Implementation Guide

Cost Optimizer for Amazon WorkSpaces

Cost Optimizer for Amazon WorkSpaces: Implementation Guide

Table of Contents

Solution overview **1**
Features and benefits 2
Dry run mode 2
Automatic billing conversion 2
Unused WorkSpaces termination 4
Opt out WorkSpaces 6
Opt in Regions 6
Deployment in an existing Amazon VPC 6
Maintenance calculation 6
Use cases 7
Architecture overview **8**
Architecture diagram 8
AWS Well-Architected design framework 10
AWS Organizations support 12
AWS services in this solution 13
Plan your deployment **15**
Supported AWS Regions 15
Cost 16
CUDOS Dashboard 16
Sample cost table: Scenario 1 16
Security 19
IAM roles 19
Hub template 19
Spoke template 20
Quotas 20
Deploy the solution **21**
Deployment process overview 21
AWS CloudFormation templates 22
Hub account 22
Spoke accounts 22
Step 1: Launch the hub stack 23
Step 2: Launch the spoke stack 29
Monitor the solution **32**
Operation Insights Dashboard 32

Update the solution ... 35
Troubleshooting ... 36
Known issue resolution ... 36
Failed WorkSpaces ... 36
FAQs ... 36
Redeploying after deletion ... 37
Contact AWS Support ... 37
Create case ... 37
How can we help? ... 37
Additional information ... 38
Help us resolve your case faster ... 38
Solve now or contact us ... 38
Uninstall the solution ... 39
Using the AWS Management Console ... 39
Using AWS Command Line Interface ... 39
Developer guide ... 40
Source code ... 40
Container image ... 40
Maintenance ... 40
Versions ... 40
Reference ... 41
Data collection ... 41
Contributors ... 41
Revisions ... 42
Notices ... 43

Monitor Amazon WorkSpaces usage and optimize costs with the Cost Optimizer for Amazon WorkSpaces solution

The Cost Optimizer for Amazon WorkSpaces solution analyzes all of your WorkSpaces usage data and automatically converts the WorkSpace to the most cost-effective billing option (hourly or monthly), depending on your individual usage. Amazon WorkSpaces is a fully managed, secure Desktop-as-a-Service (DaaS) offering that eliminates the need for you to procure, deploy, and manage complex virtual desktop environments.

This solution helps you monitor your WorkSpaces usage and optimize costs and uses AWS CloudFormation to automatically provision and configure the necessary Amazon Web Services, Inc. (AWS) services to convert the billing mode for individual WorkSpaces. This solution supports a multi-account environment in AWS Organizations and is capable of running in the AWS GovCloud (US) Regions.

This implementation guide provides architectural considerations and configuration steps for deploying this solution on the AWS Cloud. It includes links to CloudFormation templates that launch, configure, and run the AWS compute, management, storage, and other services required to deploy this solution on AWS, using AWS best practices for security and availability.

This guide is intended for solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals who want to Cost Optimizer for Amazon WorkSpaces in their environment.

Use this navigation table to quickly find answers to these questions:

If you want to . . .	Read . . .
Know the cost for running this solution The estimated cost for running this solution in the US East (N. Virginia) Region is USD $5.00 per month.	Cost
Understand the security considerations for this solution	Security

If you want to . . .	Read . . .
Know how to deploy the solution	Deploy the solution
View or download the AWS CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution	AWS CloudFormation templates
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution	GitHub repository

Features and benefits

The Cost Optimizer for Amazon WorkSpaces Solution provides the following features:

Dry run mode

We recommend that you run this solution in the dry run mode (activated by default) for a few months, review the daily and monthly reports, and manually implement any desired changes. Dry run mode provides insight into how the recommended changes may affect your WorkSpaces costs. This mode also allows you to evaluate and analyze the solution’s recommendations without the solution automatically implementing billing changes. When you are comfortable with the recommendations provided by the solution and would like these recommendations to be automatically implemented, change the template parameter **Dry Run Mode** to No. The solution will then begin to automatically implement future billing changes based on the recommendations provided in its daily and monthly reports.

Automatic billing conversion

When the **Dry Run Mode** parameter is set to No, the solution converts a WorkSpace to monthly or hourly billing model, depending on the WorkSpace utilization for the given month. If WorkSpace utilization exceeds the usage threshold, the billing model changes to monthly on the day the utilization exceeds the threshold. The conversion from monthly to hourly mode happens on the last day of the month. If WorkSpace utilization for the month was lesser than or equal to usage threshold, the billing model changes to hourly on the last day of the month.

Hourly to monthly

This solution calculates the hourly WorkSpace usage for each WorkSpace once per day, right before midnight GMT. This calculation includes that day's usage. When the **Dry Run Mode** parameter is set to No, the solution automatically converts individual WorkSpaces from an hourly billing model to a monthly billing model if the WorkSpace exceeds an hourly usage threshold for that WorkSpace type.

⚠ **Important**

> The solution is designed to calculate the hourly WorkSpace usage once per day. Do not manually trigger the ECS task multiple times per day or modify the EventBridge rule schedule, as this will result in inaccurate billable hours calculations and incorrect WorkSpace billing mode conversions.

By default, the threshold is set close to the hourly and monthly billing break-even point. However, you can use the solution's hub template parameters to change the threshold for when each WorkSpace converts from hourly to monthly billing.

ⓘ **Note**

> Because the solution cannot convert WorkSpaces before the calculation at midnight, the default configuration may result in some WorkSpaces exceeding the threshold for up to 24 hours. For example, the default threshold for a Standard instance is set to 85. If, at midnight on Monday, your usage is 84, the WorkSpace will not be converted to monthly billing. If the usage exceeds 85 after midnight on Monday, the WorkSpace will not be converted until the calculation at midnight on Tuesday.
> If several of your WorkSpaces consistently exceed the threshold before being converted, consider lowering the threshold. If several of your WorkSpaces are prematurely converted before exceeding the threshold, consider raising the threshold.
> To mitigate this potential discrepancy, we recommend carefully monitoring your usage using dry run mode and adjusting the threshold to your needs before setting the **Dry Run Mode** parameter to No.

Once the solution converts a WorkSpace from hourly billing to monthly billing, the solution will not convert the WorkSpace back to hourly billing until the beginning of the next month, if usage was below the monthly usage threshold. However, you can manually change the billing model at any time using the AWS Management Console.

Monthly to hourly

For customers who want to convert WorkSpaces from monthly billing to hourly billing immediately, the solution's CloudFormation template includes a parameter (**Simulate End of Month**) that will run these conversions at the time of deployment.

Unused WorkSpaces termination

⚠ **Important**

> Don't set the parameters **Simulate End of Month Cleanup** and **Terminate workspaces not used for a month** to `Yes` at the same time. Doing this will unexpectedly terminate WorkSpaces.
> The **Simulate End of Month Cleanup** parameter will run the solution as if it's the last day of the month and then terminate WorkSpaces unexpectedly. To use the **Terminate unused WorkSpaces** feature, set **Simulate End of Month Cleanup** to No.

The solution provides a feature to terminate unused WorkSpaces permanently. The solution automatically identifies and excludes standby workspaces from termination to preserve disaster recovery capabilities. Standby workspaces are backup workspaces associated with primary workspaces for disaster recovery purposes and are never terminated regardless of usage patterns. By default, this feature is set to No, and you can allow this by selecting `Yes` or `Dry Run` for the input parameter **Terminate Unused WorkSpaces**.

You can use the input parameter **Number of months for termination check** to set the duration for the unused period. For example, if you select a value of two months, the solution will select the WorkSpaces that were unused for two months for termination.

ⓘ **Note**

> Opting into this feature terminates unused WorkSpaces and can cause changes to your existing CloudFormation stack. Changes to existing CloudFormation resources might cause a drift between the resource state and its definition in the stack. Ensure that these changes do not impact your existing resources and applications before opting into this feature.

For a WorkSpace to be eligible for termination, it needs to meet the following criteria:

- The WorkSpace must be a primary WorkSpace (standby workspaces are automatically excluded)
- The WorkSpace is available from the first day of the unused period that you set.
- The **LastKnownUserConnectionTimestamp**, which indicates the last time a user logged into the WorkSpace, is earlier than the first day of the unused period that you set.
- The **Launch in Dry Run Mode** input parameter is set to No.

If this criteria is met, the WorkSpace will be eligible for termination. Then the solution checks for the following options for the parameter **Terminate Unused Workspaces**:

- **Yes** - You can opt into this feature by selecting `Yes` for the **Terminate Unused Workspaces** input parameter. If opted in, the solution will identify all the WorkSpaces not used during the user-defined period based on the last known user connection timestamp. The solution will only terminate the WorkSpace if it met previously listed criteria
- **Dry Run** - When you select `Dry Run` for the **Terminate Unused Workspaces** input parameter, the solution checks for all the conditions for a WorkSpace to be terminated and update the report, but it won't terminate the WorkSpace. The generated report will be marked as `Yes- Dry Run` for the WorkSpace.

Note

We recommend running this feature in Dry Run mode for the first few months and checking the monthly reports to review which WorkSpaces are marked for deletion. The reports will show WorkspaceType as PRIMARY or STANDBY to help you understand which workspaces are being evaluated.

- **No** - By default, the option is turned off (set to No). The WorkSpace will not be terminated and the report will not have any entry for this WorkSpace.

This check for terminating unused WorkSpaces will run only on the last day of the month or if the customers select `Yes` for the parameter **Simulate End of Month Cleanup.**

Opt out WorkSpaces

To prevent the solution from converting a WorkSpace between billing models or terminating it as an unused WorkSpace, apply a resource tag to the WorkSpace using the tag key `Skip_Convert` and any tag value. This solution will log tagged WorkSpaces, but it will not convert or terminate the tagged WorkSpaces. Remove the tag at any time to resume automatic conversion and termination check for that WorkSpace.

Opt in Regions

This solution provides the **List of AWS Regions** input parameter, which specifies the AWS Regions monitored by the solution. You can provide a comma-separated list of AWS Regions that you want to monitor. If this input parameter is left blank, the solution will default to monitoring WorkSpaces in all the AWS Regions for the account.

Deployment in an existing Amazon VPC

This solution creates a new Amazon Virtual Private Cloud (Amazon VPC) to run the Amazon ECS task. If desired, you can deploy the solution in an existing Amazon VPC by providing the subnet IDs and security group ID as part of the input parameters to the CloudFormation template. To run the solution in an existing Amazon VPC, the Amazon ECS task needs to run in a public subnet, or a private subnet with a route to the Internet. This route is necessary because the Amazon ECS task will pull the Docker image hosted in a public Amazon Elastic Container Registry (Amazon ECR) repository. The security group used to run the ECS task will allow you to pull this image from ECR repository. To deploy the solution in an existing VPC select `No`for the input parameter **Create New VPC**. Provide the details for you existing VPC in the input section **Existing VPC settings**. Do not change the default values for the section **New VPC Settings**.

> **Note**
>
> In order for the solution to be deployed to a PUBLIC SUBNET, the subnet itself has to have auto-assign public IPs enabled.

Maintenance calculation

If you run your patches or other maintenance scripts by changing the WorkSpaces to `ALWAYS_ON` mode, you will need to account for the `ALWAYS_ON` hours and adjust the threshold values in the

CloudFormation template accordingly. For example, if you change the mode from `AUTO_STOP` to `ALWAYS_ON` for 10 hours to run the maintenance scripts, you should subtract 10 hours from the set threshold value in the CloudFormation template.

The solution automatically accounts for default AWS maintenance activities on AUTO_STOP workspaces when maintenance mode is enabled in the directory. For directories with maintenance mode enabled, the solution adds one hour of usage to AUTO_STOP workspaces at the end of each month to account for maintenance windows.

Use cases

Device management

As workforces increasingly become virtual, companies must grapple with device security, warranties, and delivery. By establishing thorough device management and policies, companies can spend fewer resources tracking devices or handling security breaches. Device management solutions help you secure your workforce's devices and provide options to meet different cybersecurity assurance levels. These solutions offer cost savings by reducing the number of devices needed, making management more efficient, extending the life of existing devices, and facilitating a bring-your-own-device policy.

Cloud financial management

Establishing visibility and usage analysis through spend dashboards, spend limits, chargebacks, anomaly detection and response, allows customers to identify opportunities to optimize expenses for cloud services. Customers can allocate current costs, plan and forecast future spend with the cloud financial management capability. This enables customers to track, notify, and apply cost optimization techniques across their environment.

Virtual desktop delivery

Whether in a workplace or classroom setting, desktop users need to be able to access their desktops and associated applications without delay. However, IT resource constraints and quality controls often slow down desktop delivery. Virtual desktop delivery solutions allow for faster and easier desktop-based application delivery. With these solutions, you can secure applications, build resiliency, and quickly scale to meet user demands—resulting in quantifiable savings and better data governance.

Architecture overview

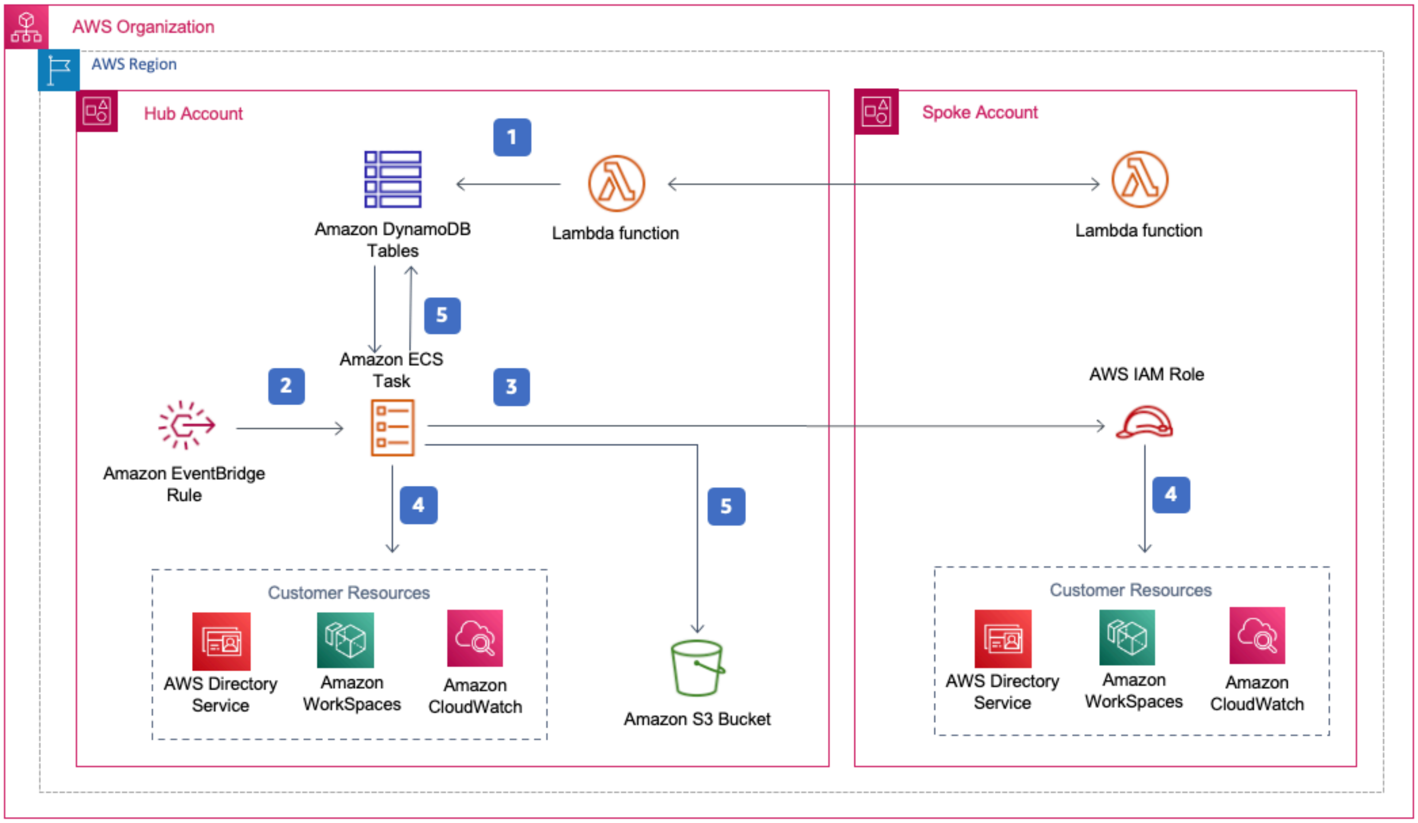
This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.

> **Note**
>
> This solution includes both a *hub account template *(deployed first) for a central account to manage the WorkSpaces and provide a centralized report, and a *spoke account template *(deployed second) for each WorkSpace account that you want to monitor. The solution generates a report per directory and an aggregated report with information about WorkSpaces from all the directories combined.

Cost Optimizer for Amazon WorkSpaces architecture



1. The spoke template creates a custom resource that invokes an AWS Lambda function to register the account as a spoke account in an Amazon DynamoDB table in the hub account.
2. The hub template creates an Amazon EventBridge rule that invokes an Amazon ECS task every 24 hours.
3. The Amazon ECS task assumes an AWS Identity and Access Management (IAM) role in each spoke account to manage WorkSpaces.
4. The Amazon ECS task polls AWS Directory Service to gather a list of all directories registered for Amazon WorkSpaces in a specific AWS Region. The task then checks the total usage for each WorkSpace that is on an hourly billing model. If a WorkSpace has met the monthly usage threshold, the solution will convert the individual WorkSpace to monthly billing.

> **Note**
>
> If a WorkSpace starts in monthly billing or the solution converts a WorkSpace from hourly to monthly billing, the solution will not convert the WorkSpace to hourly billing until the beginning of the next month if usage was below the threshold. However, you can manually change the billing model at any time using the Amazon WorkSpaces

console. Also, you can change the threshold for when each WorkSpace converts from hourly to monthly billing. For more information, refer to Automatic billing conversion

The solution also features a dry run mode (activated by default) that allows you to gain insight into how the recommended changes will affect your costs. For more information, refer to Dry run mode.

+

At the end of the month, the Amazon ECS task checks the total usage for each Workspace that is on a monthly billing model. If a WorkSpace has not met the monthly usage threshold, the solution will convert the individual WorkSpace from monthly to hourly billing at the start of the next month. . The Amazon ECS task writes the results of the execution to the DynamoDB usage table, session tables, and uploads them to an Amazon Simple Cloud Storage (Amazon S3) bucket.

Note

Check your Amazon S3 bucket frequently to track the optimizer's activity, and to view logs with error messages.

AWS Well-Architected design framework

We designed this solution with best practices from the AWS Well-Architected Framework, which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how we applied the design principles and best practices of the Well-Architected Framework when building this solution.

Operational excellence

This section describes how we applied the principles and best practices of the operational excellence pillar when designing this solution.

- The solution pushes metrics to Amazon CloudWatch to provide observability into the infrastructure, AWS Lambda functions, Amazon S3 buckets, and the rest of the solution components.

- The solution updates the daily report to indicate if the WorkSpace was skipped due to an application programming interface (API) failure.
- The solution provides a way to incrementally onboard spoke accounts when new WorkSpace workloads are added to the account.

Security

This section describes how we applied the principles and best practices of the security pillar when designing this solution.

- All inter-service communications use IAM roles.
- All multi-account communications use IAM roles.
- All roles used by the solution follow least-privilege access. In other words, they only contain minimum permissions required so that the service can function properly.
- All data storage, including Amazon S3 buckets and DynamoDB tables, have encryption at rest.

Reliability

This section describes how we applied the principles and best practices of the reliability pillar when designing this solution.

- The solution uses serverless AWS services wherever possible (such as Lambda, Amazon S3, and AWS Fargate) to ensure high availability and recovery from service failure.
- Data processing uses Lambda functions. The solution stores data in DynamoDB and Amazon S3, so it persists in multiple Availability Zones by default.

Performance efficiency

This section describes how we applied the principles and best practices of the performance efficiency pillar when designing this solution.

- All inter-service communications use IAM roles.
- The solution uses serverless AWS services wherever possible (such as Lambda, Amazon S3, and Fargate).
- The solution provides the ability to launch in any AWS Region that supports the AWS services used in this solution, such as AWS Lambda and Amazon S3.

- Automatically tested and deployed daily. This solution is reviewed by solution architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how we applied the principles and best practices of the cost optimization pillar when designing this solution.

- The solution uses serverless architecture, and customers pay only for what they use.
- The solution uses a lifecycle policy for the Amazon S3 bucket to delete objects after a year to help reduce the storage cost.
- The solution provides a feature to terminate unused WorkSpaces to help you save costs by reducing the operating workloads.

Sustainability

This section describes how we applied the principles and best practices of the sustainability pillar when designing this solution.

- The solution uses managed and serverless services to minimize the environmental impact of the backend services.
- The solution's serverless design is aimed at reducing the carbon footprint compared to the footprint of continually operating on-premises servers.
- Solution provides a feature to terminate unused WorkSpaces to help minimize the carbon footprint further.

AWS Organizations support

The solution supports AWS Organizations through a hub-and-spoke architecture. To monitor WorkSpaces across multiple accounts in your organization, allow trusted access forAWS Resource Access Manager (AWS RAM) in the management account of your Organization. For more information on how to allow trusted access for RAM, refer to AWS Resource Access Manager and AWS Organizations.

You can deploy the hub template in the central account, and then deploy the spoke template in each account that manages WorkSpaces. The spoke stacks must be deployed in the same Region as the hub stack.

For a multi-account deployment, provide the value for the **Organization ID for multi account deployment** and **Account ID of the Management Account for the Organization** input parameters. For a single-account deployment, or to manage WorkSpaces only in the central account, deploy only the hub template and leave the default value for the input parameters **Organization ID for multi account deployment** and **Account ID of the Management Account for the Organization**.

AWS services in this solution

The following AWS services are included in this solution:

AWS service	Description
Amazon WorkSpaces	The solution does not create any resources for the Amazon WorkSpaces service but monitors the existing WorkSpaces in the customer accounts.
Amazon Simple Storage Service	The solution creates an Amazon S3 bucket to store the daily and the monthly reports for the processed WorkSpaces.
AWS Organizations	The solution monitors WorkSpaces for accounts which are part of the given organizat ion.
AWS Lambda	The solution creates Lambda functions to register spoke accounts with the hub account.
Amazon Elastic Container Service	The solution creates an Amazon ECS task which is used to monitor WorkSpaces in hub and spoke accounts.
AWS Directory Service	The solution does not create any resources for the AWS Directory Service but monitors the WorkSpaces in the existing Directories.

AWS service	Description
Amazon CloudWatch	The solution creates log groups to store logs for the Amazon ECS task and Lambda functions.
AWS Fargate	The solution uses Fargate launch type to run Amazon ECS tasks.
AWS EventBridge	The solution creates Amazon EventBridge rules to trigger an Amazon ECS task on a scheduled basis.
AWS IAM	The solution creates IAM roles needed to access WorkSpaces in the hub and spoke accounts.
Amazon DynamoDB	The solution creates a DyanamoDB table to store spoke account details.
AWS Service Catalog	The solution creates a service catalog applicati on and associates the CloudFormation stack to this application.
AWS CloudFormation	The solution uses CloudFormation templates to deploy the resource needed to monitor WorkSpaces.

Plan your deployment

This section describes the cost, security, and quota considerations prior to deploying the solution.

Supported AWS Regions

You must launch the solution in an AWS Region that supports AWS Lambda, Amazon WorkSpaces, and AWS Fargate services. However, once deployed, the solution will monitor the WorkSpaces in any AWS Region. The solution can also monitor the WorkSpaces in the AWS GovCloud (US) Regions. After you deploy the solution in an AWS GovCloud (US) Region, the Amazon ECS task will pull the Docker image hosted in the AWS public Amazon ECR repository and monitor the workspaces in the AWS GovCloud (US) Region.

For the most current availability by Region, refer to the AWS Regional Services List.

Cost Optimizer for Amazon Workspaces is supported in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Seoul)
US East (N. Virginia)	Europe (Paris)
US West (Northern California)	Middle East (Bahrain)
US West (Oregon)	AWS GovCloud (US-West)
Africa (Cape Town)	Europe (Ireland)
Europe (London)	Europe (Stockholm)
Canada (Central)	Europe (Frankfurt)
Asia Pacific (Mumbai)	Asia Pacific (Osaka)
Asia Pacific (Singapore)	Asia Pacific (Sydney)
Asia Pacific (Tokyo)	South America (Sao Paulo)

Cost

You are responsible for the cost of the AWS services used while running this solution. The total cost of running this solution depends on the number of WorkSpaces that this solution monitors. As of the latest revision, the cost for running this solution with default settings in the US East (N. Virginia) Region is approximately **$42.00 per month** for 1,000 WorkSpaces. This reflects Amazon VPC, Amazon CloudWatch, Amazon DynamoDB, Amazon ECS (Fargate), Amazon S3, and AWS Lambda charges and does not include the cost of the individual WorkSpaces (refer to Amazon WorkSpaces Pricing).

The majority of this cost ($33.08/month) comes from the NAT Gateway. To reduce costs, you can deploy the solution in your existing Amazon VPC by providing your subnet IDs and security group ID during deployment.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, see the pricing webpage for each AWS service used in this solution.

CUDOS Dashboard

While this solution helps you optimize WorkSpaces costs, to see the actual cost savings, you can also deploy a CUDOS dashboard by following the steps in our CUDOS Dashboard lab and see the metrics for the cost saving. After deploying the CUDOS dashboard, you can navigate to the **End User Computing** tab to see the graphs for **Average Workspaces Cost per region** and **Workspaces spend per region**.

Sample cost table: Scenario 1

The following table provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

The cost for **Scenario 1** is based on the following assumptions:

- Number of WorkSpaces: 1,000
- Number of Amazon ECS Task runs per month: 30
- Amazon ECS task run time in seconds: 600 (10 minutes)

AWS service	Dimensions	Scenario 1 total cost (per month) [USD]
Amazon VPC	NAT Gateway hourly and data processing charges (only if deploying a new VPC)	$33.08
Amazon CloudWatch	Logs, metrics, dashboard, and API calls	$7.77
Amazon DynamoDB	Storage and request units for UsageTable, UserSessionTable, and SpokeAccountTable	$0.83
Amazon ECS (Fargate)	ECS task (0.25 vCPU, 1 GB memory)	$0.07
Amazon S3	Cost Optimizer Report bucket	$0.01
AWS Lambda	Account registration and UUID generation (within free tier)	$0.00
Total		**$41.76**

The cost for **Scenario 2** is based on the following assumptions:

- Number of WorkSpaces: 5,000
- Number of Amazon ECS Task runs per month: 30
- Amazon ECS task run time in seconds: 1,200 (20 minutes)

AWS service	Dimensions	Scenario 2 total cost (per month) [USD]
Amazon VPC	NAT Gateway hourly and data processing charges (only if deploying a new VPC)	$33.08
Amazon CloudWatch	Logs, metrics, dashboard, and API calls	$17.21
Amazon DynamoDB	Storage and request units for UsageTable, UserSessionTable, and SpokeAccountTable	$4.17
Amazon ECS (Fargate)	ECS task (0.25 vCPU, 1 GB memory)	$0.15
Amazon S3	Cost Optimizer Report bucket	$0.01
AWS Lambda	Account registration and UUID generation (within free tier)	$0.00
Total		**$54.62**

Note

The default values provided in Cost Optimizer for Amazon WorkSpaces for each of the WorkSpaces bundle type should be considered a guidance value only. Review Amazon WorkSpaces pricing for the AWS Region where you deployed the solution to determine the value for your location. Pricing differences between AWS Regions can mean a different billing break-even point between AutoStop and AlwaysOn instances.

Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

IAM roles

This solution creates IAM roles to control and isolate permissions, following the best practice of least privilege. The solution grants services the following permissions:

Hub template

`RegisterSpokeAccountsFunctionLambdaRole`

- Write permission to Amazon DynamoDB table where spoke accounts are registered

`InvokeECSTaskRole`

- Permission to create and run Amazon ECS tasks

`CostOptimizerAdminRole`

- Read permissions to an Amazon DynamoDB table where spoke accounts are registered
- Assume role permissions to `WorkspacesManagementRole` in spoke accounts
- Read only permissions to AWS Directory Service
- Write permissions to Amazon CloudWatch Logs
- Write permissions to Amazon S3
- Read and write permissions to WorkSpaces

`SolutionHelperRole`

- Permisison to invoke an AWS Lambda function to generate a universally unique identifier (UUID) for solution metrics

Spoke template

`WorkSpacesManagementRole`

- Read only permissions to AWS Directory Service
- Write permissions to Amazon CloudWatch Logs
- Write permissions to Amazon S3
- Read/write permissions to WorkSpaces

`AccountRegistrationProviderRole`

- Invoke Lambda function to register spoke account with hub account stack

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, see AWS service quotas.

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has AWS CloudFormation quotas that you should be aware of when launching the stack hub template in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the in the *AWS CloudFormation User’s Guide*

Deploy the solution

Important

Don’t set the parameters **Simulate End of Month Cleanup** and **Terminate workspaces not used for a month** to Yes at the same time. Doing this will unexpectedly terminate WorkSpaces.
The **Simulate End of Month Cleanup** parameter will run the solution as if it’s the last day of the month and then terminate WorkSpaces unexpectedly. To use the **Terminate unused WorkSpaces** feature, set **Simulate End of Month Cleanup** to No.

This solution uses AWS CloudFormation to automate the deployment of Cost Optimizer for Amazon WorkSpaces on the AWS Cloud. It includes the following AWS CloudFormation templates, which you can download before deployment.

This solution uses CloudFormation templates and stacks to automate its deployment. The CloudFormation templates describe the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the templates.

Deployment process overview

Before you launch the automated deployment, please review the cost, architecture, network security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Note

Deploy the hub template first. If you have a single-account environment, you do not need to deploy the spoke template. If you have a multi-account environment, install the spoke template for each WorkSpaces account you wish to monitor.

Time to deploy: Approximately five minutes

Step 1: Launch the hub stack

- Launch the AWS CloudFormation template in your hub account
- Enter values for the required parameters
- Review the other template parameters and adjust, if necessary

Step 2: Launch the spoke stack

- Launch the CloudFormation template in your spoke account
- Enter values for the required parameters
- Review the other template parameters and adjust, if necessary

Note

This solution includes data collection. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered through this survey. Data collection is subject to the AWS Privacy Policy.

AWS CloudFormation templates

You can download the CloudFormation templates for this solution before deploying it.

Hub account

View template

cost-optimizer-for-amazon-workspaces.template - Use this template to launch Cost Optimizer for Amazon WorkSpaces and all associated components for your hub account (or single account if you only have one account). The default configuration deploys AWS Lambda functions, an Amazon ECS task definition, an Amazon CloudWatch event, and an Amazon S3 bucket. You can customize the template based on your specific needs.

Spoke accounts

View template

cost-optimizer-for-amazon-workspaces-spoke.template – Use this template to launch Cost Optimizer for Amazon WorkSpaces and all associated components for your spoke accounts. Do not deploy this template if you only have one account. The default configuration deploys AWS Lambda functions, an Amazon ECS task definition, an Amazon CloudWatch event, and an Amazon S3 bucket. You can customize the template based on your specific needs.

> **Note**
>
> If you have previously deployed this solution, see Update the solution for update instructions.

Step 1: Launch the hub stack

This automated AWS CloudFormation template deploys Cost Optimizer for Amazon WorkSpaces on the AWS Cloud.

1. Sign in to the AWS Management Console and select the button to launch the `cost-optimizer-for-amazon-workspaces` AWS CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch Cost Optimizer for Amazon WorkSpaces in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This solution can be launched as a single account deployment or multi account deployment. For single account deployment, you only need to deploy the hub template in the account you want to monitor the workspaces for. The multi account deployment works in the hub and spoke model.For multi account deployment, you need to deploy the hub template in a central account and provide AWS Organization ID and Management Account ID in the input parameters for the hub account.

3. On the **Create Stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the *AWS Identity and Access Management User Guide*.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

> ⚠ **Important**
>
> Don't set the parameters **Simulate End of Month Cleanup** and **Terminate workspaces not used for a month** to Yes at the same time. Doing this will lead to unexpected WorkSpace termination

The **Simulate End of Month Cleanup** parameter will run the solution as if it's the last day of the month and then terminate WorkSpaces unexpectedly. To use the Terminate unused WorkSpaces feature, set **Simulate End of Month Cleanup** to No.

Parameter	Default	Description
Select New or Existing VPC for AWS Fargate		
Create New VPC	Yes	Select Yes to deploy the solution in a new Amazon VPC.
Existing VPC Settings		
Public Subnet ID	*<Optional input>*	A public subnet ID to launch the gateway. Leave this blank if you selected Yes for **Create New VPC** or enter an existing subnet ID to run an Amazon ECS task.

Parameter	Default	Description
First Public Subnet ID	*<Optional input>*	A Private Subnet ID to launch ECS task. Leave this blank if you `Yes` for **Create New VPC** .
Second private subnet ID for second subnet	*<Optional input>*	A second private subnet ID to launch an Amazon ECS task. Leave this blank if you selected `Yes` for **Create New VPC** or enter an existing subnet ID to run an Amazon ECS task.
Security group ID to launch ECS task	*<Optional input>*	Security group ID to launch an Amazon ECS task Leave this blank if you selected `Yes` for **Create New VPC** or enter an existing security group ID to run an Amazon ECS task.
New VPC Settings		
AWS Fargate VPC CIDR Block	`10.215.0.0/16`	The default VPC CIDR used to deploy the AWS Fargate container dynamically when the solution runs.
Public Subnet CIDR Block	`10.215.10.0/24`	A public subnet to which the internet gateway is deployed.
AWS Fargate Private Subnet1 CIDR Block	`10.215.30.0/24`	One of two private subnets in different Availability Zones where the AWS Fargate container is deployed.

Parameter	Default	Description
AWS Fargate Private Subnet2 CIDR Block	`10.215.40.0/24`	The second of two subnets in different Availability Zones where the AWS Fargate container is deployed.
AWS Fargate SecurityGroup CIDR Block	`0.0.0.0/0`	The CIDR block to restrict the Amazon ECS container outbound access.
Testing Parameters		
Launch in Dry Run Mode	`Yes`	Generates a change log, but does not inact any changes. For more information, refer to Dry Run Mode.
Simulate End of Month Cleanup	`No`	Overrides date and forces the solution to run as if it is the end of the month.
Log Level	`INFO`	Sets the log level for the Lambda function logs in CloudWatch.
Pricing Parameters		
ValueLimit	`81`	The number of hours a Value instance can run in a month before being converted to `ALWAYS_ON` .
StandardLimit	`85`	The number of hours a Standard instance can run in a month before being converted to `ALWAYS_ON` .

Parameter	Default	Description
PerformanceLimit	83	The number of hours a Performance instance can run in a month before being converted to ALWAYS_ON .
GraphicsG4dnLimit	217	The number of hours a Graphics instance can run in a month before being converted to ALWAYS_ON .
GraphicsProG4dnLimit	80	The number of hours a GraphicsPro instance can run in a month before being converted to ALWAYS_ON .
PowerLimit	83	The number of hours a Power instance can run in a month before being converted to ALWAYS_ON .
PowerProLimit	80	The number of hours a PowerPro instance can run in a month before being converted to ALWAYS_ON .
Container Image		
Auto-Update Container Image	Yes	Automatically use the most up to date and secure image up until the next minor release. Selecting No will pull the image as originally released, without any security updates.
List of AWS Regions		

Parameter	Default	Description
List of AWS Regions	`<Requires input>`	Comma separated list of AWS Regions for the solution to monitor. Example: `us-east-1` , `us-west-2` .
Terminate unused workspaces		
Terminate workspaces not used for a month	No	Select `Yes` to delete WorkSpaces not used for one month.
Number of months for termination check	1	Provide the number of months to check for inactive period before termination. Default value is 1 month.
Multi account deployment		
Organization ID for multi account deployment	*<Optional input>*	AWS Organizations ID to support multi-account deployment. Leave blank for single account deployments.
Account ID of the Management Account for the Organization	*<Optional input>*	Account ID for the Organizat ion's management account. Leave blank for single account deployments.

6. Choose **Next.**

7. On the **Configure stack options** page, choose **Next**.

8. On the **Review and create** page, review and confirm the settings. Select the box acknowledging that the template will create IAM resources.
9. Choose **Submit** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of CREATE_COMPLETE in approximately five minutes.

> **Note**
>
> In addition to the `CostOptimizerCreateTask` AWS Lambda function, this solution includes the `solution-helper` Lambda function, which runs only during initial configuration or when resources are updated or deleted.
> When running this solution, you will see all Lambda functions in the AWS Management Console, but only the `CostOptimizerCreateTask` function is regularly active. However, do not delete the `solution-helper` function as it is necessary to manage associated resources.

Step 2: Launch the spoke stack

This automated CloudFormation template deploys the spoke for Cost Optimizer for Amazon WorkSpaces on the AWS Cloud.

1. Sign in to the AWS Management Console and select the button to launch the `cost-optimizer-for-amazon-workspaces-spoke` AWS CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch Cost Optimizer for Amazon WorkSpaces in a different AWS Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > The spoke account template should be deployed in any account you want to monitor with from the hub account deployment. Once deployed, the spoke template will register this account with the hub account deployment. The ECS task in the hub account will now

monitor the WorkSpaces in the spoke account. You do not need to deploy the spoke account in the hub account to monitor the WorkSpaces in the hub account.

3. On the **Create Stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see IAM and AWS STS quotas, name requirements, and character limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Hub account ID	`<Requires input>`	The ID of the hub account for the solution. This stack should be deployed in the same Region as the hub stack in the hub account.
Logging level	`INFO`	Logging level.

6. Choose **Next.**
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the box acknowledging that the template will create IAM resources.
9. Choose **Submit** to deploy the stack.

You can can view the status of the stack in the CloudFormation console in the Status column. You should receive a `CREATE_COMPLETE` status in approximately five minutes

Note

In addition to the **CostOptimizerCreateTask** AWS Lambda function, this solution includes the solution-helper Lambda function, which runs only during initial configuration or when resources are updated or deleted.

When running this solution, you will see all Lambda functions in the AWS Management Console, but only the `CostOptimizerCreateTask` function is regularly active. However, don’t delete the `solution-helper` function because it’s necessary to manage associated resources.

Monitor the solution

This solution uses Operation Insights Dashboard to allow you to monitor your Cost Optimizer for Amazon Workspaces solution.

Operation Insights Dashboard

Cost Optimizer for Amazon Workspaces comes with an Operational Insights dashboard that allows you to monitor the operation of the solution and get insight into the running hours that have been saved by using this solution.

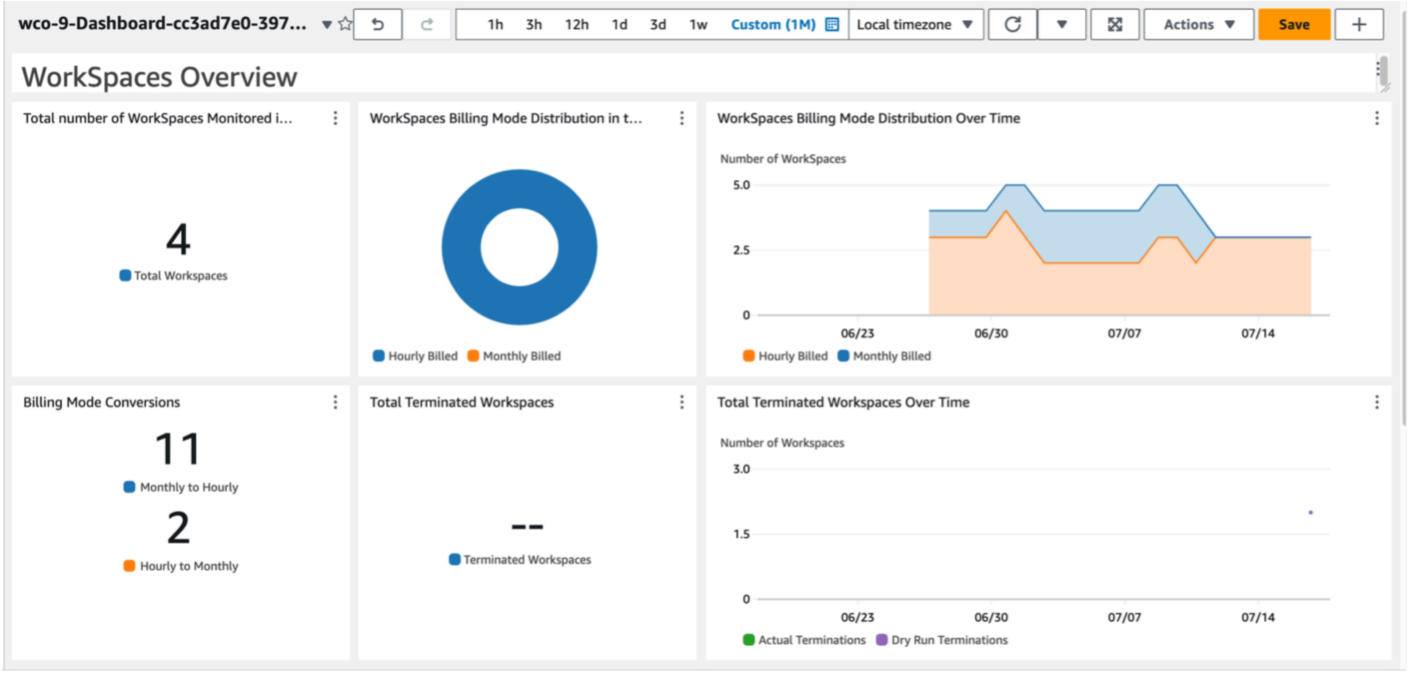
To access this dashboard:

1. Go to the AWS CloudWatch console.
2. Select **Dashboards** from the navigation menu.
3. Find and select the dashboard named `{stack-name}-Dashboard`.

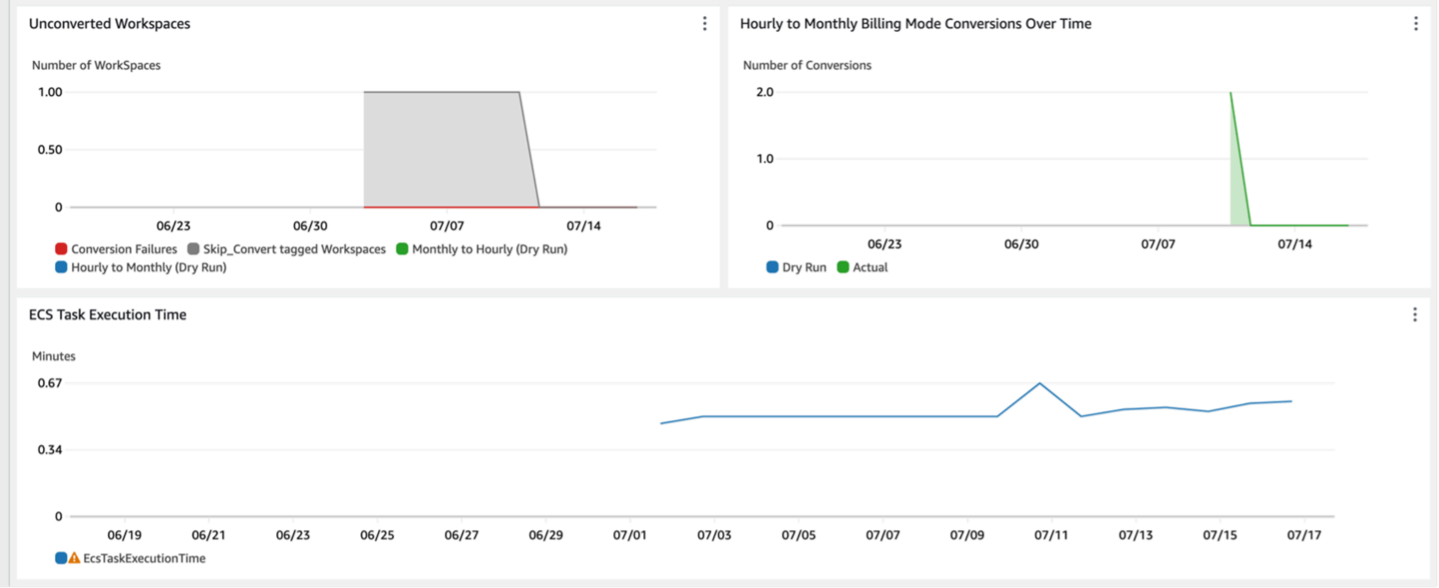
The dashboard will display various operational metrics about the operations of your solution including counts of how many Workspaces are analyzed by the solution, information on changes taken, and insights about the container that is performing the analysis.

Sample data below:

Cost Optimizer for Amazon WorkSpaces overview



Cost Optimizer for Amazon WorkSpaces insights



Additional costs associated with this feature

Service	Cost per month
Custom CloudWatch Dashboard	$3.00
Amazon ECS	$3.30

Service	Cost per month
Total	**$6.30 / month**

Update the solution

If you have previously deployed the solution, follow this procedure to update the Cost Optimizer for Amazon WorkSpaces on AWS CloudFormation stack to get the latest version of the solution's framework.

1. Log in to AWS CloudFormation console, select your existing `workspaces-cost-optimizer `CloudFormation stack, and select Update. stack, and choose **Update**.
2. Select **Replace current template**.
3. Under **Specify template:**
   - Select Amazon S3 URL
   - Copy the link of the `cost-optimizer-for-amazon-workspaces.template` AWS CloudFormation template.
   - Paste the link in the **Amazon S3 URL** box.
   - Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. Choose **Next** again.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. Refer to Step 1: Launch the stack for details about the parameters.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template might create (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Update stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a `UPDATE COMPLETE` status in approximately 15 minutes.

Troubleshooting

This section provides troubleshooting instructions for deploying and using the solution.

Known issue resolution provides instructions to mitigate known errors. If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Known issue resolution

Failed WorkSpaces

The solution will attempt to modify WorkSpace mode based on the threshold hours. When the solution cannot change the WorkSpace mode due to an exception, such as the workspace being in **UNHEALTHY** state or booting up during the conversion, the solution will skip this WorkSpace and continue to process other workspaces in the account. The skipped workspaces will be added to the daily report with the message `Failed to change mode`. You can find more details about the exception in the Amazon ECS logs. The solution will attempt to change the mode again during the next run of the Amazon ECS task.

The solution creates log group and log streams to store logs each run of the Amazon ECS task. To get detailed information on the steps that happened during the solution run, enter the Debug option for the parameter **Log Level**.

If the you think that the solution did not work as expected or if the report was generated on a certain day, you can find more information about the solution run in the logs for the Amazon ECS task. The easiest way to access the logs is to filter the resources section of the CloudFormation stack by **CostOptimizerLogs** and select the link for the logs. This will take you to the log group in the CloudWatch service console. From here, you can open the log stream for the day you want to analyze the logs for errors.

If you need more information about a particular WorkSpace, search the logs with the **WorkSpace ID** of your WorkSpace to see more information filtered down to your WorkSpace.

FAQs

Q: Where is the report that was generated for the WorkSpaces?

A: The solution uploads the report to an Amazon S3 bucket created by the solution. To access the bucket, filter the Resources section of the CloudFormation stack with CostOptimizerBucket. Navigate to the bucket by selecting the link. The reports are uploaded daily.

Q: How to find the Amazon ECS logs?

A: Filter the **Resources** section of the CloudFormation stack with **CostOptimizerLogs**

Q: How do I comfirm that a WorkSpace was terminated?

A: The report generated by the solution will have `Yes` in the **Terminated Workspaces** column.

Redeploying after deletion

If you delete the hub stack and wish to redeploy it, you must also redeploy all of the spoke stacks; redeploying the hub stack will not automatically work with the existing old spoke templates.

Contact AWS Support

If you have AWS Business Support+, AWS Enterprise Support, or Unified Operations, you can use the AWS Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case.**

How can we help?

1. Choose **Technical**
2. For **Service**, select **Solutions**.
3. For **Category**, select **Cost Optimizer for Amazon WorkSpaces**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your questions with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail, including the name of this product and the version you are using, such as this example: Cost Optimizer for Amazon WorkSpaces vX.Y.Z.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can’t resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall Cost Optimizer for Amazon WorkSpaces from the AWS Management Console or by using the AWS Command Line Interface (AWS CLI). Delete all the spoke account stacks and then delete the hub account stack.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console for spoke accounts.
2. Select this solution's spoke account main stack.
3. Choose **Delete**.
4. Sign in to the CloudFormation console for the hub account.
5. Select this solution's hub account stack.
6. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Developer guide

This section provides the source code for the solution.

Source code

You can visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Container image

This section provides information about the container image used by the solution.

Maintenance

This solution uses Docker images with fixed versions that match each solution release if automatic updates is not selected. The AWS Solutions team uses ECR Enhanced Scanning to detect Common Vulnerabilities and Exposures (CVEs) in the base image and installed packages. When possible, the team will publish patched images with the same version tag to resolve CVEs, without breaking compatibility with the released solution version. When images are patched, if they are on the same minor version, the stable tag will be automatically updated, and an additional image tag will be created in the format `<solution-version>_<date-of-fix>`. If a major or minor version is released, a full stack update will be required to get the latest image version as the stable tag will be incremented so that its version matches the version of the solution. If opting-in to automatic updates the changes to the image, including the CVEs and minor bug fixes, will automatically be applied to the image up to the latest matching minor release.

Versions

Customers on the latest solution version will receive security patches and minor, non-breaking, bug fixes automatically if they opt-in to automatic image updates. The image will automatically pull the latest image up to the latest matching minor version. In order to lock the container to a specific version, the task definition can be edited to specify the container to use a specific image version by using the tagged version of the image. Automatic updates can also be turned off by selecting **No** to automatic updates in CloudFormation when launching the stack. This will launch the image version matching the solution version.

Reference

This section includes information about data collection and a list of builders who contributed to this solution.

Data collection

This solution sends operational metrics to AWS (the "Data") about the use of this solution. We use this Data to better understand how customers use this solution and related services and products. AWS’s collection of this Data is subject to the AWS Privacy Notice.

Contributors

- Chaitanya Deolankar
- Max Granat
- Matt Juaire
- Michael Ortwein
- Joseph Seeley
- William Quan
- Sanjay Reddy Kandi
- George Lenz

Revisions

Publication date: *January 2017*

Visit the CHANGELOG.md in our GitHub repository to track version-specific improvements and fixes.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.