on the workstation. This may fail if the workstation version of Houdini does not appear in the version table above.

If you require an unsupported version of Houdini, you have the following options:

- When submitting the job from Houdini, you may override the CondaPackages queue parameter to specify a supported version to use on the worker (for example, `houdini=21.0, houdini-openjd=*`). This may or may not work, depending on the features used by your scene and how Houdini works with scenes from your workstation version.
- You may build a custom conda recipe and channel for your desired version to be installed on the worker. Use the conda recipe for a supported version linked below as a starting point, and package your desired version in a custom conda channel. For more information about creating custom conda channels, see Creating custom conda channels.

Houdini render engines

Houdini supports multiple render engines that are compatible with Deadline Cloud:

Render Engine	Description	GPU Support
Karma CPU	Modern USD-based renderer (CPU variant)	CPU-based
Karma XPU	Modern USD-based renderer (GPU variant)	GPU accelerated
Mantra	Traditional Houdini renderer	CPU-based
Arnold	Third-party Monte Carlo ray tracer	GPU/CPU hybrid
V-Ray	Third-party photorealistic renderer	GPU/CPU hybrid
Redshift	GPU-accelerated renderer	GPU optimized

These render engines are automatically detected and configured by the Houdini integrated submitter and usage is automatically licensed. The submitter maintains dependency trees between connected render output nodes (ROPs).

Open source resources

The submitter and adaptor are open source and available on GitHub. Houdini Conda recipes are available on GitHub for supported versions.

- Houdini submitter source code on GitHub
- Sample scenes and workflows on GitHub
- Conda recipes for supported versions on GitHub

File storage for Deadline Cloud

Workers must have access to the storage locations that contain the input files necessary to process a job, and to the locations that store the output. AWS Deadline Cloud provides two options for storage locations:

- With *job attachments*, Deadline Cloud transfers the input and output files for your jobs back and forth between a workstation and Deadline Cloud workers. To enable the file transfers, Deadline Cloud uses an Amazon Simple Storage Service (Amazon S3) bucket in your AWS account.

  When you use job attachments with a Linux based service-managed fleet, you can enable a virtual file system (VFS) to mount job attachments files and access them as needed instead of syncing them to the worker at the start of the job.
- With *shared storage*, you use file sharing with your operating system to provide access to files.

  When you use cross-platform shared storage, you can create a *storage profile* so that workers can map the path to files between two different operating systems.

  You can also integrate third-party cloud storage solutions, such as LucidLink, with service-managed fleets using host configuration scripts. For more information, see Set up LucidLink with service managed fleet scripts for Deadline Cloud on the AWS for M&E Blog.

Topics

Storage profiles in Deadline Cloud

When you use workstations and fleet worker hosts from multiple operating systems or with different file system mounts, you can create storage profiles in your farm to indicate where the same file systems are mounted on different systems. When Deadline Cloud runs a job on a different storage profile than on the workstation it was submitted from, it will transform file system paths that are in directories configured in the storage profiles.

Using storage profiles in your Deadline Cloud farm enables the following behaviors:

- When submitting a job to a queue, the files that the job references will be categorized by the workstation storage profile:
  - Files that are under a shared file system location will be left alone.
  - Files that are under a local file system location will be attached to the job by uploading them to the job attachments S3 bucket. Files that were previously uploaded are not uploaded again.
  - Files that are not under any file system location will be attached to the job as well. The job submitter will warn about these file paths unless they are under a known path in the local Deadline Cloud settings.
- When jobs are running on a fleet worker host with a different operating system or storage profile than the submitting workstation, file paths used by the job will be mapped from the submitting storage profile to the fleet storage profile.
- When downloading job output, jobs that were submitted for a different operating system or storage profile will have their paths mapped from the submitting storage profile to the local workstation storage profile.

For more information, see Storage profiles and path mapping in the *AWS Deadline Cloud Developer Guide*.

To create a storage profile

1. Open the Deadline Cloud console.
2. From **Get started**, choose **Go to Deadline Cloud dashboard**.
3. Choose a farm, and then choose the **Storage profiles** tab.
4. Choose **Create storage profile**.
5. From the dropdown, select an **Operating system**.
6. Enter a **Storage profile name**. The name is how you choose a storage profile for a workstation. For example, names like *Windows-Workstation* or *Windows-OnPremFleet* can make it easy to identify it later.
7. Create a file system location of **Shared** type for each shared file system that is mounted on both workstations and fleet worker hosts.

   1. Enter a **name** that identifies the mount, such as *Projects* for a shared file system that contains project data, or *Tools* for a shared file system with tools to use.
   2. Enter the mount location for the chosen shared file system on the operating system of the storage profile.

8. Create a file system location of **Local** type for each shared file system that is only for workstations. For example when your fleets are on AWS and you want job attachments to handle the data transfer. You can also create this kind of file system location for directories that are local to each workstation, to specify equivalent paths on different operating systems even if they are not mounted storage.

    1. Enter a **name** that identifies the mount, such as *Projects* for a shared file system that contains project data, or *Tools* for a shared file system with tools to use.
    2. Enter the chosen file system location on the operating system of the storage profile.

9. (Optional) To add another file system location, choose **Add new required file system location** and enter the required data.
10. After you have added all of the required file system locations, choose **Create**.

To set up your storage profile for use

1. Navigate to the queue that you want to use this storage profile with, and select the **Allowed storage profiles** tab.
2. Choose **Configure storage profiles**.
3. From the dropdown list to associate storage profiles, select the storage profile you created.
4. In the Required file system location list, select the **file system location names** that you want to ensure are available on any storage profile for the associated fleets.
5. (Optional) If you created the storage profile for a fleet, navigate to the fleet and select the **Configurations** tab.

    a. From the Storage profiles section, choose **Configure storage profile**.

    b. Select the storage profile, and then choose **Save changes**.

To configure a storage profile on a workstation

On each workstation that will submit jobs to a queue, use the settings dialog to select its default storage profile.

1. To open the Deadline Cloud settings dialog, complete one of the following steps:

    a. Select the **Settings** button in a Deadline Cloud submitter.

    OR

    b. Run the CLI command `deadline config gui`.

2. After you have configured the default farm and queue, select the default storage profile from the dropdown list.

Storage profiles for shared file systems

You can configure your Deadline Cloud fleets to mount shared file systems by using VPC resource endpoints on service-managed fleets, or by configuring the hosts of a customer-managed fleet on AWS or on-premises. When workstations have the same shared file systems mounted as your fleets, you can create file system locations of the shared type in your storage profiles to configure where each shared file system appears as a local path.

For example, suppose you have one shared file system for projects and another one for tools. Your workstations and fleets include the three operating systems Windows, macOS, and Linux. You can create one storage profile for each operating system with the following values:

- **Storage profile name**: Linux-Host, **operating system family**: Linux.
  - **File system location name**: Projects, **path**: /mnt/projects, **type**: Shared.
  - **File system location name**: Tools, **path**: /mnt/projects, **type**: Shared.
- **Storage profile name**: Windows-Host, **operating system family**: Windows.
  - **File system location name**: Projects, **path**: X:\projects, **type**: Shared.
  - **File system location name**: Tools, **path**: Z:, **type**: Shared.
- **Storage profile name**: MacOS-Host, **operating system family**: MacOS.
  - **File system location name**: Projects, **path**: /Volumes/Projects, **type**: Shared.
  - **File system location name**: Tools, **path**: /Volumes/Tools, **type**: Shared.

When you submit a job from Windows that uses a path X:\Projects\ProjectA\Textures\texture.jpg, Deadline Cloud will add a field containing the Windows-Host storage profile id to the job.

If the job runs on a Linux fleet worker host, Deadline Cloud will create two path mapping rules for the job based on corresponding file system location names: X:\Projects -> /mnt/projects, Z: -> /mnt/tools. The job will apply these rules to resolve the original paths to where the Linux host sees them.

If job attachments are also configured for your queue, any paths that are not under a file system location of type shared will be attached to the job and uploaded to the job attachments S3 bucket.

This lets you attach data files to the job instead of requiring that they always be copied to a shared file system. For example, providing auxiliary files defined by the job bundle you submit.

Storage profiles for job attachments

You can configure your Deadline Cloud queue to use job attachments for transferring asset data referenced by your jobs to and from AWS. When workstations mount the same shared file systems, but your fleets do not, you can create file system locations of the local type in your storage profiles. This configuration lets you configure where you will upload and download files from, and how to map paths between operating systems.

For example, suppose you have one shared file system for projects and another one for tools. Your workstations and fleets include the three operating systems Windows, macOS, and Linux. Everything is the same as in the **Storage profiles for shared file systems** topic except the file systems are not shared with the farm. They are for the local area network containing your workstations. You can create one storage profile for each operating system with the following values:

- **Storage profile name**: Linux-Host, **operating system family**: Linux.
  - **File system location name**: Projects, **path**: /mnt/projects, **type**: Local.
  - **File system location name**: Tools, **path**: /mnt/projects, **type**: Local.
- **Storage profile name**: Windows-Host, **operating system family**: Windows.
  - **File system location name**: Projects, **path**: X:\projects, **type**: Local.
  - **File system location name**: Tools, **path**: Z:, **type**: Local.
- **Storage profile name**: MacOS-Host, **operating system family**: MacOS.
  - **File system location name**: Projects, **path**: /Volumes/Projects, **type**: Local.
  - **File system location name**: Tools, **path**: /Volumes/Tools, **type**: Local.

When you submit a job from Windows that uses a path X:\Projects\ProjectA\Textures\texture.jpg, Deadline Cloud will add a field containing the Windows-Host storage profile id to the job and upload the file to the job attachments S3 bucket if it wasn't uploaded already.

If the job runs on a Linux fleet worker host, Deadline Cloud will make the texture file available in a local temporary directory, then create a path mapping rule from one of the directories containing the texture to the temporary directory. For example X:\Projects\ProjectA -> /sessions/session-123/projects, so that X:\Projects\ProjectA\Textures\texture.jpg maps to /sessions/session-123/

projects/Textures/texture.jpg. When a task of the job is complete, it collects the output from directories specified by the job. Suppose /sessions/session-123/projects/Output/frame0032.png is an output file. This output is recorded on the job as X:\Projects\ProjectA\Output\frame0032.jpg, matching on the storage profile for the workstation submitting the job.

When you download the job output on a macOS workstation, Deadline Cloud will create path mapping rules from the Windows workstation: X:\Projects -> /Volumes/Projects, Z: -> /Volumes/Tools. It applies the rule to all output paths, downloading the example output file to /Volumes/Projects/ProjectA/Output/frame0032.jpg.

If an output file path of a job is not contained under any of the storage profile file system locations, Deadline Cloud will not be able to determine its path for download when the storage profile is different from the submitting workstation. Depending on the command you use for download, that file will either be skipped or you will have to manually select a download directory.

Job attachments in Deadline Cloud

With *job attachments* you can transfer files back and forth between your workstation and AWS Deadline Cloud. With job attachments, you don't need to manually set up an Amazon S3 bucket for your files. Instead, when you create a queue with the Deadline Cloud console, you choose the bucket for your job attachments.

The first time that you submit a job to Deadline Cloud, all of the files for the job are transferred to Deadline Cloud. For subsequent submissions, only the files that have changed are transferred, saving both time and bandwidth.

After processing is complete, you can download the result from the job detail page, or by using the Deadline Cloud CLI `deadline job download-output` command.

You can use the same S3 bucket for multiple queues. Set a different root prefix for each queue to organize the attachments in the bucket.

When you create a queue with the console, you can either choose an existing AWS Identity and Access Management (IAM) role or you can have the console create a new role. If the console creates the role, it sets permissions to access the bucket that's specified for the queue. If you choose an existing role, you must grant the role permissions to access the S3 bucket.

Encryption for job attachment S3 buckets

Job attachment files are encrypted in your S3 bucket by default. This encryption helps secure your information from unauthorized access. You don't need to do anything to have your files encrypted with keys provided by Deadline Cloud. For more information, see Amazon S3 now automatically encrypts all new objects in the *Amazon S3 User Guide*.

You can use your own customer managed AWS Key Management Service key to encrypt the S3 bucket that contains your job attachments. To do so, you must modify the IAM role for the queue associated with the bucket to allow access to the AWS KMS key.

To open the IAM policy editor for the queue role

1. Sign in to the AWS Management Console and open the Deadline Cloud console. From the main page, in the **Get started** section, choose **View farms**.
2. From the list of farms, choose the farm that contains the queue to modify.
3. From the list of queues, choose the queue to modify.
4. In the **Queue details** section, choose the **Service role** to open the IAM console for the service role.

Next, complete the following procedure.

To update the role policy with permission for AWS KMS

1. From the list of **Permissions policies**, choose the policy for the role.
2. In the **Permissions defined in this policy** section, choose **Edit**.
3. Choose **Add new statement**.
4. Copy and paste the following policy into the editor. Change the *Region*, *accountID*, and *keyID* to your own values.

```
{
   "Effect": "Allow",
   "Action": [
       "kms:Decrypt",
       "kms:DescribeKey",
       "kms:GenerateDataKey"
   ],
   "Resource": [
       "arn:aws:kms:us-east-1:111122223333:key/keyID"
   ]
}
```

5. Choose **Next**.
6. Review the changes to the policy, and then when you're satisfied, choose **Save changes**.

Replace job attachments bucket

You can replace your current job attachments bucket with a different job attachments bucket. You will find a button under the **Job Attachments** tab in the queue details. You can use it to either change the job attachments bucket or replace the root folder inside the same bucket to upload the job attachments.

To access job attachments settings

1. Go to **Queue details**, then locate the **Job Attachments** tab.
2. From the job attachments tab, there are 2 options:

    a. Change the job attachments bucket by doing the following:

        i. Select a new S3 bucket.

        ii. Update the queue's service role policy to grant access to the new bucket.

    OR

    b. Change the root folder within an existing bucket by doing the following:

        i. Modify the root folder name.

        ii. Update the resource ARN in the queue service role.

To update the service role

1. Navigate to your farm > queue > queue service role.
2. Choose **Edit in JSON**.
3. Locate the resource ARN (default root folder is **DeadlineCloud**):

```
"arn:aws:s3:::<your-job-attachments-bucket-name>/DeadlineCloud/*"
```

```
]
```

4. Update the ARN with new bucket or folder:

```
 "arn:aws:s3:::<your-job-attachments-NEW-bucket-name>/NEW-ROOT-FOLDER-NAME/*"
]
```

5. Verify permissions after making these changes to ensure proper access.

Managing job attachments in S3 buckets

Deadline Cloud stores the job attachment files required for your job in an S3 bucket. These files accumulate over time, leading to increased Amazon S3 costs. To reduce costs, you can apply an S3 Lifecycle configuration to your S3 bucket. This configuration can automatically delete files in the bucket. Because the S3 bucket is in your account, you can choose to modify or remove the S3 Lifecycle configuration at any time. For more information, see Examples of S3 Lifecycle configuration in the *Amazon S3 User Guide*.

For a more granular S3 bucket management solution, you can set up your AWS account to expire objects in an S3 bucket based on the last time that they were accessed. For more information, see Expiring Amazon S3 objects based on last accessed date to decrease costs on the AWS Architecture Blog.

Deadline Cloud virtual file system

Virtual file system support for job attachments in AWS Deadline Cloud enables client software on workers to communicate directly with Amazon Simple Storage Service. Workers can load files only when needed instead of downloading all files before processing. Files are stored locally. This approach avoids downloading assets used more than once multiple times. All files are removed after the job completes.

- The virtual file system provides a significant performance boost for specific job profiles. In general, smaller subsets of total files with larger fleets of workers show the most benefit. Small numbers of files with fewer workers have roughly equivalent processing times.
- Virtual file system support is only available for Linux workers in service-managed fleets.
- The Deadline Cloud virtual file system supports the following operations, but is not POSIX compliant:

  - File `create`, `delete`, open, `close`, `read`, `write`, append, `truncate`, `rename`, `move`, `copy`, `stat`, `fsync`, and `falloc`
  - Directory `create`, `delete`, `rename`, `move`, `copy`, and `stat`
- The virtual file system is designed to reduce data transfer and improve performance when your tasks access only part of a large data set, and is not optimized for all workloads. You should test your workload before running production jobs.

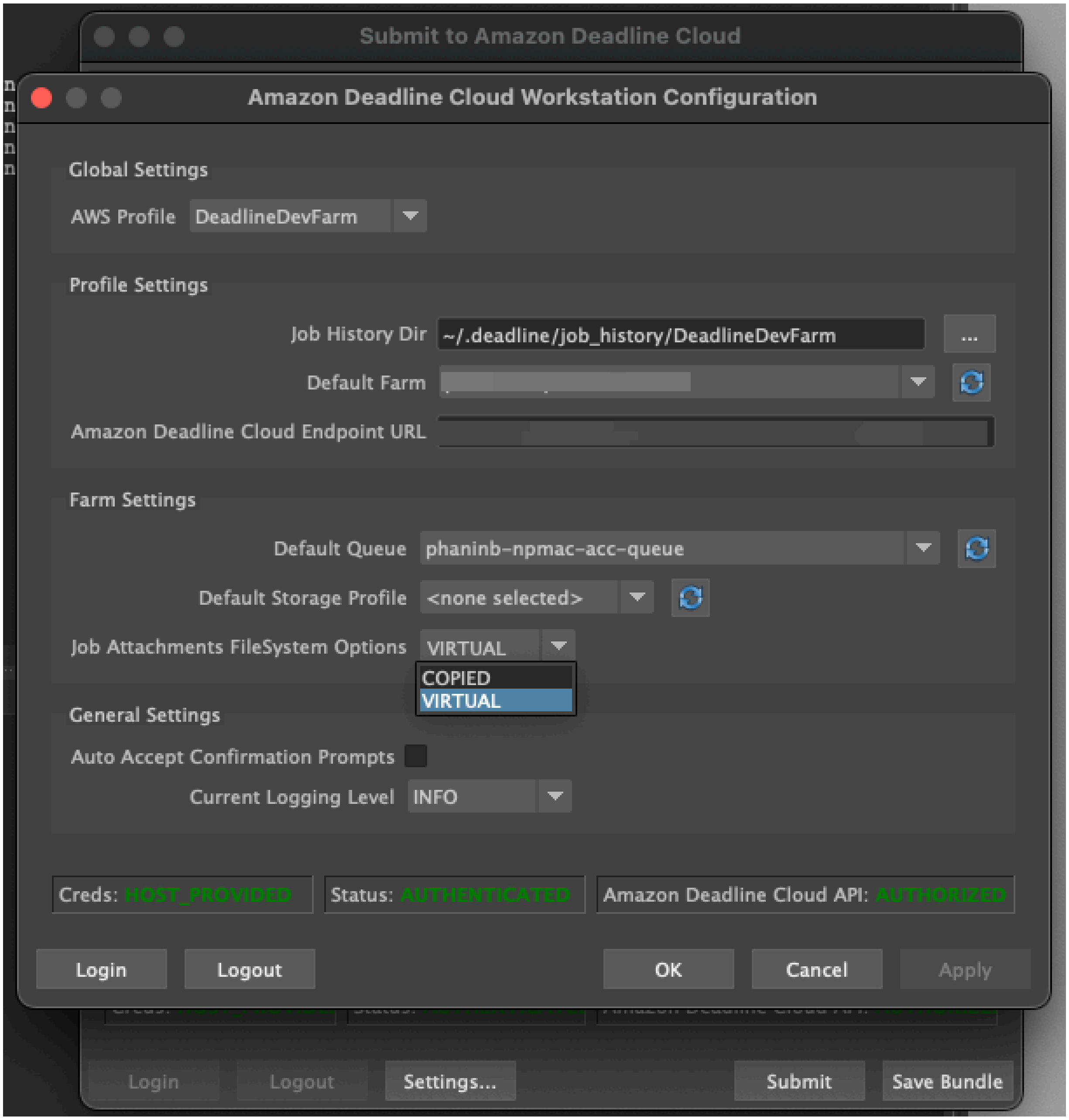
Enable VFS support

Virtual file system support (VFS) is enabled for each job. A job falls back to the default job attachments framework in these cases:

- A worker instance profile does not support a virtual file system.
- Problems prevent launching the virtual file system process.
- The virtual file system can't be mounted.

To enable virtual file system support using the submitter

1. When submitting a job, choose the **Settings** button to open the **AWS Deadline Cloud workstation configuration panel**.
2. From the **Job attachments filesystem options** dropdown, choose **VIRTUAL**.


Submit to Amazon Deadline Cloud
Amazon Deadline Cloud Workstation Configuration
COPIED
VIRTUAL
Login
Logout
OK
Cancel
Apply
Login
Logout
Settings...
Submit
Save Bundle

Global Settings

AWS Profile: DeadlineDevFarm

Profile Settings

Job History Dir: ~/.deadline/job_history/DeadlineDevFarm

Default Farm: [redacted]

Amazon Deadline Cloud Endpoint URL: [redacted]

Farm Settings

Default Queue: phaninb-npmac-acc-queue

Default Storage Profile: <none selected>

Job Attachments FileSystem Options: VIRTUAL

General Settings

Auto Accept Confirmation Prompts: □

Current Logging Level: INFO

Creds: HOST_PROVIDED

Status: AUTHENTICATED

Amazon Deadline Cloud API: AUTHORIZED

3. To save your changes, choose **OK**.

To enable virtual file system support using the AWS CLI

- Use the following command when you submit a saved job:

```
deadline bundle submit-job --job-attachments-file-system VIRTUAL
```

To verify that the virtual file system launched successfully for a particular job, review your logs in Amazon CloudWatch Logs. Look for the following messages:

```
Using mount_point mount_point
Launching vfs with command command
Launched vfs as pid PID number
```

If the log contains the following message, virtual file system support is disabled:

```
Virtual File System not found, falling back to COPIED for JobAttachmentsFileSystem.
```

Troubleshooting virtual file system support

You can view logs for your virtual file system using the Deadline Cloud monitor. For instructions, see View session and worker logs in Deadline Cloud.

Virtual file system logs are also sent to the CloudWatch Logs group that's associated with the queue shared with the worker agent output.

Automatic downloads

The Deadline CLI provides a command to download the output of all tasks in a queue that completed since the last time the same command ran. You can configure this as a cron job or scheduled task to run repeatedly. This configuration sets up automatic downloading of output on a continuous basis.

Before setting up automatic downloads, follow the steps in Storage profiles for job attachments to configure all paths of asset data for upload and download. If a job uses an output path that is not in its storage profile, then the automatic download skips downloading that output and prints warning messages to summarize the files it did not download. Similarly, if a job is submitted without a storage profile, the automatic download skips that job and prints a warning message. By default, Deadline Cloud submitters display warning messages for paths that are outside of storage profiles to help ensure correct configuration.

Configuring AWS credentials

Automatic downloads use the Deadline CLI to continuously download job outputs. To authenticate these downloads, you need long-term IAM credentials. Deadline Cloud monitor credentials expire, so you can't use them for this purpose.

Follow the steps below to set up long-term credentials.

Important

Heed the following warnings:

- **Do NOT** use your account's root credentials to access AWS resources. These credentials provide unrestricted account access and are difficult to revoke.
- **Do NOT** put literal access keys or credential information in your application files. If you do, you create a risk of accidentally exposing your credentials if, for example, you upload the project to a public repository.
- **Do NOT** include files that contain credentials in your project area.
- Secure your access keys. Do not provide your access keys to unauthorized parties, even to help find your account identifiers. By doing this, you might give someone permanent access to your account.
- Be aware that any credentials stored in the shared AWS credentials file are stored in plain text.

For more details, see Best practices for managing AWS access keys in the *AWS General Reference*.

Create an IAM user

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, select **Users** and then select **Create user**.
3. Name the user **`deadline-output-downloader`**. Clear the checkbox for **Provide user access to the AWS Management Console**, then choose **Next**.
4. Choose **Attach policies directly**.
5. Choose **Create policy** to create a custom policy with minimum required permissions.
6. In the JSON editor, specify the following permissions:

   JSON

```
{
                      "Version":"2012-10-17",
 "Statement": [
     {
         "Sid": "DeadlineCloudOutputDownload",
         "Effect": "Allow",
         "Action": [
             "deadline:AssumeQueueRoleForUser",
             "deadline:ListQueueEnvironments",
             "deadline:ListSessions",
             "deadline:ListSessionActions",
             "deadline:SearchJobs",
             "deadline:GetJob",
             "deadline:GetQueue",
             "deadline:GetStorageProfileForQueue"
         ],
         "Resource": "*"
     }
   ]
}
```

7. Name the policy **`DeadlineCloudOutputDownloadPolicy`** and choose **Create policy**.
8. Return to the user creation page, refresh the policy list, and select the **DeadlineCloudOutputDownloadPolicy** you just created, then choose **Next**.
9. Review the user details and then choose **Create user**.

Create an access key

1. From the user details page, select the **Security credentials** tab. In the **Access keys** section, choose **Create access key**.
2. Indicate that you want to use the key for Other, then choose **Next**, then choose **Create access key**.
3. On the **Retrieve access keys** page, choose **Show** to reveal the value of your user's secret access key. You can copy the credentials or download a .csv file.

Store the user access keys

- Store the user access keys in the AWS credentials file on your system:
  - On Linux, the file is located at `~/.aws/credentials`
  - On Windows, the file is located at `%USERPROFILE\.aws\credentials`

Replace the following keys:

```
[deadline-downloader]
aws_access_key_id=ACCESS_KEY_ID
aws_secret_access_key=SECRET_ACCESS_KEY
region=YOUR_AWS_REGION
```

> **Important**
>
> When you no longer need this IAM user, we recommend that you remove it to align with the AWS security best practice. We recommend that you require your human users to use temporary credentials through AWS IAM Identity Center when accessing AWS.

Prerequisites

Complete the following steps before creating a cron job or scheduled task for automatic download.

1. If you haven't already, install Python.
2. Install the Deadline CLI by running:

   ```
   python -m pip install deadline
   ```

3. Confirm the version of the Deadline CLI is 0.52.1 or newer with the following command.

   ```
   $ deadline --version
   deadline, version 0.52.1
   ```

Test the output download command

To verify the command works in your environment

1. Get the path to Deadline

Linux and macOS

```
$ which deadline
```

Windows

```
C:\> where deadline
```

PowerShell

```
PS C:\> Get-Command deadline
```

2. Run the sync-output command to bootstrap.

   ```
   /path/to/deadline queue sync-output \
   --profile deadline-downloader \
   --farm-id YOUR_FARM_ID \
   --queue-id YOUR_QUEUE_ID \
   --storage-profile-id YOUR_PROFILE_ID \
   --checkpoint-dir /path/to/checkpoint/directory \
   ```

3. You only need to do this step if your downloading machine is the same as submitting machine. Replace `--storage-profile-id YOUR_PROFILE_ID \` above with `--ignore-storage-profiles`.

4. Submit a test job.

   a. Download the .zip file from GitHub.

      i. Open the deadline-cloud-samples GitHub repository.

      ii. Choose **Code** and then, from the dropdown menu, select **Download ZIP**.

      iii. Unzip the downloaded archive to a local directory.

   b. Run

      ```
       cd /path/to/unzipped/deadline-cloud-samples-mainline/job_bundles/
      job_attachments_devguide_output
      ```

   c. Run

```
deadline bundle submit .
```

- If you don’t have the default deadline config setup, you might need to supply the following in the command line.

  ```
  --farm-id YOUR-FARM-ID --queue-id YOUR-QUEUE-ID
  ```

d. Wait for the job to complete before going to the next step.

5. Run the sync-output command again.

```
/path/to/deadline queue sync-output \
 --profile deadline-downloader \
 --farm-id YOUR_FARM_ID \
 --queue-id YOUR_QUEUE_ID \
 --storage-profile-id YOUR_PROFILE_ID \
 --checkpoint-dir /path/to/checkpoint/directory
```

6. Verify the following:

   - Your test job's outputs appear in the destination directory.
   - A checkpoint file is created in your specified checkpoint directory.

Set up scheduled downloads

Select the tab for your operating system to learn how to configure automatic downloads for every 5 minutes.

Linux

1. **Verify Deadline CLI Installation**

   Get the exact path to your deadline executable:

   ```
   $ which deadline
   ```

   Note this path (e.g., `/opt/homebrew/bin/deadline`) for use in the plist file.

2. **Create Checkpoint Directory**

Create the directory where checkpoint files will be stored. Ensure proper permissions for your user to run the command.

```
$ mkdir -p /path/to/checkpoint/directory
```

3. **Create Log Directory**

   Create a directory for cron job logs:

   ```
   $ mkdir -p /path/to/logs
   ```

   Consider setting up log rotate on the log file using https://www.redhat.com/en/blog/setting-logrotate

4. **Check Current Crontab**

   View your current crontab to see existing jobs:

   ```
   $ crontab -l
   ```

5. **Edit Crontab**

   Open your crontab file for editing:

   ```
   $ crontab -e
   ```

   If this is your first time, you may be prompted to choose an editor (nano, vim, etc.).

6. **Add Cron Job Entry**

   Add the following line to run the job every 5 minutes (replace paths with actual values from steps 1 and 2):

   ```
   */5 * * * * /path/to/deadline queue sync-output --profile deadline-downloader
    --farm-id YOUR_FARM_ID --queue-id YOUR_QUEUE_ID --storage-profile-id
    YOUR_PROFILE_ID --checkpoint-dir /path/to/checkpoint/directory >> /path/to/
   logs/deadline_sync.log 2>&1
   ```

7. **Verify Cron Job Installation**

   After saving and exiting the editor, verify the cron job was added:

```
$ crontab -l
```

You should see your new job listed.

8. **Check Cron Service Status**

   Ensure the cron service is running:

   ```
   # For systemd systems (most modern Linux distributions)
   $ sudo systemctl status cron
   # or
   $ sudo systemctl status crond

   # For older systems
   $ sudo service cron status
   ```

   If not running, start it:

   ```
   $ sudo systemctl start cron
   $ sudo systemctl enable cron  # Enable auto-start on boot
   ```

macOS

1. **Verify Deadline CLI Installation**

   Get the exact path to your deadline executable:

   ```
   $ which deadline
   ```

   Note this path (e.g., `/opt/homebrew/bin/deadline`) for use in the plist file.

2. **Create Checkpoint Directory and Log Directory**

   Create the directory where checkpoint files will be stored:

   ```
   $ mkdir -p /path/to/checkpoint/directory
   $ mkdir -p /path/to/logs
   ```

Consider setting up log rotate on the log file using https://formulae.brew.sh/formula/logrotate

3. **Create a Plist file**

   Create a configuration file at `~/Library/LaunchAgents/com.user.deadlinesync.plist` with the following content (replace `/path/to/deadline` with the actual path from step 1):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.user.deadlinesync</string>
    <key>ProgramArguments</key>
    <array>
        <string>/path/to/deadline</string>
        <string>queue</string>
        <string>sync-output</string>
        <string>--profile</string>
        <string>deadline-downloader</string>
        <string>--farm-id</string>
        <string>YOUR_FARM_ID</string>
        <string>--queue-id</string>
        <string>YOUR_QUEUE_ID</string>
        <string>--storage-profile-id</string>
        <string>YOUR STORAGE PROFILE ID</string>
        <string>--checkpoint-dir</string>
        <string>/path/to/checkpoint/dir</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>UserName</key>
    <string>YOUR_USER_NAME</string>
    <key>StandardOutPath</key>
    <string>/path/to/logs/deadline_sync.log</string>
    <key>StartInterval</key>
    <integer>300</integer>
</dict>
</plist>
```

Replace `--storage-profile-id` *`YOUR_PROFILE_ID`* above with `--ignore-storage-profiles` if your downloading machine is the same as submitting machine.

4. **Validate Plist File**

   Validate the XML syntax of your plist file:

   ```
   $ plutil -lint ~/Library/LaunchAgents/com.user.deadlinesync.plist
   ```

   This should return "OK" if the file is valid.

5. **Check for Existing Launch Agents or Launch Daemons**

   Check if a launch agent is already loaded:

   ```
   $ launchctl list | grep deadlinesync
   OR
   $ sudo launchctl list | grep deadlinesync
   ```

   If one exists, unload it first:

   ```
   $ launchctl bootout gui/$(id -u)/com.user.deadlinesync
   OR
   $ sudo launchctl bootout system/com.user.deadlinesync
   ```

6. **Create and bootstrap**

   To run this task while the user is logged in, run it as **LaunchAgent**. To run this task without a user being logged in every time the machine is running, run it as a **LaunchDaemon**.

   a. To run as **LaunchAgent:**

      i. Use the configuration created under `~/Library/LaunchAgents/com.user.deadlinesync.plist`

      ii. Then load the configuration using the bootstrap command:

      ```
      $ launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/
      com.user.deadlinesync.plist
      ```

   b. To run as **LaunchDaemon**:

i. Move the Pilst file and change permissions by running the following:

```
$ sudo mv ~/Library/LaunchAgents/com.user.deadlinesync.plist /Library/
LaunchDaemons/
$ sudo chown root:wheel /Library/LaunchDaemons/
com.user.deadlinesync.plist
$ sudo chmod 644 /Library/LaunchDaemons/com.user.deadlinesync.plist
```

ii. Load the launch agent using the modern bootstrap command:

```
$ sudo launchctl bootstrap system /Library/LaunchDaemons/
com.user.deadlinesync.plist
```

7. **Verify Status**

If you bootstrapped a LaunchAgent run the following to confirm it's loaded:

```
$ launchctl list | grep deadlinesync
```

If you bootstrapped a LaunchDaemon, confirm it is loaded by running:

```
$ sudo launchctl list | grep deadlinesync
```

The output should look like

```
SOME_PID_NUMBER 0 com.user.deadlinesync
```

For detailed status information:

```
$ launchctl print gui/$(id -u)/com.user.deadlinesync
```

This shows the current state, program arguments, environment variables, run interval, and execution history.

Windows

> **Note**
>
> The scheduled task created using these instructions only work when the user is logged in.
> To set it up at system startup without requiring user login, see the official Windows documentation.

For all steps below use Command Prompt - run as Administrator:

1. **Verify Deadline CLI Installation**

   Find the deadline executable:

   ```
   C:\> where deadline
   ```

   Note the full path (e.g., `C:\Program Files\Amazon\DeadlineCloud\deadline.exe`) for use in the task.
2. **Create Checkpoint Directory**

   Create the directory where checkpoint files will be stored:

   ```
   C:\> mkdir "path\to\checkpoint\directory"
   ```
3. **Create Log Directory**

   Create a directory for task logs:

   ```
   C:\> mkdir "path\to\logs"
   ```
4. **Create Batch File Wrapper**

   Create the batch file with the following content:

   ```
   C:\> notepad C:\path\to\deadline_sync.bat
   ```

   ```
   YOUR_PATH_TO_DEADLINE.EXE queue sync-output --profile deadline-downloader
    --farm-id YOUR_FARM_ID --queue-id YOUR_QUEUE_ID --storage-profile-
   ```

```
id YOUR_PROFILE_ID --checkpoint-dir path\to\checkpoint\checkpoints > path\to
\logs\deadline.log 2>&1
```

5. **Test Batch File**

   Test the batch file manually:

   ```
   C:\> .\path\to\deadline_sync.bat
   ```

   Check the log file was created:

   ```
   C:\> notepad path\to\logs\deadline_sync.log
   ```

6. **Check Task Scheduler Service**

   Ensure Task Scheduler service is running:

   ```
   C:\> sc query "Schedule"
   ```

   If the service doesn't exist, try alternative names:

   ```
   C:\> sc query "TaskScheduler"
   C:\> sc query "Task Scheduler"
   ```

   If not running, start it:

   ```
   C:\> sc start "Schedule"
   ```

7. **Create Scheduled Task**

   Create the task to run every 5 minutes.

   ```
   C:\> schtasks /create /tn "DeadlineOutputSync" /tr "C:\path\to
   \deadline_sync.bat" /sc minute /mo 5
   ```

   Command breakdown:

   - `/tn` - Task name
   - `/tr` - Task to run (your batch file)

  - `/sc minute /mo 5` - Schedule: every 5 minutes

8. **Verify Task Creation**

   Check that the task was created successfully:

   ```
   schtasks /query /tn "DeadlineOutputSync" /v /fo LIST
   ```

   Look for:

   - **Task To Run**: Should show your batch file path
   - **Next Run Time**: Should show a time within 5 minutes

9. **Test Task Execution**

   Run the task manually to test:

   ```
   schtasks /run /tn "DeadlineOutputSync"
   ```

   Check task status:

   ```
   schtasks /query /tn "DeadlineOutputSync"
   ```

Verify the setup

To verify the automatic downloads setup was successful, complete the following steps.

1. Submit a new test job.
2. Wait for one scheduler interval to complete, which in this case is 5 minutes.
3. Verify that new outputs are downloaded automatically.

If the outputs do not download, check the Troubleshooting section for the process logs.

Troubleshooting automatic downloads

If you encounter issues with the automatic downloads, check the following:

Storage Profile Issues

- An error like `[Errno 2] No such file or directory` or `[Errno 13] Permission denied` in the log file could be related to missing or misconfigured storage profiles.
- See Storage profiles for information about how to set up your storage profiles when the downloading machine is different from the submitting machine.
- For same-machine downloads, try the `--ignore-storage-profiles` flag.

Directory Permissions

- Ensure the scheduler service user has:
  - Read/write access to the checkpoint directory
  - Write access to the output destination directory
- For Linux and macOS, use `ls -la` to check permissions.
- For Windows, review Security settings in the Properties folder.

Checking Scheduler Logs

Linux

1. Check if cron service is running:

   ```
   # For systemd systems
   $ sudo systemctl status cron
   # or
   $ sudo systemctl status crond

   # Check if your user has cron job correctly configured
   $ crontab -l
   ```

2. View cron execution logs:

   ```
   # Check system logs for cron activity (most common locations)
   $ sudo tail -f /var/log/syslog | grep CRON
   $ sudo tail -f /var/log/cron.log | grep deadline
   ```

```
# View recent cron logs
$ sudo journalctl -u cron -f
$ sudo journalctl -u crond -f  # On some systems
```

3. Check your specific cron job logs:

```
# View the log file specified in your cron job
$ tail -100f /path/to/logs/deadline_sync.log
```

4. Search for cron job execution in system logs:

```
# Look for your specific cron job executions
$ sudo grep "deadline.*incremental-output-download" /var/log/syslog

# Check for cron job starts and completions
$ sudo grep "$(whoami).*CMD.*deadline" /var/log/syslog
```

5. Check checkpoint file updates:

```
# List checkpoint files with timestamps
$ ls -la /path/to/checkpoint/directory/

# Check when checkpoint was last modified
$ stat /path/to/checkpoint/directory/queue-*_download_checkpoint.json
```

6. Check the log file:

```
$ ls -la /path/to/log/deadline_sync.log
```

macOS

Viewing Launch Agent Execution Logs:

1. Check if the launch agent is running:

   ```
   $ sudo launchctl list | grep deadlinesync
   ```

   Output shows: `PID Status Label` (PID will be - when not currently running, which is normal for interval jobs)

2. View detailed launch agent status:

   ```
   $ sudo launchctl print system/com.user.deadlinesync
   ```

   This shows execution history, last exit code, number of runs, and current state.

3. View launch agent execution logs:

   ```
   # View recent logs (last hour)
   log show --predicate 'subsystem contains "com.user.deadlinesync"' --last 1h

   # View logs from a specific time period
   log show --predicate 'subsystem contains "com.user.deadlinesync"' --start
    '2024-08-27 09:00:00'
   ```

4. Force run the launch agent for immediate testing:

   ```
   $ sudo launchctl kickstart gui/$(id -u)/com.user.deadlinesync
   ```

   This immediately triggers the job regardless of the schedule, useful for testing.

5. Check checkpoint file updates:

   ```
   # List checkpoint files with timestamps
   $ ls -la /path/to/checkpoint/directory/
   ```

6. Check the log file:

   ```
   $ ls -la /path/to/log/deadline_sync.log
   ```

Windows

1. Check if Task Scheduler service is running:

   ```
   C:\> sc query "Schedule"
   ```

   If the service doesn't exist, try alternative names:

   ```
   C:\> sc query "TaskScheduler"
   C:\> sc query "Task Scheduler"
   ```

2. View your scheduled tasks:

   ```
   C:> schtasks /query /tn "DeadlineOutputSync"
   ```

3. Check your task's log file:

   ```
   # View the log file created by your batch script
   C:> notepad C:\path\to\logs\deadline_sync.log
   ```

4. Check checkpoint file updates:

   ```
   # List checkpoint files with timestamps
   C:> dir "C:\path\to\checkpoint\directory" /od
   ```

Track spending and usage for Deadline Cloud farms

The AWS Deadline Cloud budget manager and usage explorer are cost management tools that provide the approximate cost of using Deadline Cloud based on available information about cost variables. The cost management tools don't guarantee the amount owed for your actual use of Deadline Cloud and other AWS services.

To help you manage costs for Deadline Cloud, you can use the following features:

- **Budget manager** – With the Deadline Cloud budget manager, you can create and edit budgets to help manage project costs.
- **Usage explorer** – With the Deadline Cloud usage explorer, you can view how many AWS resources are used and the estimated costs for those resources.
- **AWS cost allocation tags** – With cost allocation tags, you can track detailed costs for all of your AWS services. For more information, see Organizing and tracking costs using AWS cost allocation tags.

Cost assumptions

The basic calculation used by the Deadline Cloud cost management tools is:

```
Cost per job =
    (CMF run time x CMF compute rate) +
    (SMF run time x SMF compute rate) +
    (License run time x license rate)
```

- *Run time* is the sum of all tasks in a job, from start time to end time.
- *Compute rate* is determined by the AWS Deadline Cloud pricing for service-managed fleets. For customer-managed fleets, the compute rate is estimated to be $1 per worker hour.
- *License rate* is determined by the Deadline Cloud base license price and is only available for service-managed fleets. Additional tiers are not included. For more information about license pricing, see AWS Deadline Cloud pricing.

The cost estimate from the Deadline Cloud cost management tools may vary from your actual costs for a number of reasons. Common reasons include:

- *Customer owned resources and their pricing*. You can choose to bring your own resources, either from AWS or externally from on-premises or other cloud providers. Actual costs of these resources are not calculated.
- *Idle worker costs*. Idle worker costs are not included when the worker status is IDLE. This situation can happen for fleets with a minimum instance count greater than zero, or when workers transition between jobs. Idle worker cost are not included in calculations.
- *Worker stop and start time*. After workers complete a job, the cost for moving from IDLE to STOPPING and from STOPPING to STOPPED is not included in Deadline Cloud cost estimates.
- *Promotional credits, discounts, and custom pricing agreements*. The cost management tools don't account for promotional credits, private pricing agreements, or other discounts. You may be eligible for other discounts that are not part of the estimate.
- *Asset storage*. Asset storage is not included in the cost and usage estimates.
- *Changes in price*. AWS offers pay-as-you-go pricing for most services. Prices may change over time. The cost management tools use the most up-to-date prices publicly available, but there may be delays after changes.
- *Taxes*. The cost management tools don't include taxes applied to our purchase of the service.
- *Rounding*. The cost management tool perform mathematical rounding of pricing data.
- *Currency*. Cost estimates are made in U.S. dollars. Global exchange rates vary over time. If you translate estimates to a different currency base on the current exchange, changes in the exchange rate affect the estimate.
- *Outside licensing*. If you choose to use pre-purchased licences (Software licensing for service-managed fleets), Deadline Cloud cost management tools can't account for this cost.

Control costs with a budget

The Deadline Cloud budget manager helps you control spending on a given resource, such as a queue, fleet, or farm. You can create budget amounts and limits, and set automated actions to help reduce or stop additional spending against the budget.

The following sections provide you with the steps for using the Deadline Cloud budget manager.

Topics

- Create a budget for a Deadline Cloud queue
- View a Deadline Cloud queue budget
- Edit a budget for a Deadline Cloud queue
- Deactivate a budget for a Deadline Cloud queue
- Monitor a budget with EventBridge events

Prerequisite

To use the Deadline Cloud budget manager, you must have `OWNER` access level. To grant `OWNER` permission, follow the steps in Managing users in Deadline Cloud.

Open the Deadline Cloud budget manager

To open the Deadline Cloud budget manager, use the following procedure.

1. Sign in to the AWS Management Console and open the Deadline Cloud console.
2. Choose **View farms**.
3. Locate the farm that you want to get information about, then choose **Manage jobs**.
4. In the Deadline Cloud monitor, in the left navigation pane, choose **Budgets**.

The budget manager summary page displays a list of both active and inactive budgets:

- **Active** budgets track against the selected resource (a queue).
- **Inactive** budgets have either expired or been canceled by a user, and are no longer tracking costs against this budget's limits.

After you choose a budget, the budget summary page contains basic information about the budget. Information provided includes the budget name, status, resources, remaining percentage, remaining amount, total budget, start date, and end date.

Create a budget for a Deadline Cloud queue

To create a budget, use the following procedure.

1. If you haven't already, sign in to the AWS Management Console, open the Deadline Cloud console, choose a farm, and then choose **Manage jobs**.

2. From the **Budget manager** page, choose **Create budget**.
3. In the details section, enter a **Budget name** for the budget.
4. (Optional) In the description field, enter a brief description of the budget.
5. From **Resource**, use the **Queue** dropdown to select the queue that you want to create a budget for.
6. For **Period**, set the start and end date for the budget by completing the following steps:

   a. For **Start date**, enter the first date of the budget tracking in YYYY/MM/DD format, or choose the **calendar** icon and select a **date**.

      The default start date is the date that the budget is created.

   b. For **End date**, enter the last date of the budget tracking in YYYY/MM/DD format or choose the **calendar** icon and select a **date**.

      The default end date is 120 days from the start date.
7. For **Budget amount**, enter the dollar amount of the budget.
8. (Optional) We recommend that you create limit alerts. In the **Limit actions** section, you can implement automated actions that occur when specific amounts remain in the budget. To do this, complete the following steps:

   a. Choose **Add new action**.

   b. For **Remaining amount**, enter the dollar amount that you want to start the action.

   c. In the **Action** dropdown, choose the action that you want. Actions include:

      - **Stop after finishing current work –** All work currently running when the threshold amount is met continue to run (and incur costs) until finished.
      - **Immediately stop work –** All work is canceled immediately when the threshold amount is met.

   d. To create additional limit alerts, choose **Add new action** and repeat the previous steps.
9. Choose **Create budget**.

View a Deadline Cloud queue budget

After you create a budget, you can view the budget on the **Budget manager** page. From there, you can view the budget's total amount and the overall cost allocated to the specific budget.

To view a budget, use the following procedure.

1. If you haven't already, sign in to the AWS Management Console, open the Deadline Cloud console, choose a farm, and then choose **Manage jobs**.
2. Choose **Budgets** from the left side navigation pane. The **Budget Manager** page appears.
3. To view an active budget, choose the **Active budgets** tab, and choose the name of the budget that you want to view. The budget details page appears.
4. To view the budget details for an expired budget, choose the **Inactive budgets** tab. Then, choose the name of the budget that you want to view. The budget details page appears.

Edit a budget for a Deadline Cloud queue

You can edit any active budget. To edit an active budget, use the following procedure.

1. If you haven't already, sign in to the AWS Management Console, open the Deadline Cloud console, choose a farm, and then choose **Manage jobs**.
2. From the **Budget Manager** page, in the **Active budgets** tab, choose the button next to the budget you want to edit.
3. From the **Actions** dropdown menu, select **Edit budget**.
4. Make the changes that you want, and then choose **Update budget**.

Deactivate a budget for a Deadline Cloud queue

You can deactivate any active budget. Deactivating a budget changes its status from **Active** to **Inactive**. When a budget is deactivated, it no longer tracks a resource to that budget’s amount.

To deactivate a budget, use the following procedure.

1. If you haven't already, sign in to the AWS Management Console, open the Deadline Cloud console, choose a farm, and then choose **Manage jobs**.
2. From the **Budget manager** page, in the **Active Budgets** tab, choose the button next to the budget that you want to deactivate.
3. From the **Actions** dropdown menu, select **Deactivate budget**. In a few moments, the selected budget will change from **Active** to **Inactive** and will move from the **Active Budgets** tab to the **Inactive Budgets** tab.

Monitor a budget with EventBridge events

Deadline Cloud sends budget-related events, using Amazon EventBridge, to your default EventBridge event bus. You can create custom functions that receive the events and act on them to send notifications to automatically notify users via email, Slack, or other channels when a budget reaches predefined levels. For example, you can send SMS messages when a budget reaches a certain threshold. These notifications help you stay on top of your spending and make informed decisions before your budget is exhausted.

Deadline Cloud periodically aggregates usage and cost data for each render farm. Then it checks to see if any of the budget thresholds has been crossed. If a threshold is crossed, Deadline Cloud triggers an event to alert you so that you can take the appropriate action. An event is triggered whenever a budget crosses one of these thresholds, specified in percent of the budget used:

- 10, 20, 30, 40, 50, 60, 70, 75, 80, 85, 90, 95, 96, 97, 98, 99, 100

The budget usage thresholds get closer together as a budget approaches 100 percent usage. This frequency helps you closely monitor usage as the budget reaches its limit. You can also set your own budget thresholds. Deadline Cloud sends an event when usage passes your custom thresholds. After your budget reaches 100 percent, Deadline Cloud stops sending events. If you adjust your budget, Deadline Cloud sends events for your thresholds based on the new budget amount.

You can use the EventBridge console (https://console.aws.amazon.com/events/) to create rules to send the Deadline Cloud events to the appropriate target for the event. For example, you can send the event to an Amazon Simple Queue Service queue and from there to multiple targets, such as AWS End User Messaging SMS or a Amazon Relational Database Service database for logging.

For examples of an EventBridge rule, see the following topics:

- Send an email when events happen using Amazon EventBridge.
- Creating an Amazon EventBridge rule that sends notifications to Amazon Q Developer in chat applications.
- Getting started with Amazon EventBridge.

For more information about budget events, see the Budget Threshold Reached event in the *Deadline Cloud Developer Guide*.

Track usage and costs with the Deadline Cloud usage explorer

With the Deadline Cloud usage explorer, you can see real-time metrics on the activity happening on each farm. You can look at the farm's costs by different variables, such as queue, job, license product, or instance types. Select various time frames to see usage during a specific period of time, and look at usage trends over the course of time. You can also see a detailed breakdown of selected data points, allowing for a closer look into metrics. Usage can be shown by time (minutes and hours) or by cost ($USD).

The following sections show you the steps for accessing and using the Deadline Cloud usage explorer.

Topics

Prerequisite

To use the Deadline Cloud usage explorer, you must have either `MANAGER` or `OWNER` farm permissions. For more information, see Understanding access levels.

> **Note**
>
> If your time zone doesn't align to a full hour, such as India Standard Time (UTC+5:30), the usage explorer doesn't show usage metrics. To see metrics, set your time zone to a zone that aligns to a full hour.

Open the usage explorer

To open the Deadline Cloud usage explorer, use the following procedure.

1. Sign in to the AWS Management Console and open the Deadline Cloud console.
2. To see all available farms, choose **View farms**.
3. Locate the farm that you want to get information about, then choose **Manage jobs**. The Deadline Cloud monitor opens in a new tab.

4. In the Deadline Cloud monitor, from the left menu, select **Usage explorer**.

Use the usage explorer

From the usage explorer page, you can select specific parameters in which the data can be displayed. By default, you see total usage in time (hours and minutes) within the last 7 days. You can change these parameters, and the information displayed changes dynamically in accordance to the parameter settings.

You can group the results based on the queue, job, user, compute usage, instance type, or license product. If you choose license product, costs are calculated for specific licenses. For all other groups the time is calculated by adding up the time taken for each task to run.

The usage explorer returns only 100 results based on the filter criteria that you set. The results are listed in descending order by the date created timestamp. If there are more than 100 results, you get an error message. You can refine your query to reduce the number of results:

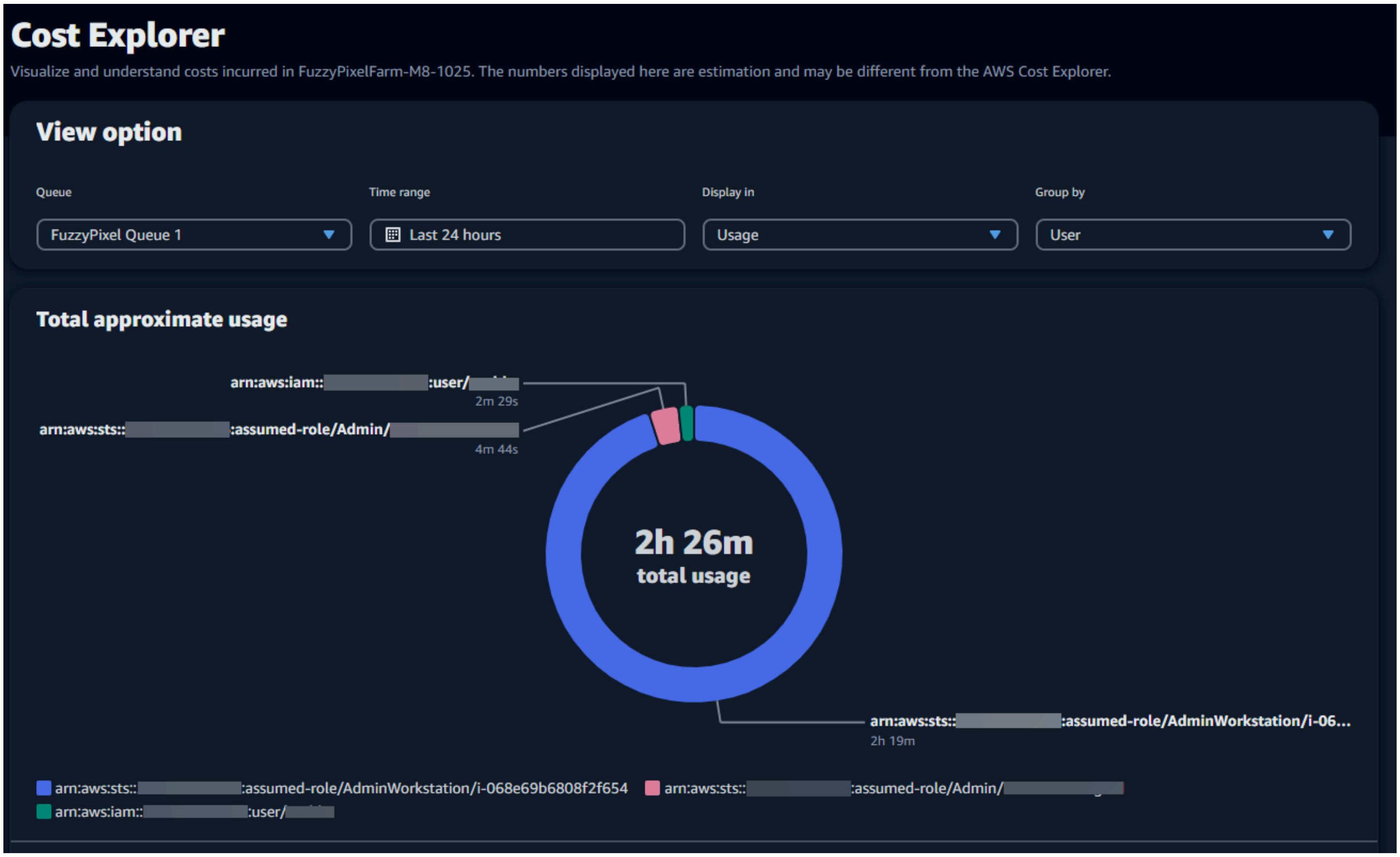
- Select a smaller time range
- Select fewer queues
- Select a different grouping, such as grouping by queue instead of job

Topics

Use visual graphs to review data

You can review data in a visual format to identify trends and potential areas that might need more analysis or attention. Usage explorer offers a pie chart that displays overall usage and cost with the option to group the totals into smaller subtotals.

> **Note**
>
> The chart *only* displays the top five results with other results combined in an "others" section. You can view all results in the breakdown section below the chart.



Cost Explorer

Visualize and understand costs incurred in FuzzyPixelFarm-M8-1025. The numbers displayed here are estimation and may be different from the AWS Cost Explorer.

View option

Queue: FuzzyPixel Queue 1

Time range: Last 24 hours

Display in: Usage

Group by: User

Total approximate usage

arn:aws:iam::[redacted]:user/[redacted]
2m 29s
arn:aws:sts::[redacted]:assumed-role/Admin/[redacted]
4m 44s
2h 26m
total usage
arn:aws:sts::[redacted]:assumed-role/AdminWorkstation/i-06...
2h 19m
🟦 arn:aws:sts::[redacted]:assumed-role/AdminWorkstation/i-068e69b6808f2f654
🟥 arn:aws:sts::[redacted]:assumed-role/Admin/[redacted]
🟩 arn:aws:iam::[redacted]:user/[redacted]

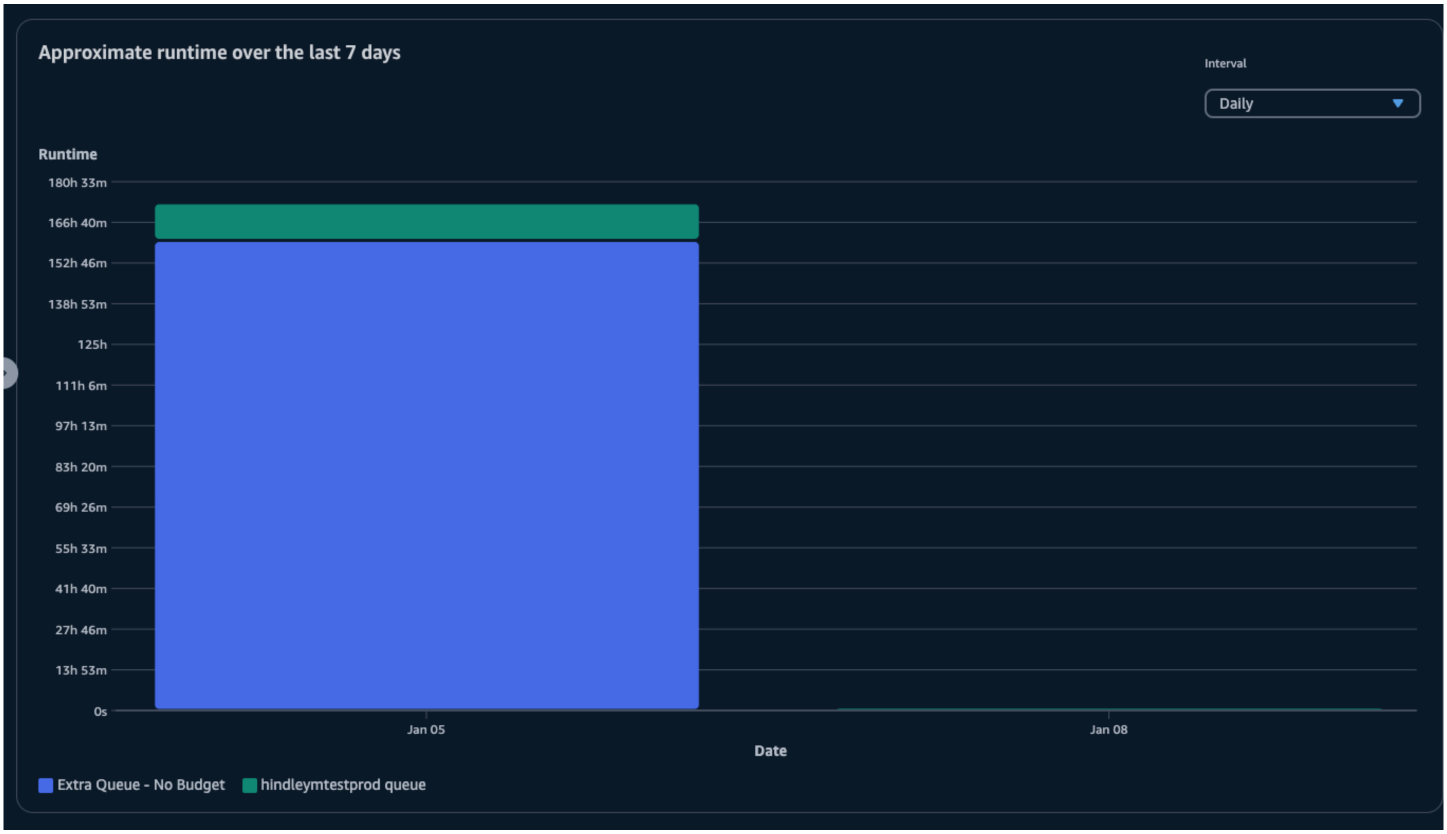
View a breakdown of metrics

Beneath the pie chart, usage explorer offers a more detailed breakdown of specific metrics, which will change as parameters change. By default, five results display in the usage explorer. You can scroll through results using the pagination arrows in the breakdown section.

Breakdown is minimized by default. To expand and display the results, select the **View all breakdown** arrow. To download the breakdown, choose **Download data**.

View approximate runtime of queues

You can also view the approximate runtime of your queues based on different intervals that you specify. The interval options are hourly, daily, weekly, and monthly. After you select an interval, the graph displays the approximate runtime of your queues.



Cost management

AWS Deadline Cloud provides budgets and the usage explorer to help you control and visualize costs for your jobs. However, Deadline Cloud uses other AWS services, such as Amazon S3. Costs for those services are not reflected in Deadline Cloud budgets or the usage explorer and are charged separately based on usage. Depending on how you configure Deadline Cloud, you may use the following AWS services, as well as others:

Service	Pricing page
Amazon CloudWatch Logs	Amazon CloudWatch Logs pricing
Amazon Elastic Compute Cloud	Amazon Elastic Compute Cloud pricing
AWS Key Management Service	AWS Key Management Service pricing
AWS PrivateLink	AWS PrivateLink pricing
Amazon Simple Storage Service	Amazon Simple Storage Service pricing

Service	Pricing page
Amazon Virtual Private Cloud	Amazon Virtual Private Cloud pricing

Cost management best practices

Using the following best practices can help you understand and control your costs when using Deadline Cloud and the tradeoffs you can make between cost and efficiency.

> **Note**
>
> The final cost of using Deadline Cloud depends on the interaction between a number of AWS services, the amount of work that you process, and the AWS Region where you run your jobs. The following best practices are guidelines and may not significantly reduce costs.

Best practices for CloudWatch Logs

Deadline Cloud sends worker and task logs to CloudWatch Logs. You are charged to collect, store, and analyze these logs. You can reduce costs by logging only the minimum amount of data required to monitor your tasks.

When you create a queue or fleet, Deadline Cloud creates a CloudWatch Logs log group with the following names:

- `/aws/deadline/<FARM_ID>/<FLEET_ID>`
- `/aws/deadline/<FARM_ID>/<QUEUE_ID>`

By default, these logs never expire. You can adjust the retention policy of log groups to remove old logs and help reduce storage costs. You can also export logs to Amazon S3. Amazon S3 storage costs are lower than those for CloudWatch. For more information, see Exporting log data to Amazon S3.

Best practices for Amazon EC2

You can use Amazon EC2 instances for both service-managed and customer-managed fleets. There are three considerations:

- For service-managed fleets, you can choose to have one or more instances available at all times by setting the minimum worker count for the fleet. When you set the minimum worker count above 0, the fleet always has this many workers running. This setting can reduce the amount of time that it takes for Deadline Cloud to start processing jobs, however you are charged for the instance's idle time.
- For service-managed fleets, set a maximum size for the fleet. This setting limits the number of instances that a fleet can auto scale to. Fleets won't grow past this size even if there are more jobs waiting to be processed.
- For both service-managed and customer-managed fleets, you can specify the Amazon EC2 instance types in your fleets. Using smaller instances costs less per minute, but may take longer to complete a job. Conversely, a larger instance costs more per minute, but can reduce the time to complete a job. Understanding the demands that your jobs place on an instance can help reduce your costs.
- When possible, choose Amazon EC2 Spot instances for your fleet. Spot instances are available for a reduced price, but may be interrupted by on-demand requests. On-demand instances are charged by the second and are not interrupted.

Best practices for AWS KMS

By default, Deadline Cloud encrypts you data with an AWS owned key. You are not charged for this key.

You may choose to use a customer managed key to encrypt your data. When you use your own key, you are charged based on how your key is used. If you use an existing key, this will be an incremental cost for the additional use.

Best practices for AWS PrivateLink

You can use AWS PrivateLink to create a connection between your VPC and Deadline Cloud using an interface endpoint. When you create a connection, you can call all of the Deadline Cloud API actions. You are charged per hour for each endpoint that you create. If you use PrivateLink, you must create at least three endpoints, and depending on your configuration, you may need as many as five.

Best practices for Amazon S3

Deadline Cloud uses Amazon S3 to store assets for processing, job attachments, output, and logs. To reduce the costs associated with Amazon S3, reduce the amount of data that you store. Some suggestions:

- Only store assets that are currently in use or that will be used shortly.
- Use an S3 Lifecycle configuration to automatically delete unused files from an S3 bucket.

Best practices for Amazon VPC

When you use usage-based licensing for your customer-managed fleet, you create a Deadline Cloud license endpoint, which is a Amazon VPC endpoint created in your account. This endpoint is charged at an hourly rate. To reduce costs, remove the endpoints when you are not using usage-based licenses.

Security in Deadline Cloud

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Deadline Cloud, see AWS services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Deadline Cloud. The following topics show you how to configure Deadline Cloud to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Deadline Cloud resources.

Topics

Data protection in Deadline Cloud

The AWS shared responsibility model applies to data protection in AWS Deadline Cloud. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Deadline Cloud or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

The data entered into name fields in Deadline Cloud job templates may also be included in billing or diagnostic logs and should not contain confidential or sensitive information.

Topics

Encryption at rest

AWS Deadline Cloud protects sensitive data by encrypting it at rest using encryption keys stored in AWS Key Management Service (AWS KMS). Encryption at rest is available in all AWS Regions where Deadline Cloud is available.

Encrypting data means sensitive data saved on disks isn't readable by a user or application without a valid key. Only a party with a valid managed key can decrypt the data.

Deadline Cloud deletes Amazon Elastic Block Store volumes when service-managed fleet worker instances terminate.

For information about how Deadline Cloud uses AWS KMS for encrypting data at rest, see Key management.

Encryption in transit

For data in transit, AWS Deadline Cloud uses Transport Layer Security (TLS) 1.2 or 1.3 to encrypt data sent between the service and workers. We require TLS 1.2 and recommend TLS 1.3. Additionally, if you use a virtual private cloud (VPC), you can use AWS PrivateLink to establish a private connection between your VPC and Deadline Cloud.

Key management

When creating a new farm, you can choose one of the following keys to encrypt your farm data:

- **AWS owned KMS key –** Default encryption type if you don't specify a key when you create the farm. The KMS key is owned by AWS Deadline Cloud. You can't view, manage, or use AWS owned keys. However, you don't need to take any action to protect the keys that encrypt your data. For more information, see AWS owned keys in the *AWS Key Management Service developer guide*.

- **Customer managed KMS key** – You specify a customer managed key when you create a farm. All of the content within the farm is encrypted with the KMS key. The key is stored in your account and is created, owned, and managed by you and AWS KMS charges apply. You have full control over the KMS key. You can perform such tasks as:
  - Establishing and maintaining key polices
  - Establishing and maintaining IAM policies and grants
  - Enabling and disabling key policies
  - Adding tags
  - Creating key aliases

  You can't manually rotate a customer owned key used with a Deadline Cloud farm. Automatic rotation of the key is supported.

  For more information, see Customer owned keys in the *AWS Key Management Service Developer Guide*.

  To create a customer managed key, follow the steps for Creating symmetric customer managed keys in the *AWS Key Management Service Developer Guide*.

How Deadline Cloud uses AWS KMS grants

Deadline Cloud requires a grant to use your customer managed key. When you create a farm encrypted with a customer managed key, Deadline Cloud creates a grant on your behalf by sending a `CreateGrant` request to AWS KMS to get access to the KMS key that you specified.

Deadline Cloud uses multiple grants. Each grant is used by a different part of Deadline Cloud that needs to encrypt or decrypt your data. Deadline Cloud also uses grants to allow access to other AWS services used to store data on your behalf, such as Amazon Simple Storage Service, Amazon Elastic Block Store, or OpenSearch.

Grants that enable Deadline Cloud to manage machines in a service-managed fleet include a Deadline Cloud account number and role in the `GranteePrincipal` instead of a service principal. While not typical, this is necessary to encrypt Amazon EBS volumes for workers in service-managed fleets using the customer managed KMS key specified for the farm.

Customer managed key policy

Key policies control access to your customer managed key. Each key must have exactly one key policy that contains statements that determine who can use the key and how they can use it. When you create you customer managed key, you can specify a key policy. For more information, see Managing access to customer managed keys in the *AWS Key Management Service Developer Guide*.

Minimal IAM policy for CreateFarm

To use your customer managed key to create farms using the console or the `CreateFarm` API operation, the following AWS KMS API operations must be permitted:

- `kms:CreateGrant` – Adds a grant to a customer managed key. Grants console access to a specified AWS KMS key. For more informations, see Using grants in the *AWS Key Management Service developer guide*.
- `kms:Decrypt` – Allows Deadline Cloud to decrypt data in the farm.
- `kms:DescribeKey` – Provides the customer managed key details to allow Deadline Cloud to validate the key.
- `kms:GenerateDataKey` – Allows Deadline Cloud to encrypt data using a unique data key.

The following policy statement grants the necessary permissions for the `CreateFarm` operation.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineCreateGrants",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234567890abcdef0",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

Minimal IAM policy for read-only operations

To use your customer managed key for read-only Deadline Cloud operations, such getting information about farms, queues, and fleets. The following AWS KMS API operations must be permitted:

- `kms:Decrypt` – Allows Deadline Cloud to decrypt data in the farm.
- `kms:DescribeKey` – Provides the customer managed key details to allow Deadline Cloud to validate the key.

The following policy statement grants the necessary permissions for read-only operations.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadOnly",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

Minimal IAM policy for read-write operations

To use your customer managed key for read-write Deadline Cloud operations, such as creating and updating farms, queues, and fleets. The following AWS KMS API operations must be permitted:

- `kms:Decrypt` – Allows Deadline Cloud to decrypt data in the farm.
- `kms:DescribeKey` – Provides the customer managed key details to allow Deadline Cloud to validate the key.
- `kms:GenerateDataKey` – Allows Deadline Cloud to encrypt data using a unique data key.

The following policy statement grants the necessary permissions for the `CreateFarm` operation.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "DeadlineReadWrite",
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "deadline.us-west-2.amazonaws.com"
                }
            }
        }
    ]
}
```

Monitoring your encryption keys

When you use an AWS KMS customer managed key with your Deadline Cloud farms, you can use AWS CloudTrail or Amazon CloudWatch Logs to track requests that Deadline Cloud sends to AWS KMS.

CloudTrail event for grants

The following example CloudTrail event occurs when grants are created, typically when you call the `CreateFarm`, `CreateMonitor`, or `CreateFleet` operation.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/Admin/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T02:05:26Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T02:05:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "operations": [
            "CreateGrant",
            "Decrypt",
            "DescribeKey",
            "Encrypt",
            "GenerateDataKey"
        ],
        "constraints": {
            "encryptionContextSubset": {
                "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
                "aws:deadline:accountId": "111122223333"
            }
        },
        "granteePrincipal": "deadline.amazonaws.com",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",
        "retiringPrincipal": "deadline.amazonaws.com"
    },
    "responseElements": {
        "grantId": "6bbe819394822a400fe5e3a75d0e9ef16c1733143fff0c1fc00dc7ac282a18a0",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "readOnly": false,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE44444"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail event for decryption

The following example CloudTrail event occurs when decrypting values using the customer managed KMS key.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:51:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY+p/5H+EuKd4Q=="
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "responseElements": null,
    "requestID": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeffffff",
    "eventID": "ffffffff-eeee-dddd-cccc-bbbbbbaaaaaa",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail event for encryption

The following example CloudTrail event occurs when encrypting values using the customer managed KMS key.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:SampleUser01",
        "arn": "arn:aws::sts::111122223333:assumed-role/SampleRole/SampleUser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE",
                "arn": "arn:aws::iam::111122223333:role/SampleRole",
                "accountId": "111122223333",
                "userName": "SampleRole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-04-23T18:46:51Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "deadline.amazonaws.com"
    },
    "eventTime": "2024-04-23T18:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "deadline.amazonaws.com",
    "userAgent": "deadline.amazonaws.com",
    "requestParameters": {
        "numberOfBytes": 32,
        "encryptionContext": {
            "aws:deadline:farmId": "farm-abcdef12345678900987654321fedcba",
            "aws:deadline:accountId": "111122223333",
            "aws-crypto-public-key": "AotL+SAMPLEVALUEiOMEXAMPLEaaqNOTREALaGTESTONLY
+p/5H+EuKd4Q=="
        },
        "keyId": "arn:aws::kms:us-
west-2:111122223333:key/abcdef12-3456-7890-0987-654321fedcba"
    },
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws::kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Deleting a customer managed KMS key

Deleting a customer managed KMS key in AWS Key Management Service (AWS KMS) is destructive and potentially dangerous. It irreversibly deletes the key material and all metadata associated with the key. After a customer managed KMS key is deleted, you can no longer decrypt the data that was encrypted by that key. Deleting the key means that the data becomes unrecoverable.

This is why AWS KMS gives customers a waiting period of up to 30 days before deleting the KMS key. The default waiting period is 30 days.

About the waiting period

Because it's destructive and potentially dangerous to delete a customer managed KMS key, we require that you set a waiting period of 7–30 days. The default waiting period is 30 days.

However, the actual waiting period might be up to 24 hours longer than the period you scheduled. To get the actual date and time when the key will be deleted, use the DescribeKey operation. You can also see the scheduled deletion date of a key in the AWS KMS console on the key’s detail page, in the **General configuration** section. Notice the time zone.

During the waiting period, the customer managed key’s status and key state is **Pending deletion**.

- A customer managed KMS key that is pending deletion can’t be used in any cryptographic operations.
- AWS KMS doesn’t rotate the backing keys of customer managed KMS keys that are pending deletion.

For more information about deleting a customer managed KMS key, see Deleting customer master keys in the *AWS Key Management Service Developer Guide*.

Inter-network traffic privacy

AWS Deadline Cloud supports Amazon Virtual Private Cloud (Amazon VPC) to secure connections. Amazon VPC provides features that you can use to increase and monitor the security for your virtual private cloud (VPC).

You can set up a customer-managed fleet (CMF) with Amazon Elastic Compute Cloud (Amazon EC2) instances that run inside a VPC. By deploying Amazon VPC endpoints to use AWS PrivateLink, traffic between workers in your CMF and the Deadline Cloud endpoint stays within your VPC. Furthermore, you can configure your VPC to restrict internet access to your instances.

In service-managed fleets, workers aren't reachable from the internet, but they do have internet access and connect to the Deadline Cloud service over the internet. Each service-managed fleet runs in its own isolated network, and worker instances remain dedicated to individual customers.

Opt out

AWS Deadline Cloud collects certain operational information to help us develop and improve Deadline Cloud. The collected data includes things such as your AWS account ID and user ID, so that we can correctly identify you if you have an issue with the Deadline Cloud. We also collect Deadline Cloud specific information, such as Resource IDs (a FarmID or QueueID when applicable), the product name (for example, JobAttachments, WorkerAgent, and more) and the product version.

You can choose to opt out from this data collection using application configuration. Each computer interacting with Deadline Cloud, both client workstations and fleet workers, needs to opt out separately.

Deadline Cloud monitor - desktop

Deadline Cloud monitor - desktop collects operational information, such as when crashes occur and when the application is opened, to help us know when you are having problems with the application. To opt out from the collection of this operational information, go to the settings page and clear **Turn on data collection to measure Deadline Cloud Monitor's performance**.

After you opt out, the desktop monitor no longer sends the operational data. Any previously collected data is retained and may still be used to improve the service. For more information, see Data Privacy FAQ.

AWS Deadline Cloud CLI and Tools

The AWS Deadline Cloud CLI, submitters, and worker agent all collect operational information such as when crashes occur and when jobs are submitted to help us know when you are having problems with these applications. To opt out from the collection of this operational information, use any of the following methods:

- In the terminal, enter `deadline config set telemetry.opt_out true`.

  This will opt out the CLI, submitters, and worker agent when running as the current user.
- When installing the Deadline Cloud worker agent, add the `--telemetry-opt-out` command line argument. For example, `./install.sh --farm-id $FARM_ID --fleet-id $FLEET_ID --telemetry-opt-out`.
- Before running the worker agent, CLI, or submitter, set an environment variable: `DEADLINE_CLOUD_TELEMETRY_OPT_OUT=true`

After you opt out, the Deadline Cloud tools no longer send the operational data. Any previously collected data is retained and may still be used to improve the service. For more information, see Data Privacy FAQ.

Identity and Access Management in Deadline Cloud

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Deadline Cloud resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user -** request permissions from your administrator if you cannot access features (see Troubleshooting AWS Deadline Cloud identity and access)
- **Service administrator -** determine user access and submit permission requests (see How Deadline Cloud works with IAM)
- **IAM administrator -** write policies to manage access (see Identity-based policy examples for Deadline Cloud)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Deadline Cloud works with IAM

Before you use IAM to manage access to Deadline Cloud, learn what IAM features are available to use with Deadline Cloud.

IAM features you can use with AWS Deadline Cloud

IAM feature	Deadline Cloud support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward access sessions (FAS)	Yes
Service roles	Yes
Service-linked roles	No

To get a high-level view of how Deadline Cloud and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Deadline Cloud

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Deadline Cloud

To view examples of Deadline Cloud identity-based policies, see Identity-based policy examples for Deadline Cloud.

Resource-based policies within Deadline Cloud

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Deadline Cloud

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Deadline Cloud actions, see Actions defined by AWS Deadline Cloud in the *Service Authorization Reference*.

Policy actions in Deadline Cloud use the following prefix before the action:

```
deadline
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "deadline:action1",
      "deadline:action2"
         ]
```

To view examples of Deadline Cloud identity-based policies, see Identity-based policy examples for Deadline Cloud.

Policy resources for Deadline Cloud

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Deadline Cloud resource types and their ARNs, see Resources defined by AWS Deadline Cloud in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Deadline Cloud.

To view examples of Deadline Cloud identity-based policies, see Identity-based policy examples for Deadline Cloud.

Policy condition keys for Deadline Cloud

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Deadline Cloud condition keys, see Condition keys for AWS Deadline Cloud in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Deadline Cloud.

To view examples of Deadline Cloud identity-based policies, see Identity-based policy examples for Deadline Cloud.

ACLs in Deadline Cloud

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Deadline Cloud

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Deadline Cloud

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for Deadline Cloud

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for Deadline Cloud

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> ⚠ **Warning**
>
> Changing the permissions for a service role might break Deadline Cloud functionality. Edit service roles only when Deadline Cloud provides guidance to do so.

Service-linked roles for Deadline Cloud

Supports service-linked roles: No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for Deadline Cloud

By default, users and roles don't have permission to create or modify Deadline Cloud resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Deadline Cloud, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Deadline Cloud in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Deadline Cloud resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more

information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Deadline Cloud console

To access the AWS Deadline Cloud console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Deadline Cloud resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Deadline Cloud console, also attach the Deadline Cloud `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Policy to access the console

To grant access to all functionality in the Deadline Cloud console, attach this identity policy to a user or role you want to have full access.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "EC2InstanceTypeSelection",
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeInstanceTypeOfferings",
            "ec2:DescribeInstanceTypes",
            "ec2:GetInstanceTypesFromInstanceRequirements",
            "pricing:GetProducts"
        ],
        "Resource": ["*"]
    },
    {
        "Sid": "VPCResourceSelection",
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
        ],
        "Resource": ["*"]
    },
    {
        "Sid": "ViewVpcLatticeResources",
        "Effect": "Allow",
        "Action": [
            "vpc-lattice:ListResourceConfigurations",
            "vpc-lattice:GetResourceConfiguration",
            "vpc-lattice:GetResourceGateway"
        ],
        "Resource": ["*"]
    },
    {
        "Sid": "ManageVpcEndpointsViaDeadline",
        "Effect": "Allow",
        "Action": [
            "ec2:CreateVpcEndpoint",
            "ec2:DescribeVpcEndpoints",
            "ec2:DeleteVpcEndpoints",
        "ec2:CreateTags"
        ],
        "Resource": ["*"],
        "Condition": {
        "StringEquals": { "aws:CalledViaFirst": "deadline.amazonaws.com" }
        }
    },
    {
        "Sid": "ChooseJobAttachmentsBucket",
        "Effect": "Allow",
        "Action": ["s3:GetBucketLocation", "s3:ListAllMyBuckets"],
        "Resource": "*"
    },
    {
        "Sid": "CreateDeadlineCloudLogGroups",
        "Effect": "Allow",
        "Action": ["logs:CreateLogGroup"],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/deadline/*",
        "Condition": {
        "StringLike": { "aws:CalledViaFirst": "deadline.amazonaws.com" }
        }
    },
    {
        "Sid": "ValidateDependencies",
        "Effect": "Allow",
        "Action": ["s3:ListBucket"],
        "Resource": "*",
        "Condition": {
        "StringLike": { "aws:CalledViaFirst": "deadline.amazonaws.com" }
        }
    },
    {
        "Sid": "RoleSelection",
        "Effect": "Allow",
        "Action": ["iam:GetRole", "iam:ListRoles"],
        "Resource": "*"
    },
    {
        "Sid": "PassRoleToDeadlineCloud",
        "Effect": "Allow",
        "Action": ["iam:PassRole"],
        "Condition": {
        "StringLike": { "iam:PassedToService": "deadline.amazonaws.com" }
    },
        "Resource": "*"
    },
    {
        "Sid": "KMSKeySelection",
        "Effect": "Allow",
        "Action": ["kms:ListKeys", "kms:ListAliases"],
        "Resource": "*"
    },
    {
        "Sid": "IdentityStoreReadOnly",
        "Effect": "Allow",
        "Action": [
            "identitystore:DescribeUser",
            "identitystore:DescribeGroup",
            "identitystore:ListGroups",
            "identitystore:ListUsers",
            "identitystore:IsMemberInGroups",
            "identitystore:ListGroupMemberships",
            "identitystore:ListGroupMembershipsForMember",
            "identitystore:GetGroupMembershipId"
    ],
        "Resource": "*"
    },
    {
        "Sid": "OrganizationAndIdentityCenterIdentification",
        "Effect": "Allow",
        "Action": [
            "sso:ListDirectoryAssociations",
            "organizations:DescribeAccount",
            "organizations:DescribeOrganization",
            "sso:DescribeRegisteredRegions",
            "sso:GetManagedApplicationInstance",
            "sso:GetSharedSsoConfiguration",
            "sso:ListInstances",
            "sso:GetApplicationAssignmentConfiguration",
            "sso:GetSSOStatus",
            "sso:ListRegions",
            "sso:DescribeRegion"
    ],
        "Resource": "*"
    },
    {
        "Sid": "ManagedDeadlineCloudIDCApplication",
        "Effect": "Allow",
        "Action": [
            "sso:CreateApplication",
            "sso:PutApplicationAssignmentConfiguration",
            "sso:PutApplicationAuthenticationMethod",
            "sso:PutApplicationGrant",
            "sso:DeleteApplication",
            "sso:UpdateApplication"
    ],
        "Resource": "*",
        "Condition": {
        "StringLike": { "aws:CalledViaFirst": "deadline.amazonaws.com" }
        }
    },
    {
        "Sid": "ChooseSecret",
        "Effect": "Allow",
        "Action": ["secretsmanager:ListSecrets"],
        "Resource": "*"
    },
    {
        "Sid": "DeadlineMembershipActions",
        "Effect": "Allow",
        "Action": [
            "deadline:AssociateMemberToFarm",
            "deadline:AssociateMemberToFleet",
            "deadline:AssociateMemberToQueue",
            "deadline:AssociateMemberToJob",
            "deadline:DisassociateMemberFromFarm",
            "deadline:DisassociateMemberFromFleet",
            "deadline:DisassociateMemberFromQueue",
            "deadline:DisassociateMemberFromJob",
            "deadline:ListFarmMembers",
            "deadline:ListFleetMembers",
            "deadline:ListQueueMembers",
            "deadline:ListJobMembers"
    ],
        "Resource": ["*"]
    },
    {
        "Sid": "DeadlineControlPlaneActions",
        "Effect": "Allow",
        "Action": [
            "deadline:CreateMonitor",
            "deadline:GetMonitor",
            "deadline:UpdateMonitor",
            "deadline:DeleteMonitor",
            "deadline:ListMonitors",
            "deadline:CreateFarm",
            "deadline:GetFarm",
            "deadline:UpdateFarm",
            "deadline:DeleteFarm",
            "deadline:ListFarms",
            "deadline:CreateQueue",
            "deadline:GetQueue",
            "deadline:UpdateQueue",
            "deadline:DeleteQueue",
            "deadline:ListQueues",
            "deadline:CreateFleet",
            "deadline:GetFleet",
            "deadline:UpdateFleet",
            "deadline:DeleteFleet",
            "deadline:ListFleets",
            "deadline:ListWorkers",
            "deadline:CreateQueueFleetAssociation",
            "deadline:GetQueueFleetAssociation",
            "deadline:UpdateQueueFleetAssociation",
            "deadline:DeleteQueueFleetAssociation",
            "deadline:ListQueueFleetAssociations",
            "deadline:CreateQueueEnvironment",
            "deadline:GetQueueEnvironment",
            "deadline:UpdateQueueEnvironment",
            "deadline:DeleteQueueEnvironment",
            "deadline:ListQueueEnvironments",
            "deadline:CreateLimit",
            "deadline:GetLimit",
            "deadline:UpdateLimit",
            "deadline:DeleteLimit",
            "deadline:ListLimits",
            "deadline:CreateQueueLimitAssociation",
            "deadline:GetQueueLimitAssociation",
            "deadline:DeleteQueueLimitAssociation",
            "deadline:UpdateQueueLimitAssociation",
            "deadline:ListQueueLimitAssociations",
            "deadline:CreateStorageProfile",
            "deadline:GetStorageProfile",
            "deadline:UpdateStorageProfile",
            "deadline:DeleteStorageProfile",
            "deadline:ListStorageProfiles",
            "deadline:ListStorageProfilesForQueue",
            "deadline:ListBudgets",
            "deadline:TagResource",
            "deadline:UntagResource",
            "deadline:ListTagsForResource",
            "deadline:CreateLicenseEndpoint",
            "deadline:GetLicenseEndpoint",
            "deadline:DeleteLicenseEndpoint",
            "deadline:ListLicenseEndpoints",
            "deadline:ListAvailableMeteredProducts",
            "deadline:ListMeteredProducts",
            "deadline:PutMeteredProduct",
            "deadline:DeleteMeteredProduct"
        ],
        "Resource": ["*"]
      }]
}
```

Policy to submit jobs to a queue

In this example, you create a scoped-down policy that grants permission to submit jobs to a specific queue in a specific farm.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SubmitJobsFarmAndQueue",
            "Effect": "Allow",
            "Action": "deadline:CreateJob",
            "Resource": "arn:aws:deadline:us-east-1:111122223333:farm/FARM_A/
queue/QUEUE_B/job/*"
        }
    ]
}
```

Policy to allow creating a license endpoint

In this example, you create a scoped-down policy that grants the required permissions to create and manage license endpoints. Use this policy to create the license endpoint for the VPC associated with your farm.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "CreateLicenseEndpoint",
        "Effect": "Allow",
        "Action": [
            "deadline:CreateLicenseEndpoint",
            "deadline:DeleteLicenseEndpoint",
            "deadline:GetLicenseEndpoint",
            "deadline:ListLicenseEndpoints",
            "deadline:PutMeteredProduct",
            "deadline:DeleteMeteredProduct",
            "deadline:ListMeteredProducts",
            "deadline:ListAvailableMeteredProducts",
            "ec2:CreateVpcEndpoint",
            "ec2:DescribeVpcEndpoints",
            "ec2:DeleteVpcEndpoints"
        ],
        "Resource": [
            "arn:aws:deadline:*:111122223333:*",
            "arn:aws:ec2:*:111122223333:vpc-endpoint/*"
        ]
    }]
}
```

Policy to allow monitoring a specific farm queue

In this example, you create a scoped-down policy that grants permission to monitor jobs in a specific queue for a specific farm.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "MonitorJobsFarmAndQueue",
        "Effect": "Allow",
        "Action": [
            "deadline:SearchJobs",
            "deadline:ListJobs",
            "deadline:GetJob",
            "deadline:SearchSteps",
            "deadline:ListSteps",
            "deadline:ListStepConsumers",
            "deadline:ListStepDependencies",
            "deadline:GetStep",
            "deadline:SearchTasks",
            "deadline:ListTasks",
            "deadline:GetTask",
            "deadline:ListSessions",
            "deadline:GetSession",
            "deadline:ListSessionActions",
            "deadline:GetSessionAction"
        ],
        "Resource": [
            "arn:aws:deadline:us-east-1:123456789012:farm/FARM_A/queue/QUEUE_B",
            "arn:aws:deadline:us-east-1:123456789012:farm/FARM_A/queue/QUEUE_B/*"
        ]
    }]
}
```

AWS managed policies for Deadline Cloud

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSDeadlineCloud-FleetWorker

You can attach the `AWSDeadlineCloud-FleetWorker` policy to your AWS Identity and Access Management (IAM) identities.

This policy grants workers in this fleet the permissions that are needed to connect to and receive tasks from the service.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows principals to manage workers in a fleet.

For a JSON listing of the policy details, see AWSDeadlineCloud-FleetWorker in the *AWS Managed Policy reference guide*.

AWS managed policy: AWSDeadlineCloud-WorkerHost

You can attach the `AWSDeadlineCloud-WorkerHost` policy to your IAM identities.

This policy grants the permissions that are needed to initially connect to the service. It can be used as an Amazon Elastic Compute Cloud (Amazon EC2) instance profile.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows the user to create workers, assume the fleet role for workers, and apply tags to workers

For a JSON listing of the policy details, see AWSDeadlineCloud-WorkerHost in the *AWS Managed Policy reference guide*.

AWS managed policy: AWSDeadlineCloud-UserAccessFarms

You can attach the `AWSDeadlineCloud-UserAccessFarms` policy to your IAM identities.

This policy allows users to access farm data based on the farms that they are members of and their membership level.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows the user to access farm data.
- `ec2` – Allows users to see details about Amazon EC2 instance types.
- `identitystore` – Allows users to see user and group names.
- `kms` – Allows users to configure AWS Key Management Service (AWS KMS) customer-managed keys for their AWS IAM Identity Center (IAM Identity Center) instance.

For a JSON listing of the policy details, see AWSDeadlineCloud-UserAccessFarms in the *AWS Managed Policy reference guide*.

AWS managed policy: AWSDeadlineCloud-UserAccessFleets

You can attach the `AWSDeadlineCloud-UserAccessFleets` policy to your IAM identities.

This policy allows users to access fleet data based on the farms that they are members of and their membership level.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows the user to access farm data.
- `ec2` – Allows users to see details about Amazon EC2 instance types.
- `identitystore` – Allows users to see user and group names.

For a JSON listing of the policy details, see AWSDeadlineCloud-UserAccessFleets in the *AWS Managed Policy reference guide*.

AWS managed policy: AWSDeadlineCloud-UserAccessJobs

You can attach the `AWSDeadlineCloud-UserAccessJobs` policy to your IAM identities.

This policy allows users to access job data based on the farms that they are members of and their membership level.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows the user to access farm data.
- `ec2` – Allows users to see details about Amazon EC2 instance types.
- `identitystore` – Allows users to see user and group names.

For a JSON listing of the policy details, see AWSDeadlineCloud-UserAccessJobs in the *AWS Managed Policy reference guide*.

AWS managed policy: AWSDeadlineCloud-UserAccessQueues

You can attach the `AWSDeadlineCloud-UserAccessQueues` policy to your IAM identities.

This policy allows users to access queue data based on the farms that they are members of and their membership level.

Permissions details

This policy includes the following permissions:

- `deadline` – Allows the user to access farm data.
- `ec2` – Allows users to see details about Amazon EC2 instance types.
- `identitystore` – Allows users to see user and group names.

For a JSON listing of the policy details, see AWSDeadlineCloud-UserAccessQueues in the *AWS Managed Policy reference guide*.

Deadline Cloud updates to AWS managed policies

View details about updates to AWS managed policies for Deadline Cloud since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Deadline Cloud Document history page.

Change	Description	Date
AWSDeadlineCloud-UserAccessFarms – Change	Deadline Cloud added new action `kms:Decrypt` so you can use an AWS KMS customer-managed key with your IAM Identity Center instance.	December 22, 2025
AWSDeadlineCloud-WorkerHost – Change	Deadline Cloud added new actions `deadline:TagResource` and `deadline:ListTagsForResource` to allow you to add and view tags associated with workers in your fleet.	May 30, 2025
AWSDeadlineCloud-UserAccessFarms – Change AWSDeadlineCloud-UserAccessJobs – Change AWSDeadlineCloud-UserAccessQueues – Change	Deadline Cloud added new actions `deadline:GetJobTemplate` and `deadline:ListJobParameterDefinitions` to allow you to resubmit jobs.	October 7, 2024
Deadline Cloud started tracking changes	Deadline Cloud started tracking changes to its AWS managed policies.	April 2, 2024

Service roles

How Deadline Cloud uses IAM service roles

Deadline Cloud automatically assumes IAM roles and provides temporary credentials to workers, jobs, and the Deadline Cloud monitor. This approach eliminates manual credential management while maintaining security through role-based access control.

When you create monitors, fleets, and queues, you specify IAM roles that Deadline Cloud assumes on your behalf. Workers and the Deadline Cloud monitor then receive temporary credentials from these roles to access AWS services.

Fleet role

Configure a fleet role to give Deadline Cloud workers the permissions they need to receive work and report progress on that work.

You usually do not have to configure this role yourself. This role can be created for you in the Deadline Cloud console to include the necessary permissions. Use the following guide to understand the specifics of this role for troubleshooting.

When creating or updating fleets programmatically, specify the fleet role ARN using the `CreateFleet` or `UpdateFleet` API operations.

What the fleet role does

The fleet role provides workers with permissions to:

- Receive new work and report progress on ongoing work to the Deadline Cloud service
- Manage worker lifecycle and status
- Record log events to Amazon CloudWatch Logs for the worker logs

Set up the fleet role trust policy

Your fleet role must trust the Deadline Cloud service and be scoped to your specific farm.

As a best practice, the trust policy should include security conditions for Confused Deputy protection. To learn more about Confused Deputy protection, see Confused Deputy in the *Deadline Cloud User Guide*.

- `aws:SourceAccount` ensures only resources from the same AWS account can assume this role.
- `aws:SourceArn` restricts role assumption to a specific Deadline Cloud farm.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDeadlineCredentialsService",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "credentials.deadline.amazonaws.com"
      },
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "YOUR_ACCOUNT_ID"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:REGION:YOUR_ACCOUNT_ID:farm/YOUR_FARM_ID"
        }
      }
    }
  ]
}
```

Attach the Fleet role permissions

Attach the following AWS managed policy to your fleet role:

AWSDeadlineCloud-FleetWorker

This managed policy provides permissions for:

- `deadline:AssumeFleetRoleForWorker` - Allows workers to refresh their credentials.
- `deadline:UpdateWorker` - Allows workers to update their status (for example, to STOPPED when exiting).
- `deadline:UpdateWorkerSchedule` - For obtaining work and reporting progress.
- `deadline:BatchGetJobEntity` - For fetching job information.
- `deadline:AssumeQueueRoleForWorker` - For accessing queue role credentials during job execution.

Add KMS permissions for encrypted farms

If your farm was created using a KMS key, add these permissions to your fleet role to ensure the worker can access encrypted data in the farm.

The KMS permissions are only necessary if your farm has an associated KMS key. The `kms:ViaService` condition must use the format `deadline.{region}.amazonaws.com`.

When creating a fleet, a CloudWatch Logs log group is created for that fleet. The worker's permissions are used by the Deadline Cloud service to create a log stream specifically for that particular worker. After the worker is set up and running, the worker will use these permissions to send log events directly to CloudWatch Logs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateLogStream",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:REGION:YOUR_ACCOUNT_ID:log-group:/aws/
deadline/YOUR_FARM_ID/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "deadline.REGION.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "ManageLogEvents",
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Resource": "arn:aws:logs:REGION:YOUR_ACCOUNT_ID:log-group:/aws/
deadline/YOUR_FARM_ID/*"
    },
    {
      "Sid": "ManageKmsKey",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:GenerateDataKey"
      ],
      "Resource": "YOUR_FARM_KMS_KEY_ARN",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "deadline.REGION.amazonaws.com"
        }
      }
    }
  ]
}
```

Modifying the fleet role

Permissions for the fleet role are not customizable. The described permissions are always required and adding additional permissions has no effect.

Customer-managed fleet host role

Set up a WorkerHost role if you use customer-managed fleets on Amazon EC2 instances or on-premises hosts.

What the WorkerHost role does

The WorkerHost role bootstraps workers on customer-managed fleet hosts. It provides the minimal permissions needed for a host to:

- Create a worker in Deadline Cloud
- Assume the fleet role to fetch operational credentials
- Tag workers with fleet tags (if tag propagation is enabled)

Set up WorkerHost role permissions

Attach the following AWS managed policy to your WorkerHost role:

AWSDeadlineCloud-WorkerHost

This managed policy provides permissions for:

- `deadline:CreateWorker` - Allows the host to register a new worker.
- `deadline:AssumeFleetRoleForWorker` - Allows the host to assume the fleet role.
- `deadline:TagResource` - Allows tagging workers during creation (if enabled).
- `deadline:ListTagsForResource` - Allows reading fleet tags for propagation.

Understand the bootstrap process

The WorkerHost role is only used during initial worker startup:

1. The worker agent starts on the host using WorkerHost credentials.
2. It invokes `deadline:CreateWorker` to register with Deadline Cloud.
3. It then invokes `deadline:AssumeFleetRoleForWorker` to fetch fleet role credentials.
4. From this point forward, the worker uses only fleet role credentials for all operations.

The WorkerHost role is not used after the worker starts running. This policy is not required for Service-managed fleets. In Service-managed fleets, bootstrapping is performed automatically.

Queue role

The queue role is assumed by the worker when processing a task. This role provides the permissions needed to complete the task.

When creating or updating queues programmatically, specify the queue role ARN using the `CreateQueue` or `UpdateQueue` API operations.

Set up the queue role trust policy

Your queue role must trust the Deadline Cloud service.

As a best practice, the trust policy should include security conditions for Confused Deputy protection. To learn more about Confused Deputy protection, see Confused Deputy in the *Deadline Cloud User Guide*.

- `aws:SourceAccount` ensures only resources from the same AWS account can assume this role.
- `aws:SourceArn` restricts role assumption to a specific Deadline Cloud farm.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "credentials.deadline.amazonaws.com",
          "deadline.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "YOUR_ACCOUNT_ID"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:deadline:us-west-2:123456789012:farm/{farm-id}"
        }
      }
    }
  ]
}
```

Understand queue role permissions

The queue role doesn't use a single managed policy. Instead, when you configure your queue in the console, Deadline Cloud creates a custom policy for your queue based on your configuration.

This automatically created policy provides access to:

Job attachments

Read and write access to your specified Amazon S3 bucket for job input and output files:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject",
    "s3:PutObject",
    "s3:ListBucket",
    "s3:GetBucketLocation"
  ],
  "Resource": [
    "arn:aws:s3:::YOUR_JOB_ATTACHMENTS_BUCKET",
    "arn:aws:s3:::YOUR_JOB_ATTACHMENTS_BUCKET/YOUR_PREFIX/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceAccount": "YOUR_ACCOUNT_ID"
    }
  }
}
```

Job logs

Read access to CloudWatch Logs for jobs in this queue. Each queue has its own log group and each session has its own log stream:

```
{
  "Effect": "Allow",
  "Action": [
    "logs:GetLogEvents"
  ],
  "Resource": "arn:aws:logs:REGION:YOUR_ACCOUNT_ID:log-group:/aws/
deadline/YOUR_FARM_ID/*"
}
```

Third-party software

Access to download third-party software supported by Deadline Cloud (such as Maya, Blender, and others):

```
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "s3:GetObject"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "s3:DataAccessPointArn": "arn:aws:s3:*:*:accesspoint/deadline-software-*"
    },
    "StringEquals": {
      "s3:AccessPointNetworkOrigin": "VPC"
    }
  }
}
```

Add permissions for your jobs

Add permissions to your queue role for AWS services that your jobs need to access. When writing OpenJobDescription step scripts, the AWS CLI and SDK will automatically use credentials from your queue role. Use this to access additional services needed to complete your job.

Example use cases include:

- for fetching custom data
- SSM permissions to tunnel to a custom license server
- CloudWatch for emitting custom metrics
- Deadline Cloud permission to create new jobs for dynamic workflows

How queue role credentials are used

Deadline Cloud provides queue role credentials to:

- Workers during job execution
- Users via Deadline Cloud CLI and monitor when interacting with job attachments and logs

Deadline Cloud creates separate CloudWatch Logs log groups for each queue. Jobs use queue role credentials to write logs to their queue's log group. The Deadline Cloud CLI and monitor use the queue role (through `deadline:AssumeQueueRoleForRead`) to read job logs from the queue's log group. The Deadline Cloud CLI and monitor use the queue role (through `deadline:AssumeQueueRoleForUser`) to upload or download job attachments data.

Monitor role

Configure a monitor role to give the Deadline Cloud monitor web and desktop applications access to your Deadline Cloud resources.

When creating or updating monitors programmatically, specify the monitor role ARN using the `CreateMonitor` or `UpdateMonitor` API operations.

What the monitor role does

The monitor role enables Deadline Cloud monitor to provide end users with access to:

- Basic functionality required for the Deadline Cloud Integrated Submitters, CLI and monitor
- Custom functionality for end users

Set up the monitor role trust policy

Your monitor role must trust the Deadline Cloud service.

As a best practice, the trust policy should include security conditions for Confused Deputy protection. To learn more about Confused Deputy protection, see Confused Deputy in the *Deadline Cloud User Guide*.

`aws:SourceAccount` ensures only resources from the same AWS account can assume this role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "credentials.deadline.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "YOUR_ACCOUNT_ID"
        }
      }
    }
  ]
}
```

Attach monitor role permissions

Attach all of the following AWS managed policies to your monitor role for basic operation:

- AWSDeadlineCloud-UserAccessFarms
- AWSDeadlineCloud-UserAccessFleets

- AWSDeadlineCloud-UserAccessJobs
- AWSDeadlineCloud-UserAccessQueues

How the monitor role works

When using the Deadline Cloud monitor, a service user signs in using AWS IAM Identity Center (IAM Identity Center), and the monitor role is assumed. The assumed role credentials are used by the monitor application to display the monitor UI, including the list of farms, fleets, queues, and other information.

When using the Deadline Cloud monitor desktop application, these credentials are additionally made available on the workstation using a named AWS credential profile corresponding to the profile name provided by the end user. Learn more about named profiles in the AWS SDK and Tools reference guide.

This named profile is how the Deadline CLI and submitters access Deadline Cloud resources.

Customizing the monitor role for advanced use cases

You can customize the monitor role to modify what users can do at each access level (Viewer, Contributor, Manager, Owner) or to add permissions for advanced workflows.

Customizing access level permissions

The four AWS managed policies attached to the monitor role control what each access level can do. You can add custom policies to the monitor role to grant or restrict permissions for specific access levels using the `deadline:MembershipLevel` condition key.

For example, to allow Contributors to update and cancel jobs (which is normally restricted to Managers and Owners), add a policy like the following:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "deadline:UpdateJob",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "deadline:MembershipLevel": "CONTRIBUTOR"
          }
        }
      }
    ]
  }
```

With this policy, Contributors can update and cancel jobs in addition to submitting them.

Adding permissions for advanced workflows

You can add custom IAM policies to the monitor role to grant additional permissions to all monitor users. This is useful for advanced scripting workflows where users need access to AWS services beyond the standard Deadline Cloud functionality.

Follow these guidelines when modifying your monitor role:

- Don't remove any of the managed policies. Removing these policies breaks monitor functionality.

How Deadline Cloud monitor uses monitor role credentials

Deadline Cloud monitor automatically obtains monitor role credentials when you authenticate. This capability enables the desktop application to provide enhanced monitoring capabilities beyond what's available in a standard web browser.

When you log in with Deadline Cloud monitor, it automatically creates a profile that you can use with the AWS CLI or any other AWS tool. This profile uses the monitor role credentials, giving you programmatic access to AWS services based on the permissions in your monitor role.

Deadline Cloud submitters work the same way - they use the profile created by Deadline Cloud monitor to access AWS services with the appropriate role permissions.

Advanced customization of Deadline Cloud roles

You can extend Deadline Cloud roles with additional permissions to enable advanced use cases beyond basic rendering workflows. This approach leverages Deadline Cloud's access management system to control access to additional AWS services based on queue membership.

Team collaboration with AWS CodeCommit

Add AWS CodeCommit permissions to your Queue role to enable team collaboration on project repositories. This approach uses Deadline Cloud's access management system for additional use cases - only users with access to the specific queue will receive these AWS CodeCommit permissions, allowing you to manage per-project repository access through Deadline Cloud queue membership.

This is useful for scenarios where artists need to access project-specific assets, scripts, or configuration files stored in AWS CodeCommit repositories as part of their rendering workflow.

Add AWS CodeCommit permissions to queue role

Add the following permissions to your queue role to enable AWS CodeCommit access:

```
{
  "Effect": "Allow",
  "Action": [
    "codecommit:GitPull",
    "codecommit:GitPush",
    "codecommit:GetRepository",
    "codecommit:ListRepositories"
  ],
  "Resource": "arn:aws:codecommit:REGION:YOUR_ACCOUNT_ID:PROJECT_REPOSITORY"
}
```

Set up credential provider on artist workstations

Configure each artist workstation to use Deadline Cloud queue credentials for AWS CodeCommit access. This setup is done once per workstation.

To configure the credential provider

1. Add a credential provider profile to your AWS config file (`~/.aws/config`):

```
[profile queue-codecommit]
credential_process = deadline queue export-credentials --farm-id farm-
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX --queue-id queue-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

2. Configure Git to use this profile for AWS CodeCommit repositories:

```
git config --global credential.https://git-codecommit.REGION.amazonaws.com.helper
 '!aws codecommit credential-helper --profile queue-codecommit $@'
git config --global credential.https://git-
codecommit.REGION.amazonaws.com.UseHttpPath true
```

Replace `farm-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX` and `queue-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX` with your actual farm and queue IDs. Replace `REGION` with your AWS region (for example, `us-west-2`).

Using AWS CodeCommit with queue credentials

Once configured, Git operations will automatically use the queue role credentials when accessing AWS CodeCommit repositories. The `deadline queue export-credentials` command returns temporary credentials that look like this:

```
{
  "Version": 1,
  "AccessKeyId": "ASIA...",
  "SecretAccessKey": "...",
  "SessionToken": "...",
  "Expiration": "2025-11-10T23:02:23+00:00"
}
```

These credentials are automatically refreshed as needed, and Git operations will work seamlessly:

```
git clone https://git-codecommit.REGION.amazonaws.com/v1/repos/PROJECT_REPOSITORY
git pull
git push
```

Artists can now access project repositories using their queue permissions without needing separate AWS CodeCommit credentials. Only users with access to the specific queue will be able to access the associated repository, enabling fine-grained access control through Deadline Cloud's queue membership system.

Troubleshooting AWS Deadline Cloud identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Deadline Cloud and IAM.

Topics

I am not authorized to perform an action in Deadline Cloud

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `deadline:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 deadline:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `deadline:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Deadline Cloud.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Deadline Cloud. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Deadline Cloud resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Deadline Cloud supports these features, see How Deadline Cloud works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for Deadline Cloud

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in Deadline Cloud

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

AWS Deadline Cloud does not back up data stored in your job attachments S3 bucket. You can enable backups of your job attachments data using any standard Amazon S3 backup mechanism, such as S3 Versioning or AWS Backup.

Infrastructure security in Deadline Cloud

As a managed service, AWS Deadline Cloud is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access Deadline Cloud through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Deadline Cloud doesn't support using AWS PrivateLink virtual private cloud (VPC) endpoint policies. It uses the AWS PrivateLink default policy, which grants full access to the endpoint. For more information, see Default endpoint policy in the *AWS PrivateLink user guide*.

Configuration and vulnerability analysis in Deadline Cloud

AWS handles basic security tasks like guest operating system (OS) and database patching, firewall configuration, and disaster recovery. These procedures have been reviewed and certified by the appropriate third parties. For more details, see the following resources:

- Shared Responsibility Model
- Amazon Web Services: Overview of Security Processes (whitepaper)

AWS Deadline Cloud manages tasks on service-managed or customer-managed fleets:

- For service-managed fleets, Deadline Cloud manages the guest operating system.
- For customer-managed fleets, you are responsible for managing the operating system.

For additional information about configuration and vulnerability analysis for AWS Deadline Cloud, see

- Security best practices for Deadline Cloud

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it should not otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the aws:SourceArn and aws:SourceAccount global condition context keys in resource policies to limit the permissions that AWS Deadline Cloud gives another service to the resource. Use `aws:SourceArn` if you want only one resource to be associated with the cross-service access. Use `aws:SourceAccount` if you want to allow any resource in that account to be associated with the cross-service use.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full Amazon Resource Name (ARN) of the resource. If you don't know the full ARN of the resource or if you are specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcard characters (*) for the unknown portions of the ARN. For example, `arn:aws:deadline:*:123456789012:*`.

If the `aws:SourceArn` value does not contain the account ID, such as an Amazon S3 bucket ARN, you must use both global condition context keys to limit permissions.

The following example shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Deadline Cloud to prevent the confused deputy problem.

JSON

```json
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "deadline.amazonaws.com"
    },
    "Action": "deadline:CreateFarm",
    "Resource": [
      "*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:deadline:*:111122223333:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
}
```

Access AWS Deadline Cloud using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and AWS Deadline Cloud. You can access Deadline Cloud as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or Direct Connect connection. Instances in your VPC don't need public IP addresses to access Deadline Cloud.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for Deadline Cloud.

Deadline Cloud also has dual-stack endpoints available. Dual-stack endpoints support requests over IPv6 and IPv4.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Considerations for Deadline Cloud

Before you set up an interface endpoint for Deadline Cloud, see Access an AWS service using an interface VPC endpoint in the *AWS PrivateLink Guide*.

Deadline Cloud supports making calls to all of its API actions through the interface endpoint.

By default, full access to Deadline Cloud is allowed through the interface endpoint. Alternatively, you can associate a security group with the endpoint network interfaces to control traffic to Deadline Cloud through the interface endpoint.

Deadline Cloud also supports VPC endpoint policies. For more information, see Control access to VPC endpoints using endpoint policies in the *AWS PrivateLink Guide*.

Deadline Cloud endpoints

Deadline Cloud uses four endpoints for access to the service using AWS PrivateLink - two for IPv4 and two for IPv6.

Workers use the `scheduling.deadline.region.amazonaws.com` endpoint to get tasks from the queue, report progress to Deadline Cloud, and to send task output back. If you are using a customer-managed fleet, the scheduling endpoint is the only endpoint that you need to create unless you are using management operations. For example, if a job creates more jobs, you need to enable the management endpoint to call the `CreateJob` operation.

The Deadline Cloud monitor uses the `management.deadline.`*`region`*`.amazonaws.com` to manage the resources in your farm, such as creating and modifying queues and fleets or getting lists of jobs, steps, and tasks.

The AWS SDKs and CLI automatically add the `management` and `scheduling` prefixes to the endpoint. If you want to disable this behavior, see the host prefix injection section in the *AWS SDKs and Tools Reference Guide*.

Deadline Cloud also requires endpoints for the following AWS service endpoints:

- Deadline Cloud uses AWS STS to authenticate workers so that they can access job assets. For more information about AWS STS, see Temporary security credentials in IAM in the *AWS Identity and Access Management User Guide*.
- If you set up your customer-managed fleet in a subnet with no internet connection you must create a VPC endpoint for Amazon CloudWatch Logs so that workers can write logs. For more information, see Monitoring with CloudWatch.
- If you use job attachments, you must create a VPC endpoint for Amazon Simple Storage Service (Amazon S3) so that workers can access the attachments. For more information, see Job attachments in Deadline Cloud.

Create endpoints for Deadline Cloud

You can create interface endpoints for Deadline Cloud using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create management and scheduling endpoints for Deadline Cloud using the following service names. Replace *`region`* with the AWS Region where you've deployed Deadline Cloud.

```
com.amazonaws.region.deadline.management
```

```
com.amazonaws.region.deadline.scheduling
```

Deadline Cloud supports dual-stack endpoints.

If you enable private DNS for the interface endpoints, you can make API requests to Deadline Cloud using its default Regional DNS name. For example, `scheduling.deadline.us-east-1.amazonaws.com` for worker operations, or `management.deadline.us-east-1.amazonaws.com` for all other operations.

You must also create an endpoint for AWS STS using the following service name:

```
com.amazonaws.region.sts
```

If your customer-managed fleet is on a subnet without an internet connection, you must create a CloudWatch Logs endpoint using the following service name:

```
com.amazonaws.region.logs
```

If you use job attachments to transfer files, you must create an Amazon S3 endpoint using the following service name:

```
com.amazonaws.region.s3
```

Restricted network environments

Deadline Cloud provides tools that are used by artists or other users on their local workstations. These tools require access to AWS API and web endpoints to perform their function. If you filter access to specific AWS domains or URL endpoints by using a web content filtering solution such as next-generation firewalls (NGFW) or Secure Web Gateways (SWG), you must add the following domains or URL endpoints to your web-content filtering solution allowlists.

AWS API endpoints to allowlist

Deadline Cloud client tools, such as the AWS Management Console, monitor, CLI, and integrated submitters, require access to AWS APIs in addition to Deadline Cloud. These endpoints only support IPv4.

- `scheduling.deadline.[Region].amazonaws.com`
- `management.deadline.[Region].amazonaws.com`
- `logs.[Region].amazonaws.com`

- `ec2.`*`[Region]`*`.amazonaws.com`
- `s3.`*`[Region]`*`.amazonaws.com`
- `sts.`*`[Region]`*`.amazonaws.com`
- `identitystore.`*`[Region]`*`.amazonaws.com`

Web domains to allowlist

The Deadline Cloud monitor requires access to the following domains to operate.

For additional information about allowlisting domains for AWS Sign-In, see Domains to add to your allow list in the *AWS Sign-In User Guide*.

- `downloads.deadlinecloud.amazonaws.com`
- `d2ev1rdnjzhmnr.cloudfront.net`
- `prod.log.shortbread.aws.dev`
- `prod.tools.shortbread.aws.dev`
- `prod.log.shortbread.analytics.console.aws.a2z.com`
- `prod.tools.shortbread.analytics.console.aws.a2z.com`
- `global.help-panel.docs.aws.a2z.com`
- *`[Region]`*`.signin.aws`
- *`[Region]`*`.signin.aws.amazon.com`
- `sso.`*`[Region]`*`.amazonaws.com`
- `portal.sso.`*`[Region]`*`.amazonaws.com`
- `oidc.`*`[Region]`*`.amazonaws.com`
- `assets.sso-portal.`*`[Region]`*`.amazonaws.com`

The Deadline Cloud submitter requires access to the following domains to download GUI dependencies.

- `pypi.python.org`
- `pypi.org`
- `pythonhosted.org`

- `files.pythonhosted.org`

Environment-specific endpoints to allowlist

These domains vary depending on the specific configuration of Deadline Cloud. If additional Deadline Cloud monitors or queues are created, additional domains will need to be allowlisted.

- *`[Directory ID or alias]`*`.awsapps.com`

  This domain is tied to the IAM Identity Center setup and should be the same for all setups in this using the same IAM Identity Center instance. The exact value can be found by the enterprise admin in the IAM Identity Center console under *Settings* → *AWS access portal URL*.
- *`[Monitor alias]`*`.`*`[Region]`*`.deadlinecloud.amazonaws.com`

  This domain is for the Monitor setup in Deadline Cloud. Artists enter this link into their browser or Deadline Cloud monitor application. If Deadline Cloud is set up in additional accounts or regions in the future, this domain will change. You can find this value in the Deadline Cloud console in the *Dashboard* → *Monitor overview* → *Monitor details* → *URL*.
- *`[Bucket name]`*`.`*`[Region]`*`.s3.amazonaws.com`

  This is the domain for the job attachments bucket used by Deadline Cloud queues. Each queue can have its own job attachments bucket configured. The exact bucket name can be found in the Deadline Cloud console under *Queues* → *Queue details* → *Job attachments*. For more information about job attachments, see the queues documentation.

Security best practices for Deadline Cloud

AWS Deadline Cloud (Deadline Cloud) provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Note

For more information about the importance of many security topics, see the Shared Responsibility Model.

Data protection

For data protection purposes, we recommend that you protect AWS account credentials and set up individual accounts with AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon Simple Storage Service (Amazon S3).
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This recommendation includes when you work with AWS Deadline Cloud or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Deadline Cloud or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don’t include credentials information in the URL to validate your request to that server.

AWS Identity and Access Management permissions

Manage access to AWS resources using users, AWS Identity and Access Management (IAM) roles, and by granting the least privilege to users. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see IAM Best Practices in the *IAM User Guide*.

Run jobs as users and groups

When using queue functionality in Deadline Cloud, it’s a best practice to specify an operating system (OS) user and its primary group so that the OS user has least-privilege permissions for the queue’s jobs.

When you specify a “Run as user” (and group), any processes for jobs submitted to the queue will be run using that OS user and will inherit that user’s associated OS permissions.

The fleet and queue configurations combine to establish a security posture. On the queue side, the “Job run as user” and IAM role can be specified to use the OS and AWS permissions for the queue’s jobs. The fleet defines the infrastructure (worker hosts, networks, mounted shared storage) that, when associated to a particular queue, run jobs within the queue. The data available on the worker hosts needs to be accessed by jobs from one or more associated queues. Specifying a user or group helps protect the data in jobs from other queues, other installed software, or other users with access to the worker hosts. When a queue is without a user, it runs as the agent user which can impersonate (`sudo`) any queue user. In this way, a queue without a user can escalate privileges to another queue.

Networking

To prevent traffic from being intercepted or redirected, it's essential to secure how and where your network traffic is routed.

We recommend that you secure your networking environment in the following ways:

- Secure Amazon Virtual Private Cloud (Amazon VPC) subnet route tables to control how IP layer traffic is routed.
- If you are using Amazon Route 53 (Route 53) as a DNS provider in your farm or workstation setup, secure access to the Route 53 API.
- If you connect to Deadline Cloud outside of AWS such as by using on-premises workstations or other data centers, secure any on-premises networking infrastructure. This includes DNS servers and route tables on routers, switches, and other networking devices.

Jobs and job data

Deadline Cloud jobs run within sessions on worker hosts. Each session runs one or more processes on the worker host, which generally require that you input data to produce output.

To secure this data, you can configure operating system users with queues. The worker agent uses the queue OS user to run session sub-processes. These sub-processes inherit the queue OS user's permissions.

We recommend that you follow best practices to secure access to the data these sub-processes access. For more information, see Shared responsibility model.

Farm structure

You can arrange Deadline Cloud fleets and queues many ways. However, there are security implications with certain arrangements.

A farm has one of the most secure boundaries because it can't share Deadline Cloud resources with other farms, including fleets, queues, and storage profiles. However, you can share external AWS resources within a farm, which compromises the security boundary.

You can also establish security boundaries between queues within the same farm using the appropriate configuration.

Follow these best practices to create secure queues in the same farm:

- Associate a fleet only with queues within the same security boundary. Note the following:
  - After job runs on the worker host, data may remain behind, such as in a temporary directory or the queue user's home directory.
  - The same OS user runs all the jobs on a service-owned fleet worker host, regardless of which queue you submit the job to.
  - A job might leave processes running on a worker host, making it possible for jobs from other queues to observe other running processes.
- Ensure that only queues within the same security boundary share an Amazon S3 bucket for job attachments.
- Ensure that only queues within the same security boundary share an OS user.
- Secure any other AWS resources that are integrated into the farm to the boundary.

Job attachment queues

Job attachments are associated with a queue, which uses your Amazon S3 bucket.

- Job attachments write to and read from a root prefix in the Amazon S3 bucket. You specify this root prefix in the `CreateQueue` API call.
- The bucket has a corresponding `Queue Role`, which specifies the role that grants queue users access to the bucket and root prefix. When creating a queue, you specify the `Queue Role` Amazon Resource Name (ARN) alongside the job attachments bucket and root prefix.

- Authorized calls to the `AssumeQueueRoleForRead`, `AssumeQueueRoleForUser`, and `AssumeQueueRoleForWorker` API operations return a set of temporary security credentials for the `Queue Role`.

If you create a queue and reuse an Amazon S3 bucket and root prefix, there is a risk of information being disclosed to unauthorized parties. For example, QueueA and QueueB share the same bucket and root prefix. In a secure workflow, ArtistA has access to QueueA but not QueueB. However, when multiple queues share a bucket, ArtistA can access the data in QueueB data because it uses the same bucket and root prefix as QueueA.

The console sets up queues that are secure by default. Ensure that the queues have a distinct combination of Amazon S3 bucket and root prefix unless they're part of a common security boundary.

To isolate your queues, you must configure the `Queue Role` to only allow queue access to the bucket and root prefix. In the following example, replace each *`placeholder`* with your resource-specific information.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME",

 "arn:aws:s3:::JOB_ATTACHMENTS_BUCKET_NAME/JOB_ATTACHMENTS_ROOT_PREFIX/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "111122223333"
                }
            }
        },
        {
            "Action": [
                "logs:GetLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
deadline/FARM_ID/*"
        }
    ]
}
```

You must also set a trust policy on the role. In the following example, replace the *placeholder* text with your resource-specific information.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "deadline.amazonaws.com"
            },
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:deadline:us-
east-1:111122223333:farm/FARM_ID"
                }
            }
        },
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "credentials.deadline.amazonaws.com"
            },
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:deadline:us-east-1:111122223333:farm/FARM_ID"
                }
            }
        }
    ]
}
```

Custom software Amazon S3 buckets

You can add the following statement to your `Queue Role` to access custom software in your Amazon S3 bucket. In the following example, replace *SOFTWARE_BUCKET_NAME* with the name of your S3 bucket and *BUCKET_ACCOUNT_OWNER* with the AWS account ID that owns the bucket.

```
"Statement": [
    {
        "Action": [
            "s3:GetObject",
            "s3:ListBucket"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME",
            "arn:aws:s3:::SOFTWARE_BUCKET_NAME/*"
        ],
        "Condition": {
         "StringEquals": {
            "aws:ResourceAccount": "BUCKET_ACCOUNT_OWNER"
         }
      }
    }
]
```

For more information about Amazon S3 security best practices, see [Security best practices for Amazon S3]() in the *Amazon Simple Storage Service User Guide*.

Worker hosts

Secure worker hosts to help ensure that each user can only perform operations for their assigned role.

We recommend the following best practices to secure worker hosts:

- Using a *host configuration script* can change the security and operations of a worker. An incorrect configuration may cause the worker to be unstable or to stop working. It is your responsibility to debug such failures.
- Don’t use the same `jobRunAsUser` value with multiple queues unless jobs submitted to those queues are within the same security boundary.
- Don’t set the queue `jobRunAsUser` to the name of the OS user that the worker agent runs as.
- Grant queue users least-privileged OS permissions required for the intended queue workloads. Ensure that they don't have filesystem write permissions to work agent program files or other shared software.
- Ensure only the root user on Linux and the `Administrator` owns account on Windows owns and can modify the worker agent program files.
- On Linux worker hosts, consider configuring a `umask` override in `/etc/sudoers` that allows the worker agent user to launch processes as queue users. This configuration helps ensure other users can't access files written to the queue.
- Grant trusted individuals least-privileged access to worker hosts.
- Restrict permissions to local DNS override configuration files (`/etc/hosts` on Linux and `C:\Windows\system32\etc\hosts` on Windows), and to route tables on workstations and worker host operating systems.
- Restrict permissions to DNS configuration on workstations and worker host operating systems.
- Regularly patch the operating system and all installed software. This approach includes software specifically used with Deadline Cloud such as submitters, adaptors, worker agents, OpenJD packages, and others.
- Use strong passwords for the Windows queue `jobRunAsUser`.

- Regularly rotate the passwords for your queue `jobRunAsUser`.
- Ensure least privilege access to the Windows password secrets and delete unused secrets.
- Don't give the queue `jobRunAsUser` permission the schedule commands to run in the future:
  - On Linux, deny these accounts access to `cron` and `at`.
  - On Windows, deny these accounts access to the Windows task scheduler.

> **Note**
>
> For more information about the importance of regularly patching the operating system and installed software, see the Shared Responsibility Model.

Host configuration script

- Using a host configuration script can change the security and operations of a worker. An incorrect configuration may cause the worker to be unstable or to stop working. It is your responsibility to debug such failures.

Workstations

It's important to secure workstations with access to Deadline Cloud. This approach helps ensure that any jobs you submit to Deadline Cloud can't run arbitrary workloads billed to your AWS account.

We recommend the following best practice to secure artist workstations. For more information, see the Shared Responsibility Model.

- Secure any persisted credentials that provide access to AWS, including Deadline Cloud. For more information, see Managing access keys for IAM users in the *IAM User Guide*.
- Only install trusted, secure software.
- Require users federate with an identity provider to access AWS with temporary credentials.
- Use secure permissions on Deadline Cloud submitter program files to prevent tampering.
- Grant trusted individuals least-privileged access to artist workstations.
- Only use submitters and adaptors that you obtain through the Deadline Cloud Monitor.

- Restrict permissions to local DNS override configuration files (`/etc/hosts` on Linux and macOS, and `C:\Windows\system32\etc\hosts` on Windows), and to route tables on workstations and worker host operating systems.
- Restrict permissions to `/etc/resolve.conf` on workstations and worker host operating systems.
- Regularly patch the operating system and all installed software. This approach includes software specifically used with Deadline Cloud such as submitters, adaptors, worker agents, OpenJD packages, and others.

Verify the authenticity of downloaded software

Verify your software's authenticity after downloading the installer to protect against file tampering. This procedure works for both Windows and Linux systems.

Windows

To verify the authenticity of your downloaded files, complete the following steps.

1. In the following command, replace *file* with the file that you want to verify. For example, ***C:\PATH\TO\MY\*****DeadlineCloudSubmitter-windows-x64-installer.exe** . Also, replace *signtool-sdk-version* with the version of the SignTool SDK installed. For example, **10.0.22000.0**.

```
"C:\Program Files (x86)\Windows Kits\10\bin\signtool-sdk-
version\x86\signtool.exe" verify /vfile
```

2. For example, you can verify the Deadline Cloud submitter installer file by running the following command:

```
"C:\Program Files (x86)\Windows Kits\10\bin
\10.0.22000.0\x86\signtool.exe" verify /v DeadlineCloudSubmitter-
windows-x64-installer.exe
```

Linux

To verify the authenticity of your downloaded files, use the gpg command line tool.

1. Import the `OpenPGP` key by running the following command:

```
 gpg --import --armor <<EOF
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGlANDUBEACg6zffjN43gqe5ryPhk+wQM10rEdvmItw4WPWaVsN+/at/OIJw
MGCagSYXcgR+jKbsHQOQoEQdo5SrxxHjpKTEs3KQhGvf+ehrU1Ac7koXKIBWtes+
BI9F0slRECz0nXTOy/cd/90RXjpF07mreTLIKNIbybULfad82nYykpITjFr5XRGj
/shYkucxRQZdwkgkIYyV25pPICPd2RsX+Zua85jV8mCqVffDfRXvgcPe3+ofClj/
2CE8UfUIqO8Csua4YEkSqr3aeoTOEFT4kuQR5nFXVzorOEkQtO3gB35KNWKMlIOU
2vA+wyoL7nWSii4yfYtW3EZ+3gq6HxvnT9Zs8MC53uTOiOdamASXecYREwGmY/io
6n5XTEA/35LNbl4A756vSTZ7h4VFJAN5BpuqxstI1D7ou94skoSmcPoC/iniTvY9
kZylU5OCH/nifMAHM2a5jrQel80cW4oko9eyc8ENQpSy15JElFOKFf7D/4tcZJLF
F0VBTXbhfvq3dPfoq94IWt7p54Ovwj0S//CEu3jZYbNl2QC/3YiHE2H2XyGCQbq6
2MjcuxLnEapoRIqfbi8GPtCWVPzm28WGyKIDofWICczzeJFFJnvzrY3wRG64ibKJ
bR/uedwua1UuiC482V1FD5ffmzSSs8ktTp9hgj7RGDXlc9NTcF1jHxG9hwARAQAB
tCxBV1MgRGVhZGxpbmUgQ2xvdWQgPGF3cy1kZWFkbGluZUBhbWF6b24uY29tPokC
VwQTAQgAQRYhBJmXd7So2csyehiIYsg71N18bhtjBQJpQDQ1AhsvBQkDwmcABQsJ
CAcCAiICBhUKCQgLAgQWAgMBAh4HAheAAAoJEMg71N18bhtjk2UP/3h4KlEzZO/7
BxRmkbixuo1QuqOGvA6tXbSWaM8QH5jglcvL12PZLALklLT4v82uCsLR1lF8/Tch
cCl0SZEOFIS+XxAaw1Xfai6jlyLhabOwKF2ylq5eJlLcw1lh2nAArDRb4fLD0m1g
Dfqetq/XEpyXpOSkWxGRV4RlUdjQfytxrmcUnsT5/fk5f9VDdblu6K/lEmwfyYjB
lXv0uUCkqPot0SmbvOh3PY3Hi3n54ncy8NfTeV+TUvSe3C1s1zNl8aqHoTxJB/eU
kp+LFZ9m+igpSYnKeglKnytylH3KGCjTHglT/QXnI1wNTqmj1kFBVwtt/y1mtnA+
CPIUHP1CtbKsHaLtpp4llBm5TVtPN/Wqqicn5QLl4khg7R4K+V2aaA4ubY6p1tG9
0fFhN5tTnHDSKWMfmb83wfh5Zkcg85c3egjoit+wgGQRAQVqbznx7NqAHs9VoDIu
SPcAr+C329AOBzod4gyNGH7Ah5DkMITo4O4+axnAU9yhFOHcMJmTIask/fNg1Aum
OqYPMUwcgv1GZjLaTJyfGGC1xALsYR0KHnwIehD06MHR/Z98bGkcV8+Y0q8UPsd1
VN1fc1rjCJh/AT3w6owvG4DaEwspseSjzHv16mW4e2N6Uu23SPzgQsJ5qYN2g8D+
P7N9LGDfP8DaYc5JM9mlyFmYI2Q94ufL
=rY5l
-----END PGP PUBLIC KEY BLOCK-----
EOF
```

2. Determine whether to trust the OpenPGP key. Some factors to consider when deciding whether to trust the above key include the following:

   - The internet connection you've used to obtain the GPG key from this website is secure.
   - The device that you are accessing this website on is secure.
   - AWS has taken measures to secure the hosting of the OpenPGP public key on this website.

3. If you decide to trust the OpenPGP key, edit the key to trust with gpg similar to the following example:

```
$ gpg --edit-key 0xB840C08C29A90796A071FAA5F6CD3CE6B76F3CEF

    gpg (GnuPG) 2.0.22; Copyright (C) 2013 Free Software Foundation, Inc.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.


    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud example@example.com

    gpg> trust
    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: unknown       validity: unknown
    [ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com

    Please decide how far you trust this user to correctly verify other users'
 keys
    (by looking at passports, checking fingerprints from different sources,
 etc.)

      1 = I don't know or won't say
      2 = I do NOT trust
      3 = I trust marginally
      4 = I trust fully
      5 = I trust ultimately
      m = back to the main menu

    Your decision? 5
    Do you really want to set this key to ultimate trust? (y/N) y

    pub  4096R/4BF0B8D2  created: 2023-06-23  expires: 2025-06-22  usage: SCEA
                         trust: ultimate      validity: unknown
    [ unknown] (1). AWS Deadline Cloud aws-deadline@amazon.com
    Please note that the shown key validity is not necessarily correct
    unless you restart the program.

    gpg> quit
```

4. **Verify the Deadline Cloud submitter installer**

   To verify the Deadline Cloud submitter installer, complete the following steps:

a. Download the signature file for the Deadline Cloud submitter installer.

   [Download signature file (.sig)]

b. Verify the signature of the Deadline Cloud submitter installer by running:

```
gpg --verify ./DeadlineCloudSubmitter-linux-x64-installer.run.sig ./
DeadlineCloudSubmitter-linux-x64-installer.run
```

5. **Verify the Deadline Cloud monitor**

> **Note**
>
> You can verify the Deadline Cloud monitor download using signature files or platform specific methods. For platform specific methods, see the Linux (Debian) tab, the Linux (RPM) tab, or the Linux (AppImage) tab based on your downloaded file type.

To verify the Deadline Cloud monitor desktop application with signature files, complete the following steps:

a. Download the corresponding signature file for your Deadline Cloud monitor installer:

   - Download .deb signature file
   - Download .rpm signature file
   - Download .AppImage signature file

b. Verify the signature:

   For .deb:

```
gpg --verify ./deadline-cloud-monitor_amd64.deb.sig ./deadline-cloud-
monitor_amd64.deb
```

   For .rpm:

```
gpg --verify ./deadline-cloud-monitor.x86_64.rpm.sig ./deadline-cloud-
monitor.x86_64.rpm
```

For .AppImage:

```
gpg --verify ./deadline-cloud-monitor_amd64.AppImage.sig ./deadline-cloud-monitor_amd64.AppImage
```

c. Confirm that the output looks similar to the following:

```
gpg: Signature made Mon Apr 1 21:10:14 2024 UTC

gpg: using RSA key B840C08C29A90796A071FAA5F6CD3CE6B7
```

If the output contains the phrase `Good signature from "AWS Deadline Cloud"`, it means that the signature has successfully been verified and you can run the Deadline Cloud monitor installation script.

Historical Keys

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGX6GQsBEADduUtJgqSXI+q76O6fsFwEYKmbnlyL0xKvlq32EZuyv0otZo5L
le4m5Gg52AzrvPvDiUTLooAlvYeozaYyirIGsK08Ydz0Ftdjroiuh/mw9JSJDJRI
rnRn5yKet1JFezkjopA3pjsTBP6lW/mb1bDBDEwwwtH0x9lV7A03FJ9T7Uzu/qSh
qO/UYdkafro3cPASvkqgDt2tCvURfBcUCAjZVFcLZcVD5iwXacxvKsxxS/e7kuVV
I1+VGT8Hj8XzWYhjCZxOLZk/fvpYPMyEEujN0fYUp6RtMIXve0C9awwMCy5nBG2J
eE2Ol5DsCpTaBd4Fdr3LWcSs8JFA/YfP9auL3NczOozPoVJt+fw8CBlVIXO0J7l5
hvHDjcC+5v0wxqAlMG6+f/SX7CT8FXK+L3iOJ5gBYUNXqHSxUdv8kt76/KVmQa1B
Akl+MPKpMq+lhw++S3G/lXqwWaDNQbRRw7dSZHymQVXvPp1nsqc3hV7KlOM+6s6g
1g4mvFY4lf6DhptwZLWyQXU8rBQpojvQfiSmDFrFPWFi5BexesuVnkGIolQoklKx
AVUSdJPVEJCteyy7td4FPhBaSqT5vW3+ANbr9b/uoRYWJvn17dN0cc9HuRh/Ai+I
nkfECo2WUDLZ0fEKGjGyFX+todWvJXjvc5kmE9Ty5vJp+M9Vvb8jd6t+mwARAQAB
tCxBV1MgRGVhZGxpbmUgQ2xvdWQgPGF3cy1kZWFkbGluZUBhbWF6b24uY29tPokC
VwQTAQgAQRYhBLhAwIwpqQeWoHH6pfbNPOa3bzzvBQJl+hkLAxsvBAUJA8JnAAUL
CQgHAgIiAgYVCgkICwIDFgIBAh4HAheAAAoJEPbNPOa3bzzvKswQAJXzKSAY8sY8
F6Eas2oYwIDDdDurs8FiEnFghjUEO6MTt9AykF/jw+CQg2UzFtEyObHBymhgmhXE
3buVeom96tgM3ZDfZu+sxi5pGX6oAQnZ6riztN+VpkpQmLgwtMGpSMLl3KLwnv2k
WK8mrR/fPMkfdaewB7A6RIUYiW33GAL4KfMIs8/vIwIJw99NxHpZQVoU6dFpuDtE
1OuxGcCqGJ7mAmo6H/YawSNp2Ns80gyqIKYo7o3LJ+WRroIRlQyctq8gnR9JvYXX
42ASqLq5+OXKo4qh81blXKYqtc176BbbSNFjWnzIQgKDgNiHFZCdcOVgqDhwO15r
NICbqqwwNLj/Fr2kecYx180Ktpl0jOOw5IOyh3bf3MVGWnYRdjvA1v+/CO+55N4g
z0kf50Lcdu5RtqV10XBCifn28pecqPaSdYcssYSRl5DLiFktGbNzTGcZZwITTKQc
```

```
af8PPdTGtnnb6P+cdbW3bt9MVtN5/dgSHLThnS8MPEuNCtkTnpXshuVuBGgwBMdb
qUC+HjqvhZzbwns8dr5WI+6HWNBFgGANn6ageYl58vVp0UkuNP8wcWjRARciHXZx
ku6W2jPTHDWGNrBQO2Fx7fd2QYJheIPPAShHcfJO+xgWCof45D0vAxAJ8gGg9Eq+
gFWhsx4NSHn2gh1gDZ41Ou/4exJ1lwPM
=uVaX
-----END PGP PUBLIC KEY BLOCK-----
EOF
```

Linux (AppImage)

To verify packages that use a Linux .AppImage binary, first complete steps 1-3 in the Linux tab, then complete the following steps.

1. From the AppImageUpdate page in GitHub, download the **validate-x86_64.AppImage** file.
2. After downloading the file, to add execute permissions, run the following command.

   ```
   chmod a+x ./validate-x86_64.AppImage
   ```

3. To add execute permissions, run the following command.

   ```
   chmod a+x ./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
   ```

4. To verify the Deadline Cloud monitor signature, run the following command.

   ```
   ./validate-x86_64.AppImage ./deadline-cloud-monitor_<APP_VERSION>_amd64.AppImage
   ```

   If the output contains the phrase `Validation successful`, it means that the signature has successfully been verified and you can safely run the Deadline Cloud monitor installation script.

Linux (Debian)

To verify packages that use a Linux .deb binary, first complete steps 1-3 in the Linux tab.

dpkg is the core package management tool in most debian based Linux distributions. You can verify the .deb file with the tool.

1. Download the Deadline Cloud monitor .deb file:

   Download Deadline Cloud monitor (.deb)

2. Verify the .deb file:

   ```
   dpkg-sig --verify deadline-cloud-monitor_amd64.deb
   ```

3. The output will be similar to:

   ```
   Processing deadline-cloud-monitor_amd64.deb...
   GOODSIG _gpgbuilder B840C08C29A90796A071FAA5F6CD3C 171200
   ```

4. To verify the .deb file, confirm that `GOODSIG` is present in the output.

Linux (RPM)

To verify packages that use a Linux .rpm binary, first complete steps 1-3 in the Linux tab.

1. Download the Deadline Cloud monitor .rpm file:

   Download Deadline Cloud monitor (.rpm)

2. Verify the .rpm file:

   ```
   gpg --export --armor "Deadline Cloud" > key.pub
   sudo rpm --import key.pub
   rpm -K deadline-cloud-monitor.x86_64.rpm
   ```

3. The output will be similar to:

   ```
   deadline-cloud-monitor.x86_64.rpm: digests signatures OK
   ```

4. To verify the .rpm file, confirm that `digests signatures OK` is in the output.

Monitoring AWS Deadline Cloud

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Deadline Cloud (Deadline Cloud) and your AWS solutions. Collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. Before you start monitoring Deadline Cloud, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- Which resources will you monitor?
- How often will you monitor these resources?
- Which monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

AWS and Deadline Cloud provide tools that you can use to monitor your resources and respond to potential incidents. Some of these tools do the monitoring for you, some of the tools require manual intervention. You should automate monitoring tasks as much as possible.

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.

  Deadline Cloud has three CloudWatch metrics.
- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.
- *Amazon EventBridge* can be used to automate your AWS services and respond automatically to system events, such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near real time. You can write simple rules to indicate

which events are of interest to you and which automated actions to take when an event matches a rule. For more information, see Amazon EventBridge User Guide.

- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

For more information, see the following topics in the *Deadline Cloud Developer Guide*:

- CloudTrail logs
- Managing events using EventBridge
- Monitoring with CloudWatch

Quotas for Deadline Cloud

AWS Deadline Cloud provides resources, such as farms, fleets, and queues, that you can use to process jobs. When you create your AWS account, we set default quotas on these resources for each AWS Region.

Service Quotas is a central location where you can view and manage your quotas for AWS services. You can also request a quota increase for many of the resources that you use.

To view the quotas for Deadline Cloud, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **Deadline Cloud**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, use the service quota increase form.

Your AWS account has the following quotas related to Deadline Cloud.

Name	Default	Adjuse	Description
Associated members per farm	Each supported Region: 75	No	The maximum number of members that can be associated to each farm in the current AWS Region.
Associated members per fleet	Each supported Region: 75	No	The maximum number of members that can be associated to each fleet in the current AWS Region.
Associated members per job	Each supported Region: 75	No	The maximum number of members that can be associated to each job in the current AWS Region.
Associated members per queue	Each supported Region: 75	No	The maximum number of members that can

Name	Default	Adjus e	Description
			be associated to each queue in the current AWS Region.
Budgets per farm	Each supported Region: 20	Yes	The maximum number of budgets per farm in the current AWS Region
Farms per region	Each supported Region: 2	Yes	The maximum number of farms that can be created in the current AWS Region.
Fleets per farm	Each supported Region: 5	Yes	The maximum number of fleets that can be created for each farm in the current AWS Region.
Jobs per farm	Each supported Region: 100,000	Yes	The maximum number of jobs per farm in the current AWS Region.
License endpoints per region	Each supported Region: 5	Yes	The maximum number of license endpoints in the current AWS Region.
License sessions per license endpoint	Each supported Region: 500	Yes	The maximum number of license sessions per license endpoint in the current AWS Region.
Limits per farm	Each supported Region: 50	Yes	The maximum number of limits that can be created for each farm in the current AWS Region.

Name	Default	Adjustable	Description
Monitors per region	Each supported Region: 1	No	The maximum number of monitors in the current AWS Region.
OnDemand G instance GPUs per region	Each supported Region: 1	Yes	The maximum number of on-demand G instance GPUs that can be provisioned across all service-managed fleets in the current AWS Region.
OnDemand vCPUs per region	Each supported Region: 50	Yes	The maximum number of on-demand vCPUs that can be provisioned across all service-managed fleets in the current AWS Region.
Queue environments per queue	Each supported Region: 10	No	The maximum number of queue environments that can be created for each queue in the current AWS Region.
Queue fleet associations per farm	Each supported Region: 100	Yes	The maximum number of queue fleet associations per farm in the current AWS Region
Queue limit associations per queue	Each supported Region: 10	Yes	The maximum number of limits that can be associated with each queue in the current AWS Region.

Name	Default	Adjus e	Description
Queues per farm	Each supported Region: 20	Yes	The maximum number of queues that can be created for each farm in the current AWS Region.
Resource configurations per fleet	Each supported Region: 1	Yes	The maximum number of VPC Lattice resource configurations that can be added to each fleet.
Spot G Instance GPUs per region	Each supported Region: 1	Yes	The maximum number of spot G instance GPUs that can be provision ed across all service-managed fleets in the current AWS Region.
Spot vCPUs per region	Each supported Region: 500	Yes	The maximum number of spot vCPUs that can be provisioned across all service-managed fleets in the current AWS Region.
Steps per job	Each supported Region: 200	Yes	The maximum number of steps per job in the current AWS Region.
Storage for General Purpose SSD (gp3) volumes, in TiB	Each supported Region: 50	Yes	The maximum aggregate d amount of EBS storage, measured in TiB, that can be used across all fleets in the current AWS Region.

Name	Default	Adjus e	Description
Storage profiles per farm	Each supported Region: 50	No	The maximum number of storage profiles that can be created for each farm in the current AWS Region.
Tasks per chunk	Each supported Region: 150	No	The maximum number of tasks that can be combined into a single chunk when submitting a job.
Tasks per job	Each supported Region: 10,000	Yes	The maximum number of tasks per job in the current AWS Region.
Tasks per step	Each supported Region: 10,000	Yes	The maximum number of tasks per step in the current AWS Region.
Wait-and-save vCPUs per region	Each supported Region: 50	Yes	The maximum number of wait-and-save vCPUs that can be provision ed across all service-managed fleets in the current AWS Region.
Workers per farm	Each supported Region: 7,500	Yes	The maximum number of workers per farm in the current AWS Region.

Creating AWS Deadline Cloud resources with AWS CloudFormation

AWS Deadline Cloud is integrated with AWS CloudFormation, a service that helps you to model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want (such as farms, queues, and fleets), and CloudFormation provisions and configures those resources for you.

When you use CloudFormation, you can reuse your template to set up your Deadline Cloud resources consistently and repeatedly. Describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

Deadline Cloud and CloudFormation templates

To provision and configure resources for Deadline Cloud and related services, you must understand CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use CloudFormation Designer to help you get started with CloudFormation templates. For more information, see What is CloudFormation Designer? in the *AWS CloudFormation User Guide*.

Deadline Cloud supports creating farms, queues, and fleets in CloudFormation. For more information, including examples of JSON and YAML templates for farms, queues, and fleets, see the AWS Deadline Cloud in the *AWS CloudFormation User Guide*.

Learn more about CloudFormation

To learn more about CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- CloudFormation API Reference
- AWS CloudFormation Command Line Interface User Guide

Troubleshooting

The following procedures and tips can help you troubleshoot issues with your AWS Deadline Cloud farms and resources.

Topics

Why can a user not see my farm, fleet, or queue?

User access

When your users are not seeing your farms, fleets, or queues in the Deadline Cloud monitor, there might be an issue with their access to your farm and resources.

Users without access to any farms receive the message "No farms available" in the Deadline Cloud monitor.

To confirm you have the correct user or group assigned to your farm, fleet, or queue

1. In the AWS Deadline Cloud console, find your farm, fleet, or queue, and then choose **Access management**.
2. The groups tab is selected by default. If you're assigning permissions by groups, which is recommended, your group should display in the list and have an assigned access level.

   If the group is not in the list, choose **Add group** to assign permission for the group.
3. If you're assigning permissions by user, select the **Users** tab. Your user should display in the list and have an assigned access level.

   If your user is not in the list, choose **Add user** to assign permission for the user.

To confirm you have the user assigned to your group

1. In the AWS Deadline Cloud console, find your farm, fleet, or queue, and then choose **Access management**.
2. The groups tab is selected by default. Select the group name to view its members.
3. If the user is not listed in the group, they must be added.

   If you're using the default identity setup, you can directly add the user to the group in the Identity Center console. If you're connected to an external identity provider such as Okta or Google Workspace, you can add your user to the group in your identity provider.

   > **Note**
   >
   > Some external identity providers sync users but not groups to Identity Center. In this case, consider assigning permissions to a user directly instead of by group.

For more information about managing user access to Deadline Cloud, see Managing users in Deadline Cloud.

Why are workers not picking up my jobs?

Fleet role configuration

Sometimes when workers are created but do not complete initialization and do not start working on jobs, it's because the fleet role was not configured correctly.

To verify this is what is happening, check your CloudTrail logs for any access denied errors. After you confirm the access denied issue, go to your fleet and update the role configuration to the correct permissions. For more information, see CloudTrail logs in the Deadline Cloud developer guide.

Why is my worker stuck running?

Worker stuck exiting OpenJD environment

Workers can get stuck in long-running `envExit` session actions. This might happen if you use a job template that overrides the OpenJD template and sets the environment exit actions timeout to more than 5 minutes. The Deadline Cloud monitor provides some visibility into workers stuck in this situation, but it requires cross-referencing `RUNNING` workers against available work in the associated queues.

To find stuck workers, go through all fleets in the Deadline Cloud monitor and complete the following steps:

1. In the worker status column, find `RUNNING` workers.
2. From the Fleet details section, navigate to each associated queue.
3. In each associated queue, search for jobs that are `RUNNING`, `READY`, or `PENDING`. If all associated queues don't have any jobs in those states, then the worker is running an environment exit.

To stop a worker stuck in this state, use the following AWS CLI command:

```
aws deadline update-worker \
    --farm-id $FARM_ID     \
    --fleet-id $FLEET_ID   \
    --worker-id $WORKER_ID \
    --status STOPPED
```

After running the command, the worker agent restarts when the program exits. Workers then come back online and run more jobs from associated queues. If the queue contains more jobs with environment exit action timeouts longer than 5 minutes, the worker will get stuck again. If this happens, you will need to repeat this process until no more workers are stuck exiting.

To avoid this issue, set the timeout option to no more than 5 minutes when using a job template.

Troubleshooting Deadline Cloud jobs

For information about common problems with jobs in AWS Deadline Cloud, see the following topics.

Why did creating my job fail?

Quota validation

Some possible reasons that a job can fail validation checks include the following:

- The job template doesn't follow the OpenJD specification.
- The job contains too many steps.
- The job contains too many total tasks.
- There was an internal service error that prevents the job from being created.

To see the quotas for the maximum number of steps and tasks in a job, use the Service Quotas console. For more information, see Quotas for Deadline Cloud.

CHUNK[INT] task parameter error

If job creation fails with the following error message, you need to add the TASK_CHUNKING extension to your job template.

```
The CHUNK[INT] task parameter requires the TASK_CHUNKING extension.
```

To resolve this issue, add the following to your job template:

```
extensions:
  - TASK_CHUNKING
```

Why is my job not compatible?

Common reasons that jobs are not compatible with queues include the following:

- No fleets are associated with the queue that the job was submitted to. Open the Deadline Cloud monitor, and check that the queue has associated fleets. For more information about how to view queues, see View queue and fleet details in Deadline Cloud.
- The job has host requirements that are not satisfied by any of the fleets associated with the queue. To check, compare the hostRequirements entry in the job template with the configuration of the fleets in your farm. Make sure that one of the fleets satisfies the host requirements. For more information about fleet compatibility, see Determine fleet compatibility. To view fleet configuration, see View queue and fleet details in Deadline Cloud.

Why is my job stuck in ready?

Possible reasons for your job appearing to be stuck in the READY state include the following:

- The maximum worker count for fleets associated with the queue is set to zero. To check, see View queue and fleet details in Deadline Cloud.
- There is a higher priority job in the queue. To check, see View queue and fleet details in Deadline Cloud.
- For customer-managed fleets, check the auto scaling configuration. For more information, see Create fleet infrastructure with an Amazon EC2 Auto Scaling group in the *Deadline Cloud Developer Guide*.

Why did my job fail?

A job can fail for many reasons. To search for the issue, open the Deadline Cloud monitor and choose the failing job. Choose a task that failed and then view the logs for the task. For instructions, see View session and worker logs in Deadline Cloud.

- If you see license errors or if you get a watermark that occurs because the software doesn't have a valid license, make sure that the worker can connect to the required license server. For more information, see Connect customer-managed fleets to a license endpoint in the *Deadline Cloud Developer Guide*.
- The last session action message or the process exit code may provide information about why you job failed. If you are using Windows and your exit code is negative, try searching for the unsigned version of your exit code:

```
2,147,483,647 - |your exit code|
```

Why is my step pending?

Steps may stay in the `PENDING` state when one or more of their dependencies are not complete. You can check the state of dependencies using the Deadline Cloud monitor. For instructions, see View a step in Deadline Cloud.

Deadline Cloud monitor desktop application logs

The Deadline Cloud monitor desktop application writes diagnostic logs that you can use to investigate crashes or other unexpected behavior. When reporting an issue with the desktop application, include the relevant log files to help with diagnosis.

The location of the log files depends on your operating system:

Windows

```
%APPDATA%\com.amazonaws.deadline.monitor\logs
```

macOS

```
~/Library/Logs/com.amazonaws.deadline.monitor/
```

Linux

```
~/.config/com.amazonaws.deadline.monitor/logs
```

Additional resources

You can find additional information and resources on GitHub.

Deadline Cloud release notes

This page contains information about the latest releases and updates to AWS Deadline Cloud.

Date	Title	Description
2026-03-02	Submitter Installer v2026-03-02 Released	A new submitter installer has been released which updates the following components: • blender: 0.6.0 → 0.6.1 (release notes) • deadline-cloud: 0.54.0 → 0.54.1 (release notes)
2026-02-24	Deadline Cloud documentation now includes supported software user guide	Deadline Cloud user guide now includes dedicated subpages for each supported application, providing in-depth details on version compatibility and feature support.
2026-02-24	Deadline Cloud monitor usage explorer now supports grouping usage by user	Analyze usage patterns per user and attribute costs across your team with the usage explorer.
2026-02-24	Deadline Cloud now supports task chunking for improved rendering efficiency	AWS Deadline Cloud now supports task chunking, which groups multiple frames into a single task run. This feature reduces overhead by loading the application and scene once per chunk instead of once per frame. You can specify a default chunk size or

Date	Title	Description
		let Deadline Cloud dynamically adjust chunk sizes based on a target runtime.
2026-02-19	Submitter Installer v2026-02-19 Released	A new submitter installer has been released which updates the Autodesk Maya submitter from 0.15.12 to 0.15.13.
2026-02-13	OpenJD Specifications now includes a Claude and Kiro skill for RFC review	The Open Job Description specifications repository now includes a Kiro skill for AI-assisted review of RFC proposals, checking for completeness, clarity, tenet alignment, and compatibility with existing specifications.
2026-02-13	Deadline Cloud for 3ds Max adds Kiro powers for AI-assisted development	The Deadline Cloud for 3ds Max repository now includes Kiro powers that provide AI-assisted setup, design, and development workflows with built-in guardrails and best practices.
2026-02-06	Deadline Cloud adds job tagging for access control	Job resources now support tagging and Attribute-Based Access Control (ABAC). IAM policies can reference job tags using condition keys, enabling tag-based authorization patterns — for example, restricting GetJob API calls to jobs with a particular team tag.

Date	Title	Description
2026-02-05	Deadline Cloud now supports IAM Identity Center multi-region replication	AWS Deadline Cloud now supports IAM Identity Center's multi-region replication feature, giving studios more flexibility in where they set up Deadline Cloud relative to their Identity Center instance. Studios can create a Deadline Cloud farm in regions that align with their rendering needs while administrators continue managing Identity Center from a primary region.
2026-02-04	Sample job bundle for FLUX.2 Klein LoRA training now available	A sample job bundle is now available that demonstrates how to train custom LoRA adapters on the FLUX.2 Klein model using 20-50 images. This enables you to create personalized image generators for products, characters, or brand assets without requiring deep machine learning expertise. The LoRA fine-tuning approach creates small, portable model adapters that are efficient to train and easy to share across your team.

Date	Title	Description
2026-01-29	V-Ray Standalone tiled rendering job bundle now available	A new job bundle for tiled rendering of exported V-Ray scenes is now available. This job bundle enables efficient rendering of high-resolution images by splitting them into tiles that can be processed in parallel across your render farm. Customers using 3ds Max and V-Ray can export V-Ray scenes locally and submit them using this bundle to Linux workers instead of needing to use Windows.
2026-01-27	Deadline Cloud now supports editing job name and description	AWS Deadline Cloud now supports editing job names and descriptions after submission. This new feature makes it easier to organize and identify jobs after submission by updating names or adding useful tracking details in the description field.
2026-01-22	Redshift 2026 support on Deadline Cloud for Maya	Redshift 2026 is now supported on Linux service-managed fleets with Deadline Cloud for Maya.

Date	Title	Description
2026-01-22	Deadline Cloud now supports machine learning training using Foundry Nuke CopyCat	Deadline Cloud now integrate s with Foundry Nuke CopyCat, enabling you to run ML training jobs for visual effects in the cloud. CopyCat learns adjustments from sample frames and applies them across entire sequences . Submit training jobs to your Deadline Cloud render farm, scale workloads in parallel, and free up your artist workstations.
2026-01-15	Deadline Cloud SDKs now include waiters for job completion	AWS Deadline Cloud SDKs now include JobComplete and JobSucceeded waiters that simplify polling for job status. The JobComplete waiter polls until a job reaches any terminal state (SUCCEEDED, FAILED, or CANCELED), while the JobSucceeded waiter polls until a job succeeds. These waiters eliminate the need to write custom polling logic, making it easier to build automation workflows that depend on job completion.

Date	Title	Description
2026-01-15	Deadline Cloud now supports tagging Budgets	AWS Deadline Cloud customers can now apply tags to Budget resources and use Attribute-Based Access Control (ABAC) for fine-grained permissions management. This new capability allows customers to organize, manage, and control access to their Deadline Cloud budgets using tags, enabling consistent authorization patterns across their AWS resources. Customers can now tag budgets during creation and use these tags in IAM policies to control who can access specific budgets based on tag values.
2026-01-15	Deadline Cloud monitor search now supports multi-select filtering	When using the Deadline Cloud monitor, you can now select up to 16 values for any search filter, including user names and job status. This allows you to quickly find jobs across multiple users or filter for several statuses at once. This functionality is also available in the Deadline Cloud API through the new StringListFilterExpression for Jobs, Steps, Tasks, and Workers.

Date	Title	Description
2026-01-07	Deadline Cloud documentation now includes direct Deadline Cloud Monitor and submitter installer download links	Users can now download the Deadline Cloud Monitor desktop application and submitter installer directly from the Deadline Cloud documentation. This enables users without AWS console access to download the software they need to get started with Deadline Cloud.
2025-12-19	Submitter Installer v2025-12-19 Released	A new submitter installer has been released which updates the following components: • cinema-4d: 0.9.0 → 0.9.2 (release notes) • deadline-cloud: 0.53.3 → 0.54.0 (release notes) • nuke: 0.18.13 → 0.18.14 (release notes)

Date	Title	Description
2025-12-17	Deadline Cloud Monitor 1.1.7 - Integrated Job Submission	The latest Deadline Cloud Monitor desktop application release includes: - Support for submitting jobs directly from the Deadline Cloud Monitor desktop application. - Simpler workstation setup. - Improved proxy support. - Bug fixes for edge cases when reading and writing to and from the Deadline Cloud profile config file.
2025-12-11	Deadline Cloud developer guide now includes guidance on using AI agents	The Deadline Cloud developer guide now includes best practices for using AI agents with AWS Deadline Cloud in order to write job bundles, develop conda packages, and troubleshoot jobs more efficiently.
2025-12-10	User guide now available for Autodesk VRED submitter	Documentation for the AWS Deadline Cloud submitter for Autodesk VRED is now available. The guide covers how to install the submitter and submit rendering jobs to Deadline Cloud. This helps VRED users get started quickly with cloud rendering.

Date	Title	Description
2025-12-10	Deadline Cloud documentation now includes LicensesInUse metric	The Deadline Cloud documentation now includes information about the LicensesInUse metric. This metric helps you monitor how many licenses your jobs are currently consuming across your fleets. You can use this information to optimize license usage and to avoid running out of licenses when scaling out workloads.
2025-12-10	Cinema 4D 2026.1 Support on Service-Managed Fleets	Maxon Cinema 4D 2026.1 is now supported on Linux and Windows service-managed fleets. This release includes Redshift 2026.2.0. Cross-platform font rendering support has also been added for all versions of Cinema 4D. This release allows customers to use the latest Cinema 4D features. It also enables customers to use custom fonts on cross-platform setups, such as when submitting jobs from Windows while using the faster startup time and lower cost of Linux workers.

Date	Title	Description
2025-12-09	Enhanced Autodesk Maya Submitter Setup and Usage Documentation	New setup and usage documentation added for the AWS Deadline Cloud Submitter for Autodesk Maya.
2025-12-09	After Effects submitter 0.4.4 improves macOS installation and font support	The After Effects submitter now automatically installs to the user preferences directory on macOS, removing the need for manual installation. This release also adds support for most TrueType Collectio n (TTC) font files, allowing you to submit and render jobs that use these fonts. These improvements simplify setup and expand font compatibility for After Effects users.
2025-12-08	Deadline Cloud Release Notes	All major changes to Deadline Cloud features, applicati ons, integrations, samples and documentation will now be listed on the Release Notes page in the User Guide going forward. You can find previous major Deadline Cloud releases at AWS What's New and CLI/ Worker/integration-specific release notes in the repositor ies of the Deadline Cloud github organization

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.