Securing the Microsoft Platform on Amazon Web Services

August 2019

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

Deploying Microsoft products on Amazon Web Services (AWS) is fast, easy, and cost-effective. Before deploying these applications to production, it is helpful to have guidance on approaches for securing them. The paper outlines the principles for protecting the runtime environment of applications running on AWS with a focus on risk assessment, reducing attack surface, adhering to the principle of least privilege, and protecting data. This document provides Microsoft Workloads Administrators, Security Experts, DBAs, Cloud and Solutions Architects, and Systems Engineers prescriptive best practices guidance for how to secure the Microsoft Platform on AWS.

Introduction to Amazon Web Services and the Microsoft Platform

Amazon Web Service (AWS) appeals to customers and IT professionals in the Microsoft community because AWS has been running Windows Workloads on AWS for over a decade (since Oct 2008). This experience has earned trust with the Microsoft community, resulting in the continued growth of AWS enterprise customers using Amazon EC2 for Windows Server. According to an IDC report, AWS hosts nearly 2x the number of Windows Server instances than the next largest cloud provider. Additionally, the broader AWS platform provides a rich set of services, that when combined with Microsoft workloads, can further enhance the experience of your users.

Microsoft products together make up the Microsoft platform, a well-integrated set of operating systems, server roles, applications, and development tools that provide flexibility and business agility. The deep integration between applications and infrastructure makes it simple to manage and share enterprise data. The Microsoft platform has a strong focus on user experience, which reduces friction for users, managers, administrators, and developers. AWS is a perfect complement to the Microsoft platform because it allows administrators to rapidly provision pay-as-you-go infrastructure that powers the Microsoft platform and applications designed to run on it.

AWS and Microsoft have worked together to enable customers to deploy enterprise-class workloads involving Windows Server® and Microsoft SQL Server® on a pay-as-you-go, on-demand, elastic infrastructure. This approach eliminates the capital cost for server hardware and greatly reduces the provisioning time required to create or extend, for example, a SharePoint server farm. This joint effort has further resulted in the ability to license and run SharePoint Server and other Microsoft Server products on AWS under provisions in the License Mobility program.

AWS Shared Responsibility Model

To understand the security controls identified and described this whitepaper, you must first be familiar with the AWS Shared Responsibility Model, which requires AWS and customers to work together to achieve common security objectives.

The security controls available in the AWS Cloud offer the same security isolations found in traditional on-premises and data center environments. These include physical data center security, separation of the network, isolation of the server hardware, and isolation of storage.

In the Shared Responsibilities Model, AWS manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. As a customer, you are responsible for building secure solutions and applications on top of the services AWS operates, manages, and controls.

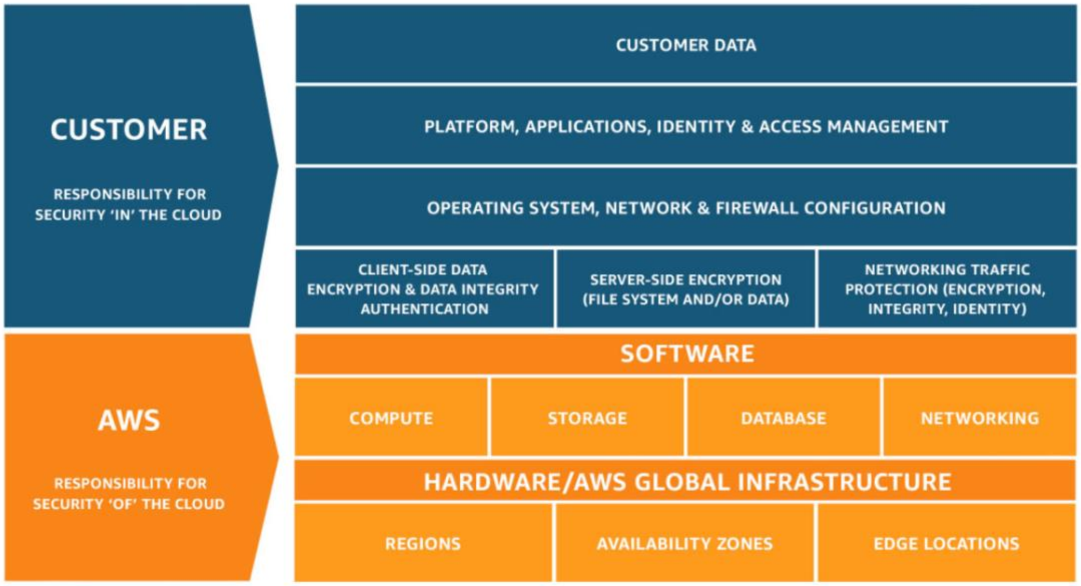

CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

Figure 1: AWS Shared Responsibility Model

Figure 1 shows the differentiation of the responsibilities commonly known as security *of* the cloud versus security *in* the cloud, AWS responsibility versus AWS customer's responsibility, respectively.

AWS customers benefit from inheriting all of the best practices of AWS policies, architecture, and operation processes built to satisfy the requirements of the most security sensitive customers. Security and compliance reports and select online agreements are available for download from the AWS Artifact.

Cloud Security

Cloud security at AWS is the highest priority; providing the same security isolations as traditional data centers, including physical and logical controls and safeguards, which help protect confidentiality, integrity, and availability of customers' solutions. As an AWS customer, it is also critical that you understand the controls and safeguards at your disposal, so you can ensure that you meet your solution's security objectives, from data security and privacy to compliance, cost, and scale.

Resources for Cloud Security

Familiarizing yourself with the services available in AWS and how they interact with each other is an important step in understanding how to protect your system resources and data. The Overview of AWS Security - Network Security whitepaper is a great starting point for understanding the various services and how they are secured or isolated in their default state. The *Overview* whitepaper also helps you understand the protections that are in place and to determine whether you want to take additional steps to protect your resources and data. We also recommend that you review the Security Checklist, as it outlines some additional best practices to secure your account, for services outside of the Microsoft purview.

This document focuses on the controls available for protecting Microsoft Workloads on AWS. Just as critical or more important, is the need to have a well-defined and mature security program, which can be built based on guidance provided in the AWS Cloud Adoption Framework, Security Pillar of the AWS Well-Architected Framework, Introduction to AWS Security, and AWS Security Best Practices.

Risk Assessment

There must be a balance between existing security controls, functional capabilities, and the economic cost of the system. Risk assessment is a technique for identifying and understanding risk, and more specifically the process of studying, analyzing, and describing the set of outcomes for particular threats.

The first step in assessing the risk of your application infrastructure is to make sure that you are deeply familiar with the logical and physical architecture of the system. This architecture includes the application tiers or subsystems and the network communication between the tiers and subsystems. It includes required resources, such as SQL Server, and the underlying configuration of AWS resources, such as Amazon Elastic Block Store (EBS) volumes. It also includes an understanding of who will be interacting with various system interfaces and from where.

Another important step in assessing risk is to complete threat modeling to understand the potential threats to the system, determine the risk, and then develop the appropriate mitigations for the risk. A simple way to do threat modeling is to draw a diagram of your applications' subsystems and network entry points. You can then identify the threats for each interface and interaction, and then address each threat individually until your system is covered. This process is useful because it forces you to think about potential threats upfront, and it provides an opportunity to place controls into the infrastructure at that time.

When considering controls, it is important to adhere to the principle of *least privilege*. This principle refers to users of the system having only the necessary set of permissions to perform their job function and no more. It also refers to the processes in the system having the least possible authority necessary to perform job tasks. This approach helps reduce the attack surface of the system and your environment, making it much harder for an adversary to exploit.

An attack surface can be defined as the set of exploitable vulnerabilities in your environment; these would include the network, software, and users who are involved in the ongoing operation of the system. Always look to reduce the attack surface of the system by exposing the absolute minimal set interfaces, such as reducing the number of ports to the network while also restricting the source network or IP address that will have access to your systems.

Another way of mitigating risk is to have the controls necessary to protect against the accidental or deliberate misuse of confidential data. This refers to protecting the confidentiality, integrity, and availability of data in the system. Consequences for not adequately protecting data can range from the embarrassing (someone vandalizing website content) to the severe (compromise and leak of sensitive data). The following topics address a set of controls that you can implement using AWS capabilities, alongside the functionality that exists in the Microsoft platform. We also cover strategies for using controls to protect sensitive data in the system, whether it's in transit or at rest.

Controls Available in AWS

AWS provides a set of building blocks that customers can use to build infrastructure for their applications. In this model, some security capabilities such as physical security are the responsibility of AWS and are highlighted in the Introduction to AWS Security Processes whitepaper. Other areas, such as controlling access to applications, fall squarely on the application developer and the tools in the Microsoft platform. There is a gray area in between, where AWS has security capabilities that exist in the platform and can be configured to mitigate additional risks in the environment. We address these capabilities in this section.

Resource and Access Management

AWS Identity and Access Management (IAM) and IAM Roles

When running an application infrastructure on AWS, there are two different administrative roles; the AWS administrators, and the application/server administrators. The AWS administrator is responsible for interacting with AWS resources; the application/server administrator is responsible for connecting to and managing Windows Servers. AWS Identity and Access Management (IAM) was designed to provide control over the former set of administrators; those who are responsible for the AWS environment. IAM provides the ability to specify policies for access control to create guardrails for the first set of administrators.

Let’s use the example of an IT organization with an infrastructure team and an application team. The infrastructure team controls the network topology, and the application team is not allowed to modify network resources as a matter of policy. Separate IAM groups can be used for infrastructure and application administrators. Further, the application administrators can be restricted from creating, deleting, or modifying any Amazon Virtual Private Cloud (VPC) resources (subnets, security groups, ACLs). Adhering to best practices, you might then decide that you want to limit the infrastructure team and prevent them from starting or stopping specific Amazon Elastic Compute Cloud (Amazon EC2) instances. You could also restrict the infrastructure team members’ ability to create and attach Amazon EBS volumes, and also prevent modification of Auto Scaling groups. Lastly, you may want to ensure that none of these administrators have access to the rolled-up billing information for the account. All of these approaches can be achieved with AWS IAM users and groups with appropriate policies restricting access to the various areas described.

Since AWS offers a wide range of services that applications can consume, applications are often required to possess credentials for those services. Rather than embedding IAM users’ credentials on an Amazon EC2 instance to access a particular service, use IAM Roles. IAM Roles allow you to launch an Amazon EC2 instance with a predetermined set of IAM authorizations, where credentials are made available to the Amazon EC2 instance. The credentials are automatically rotated several times a day, and they can be retrieved programmatically via the EC2 Instance Metadata Service.

For example, to ensure that all EC2 instances launched join an Active Directory domain, you could create a new IAM role called **EC2RoleforSSM** and attach the

AmazonEC2RoleforSSM policy. This policy enables instances to communicate with the Systems Manager API, which can then be used to domain join instances. Another common use is granting access of the EC2 instance to S3 buckets, removing the need to store credentials in your application.

Directory Services

The controls available for the application/server administrator responsible for administering Windows Servers, Active Directory, and other domain resources are covered in this section.

Active Directory is a foundation of security and access control for Windows-based workloads used by the majority of large enterprises. AWS supports several options for deploying Directory Services in the cloud and integrating with on-premises infrastructure and other identity providers. Maintaining security of Directory Services is critical for the entire infrastructure of any solution.

This whitepaper covers only security-related aspects of deploying Directory Services. For more information about selecting the right Directory Service, deployment topology and network configuration, see the Active Directory Domain Services on AWS whitepaper.

AWS Directory Service for Microsoft Active Directory

AWS Directory Service for Microsoft Active Directory, also known as AWS Managed Microsoft AD, lets you run Microsoft Active Directory (AD) as a managed service. When you deploy AWS Managed Microsoft AD, it creates a highly available pair of domain controllers connected to your virtual private cloud (VPC). The domain controllers run in different Availability Zones in a Region of your choice.

Applying the Shared Responsibility Model to AWS Managed Microsoft AD means that AWS is responsible for deploying and managing the domain controllers and DNS service, performing Active Directory maintenance and administrative tasks. The customer is responsible for managing data and delegated permissions to the designated Active Directory Organization Unit. With this approach, the infrastructure provisioning, management, and maintenance of the directory is handled by AWS. The customer is responsible for configuring and maintaining the resource and service access management for resources in AWS. The methodology to secure the resources in the managed domain should follow the best practice of least privilege, permitting administrators to perform only role-specific functions (i.e. complete tasks) and no more.

Forest Models

There are two forest design patterns that are prevalent with AWS Managed Microsoft AD. The Restricted Access Forest Model and the Resource Forest Model.

Restricted Access Forest Model

The Restricted Access Forest Model is an isolated and completely independent forest within AWS. This model creates a separate (and disconnected) security boundary between your on-premises domain and resources and the AWS based resources, which may be a requirement for some customers.

Resource Forest Model

The Resource Forest model allows on-premises users to access AWS bound resources. This design provides similar security boundary isolations as the previous design, while at the same time allows the use of on-premises based credentials to access resources in AWS, via Trusts. In this model, there is no access by default, and the trust is used to read objects in the directory.

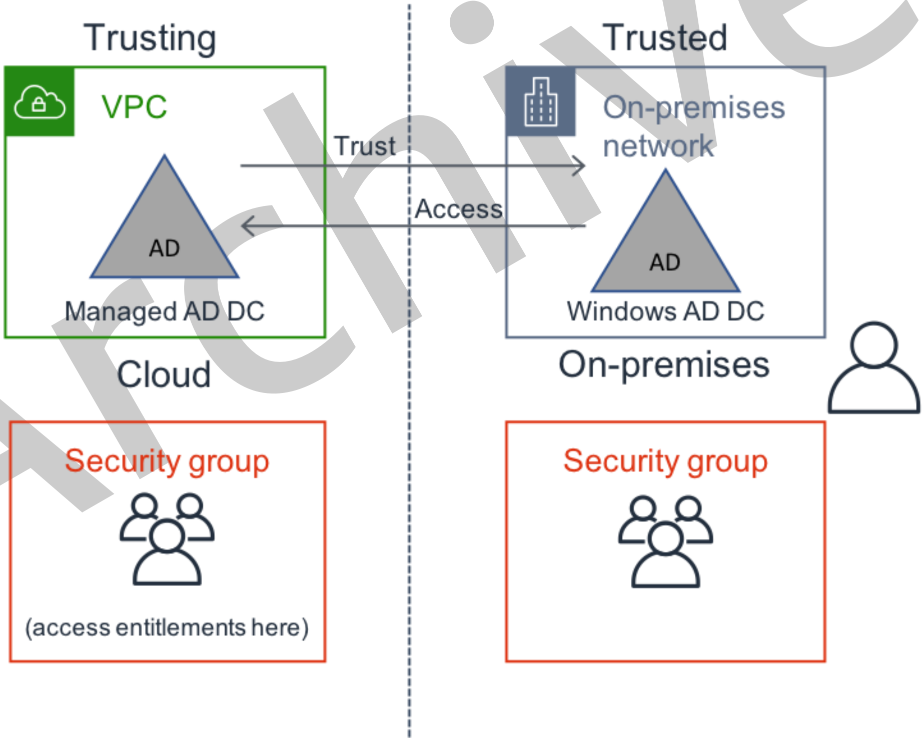


Figure 2: Forest Trust

> **NOTE:** Forest Trust requires private network connectivity between on-premises domain controllers and the managed domain controllers, using the VPN-VPC or Direct Connect (DX) connections.

Restricting Access to AWS Managed Microsoft AD

Despite of the fact that AWS Managed Microsoft AD does not provide you service level administrative privileges, it is still important to protect user accounts managing your Directory. We recommend that you create a separate AWS account to centralize common resources used by the entire organization (i.e. shared services). Deploy Active Directory service in this account and share it with other accounts in your organization. Only a limited group of users should be able to administer services in this account. We highly recommend that you enable multi-factor authentication (MFA) for an extra layer of protection.

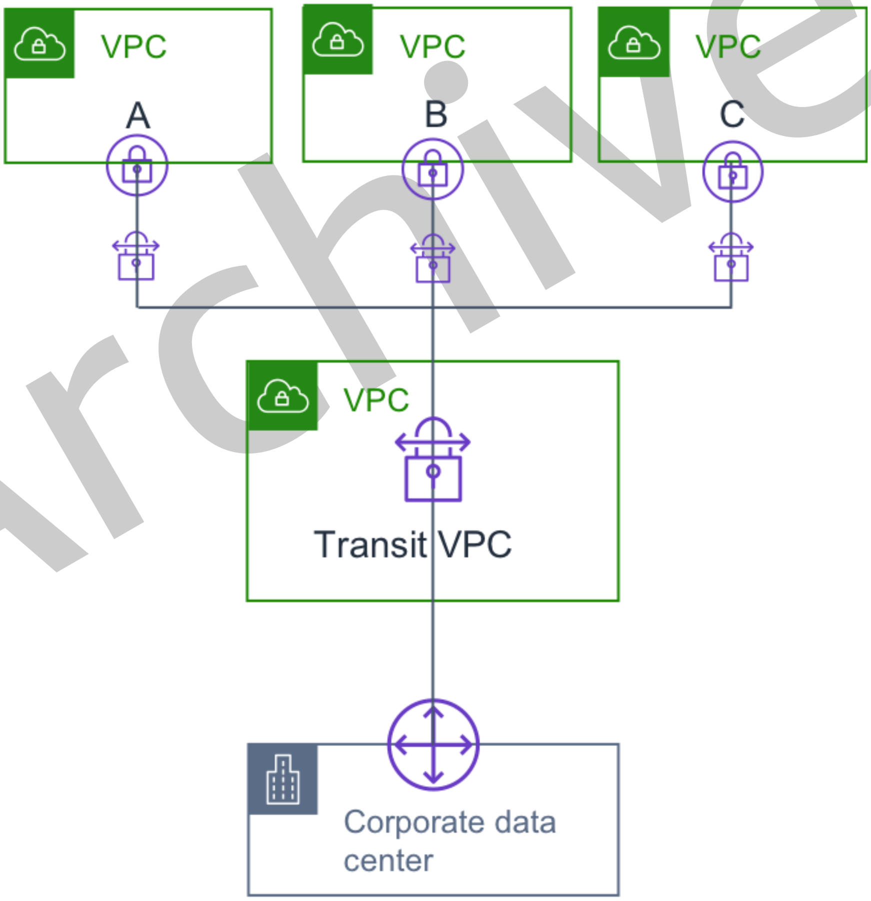


Figure 3: Using separate account for shared services

Generally, you should treat Active Directory in the cloud in the same manner as on-premises and limit administrative access to the AWS account the same way you control access to the physical data center with your Active Directory domain controllers.

Create additional AWS accounts for developers and IT Pros groups in your organization and share the AWS Managed Microsoft AD with them. Once you have successfully shared Active Directory and configured routing, the groups can use it to join EC2 instances, but you still maintain control of all administrative tasks.

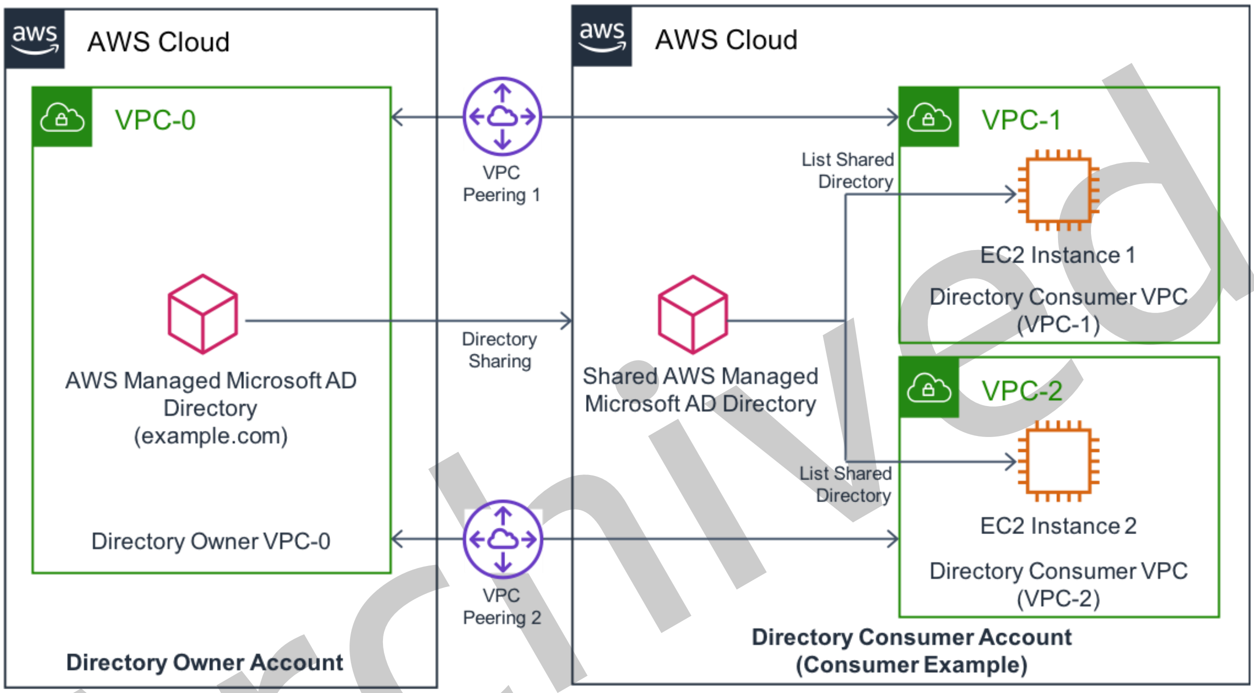

AWS Cloud
VPC-0
AWS Managed Microsoft AD Directory (example.com)
Directory Owner VPC-0
Directory Owner Account
VPC Peering 1
Directory Sharing
VPC Peering 2
AWS Cloud
Shared AWS Managed Microsoft AD Directory
List Shared Directory
VPC-1
EC2 Instance 1
Directory Consumer VPC (VPC-1)
VPC-2
List Shared Directory
EC2 Instance 2
Directory Consumer VPC (VPC-2)
Directory Consumer Account (Consumer Example)

Figure 4: Sharing single AWS Managed Microsoft AD with another account

Active Directory Permissions Delegation

When you use AWS Managed Microsoft AD, part of the Service Owners responsibilities is assumed by AWS so you can focus on other business critical tasks. Service-level tasks are automatically performed by the AWS managing service.

With AWS Managed Microsoft AD, you can delegate administrative permissions to some groups in your organization. These permissions include managing user accounts, joining computers to the domain, managing Group Policy and password policies, managing DNS, DHCP, DFS, RAS, CA, and other services. The full list of delegated permissions is described in the AWS Directory Service Administration Guide.

Work with all teams using Active Directory services in your organization and create a list with all permissions that must be delegated. Plan security groups for different administrative roles and use AWS Microsoft Managed AD Delegated Groups to assign permissions. Check the AWS Directory Service Administration Guide to be sure that it is possible to delegate all required permissions.

Multi-Factor Authentication

Multi-Factor Authentication (MFA) is a best practice that adds an extra layer of protection on top of your user name and password. When a user signs into the AWS Console with MFA enabled, the user is prompted for a user name and password (the first factor—what they know), and an authentication response from an AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account. We recommend that you enable MFA on all of your privileged accounts, whether you are using IAM or federating through AWS Single Sign-On (SSO).

To enable MFA, you must have an MFA solution that is a Remote Authentication Dial-In User Service (RADIUS) server, or you must have an MFA plugin to a RADIUS server already implemented in your on-premises infrastructure. Your MFA solution should implement One Time Passcodes (OTP) that users obtain from a hardware device or from software running on a device, such as a mobile phone.

Complete information about requirements and settings for enabling MFA for AWS Managed Microsoft AD is available on the technical documentation page.

Self-Managed Microsoft Active Directory

Self-managed Active Directory (AD) Domain Controllers can run on Amazon EC2 instances, and can be replicated from on-premises domain controllers using the VPN-VPC or AWS Direct Connect (DX) connections. This setup allows EC2 Instances to authenticate with local domain controllers and still authenticate corporate identities and credentials. Although it is possible to directly authenticate to corporate domain controllers over VPN-VPC or Direct Connect (DX) connections, replicating to AWS provides for better performance. It is also the recommended approach if you plan to have a single, flat domain with a single security boundary. It is best practice to replicate and deploy your DCs across Availability Zones (as with your other resources) to provide high availability.

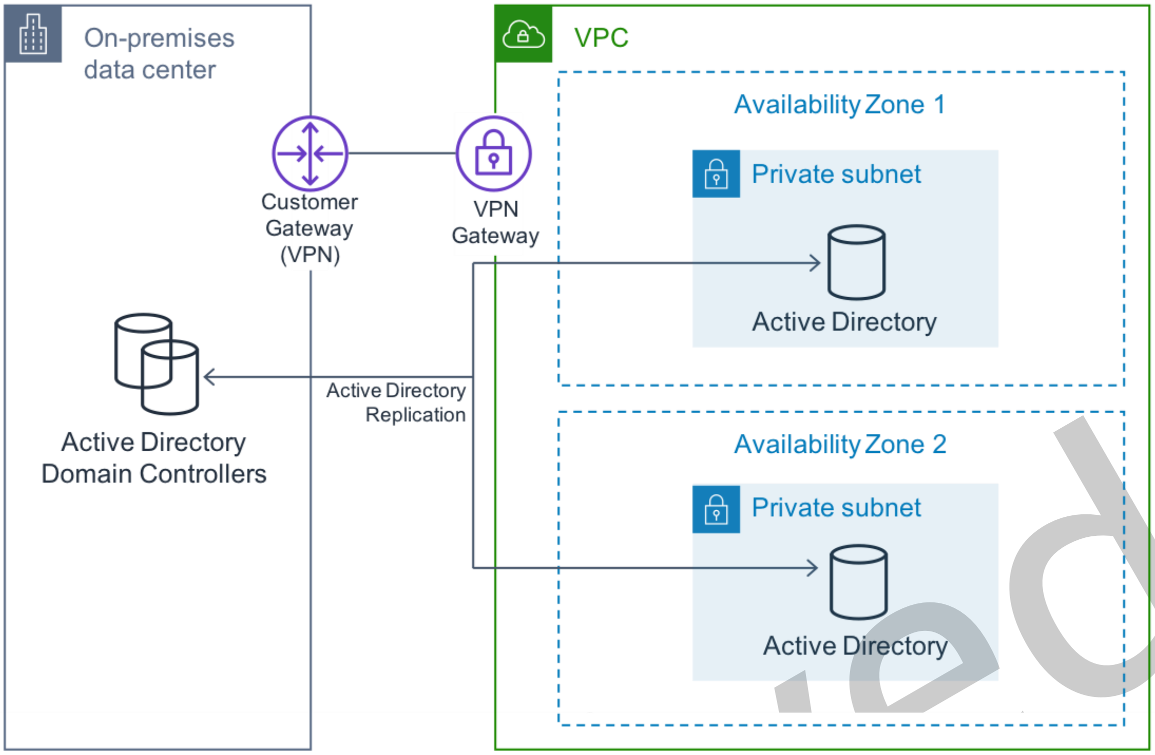


Figure 5: Active Directory Deployment in AWS with Replication

Whether authenticating locally or remotely, via VPN-VPC or Direct Connect (DX) connections, ensure the privileges assigned to your Active Directory Security Groups do not grant more permissions than what is necessary. Doing so ensures that your AD groups do not contain an unnecessary number of privileged users or users with unnecessary privileges, which may increase the risk exposure to Active Directory, Member Servers of the domain, Workstations, Applications, and even Data Repositories.

Active Directory (AD) Connector

AD Connector is another component of AWS Directory Services, which allows customers to authenticate to AWS-based resources using their on-premises AD credentials. AD Connector is a directory gateway, which redirects authentication requests from the AWS Cloud to an existing Active Directory domain, without caching credentials or replicating any of your directory data.

Network connectivity between the on-premises domain controllers and the AD Connector components is required for both VPN-VPC or Direct Connect (DX) connections.

NOTE: Although AD Connector can be used for extending functionality, including authenticating to the AWS Console and accessing the suite of Amazon Enterprise Applications (e.g. Amazon WorkSpaces, Amazon WorkMail, Amazon Chime, and Amazon WorkDocs), Amazon QuickSight, and AWS Single Sign-On (SSO); it is not suitable for .NET or Windows based applications. Active Directory should be used instead.

Securing Active Directory Domain Controllers

Domain controllers provide the physical storage for the AD DS database, in addition to providing the services and data that allow enterprises to effectively manage their servers, workstations, users, and applications. If privileged access to a domain controller is obtained by an unauthorized or malicious user, that user can modify, corrupt, or destroy the AD DS database and, by extension, all of the systems and accounts that are managed by Active Directory. Because domain controllers can read from and write to anything in the AD DS database, a compromise of a domain controller means that your Active Directory forest can never be considered trustworthy again, unless it can be recovered using a known good backup and close the gaps that allowed the compromise in the process. The following are best practices to secure Domain Controllers running on AWS:

- Restrict AWS user account (IAM) permissions to only authorized users for create/access EBS snapshots, launch/terminate EC2 Instances, and create/copy EBS volumes.
- Deploy your Domain Controllers in a private subnet. Ensure that the Route Table does not route to a NAT gateway or other device that would provide outbound internet access.
- Restrict allowed ports and protocols into the Domain Controllers, using security groups. Allow remote management (RDP, WinRM) only from trusted networks.
- Leverage EBS encryption on the root and additional volumes using AWS Key Management Service (AWS KMS).
- Implement backup routines and keep your backups in a location with limited access.
- Keep your security patches up-to-date. Test patches in non-production environments before installing patches on production environments. Makes Focus on testing security patches as they can change an environment’s behavior.

- Staying up-to-date with Microsoft’s security best practices, as new techniques to secure an environment or implications may be identified.

Protecting Administrative Accounts in Active Directory

By default, Active Directory Domain Controllers have a secure configuration. A compromise can start from compromising domain-joined computers and administrative workstations or stealing credentials using different techniques.

To limit the possibility of compromise, follow these best practices:

- To prevent possible pass-the-hash attacks, use isolated secure workstations and jump-boxes that are only used for administering Active Directory.
- On domain-based accounts, enable the *Account is sensitive and cannot be delegated* attribute to disable account delegation for critical accounts.
- Use smart cards for administrative accounts.
- Do not add domain users’ day-to-day accounts to any AD administrative groups. Create separate accounts, which should only be used for AD management activities.
- Use role-based access control (RBAC) to delegate administrative permissions to user groups, if necessary. For example, use DNS Administrators security group to give network administrators permissions to manage DNS zones on domain controllers.

IAM with Federation

Active Directory Federation Services

It is also possible to support a scenario in which corporate environments do not use AD, but rather another Lightweight Directory Access Protocol (LDAP)–based directory service. In this case, you can use Active Directory Federation Services (AD FS) to facilitate federated authentication using Security Assertion Markup Language 1.1. and 2.0 (SAML), OAuth, or OpenID Connect. AWS provides a detailed blog post on how to set up and configure AD FS in AWS to support federated authentication. AD FS can be used both with AWS Managed Microsoft AD and self-managed Active Directory. Additionally, ADFS can be valuable in a case of using applications, that does not support SAML 2.0, like SharePoint 2013 and 2016.

AWS Single Sign-On

AWS Single Sign-On (SSO) is similar to AD FS, in that it provides federated authentication services for Active Directory based credentials, with the difference being that you do not need to manage or maintain the federation services infrastructure because AWS SSO is a cloud-based service, which makes it easy to centrally manage SSO access to multiple AWS accounts and business applications. It enables users to sign in to a user portal with credentials they configure in AWS SSO or use their existing corporate credentials to access all of their assigned accounts and applications from one place.

With AWS SSO, you can easily manage SSO access and user permissions to all of your accounts in AWS Organizations centrally. Using the AWS SSO application configuration wizard, Security Assertion Markup Language (SAML) 2.0 integrations can be created and extend SSO access to any SAML-enabled applications. You can also obtain credentials for use with the AWS SDK and CLI, and use preconfigured SAML integrations to sign into many cloud applications. By adding Azure AD Connect, and optionally Active Directory Federation Service (AD FS), you can sign in to Microsoft Office 365 and other cloud applications with credentials stored in AWS Managed Microsoft AD.

Integration with Office 365

AWS Managed Microsoft AD makes it possible and easy for you to build a Windows environment in AWS, synchronize your AWS Microsoft AD users into Microsoft Azure AD, and use Office 365, all without needing to create and manage AD domain controllers. For more information, see the AWS Security Blog post How to Enable Your Users to Access Office 365 with AWS Managed Microsoft AD.

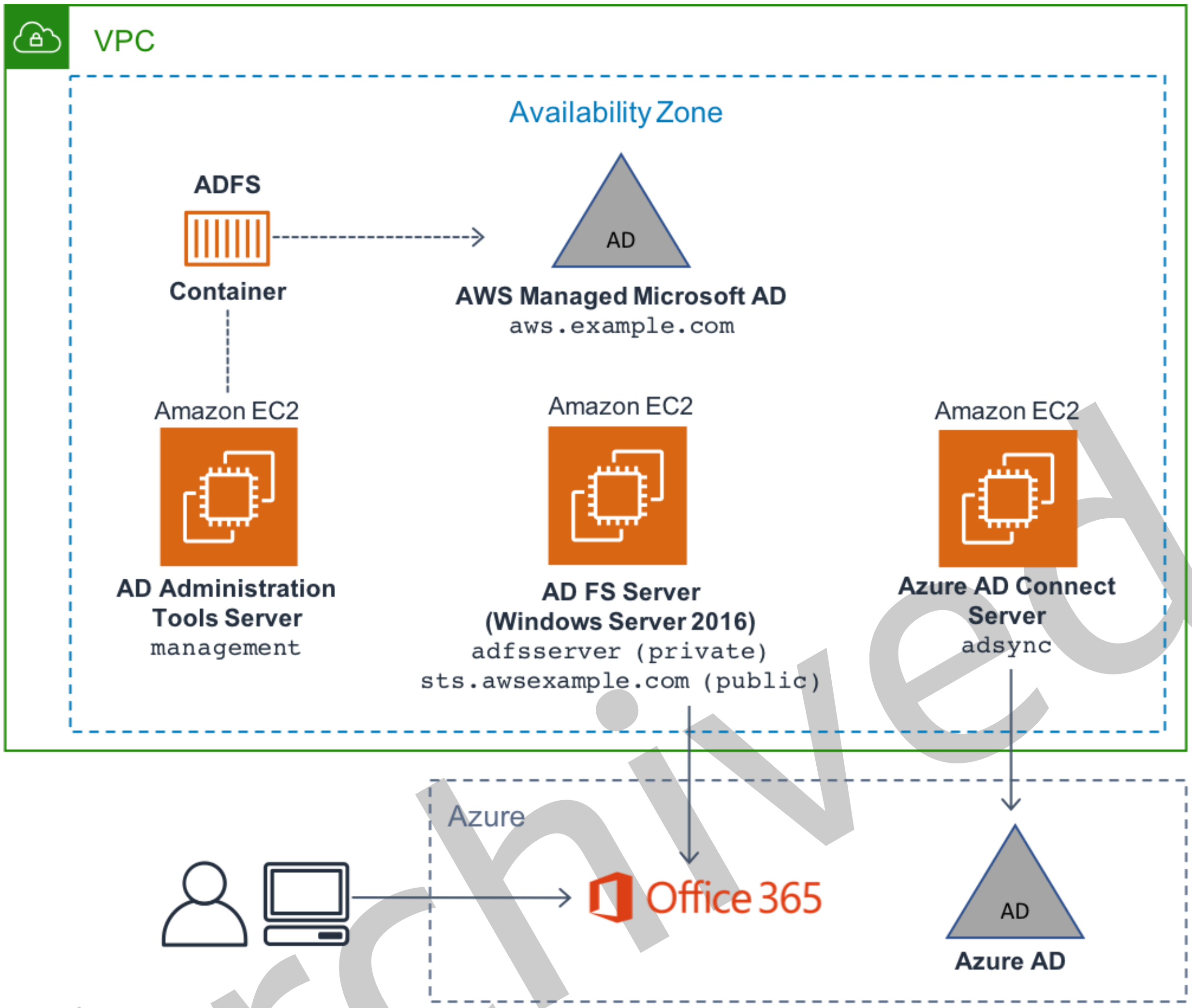


Figure 6: Enabling SSO with Office 365 and Azure AD

Infrastructure Security

VPC Networking

One of the critical components of an AWS environment is its networking infrastructure. With Amazon Virtual Private Cloud (Amazon VPC), you can deploy your AWS resources into a defined virtual network. Microsoft workloads are a natural fit and can function the same as they would in a traditional environment.

Each Amazon VPC consists of networking components to isolate and secure your network. This includes IPv4 and/or IPv6 private network (RFC 1918) addresses in the range of your choice (e.g., 10.0.0.0/16) that can be divided into subnets. Each subnet has a configurable Route Table and Network ACLs. You can enable gateways and endpoints to allow or restrict access to public and private resources. Amazon VPC also includes solutions for VPN and connecting to remote locations. DHCP options sets allow you to specify custom Domain and DNS settings in a way that is friendly to both Microsoft Active Directory and the many applications that rely on it. You can assign IPs statically or via DHCP, you can use AWS provided Public IPs, or even bring your own.

Amazon VPC provides isolation and familiar multi-layer networks, and also supports multiple connectivity options; some examples of this are:

- Resources in a subnet can be allowed to send, receive, or to send and receive traffic from the internet.
- VPC Peering allows the routing of traffic to other VPCs, Regions, and AWS Accounts.
- Multiple office locations can be connected via site-to-site VPN tunnels, using the AWS Cloud as a router.
- VPN-VPC connectivity is supported to extend your corporate data center to the AWS Cloud; allowing corporate users to interact with AWS resources and applications in a relatively transparent way.

Public and Private Subnets

In *Figure 7*, multiple subnets are defined to isolate the tiers of an infrastructure. Two *public subnets* route to an internet gateway. The Elastic Load Balancers (ELBs) are placed in this internet gateway because they are serving public internet traffic. All other subnets are *private* in that they do not have a direct route to the internet or an internet gateway, and resources are not assigned public IPs. A private subnet can be granted internet access with the use of a NAT gateway or device, while also restricting the public inbound connections.

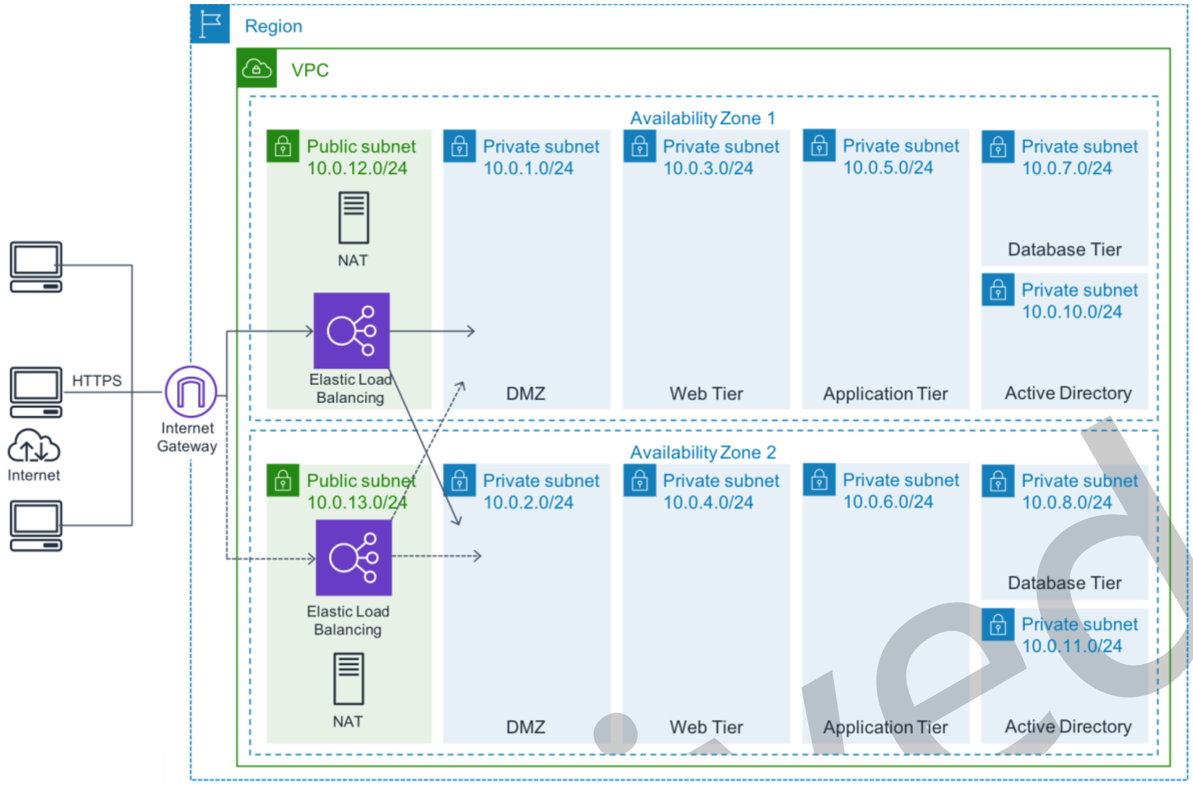

Region
VPC
Availability Zone 1
Public subnet 10.0.12.0/24
NAT
Elastic Load Balancing
Private subnet 10.0.1.0/24
DMZ
Private subnet 10.0.3.0/24
Web Tier
Private subnet 10.0.5.0/24
Application Tier
Private subnet 10.0.7.0/24
Database Tier
Private subnet 10.0.10.0/24
Active Directory
Availability Zone 2
Public subnet 10.0.13.0/24
Elastic Load Balancing
NAT
Private subnet 10.0.2.0/24
DMZ
Private subnet 10.0.4.0/24
Web Tier
Private subnet 10.0.6.0/24
Application Tier
Private subnet 10.0.8.0/24
Database Tier
Private subnet 10.0.11.0/24
Active Directory
HTTPS
Internet
Internet Gateway

Figure 7: Isolated environment with full internet access

Routing

Each VPC Subnet is associated with a route table, which defines the flow of traffic for the associated subnet. The route table determines if a subnet is public or private, and can be assigned to one or more subnets.

A well architected route table, will ensure applications that do not require inbound traffic directly from internet, such as domain controllers or backend SQL Servers, run in a private subnet.

As depicted, routing for each subnet is designed to allow communication between subnets so that the application tier can send traffic to the database tier, but there is no reason for the web tier to ever directly access the database tier.

In a more sophisticated network topology, connectivity from the private subnets can be allowed to directly access a remote data center network and can be used to integrate with existing internal systems. Using VPN-VPC connectivity can grant administrative access without the need to allow traffic to or from the public internet.

A common scenario for organizations running Microsoft workloads is to prevent direct access to the public internet; all traffic must be routed through a managed proxy or firewall. Once the VPN-VPC connectivity has been established, a default route (0.0.0.0/0) can be used to send all traffic back over the VPN to your local network. A proxy can also be configured directly in Windows redirecting traffic as needed, while still allowing your instance some access to the public internet in order to receive Windows Server or application updates, reducing the bandwidth requirement on your VPN.

VPC Endpoints

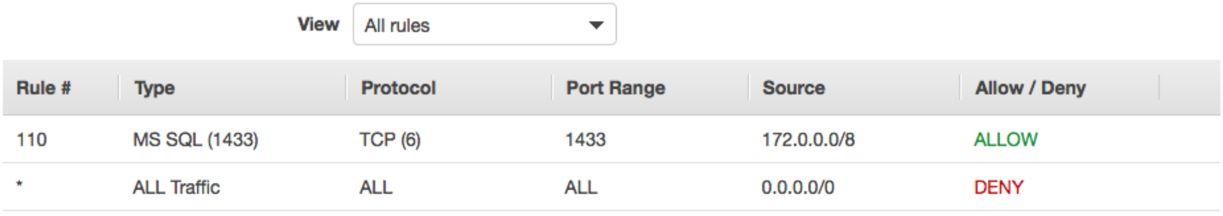
Another method of limiting the exposure of your environments to the public internet while still retaining access to AWS resources is with the use of VPC Endpoints. A VPC endpoint allows you to privately connect your VPC to supported AWS services without impacting availability or bandwidth constraints. Traffic to the supported AWS services via an endpoint remains on the Amazon network.

VPC Security

Network Access Control Lists (ACLs)

Network access control lists (ACLs) are attached to each VPC subnet and provide an optional layer of security via stateless filtering of network traffic, similar to a firewall. Network ACLs can be used for inbound or outbound traffic and provide an effective way to blacklist a CIDR block or individual IP addresses. Network ACLs consist of ordered rules that allow or deny traffic based upon IP protocol, service port, or source/destination IP address.

The figure below is an example of a network ACL rule that would allow administrative traffic to come inbound on RDP port 1433 from a specific CIDR block (172.0.0.0/8). See also Recommended Network ACL Rules for Your VPC.



View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
110	MS SQL (1433)	TCP (6)	1433	172.0.0.0/8	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Figure 8: Network ACL to allow range of IP addresses to access SQL Server port inbound

Security Groups

Security groups are one of the most critical tools available to isolate and secure VPC network traffic and the AWS infrastructure. Security groups allow for more granular control at the elastic network interface level, which allows you to further reduce the attack surface. Similar to network ACLs, security groups consist of a set of rules that allow traffic based upon IP protocol, service port, and source/destination IP address. Unlike network ACLs, security groups are stateful (stateless requires matching inbound and outbound rules) and act at the instance level rather than the subnet level.

To function correctly, all Amazon EC2 instances belong to one or more security groups. In Amazon VPC, every instance runs over a stateful firewall that runs on the host with all ports closed by default. If a security group rule does not exist to allow the traffic, the request is denied. The security group is responsible for opening up ingress and egress ports on that firewall. For example, you could have a security group called "*webtier"* that has rules to open port 80 and 443. You could then run 10 web servers that are all part of the "webtier" security group. If you later decide that you just want to support HTTPS traffic from the web server, you can simply remove the port 80 rule in the "webtier" security group. All 10 instances immediately respect this change and will deny traffic surfacing on HTTP port 80.

Security groups provide much more than firewall policies, you can use them to group and isolate the tiers of your environment. A security group can be created just for your SQL Servers, where you can specify to allow traffic on port 1433, but only from the security group containing your SharePoint servers. All of these topologies can be created with Amazon VPC in a manual or automated fashion. They can serve as the foundation for any Microsoft platform application built on AWS, thereby allowing you to control all network flow to and from your Microsoft workloads running in AWS.

Amazon EC2 Windows Firewall Usage

Since every Amazon EC2 instance in a VPC has a built-in stateful ingress and egress firewall that by default denies access to all ports, it is at the administrator's discretion which ports are opened using security group rules. We recommend that you use security groups but also leverage the Windows firewall as an added layer of security. For example, members of the same security groups can communicate with each other on any port. By placing AD-01 and AD-02 into a single AD-SG security group, the two modules are able to communicate with each other on any and all ports. For this reason, you can enforce a restrictive policy by using the Windows firewall to explicitly open only the necessary ports for AD.

In particular, the Windows firewall gives you audit details about packet drops, which may be important to meet your security policy or compliance requirements. VPC Flow Logs are also available to capture information about the traffic going to and from Elastic Network Interfaces (ENIs).

Remote Access

There are a few different solutions that can be used to gain remote access to your AWS resources and Microsoft Windows EC2 instances. The most straightforward way is to simply connect via Remote Desktop Protocol (RDP) over port 3389; a security group rule is needed to allow the inbound traffic.

> **NOTE:** If connecting over the public internet, only known good IP addresses should be allowed. Due to security risks, never open RDP to the entire internet—not even temporarily or for testing purposes.

Remote Desktop Gateway

Rather than connecting directly to administer each machine, Remote Desktop Gateway (RD Gateway) can be used to proxy connections. This Windows Server role service provides an additional layer of security and other controls that can help reduce or mitigate threats related to administrative access. RD Gateway offers authentication mechanisms based on SSL/TLS mutual authentication and the ability to apply authorization policies. For example, you can specify that clients must be members of specific Active Directory groups and which network resources those groups can connect to. Depending on business needs, you can place RD Gateway in a public subnet and configure it to allow remote access over an internet connection, through a VPN-VPC connection or AWS Direct Connect. RD Gateway uses HTTPS (port 443) to communicate, and then proxies to the backend server via the typical Remote Desktop Protocol (RDP).For more information, see Remote Desktop Gateway on the AWS Cloud: Quick Start Reference Deployment and RD Gateway on AWS Quick Start.

Bastion Server Environment

Another common security practice with regard to remote access is the use of a *bastion* server (or *jump*) box. In the same way that RD Gateway acts as a proxy, a bastion host is a special-purpose instance that administrators are required to connect and authenticate with before proceeding to connect and access the intended target.

Following best practice, place these proxy/bastion environments in a segregated network and use multi-factor authentication (MFA).

There are a number of options and third-party solutions for implementing a bastion environment. Most of these require additional management and resources that are always running. AWS provides a couple-managed services that can be used as a proxy/bastion environment to help limit the amount of time, and effort required.

Amazon WorkSpaces is a managed, secure cloud desktop service that can be used as a bastion environment. Users can connect to their personal WorkSpace using Web Access or with the WorkSpaces client. The client creates a secure connection using HTTPS (TCP 443) and PCoIP (UDP and TCP), Web Access uses port 443 over UDP and falls back to TCP if UDP is not available. Access to a WorkSpace can be restricted so that only trusted devices, from any IP address or only from certain IPs, are permitted.

To authenticate, WorkSpaces relies on the use of an AWS Directory Service so that users can retain the same credentials and use MFA, if enabled. Since WorkSpaces is joined to an Active Directory Domain, you can use the same tools to manage a WorkSpace that you use to manage on premises workstations/desktops.

Once connected, the WorkSpace user can connect/RDP and administer EC2 instances in the secure private VPC networks as needed. When the user is finished with their task, they can simply disconnect. WorkSpaces can be set to stop/sleep after a set time if a user is not connected, or they can be left in an always-on mode. When the user next connects, they connect to the same instance. If stopped, the WorkSpace starts/resumes in the same way a user logs in and out of a local workstation/desktop each day.

Amazon AppStream 2.0

Amazon AppStream 2.0 is a fully managed application streaming service that provides users with instant access to desktop applications via an HTML5 capable web browser or the AppStream 2.0 client. Applications and data remain on AWS; only encrypted pixels are streamed.

AppStream 2.0 allows you to build a more traditional bastion host, in that only your desired applications (RDP client/manager) are available to the user, rather than a full Windows Desktop experience. Each time a user connects, they do so to a dedicated streaming instance, once a pre-defined session limit is reached, or the user ends the session, the instance is terminated. By default, no data or settings persist between sessions; storage, application, user, and Region settings can be configured to persist.

There are a few different options to grant and control access to an AppStream 2.0 streaming instance. You can use an external identity provider (IdP) that supports SAML

2.0, or you can create and manage users directly in the AppStream 2.0 console via a local User Pool. AppStream 2.0 also supports joining the fleet instances to an Active Directory Domain. In this scenario, your AD users rely on AD FS and SAML for authentication and access.

For more information, see How to use Amazon AppStream 2.0 to reduce your bastion host attack surface on the AWS Security Blog.

WorkSpaces and AppStream 2.0 are great options for providing remote access to your environments for users that do not require daily or frequent access. Resources can be stopped and started on-demand, greatly reducing the cost of a secure bastion host environment. If you so choose, users can also securely connect from anywhere without the need of a VPN or even a computer. Remote access is not limited to RDP. SQL Server Management Studio (SSMS) and other management tools can be added to a WorkSpace or AppStream 2.0 Amazon Machine Image (AMI), so that Database Administrators and developers have access to the applications they need using a secure auditable connection. In the preceding remote access scenarios, AWS security groups are the key component in stopping administrators and/or malicious traffic from connecting directly to EC2 instances. By placing the proxy/bastion environment into its own unique security group, rules can be created in other security groups to accept traffic only on the necessary port (RDP 3389) from the proxy/bastion security group. Taken a step further, the bastion/proxy solution can be deployed in a separate AWS account, reducing the blast radius should any part of the environment become compromised.

AWS Systems Manager – Session Manager

Session Manager is an agent-based, fully managed AWS Systems Manager capability that lets you gain remote access to your Amazon EC2 (Windows or Linux) instances through an interactive browser-based shell or through the AWS CLI. Session Manager is secure and auditable without the need to open any inbound ports or maintain bastion hosts. Systems Manager supports VPC Endpoints using AWS PrivateLink. You do not need to expose any portion of your environment to external traffic.

AWS Systems Manager is a collection of capabilities for configuring and managing your instances, on-premises servers and virtual machines, and other AWS resources at scale. Systems Manager and Session Manager integrate with a number of other AWS services that provide auditing (AWS CloudTrail), logging (Amazon CloudWatch), notifications (Amazon Simple Notification Service [Amazon SNS]), and log storage (Amazon Simple Storage Service [Amazon S3]) with or without encryption. Access to the service can be granted or revoked through the use of AWS IAM policies, Administrators are able to control which IAM users can open a session, to which instances they can access, and on either a permanent or temporary schedule. Access to the service can also be granted via SAML-based 2.0 federation.

From the AWS Systems Manager console, you can open a session quickly with a single click. Once connected, an interactive PowerShell session opens running directly on the EC2 Windows Instance. A local Windows user is created and supplies the credentials for the interactive session while active.

Session Manager is great for quick or scripted administrative tasks that require an administrator to directly access an instance, to check the status of an application or service, or to ensure the health of an environment before and after a change. No additional resources are required outside of the EC2 instance, so administrators do not need to wait for anything else to start or to provide multiple sets of credentials. With the use of Session Manager, you no longer even need to enable Remote Desktop on your Windows Server. Session Manager provides a solution for organizations that have a need to remove the use of a key pair attached to EC2 Instances, preventing users from being able to obtain and decrypt the Windows Local Admin credentials.

Each of the preceding remote access methods has its own unique benefits. It may be useful to deploy one or more for access to different environments, for different teams, or from different geographical locations.

Configuration Management

One way to increase the ease in which Amazon EC2 instances are managed and kept compliant, is with AWS Systems Manager; which includes a unified interface that allows you to easily centralize operational data and automate tasks across your AWS resources.

Systems Manager also shortens the time to detect and resolve operational problems in your infrastructure, as it provides a complete view of your infrastructure performance and configuration, resource and application management simplification, and makes it easy to operate and manage infrastructure at scale.

In this section we cover AWS Systems Manager capabilities, as they relate to configuration management. When the capabilities are used together, they create a powerful management system that you can use to manage many aspects of your infrastructure, including security.

Configuration Compliance

Systems Manager Configuration Compliance can be used to scan your fleet of managed instances for patch compliance and configuration inconsistencies. You can collect and aggregate data from multiple AWS accounts and Regions, and then drill down into specific resources that aren't compliant. By default, Configuration Compliance displays compliance data about Patch Manager patching and State Manager associations. You can also customize the service and create your own compliance types based on your IT or business requirements.

Desired State Configuration

An important aspect of maintaining security compliance for EC2 instances is the ability to set and maintain desired configuration, and automatically remediate configuration drift, if unauthorized changes occur. For Windows systems, Windows PowerShell Desired State Configuration (DSC) is a management platform that enables infrastructure configuration as code. It can ensure that systems settings remain compliant, such as ensuring network ports remain set to a secure state, Windows Registry values follow best practices, or removing unauthorized applications or services from a server.

AWS Systems Manager includes the Windows PowerShell Desired State Configuration runtime framework and can execute compiled configuration files (.MOF file extension) written in declarative form against a set of Systems Manager Managed resources. To execute PowerShell DSC configuration files, Systems Manager provides an automation document with the name AWS-ApplyDSCMofs that executes the configuration file from an S3 bucket or from an HTTP-based location. For more information, see Run compliance enforcement and view compliant and non-compliant instances using AWS Systems Manager and PowerShell DSC on the AWS Management Tools Blog.

Patch Management

Keeping your patches up-to-date is an important part of protecting your infrastructure, because it helps to ensure operating system vulnerabilities are addressed when a vendor releases code fixes. You always have the option of pointing your servers at a WSUS server that is running on your own data center network. The key factor in deciding to deploy a self-managed WSUS on premise, is the amount of infrastructure that must be updated and its potential to saturate the bandwidth of the VPN-VPC or AWS Direct Connect connection. For example, having hundreds of servers in the VPC grabbing updates at the same time would likely saturate the available bandwidth in the connection.

Patch Manager is another AWS Systems Manager capability that can be used to automate the process of patching Amazon EC2 or on-premises managed instances. This capability enables you to scan instances for missing patches and apply them individually or to large groups of instances by using Amazon EC2 instance tags. For security patches, Patch Manager uses patch baselines that include rules for auto-approving patches within days of their release, as well as a list of approved and rejected patches. Security patches are installed from the default repository for patches configured for the instance this could be a custom WSUS or direct from Microsoft. You can install Microsoft application and/or Windows security patches on a regular basis by scheduling patching to run as a Systems Manager Maintenance Window task.

Maintenance Windows

Use Maintenance Windows to set up recurring schedules for managed instances to execute administrative tasks, such as installing patches and updates without interrupting business-critical operations.

State Management

Use Systems Manager State Manager to automate the process of keeping your managed instances in a defined state. You can use State Manager to ensure that your instances are bootstrapped with specific software at startup, joined to a Windows domain, or patched with specific software updates.

Managed Instances

A managed instance is any Amazon EC2 instance or on-premises machine, server, or virtual machine (VM) in your hybrid environment that is configured for Systems Manager. To set up managed instances, you must install the AWS Systems Manager agent (SSM Agent) (if not installed by default) and configure AWS Identity and Access Management (IAM) permissions. On-premises machines also require an activation code.

As described in this section, Systems Manager can be one of the key services for protecting your Windows-based Amazon EC2 instances. The following sections cover other services that can complement Systems Manager.

AWS Config and AWS Config Rules

State Manager and Desired State Configuration (DSC) provide a way to ensure your application and Windows Servers stay compliant. AWS Config provides a way to keep your AWS environment compliant, as well as a detailed view of AWS resources in your AWS account. These services together enable you to assess, audit, and evaluate the configurations of your Windows based AWS resources, and see how the configurations change over time.

To help you get started, AWS Config provides predefined rules, called managed rules, which are customizable; custom rules can also be created from scratch. The AWS Config Rules represent your ideal configuration settings.

Although AWS Config continuously tracks the configuration changes that occur among your resources, it also checks whether these changes violate any of the conditions in the rules. If a resource violates a rule, AWS Config flags the resource and the rule as *noncompliant.*

For example, when an EC2 instance is launched, AWS Config can evaluate whether the instance is on a compliant patch level using the ec2-managedinstance-patch-compliance-status-check managed rule. If noncompliant, AWS Config flags the instance and the rule as noncompliant.

AWS Config can also check all of your resources for account-wide requirements. AWS Config can check whether the EC2 instances in an account have Amazon CloudWatch Detailed Monitoring enabled (managed rule ec2-instance-detailed-monitoring-enabled), or whether an instance is an AWS Systems Manager managed instance (managed rule ec2-instance-managed-by-systems-manager). See also List of AWS Config Managed Rules.

AWS Config can remediate noncompliant resources that are evaluated by AWS Config Rules. AWS Config applies remediation using AWS Systems Manager Automation documents. These documents define the actions to be performed. AWS Config provides a set of managed automation documents with remediation actions. You can also create and associate custom automation documents with AWS Config Rules.

AWS CloudFormation

The provisioning of Microsoft workloads is just as important or even more important than configuration management. If done improperly or manually, you can introduce risks into an environment that can affect the confidentiality, integrity, and availability of an application or your customer’s data. When deploying a Microsoft resource in AWS, you can deploy these resources either individually and manually or as an AWS CloudFormation stack, which comprises all of the components that make up a solution or application (i.e. Infrastructure as Code).

AWS CloudFormation is a service that helps you model and set up your AWS resources, in a secure and repeatable fashion, so that you can spend less time managing those resources and more time focusing on your applications and security. A template is created that describes all of the desired AWS resources (such as Amazon EC2 instances or Amazon Relational Database Service (RDS) instances), AWS CloudFormation takes care of provisioning and configuring those resources. You do not need to individually create and configure AWS resources and figure out what's dependent on what; AWS CloudFormation takes care of it for you.

For Windows workloads and scalable web applications that also include a backend database, you might use an Auto Scaling Group, an Elastic Load Balancing load balancer, and an Amazon Relational Database Service database instance. Normally, each individual service is used to provision each resource. After the resources are created, they are configured to work together. All of these tasks add complexity and time to get your application up and running.

Instead, you can create or modify an existing Microsoft Windows stack running on Windows Server instances as a base for your AWS CloudFormation template. This approach provisions the Auto Scaling group, load balancer, and database. Once the stack is successfully created, the AWS resources are up and running. You can delete the stack just as easily, which deletes all of the resources in the stack. By using AWS CloudFormation, you easily manage a collection of resources as a single unit.

When you provision your infrastructure with AWS CloudFormation, the AWS CloudFormation template describes exactly what resources are provisioned and their settings. Because these templates are text files, you simply track differences in your templates to track changes to your infrastructure, similar to the way developers control revisions to source code. For example, you can use a version control system with your templates so that you know exactly what changes were made, who made them, and when. If at any point you need to reverse changes to your infrastructure, you can use a previous version of your template. A developer may not be granted IAM policies to launch new EC2 instance, but can be granted access to do so via specific templates. In this way, CloudFormation also provides another method to control AWS permissions.

Once a CloudFormation stack has been deployed, a user with privileges could access these resources and change the expected templatized settings. For example, a user can go into the AWS Console and change a security group associated with a set of EC2 instances. For this type of scenario, you can use CloudFormation Drift Detection to identify the difference between the expected configuration value of stack resources defined in CloudFormation templates and the actual configuration values of the resources in the corresponding CloudFormation stacks. This approach allows you to better manage your CloudFormation stacks and ensure consistency in your resource configuration. For more information, see CloudFormation Drift Detection on the AWS News Blog.

As a best practice, we recommend that all Microsoft workloads be provisioned using a repeatable manner as Infrastructure as Code (IaC). AWS CloudFormation provides a common language for you to describe and provision all of the infrastructure resources in your cloud environment. CloudFormation allows you to use a simple text file to model and provision, in an automated and secure manner, all of the resources required for your applications across all Regions and accounts. This file serves as the single source of truth for your cloud environment.

Data Encryption

This section covers the data encryption services available in AWS that directly impact the ability to secure Microsoft workloads. Data encryption is the process by which clear or plaintext data is transformed into an unreadable format using mathematical algorithms. Data encryption provides service providers and customers the ability to protect sensitive information, such as health records. Whether you are a data custodian or a data owner, ensure that the proper safeguards and controls are in place to protect the confidentiality of the data being entrusted to you.

Data should be protected at all times, both at rest and in transit. Data at rest is inactive, such as when stored on the appropriate storage service (e.g. Amazon EBS). Data in transit is active, such as when moving from client to server or between services. In AWS, there are various ways to protect customer data both at rest and in transit. It is critical that you take the necessary precautions to protect data when it is in both states.

Data at Rest

In this section, we cover the controls available to secure data at rest. The services that are covered include services that provide the storage capabilities, as well as the services that can be overlaid on top of the storage services to encrypt the data.

AWS Key Management Service (AWS KMS)

AWS KMS is a managed service that makes it easy for you to create and manage keys and control the use of encryption across a wide range of AWS services and in your applications. AWS KMS is a secure and resilient service that uses FIPS 140-2 validated hardware security modules to protect your keys. AWS KMS is integrated with AWS CloudTrail to provide you with logs of all key usage to help meet your regulatory and compliance needs.

If you are a developer who needs to encrypt data in your applications, use the AWS Encryption SDK with AWS KMS support to easily use and protect encryption keys. If you're an IT administrator looking for a scalable key management infrastructure to support your developers and their growing number of applications, use AWS KMS to reduce your licensing costs and operational burden. If you're responsible for proving data security for regulatory or compliance purposes, use AWS KMS to verify that data is encrypted consistently across the applications where it is used and stored.

The easiest way to get started using AWS KMS is to choose to encrypt your data within supported AWS services using AWS managed master keys that are automatically created in your account for each service. If you want full control over the management of your keys, including the ability to share access to keys across accounts or services, you can create your own customer-managed master keys in KMS. You can also use the master keys that you create in KMS directly within your own applications. Using AWS KMS, you can create encryption keys and define the policies that control how these keys can be used. AWS KMS supports AWS CloudTrail, so you can audit key usage to verify that keys are being used appropriately.

AWS KMS is seamlessly integrated with most other AWS services to make encrypting data in those services as easy as selecting a check box. In some cases, data is encrypted by default using keys that are stored in KMS but owned and managed by the AWS service. In many cases, the master keys are owned and managed by you within your account. Some services give you the choice of managing the keys yourself or you can allow the service to manage the keys on your behalf. See the list of AWS services currently integrated with KMS. See also the AWS Key Management Service Developer Guide for more information on how integrated services use AWS KMS.

AWS CloudHSM

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use your own encryption keys on AWS. In a similar way to KMS, AWS CloudHSM helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated Hardware Security Module (HSM) instances within AWS. AWS and AWS Marketplace partners offer a variety of solutions for protecting sensitive data within the AWS platform. However, for some applications and data subject to contractual or regulatory mandates for managing cryptographic keys, additional protection may be necessary. CloudHSM complements existing data protection solutions and allows you to protect your encryption keys within hardware security modules (HSMs) that are designed and validated to government standards for secure key management. CloudHSM allows you to securely generate, store, and manage cryptographic keys used for data encryption in a way that keys are accessible only by you.

Amazon EBS Encryption

Amazon EBS encryption offers a simple encryption solution for your EBS volumes without the need to build, maintain, and secure your own key management infrastructure. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume, the host, and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

Encryption is supported by all EBS volume types. You can expect the same IOPS performance on encrypted volumes as on unencrypted volumes, with a minimal effect on latency. You can access encrypted volumes the same way that you access unencrypted volumes. Encryption and decryption are handled transparently and they require no additional action from you or your applications.

Amazon S3 Encryption

Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS).

When you use server-side encryption, Amazon S3 encrypts an object before saving it to disk in its data centers and decrypts it when you download the objects. For more information about protecting data using server-side encryption and encryption key management, see Protecting Data Using Encryption.

Default encryption works with all existing and new S3 buckets. Without default encryption, to encrypt all objects stored in a bucket, you must include encryption information with every object storage request. Also set up an S3 bucket policy to reject storage requests that don't include encryption information.

SQL Server on AWS

SQL Server on AWS is available in a customer self-managed model and also an AWS managed offering. Regardless of the management model that is selected, securing SQL Server data in transit and at rest is important and often required for compliance reasons. In this section, we cover the available options for securing SQL Server data at rest for both the self-managed and AWS-managed options.

SQL Server on RDS

SQL Server is a relational database management system developed by Microsoft. Amazon RDS for SQL Server makes it easy to set up, operate, and scale SQL Server deployments in the cloud. With Amazon RDS, you can deploy multiple editions of SQL Server (2008 R2, 2012, 2014, 2016, and 2017) including Express, Web, Standard and Enterprise, in minutes with cost-efficient and re-sizable compute capacity. Amazon RDS frees you up to focus on application development by managing time-consuming database administration tasks including provisioning, backups, software patching, monitoring, and hardware scaling.

RDS Storage Level Encryption with AWS Key Management Service

You can encrypt your Amazon RDS DB instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instances. Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, Read Replicas, and snapshots.

Amazon RDS encrypted DB instances use the industry standard AES-256 encryption algorithm to encrypt your data on the server that hosts your Amazon RDS DB instances. Once your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You don't need to modify your database client applications to use encryption.

Amazon RDS encrypted DB instances provide an additional layer of data protection by securing your data from unauthorized access to the underlying storage. You can use

Amazon RDS encryption to increase data protection of your applications deployed in the cloud, and to fulfill compliance requirements for data at rest encryption.

To manage the keys used for encrypting and decrypting your Amazon RDS resources, you use AWS Key Management Service (KMS). For an Amazon RDS encrypted DB instance, all logs, backups, and snapshots are encrypted. A Read Replica of an Amazon RDS encrypted instance is also encrypted using the same key as the master instance when both are in the same Region. If the master and Read Replica are in different Regions, you encrypt using the encryption key for that Region.

If you disable the key for an encrypted DB instance, you cannot read from or write to that DB instance. When Amazon RDS encounters a DB instance encrypted by a key that Amazon RDS doesn't have access to, Amazon RDS puts the DB instance into a terminal state. In this state, the DB instance is no longer available and the current state of the database can't be recovered. To restore the DB instance, you must re-enable access to the encryption key for Amazon RDS, and then restore the DB instance from a backup. One of the benefits for using the KMS storage-based encryption is that most of the SQL Server Editions are supported, without requiring the use of SQL Server Enterprise. Encryption at rest is not available for DB instances running SQL Server Express Edition.

Application Level Encryption with Transparent Data Encryption (TDE)

Amazon RDS also supports using Transparent Data Encryption (TDE) to encrypt stored data on your DB instances running Microsoft SQL Server. TDE automatically encrypts data before it is written to storage, and automatically decrypts data when the data is read from storage. For more information, see Microsoft SQL Server Transparent Data Encryption Support for current supported SQL versions.

To enable Transparent Data Encryption for a DB instance that is running SQL Server, specify the **TDE** option in an Amazon RDS option group that is associated with that DB instance.

Transparent Data Encryption for SQL Server provides encryption key management by using a two-tier key architecture. A certificate, which is generated from the database master key, is used to protect the data encryption keys. The database encryption key performs the actual encryption and decryption of data on the user database. Amazon RDS backs up and manages the database master key and the TDE certificate.

You can use Transparent Data Encryption in scenarios where you must encrypt sensitive data in data files, and backups are obtained by a third party or when you need to address security-related regulatory compliance issues.

> **NOTE:** You cannot encrypt the system databases for SQL Server, such as the model or master database.

A detailed discussion of Transparent Data Encryption is beyond the scope of this guide, but you should understand the security strengths and weaknesses of each encryption algorithm and key. For information about Transparent Data Encryption for SQL Server, see Transparent Data Encryption (TDE) on the Microsoft website.

The TDE option is a persistent option and cannot be removed from an option group unless all DB instances and backups are disassociated from the option group. Once you add the TDE option to an option group, the option group can only be associated with DB instances that use TDE. For more information about persistent options in an option group, see Option Groups Overview.

SQL Server on EC2

SQL Server on EC2 also supports the AWS KMS and TDE-based encryptions used with Amazon RDS. However, in terms of KMS storage-based encryption, the biggest differentiation between Amazon RDS and Amazon EC2 is that since the EC2 instances are self-managed, administrators have file and operating system access to the instances hosting SQL Server. Since there is additional access, there is an additional attack surface. Therefore, complementary controls must be put in place to ensure that the environment is secure at every layer.

Amazon FSx for Windows File Server

All Amazon FSx file systems are encrypted at rest with keys-managed using AWS Key Management Service (AWS KMS). Data is automatically encrypted before being written to the file system, and automatically decrypted as it is read. These processes are handled transparently by Amazon FSx, so you don’t have to modify your applications.

Data in Transit

In this section, we cover the controls available to secure data in transit, including SSL/TLS certificates used to encrypt the transport of your traffic to and from AWS resources or into and out of AWS.

Amazon FSx for Windows File Server

Encryption of data in transit is supported on file shares that are mapped on a compute instance that supports SMB protocol 3.0 or newer. This includes all Windows versions starting from Windows Server 2012 and Windows 8, and all Linux clients with Samba client version 4.2 or newer. Amazon FSx for Windows File Server automatically encrypts data in transit (using SMB Kerberos session keys) as you access your file system without the need for you to modify your applications.

AWS Certificate Manager

AWS Certificate Manager is a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. SSL/TLS certificates are used to secure network communications and establish the identity of websites over the internet as well as resources on private networks.

AWS Certificate Manager removes the time-consuming manual process of purchasing, uploading, and renewing SSL/TLS certificates. With AWS Certificate Manager, you can quickly request a certificate, deploy it on AWS resources such as Elastic Load Balancers, Amazon CloudFront distributions, and APIs on Amazon API Gateway, and let AWS Certificate Manager handle certificate renewals. It also enables you to create private certificates for your internal resources and manage the certificate lifecycle centrally. Public and private SSL/TLS certificates provisioned through AWS Certificate Manager and used exclusively with ACM-integrated services, such as Elastic Load Balancing, Amazon CloudFront, and Amazon API Gateway, are free. You pay for the AWS resources you create to run your application. You pay a monthly fee for the operation of each private CA until you delete it, and for the private certificates you issue that are not used exclusively with ACM-integrated services.

Web Tier - Web Servers (e.g. Internet Information Services)

When encrypting web traffic, determine whether the traffic should be encrypted at the load balancer in front of the web servers or at the web servers themselves. In most cases, encrypting web traffic at the load balancer is sufficient; however, there are circumstances in which traffic must be encrypted at all hops, end-to-end. When this approach is required, encrypt both the connection between the client machine's browser and the load balancer with an SSL/TLS certificate, and the internal connection from the load balancer to the web servers. If required by your company's security policy, follow Microsoft's documentation for enabling SSL on IIS.

Database Tier

SQL Server on RDS

You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS Regions for all supported SQL Server editions.

When you create a SQL Server DB instance, Amazon RDS creates an SSL certificate for it. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

There are two ways to use SSL to connect to your SQL Server DB instance: The first and recommended approach is to force SSL for all connections—this happens transparently to the client, and the client doesn't have to do any work to use SSL. The second approach is to encrypt specific connections, which set up an SSL connection from a specific client computer. With this second approach, you must do work on the client to encrypt connections.

SQL Server on EC2

Since SQL Server on EC2 is self-managed, the required steps to configure encryption of data in transit is similar to the steps outlined for SQL Server on RDS and the customer’s responsibility. In short, the customer must install a certificate in the local machines certificate store, and then configure SQL Server to require all connections to be encrypted. See Enable Encrypted Connections to the Database Engine on the Microsoft documentation site for detailed instructions.

Increasing Availability

Active redundancy is commonly used to provide high availability for enterprise applications on the Microsoft platform. You can implement active redundancy by having an active-active configuration of application components. In this scenario, if one component fails, the other is available to keep the application running. This principle does not change when running on AWS, but the scale of the AWS infrastructure allows for an additional level of protection called Availability Zones. Availability Zones are fault separated areas of an AWS Region where you can place application infrastructure.

Availability Zones have different utility power providers and are located in geographically separated physical data centers. By separating the active-active components in two Availability Zones, you add additional resiliency that you couldn't achieve with a single data center.

A load balancer balances traffic between components in both Availability Zones. Load balancing provides a loose coupling between the application consumer and the infrastructure. This abstraction allows the infrastructure to grow or shrink without affecting the ability to serve clients. AWS provides a load balancing service called Elastic Load Balancing that can automatically scale based on the traffic it receives. You simply specify that you want a load balancer, but you don't need to worry about sizing it. You can place a load balancer into each Availability Zone when using Amazon VPC and assign a security group to it. This allows you to lock the frontend of your application to the load balancer's security group. Although your load balancer is capable of automatically scaling as needed, you must still consider how to scale the infrastructure running behind it.

AWS Auto Scaling can automatically scale up or scale down infrastructure based on a pattern, such as network traffic. You can also place PowerShell scripts on the instance so that when it is auto scaled, it can be configured properly in the environment before accepting traffic from the load balancer.

Auditing and Logging

In this section, we cover the auditing and logging options that can be used with your Microsoft workloads. It is a best practice to isolate application infrastructure entirely and tightly control access for both internal and external users and administrators. Having a central location to enforce policy for the perimeter of the application infrastructure is an important part of protecting a Microsoft environment.

Amazon CloudWatch

Amazon CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers. You can use CloudWatch to set high-resolution alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to optimize your applications, and ensure that they are running smoothly.

Amazon CloudWatch Logs

You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, Amazon Route 53, and other sources. You can then retrieve the associated log data from CloudWatch Logs.

EC2 Windows Logs

You can also set up a unified CloudWatch agent on your Windows instances. With the unified CloudWatch agent, you can monitor both in-guest metrics and also Windows Event Logs, IIS logs, or SQL logs. Amazon CloudWatch alarms can be configured to perform action/s, as well as triggering alerts to send notifications based on watched metrics or logged events, such as high CPU usage, low available memory, SQL errors, Windows Security audit events, IIS application pool or .Net warnings.

AWS Managed Microsoft AD

You can use either the AWS Directory Service console or APIs to forward AWS Managed Microsoft AD domain controller security event logs to Amazon CloudWatch Logs. This approach helps you to meet your security monitoring, audit, and log retention policy requirements by providing transparency of the security events in your directory. CloudWatch Logs can also forward these events to other AWS accounts, AWS services, or third-party applications. This approach makes it easier for you to centrally monitor and configure alerts to detect and respond proactively to unusual activities in near-real time.

Once enabled, you can then use the CloudWatch Logs console to retrieve the data from the log group you specified when you enabled the service. This log group contains the security logs from your domain controllers.

AWS CloudTrail

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting.

VPC Flow Logs

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon CloudWatch Logs and Amazon S3. After you've created a flow log, you can retrieve and view its data in the chosen destination.

Flow logs help you with a number of tasks; for example, to troubleshoot why specific traffic is not reaching an instance, which in turn helps you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your instance.

Security Information and Event Management

Amazon CloudWatch Events

Amazon CloudWatch Events delivers a near-real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. Using simple rules that you can quickly set up, you can match events and route them to one or more target functions or streams. CloudWatch Events becomes aware of operational changes as they occur. CloudWatch Events responds to these operational changes and takes corrective action as necessary, by sending messages to respond to the environment, activating functions, making changes, and capturing state information.

Governance

An important aspect of any security program is the governance and the structure that can be built into the system to segregate the different components and maintain accountability. In this section, we include services and solutions that can be used to set up a governance structure in your account.

Multi-Account Management

AWS Organizations

AWS Organizations offers policy-based management for multiple AWS accounts. With Organizations, you can create groups of accounts, automate account creation. Apply and manage policies for those groups. Organizations enables you to centrally manage policies across multiple accounts, without requiring custom scripts and manual processes.

Using AWS Organizations, you can create Service Control Policies (SCPs) that centrally control AWS service use across multiple AWS accounts. You can also use Organizations to help automate the creation of new accounts through APIs. Organizations helps simplify the billing for multiple accounts by enabling you to set up a single payment method for all the accounts in your organization through consolidated billing. AWS Organizations is available to all AWS customers at no additional charge.

AWS Landing Zone 2.0 and AWS Control Tower

As thousands of Enterprise customers matured in AWS, best practices were integrated into solutions to better manage a multi-account environment, along with security isolation, centralized logging, and shared services, such as authentication.

AWS Landing Zone 2.1

AWS Landing Zone is a solution that helps customers more quickly set up a secure, multi-account AWS environment based on AWS best practices. With the large number of design choices, setting up a multi-account environment can take a significant amount of time, involve the configuration of multiple accounts and services, and require a deep understanding of AWS services.

This solution helps save time by automating environment setup for running secure and scalable workloads while implementing an initial security baseline through the creation of core accounts and resources. It also provides a baseline environment to get started with a multi-account architecture, identity and access management, governance, data security, network design, and logging.

AWS Control Tower

AWS Control Tower automates the set-up of a baseline environment, or landing zone, that is a secure, well-architected multi-account AWS environment. The configuration of the landing zone is based on best practices that have been established by working with thousands of enterprise customers to create a secure environment that makes it easier to govern AWS workloads with rules for security, operations, and compliance.

As enterprises migrate to AWS, they typically have a large number of applications and distributed teams. They often want to create multiple accounts to allow their teams to work independently, while still maintaining a consistent level of security and compliance.

In addition, they use AWS's management and security services, like AWS Organizations, AWS Service Catalog and AWS Config, that provide granular controls over their workloads. They want to maintain this control, but they also want a way to centrally govern and enforce the best use of AWS services across all the accounts in their environment.

Control Tower automates the set-up of their landing zone and configures AWS management and security services based on established best practices in a secure, compliant, multi-account environment. Distributed teams are able to provision new AWS accounts quickly, while central teams have the peace of mind knowing that new accounts are aligned with centrally established, company-wide compliance policies. This gives you control over your environment, without sacrificing the speed and agility AWS provides your development teams.

Additional Considerations

Vulnerability Scans and Penetration Testing

A vulnerability scan is a process that can assess resources and interfaces to enumerate vulnerabilities present in the target. For example, a vulnerability scanner can detect that you left a port open to the entire internet or find that you have weak crypto algorithms enabled for an SSL interface. A penetration test is different in that it can find vulnerabilities, and then exploit them to determine their significance.

There is a process in place if you want to run a vulnerability scan or penetration test of your own Amazon EC2 Instances.

> **NOTE:** AWS customers are welcome to carry out security assessments or penetrations tests against their AWS infrastructure without prior approval for 8 services. See the penetration testing documentation for more information.
>
> **The AWS Acceptable Use Policy strictly forbids doing any kind of port scanning.**

For vulnerability scanning and penetration testing outside the approved services, you will need to contact AWS Security. Once you have your infrastructure standing, you can use a vulnerability scanning tool to evaluate the perimeter of your application and detect any vulnerabilities.

Antivirus

Best practice dictates the use of antivirus software on all domain controllers and any server that interacts with the domain controllers. See Microsoft recommendations for virus scanning. This same guidance applies to your servers running in AWS, as it is important to stop malware as early as possible. Antivirus software has an impact on performance for files that rapidly change, such as database storage files. For this reason, Microsoft provides guidance on files to exclude on the server when running antivirus scans. Enterprise Server Applications, including SQL Server and SharePoint, require additional sets of exclusions. Therefore, be sure to review these additional requirements when configuring the antivirus for your specific applications.

DDoS Mitigation

AWS Shield and AWS Shield Advanced

AWS Shield provides protection against distributed denial of service attacks (DDoS). A denial of service (DoS) attack is a malicious attempt to affect the availability of a targeted system, such as a website or application, to legitimate end users. Typically, attackers generate large volumes of packets or requests ultimately overwhelming the target system.

All AWS customers benefit from the automatic protections provided by AWS Shield Standard, which defends against the most common network and transport (layer 3 and 4) DDoS attacks.

AWS Shield Advanced is a subscription offering, that provides a higher level of protection at the network, transport, and application layers (3, 4, and 7), grants access to a 24x7 DDoS response team, as well as real-time metrics and reports. A DDoS attack can cause a significant increase in your bandwidth, AWS Shield Advanced also offers some cost protections resulting from related spikes in your AWS bill.

You may want to consider AWS Shield Advanced if your business or industry is a likely target, or if you have experienced a DDoS attack in the past. For more information, see the Best Practices for DDoS Resiliency whitepaper. To take full advantage of the benefits of AWS Shield, we recommend using Amazon CloudFront and Amazon Route 53.

Amazon CloudFront

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. When a user requests content (static or dynamic) that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (reducing any time delay), so that content is delivered with the best possible performance.

Beyond simply speeding up the delivery of your content, your user's requests take advantage of the AWS backbone network and CloudFront edge servers to give them a fast, safe, and reliable experience when they visit your website.

For additional security on top of HTTPS, you can add field-level encryption to protect specific data throughout system processing, so that only certain applications at your origin can see the data.

The use of CloudFront provides an additional layer of network infrastructure that help absorb DDoS attacks, limiting the impact to your environment and legitimate users.

Amazon Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service that can be used to direct public internet traffic as well as private VPC traffic. You can use Amazon Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. It includes many advanced features like traffic flow, latency-based routing, Geo DNS, and monitoring. Amazon Route 53 is one of the few services with an SLA of 100%. AWS will use commercially reasonable efforts to make Amazon Route 53 100% Available; "100% Available" means that Amazon Route 53 did not fail to respond to your DNS queries during a monthly billing cycle.

Securing your application, database, user credentials, and network are all important, but if DNS is left vulnerable, traffic may never reach your secure environment. Successful DNS attacks (spoofing, poisoning, and hijacking) could potentially re-route traffic somewhere else, or render the DNS server unable to reply resulting in traffic without a place to go. Amazon Route 53 uses shuffle sharding and anycast so end users can access your application, even if the DNS server is targeted by a DDoS attack.

Other Security Services

AWS WAF

AWS WAF is a web application firewall that helps protect your web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. AWS WAF provides monitoring and protection of HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront, or an Application Load Balancer. You can also use AWS WAF to block or allow requests based on conditions, such as the origin IP or the values of query strings. AWS WAF is included with AWS Shield Advanced.

AWS Firewall Manager

AWS Firewall Manager simplifies your AWS WAF administration and maintenance tasks across multiple accounts and resources. With Firewall Manager, you set up your firewall rules just once. The service automatically applies your rules across your accounts and resources, even as you add new resources. Firewall Manager is useful when you have a large number of resources that you want to protect with AWS WAF, or if you frequently add new resources that you want to protect.

Amazon GuardDuty

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as AWS CloudTrail, Amazon VPC Flow Logs, and DNS logs. It uses threat intelligence feeds, such as lists of malicious IPs and domains, and machine learning to identify unexpected and potentially unauthorized and malicious activity within your AWS environment. It also monitors AWS account access behavior for signs of compromise, such as unauthorized infrastructure deployments, like instances deployed in a Region that has never been used, or unusual API calls, like a password policy change to reduce password strength.

Amazon Inspector

Amazon Inspector is a security vulnerability assessment service that can automatically assesses resources for vulnerabilities or deviations from best practices, and then produces a detailed list of security findings prioritized by level of severity. Amazon Inspector includes a knowledge base of hundreds of rules mapped to common security standards and vulnerability definitions that are regularly updated by AWS security researchers.

Amazon Inspector also offers predefined software called an *agent* that you can optionally install in the operating system of the EC2 instances that you want to assess. The agent monitors the behavior of the EC2 instances, including network, file system, and process activity. It also collects a wide set of behavior and configuration data (telemetry).

AWS Security Hub

AWS Security Hub gives you a comprehensive view of your high-priority security alerts and compliance status across AWS accounts. Security Hub is the single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie, as well as from AWS Partner solutions.

Amazon Macie

Amazon Macie is a security service that uses machine learning to automatically discover, classify, and protect sensitive data in AWS.

Conclusion

As we discussed in this document, protecting the runtime environment of applications running on AWS is critical to providing a secure service for your customers. There should always be an emphasis on risk assessment, reducing attack surface, adhering to the principle of least privilege, and data protection. It should also be understood that security is a shared responsibility between you and AWS and that the safeguards and controls covered in this document are point in time to when the document was written. As with anything related to information security and the fact that it is ever evolving, our recommendation is to use this document as the first part of your Microsoft Workloads Security planning and supplement it with the additional reading we referenced, as well as any new developments or services available in the security space.

Contributors

Contributors to this document include:

- Alex Moore, Senior Solutions Architect, Amazon Web Services
- Rodney Bozo, Senior Solutions Architect, Amazon Web Services
- Siavash Irani, Senior Solutions Architect, Amazon Web Services
- Vladimir Provorov, Senior Solutions Architect, Amazon Web Services

Further Reading

For additional information, see:

- Windows on Amazon EC2 Security Guide
- AWS Cloud Security
- Vulnerability and Penetration Testing
- Microsoft Recommendation on Virus Scanning
- Amazon VPC User Guide
- AWS Security Best Practices
- Microsoft Implementing Least Privilege Administrative Models
- AWS Single Sign-On
- Microsoft Best Practices for Securing Active Directory
- Automate IIS and HttpErr Logs to Amazon CloudWatch Using EC2 Systems Manager

Document Revisions

Date	Description
August 2019	Updated to reflect current technologies.
August 2012	First publication

Appendix A: Subsystem Interface Port Mappings

Subsystem	Inbound Interface	Ports
NAT	WSS -> NAT	TCP 443
WSS	AD, RDG, TMG, WFE, SS, SQL -> WSS RDG -> WSS	TCP8531 TCP3389
AD	WSS, RDG, TMG, WFE, SS, SQL -> AD RDG -> AD	UDP123, TCP135, UDP135, TCP1024-65535, TCP137, UDP137, TCP139, TCP445, UDP445, TCP389, TCP636, TCP88, UDP88, TCP53, UDP53, ICMP8, ICMP13, ICMP15, ICMP17 TCP3389
RDG	VPN -> RDG	TCP443
TMG	RDG -> TMG ELB -> TMG	TCP3389 TCP80
WFE	RDG -> WFE TMG ->WFE	TCP3389 TCP80
SS	RDG -> SS WFE ->SS	TCP3389 TCP56737
SQL	RDG -> SQL SS -> SQL	TCP3389 TCP1433
ELB	Internet -> ELB	TCP443

* Active Directory incorporates many different services and protocols; for a complete list of ports used, see http://support.microsoft.com/kb/832017. Many ports must be open to authenticate between an instance and the primary domain controller. It is a common practice to open all ports on the AD server and use source traffic restrictions to control access. For example, you could open all ports on the AD server to the SQL Server's Security Group. This practice is not following our principle of least privilege though, and it is a far better practice to get familiar with the specific ports you need and enable those to be open and no more.

Appendix B: Module Interface Port Mappings

Module	Inbound Interface	Ports
SQL-01	SQL-02 -> SQL-01	TCP5022
SQL-02	SQL-01 -> SQL-02	TCP5022
AD-01	AD-02 -> AD01	TCP135, UDP135, TCP137, UDP137, UDP138, TCP139, TCP1024-65535, TCP445, UDP445, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, TCP53, UDP53
AD-02	AD-01 -> AD02	TCP135, UDP135, TCP137, UDP137, UDP138, TCP139, TCP1024-65535, TCP445, UDP445, TCP389, UDP389, TCP636, TCP3268, TCP3269, TCP88, UDP88, TCP53, UDP53

Appendix C: Subsystems with External Port Mappings

Security Group	Subsystem	Source IP/SG	Inbound Port Rules
nat-sg	NAT	wss-sg	TCP 443
wss-sg	WSS	ad-sg, rdg-sg, tmg-sg, wfe-sg, ss-sg, sql-sg -> wss-sg rdg-sg	TCP8531 TCP3389
ad-sg	AD	wss-sg, rdg-sg, tmg-sg, wfe-sg, ss-sg, sql-sg -> wss-sg rdg-sg	UDP123, TCP135, UDP135, TCP1024-65535, TCP137, UDP137, TCP139, TCP445, UDP445, TCP389, TCP636, TCP88, UDP88, TCP53, UDP53, ICMP8, ICMP13, ICMP15, ICMP17 TCP3389 TCP3389
rdg-sg	RDG	(IP range of Admins)	TCP443
tmg-sg	TMG	elb-sg rdg-sg	TCP80 TCP3389
wfe-sg	WFE	tmg-sg rdg-sg	TCP80 TCP3389
ss-sg	SS	wfe-sg rdg-sg	TCP56737 TCP3389
sql-sg	SQL	ss-sg rdg-sg	TCP1433 TCP3389
elb-sg	ELB	0.0.0.0/0	TCP443