Technology Partner Integration Guide

Amazon WorkSpaces Core

Amazon WorkSpaces Core: Technology Partner Integration Guide

Table of Contents

Introduction

How Amazon WorkSpaces Core works

Amazon WorkSpaces Core provides managed virtual desktop infrastructure (VDI) that integrates with third-party management solutions. Amazon WorkSpaces Core enables technology partners to build flexible and customizable desktop solutions while benefiting from the security, global availability, and cost efficiency of AWS infrastructure.

Amazon WorkSpaces Core supports two provisioning options:

- **Amazon WorkSpaces Core bundle configurations** – Similar to an Amazon WorkSpaces Personal bundle, an Amazon WorkSpaces Core bundle is a preconfigured combination of compute, storage, and software resources, along with an operating system that you can use to launch a virtual desktop in Amazon WorkSpaces Core. Use this option to deliver persistent desktops where user data and settings are preserved. Amazon WorkSpaces Core manages the underlying infrastructure on your behalf, including Amazon Machine Image (AMI), Amazon EC2 instances, and Amazon EBS volumes. Pricing is all-inclusive and available on hourly or monthly billing terms. For details about available public bundles, see Amazon WorkSpaces Bundles.
- **Amazon WorkSpaces Core Managed Instances** – An Amazon WorkSpaces Core Managed Instance is an Amazon EC2 instance that is launched and managed by Amazon WorkSpaces Core. Use this option if you prefer to have more visibility and control of the types of Amazon EC2 instances managed by WorkSpaces Core. You can deliver both persistent and non-persistent desktops while maintaining direct control over your Amazon EC2 instances. This option allows you to leverage Amazon EC2 pricing models, including Reserved Instances and Savings Plans, for cost optimization. You are billed for an hourly service fee in addition to charges for any AWS resources used (such as Amazon EC2 and Amazon EBS). For details about support instances, see Amazon WorkSpaces Core Managed Instances .

Who should use Amazon WorkSpaces Core

This guide is intended for third-party VDI solution providers who want to build custom desktop experiences on AWS using Amazon WorkSpaces Core. Providers can use the Amazon WorkSpaces Core API to integrate desktop provisioning, management, and monitoring capabilities into their solutions.

If you're a customer interested in using a VDI or desktop as a service (DaaS) solution built on Amazon WorkSpaces Core, see Amazon WorkSpaces Core and choose **WorkSpaces Core Partners** to learn more.

Amazon WorkSpaces Core is part of the Amazon WorkSpaces Family. For more information, see Amazon WorkSpaces Family.

Shared responsibility model

Security and compliance is a shared responsibility between AWS and its partners. This shared model can help relieve your operational burden. AWS operates, manages and controls the components from the host operating system and visualization layer to the physical security of the facilities in which the service operates. The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software, and the configuration of the security group firewall that's provided by AWS.

Customers should carefully consider the services that they choose. Their responsibilities vary depending on the services used, the integration of those services into their IT environment, and applicable laws and regulations. The nature of this shared responsibility also provides the flexibility and customer control that permits the deployment. For more information, see Shared Responsibility Model.

Topics

Shared responsibilities with Amazon WorkSpaces Core

The following responsibilities are shared between your company and Amazon WorkSpaces Core:

- Compliance validation.
- Amazon WorkSpaces image management for Amazon WorkSpaces Core bundles. However, customers are responsible for image managed for Amazon WorkSpaces Core Managed Instances.
- AWS Identity and Access Management (IAM) for WorkSpaces. This responsibility includes IAM configurations and policies. This responsibility doesn't include access to the desktop through the customer and/or partner directory, or gateway services.

Amazon WorkSpaces Core responsibilities

The following responsibilities belong to Amazon WorkSpaces Core:

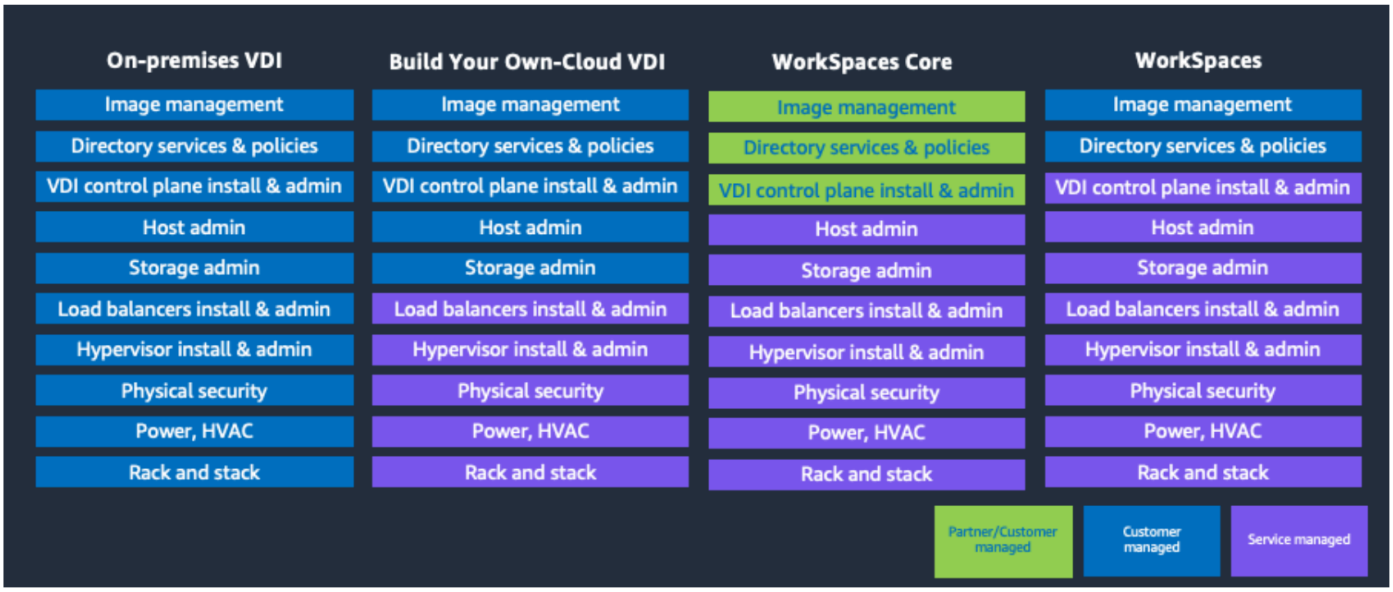
- Infrastructure security.
- Encryption at rest (which must be enabled) for Amazon WorkSpaces Core bundles. For more information, see Encrypted WorkSpaces in the *Amazon WorkSpaces Administration Guide*.
- Resilience in Amazon WorkSpaces Core bundles (except for cross-Region redirection).
- WorkSpaces API operations, AWS Command Line Interface (AWS CLI), SDK, CDK, and console.
- WorkSpaces based monitoring.
- WorkSpaces dedicated hardware requirements.
- Windows operating system (OS) updates and security patches for WorkSpaces Core bundles.

Customer and partner responsibilities

The following responsibilities belong to your company:

- Lifecycle of the Amazon WorkSpaces Core desktop, including calling our API, CLI, or console to provision the desktop, receiving any status, and calling our API, CLI, or console to terminate the desktop.
- Registration of Amazon WorkSpaces Core desktops within the customer or partner solution.
- Brokering Active Directory users to the Amazon WorkSpaces Core desktop.
- Gateway services for securely accessing the Amazon WorkSpaces Core desktop.
- Multi-Region resilience.
- Customers are responsible for Windows OS updates and security patches for WorkSpaces Core Managed Instances.
- Customers must provision and attach encrypted Amazon EBS volumes for Amazon WorkSpaces Core managed instances. For more information, refer to Encryption at Rest for EBS Storage. For more information, see Data Protection in Amazon EC2.
- Additional monitoring, security, and analytic solutions. These solutions are also the responsibility of the customer or partner operating the solution.

The following images show the shared responsibility model and shared responsibility with AWS and your partner.


On-premises VDI
Image management
Directory services & policies
VDI control plane install & admin
Host admin
Storage admin
Load balancers install & admin
Hypervisor install & admin
Physical security
Power, HVAC
Rack and stack
Build Your Own-Cloud VDI
Image management
Directory services & policies
VDI control plane install & admin
Host admin
Storage admin
Load balancers install & admin
Hypervisor install & admin
Physical security
Power, HVAC
Rack and stack
WorkSpaces Core
Image management
Directory services & policies
VDI control plane install & admin
Host admin
Storage admin
Load balancers install & admin
Hypervisor install & admin
Physical security
Power, HVAC
Rack and stack
WorkSpaces
Image management
Directory services & policies
VDI control plane install & admin
Host admin
Storage admin
Load balancers install & admin
Hypervisor install & admin
Physical security
Power, HVAC
Rack and stack
Partner/Customer managed
Customer managed
Service managed

Prerequisites

To deploy a Amazon WorkSpaces Core-based virtual desktop infrastructure (VDI) using WorkSpaces Core bundles or Managed Instances, customers must meet the following requirements:

- Customers must either work with a technology partner or build and manage their own control plane. This includes brokering, orchestration, and pixel streaming capabilities.
- For customers who choose to deploy using WorkSpaces Core bundles, Active Directory integration is required. This can be achieved by deploying AWS Managed Microsoft AD in the customer's account or by using an existing on-premises or self-managed directory.
- Customers are expected to supply their own pixel streaming protocol, regardless of the provisioning model used.
- If the deployment includes the Windows Client operating system, customers must meet the requirements of the Bring Your Own License (BYOL) model. This includes using eligible Windows desktop licenses that are covered under Microsoft's licensing terms for AWS. Detailed information about BYOL eligibility and deployment can be found in the Amazon WorkSpaces Administration Guide.
- If customers are deploying desktops using the Windows Server operating system, they will need to provide Remote desktop licensing.
  - Amazon WorkSpaces Core includes a license that permits two Remote Desktop connections for administrative use only. To support additional concurrent user sessions, customers must acquire Microsoft Remote Desktop Services (RDS) Client Access Licenses (CALs) with active Software Assurance. These licenses can be brought to AWS through Microsoft's License Mobility program.
  - If customers have Microsoft Software Assurance with License Mobility, they might be able to bring their Microsoft RDS CALs and then use them with Amazon WorkSpaces Core. For more information about how to sign up for and complete a license verification process, and to view eligibility requirements, see License Mobility.
  - To verify license eligibility through License Mobility, complete the following steps:
    1. Confirm that your Microsoft licenses include Software Assurance and are eligible for License Mobility.
    2. Go to the Microsoft License Mobility Verification form.
    3. Fill out the form using the following AWS partner details:
       - Email Address: **`microsoft@amazon.com`**

- Partner Name: **`Amazon Web Services`**
- Partner Website: **`aws.amazon.com`**

4. Submit the form to Microsoft.
5. Wait for confirmation from Microsoft, which will be sent to both you and AWS upon successful verification. For more information, see Microsoft licensing on AWS.

For more information, see Microsoft licensing on AWS.

Deployment with WorkSpaces Core bundles

To deploy a Amazon WorkSpaces Core-based virtual desktop infrastructure (VDI) using WorkSpaces Core bundles, customers must meet the following requirements:

Contents

Infrastructure setup

Use the following steps to set up your customer's AWS account. As the technology partner, you perform some steps, and your customer also performs some steps.

Topics

Enable AWS account for Bring Your Own Protocol

To enable the customer AWS account for BYOP, customers must contact their AWS account manager. For select technology partners with hosted managed solutions, BYOP might be enabled at the technology partner solution level. In that case, the customer account won't need to have BYOP enabled within their account.

Grant partner solution access to AWS account

Partner step and Customer step – Create a technology partner solution connection to the customer's AWS account.

For more information, see AWS security credentials in the IAM User Guide. This connection can be done with secret and access keys for self-managed solutions. The preferred method is to use an assume role capability. For more information, see How to Use External ID When Granting Access to Your AWS Resources at the AWS Security Blog.

If assume role access is being used, the technology partner creates an assume role from the technology partner solution's AWS account to the customer's AWS account. You can provide the customer with an AWS CloudFormation template to automate creation of the role with permissions or instructions on permissions as needed.

If assume role access is being used, instruct your customer to use tag-based authorization. This limits exposure to customer resources from the role granted to the partner solution. For more information, see Tag-based authorization guidelines.

Enable the account for BYOL and configure the BYOL CIDR block (Windows client OS ONLY)

Follow these steps to enable Bring Your Own Licenses (BYOL), configure the BYOL Classless Inter-Domain Routing (CIDR) block, and register the directory.

1. *(Customer step)* – Enable BYOL.
   a. For information on how to enable BYOL see Bring Your Own Windows desktop licenses in the *Amazon WorkSpaces Administration Guide*.
2. *(Partner step)* – List and configure the management CIDR ranges.
   a. This is the management CIDR block that is required for the WorkSpaces dedicated control plane. WorkSpaces desktops have two elastic network interfaces: one network interface for the management network and another for access to a customer's virtual private cloud (VPC).

      First use the DescribeAccountModifications API to see if the customer has configured the CIDR block already. If they haven't, use the ListAvailableManagementCidrRanges API to provide a list of CIDR block ranges for the customer to select. Then use the ModifyAccount API to configure BYOL and provide the CIDR block.

> **Important**
>
> This action can not be changed once configured.

Import the Windows Client OS image (BYOL-BYOP)

Use the following steps to import the image.

1. *(Customer step)* – The customer must have an image within Amazon Elastic Compute Cloud (Amazon EC2) as an Amazon Machine Image (AMI). For more information, see Importing a VM as an image using VM Import/Export in the *VM Import/Export User Guide*.
2. *(Partner step)* – List the AMIs and display them to the customer admin by using the DescribeImages API.

```
describe-images - (EC2)
"VirtualizationType" (filter)
"Description" (display)
"PlatformDetails" (display)
"EnaSupport" (display) - instance types limit
"Hypervisor" (display) - instance types limit
"State" (filter)
"ImageId" (display)
"VolumeType" (display)
"VolumeSize" (display) - make sure meets WS requirements
"Encrypted" (display and filter) not supported
"OwnerId" (display)
"ImageType": "machine" (filter)
"Name" (display)
```

3. *(Customer step)* – Select the Amazon EC2 AMI.
4. *(Partner step)* – Import the image. Make sure to use the BYOP import ingestion process with the ImportWorkspaceImage Amazon WorkSpaces Core API. When doing so, choose an ingestion process option that meets your needs. For more information about the ingestion process options available, see IngestionProcess in the *WorkSpaces API Reference*.

   Following is an example command using the AWS CLI:

```
aws workspaces import-workspace-image --ec2-image-id ami-example123 --ingestion-
process BYOL_REGULAR_BYOP --image-name win10-ent-img01 --image-description “Windows
 10 Enterprise”
```

5. *(Partner step)* – Display the status of the import by using the DescribeWorkspaceImages API.

Configure the directory

Complete the following steps to configure the directory.

1. *(Partner step)* – Present the directories that the customer admin would choose for WorkSpaces using the DescribeWorkspaceDirectories API. Amazon WorkSpaces requires that you pre-configure a directory within the AWS Directory Service.
2. *(Partner step)* – Register the directory to AWS for this WorkSpaces to access using the RegisterWorkspaceDirectory API. This step is used for adding the desktop to Active Directory. Note that BYOL requires a tenancy of DEDICATED, all others must use SHARED

Add a security group to a WorkSpaces directory

You must allow for access from the customer VPC into the Amazon WorkSpaces Core desktop. WorkSpaces desktops, including Amazon WorkSpaces Core desktops, have a security group attached to the customer VPC elastic network interface. By default, this security group blocks all traffic.

For Remote Desktop Protocol (RDP) access or access from any other protocol that will be accessing the desktop, you must add or modify a security group to the WorkSpaces directory. For more information, see Security groups for your WorkSpaces in the *Amazon WorkSpaces Administration Guide*.

You can also add this new default security group to existing WorkSpaces without rebuilding them. For more information, see To add a security group to an existing WorkSpace in the *Amazon WorkSpaces Administration Guide*. Use caution when modifying or deleting these security groups. Customers are responsible for the “security in the cloud." For more information, see Shared Responsibility Model.

Deploy Amazon WorkSpaces Core desktops

Complete the following steps to deploy the Amazon WorkSpaces Core desktops.

1. *(Partner and customer step)* – Create a bundle using the CreateWorkspaceBundle API. Initially only needed for BYOL deployments. BYOL customers import their image first. They will need to create a bundle to deploy desktops. Unlike shared tenancy deployments where WorkSpaces provides a bundle which includes an image.

```
CreateWorkspaceBundle (Amazon WorkSpaces)
   "BundleDescription"
   "BundleName"
   "ComputeType"
   "ImageId"
   "RootStorage" - "Capacity"
   "Tags": [
   "UserStorage"
      "Capacity"
```

2. *(Partner and customer step)* – Create a WorkSpace using the CreateWorkspaces API.

> **Note**
>
> Amazon WorkSpaces Core (BYOP) supports user-decoupled and regular user-assigned WorkSpaces.

Following is an example command using the AWS CLI:

```
aws workspaces create-workspaces --workspaces DirectoryId=d-
example123,UserName='"[UNDEFINED]"',WorkspaceName=desktop1,BundleId=wsb-example123
```

For RunningMode, the AUTO_STOP mode isn't available for Amazon WorkSpaces Core. Instead, a new running mode value of MANUAL is available for technology partner solutions to power manage the workspace and offer hourly usage of the instance. With the MANUAL mode, technology partner solutions use the StartWorkSpaces and StopWorkSpaces API operations to manage the workspaces. The customer is only charged for the hours when the WorkSpace is in the AVAILABLE state.

> **Note**
>
> To ensure that no workspaces are inadvertently charging the customer for unknown periods of time, manual workspaces in the AVAILABLE state will be stopped after a

sufficiently long period of time (greater than or equal to 48 hours). Manual workspaces are subject to an automatic maintenance window schedule once a month, similar to the current `AUTO_STOP` workspaces detailed here. You can opt out of this maintenance schedule by using the `ModifyWorkspaceCreationProperties` API operation.

Custom images

After you deploy a WorkSpace, you can customize the image being used by customers moving forward. For example, if you use a shared tenancy bundle for BYOP and you'd like to install a partner solution agent, or install productivity or proprietary applications within an image. This is often referred to as *golden image creation*.

You can customize an image using the CreateWorkspaceImage API. You can then use use the CreateWorkspaceBundle or UpdateWorkspaceBundle API. Then deploy WorkSpaces as described within this document.

Tag-based authorization guidelines

Tag-based authorization can prevent you from modifying customer resources. This strategy utilizes IAM tag conditions. You assume a role in your customer's account, and the role will have IAM policies based on tag conditions. When you create a resource in your customer's account, the policy requires a specific tag to be added. And when you modify a resource in your customer's account, the policy ensures that it only allows modification on resources with the specified tags. You should not have permission to modify or delete tags on a resource. To create a complete IAM policy for the assume role, the customer can use the following examples.

Topics

Tag conditions

TagKeys condition

To ensure that only a specific tag key can be used in a request, use the `aws:TagKeys` condition key.

RequestTag condition

To ensure that a specific tag key and value will be put on the resource, use a combination of the `aws:TagKeys` and `aws:RequestTag` condition keys. This applies to resource creation API actions, such as CreateWorkspaces.

The following tag keys policy example only allows API actions to use tag keys “PartnerManaged.”

```
{
"Version":"2012-10-17",
"Statement":[
{
"Effect":"Allow",
"Action":[
ws:CreateWorkspaces
],
"Resource":"*",
"Condition":{
"StringEquals": {
"aws:RequestTag/PartnerManaged": "true"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": "PartnerManaged"
}
}
}
]
}
```

ResourceTag condition

To control access to a customer’s resources based on the tag key and value use a combination of the `aws:TagKeys` and `aws:ResourceTag` condition keys. This applies to modifications related to API actions, such as ModifyWorkspaceProperties.

The following resource tag policy example ensures that modifications can only happen on resources with the tag “Key=PartnerManaged, Value=true”.

```
{
"Version":"2012-10-17",
"Statement":[
{
```

```
"Effect":"Allow",
"Action":[
ws:ModifyWorkspaceProperties
],
"Resource":"*",
"Condition":{
"StringEquals":{
"aws:ResourceTag/PartnerManaged":"true"
},
"ForAllValues:StringEquals": {
"aws:TagKeys": "PartnerManaged"
}
}
}
]
}
```

Additional examples

API name	Tag condition request	Assumed role policy for UserTag	Note
CreateWorkSpaces	TagKeys + RequestTag	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "workspaces:CreateWorkspaces" ], "Resource":"*", "Condition":{ "StringEquals": { "aws:RequestTag/PartnerManaged":"true" },`	With this policy, you can only create a workspace if you provide a tag key "PartnerManaged" and value "true" in the request.

API name	Tag condition request	Assumed role policy for UserTag	Note

		"ForAllV
		alues:Str
		ingEquals":{
		"aws:TagKeys":
		"PartnerM
		anaged"
		}
		}
		}
		]
		}	

API name	Tag condition request	Assumed role policy for UserTag	Note
TerminateWorkSpaces	TagKeys + RequestTag	{ "Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":["workspaces:TerminateWorkspaces"], "Resource":"*", "Condition":{ "StringEquals": { "aws:ResourceTag/PartnerManaged":"true" }, "ForAllValues:StringEquals":{ "aws:TagKeys":"PartnerManaged" } } }] }	With this policy, you can only terminate a workspace if the workspace has a tag key "PartnerManaged" and value "true".

WorkSpaces Core bundles management

To perform various actions for Amazon WorkSpaces Core, use the following API operations. To help you create your workflow, we have provided a recommendation for each API operation. We

recommend partners solutions use as many of these APIs as possible so that admin customers don't need to access the WorkSpaces console.

- Deployment and setup
  - CreateTags
  - DescribeAccount
  - DescribeAccountModifications
  - ImportWorkspaceImage
  - ModifyAccount
  - ListAvailableManagementCidrRanges
  - RegisterWorkspaceDirectory
- Operations
  - CopyWorkspaceImage – Supports an `UpdateWorkspaceBundle` image process and copying from one AWS Region to another Region.
  - CreateWorkspaceImage – Supports custom images and workflows for day-two operations.
  - DescribeTags
  - DescribeWorkspaceBundles
  - DescribeWorkspaceDirectories
  - DescribeWorkspaceImagePermissions
  - DescribeWorkspaceImages
  - DescribeWorkspaces
  - DescribeWorkspaceSnapshots
  - MigrateWorkspace
  - ModifyWorkspaceCreationProperties
  - ModifyWorkspaceProperties – Supports modification of the following properties:
    - ComputeTypeName
    - RootVolumeSizeGib
    - RunningMode – BYOP must use `ALWAYS_ON` or `MANUAL`.
    - UserVolumeSizeGib
  - ModifyWorkspaceState
  - RebootWorkspaces

- RebuildWorkspaces
- RestoreWorkspace
- StartWorkspaces
- StopWorkspaces
- UpdateWorkspaceBundle
- UpdateWorkspaceImagePermission

- Termination
  - DeleteTags
  - DeleteWorkspaceBundle
  - DeleteWorkspaceImage
  - DeregisterWorkspaceDirectory
  - TerminateWorkspaces

Deployment with WorkSpaces Core Managed Instances

To deploy a Amazon WorkSpaces Core-based virtual desktop infrastructure (VDI) using WorkSpaces Managed Instances, customers must meet the following requirements:

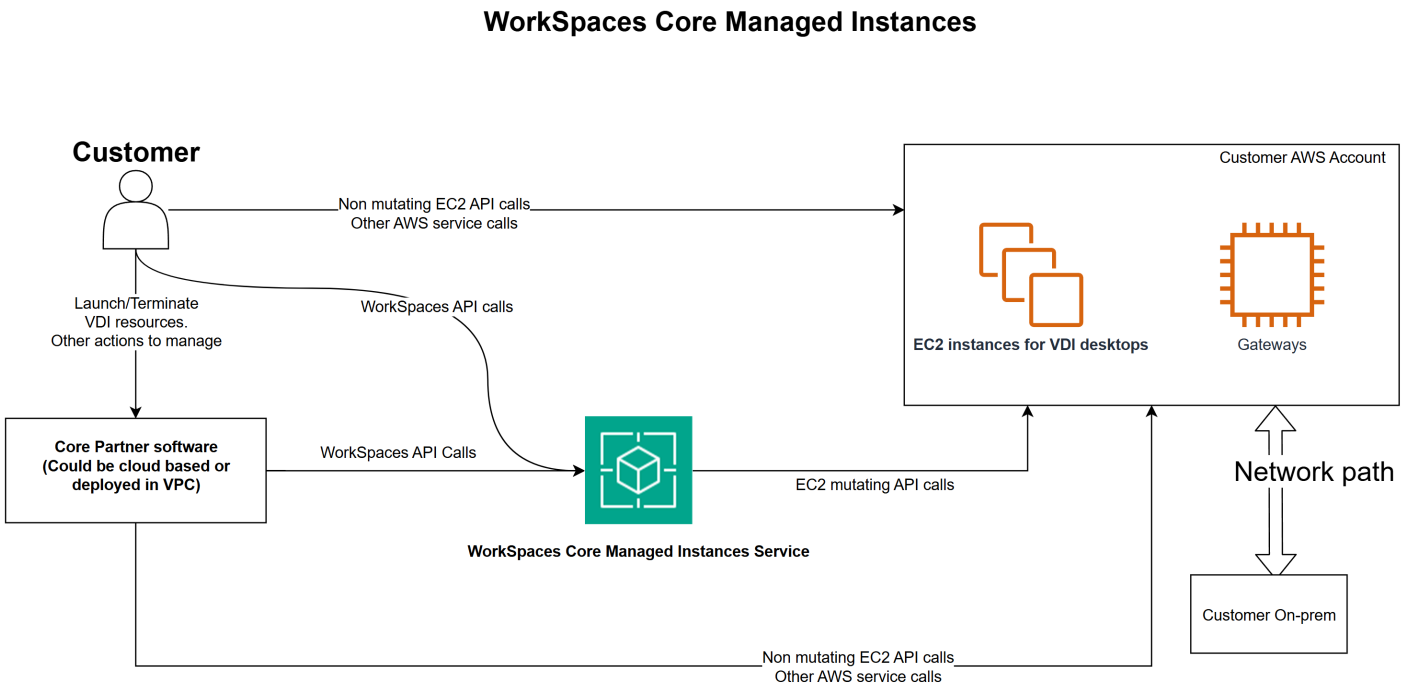
Contents

Architecture

Amazon WorkSpaces Core Managed Instances introduce an updated architecture that allows EC2 instances to be launched directly into the customer's AWS account, rather than into an account owned by Amazon WorkSpaces Core. These instances are referred to as WorkSpaces Core managed EC2 instances. Under this model, WorkSpaces Core partners have direct access to most EC2 APIs within the customer's account. For mutable operations such as launch and terminate an instance, partners must use the Amazon WorkSpaces Core SDK instead of native EC2 APIs.

WorkSpaces Core Managed Instances operate with:

- Direct EC2 instance deployment in your AWS account
- Native AWS feature support (AMIs, KMS, Systems Manager)
- WorkSpaces Core SDK for instance lifecycle management



This model differs significantly from Amazon WorkSpaces Core bundles, which rely on pre-defined infrastructure launched within Amazon WorkSpaces owned accounts. Concepts such as directories, bundles, and images from Amazon WorkSpaces Core bundles do not apply here.

API Operations

For Amazon WorkSpaces Instances API information see WorkSpaces Instances API Reference.

Setting up partner access to AWS accounts

This section explains how to deploy WorkSpaces Core Managed Instances in your AWS environment. You can enable partner solutions to access your AWS account by completing a connection setup with your technology partner. This connection establishes secure access to your AWS environment.

WorkSpaces Core Managed Instances allow EC2 instances to launch directly in your AWS account rather than in an Amazon WorkSpaces Core-owned account. This architecture provides more flexibility and direct access to AWS services while maintaining WorkSpaces Core management capabilities.

For general information about security credentials, see AWS security credentials the *IAM User Guide*.

Contents

Grant authorization and permissions

Authorization and permissions in WorkSpaces Core Managed Instances determine who deploys your WorkSpaces resources. IAM (Identity and Access Management) controls permissions, allowing administrators to define specific roles and policies that govern user actions and resource access.

AWS recommends using IAM Roles for partners to get access to the customer's environment. This avoids inputting long-term access keys and secrets into external systems. For more information on how to set this up, refer to your partner specific guides.

Customer and WorkSpaces Core Partnership

Customers must grant appropriate IAM permissions to the Core partner software to perform required AWS API calls. These include:

- Existing permissions already used in EC2-based partner integrations.
- New permissions to call the WorkSpaces Core APIs listed above.

Required IAM permissions

The WorkspacesInstances APIs will be called using an IAM role or user credentials from the WorkSpaces Core partner's account. For more information, see Identity and access management for WorkSpaces Instances in the *Amazon WorkSpaces Core Administration Guide*.

Create a Service-Linked Role

WorkSpaces Core Managed Instances require an IAM service-linked role. This role:

- Contains predefined trust and permissions policies.
- Can only be assumed by WorkSpaces Instances.
- Must be removed after associated resources are deleted.

For more information on service linked roles, see Using service-linked roles for Amazon WorkSpaces Instances

Solution deployment guide example

As a partner who is building a solution using Amazon WorkSpaces Core, it's your responsibility to document how your customers can deploy your solution to their environments. We recommend that you create a deployment guide, with the following suggested table of contents. Some topics might not be relevant to your solution, so revise the topics as necessary.

It's also a good practice to link to other AWS documentation where relevant. For example, refer your customers to the Amazon WorkSpaces Administration Guide for sections related to Bring Your Own License (BYOL) image import, directory setup, and virtual private cloud (VPC) setup. Specific details of your deployment guide and steps will vary, depending on the level of integration of your solution with the WorkSpaces API, and what steps customers must take manually using the AWS Management Console or AWS Command Line Interface.

As a partner, you're responsible for hosting and publishing the deployment guides on your website. Amazon WorkSpaces Core can link to these guides from the **WorkSpaces Core Partners** section at Amazon WorkSpaces Core, where customers can easily find them.

Following is a suggested table of contents for an Amazon WorkSpaces Core solution deployment guide:

- Chapter 1: Introduction
- Chapter 2: Getting started
  - Overview
  - Setting up security groups
  - Configuring the directory services security group
  - Configuring a VPC
- Chapter 3: Installing <your service> in Amazon EC2
  - Required AWS permissions
  - Launching a connection broker instance
  - Upgrading the <your service> connection broker
  - Lauching a <your service> gateway instance
  - Obtaining your <your service> license
- Chapter 4: Preparing WorkSpaces Core images
- Chapter 5: Integrating with your AWS infrastructure

- Connecting to your Amazon diretory services
- Connecting to your Amazon WorkSpaces account
- Attaching the <your service> gateway to a connection broker

- Chapter 6: Launching new workspaces
  - Loading users
  - Deploying new workspaces
- Chapter 7: Connecting users to WorkSpaces
  - Amazon WorkSpaces pools
  - Protocol plans
  - Power control plans
  - Release plans
  - Building user policies
  - Assigning policies to users
  - Testing your connection broker configuration
  - Connecting to WorkSpaces

Document history for the Amazon WorkSpaces Core Technology Partner Integration Guide

The following table describes the documentation releases for Amazon WorkSpaces Core.

Change	Description	Date
Added new topic	Added "Deployment with WorkSpaces Core Managed Instances" topic	June 23, 2025
Added new topic	Added "Tag-based authorization guidelines" topic	April 1, 2024
Initial release	Initial release of the Amazon WorkSpaces Core Technology Partner Integration Guide.	September 20, 2023