Best Practices for Deploying Amazon WorkSpaces

Network Access, Directory Services, Cost Optimization and Security

June 2020

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 current AWS 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. The responsibilities and liabilities of AWS 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.

Contents

Abstract

This whitepaper outlines a set of best practices for the deployment of Amazon WorkSpaces. The paper covers network considerations, directory services and user authentication, security, and monitoring and logging.

The document is broken into four categories to enable quicker access to relevant information. This document is intended for a network engineer, directory engineer, or security engineer.

Introduction

Amazon WorkSpaces is a managed desktop computing service in the cloud. Amazon WorkSpaces removes the burden of procuring or deploying hardware or installing complex software, and delivers a desktop experience with either a few clicks on the AWS Management Console, using the AWS command line interface (CLI), or by using the application programming interface (API). With Amazon WorkSpaces, you can launch a Microsoft Windows or Amazon Linux desktop within minutes, and connect to and access your desktop software from on-premises or an external network securely, reliably, and quickly. You can:

- Leverage your existing on-premises Microsoft Active Directory (AD) by using AWS Directory Service: AD Connector.
- Extend your directory to the AWS Cloud.
- Build a managed directory with AWS Directory Service: Microsoft AD or Simple AD, to manage your users and WorkSpaces.
- Leverage your on-premises or cloud-hosted RADIUS server with AD Connector to provide multi-factor authentication (MFA) to your WorkSpaces.

You can automate the provisioning of Amazon WorkSpaces by using the CLI or API, which enables you to integrate Amazon WorkSpaces into your existing provisioning workflows.

For security, in addition to the integrated network encryption that the Amazon WorkSpaces service provides, you can also enable encryption at rest for your WorkSpaces (see Encrypted WorkSpaces in the security section).

You can deploy applications to your WorkSpaces by using your existing on-premises tools, such as Microsoft System Center Configuration Manager (SCCM), Puppet Enterprise, or Ansible.

The following sections provide details about Amazon WorkSpaces, explain how the service works, describe what you need to launch the service, and lets you know what options and features are available for you to use.

WorkSpaces Requirements

The Amazon WorkSpaces service requires three components to deploy successfully:

- **WorkSpaces client application**. An Amazon WorkSpaces-supported client device. Find a full list here: Supported Platforms and Devices.

  You can also use Personal Computer over Internet Protocol (PCoIP) Zero Clients to connect to WorkSpaces. For a list of available devices, see PCoIP Zero Clients for Amazon WorkSpaces.

- **A directory service to authenticate users and provide access to their WorkSpace**. Amazon WorkSpaces currently works with AWS Directory Service and Microsoft Active Directory. You can use your on-premises Active Directory server with AWS Directory Service to support your existing enterprise user credentials with Amazon WorkSpaces.

- **Amazon Virtual Private Cloud (Amazon VPC) in which to run your Amazon WorkSpaces**. You'll need a minimum of two subnets for an Amazon WorkSpaces deployment because each AWS Directory Service construct requires two subnets in a Multi-AZ deployment.

Network Considerations

Each WorkSpace is associated with the specific Amazon VPC and AWS Directory Service construct that you used to create it. All AWS Directory Service constructs (Simple AD, AD Connector, and Microsoft AD) require two subnets to operate, each in different Availability Zones. Subnets are permanently affiliated with a Directory Service construct and can't be modified after it is created. Therefore, it's imperative that you determine the right subnet sizes before you create the Directory Services construct. Carefully consider the following before you create the subnets:

- How many WorkSpaces will you need over time?
- What is the expected growth?
- What types of users will you need to accommodate?
- How many Active Directory domains will you connect?
- Where do your enterprise user accounts reside?

Amazon recommends defining user groups, or personas, based on the type of access and the user authentication you require as part of your planning process. Answers to these questions are helpful when you need to limit access to certain applications or resources. Defined user personas can help you segment and restrict access using AWS Directory Service, network access control lists, routing tables, and VPC security groups.

Each AWS Directory Service construct uses two subnets and applies the same settings to all WorkSpaces that launch from that construct. For example, you can use a security group that applies to all WorkSpaces attached to an AD Connector to specify whether MFA authentication is required, or whether an end-user can have local administrator access on their WorkSpace.

> **Note:** Each AD Connector connects to one Microsoft Active Directory organizational unit (OU). To take advantage of this capability, you must construct your Directory Service to take your user personas into consideration.

VPC Design

This section describes best practices for sizing your VPC and subnets, traffic flow, and implications for directory services design.

Here are a few things to consider when designing the VPC, subnets, security groups, routing policies, and network ACLs for your Amazon WorkSpaces so that you can build your WorkSpaces environment for scale, security, and ease of management:

- **VPC**. We recommend using a separate VPC specifically for your WorkSpaces deployment. With a separate VPC, you can specify the necessary governance and security guardrails for your WorkSpaces by creating traffic separation.
- **Directory Services**. Each AWS Directory Service construct requires a pair of subnets that provides a highly available directory service split between Amazon AZs.
- **Subnet size**. WorkSpaces deployments are tied to a directory construct and reside in the same VPC subnets as your chosen AWS Directory Service. A few considerations:
    - Subnet sizes are permanent and cannot change. You should leave ample room for future growth.
    - You can specify a default security group for your chosen AWS Directory Service. The security group applies to all WorkSpaces that are associated with the specific AWS Directory Service construct.
    - You can have multiple AWS Directory Services use the same subnet.

Consider future plans when you design your VPC. For example, you might want to add management components, such as an antivirus server, a patch management server, or an Active Directory or RADIUS MFA server. It's worth planning for additional available IP addresses in your VPC design to accommodate such requirements.

For in-depth guidance and considerations for VPC design and subnet sizing, see the **re:Invent** presentation How Amazon.com is Moving to Amazon WorkSpaces.

Network Interfaces

Each WorkSpace has two elastic network interfaces (ENIs), a management network interface (eth0), and a primary network interface (eth1). AWS uses the management network interface to manage the WorkSpace—it's the interface on which your client connection terminates. AWS uses a private IP address range for this interface. For network routing to work properly, you can't use this private address space on any network that can communicate with your WorkSpaces VPC.

For a list of the private IP ranges that we use on a per region basis, see Amazon WorkSpaces Details.

> **Note:** Amazon WorkSpaces and their associated management network interfaces do not reside in your VPC, and you cannot view the management network interface or the Amazon Elastic Compute Cloud (Amazon EC2) instance ID in your AWS Management Console (see Figures 4, 5, and 6). However, you can view and modify the security group settings of your primary network interface (eth1) in the console. Also, the primary network interface of each WorkSpace does count toward your ENI Amazon EC2 resource quotas. For large deployments of Amazon WorkSpaces, you would need to open a support ticket via the AWS Management Console to increase your ENI quotas.

Traffic Flow

You can break down Amazon WorkSpaces traffic into two main components:

- The traffic between the client device and the Amazon WorkSpaces service
- The traffic between the Amazon WorkSpaces service and customer network traffic

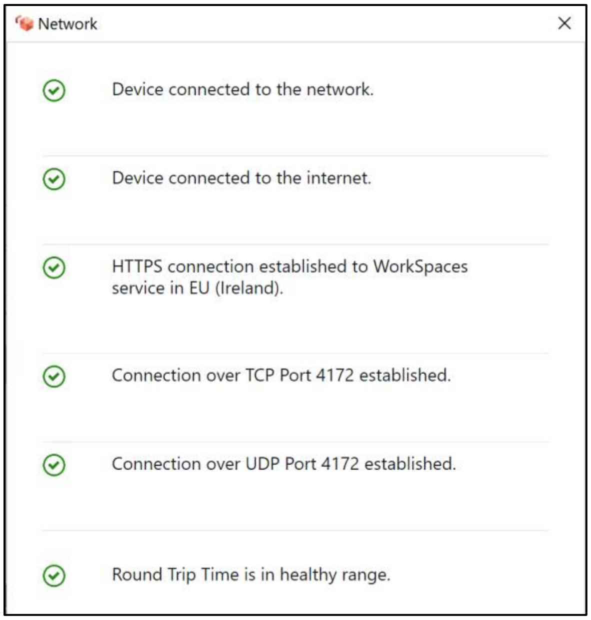
In the next section, we discuss both of these components.

Client Device to WorkSpace

Regardless of its location (on premises or remote), the device running the Amazon WorkSpaces client uses the same two ports for connectivity to the Amazon WorkSpaces service. The client uses port 443 (HTTPS port) for all authentication and session-related information, and port 4172 (PCoIP port), with both TCP and UDP, for pixel streaming to a given WorkSpace and network health checks. Traffic on both ports is encrypted. Port 443 traffic is used for authentication and session information and uses TLS for encrypting the traffic. Pixel streaming traffic uses AES-256-bit encryption for communication between the client and eth0 of the WorkSpace, via the streaming gateway. More information can be found in the Security section, later in this document.

We publish per-region IP ranges of our PCoIP streaming gateways and network health check endpoints. You can limit outbound traffic on port 4172 from your corporate network to the AWS streaming gateway and network health check endpoints by allowing only outbound traffic on port 4172 to the specific AWS Regions in which you're using Amazon WorkSpaces. For the IP ranges and network health check endpoints, see Amazon WorkSpaces PCoIP Gateway IP Ranges.

The Amazon WorkSpaces client has a built-in network status check. This utility shows users whether their network can support a connection by way of a status indicator on the bottom right of the application. A more detailed view of the network status can be accessed by selecting **Network** on the bottom-right side of the client, the result of which is shown in Figure 1.



✓ Device connected to the network.

✓ Device connected to the internet.

✓ HTTPS connection established to WorkSpaces service in EU (Ireland).

✓ Connection over TCP Port 4172 established.

✓ Connection over UDP Port 4172 established.

✓ Round Trip Time is in healthy range.

Figure 1: WorkSpaces Client — network check

A user initiates a connection from their client to the Amazon WorkSpaces service by supplying their login information for the directory used by the Directory Service construct, typically your corporate directory. The login information is sent via HTTPS to the authentication gateways of the Amazon WorkSpaces service in the Region where the WorkSpace is located. The authentication gateway of the Amazon WorkSpaces service then forwards the traffic to the specific AWS Directory Service construct associated with your WorkSpace.

For example, when using the AD Connector, the AD Connector forwards the authentication request directly to your Active Directory service, which could be on premises or in an AWS VPC (see AD DS Deployment Scenarios). The AD Connector does not store any authentication information and acts as a stateless proxy. As a result, it’s imperative that the AD Connector has connectivity to an Active Directory server. The AD Connector determines which Active Directory server to connect to by using the DNS servers that you define when you create the AD Connector.

If you’re using an AD Connector and you have MFA enabled on the directory, the MFA token is checked before the directory service authentication. Should the MFA validation fail, the user’s login information is not forwarded to your AWS Directory Service.

Once a user is authenticated, the streaming traffic starts by using port 4172 (PCoIP port) through the AWS streaming gateway to the WorkSpace. Session-related information is still exchanged via HTTPS throughout the session. The streaming traffic uses the first ENI on the WorkSpace (eth0 on the WorkSpace) that is not connected to your VPC. The network connection from the streaming gateway to the ENI is managed by AWS. In the event of a connection failure from the streaming gateways to the WorkSpaces streaming ENI, a CloudWatch event is generated (see Monitoring or Logging Using Amazon CloudWatch section of this whitepaper).

The amount of data that is sent between the Amazon WorkSpaces service and the client depends on the level of pixel activity. To ensure an optimal experience for users, we recommend that the round-trip time (RTT) between the WorkSpaces client and the AWS Region where your WorkSpaces are located is less than 100 ms. Typically this means your WorkSpaces client is located less than two thousand miles from the Region in which the WorkSpace is being hosted. We provide a Connection Health Check webpage that you can refer to in order to determine the most optimal AWS Region to connect to the Amazon WorkSpaces service.

Amazon WorkSpaces Service to VPC

After a connection is authenticated from a client to a WorkSpace and streaming traffic is initiated, your WorkSpaces client will display a Windows desktop (your WorkSpace) that is connected to your VPC, and your network should show that you have established that connection. The WorkSpace’s primary ENI, identified as eth1, will have an IP address assigned to it from the Dynamic Host Configuration Protocol (DHCP) service that is provided by your VPC, typically from the same subnets as your AWS Directory Service. The IP address stays with the WorkSpace for the duration of the life of the WorkSpace. The ENI that is in your VPC has access to any resource in the VPC and to any network that you have connected to your VPC (via a VPC peering, an AWS Direct Connect connection, or VPN connection).

ENI access to your network resources is determined by the default security group (see more on security groups here) that your AWS Directory Service configures for each WorkSpace and any additional security groups that you assign to the ENI. You can add security groups to the ENI facing your VPC at any time by using the AWS Management Console or AWS CLI. In addition to security groups, you can use your preferred host-based firewall on a given WorkSpace to limit network access to resources within the VPC.

Figure 4 in AD DS Deployment Scenarios, later in this whitepaper, shows the traffic flow described.

Example of a Typical Configuration

Let's consider a scenario where you have two types of users and your AWS Directory Service uses a centralized Active Directory for user authentication:

- **Workers who need full access from anywhere** (for example, full-time employees). These users will have full access to the internet and the internal network, and they will pass through a firewall from the VPC to the on-premises network.
- **Workers who should have only restricted access from inside the corporate network** (for example, contractors and consultants). These users have restricted internet access through a proxy server to specific websites in the VPC, and will have limited network access in the VPC and to the on-premises network.

You'd like to give full-time employees the ability to have local administrator access on their WorkSpace to install software and you would like to enforce two-factor authentication with MFA. You also want to allow full-time employees to access the internet without restrictions from their WorkSpace.

For contractors, you want to block local administrator access so that they can only use specific pre-installed applications. You want to apply very restrictive network access controls using security groups for these WorkSpaces. You need to open ports 80 and 443 to specific internal websites only, and you would like to entirely block their access to the internet.

In this scenario, there are two completely different types of user personas with different requirements for network and desktop access. It's a best practice to manage and configure their WorkSpaces differently. You will need to create two AD Connectors, one for each user persona. Each AD Connector requires two subnets that have enough IP addresses available to meet your WorkSpaces usage growth estimates.

> **Note:** Each AWS VPC subnet consumes five IP addresses (the first four and the last IP address) for management purposes and each AD Connector consumes one IP address in each subnet in which it persists.

Further considerations for this scenario are as follows:

- AWS VPC subnets should be private subnets, so that traffic, such as internet access, can be controlled through either a NAT Gateway, Proxy-NAT server in the cloud, or routed back through your on-premises traffic management system.

- A firewall is in place for all VPC traffic bound for the on-premises network.
- Microsoft Active Directory server and the MFA RADIUS servers are either on-premises (see Scenario 1: Using AD Connector to Proxy Authentication to On-Premises AD DS) or part of the AWS Cloud implementation (see Scenarios 2 and 3, AD DS Deployment Scenarios).

Given that all WorkSpaces are granted some form of internet access, and given that they are hosted in a private subnet, you also must create public subnets that can access the internet through an internet gateway. You need a NAT gateway for the full-time employees allowing them to access the internet, and a Proxy-NAT server for the consultants and contractors to limit their access to specific internal websites. To plan for failure, design for high availability, and limit cross-AZ traffic charges, you should have two NAT gateways and NAT or proxy servers in two different subnets in a Multi-AZ deployment. The two Availability Zones that you select as public subnets will match the two Availability Zones that you use for your WorkSpaces subnets, in Regions that have more than two zones. You can route all traffic from each WorkSpaces AZ to the corresponding public subnet to limit cross-AZ traffic charges and provide easier management. Figure 2 shows the VPC configuration.

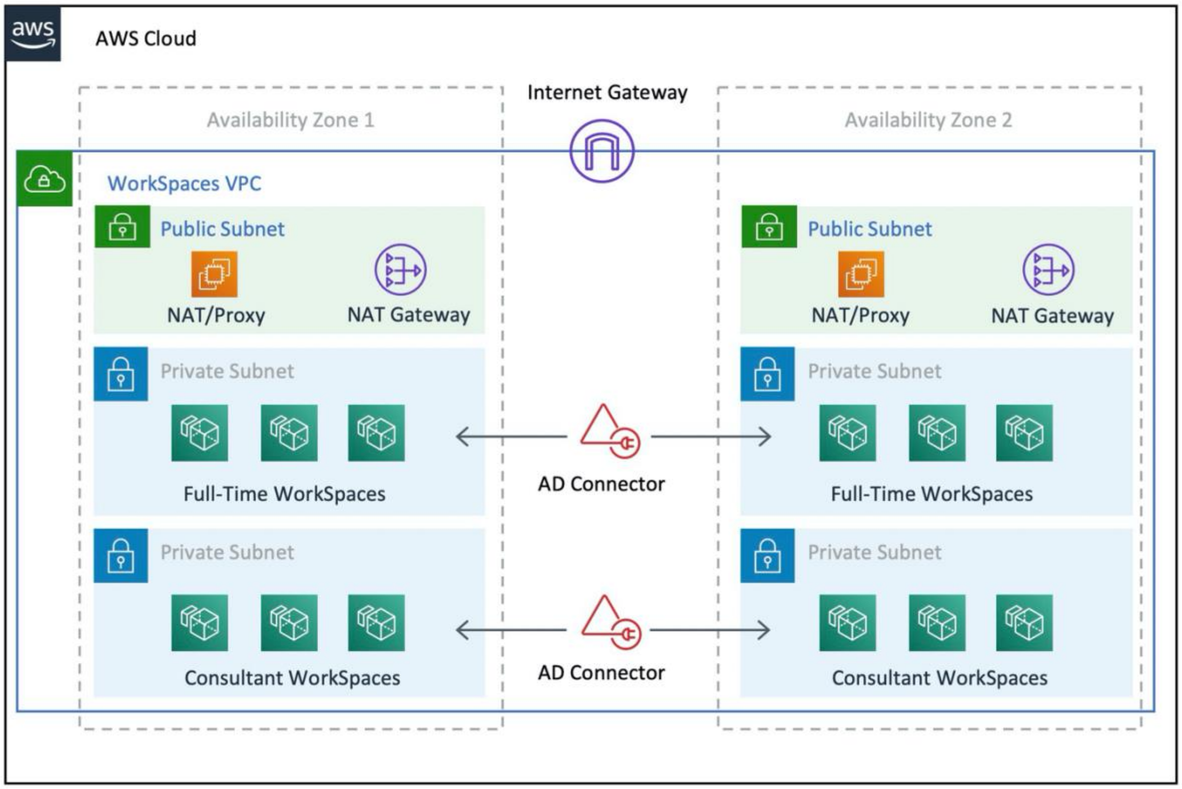


Figure 2: High-level VPC design

The following information describes how to configure the two different WorkSpaces types described earlier.

- **Full-time employees:** In the Amazon WorkSpaces Management Console, select the **Directories** option on the menu bar, select the directory that hosts your full-time employees, and then select **Local Administrator Setting**. By enabling this option, any newly created WorkSpace will have local administrator privileges. To grant internet access, you should configure Network Address Translation (NAT) for outbound internet access from your VPC. To enable MFA, you need to specify a RADIUS server, server IPs, ports, and a pre-shared key.

  For full-time employees' WorkSpaces, inbound traffic to the WorkSpace would be limited to Remote Desktop Protocol (RDP) from the Helpdesk subnet by applying a default security group via the AD Connector settings.

- **Contractors and consultants:** In the Amazon WorkSpaces Management Console, disable **Internet Access** and the **Local Administrator Setting**. Then add a security group under the **Security Group** settings section to enforce a security group for all new WorkSpaces created under that directory.

For consultants' WorkSpaces, limit outbound and inbound traffic to the WorkSpaces by applying a default Security group via the AD Connector settings to all WorkSpaces associated with the AD Connector. The security group prevents outbound access from the WorkSpaces to anything other than HTTP and HTTPS traffic, and inbound traffic to RDP from the Helpdesk subnet in the on-premises network.

> **Note:** The security group applies only to the ENI that is in the VPC (eth1 on the WorkSpace), and access to the WorkSpace from the WorkSpaces client is not restricted as a result of a security group. Figure 3 shows the final WorkSpaces VPC design described earlier.

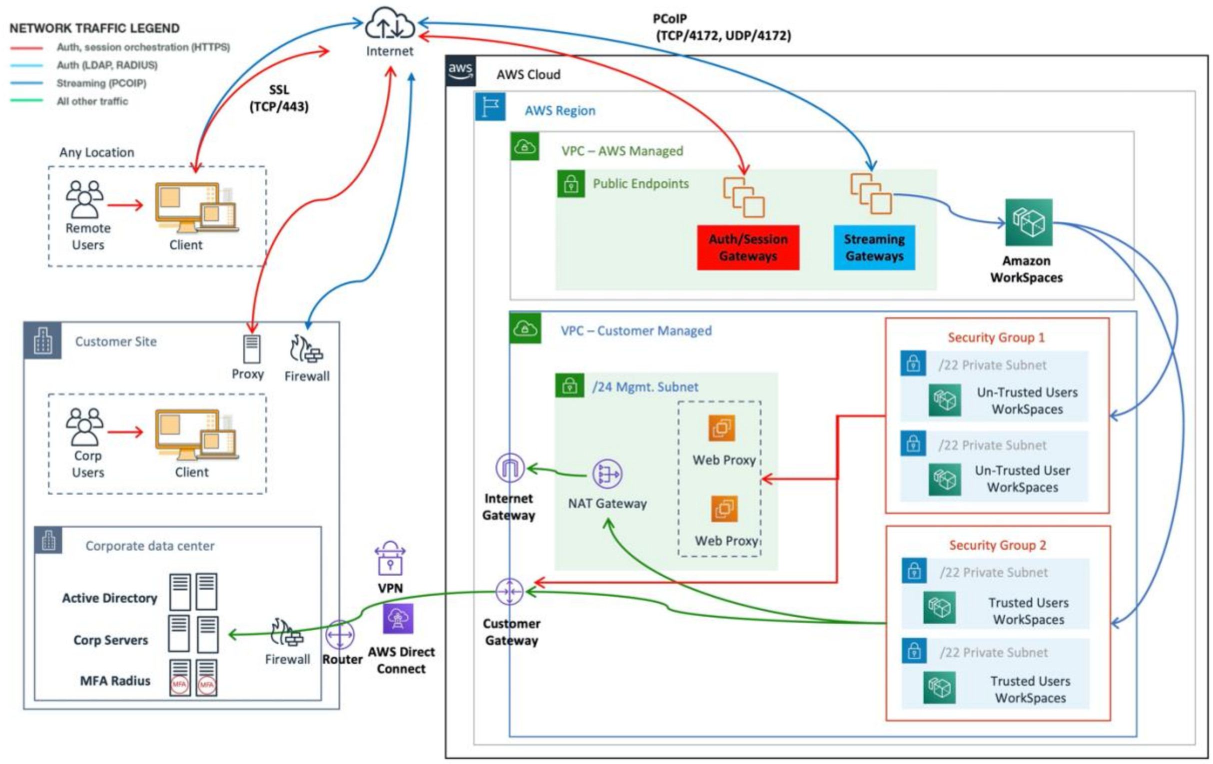


Figure 3: WorkSpaces design with user personas

AWS Directory Service

As mentioned in the introduction, AWS Directory Service is a core component of Amazon WorkSpaces. With AWS Directory Service, you can create three types of directories with Amazon WorkSpaces:

- **AWS Managed Microsoft AD**, which is a managed Microsoft Active Directory, powered by Windows Server 2012 R2. AWS Managed Microsoft AD is available in Standard or Enterprise Edition.
- **Simple AD** is standalone, Microsoft Active Directory-compatible, managed directory service powered by Samba 4.
- **AD Connector** is a directory proxy for redirecting authentication requests and user or group lookups to your existing on-premises Microsoft Active Directory.

The following section describes communication flows for authentication between the Amazon WorkSpaces brokerage service and AWS Directory Service, best practices for implementing WorkSpaces with AWS Directory Service, and advanced concepts, such as MFA. It also discusses infrastructure architecture concepts for Amazon WorkSpaces at scale, requirements on Amazon VPC, and AWS Directory Service, including integration with on-premises Microsoft Active Directory Domain Services (AD DS).

AD DS Deployment Scenarios

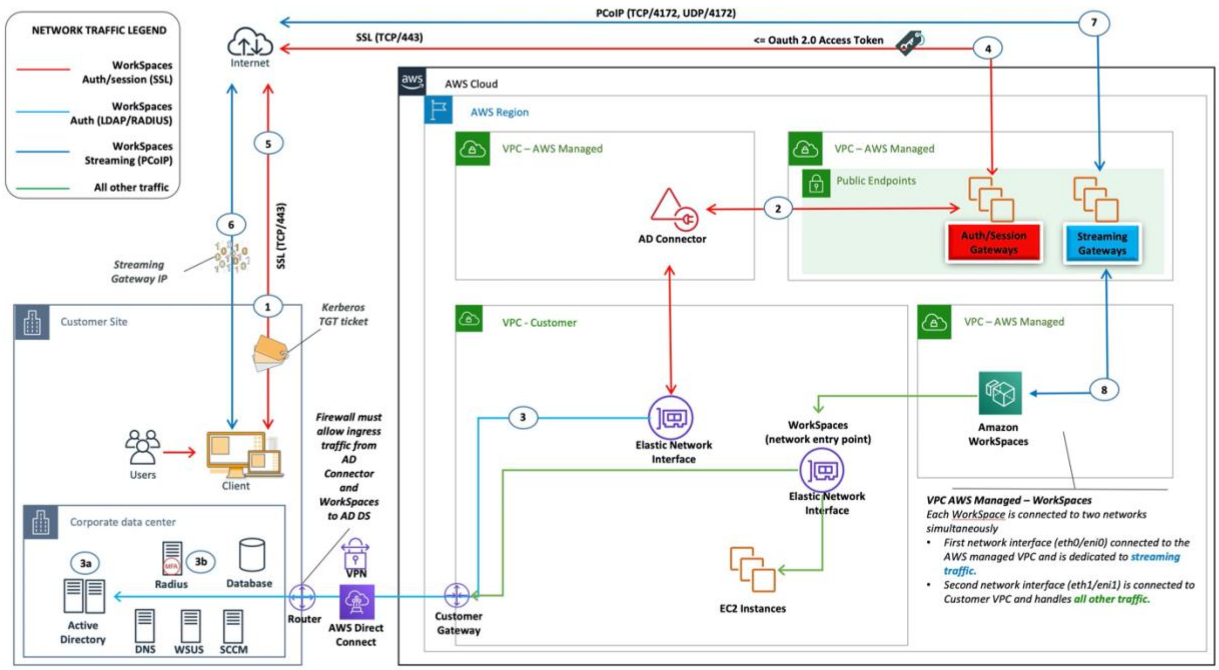
Backing Amazon WorkSpaces is the AWS Directory Service, and the proper design and deployment of the directory service is critical. The following three scenarios build upon the *Active Directory Domain Services on AWS* Quick Start guide, and describe the best practice deployment options for AD DS when used with Amazon WorkSpaces. The *Design Considerations* section details the specific requirements and best practices of using AD Connector for WorkSpaces, which is an integral part of the overall WorkSpaces design concept.

- **Scenario 1: Using AD Connector to proxy authentication to on-premises AD DS.** In this scenario, network connectivity (VPN/Direct Connect) is in place to the customer, with all authentication proxied via AWS Directory Service (AD Connector) to the customer on-premises AD DS.
- **Scenario 2: Extending on-premises AD DS into AWS (Replica).** This scenario is similar to scenario 1, but here a replica of the customer AD DS is deployed on AWS in combination with AD Connector, reducing latency of authentication/query requests to AD DS and the AD DS global catalog.
- **Scenario 3: Standalone isolated deployment using AWS Directory Service in the AWS Cloud.** This is an isolated scenario and doesn't include connectivity back to the customer for authentication. This approach uses AWS Directory Service (Microsoft AD) and AD Connector. Although this scenario doesn't rely on connectivity to the customer for authentication, it does make provision for application traffic where required over VPN or Direct Connect.
- **Scenario 4: AWS Microsoft AD and a Two-Way Transitive Trust to On-Premises.** This scenario includes the AWS Managed Microsoft Active Directory Service (MAD) with a two-way transitive trust to the on-premises Microsoft AD forest.
- **Scenario 5: AWS Microsoft AD using a Shared Services Virtual Private Cloud (VPC).** This scenario uses AWS Managed Microsoft AD in a Shared Services VPC to be used as an Identity Domain for multiple AWS Services (Amazon EC2, Amazon WorkSpaces, etc.) while using the AD Connector to proxy LDAP user authentication requests to the AD domain controllers.

- **Scenario 6: AWS Microsoft AD, Shared Services VPC, and a One-Way Trust to On-Premises AD.** This scenario is similar to Scenario 5, but it includes disparate identity and resource domains using a one-way trust to on-premises.

Scenario 1: Using AD Connector to Proxy Authentication to On-Premises Active Directory service

This scenario is for customers who don't want to extend their on-premises Active Directory service into AWS, or where a new deployment of AD DS is not an option. Figure 4: AD Connector to on-premises Active Directory depicts, at a high level, each of the components and shows the user authentication flow.



VPC AWS Managed – WorkSpaces
Each WorkSpace is connected to two networks simultaneously

- *First network interface (eth0/eni0) connected to the AWS managed VPC and is dedicated to streaming traffic.*
- *Second network interface (eth1/eni1) is connected to Customer VPC and handles all other traffic.*

Figure 4: AD Connector to on-premises Active Directory

In this scenario, AWS Directory Service (AD Connector) is used for all user or MFA authentication that is proxied through the AD Connector to the customer on-premises AD DS (Figure 5). For details on the protocols or encryption used for the authentication process, see the Security section of this whitepaper.

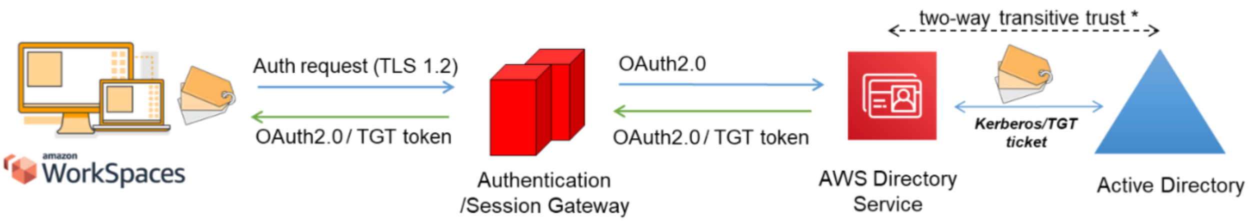


Figure 5: User Authentication via the Authentication Gateway

Scenario 1 shows a hybrid architecture where the customer might already have resources in AWS, as well as resources in an on-premises data center that could be accessed via Amazon WorkSpaces. The customer can leverage their existing on-premises AD DS and RADIUS servers for user and MFA authentication.

This architecture uses the following components or constructs.

AWS:

- Amazon VPC: Creation of an Amazon VPC with at least two private subnets across two Availability Zones.
- DHCP Options Set: Creation of an Amazon VPC DHCP Options Set. This allows customer-specified domain name and domain name servers (DNS) (on-premises services) to be defined. (For more information, see DHCP Options Sets.)
- Amazon virtual private gateway: Enable communication with your own network over an IPsec VPN tunnel or an AWS Direct Connect connection.
- AWS Directory Service: AD Connector is deployed into a pair of Amazon VPC private subnets.
- Amazon WorkSpaces: WorkSpaces are deployed in the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Network connectivity:** corporate VPN or Direct Connect endpoints.
- **AD DS:** corporate AD DS.
- **MFA (optional):** corporate RADIUS server.
- **End user devices:** Corporate or BYOL end user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

Although this solution is great for customers who don't want to deploy AD DS into the cloud, it does come with some caveats:

- **Reliance on connectivity:** If connectivity to the data center is lost, users cannot log in to their respective WorkSpaces, and existing connections will remain active for the Kerberos/TGT lifetime.
- **Latency:** If latency exists via the connection (this is more the case with VPN than Direct Connect), then WorkSpaces authentication and any AD DS-related activity, such as Group Policy (GPO) enforcement, will take more time.
- **Traffic costs:** All authentication must traverse the VPN or Direct Connect link, and so it depends on the connection type. This is either Data Transfer Out from Amazon EC2 to internet or Data Transfer Out (Direct Connect).

> **Note:** AD Connector is a proxy service. It doesn't store or cache user credentials. Instead, all authentication, lookup, and management requests are handled by your Active Directory. An account with delegation privileges is required in your directory service with rights to read all user information and join a computer to the domain.

For details about how to configure a user in the customer directory for AD Connector, see Delegating Connect Privileges.

In general, the WorkSpaces experience is highly dependent on item 5 shown in Figure 4. For this scenario, the WorkSpaces authentication experience is highly dependent on the network link between the customer Active Directory and the WorkSpaces VPC. The customer should ensure the link is highly available.

Scenario 2: Extending On-Premises AD DS into AWS (Replica)

This scenario is similar to scenario 1. However, in this scenario, a replica of the customer AD DS is deployed on AWS in combination with AD Connector. This reduces latency of authentication or query requests to AD DS. Figure 6 shows a high-level view of each of the components and the user authentication flow.

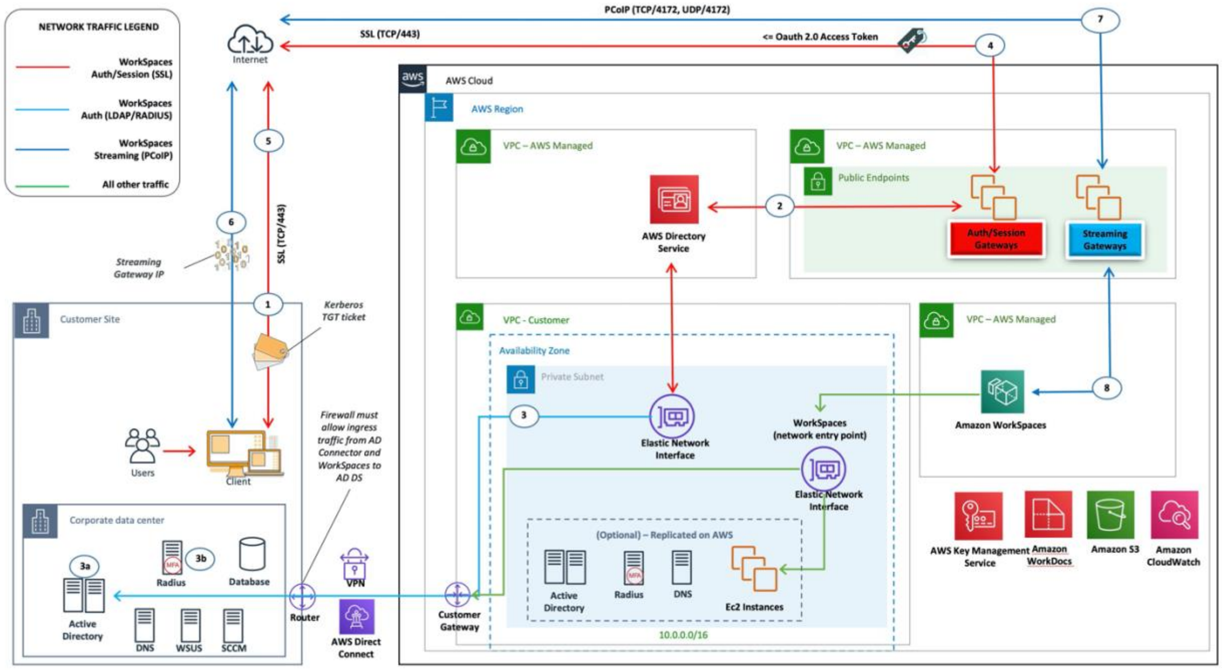


Figure 6: Extend customer Active Directory Domain to the cloud

As in scenario 1, AD Connector is used for all user or MFA authentication, which in turn is proxied to the customer AD DS (Figure 5). In this scenario, the customer AD DS is deployed across Availability Zones on Amazon EC2 instances that are promoted to be domain controllers in the customer's on-premises Active Directory forest, running in the AWS Cloud. Each domain controller is deployed into VPC private subnets to make AD DS highly available in the AWS Cloud. For best practices for deploying AD DS in the AWS Cloud, see Design Considerations later in this whitepaper.

After WorkSpaces instances are deployed, they have access to the cloud-based domain controllers for secure, low-latency directory services and DNS. All network traffic, including AD DS communication, authentication requests, and Active Directory replication, is secured either within the private subnets or across the customer VPN tunnel or Direct Connect.

This architecture uses the following components or constructs.

AWS:

- **Amazon VPC:** Creation of an Amazon VPC with at least four private subnets across two Availability Zones (two for the customer AD DS, two for AD Connector or Amazon WorkSpaces).

- **DHCP Options Set:** Creation of an Amazon VPC DHCP options set. This allows the customer to define a specified domain name and DNSs (AD DS local). For more information, see DHCP Options Sets.
- **Amazon virtual private gateway:** Enable communication with a customer-owned network over an IPsec VPN tunnel or AWS Direct Connect connection.
- **Amazon EC2:**
  - Customer corporate AD DS domain controllers deployed on Amazon EC2 instances in dedicated private VPC subnets.
  - Customer "optional" RADIUS servers for MFA on Amazon EC2 instances in dedicated private VPC subnets.
- **AWS Directory Services**: AD Connector is deployed into a pair of Amazon VPC private subnets.
- **Amazon WorkSpaces:** WorkSpaces are deployed into the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Network connectivity:** Corporate VPN or AWS Direct Connect endpoints.
- **AD DS:** Corporate AD DS (required for replication).
- **MFA "optional":** Corporate RADIUS server.
- **End user devices:** Corporate or BYOL end user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

Unlike scenario 1, this solution doesn't have the same caveats. Therefore, WorkSpaces and AWS Directory Service have no reliance on the connectivity in place.

- **Reliance on connectivity:** If connectivity to the customer data center is lost, end users can continue to work because authentication and "optional" MFA are processed locally.
- **Latency:** With the exception of replication traffic (see *Design Considerations:* AD DS Sites and Services), all authentication is local and low latency.
- **Traffic costs:** In this scenario, authentication is local, with only AD DS replication having to traverse the VPN or Direct Connect link, reducing data transfer.

In this scenario, the WorkSpaces authentication experience is not highly dependent on the network link between the customer Active Directory as the AD Directory Services are available in AWS. This becomes even more the case when a customer wants to scale WorkSpaces to thousands of desktops, especially in relation to AD DS global catalog queries, as this traffic remains local to the WorkSpaces environment.

Scenario 3: Standalone Isolated Deployment Using AWS Directory Service in the AWS Cloud

This scenario, shown in Figure 7, has AD DS deployed in the AWS Cloud in a standalone isolated environment. AWS Directory Service is used exclusively in this scenario. Instead of fully managing AD DS, customers can rely on AWS Directory Service for tasks such as building a highly available directory topology, monitoring domain controllers, and configuring backups and snapshots.

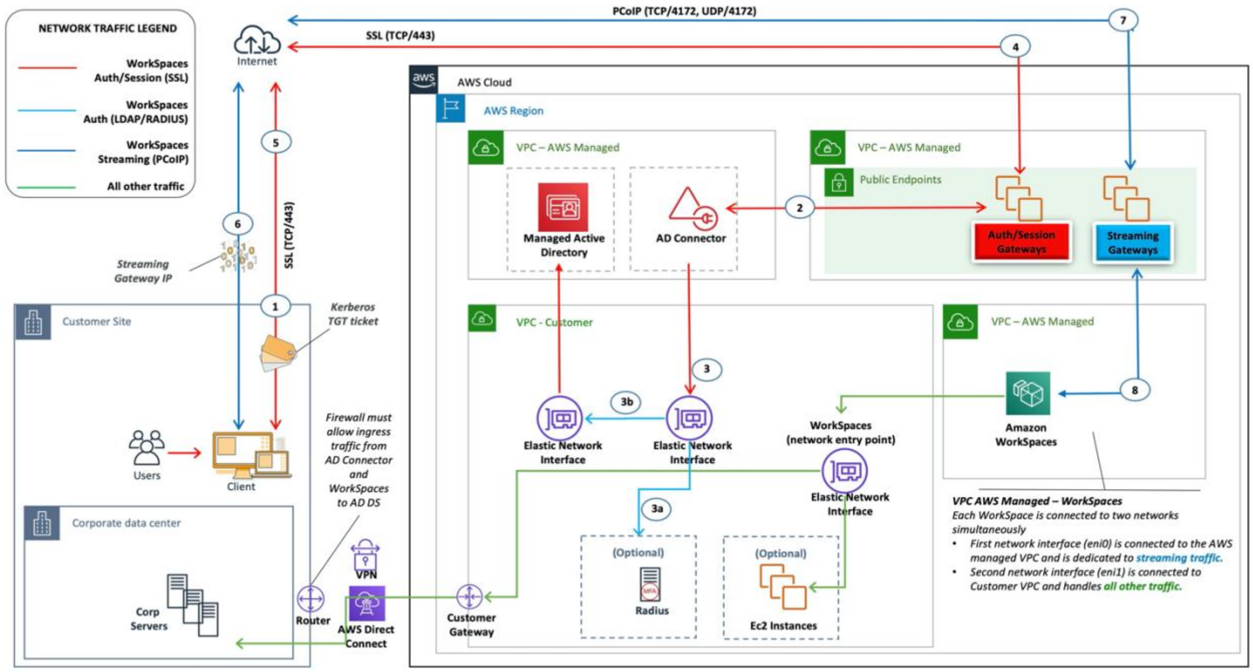

NETWORK TRAFFIC LEGEND
WorkSpaces Auth/Session (SSL)
WorkSpaces Auth (LDAP/RADIUS)
WorkSpaces Streaming (PCoIP)
All other traffic
Internet
PCoIP (TCP/4172, UDP/4172)
SSL (TCP/443)
SSL (TCP/443)
1
2
3
3a
3b
4
5
6
7
8
Streaming Gateway IP
Kerberos TGT ticket
Customer Site
Users
Client
Corporate data center
Corp Servers
Firewall must allow ingress traffic from AD Connector and WorkSpaces to AD DS
Router
VPN
AWS Direct Connect
Customer Gateway
AWS Cloud
AWS Region
VPC – AWS Managed
Managed Active Directory
AD Connector
VPC – AWS Managed
Public Endpoints
Auth/Session Gateways
Streaming Gateways
VPC - Customer
Elastic Network Interface
Elastic Network Interface
WorkSpaces (network entry point)
Elastic Network Interface
(Optional)
Radius
(Optional)
Ec2 Instances
VPC – AWS Managed
Amazon WorkSpaces
VPC AWS Managed – WorkSpaces
Each WorkSpace is connected to two networks simultaneously
• First network interface (eni0) is connected to the AWS managed VPC and is dedicated to streaming traffic.
• Second network interface (eni1) is connected to Customer VPC and handles all other traffic.

Figure 7: Cloud only — AWS Directory Services (Microsoft AD)

As in scenario 2, the AD DS (Microsoft AD) is deployed into dedicated subnets that span two Availability Zones, making AD DS highly available in the AWS Cloud. In addition to Microsoft AD, AD Connector (in all three scenarios) is deployed for WorkSpaces authentication or MFA. This ensures separation of roles or functions within the Amazon VPC, which is a standard best practice (see *Design Considerations*: Partitioned Network section).

Scenario 3 is a standard all-in configuration that works well for customers who want to have AWS manage the deployment, patching, high availability, and monitoring of the AWS Directory Service. The scenario also works well for proof of concepts, lab, and production environments because of its isolation mode.

In addition to the placement of AWS Directory Service, Figure 7 shows the flow of traffic from a user to a workspace and how the workspace interacts with the AD server and MFA server.

This architecture uses the following components or constructs.

AWS:

- **Amazon VPC:** Creation of an Amazon VPC with at least four private subnets across two Availability Zones (two for AD DS *Microsoft AD*, two for AD Connector or WorkSpaces). "*Separation of roles.*"
- **DHCP options set:** Creation of an Amazon VPC DHCP options set. This allows a customer to define a specified domain name and DNS(s) (Microsoft AD). For more information, see DHCP Options Sets.
- **Optional: Amazon virtual private gateway:** Enable communication with a customer-owned network over an IPsec VPN tunnel (VPN) or AWS Direct Connect connection. Use for accessing on-premises back-end systems.
- **AWS Directory Service:** Microsoft AD deployed into a dedicated pair of VPC subnets (AD DS Managed Service).
- **Amazon EC2:** Customer "Optional" RADIUS Servers for MFA.
- **AWS Directory Services**: AD Connector is deployed into a pair of Amazon VPC private subnets.
- **Amazon WorkSpaces:** WorkSpaces are deployed into the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Optional: Network Connectivity:** corporate VPN or AWS Direct Connect endpoints.

- **End user devices:** Corporate or BYOL end-user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

Like scenario 2, this scenario doesn't have issues with reliance on connectivity to the customer on-premises data center, latency, or data out transfer costs (except where internet access is enabled for WorkSpaces within the VPC) because, by design, this is an isolated or cloud-only scenario.

Scenario 4: AWS Microsoft AD and a Two-Way Transitive Trust to On-Premises

This scenario, shown in Figure 8, has AWS Managed AD deployed in the AWS Cloud, which has a two-way transitive trust to the customer on-premises Active Directory. User accounts and WorkSpaces are created in the Managed AD, with the Active Directory trust enabling resources to be accessed in the on-premises environment.

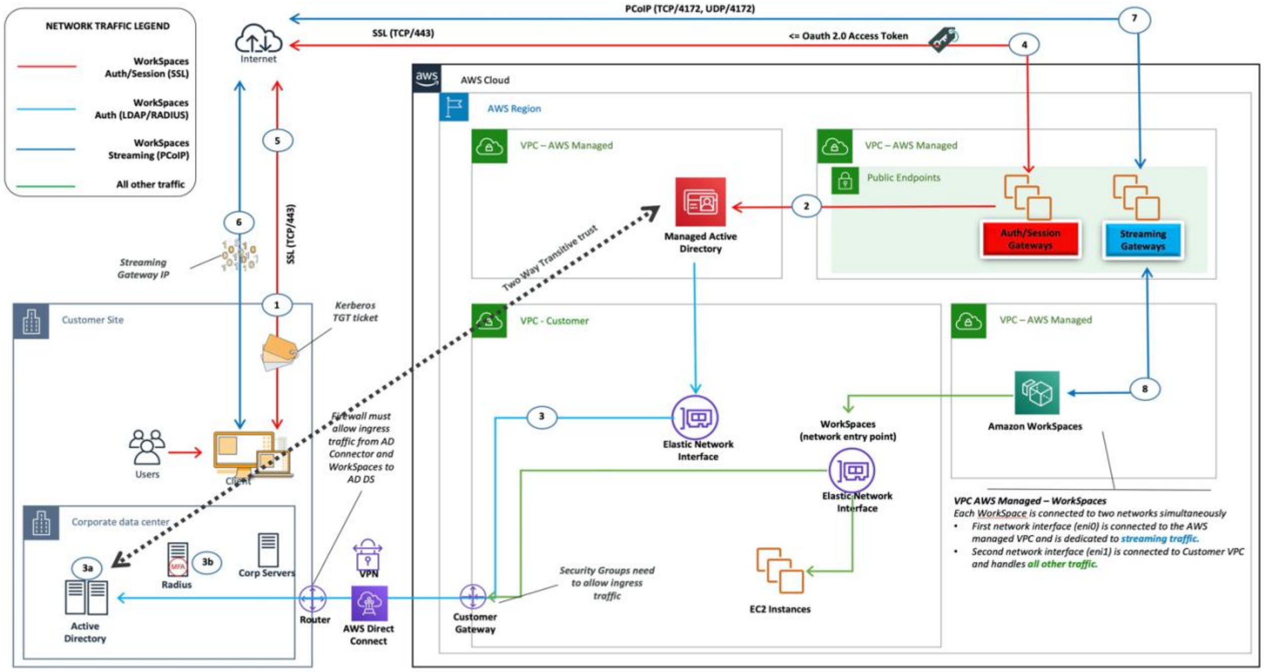

NETWORK TRAFFIC LEGEND
WorkSpaces Auth/Session (SSL)
WorkSpaces Auth (LDAP/RADIUS)
WorkSpaces Streaming (PCoIP)
All other traffic
Internet
PCoIP (TCP/4172, UDP/4172)
SSL (TCP/443)
<= Oauth 2.0 Access Token
7
4
5
6
SSL (TCP/443)
Streaming Gateway IP
1
Kerberos TGT ticket
Customer Site
Users
Client
Firewall must allow ingress traffic from AD Connector and WorkSpaces to AD DS
Corporate data center
3a
3b
MFA
Radius
Corp Servers
Active Directory
Router
VPN
AWS Direct Connect
Customer Gateway
Security Groups need to allow ingress traffic
AWS Cloud
AWS Region
VPC – AWS Managed
Two Way Transitive trust
Managed Active Directory
2
VPC – AWS Managed
Public Endpoints
Auth/Session Gateways
Streaming Gateways
VPC - Customer
3
Elastic Network Interface
WorkSpaces (network entry point)
Elastic Network Interface
EC2 Instances
VPC – AWS Managed
Amazon WorkSpaces
8
VPC AWS Managed – WorkSpaces
Each WorkSpace is connected to two networks simultaneously
• First network interface (eni0) is connected to the AWS managed VPC and is dedicated to streaming traffic.
• Second network interface (eni1) is connected to Customer VPC and handles all other traffic.

Figure 8: AWS Microsoft AD and a two-way transitive trust to on-premises

As in scenario 3, the AD DS (Microsoft AD) is deployed into dedicated subnets that span two Availability Zones, making AD DS highly available in the AWS Cloud.

This scenario works well for customers who want to have a fully managed AWS Directory Service, including deployment, patching, high availability, and monitoring of their AWS Cloud. This scenario also allows WorkSpaces users to access AD-joined resources on their existing networks. This scenario requires a domain trust to be in place. Security groups and firewall rules need to allow communication between the two active directories.

In addition to the placement of AWS Directory Service, Figure 8 shows the flow of traffic from a user to a workspace and how the workspace interacts with the AD server and MFA server.

This architecture uses the following components or construct.

AWS:

- **Amazon VPC:** Creation of an Amazon VPC with at least four private subnets across two Availability Zones (two for AD DS *Microsoft AD*, two for AD Connector or WorkSpaces). "*Separation of roles.*"
- **DHCP options set:** Creation of an Amazon VPC DHCP options set. This allows a customer to define a specified domain name and DNS(s) (Microsoft AD). For more information, see DHCP Options Sets.
- **Optional: Amazon virtual private gateway:** Enable communication with a customer-owned network over an IPsec VPN tunnel (VPN) or AWS Direct Connect connection. Use for accessing on-premises back-end systems.
- **AWS Directory Service:** Microsoft AD deployed into a dedicated pair of VPC subnets (AD DS Managed Service).
- **Amazon EC2:** Customer "Optional" RADIUS Servers for MFA.
- **Amazon WorkSpaces:** WorkSpaces are deployed into the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Network Connectivity:** corporate VPN or AWS Direct Connect endpoints.
- **End user devices:** Corporate or BYOL end-user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

This solution requires connectivity to the customer on-premises data center to allow the trust process to operate. If WorkSpaces users are using resources on the on-premises network, then latency and outbound data transfer costs need to be considered.

Scenario 5: AWS Microsoft AD using a Shared Services Virtual Private Cloud (VPC)

This scenario, shown in Figure 9, has an AWS Managed AD deployed in the AWS Cloud, providing authentication services for workloads that are either already hosted in AWS or are planned to be as part of a broader migration. The best practice recommendation is to have Amazon WorkSpaces in a dedicated VPC. Customers should also create a specific Active Directory organizational unit (OU) to organize the WorkSpaces computer objects.

To deploy WorkSpaces with a shared services VPC hosting Managed Active Directory, deploy an Active Directory Connector with an ADC service account created in the Managed AD. The service account requires permissions to create computer objects in the WorkSpaces designated OU in the shared services Managed AD.

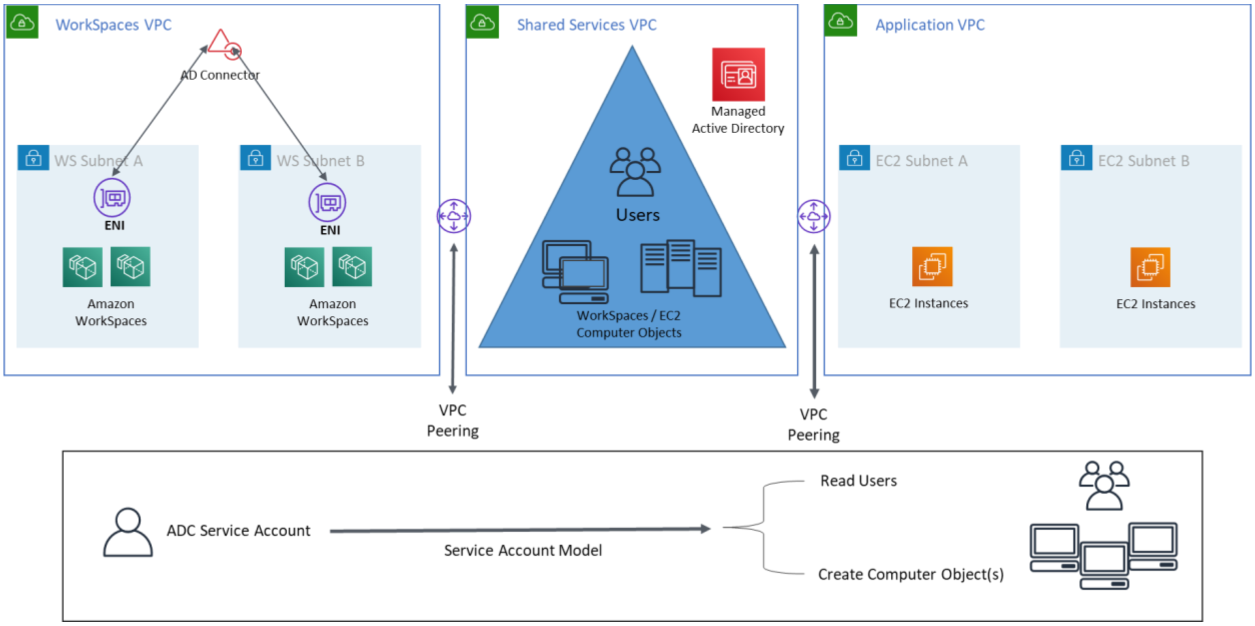


Figure 9: AWS Microsoft AD using a Shared Services Virtual Private Cloud (VPC)

This architecture uses the following components or constructs.

AWS:

- **Amazon VPC:** Creation of an Amazon VPC with at least two private subnets across two Availability Zones (two for AD Connector and WorkSpaces).
- **DHCP options set:** Creation of an Amazon VPC DHCP options set. This allows a customer to define a specified domain name and DNS(s) (Microsoft AD). For more information, see DHCP Options Sets.
- **Optional: Amazon virtual private gateway:** Enable communication with a customer-owned network over an IPsec VPN tunnel (VPN) or AWS Direct Connect connection. Use for accessing on-premises back-end systems.
- **AWS Directory Service:** Microsoft AD deployed into a dedicated pair of VPC subnets (AD DS Managed Service), AD Connector
- **AWS Transit Gateway/VPC Peering:** enable connectivity between Workspaces VPC and the Shared Services VPC
- **Amazon EC2:** Customer “Optional” RADIUS Servers for MFA.
- **Amazon WorkSpaces:** WorkSpaces are deployed into the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Network Connectivity:** corporate VPN or AWS Direct Connect endpoints.
- **End user devices:** Corporate or BYOL end-user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

Scenario 6: AWS Microsoft AD, Shared Services VPC, and a One-Way Trust to On Premises

This scenario, as shown in Figure 10, uses an existing on-premises Active Directory for user accounts, and introduces a separate Managed Active Directory in AWS Cloud to host the computer objects associated with the WorkSpaces.

This scenario allows the computer objects and Active Directory group policies to be managed independently from the corporate Active Directory. This scenario is useful when a third party wants to manage WorkSpaces on a customer’s behalf as it allows the third party to define and control the WorkSpaces and policies associated with them, without a need to grant the third-party access to the customer AD. In this scenario, a specific Active Directory organizational unit (OU) is created to organize the WorkSpaces computer objects in the Shared Services AD. To deploy WorkSpaces with the computer objects created in the Shared Services VPC hosting Managed Active Directory using user accounts from the customer domain, deploy an Active Directory Connector referencing the corporate AD. Use an ADC Service Account created in the corporate AD that has permissions to create computer objects in the OU that was configured for WorkSpaces in the Shared Services Managed AD.

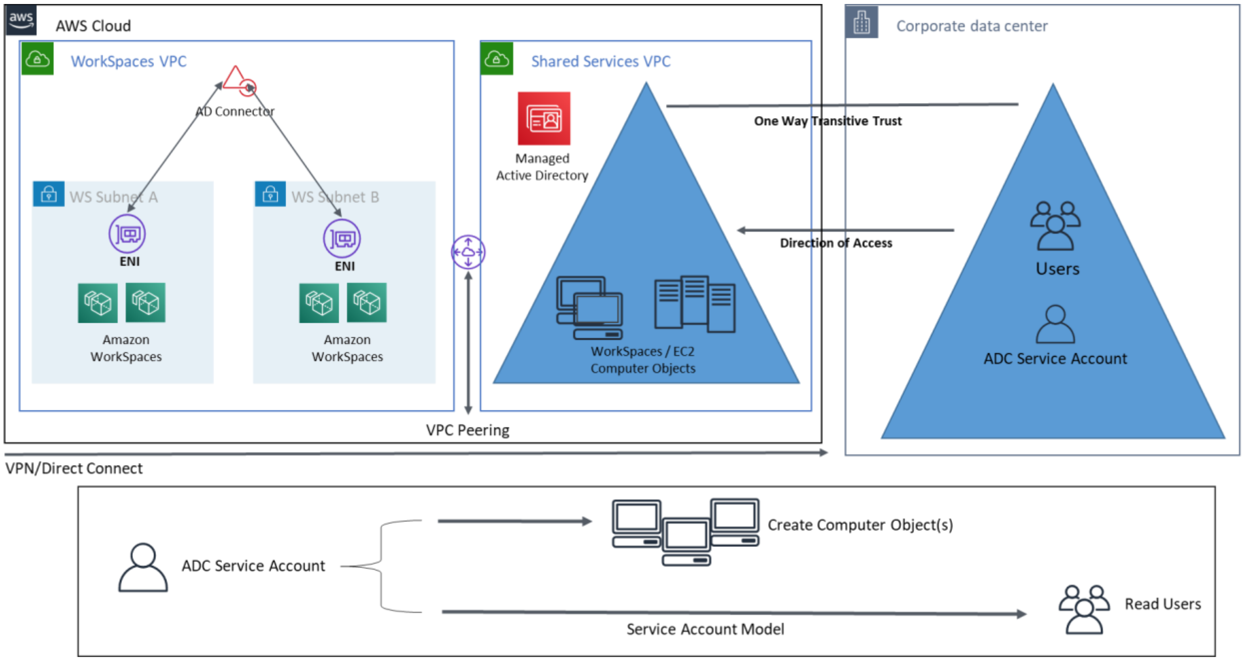


Figure 10: AWS Microsoft, Shared Services VPC and a one-way trust to AD on-premises

This architecture uses the following components or constructs.

AWS:

- **Amazon VPC:** Creation of an Amazon VPC with at least two private subnets across two Availability Zones (two for AD Connector and WorkSpaces).
- **DHCP options set:** Creation of an Amazon VPC DHCP options set. This allows a customer to define a specified domain name and DNS(s) (Microsoft AD). For more information, see DHCP Options Sets.
- **Optional: Amazon virtual private gateway:** Enable communication with a customer-owned network over an IPsec VPN tunnel (VPN) or AWS Direct Connect connection. Use for accessing on-premises back-end systems.
- **AWS Directory Service:** Microsoft AD deployed into a dedicated pair of VPC subnets (AD DS Managed Service), AD Connector.

- **Transit Gateway/VPC Peering:** enable connectivity between Workspaces VPC and the Shared Services VPC.
- **Amazon EC2:** Customer “Optional” RADIUS Servers for MFA.
- **Amazon WorkSpaces:** WorkSpaces are deployed into the same private subnets as the AD Connector (see Design Considerations, AD Connector).

Customer:

- **Network Connectivity:** corporate VPN or AWS Direct Connect endpoints.
- **End user devices:** Corporate or BYOL end-user devices (such as Windows, Macs, iPads, Android tablets, zero clients, and Chromebooks) used to access the Amazon WorkSpaces service (see Supported Platforms and Devices).

Design Considerations

A functional AD DS deployment in the AWS Cloud requires a good understanding of both Active Directory concepts and specific AWS services. In this section, we discuss key design considerations when deploying AD DS for Amazon WorkSpaces, VPC best practices for AWS Directory Service, DHCP and DNS requirements, AD Connector specifics, and Active Directory sites and services.

VPC Design

As we discussed in the Network Considerations section of this document and documented earlier for scenarios 2 and 3, customers should deploy AD DS in the AWS Cloud into a dedicated pair of private subnets, across two Availability Zones, and separated from AD Connector or WorkSpaces subnets. This construct provides highly available, low latency access to AD DS services for WorkSpaces, while maintaining standard best practices of separation of roles or functions within the Amazon VPC.

Figure 11 shows the separation of AD DS and AD Connector into dedicated private subnets (scenario 3). In this example all services reside in the same Amazon VPC.

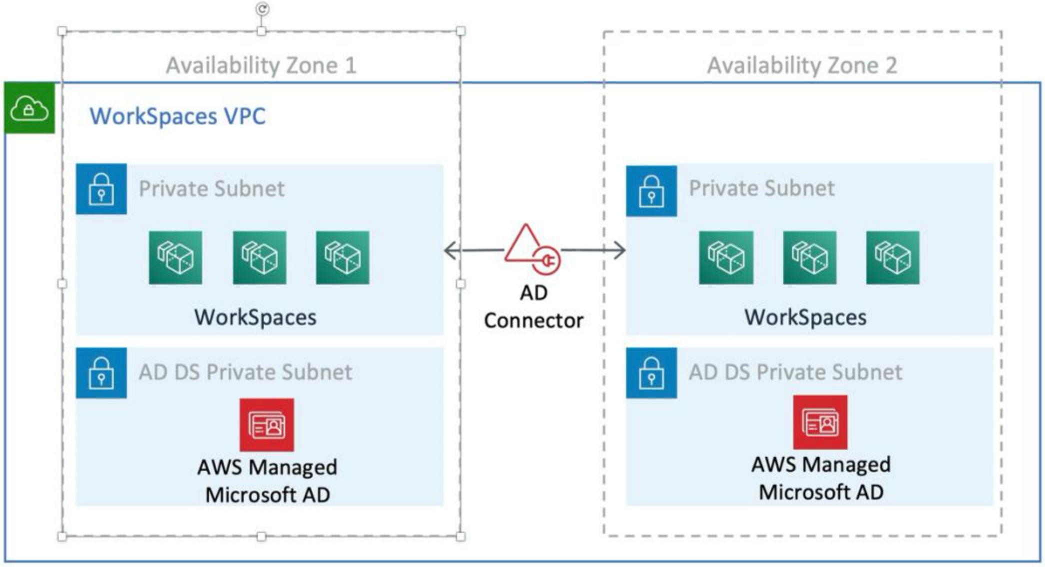


Figure 11: AD DS network segregation

Figure 12 shows a design similar to scenario 1, however, in this scenario the on-premises portion resides in a dedicated Amazon VPC.

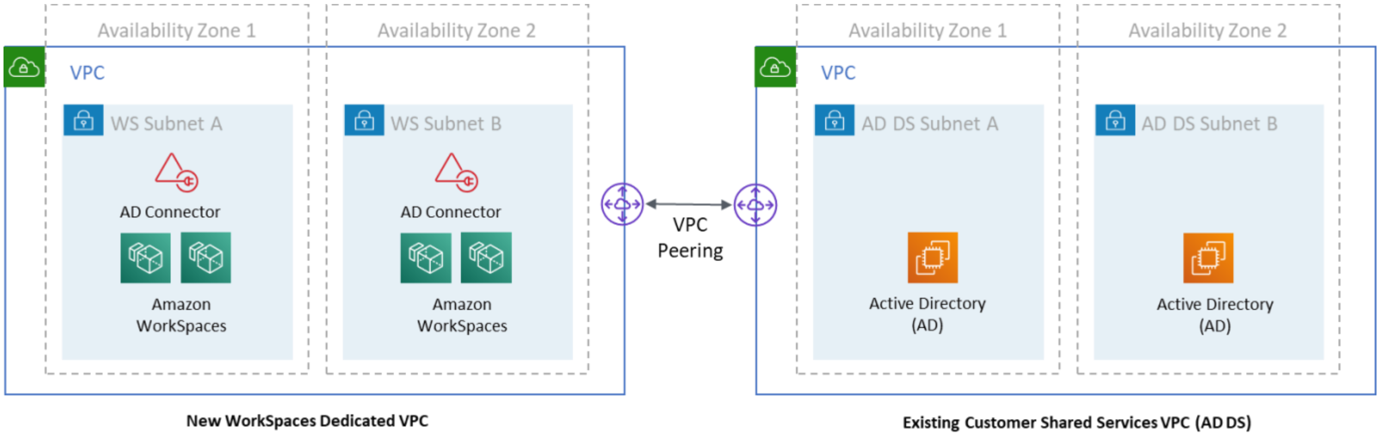


Figure 12: Dedicated WorkSpaces VPC

Note: For customers who have an existing AWS deployment where AD DS is being used, we recommend that customers locate their WorkSpaces in a dedicated VPC and use VPC peering for AD DS communications.

In addition to the creation of dedicated private subnets for AD DS, domain controllers and member servers require several Security Group rules to allow traffic for services, such as AD DS replication, user authentication, Windows Time services, and distributed file system (DFS).

> **Note:** Best practice is to restrict the required security group rules to the WorkSpaces private subnets and, in the case of scenario 2, allow for bidirectional AD DS communications on-premises to and from the AWS Cloud, as shown in the following table.

Protocol	Port	Use	Destination
tcp	53, 88, 135, 139, 389, 445, 464, 636	Auth (primary)	Active Directory (private data center or Amazon EC2)*
tcp	49152 – 65535	RPC High Ports	Active Directory (private data center or Amazon EC2)**
tcp	3268-3269	Trusts	Active Directory (private data center or Amazon EC2)*
tcp	9389	Remote Microsoft Windows PowerShell (optional)	Active Directory (private data center or Amazon EC2)*
udp	53, 88, 123, 137, 138, 389, 445, 464	Auth (primary)	Active Directory (private data center or Amazon EC2)*
udp	1812	Auth (MFA) (optional)	RADIUS (private data center or Amazon EC2)*

*See Active Directory and Active Directory Domain Services Port Requirements

**See Service overview and network port requirements for Windows

For step-by-step guidance for implementing rules, see Adding Rules to a Security Group in the *Amazon Elastic Compute Cloud User Guide*.

VPC Design: DHCP and DNS

With an Amazon VPC, DHCP services are provided by default for your instances. By default, every VPC provides an internal DNS server that is accessible via the Classless Inter-Domain Routing (CIDR) +2 address space and is assigned to all instances via a default DHCP options set.

DHCP options sets are used within an Amazon VPC to define scope options, such as the domain name or the name servers that should be handed to customer instances via DHCP. Correct functionality of Windows services within a customer VPC depends on this DHCP scope option. In each of the scenarios defined earlier, customers create and assign their own scope that defines the domain name and name servers. This ensures that domain-joined Windows instances or WorkSpaces are configured to use the Active Directory DNS.

The following table is an example of a custom set of DHCP scope options that must be created for Amazon WorkSpaces and AWS Directory Services to function correctly.

Parameter	Value
Name tag	Creates a tag with key = **name** and **value** set to a specific string Example: example.com
Domain name	example.com
Domain name servers	DNS server address, separated by commas Example: 192.0.2.10, 192.0.2.21

NTP servers	Leave this field blank
NetBIOS name servers	Enter the same comma separated IPs as per domain name servers Example: 192.0.2.10, 192.0.2.21
NetBIOS node type	2

For details on creating a custom DHCP option set and associating it with an Amazon VPC, see Working with DHCP Options Sets in the *Amazon Virtual Private Cloud User Guide*.

In scenario 1, the DHCP scope would be the on-premises DNS or AD DS. However, in scenarios 2 or 3, this would be the locally deployed directory service (AD DS on Amazon EC2 or AWS Directory Services: Microsoft AD). We recommend each domain controller that resides in the AWS Cloud be a global catalog and Directory-Integrated DNS server.

Active Directory: Sites and Services

For scenario 2, sites and services are critical components for the correct function of AD DS. Site topology controls Active Directory replication between domain controllers within the same site and across site boundaries. In scenario 2, at least two sites are present: on premises and the Amazon WorkSpaces in the cloud.

Defining the correct site topology ensures client affinity, meaning that clients (in this case, WorkSpaces) use their preferred local domain controller.

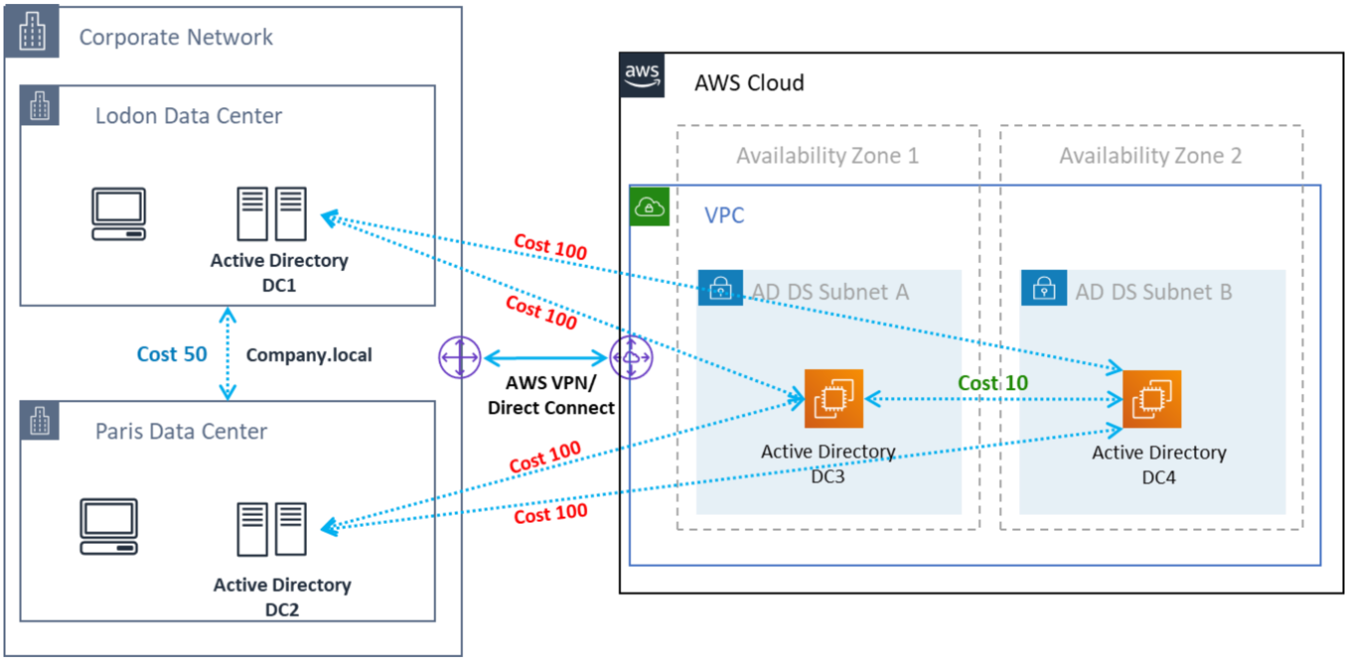


Figure 13: Active Directory sites and services: client affinity

Best practice: Define high cost for site links between on-premises AD DS and the AWS Cloud. Figure 10 is an example of what costs to assign to the site links (cost 100) to ensure site-independent client affinity.

These associations help ensure that traffic—such as AD DS replication, and client authentication—uses the most efficient path to a domain controller. In the case of scenarios 2 and 3, this helps ensure lower latency and cross-link traffic.

Multi-Factor Authentication (MFA)

Implementing MFA requires the Amazon WorkSpaces infrastructure to use AD Connector as its AWS Directory Service and have a RADIUS server. Although this document doesn’t discuss the deployment of a RADIUS server, the previous section, AD DS Deployment Scenarios describes the placement of RADIUS within each scenario.

MFA – Two-Factor Authentication

Amazon WorkSpaces supports MFA through AWS Directory Service: AD Connector and a *customer owned* RADIUS server. Once enabled, users are required to provide

Username, **Password**, and **MFA Code** to the WorkSpaces client for authentication to their respective WorkSpaces desktops.

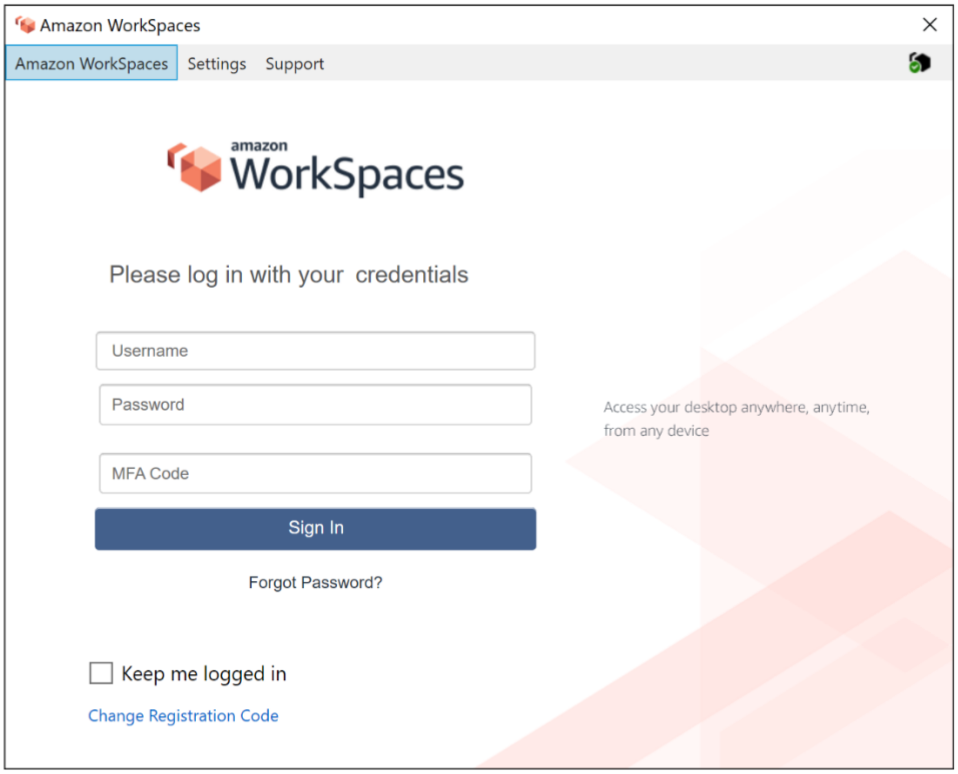


Figure 14: WorkSpaces client with MFA enabled

Hard rule: Implementing MFA authentication requires customers to use AD Connector. AD Connector doesn’t support selective “per user” MFA, as this is a global per AD Connector setting. If selective “per user” MFA is required, users must be separated by an AD Connector.

WorkSpaces MFA requires one or more RADIUS servers. Typically, these are existing solutions, for example, RSA, or the servers can be deployed within a VPC (see AD DS Deployment Scenarios). If deploying a new RADIUS solution, several implementations exist, such as FreeRADIUS, and cloud services, such as Duo Security.

For a list of prerequisites to implement MFA with Amazon WorkSpaces, see the *Amazon WorkSpaces Administration Guide*, Preparing Your Network for an AD Connector Directory. The process for configuring AD Connector for MFA is described in

Managing an AD Connector Directory: Multi-factor Authentication, in the *Amazon WorkSpaces Administration Guide*.

Security

This section explains how to secure data by using encryption when using Amazon WorkSpaces services. We describe encryption in transit and at rest, and the use of security groups to protect network access to the WorkSpaces. This section also provides information on how to control end device access to WorkSpaces by using Trusted Devices, and IP Access Control Groups.

Additional information on authentication (including MFA support) in the AWS Directory Service can be found in this section.

Encryption in Transit

Amazon WorkSpaces uses cryptography to protect confidentiality at different stages of communication (in transit) and also to protect data at rest (encrypted WorkSpaces). The processes in each stage of the encryption used by Amazon WorkSpaces in transit is described in the following sections.

For information about the encryption at rest, see the Encrypted WorkSpaces section later in this whitepaper.

Registration and Updates

The desktop client application communicates with Amazon for updates and registration using HTTPS.

Authentication Stage

The desktop client initiates authentication by sending credentials to the authentication gateway. The communication between the desktop client and authentication gateway uses HTTPS. At the end of this stage, if the authentication succeeds, the authentication gateway returns an OAuth 2.0 token to the desktop client, through the same HTTPS connection.

> **Note:** The desktop client application supports the use of a proxy server for port 443 (HTTPS) traffic, for updates, registration, and authentication

After receiving the credentials from the client, the authentication gateway sends an authentication request to AWS Directory Service. The communication from thje authentication gateway to AWS Directory Service takes place over HTTPS, so no user credentials are transmitted in plaintext.

Authentication — AD Connector

AD Connector uses Kerberos to establish authenticated communication with on-premises AD, so it can bind to LDAP and execute subsequent LDAP queries. The AWS Directory Service also supports LDAP with TLS (LDAPs). No user credentials are transmitted in plaintext at any time. For increased security, it is possible to connect a WorkSpaces VPC with the on-premises network (where AD resides) using a VPN connection. When using an AWS hardware VPN connection, customers can set up encryption in transit by using standard IPSEC (IKE and IPSEC SAs) with AES-128 or AES-256 symmetric encryption keys, SHA-1 or SHA-256 for integrity hash, and DH groups (2,14-18, 22, 23 and 24 for phase 1; 1,2,5, 14-18, 22, 23 and 24 for phase 2) using PFS.

Broker Stage

After receiving the OAuth 2.0 token (from the authentication gateway, if the authentication succeeded), the desktop client will query Amazon WorkSpaces services (Broker Connection Manager) using HTTPS. The desktop client authenticates itself by sending the OAuth 2.0 token and, as a result, the client will receive the endpoint information of the WorkSpaces streaming gateway.

Streaming Stage

The desktop client requests to open a PCoIP session with the streaming gateway (using the OAuth 2.0 token). This session is AES-256 encrypted and uses the PCoIP port for communication control (that is, 4172/tcp).

Using the OAuth2.0 token, the streaming gateway requests the user-specific WorkSpaces information from the Amazon WorkSpaces service, over HTTPS.

The streaming gateway also receives the TGT from the client (which is encrypted using the client user’s password) and, by using Kerberos TGT pass-through, the gateway initiates a Windows login on the WorkSpace, using the user’s retrieved Kerberos TGT.

The WorkSpace then initiates an authentication request to the configured AWS Directory Service, using standard Kerberos authentication.

After the WorkSpace is successfully logged in, the PCoIP streaming starts. The connection is initiated by the client on port tcp 4172 with the return traffic on port udp 4172. Additionally, the initial connection between the streaming gateway and a WorkSpaces desktop over the management interface is via UDP 55002. (See the Amazon Workspaces documentation, Amazon WorkSpaces Details. The initial outbound UDP port is 55002.) The streaming connection, using ports 4172 (tcp and udp), is encrypted by using AES 128- and 256-bit ciphers, but default to 128-bit. Customers can actively change this to 256-bit either using PCoIP-specific Active Directory Group Policy settings (pcoip.adm) for Windows WorkSpaces, or with the pcoip-agent.conf file for Amazon Linux WorkSpaces.

Network Interfaces

Each Amazon WorkSpace has two network interfaces, called the primary network interface and management network interface.

The primary network interface provides connectivity to resources inside the customer VPC, such as access to AWS Directory Service, the internet, and the customer corporate network. It is possible to attach security groups to this primary network interface. Conceptually, we differentiate the security groups attached to this ENI based on the scope of the deployment: WorkSpaces security group and ENI security groups.

Management Network Interface

The management network interface cannot be controlled via security groups, however, customers can use a host-based firewall on WorkSpaces to block ports or control access. We don’t recommend applying restrictions on the management network interface. If a customer decides to add host-based firewall rules to manage this interface, a few ports should be open so the Amazon WorkSpaces service can manage the health and accessibility to the WorkSpace as defined in the Amazon WorkSpaces Administration Guide.

WorkSpaces Security Group

A default security group is created per AWS Directory Service and is automatically attached to all WorkSpaces that belong to that specific directory.

As with any other security group, it is possible to modify the rules of a WorkSpaces security group. The results take effect immediately after the changes are applied.

It is also possible to change the default WorkSpaces security group attached to an AWS

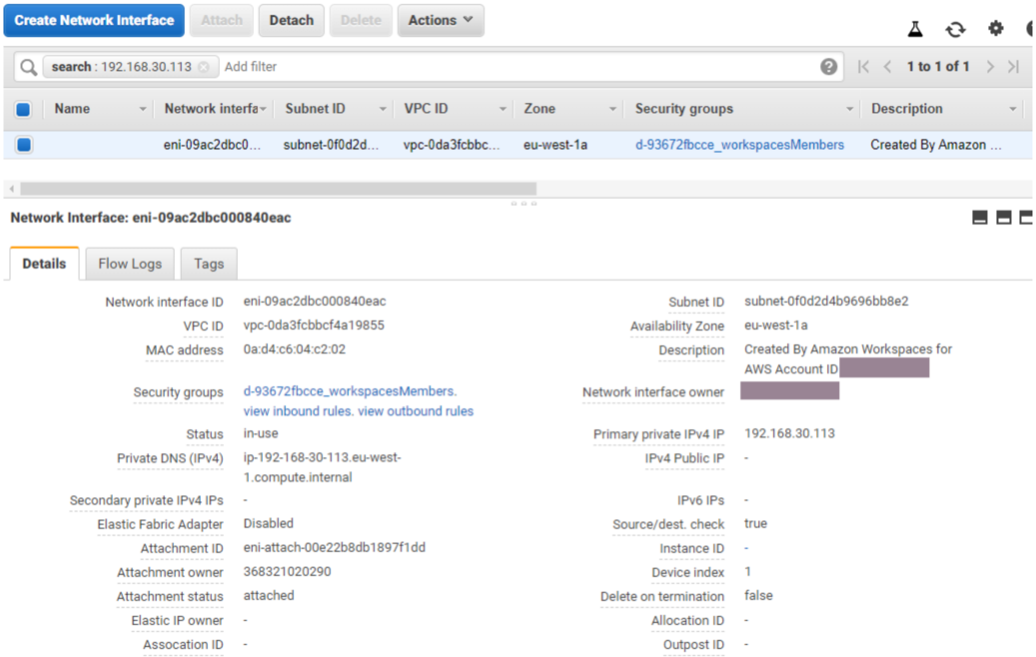
Directory Service by changing the WorkSpaces security group association.

> **Note:** A newly associated security group will be attached only to WorkSpaces created or rebuilt after the modification.

ENI Security Groups

Because the primary network interface is a regular ENI, it can be managed by using the different AWS management tools (see Elastic Network Interfaces (ENI). In particular, look for the WorkSpace IP address (in the WorkSpaces page in the Amazon WorkSpaces console), and then use that IP address as a filter to find the corresponding ENI (in the Network Interfaces section of the Amazon EC2 console).

Once the ENI is located, it can be directly managed by security groups. When manually assigning security groups to the primary network interface, consider the port requirements of Amazon WorkSpaces, as explained in Amazon WorkSpaces Details.



search : 192.168.30.113

☐	Name	Network interfa	Subnet ID	VPC ID	Zone	Security groups	Description
☐		eni-09ac2dbc0...	subnet-0f0d2d...	vpc-0da3fcbbc...	eu-west-1a	d-93672fbcce_workspacesMembers	Created By Amazon ...

Network Interface: eni-09ac2dbc000840eac

Network interface ID: eni-09ac2dbc000840eac
VPC ID: vpc-0da3fcbbcf4a19855
MAC address: 0a:d4:c6:04:c2:02
Security groups: d-93672fbcce_workspacesMembers. view inbound rules. view outbound rules
Status: in-use
Private DNS (IPv4): ip-192-168-30-113.eu-west-1.compute.internal
Secondary private IPv4 IPs: -
Elastic Fabric Adapter: Disabled
Attachment ID: eni-attach-00e22b8db1897f1dd
Attachment owner: 368321020290
Attachment status: attached
Elastic IP owner: -
Assocation ID: -

Subnet ID: subnet-0f0d2d4b9696bb8e2
Availability Zone: eu-west-1a
Description: Created By Amazon Workspaces for AWS Account ID [redacted]
Network interface owner: [redacted]
Primary private IPv4 IP: 192.168.30.113
IPv4 Public IP: -
IPv6 IPs: -
Source/dest. check: true
Instance ID: -
Device index: 1
Delete on termination: false
Allocation ID: -
Outpost ID: -

Figure 15: Managing security group associations

Encrypted WorkSpaces

Each Amazon WorkSpace is provisioned with a root volume (`C:` drive for Windows

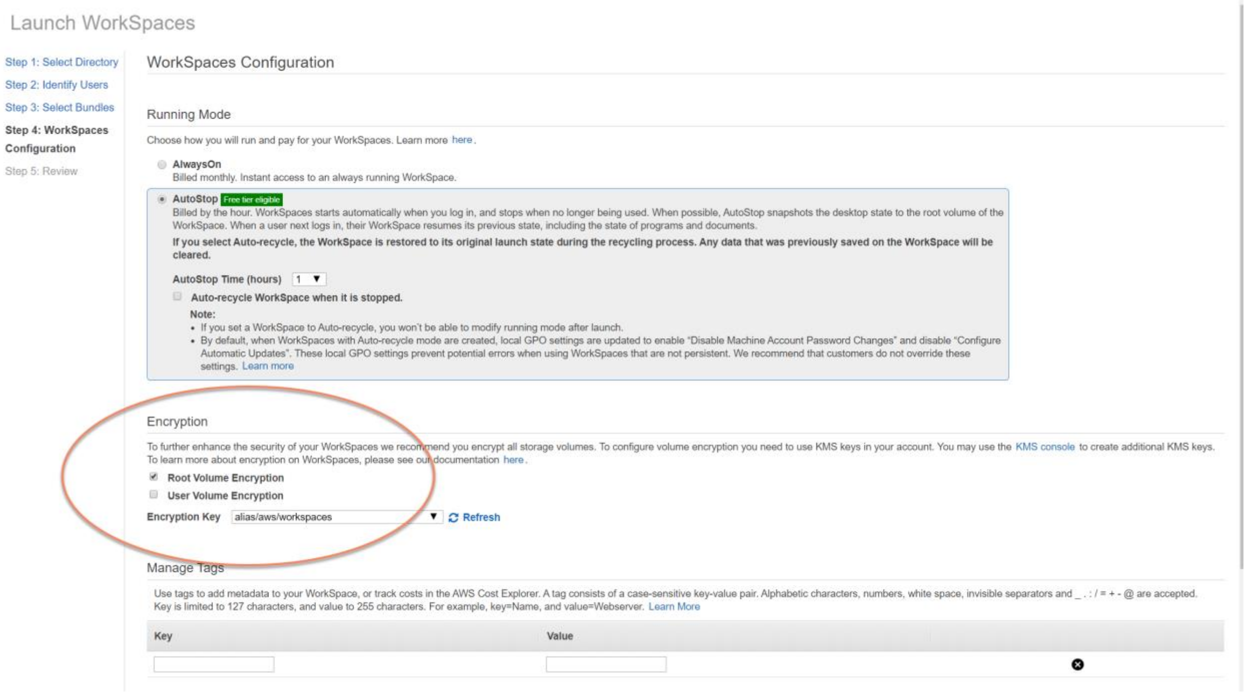
WorkSpaces, `root` for Amazon Linux WorkSpaces) and a user volume (`D:` drive for Windows WorkSpaces, `/home` for Amazon Linux WorkSpaces). The encrypted WorkSpaces feature enables one or both volumes to be encrypted.

What is Encrypted?

The data stored at rest, disk I/O to the volume, and snapshots created from encrypted volumes are all encrypted.

When Does Encryption Occur?

Encryption for a WorkSpace should be specified when launching (creating) the WorkSpace. WorkSpaces volumes can be encrypted only at launch time: after launch, the volume encryption status cannot be changed. Figure 13 shows the Amazon WorkSpaces console page for choosing encryption during the launching of a new WorkSpace.



Launch WorkSpaces

- Step 1: Select Directory
- Step 2: Identify Users
- Step 3: Select Bundles
- **Step 4: WorkSpaces Configuration**
- Step 5: Review

WorkSpaces Configuration

Running Mode

Choose how you will run and pay for your WorkSpaces. Learn more here.

○ **AlwaysOn**
Billed monthly. Instant access to an always running WorkSpace.

● **AutoStop** Free tier eligible
Billed by the hour. WorkSpaces starts automatically when you log in, and stops when no longer being used. When possible, AutoStop snapshots the desktop state to the root volume of the WorkSpace. When a user next logs in, their WorkSpace resumes its previous state, including the state of programs and documents.

If you select Auto-recycle, the WorkSpace is restored to its original launch state during the recycling process. Any data that was previously saved on the WorkSpace will be cleared.

AutoStop Time (hours) 1 ▼

□ **Auto-recycle WorkSpace when it is stopped.**

Note:

- If you set a WorkSpace to Auto-recycle, you won't be able to modify running mode after launch.
- By default, when WorkSpaces with Auto-recycle mode are created, local GPO settings are updated to enable "Disable Machine Account Password Changes" and disable "Configure Automatic Updates". These local GPO settings prevent potential errors when using WorkSpaces that are not persistent. We recommend that customers do not override these settings. Learn more

Encryption

To further enhance the security of your WorkSpaces we recommend you encrypt all storage volumes. To configure volume encryption you need to use KMS keys in your account. You may use the KMS console to create additional KMS keys. To learn more about encryption on WorkSpaces, please see our documentation here.

☑ **Root Volume Encryption**

□ **User Volume Encryption**

Encryption Key alias/aws/workspaces ▼ Refresh

Manage Tags

Use tags to add metadata to your WorkSpace, or track costs in the AWS Cost Explorer. A tag consists of a case-sensitive key-value pair. Alphabetic characters, numbers, white space, invisible separators and _ . : / = + - @ are accepted. Key is limited to 127 characters, and value to 255 characters. For example, key=Name, and value=Webserver. Learn More

Key	Value

Figure 16: Encrypting WorkSpace root volumes

How is a New WorkSpace Encrypted?

A customer can choose the Encrypted WorkSpaces option from either the Amazon WorkSpaces console or AWS CLI, or by using the Amazon WorkSpaces API when a customer launches a new WorkSpace.

To encrypt the volumes, Amazon WorkSpaces uses a customer master key (CMK) from AWS Key Management Service (AWS KMS). A default AWS KMS CMK is created the first time a WorkSpace is launched in a region (CMKs have a region scope).

A customer can also create a customer-managed CMK to use with encrypted WorkSpaces. The CMK is used to encrypt the data keys that are used by Amazon WorkSpaces service to encrypt each of the WorkSpace volumes (in a strict sense, it will be Amazon Elastic Block Store (Amazon EBS) service that will encrypt the volumes). For current CMK limits, review the AWS KMS Resource Quotas documentation.

> **Note:** Creating custom images from an encrypted WorkSpace is not supported. Also, WorkSpaces launched with root volume encryption enabled can take up to an hour to be provisioned.

For a detailed description of the WorkSpaces encryption process, see Overview of Amazon WorkSpaces Encryption Using AWS KMS. Consider how the use of CMK will be monitored to ensure that a request for an encrypted WorkSpace is serviced correctly. For additional information about AWS KMS customer master keys and data keys, see AWS Key Management Service Concepts.

Access Control Options and Trusted Devices

Amazon WorkSpaces provides customers options to manage which client devices can access WorkSpaces. Customers can limit WorkSpaces access to trusted devices only. Access to WorkSpaces can be allowed from macOS and Microsoft Windows PCs using digital certificates. It can also allow or block access for iOS, Android, Chrome OS, Linux, and zero clients, as well as the WorkSpaces Web Access client. With these capabilities, it can further improve the security posture.

Access control options are enabled for new deployments for users to access their WorkSpaces from clients on Windows, MacOS, iOS, Android, ChromeOS and Zero Clients. Access using Web Access or a Linux WorkSpaces client is not enabled by default for a new Workspaces deployment, and will need to be enabled.

If there are limits on corporate data access from trusted devices (also known as managed devices), WorkSpaces access can be restricted to trusted devices with valid certificates. When this feature is enabled, Amazon WorkSpaces uses certificate-based authentication to determine whether a device is trusted. If the WorkSpaces client application can't verify that a device is trusted, it blocks attempts to log in or reconnect from the device.

For more information about controlling which devices can access WorkSpaces see Restrict WorkSpaces Access to Trusted Devices.

> **Note: Certificates for trusted devices only apply to the Amazon WorkSpaces Windows and macOS clients.** This feature does not apply to the Amazon WorkSpaces Web Access client, or any third-party clients, including but not limited to Teradici PCoIP software and mobile clients, Teradici PCoIP zero clients, RDP clients, and remote desktop applications.

IP Access Control Groups

Using IP address-based control groups, customers can define and manage groups of trusted IP addresses, and only allow users to access their WorkSpaces when they're connected to a trusted network. This feature helps customers gain greater control over the security posture.

IP access control groups can be added at the WorkSpaces directory level. There are two ways to get started using IP access control groups. First, from the WorkSpaces management console, IP access control groups can be created on the **IP Access Controls** page. Rules can be added to these groups by entering the IP addresses or IP ranges from which WorkSpaces can be accessed. These groups can then be added to directories on the **Update Details** page. Secondly, WorkSpaces APIs can be used to create, delete, and view groups; create or delete access rules; or to add and remove groups from directories.

For a detailed description of the using IP access control groups with the Amazon WorkSpaces encryption process, see IP Access Control Groups for Your WorkSpaces.

Monitoring or Logging Using Amazon CloudWatch

Monitoring is an integral part of any infrastructure, be that network, servers, or logs. Customers who deploy Amazon WorkSpaces need to monitor their deployments, specifically the overall health and connection status of individual WorkSpaces.

Amazon CloudWatch Metrics for WorkSpaces

CloudWatch metrics for WorkSpaces is designed to provide administrators with additional insight into the overall health and connection status of individual WorkSpaces. Metrics are available per WorkSpace or aggregated for all WorkSpaces in an organization within a given directory (*AD Connector, see Identity*).

These metrics, like all CloudWatch metrics, can be viewed in the AWS Management Console (Figure 17), accessed via the CloudWatch APIs, and monitored by CloudWatch alarms and third-party tools.

By default, the following metrics are enabled and are available at no extra cost:

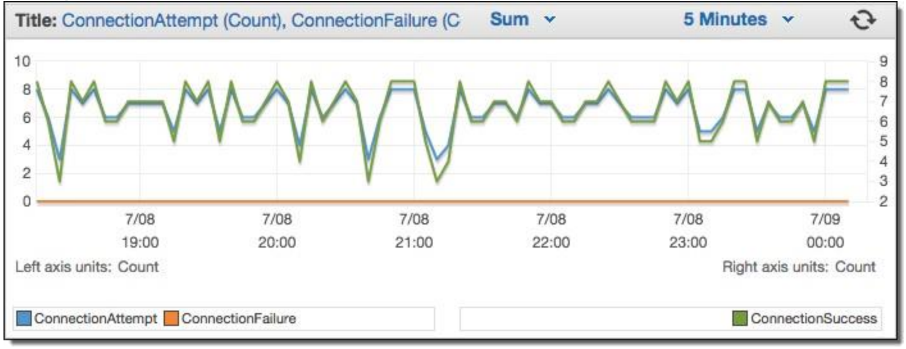
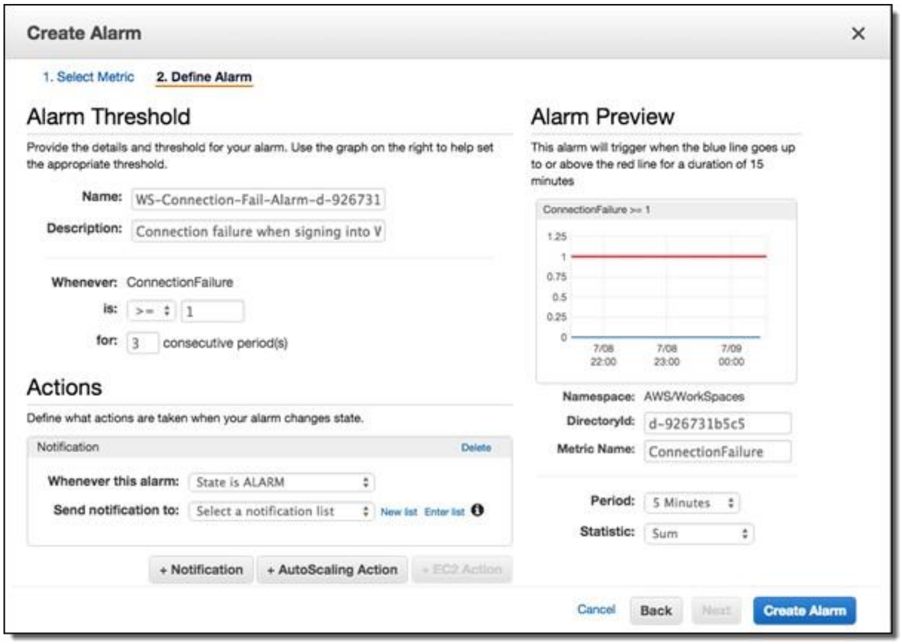


Figure 17: CloudWatch metrics – ConnectionAttempt/ConnectionFailure

- **Available**: WorkSpaces that respond to a status check are counted in this metric.
- **Unhealthy**: WorkSpaces that don’t respond to the same status check are counted in this metric.
- **ConnectionAttempt**: The number of connection attempts made to a WorkSpace.
- **ConnectionSuccess**: The number of successful connection attempts.
- **ConnectionFailure**: The number of unsuccessful connection attempts.
- **SessionLaunchTime**: The amount of time taken to initiate a session, as measured by the WorkSpaces client.
- **InSessionLatency**: The round-trip time between the WorkSpaces client and WorkSpaces, as measured and reported by the client.
- **SessionDisconnect**: The number of user-initiated and automatically closed sessions.

Additionally, alarms can be created, as shown in Figure 17.



Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: WS-Connection-Fail-Alarm-d-926731

Description: Connection failure when signing into V

Whenever: ConnectionFailure

is: >= 1

for: 3 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM

Send notification to: Select a notification list

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 15 minutes

Namespace: AWS/WorkSpaces

DirectoryId: d-926731b5c5

Metric Name: ConnectionFailure

Period: 5 Minutes

Statistic: Sum

Figure 18: Create CloudWatch alarm for WorkSpaces connection errors

Amazon CloudWatch Events for WorkSpaces

Events from Amazon CloudWatch Events can be used to view, search, download, archive, analyze, and respond to successful logins to WorkSpaces. The service can monitor client WAN IP addresses, Operating System, WorkSpaces ID, and Directory ID information for users' logins to WorkSpaces. For example, it can use events for the following purposes:

- Store or archive WorkSpaces login events as logs for future reference, analyze the logs to look for patterns, and take action based on those patterns.
- Use the WAN IP address to determine where users are logged in from, and then use policies to allow users access only to files or data from WorkSpaces that meet the access criteria found in the CloudWatch Event type of `WorkSpaces Access`.
- Use policy controls to block access to files and applications from unauthorized IP addresses.

For more information on how to use CloudWatch Events, see the Amazon CloudWatch Events User Guide. To learn more about CloudWatch Events for WorkSpaces, see Monitor your WorkSpaces using Cloudwatch Events

Cost Optimization

Self-Service WorkSpace Management Capabilities

In Amazon WorkSpaces, self-service WorkSpace management capabilities can be enabled for users to provide them with more control over their experience. Allowing users self-service capability can also reduce your IT support staff workload for Amazon WorkSpaces. self-service capabilities are enabled, it allows users to perform one or more of the following tasks directly from their Windows, macOS, or Linux client for Amazon WorkSpaces:

- Cache their credentials on their client. This lets them reconnect to their WorkSpace without re-entering their credentials.
- Restart their WorkSpace.
- Increase the size of the root and user volumes on their WorkSpace.
- Change the compute type (bundle) for their WorkSpace.
- Switch the running mode of their WorkSpace.
- Rebuild their WorkSpace.

There are no on-going cost implications for allowing users the **Restart** and **Rebuild** options for their WorkSpaces. Users should be aware that a **Rebuild** of their WorkSpace will cause their WorkSpace to be unavailable, for up to an hour, as the rebuild process takes place.

Options to *increase the size of the volumes*, *change the compute type* and *switch the running mode* can incur additional costs for WorkSpaces. A best practice is to enable self-service to reduce the workload for the support team. Self-service for additional cost items should be allowed within a workflow process that ensures that authorization for additional charges has been obtained. This could be through a dedicated self-service portal for WorkSpaces, or by integration with existing Information Technology Service Manage (ITSM) services, such as ServiceNow.

For more detailed information, read Enabling Self-Service WorkSpace Management Capabilities for Your Users. For an example describing enabling a structured portal for user self-service, read Automate Amazon WorkSpaces with a Self-Service Portal.

Amazon WorkSpaces Cost Optimizer

The *running mode* of a WorkSpace determines its immediate availability and how it will be billed. Here are the current running WorkSpaces running mode:

- **AlwaysOn** — Use when paying a fixed monthly fee for unlimited usage of WorkSpaces. This mode is best for users who use their WorkSpace full time as their primary desktop.
- **AutoStop** — Use when paying for WorkSpaces by the hour. With this mode, WorkSpaces stop after a specified period of inactivity and the state of apps and data is saved. To set the automatic stop time, use **AutoStop Time (hours)**.

A best practice is to monitor usage and set the WorkSpaces' running mode to be the most cost effective. This can be done with the Amazon WorkSpaces Cost Optimizer. This solution deploys an Amazon CloudWatch event that invokes an AWS Lambda function every 24 hours.

This solution can convert individual WorkSpaces from an hourly billing model to a monthly billing model on any day after the threshold is met. If the solution converts a WorkSpace from hourly billing to monthly billing, the solution does not convert the WorkSpace back to hourly billing until the beginning of the next month, and only if usage was below the threshold. However, the billing model can be manually change at any time using the AWS Management Console. The solution's AWS CloudFormation template includes parameters that will execute these conversions.

Opting Out with Tags

To prevent the solution from converting a WorkSpace between billing models, 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 the tagged WorkSpaces. Remove the tag at any time to resume automatic conversion for that WorkSpace.

For more details, read Amazon WorkSpaces Cost Optimizer.

Troubleshooting

Common administration and client issues, such as error messages like "Your device is not able to connect to the WorkSpaces Registration service" or “Can't connect to a WorkSpace with an interactive logon banner”, can be found on the Client and Admin Troubleshooting pages in the *Amazon WorkSpaces Administration Guide*.

AD Connector Cannot Connect to Active Directory

For AD Connector to connect to the on-premises directory, the firewall for the on-premises network must have certain ports open to the CIDRs for both subnets in the VPC (see AD Connector). To test if these conditions are met, perform the following steps.

To test the connection:

1. Launch a Windows instance in the VPC and connect to it over RDP.

   The remaining steps are performed on the VPC instance.

2. Download and unzip the DirectoryServicePortTest test application. The source code and Microsoft Visual Studio project files are included to modify the test application, if desired.

3. From a Windows command prompt, run the DirectoryServicePortTest test application with the following options:

```
DirectoryServicePortTest.exe -d <domain_name>
-ip <server_IP_address> -tcp "53,88,135,139,389,445,464,636,49152"
-udp "53,88,123,137,138,389,445,464" <domain_name>
```

<domain_name>

The fully qualified domain name, used to test the forest and domain functional levels. If the domain name is excluded, the functional levels won't be tested.

<server_IP_address>

The IP address of a domain controller in the on-premises domain. The ports are tested against this IP address. If the IP address is excluded, the ports won't be tested.

This test determines if the necessary ports are open from the VPC to the domain. The test app also verifies the minimum forest and domain functional levels.

Troubleshooting a WorkSpace Custom Image Creation Error

If a Windows or Amazon Linux WorkSpace has been launched and customized, a custom image can be created from that WorkSpace. A custom image contains the operating system, application software, and settings for the WorkSpace.

Review the requirements to create a Windows custom image or the requirements to create an Amazon Linux custom image. The image creation requires that all prerequisites are met before image creation can start.

To confirm that the Windows WorkSpace meets the requirements for image creation, we recommend running the Image Checker. The Image Checker performs a series of tests on the WorkSpace when an image is created, and provides guidance on how to resolve any issues it finds. For detailed information read installing and configuring the image checker

After the WorkSpace passes all tests, a **Validation Successful** message appears. A custom bundle can now be created. Otherwise, resolve any issues that cause test failures and warnings, and repeat the process of running the Image Checker until the WorkSpace passes all tests. All failures and warnings must be resolved before an image can be created.

Follow the tips for resolving issues detected by the Image Checker for additional details.

Troubleshooting a Windows WorkSpace Marked as Unhealthy

The Amazon WorkSpaces service periodically checks the health of a WorkSpace by sending it a status request. The WorkSpace is marked as Unhealthy if a response isn’t received from the WorkSpace in a timely manner. Common causes for this problem are:

- An application on the WorkSpace is blocking network connection between the Amazon WorkSpaces service and the WorkSpace.
- High CPU utilization on the WorkSpace.
- The computer name of the WorkSpace is changed.

- The agent or service that responds to the Amazon WorkSpaces service isn't in running state.

The following troubleshooting steps can return the WorkSpace to a healthy state:

- First, reboot the WorkSpace from the Amazon WorkSpaces console. If rebooting the WorkSpace doesn't resolve the issue, either connect to a Windows WorkSpace using Remote Desktop Connection (RDP), or connect to an Amazon Linux WorkSpace using SSH
- If the WorkSpace is unreachable by a different protocol, rebuild the WorkSpace from the Amazon WorkSpaces console.
- If a WorkSpaces connection cannot be established, verify the following:

  Verify CPU Utilization

  Open Task Manager to determine if the WorkSpace is experiencing high CPU utilization. If it is, try any of the following troubleshooting steps to resolve the issue:

  - Stop any service that is consuming a high amount of CPU
  - Resize the WorkSpace to a compute type greater than what is currently used
  - Reboot the WorkSpace

> Note: To diagnose high CPU utilization, and for guidance if the above steps don't resolve the high CPU utilization issue, see How do I diagnose high CPU utilization on my EC2 Windows instance when my CPU is not throttled?

  Verify the Computer Name of the WorkSpace

  If the computer name of the WorkSpacewas changed, change it back to the original name.

  1. Open the Amazon WorkSpaces console, and then expand the Unhealthy WorkSpace to show details.
  2. Copy the **Computer Name**.
  3. Connect to the WorkSpace using RDP.

4. Open a command prompt, and then enter hostname to view the current computer name.
   If the name matches the Computer Name from step 2, skip to the next troubleshooting section.
   If the names don't match, enter sysdm.cpl to open system properties, and then follow the remaining steps in this section.

5. Choose **Change**, and then paste the Computer Name from step 2.

6. Enter the domain user credentials if prompted.

Confirm that SkyLightWorkspaceConfigService is in Running State

From Services, verify if the WorkSpace service SkyLightWorkspaceConfigService is in running state. If it's not, start the service.

Verify Firewall Rules

Confirm that the Windows Firewall and any third-party firewall that is running have rules to allow the following ports:

- Inbound TCP on port 4172: Establish the streaming connection.
- Inbound UDP on port 4172: Stream user input.
- Inbound TCP on port 8200: Manage and configure the WorkSpace.
- Outbound UDP on port 55002: PCoIP streaming.

If the firewall uses stateless filtering, then open ephemeral ports 49152-65535 to allow return communication.

If the firewall uses stateful filtering, then ephemeral port 55002 is already open.

Collecting a WorkSpaces Support Log Bundle for Debugging

When troubleshooting WorkSpaces issues, it will be necessary to gather the log bundle from the affected WorkSpace and the host where the WorkSpaces client is installed. There are two fundamental categories of logs:

- **Server-side logs**: The WorkSpace is the server in this scenario, so these are logs that live on the WorkSpace itself.
- **Client-side logs**: These will be on the device that the end user is using to connect to the WorkSpace.
  - Note that only Windows and macOS clients write logs locally.
  - Zero clients and iOS clients do not log.
  - Android logs are encrypted on the local storage and uploaded automatically to the WorkSpaces client engineering team. Only that team can review the logs for Android devices.

PcoIP Server-Side Logs

All of the PCoIP components write their log files into one of two folders:

- Primary location: `C:\ProgramData\Teradici\PCoIPAgent\logs`
- Archive location: `C:\ProgramData\Teradici\logs`

Sometimes when working with AWS Support on a complex issue, it will be necessary to put the PCoIP Server agent into verbose logging mode. To enable this:

1. Open the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Teradici\PCoIP\pcoip_admin_defaults`
2. In the `pcoip_admin_defaults` key create the following 32bit DWORD: `pcoip.event_filter_mode`
3. Set the value for `pcoip.event_filter_mode` to “3” (Dec or Hex)

For reference, these are the log thresholds which can be defined in this DWORD.

0 — (CRITICAL)
1 — (ERROR)
2 — (INFO)
3 — (Debug)

If the `pcoip_admin_default` DWORD doesn’t exist, the log level is 2 by default. It is recommended to restore a value of 2 to the DWORD after it no longer need verbose logs, as they are much larger and will consume disk space unnecessarily.

WebAccess Server-Side Logs

The WorkSpaces web access client uses the STXHD service. The logs for WebAccess is stored at `C:\ProgramData\Amazon\Stxhd\Logs`.

Client-Side Logs

These logs come from the WorkSpaces client that the user connects with, and as such the logs will be on the end user's computer. The log file locations for Windows and Mac are shown below.

Windows: `"%LOCALAPPDATA%\Amazon Web Services\Amazon WorkSpaces\Logs"`
macOS: `~/Library/Logs/Amazon Web Services/`
Linux: `~/.local/share/Amazon Web Services/Amazon WorkSpaces/logs`

To help troubleshoot issues that the users might experience, enable advanced logging can be used on any Amazon WorkSpaces client. Advanced logging is enabled for every subsequent client session until it is disabled.

1. Before connecting to the WorkSpace, the end user should enable advanced logging for their WorkSpaces client

2. The end user should then connect as normal and use their WorkSpace, and attempt to reproduce the issue.

3. Advanced logging generates log files that contain diagnostic information and debugging-level details, including verbose performance data.

This setting persists until explicitly turned off. Once the user has been able to reproduce the issue with verbose logging on, this setting should be disabled, as it generates large log files.

Automated Server Side Log Bundle Collection for Windows

The **Get-WorkSpaceLogs.ps1** script is very helpful for quickly gathering a server-side log bundle for AWS Premium Support. The script can be requested from AWS Premium Support by requesting it in a support case.

1. Connect to the WorkSpace using the client or using Remote Desktop Protocol (RDP)

2. Start an administrative Command Prompt (i.e. Run as administrator).

3. Launch the script from the Command Prompt with the following command:

```
powershell.exe -NoLogo -ExecutionPolicy RemoteSigned -NoProfile -
File "C:\Program Files\Amazon\WorkSpacesConfig\Scripts\Get-
WorkSpaceLogs.ps1"
```

4. The script will create a log bundle on the user's desktop.

The script creates a zip file with the following folders:

- **C**: it contains the files from Program Files, Program Files (x86), ProgramData, and Windows related to Skylight, EC2Config, Teradici, Event viewer, and Windows logs (Panther and others)
- **CliXML**: it contains XML files that can be imported in PowerShell by using Import-CliXML for interactive filtering (https://msdn.microsoft.com/en-us/powershell/reference/5.1/microsoft.powershell.utility/import-clixml)
- **Config**: detailed logs for each check that is performed
- **ScriptLogs**: logs about the script execution (not relevant to the investigation, but useful to debug what the script does)
- **tmp**: temporary folder (it should be empty)
- **Traces**: packet capture done during the log collection

How to Check Latency to Closest AWS Region

The Connection Health Check website quickly checks whether all of the required services that use Amazon WorkSpaces can be reached. It also does a performance check to each AWS Region where Amazon WorkSpaces is available, and lets users know which one will be the fastest for them.

Conclusion

We're seeing a strategic shift in end-user computing as organizations strive to be more agile, better protect their data, and help their workers be more productive. Many of the benefits already realized with cloud computing also apply to end user computing. By moving their Windows or Linux desktops to the AWS Cloud with Amazon WorkSpaces, organizations can quickly scale as they add workers, improve their security posture by keeping data off devices, and offer their workers a portable desktop with anywhere access from the device of their choice.

Amazon WorkSpaces is designed to be integrated into existing IT systems and processes, and this whitepaper described the best practices for doing this. The result of following the guidelines in this whitepaper is a cost-effective cloud desktop deployment that can securely scale with your business on the AWS global infrastructure.

Contributors

Contributors to this document include:

- Naviero Magee, Sr. EUC Solutions Architect, Amazon Web Services
- Andrew Wood, Sr. EUC Solutions Architect, Amazon Web Services
- Dzung Nguyen, Sr. Consultant, Amazon Web Services

Further Reading

For additional information, see:

- Amazon WorkSpaces Administration Guide
- Amazon WorkSpaces Developer Guide
- Amazon WorkSpaces Clients
- Managing Amazon Linux 2 Amazon WorkSpaces with AWS OpsWorks for Puppet Enterprise
- Customizing the Amazon Linux WorkSpace
- How to improve LDAP Security in AWS Directory Service with client-side LDAPs
- Use Amazon CloudWatch Events with Amazon WorkSpaces and AWS Lambda for greater fleet visibility
- How Amazon WorkSpaces Use AWS KMS
- AWS CLI Command Reference – WorkSpaces
- Monitoring Amazon WorkSpaces Metrics
- MATE Desktop Environment

- Troubleshooting AWS Directory Service Administration Issues
- Troubleshooting Amazon WorkSpaces Administration Issues
- Troubleshooting Amazon WorkSpaces Client Issues
- Automate Amazon WorkSpaces with a Self-Service Portal

Document Revisions

Date	Description
June 2020	Corrected link and minor text updates.
May 2020	Updated content since first publication and added new diagrams.
July 2016	First publication