User Guide

Amazon Elastic VMware Service

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Elastic VMware Service: User Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is Amazon Elastic VMware Service?

ⓘ **Note**

Amazon EVS is in public preview release and is subject to change.

You can use Amazon Elastic VMware Service (Amazon EVS) to deploy and run a VMware Cloud Foundation (VCF) environment directly on EC2 bare metal instances within Amazon Virtual Private Cloud (VPC).

Topics

Features of Amazon EVS

The following are key features of Amazon EVS:

Simplify and accelerate your migration to AWS

Remove migration friction and ensure operational consistency with subscription portability and automated deployment of VMware Cloud Foundation (VCF) in the cloud. Extend on-premises networks and migrate workloads without having to change IP addresses, retrain staff, or re-write operational runbooks.

Retain control of your VMware architecture in the cloud

Keep complete control over your VMware architecture and optimize a virtualization stack that meets the unique demands of your applications, including add-ons and third-party solutions.

Self-manage or leverage AWS Partners for a managed experience

Unlock choice and flexibility to self-manage, or leverage the expertise of AWS Partners to manage and operate your VCF environment on AWS to meet your business goals across talent, time, and costs.

Scale and protect your business from disruptions

Enhance scalability on the most secure, scalable, and resilient cloud for migrating and operating your VMware-based workloads.

Embrace AWS innovation to transform your applications and infrastructure

As an AWS-native service, Amazon EVS simplifies extending and expanding your VMware environment with 200+ services (including managed databases, analytics, serverless and containers, and generative AI) to transform your business.

Get started with Amazon EVS

To create your first Amazon EVS environment, see *Getting started*. In general, getting started with Amazon EVS involves completing the following steps.

1. Complete prerequisites. For more information, see *Setting up Amazon Elastic VMware Service*.
2. Create an Amazon EVS environment. During environment creation, Amazon EVS creates the required VLAN subnets using the CIDR ranges that you specify and adds hosts to the environment.
3. Customize VCF. Configure your environment in the vSphere user interface according to your needs. This may include setting up logins, policies, monitoring, and more.
4. Connect and migrate. Connect your environment to your on-premises data center and migrate your VCF workloads to Amazon EVS.

Accessing Amazon EVS

You can define and configure your Amazon EVS deployments using the following interfaces:

- Amazon EVS console - Provides a web interface to create Amazon EVS environments.
- AWS CLI - Provides commands for a broad set of AWS services and is supported on Windows, macOS, and Linux. For more information, see AWS Command Line Interface.

- AWS CloudFormation - Provides a specification for each resource type, such as `AWS::EVS::Environment`. You create a template using the resource specification, and CloudFormation takes care of provisioning and configuring the resources for you.

Concepts and components of Amazon EVS

> **Note**
>
> Amazon EVS is in public preview release and is subject to change.

This section explains some key Amazon EVS concepts and components.

Amazon EVS environment

An Amazon EVS *environment* is a logical container for VMware Cloud Foundation (VCF) resources, such as vSphere hosts, vSAN, NSX, and SDDC Manager. An environment contains a consolidated VCF domain with a vSphere cluster that hosts the components for managing, monitoring, and instantiating the VCF software stack. Each environment directly maps to an SDDC Manager appliance. For more information, see the section called “Architecture”.

Amazon EVS host

An Amazon EVS *host* is a VMware ESXi host that runs on Amazon EC2 bare metal instances.

Service access subnet

The *service access subnet* is a standard VPC subnet that allows Amazon EVS to access the VCF deployment. During Amazon EVS environment creation, you specify the VPC and subnet for Amazon EVS to use for service access.

When you create an Amazon EVS environment, Amazon EVS provisions elastic network interfaces into the service access subnet to facilitate management connectivity to VCF appliances and ESXi hosts. This connectivity is required for Amazon EVS to be able to deploy, manage, and monitor the VCF deployment.

Amazon EVS VLAN subnet

An *Amazon EVS VLAN subnet* is an Amazon VPC subnet that is managed by Amazon EVS. VLAN subnets provide VPC connectivity for Amazon EVS hosts, and VCF appliances such as VMware NSX, VMware HCX, and VMware vCenter Server. Each VLAN subnet has a VLAN tag to allow VLAN network traffic to be segmented logically.

Amazon EVS creates all of the VLAN subnets that the service uses when the Amazon EVS environment is created. You provide the CIDR block inputs that the VLAN subnets use. You should ensure that your VLAN subnet CIDR blocks are properly sized according to the number of hosts that will be configured, taking into account future scaling needs. For more information, see the section called "Amazon EVS networking considerations".

⚠ **Important**

Amazon EVS VLAN subnets can only be created during Amazon EVS environment creation, and cannot be modified after the environment is created. You must ensure that the VLAN subnet CIDR blocks are properly sized before creating the environment. You will not be able to add VLAN subnets after the environment is deployed.

⚠ **Important**

EC2 security group rules are not enforced on Amazon EVS elastic network interfaces that are attached to VLAN subnets. To control traffic to and from VLAN subnets, you must use a network access control list.

ⓘ **Note**

Amazon EVS does not support IPv6 at this time.

Host VMkernel management VLAN subnet

The *host VMkernel management VLAN subnet* separates management traffic from user traffic, and allows for remote management of hosts. The EVS host management vmkernel network interface connects to this subnet.

vMotion VLAN subnet

The *vMotion VLAN subnet* logically segments VMware vMotion traffic, and is used during a vMotion process to move virtual machines between hosts.

vSAN VLAN subnet

The *vSAN VLAN subnet* is used by VMware vSAN to separate traffic related to vSAN's storage operations from other network traffic.

VTEP VLAN subnet

The *VTEP VLAN subnet* uses VMware NSX virtual tunnel endpoints (VTEP) to encapsulate and decapsulate overlay network traffic for the Amazon EVS ESXi hosts.

Edge VTEP VLAN subnet

The *Edge VTEP VLAN subnet* is a specialized VTEP VLAN subnet that is dedicated for NSX Edge appliance overlay traffic. This VLAN is used for overlay communication between NSX edges and ESXi hosts.

VM management VLAN subnet

The *VM management VLAN subnet* is used for managing virtual appliances, including NSX Manager, vCenter Server, and SDDC Manager.

HCX uplink VLAN subnet

The *HCX uplink VLAN subnet* is used for communication between the HCX Interconnect (HCX-IX) and HCX Network Extension (HCX-NE) appliances, and enables the creation of the HCX service mesh uplink.

NSX uplink VLAN subnet

The *NSX uplink VLAN subnet* is used for connecting your NSX overlay networks to the rest of your VPC and any other external networks that you configure. The NSX uplink VLAN subnet is configured on the NSX Edge node uplinks.

Expansion VLAN subnet

The *expansion VLAN subnet* can be used to enable additional VCF-supported functions, such as NSX Federation. Amazon EVS creates two expansion VLAN subnets during environment creation.

VMware NSX

VMware NSX is a software-defined networking (SDN) platform that enables network virtualization. Amazon EVS uses VMware NSX to create and manage the overlay network where VMware Cloud Foundation (VCF) appliances and workloads run. Amazon EVS deploys a pair of active/standby NSX Edge nodes, along with an NSX overlay network. Amazon EVS automatically configures all of the NSX routing and uplinks on your behalf as part of deployment. For more information about common NSX concepts, see Key Concepts in the *VMware NSX Installation Guide*.

VMware Hybrid Cloud Extension (HCX)

VMware Hybrid Cloud Extension (VMware HCX) is an application mobility platform designed for simplifying application migration, rebalancing workloads, and optimizing disaster recovery across data centers and clouds. You can use HCX to migrate your VMware-based workloads to Amazon EVS.

You can configure connectivity for VMware HCX using AWS Direct Connect with an associated transit gateway, or using an AWS Site-to-Site VPN attachment to a transit gateway. For more information, see *Migration*.

Amazon EVS architecture

Note

Amazon EVS is in public preview release and is subject to change.

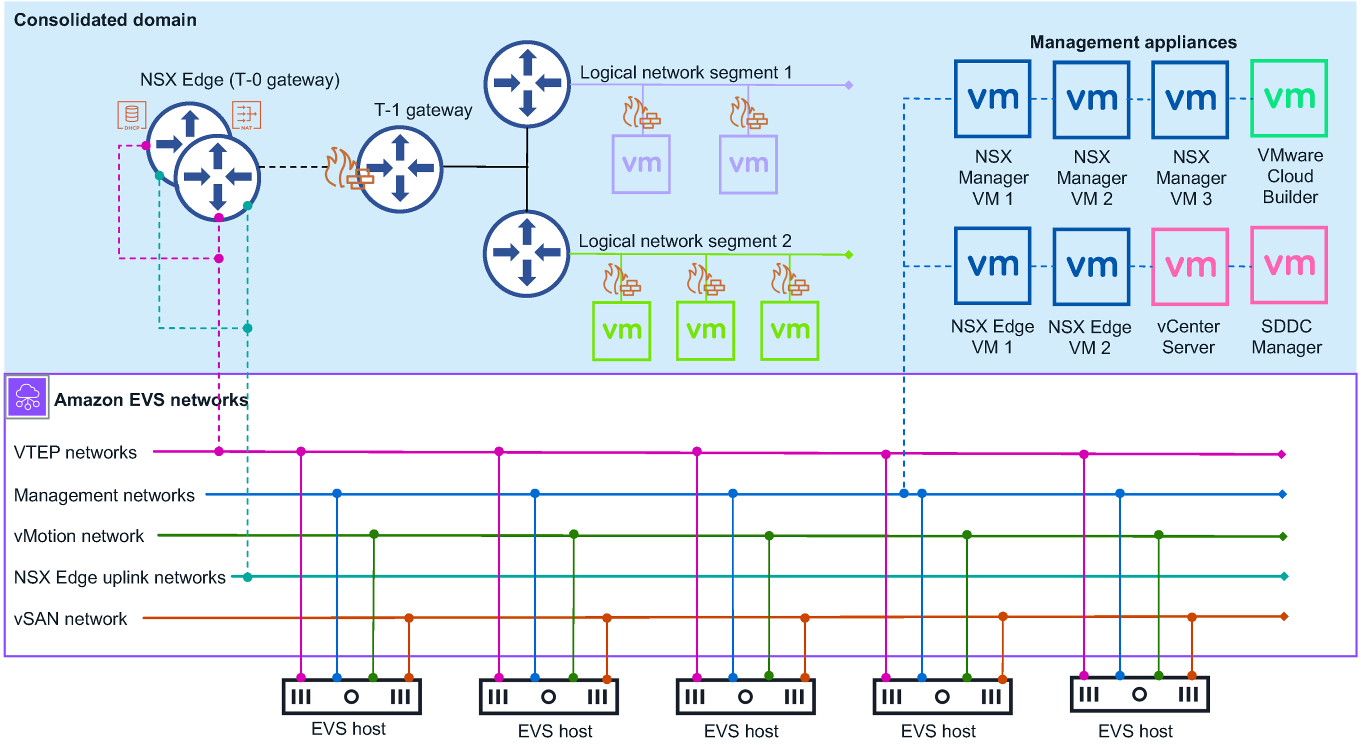
Amazon EVS implements a VMware Cloud Foundation (VCF) consolidated architecture model. In this model, VCF management components and customer workloads run together on a consolidated domain. The Amazon EVS environment is managed from a single vCenter Server with vSphere resource pools that provide isolation between management and customer workloads.

The consolidated domain that Amazon EVS deploys contains the following VCF management components:

- ESXi hosts
- vCenter Server instance

- SDDC Manager
- vSAN datastore
- Three-node NSX Manager cluster
- vSphere cluster
- NSX Edge cluster

The following diagram shows an example Amazon EVS architecture that's been deployed in an Amazon EVS environment, and shows how the components in the environment are connected. In the diagram, the Amazon EVS environment with a consolidated domain architecture is shaded in blue. The underlying Amazon EVS network topology is illustrated within the solid purple line.



Network topology

An Amazon EVS environment has two separate management network layers:

Amazon VPC

> The Amazon VPC and the Amazon EVS VLAN subnets that are created in the VPC during environment creation form the underlay network for your VCF deployment. This infrastructure

provide connectivity for NSX overlay networks, host management, vMotion, and VSAN. Amazon VPC Route Server enables dynamic routing between the underlay network and overlay networks. For more information, see the section called "Concepts and components".

Note

Amazon EVS VLAN subnets are used to facilitate VCF underlay communication only. Guest virtual machines running customer workloads must be deployed on NSX overlay networks. Deployment of guest virtual machines on the Amazon EVS VLAN subnet underlay network is not supported.

VMware NSX overlay network

Amazon EVS configures an NSX overlay network on your behalf as part of the deployment. You can configure additional NSX overlay networks to achieve network isolation between different workloads or applications within your Amazon EVS environment. For more information, see Overlay Design for VMware Cloud Foundation in the VMware Cloud Foundation product documentation.

Note

Amazon EVS supports only one tier-0 gateway for an Active/Standby NSX Edge cluster with two NSX Edge nodes. This tier-0 gateway connects to and advertises all overlay networks that you configure for use with Amazon EVS.

The two network layers are connected by an Active/Standby NSX Edge cluster with two NSX Edge nodes. The NSX Edge nodes enable communication over the VPC between virtual machines in the VLANs, as well as internet connectivity, and private connectivity using AWS Direct Connect or AWS Site-to-Site VPN with a transit gateway.

Amazon EVS networking considerations

The management network requires the following networking resource configurations. You provide these inputs during Amazon EVS environment creation. For more information, see the section called "Concepts and components".

- An Amazon VPC. Ensure that your VPC IPv4 CIDR block is sized appropriately to accommodate the required VPC subnet and Amazon EVS VLAN subnets that Amazon EVS provisions during environment creation. For more information, see the section called “Amazon EVS VLAN subnet”.

  > **Note**
  >
  > Amazon EVS does not support IPv6 at this time.

- A service access subnet in your VPC. Amazon EVS uses this subnet to maintain a persistent connection to your SDDC Manager appliance. For more information, see service access subnet.

  > **Note**
  >
  > Amazon EVS only supports Single-AZ deployments at this time. All VPC subnets that Amazon EVS uses must exist in a single Availability Zone in a Region where the service is available.

  > **Note**
  >
  > All VPC subnets require associated route tables that are configured according to your organization’s networking requirements.

- A primary DNS server IP address and a secondary DNS server IP address in the VPC’s DHCP option set to resolve host IP addresses. Amazon EVS also requires that you create a DNS forward lookup zone with A records and a reverse lookup zone with PTR records for each VCF management appliance and Amazon EVS host in your deployment. For more information, see the section called “DNS server configuration”.
- Amazon EVS VLAN subnet CIDR blocks for each VLAN subnet that Amazon EVS provisions for you during environment creation. CIDR blocks must have a minimum size of /28 netmask and a maximum size of /24 netmask. CIDR blocks must be non-overlapping.
- An Amazon VPC Route Server instance with Route Server propagation enabled.
- Two Route Server endpoints in the service access subnet.
- Two Route Server peers that peer the NSX Edge nodes that Amazon EVS provisions with Route Server endpoints.

Tier-0 gateway

The tier-0 gateway handles all north-south traffic between the logical and physical networks and is created on the NSX overlay network. This tier-0 gateway is created as a part of Amazon EVS deployment.

> **Note**
>
> Amazon EVS supports only one tier-0 gateway for an Active/Standby NSX Edge cluster with two NSX Edge nodes.

Tier-1 gateway

The tier-1 gateway handles east-west traffic between routed network segments within an environment and is created on the NSX overlay network. The tier-1 gateway has downlink connections to segments and uplink connections to the tier-0 gateway. You can create and configure additional Tier-1 gateways if you need them.

NSX Edge cluster

Amazon EVS uses the NSX Manager interface to deploy an NSX Edge cluster with two NSX Edge nodes that run in Active/Standby mode. This NSX Edge cluster provides the platform on which the Tier-0 and Tier-1 gateways run, along with IPsec VPN connections and their BGP routing machinery.

Amazon EVS resources

Amazon EVS provisions the following AWS resources during environment creation. These resources appear in the VPC that you allow Amazon EVS to access, and are visible in the AWS Management Console and AWS CLI after they are created.

> **Important**
>
> Modification of these resources outside of the Amazon EVS console and API could impact the availability and stability of your Amazon EVS environment.

- Amazon EVS elastic network interfaces that enable connectivity to your VCF appliances and hosts.

- Amazon EVS ESXi hosts that run on Amazon EC2 bare metal instances. For more information, see the section called “Amazon EVS host”.

> ⚠ **Important**
>
> Your Amazon EVS environment must have a minimum of 4 hosts and no more than 16 hosts. Amazon EVS only support environments with 4-16 hosts.

- Amazon EVS VLAN subnets that connect your VPC to VCF appliances. For more information, see the section called “Amazon EVS VLAN subnet”.

Setting up Amazon Elastic VMware Service

Note

Amazon EVS is in public preview release and is subject to change.

To use Amazon EVS, you will need to configure other AWS services, as well as set up your environment to meet VMware Cloud Foundation (VCF) requirements.

Topics

Sign up for AWS

If you don’t have an AWS account, complete the following steps to create one.

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

Create an IAM user

1. Sign in to the IAM console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   > **Note**
   >
   > We strongly recommend that you adhere to the best practice of using the `Administrator` IAM user below and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.

2. In the navigation pane, choose **Users** and then choose **Create user**.
3. For **User name**, enter `Administrator`.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter `Administrators`.
10. Choose **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    > **Note**
    >
    > You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.

12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.

14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM Entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access Management and Example Policies.

Create an IAM role to delegate Amazon EVS permission to an IAM user

You can use roles to delegate access to your AWS resources. With IAM roles, you can establish trust relationships between your trusting account and other AWS trusted accounts. The trusting account owns the resource to be accessed, and the trusted account contains the users who need access to the resource.

After you create the trust relationship, an IAM user or an application from the trusted account can use the AWS Security Token Service (AWS STS) `AssumeRole` API operation. This operation provides temporary security credentials that enable access to AWS resources in your account. For more information, see Create a role to delegate permissions to an IAM user in the *AWS Identity and Access Management User Guide*.

Follow these steps to create an IAM role with a permissions policy that allows access to Amazon EVS operations.

> **Note**
>
> Amazon EVS does not support the use of an instance profile to pass an IAM role to an EC2 instance.

Example

IAM console

1. Go the IAM console.

2. On the left menu, choose **Policies**.
3. Choose **Create policy**.
4. In the policy editor, create a permissions policy that enables Amazon EVS operations. For an example policy, see the section called “Create and manage an Amazon EVS environment”. To view all available Amazon EVS actions, resources, and condition keys, see Actions in the *Service Authorization Reference*.
5. Choose **Next**.
6. Under **Policy name**, enter a meaningful policy name to identify this policy.
7. Review the permissions defined in this policy.
8. (Optional) Add tags to help identify, organize, or search for this resource.
9. Choose **Create policy**.
10. On the left menu, choose **Roles**.
11. Choose **Create role**.
12. For **Trusted entity type**, choose AWS account.
13. Under **An AWS account** , specify the account that you want to perform Amazon EVS actions and choose **Next**.
14. On the **Add permissions** page, select the permissions policy that you previously created and choose **Next**.
15. Under **Role name**, enter a meaninful name to identify this role.
16. Review the trust policy and ensure that the correct AWS account is listed as the principal.
17. (Optional) Add tags to help identify, organize, or search for this resource.
18. Choose **Create role**.

AWS CLI

1. Copy the following contents to a trust policy JSON file. For the principal ARN, replace the example AWS account ID and `service-user` name with your own AWS account ID and IAM user name.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/service-user"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Create the role. Replace `evs-environment-role-trust-policy.json` with your trust policy file name.

   ```
   aws iam create-role \
     --role-name myAmazonEVSEnvironmentRole \
     --assume-role-policy-document file://"evs-environment-role-trust-policy.json"
   ```

3. Create a permissions policy that enables Amazon EVS operations and attach the policy to the role. Replace `myAmazonEVSEnvironmentRole` with your role name. For an example policy, see the section called "Create and manage an Amazon EVS environment". To view all available Amazon EVS actions, resources, and condition keys, see Actions in the *Service Authorization Reference*.

   ```
   aws iam attach-role-policy \
     --policy-arn arn:aws:iam::aws:policy/AmazonEVSEnvironmentPolicy \
     --role-name myAmazonEVSEnvironmentRole
   ```

Sign up for an AWS Business, AWS Enterprise On-Ramp, or AWS Enterprise Support plan

Amazon EVS requires that customers are enrolled in an AWS Business, AWS Enterprise On-Ramp, or AWS Enterprise Support plan to receive continuous access to Amazon EVS technical support and architectural guidance. If you have business-critical workloads, we recommend enrolling in AWS Enterprise On-Ramp or AWS Enterprise Support plans. For more information, see Compare AWS Support Plans.

Important

Amazon EVS environment creation fails if you do not sign up for an AWS Business, AWS Enterprise On-Ramp, or an AWS Enterprise Support plan.

Check quotas

To enable Amazon EVS environment creation, ensure that your account has the required minimum account-level quotas. For more information, see the section called “Service quotas”.

⚠ **Important**

Amazon EVS environment creation fails if the host count per EVS environment quota value is not at least 4.

Plan VPC CIDR sizes and configure VPC components

To enable Amazon EVS environment creation, you must provide Amazon EVS with a VPC that contains a subnet and enough IP address space for Amazon EVS to create the VLAN subnets that connect to your VCF appliances. For more information about VPC creation requirements, see the section called “Create a VPC with subnets and route tables”.

Main route table

Amazon EVS subnets are implicitly associated to your VPC’s main route table when they are created. To enable connectivity to dependent services such as DNS or on-premises systems for successful environment deployment, you must configure the VPC’s main route table to allow traffic to these systems. For more information about main route table configuration for Amazon EVS, see the section called “Configure the VPC main route table”.

DHCP option set

Amazon EVS uses your VPC’s DHCP option set to retrieve the following:

- Domain Name System (DNS) servers that are used to resolve host IP addresses.
- Network Time Protocol (NTP) servers that are used to avoid time synchronization issues in the SDDC.

To successfully deploy an Amazon EVS environment, your VPC’s DHCP option set must have the following DNS settings:

- A primary DNS server IP address and a secondary DNS server IP address in the DHCP option set.

- A DNS forward lookup zone with A records for each VCF management appliance and Amazon EVS host in your deployment as detailed in the section called “Create an Amazon EVS environment”.
- A reverse lookup zone with PTR records for each VCF management appliance and Amazon EVS host in your deployment as detailed in the section called “Create an Amazon EVS environment”.

For NTP configuration, you can use the the default Amazon NTP address `169.254.169.123`, or another IPv4 address that you prefer.

For more information about Amazon EVS supported options for DNS and NTP server configuration, see the section called “Configure DNS and NTP servers using the VPC DHCP option set”.

Create VPC Route Server infrastructure

Amazon EVS uses Amazon VPC Route Server to to enable BGP-based dynamic routing to your VPC underlay network. For information about setting up Route Server for Amazon EVS usage, see the section called “Set up a VPC Route Server instance with endpoints and peers”.

Create a transit gatway for on-premesis connectivity

You can configure connectivity for your on-premises data center to your AWS infrastructure using AWS Direct Connect with an associated transit gateway, or using an AWS Site-to-Site VPN attachment to a transit gateway. For more information, see the section called “(Optional) Configure on-premises network connectivity”.

Create an Amazon EC2 Capacity Reservation

Amazon EVS launches Amazon EC2 i4i.metal instances that represent ESXi hosts in your Amazon EVS environment. To ensure that you have sufficient i4i.metal instance capacity available when you need it, we recommend that you request an Amazon EC2 Capacity Reservation. You can create a Capacity Reservation at any time, and you can choose when it starts. You can request a Capacity Reservation for immediate use, or you can request a Capacity Reservation for a future date. For more information, see Reserve compute capacity with EC2 On-Demand Capacity Reservations in the *Amazon Elastic Compute Cloud User Guide*.

Set up the AWS CLI

The AWS CLI is a command line tool for working with AWS services, including Amazon EVS. It is also used to authenticate IAM users or roles for access to the Amazon EVS virtualization environment and other AWS resources from your local machine. To provision AWS resources from the command line, you need to obtain an AWS access key ID and secret key to use in the command line. Then you need to configure these credentials in the AWS CLI. For more information, see Set up the AWS CLI in the *AWS Command Line Interface User Guide for Version 2*.

Create an Amazon EC2 key pair

Amazon EVS uses an Amazon EC2 key pair that you provide during environment creation to connect to your hosts. To create a key pair, follow the steps on Create a key pair for your Amazon EC2 instance in the Amazon Elastic Compute Cloud User Guide.

Prepare your environment for VMware Cloud Foundation (VCF)

Before you deploy your Amazon EVS environment, your environment must meet VMware Cloud Foundation (VCF) infrastructure requirements. For detailed VCF prerequisites, see the Planning and Preparation Workbook in the VMware Cloud Foundation product documentation.

You should also familiarize yourself with VCF 5.2.1 requirements. For more information, see the VCF 5.2.1 release notes

> **Note**
>
> Amazon EVS only supports VCF version 5.2.1.x at this time.

Acquire VCF license keys

To use Amazon EVS, you need to provide a VCF solution key and a vSAN license key. The VCF solution key must have at least 256 cores. The vSAN license key must have at least 110 TiB of vSAN capacity. For more information about VCF licenses, see Managing License Keys in VMware Cloud Foundation in the *VMware Cloud Foundation Administration Guide*.

ⓘ **Note**

Your VCF license will be available to Amazon EVS across all AWS Regions for license compliance. Amazon EVS does not validate license keys. To validate license keys, visit Broadcom support.

ⓘ **Note**

Use the SDDC Manager user interface to manage VCF solution and vSAN license keys. Amazon EVS requires that you maintain valid VCF solution and vSAN license keys in SDDC Manager for the service to function properly. If you manage these keys using the vSphere Client, you must make sure that those keys also appear in the licensing screen of the SDDC Manager user interface.

VMware HCX prerequisites

You can use VMware HCX to migrate your existing VMware-based workloads to Amazon EVS. Before you use VMware HCX with Amazon EVS, make sure that the following prerequiste tasks have been completed.

- Before you can use VMware HCX with Amazon EVS, minimum network underlay requirements must be met. For more information, see Network Underlay Minimum Requirements in the *VMware HCX User Guide*.
- VMware NSX is installed and configured in your environment. For more information, see the VMware NSX Installation Guide.
- VMware HCX is activated and installed in your environment. For more information, see About Getting Started with VMware HCX in the *Getting Started with VMware HCX Guide*.

Getting started with Amazon Elastic VMware Service

Note

Amazon EVS is in public preview release and is subject to change.

Use this guide to get started with Amazon Elastic VMware Service (Amazon EVS). You'll learn how to create an Amazon EVS environment with hosts within your own Amazon Virtual Private Cloud (VPC).

After you're finished, you'll have an Amazon EVS environment that you can use to migrate your VMware vSphere-based workloads to the AWS Cloud.

Important

To get started as simply and quickly as possible, this topic includes steps to create a VPC, and specifies minimum requirements for DNS server configuration and Amazon EVS environment creation. Before creating these resources, we recommend that you plan out your IP address space and DNS record setup that meets your requirements. You should also familiarize yourself with VCF 5.2.1 requirements. For more information, see the VCF 5.2.1 release notes.

Important

Amazon EVS only supports VCF version 5.2.1.x at this time.

Topics

- Create an Amazon EVS environment
- Verify Amazon EVS environment creation
- Explicitly associate Amazon EVS VLAN subnets to a VPC route table
- (Optional) Configure transit gateway route tables and Direct Connect prefixes for on-premises connectivity
- Create a network ACL to control Amazon EVS VLAN subnet traffic
- Retrieve VCF credentials and access VCF management appliances
- Configure the EC2 Serial Console
- Clean up
- Next steps

Prerequisites

Before getting started, you must complete the Amazon EVS prerequisite tasks. For more information, see *Setting up Amazon Elastic VMware Service*.

Create a VPC with subnets and route tables

> **Note**
>
> The VPC, subnets, and Amazon EVS environment must all be created in the same account. Amazon EVS does not support cross-account sharing of VPC subnets or Amazon EVS environments.

1. Open the Amazon VPC console.
2. On the VPC dashboard, choose **Create VPC**.
3. For **Resources to create**, choose **VPC and more**.
4. Keep **Name tag auto-generation** selected to create Name tags for the VPC resources, or clear it to provide your own Name tags for the VPC resources.
5. For **IPv4 CIDR block**, enter an IPv4 CIDR block. A VPC must have an IPv4 CIDR block. Ensure that you create a VPC that is adequately sized to accommodate the Amazon EVS subnets. For more information, see the section called “Amazon EVS networking considerations”

> **Note**
>
> Amazon EVS does not support IPv6 at this time.

6. Keep **Tenancy** as `Default`. With this option selected, EC2 instances that are launched into this VPC will use the tenancy attribute specified when the instances are launched. Amazon EVS launches bare metal EC2 instances on your behalf.
7. For **Number of Availability Zones (AZs)**, choose **1**.

> **Note**
>
> Amazon EVS only supports Single-AZ deployments at this time.

8. Expand **Customize AZs** and choose the AZ for your subnets.

> **Note**
>
> You must deploy in an AWS Region where Amazon EVS is supported. For more information about Amazon EVS Region availability, see *Endpoints and quotas*.

9. (Optional) If you need internet connectivity, for **Number of public subnets**, choose **1**.
10. For **Number of private subnets**, choose **1**.
11. To choose the IP address ranges for your subnets, expand **Customize subnets CIDR blocks**.

> **Note**
>
> Amazon EVS VLAN subnets will also need to be created from this VPC CIDR space. Ensure that you leave enough space in the VPC CIDR block for the VLAN subnets that the service requires. For more information, see the section called “Amazon EVS networking considerations”

12. (Optional) To grant internet access over IPv4 to resources, for **NAT gateways**, choose **In 1 AZ**. Note that there is a cost associated with NAT gateways. For more information, see Pricing for NAT gateways.

> **Note**
>
> Amazon EVS requires the use of a NAT gateway to enable outbound internet connectivity.

13. For **VPC endpoints**, choose **None**.

> **Note**
>
> Amazon EVS does not support gateway VPC endpoints for Amazon S3 at this time. To enable Amazon S3 connectivity, you must set up an interface VPC endpoint using AWS PrivateLink for Amazon S3. For more information, see AWS PrivateLink for Amazon S3 in the *Amazon Simple Storage Service User Guide*.

14. For **DNS options**, keep the defaults selected. Amazon EVS requires your VPC to have DNS resolution capability for all VCF components.
15. (Optional) To add a tag to your VPC, expand **Additional tags**, choose **Add new tag**, and enter a tag key and a tag value.
16. Choose **Create VPC**.

> **Note**
>
> During VPC creation, Amazon VPC automatically creates a main route table and implicitly associates subnets to it by default.

Configure the VPC main route table

Amazon EVS subnets are implicitly associated to your VPC's main route table when they are created. To enable connectivity to dependent services such as DNS or on-premises systems for successful environment deployment, you must configure the main route table to allow traffic to these systems. For more information about managing subnet route tables, see Manage subnet route tables in the *Amazon VPC User Guide*.

After the Amazon EVS environment deploys, you can configure explicit route table associations to enable connectivity through a custom route table. For more information, see Replace the main route table in the *Amazon VPC User Guide*.

⚠ **Important**

Amazon EVS supports the use of a custom route table only after the Amazon EVS environment is created. Custom route tables should not be used during Amazon EVS environment creation, as this may result in connectivity issues.

Configure DNS and NTP servers using the VPC DHCP option set

Amazon EVS uses your VPC's DHCP option set to retrieve the following:

- Domain Name System (DNS) servers that are used to resolve host IP addresses.
- Network Time Protocol (NTP) servers that are used to avoid time synchronization issues in the SDDC.

You can create a DHCP option set using the Amazon VPC console or AWS CLI. For more information, see Create a DHCP option set in the *Amazon VPC User Guide*.

To enable DNS connectivity for successful environment deployment, you must first configure the VPC's main route table to allow DNS traffic. For more information, see the section called "Configure the VPC main route table".

DNS server configuration

You can enter IPv4 addresses of up to four Domain Name System (DNS) servers. You can use Route 53 as your DNS server provider, or you can provide your own custom DNS servers. For more information about configuring Route 53 as your DNS service for an existing domain, see Making Route 53 the DNS service for a domain that's in use.

ⓘ **Note**

Using both Route 53 and a custom Domain Name System (DNS) server may cause unexpected behavior.

Note

Amazon EVS does not support IPv6 at this time.

To successfully deploy an environment, your VPC’s DHCP option set must have the following DNS settings:

- A primary DNS server IP address and a secondary DNS server IP address in the DHCP option set.
- A DNS forward lookup zone with A records for each VCF management appliance and Amazon EVS host in your deployment as detailed in the section called “Create an Amazon EVS environment”.
- A reverse lookup zone with PTR records for each VCF management appliance and Amazon EVS host in your deployment as detailed in the section called “Create an Amazon EVS environment”.

For more information about configuring DNS servers in a DHCP option set, see Create a DHCP option set.

Note

If you use custom DNS domain names defined in a private hosted zone in Route 53, or use private DNS with interface VPC endpoints (AWS PrivateLink), you must set both the `enableDnsHostnames` and `enableDnsSupport` attributes to `true`. For more information, see DNS attributes for your VPC.

NTP server configuration

NTP servers provide the time to your network. You can enter the IPv4 addresses of up to four Network Time Protocol (NTP) servers. For more information about configuring NTP servers in a DHCP option set, see Create a DHCP option set.

Note

Amazon EVS does not support IPv6 at this time.

You can specify the Amazon Time Sync Service at IPv4 address `169.254.169.123`. By default, the Amazon EC2 instances that Amazon EVS deploys use the Amazon Time Sync Service at IPv4 address `169.254.169.123`.

For more information about NTP servers, see RFC 2123. For more information about the Amazon Time Sync Service, see Set the time for your instance in the *Amazon EC2 User Guide*.

(Optional) Configure on-premises network connectivity

You can configure connectivity for your on-premises data center to your AWS infrastructure using AWS Direct Connect with an associated transit gateway, or using an AWS Site-to-Site VPN attachment to a transit gateway. AWS Site-to-Site VPN creates an IPsec VPN connection to the transit gateway over the internet. AWS Direct Connect creates an IPsec VPN connection to the transit gateway over a private dedicated connection. After the Amazon EVS environment is created, you can use either option to connect your on-premises data center firewalls to the VMware NSX environment.

To enable connectivity to on-premises systems for successful environment deployment, you must configure the VPC’s main route table to allow traffic to these systems. For more information, see the section called “Configure the VPC main route table”.

After the Amazon EVS environment is created, you must update the transit gateway route tables with the VPC CIDRs created within the Amazon EVS environment. For more information, see the section called “(Optional) Configure transit gateway route tables and Direct Connect prefixes for on-premises connectivity”.

For more information about setting up an AWS Direct Connect connection, see AWS Direct Connect gateways and transit gateway associations. For more information about using AWS Site-to-Site VPN with AWS Transit Gateway, see AWS Site-to-Site VPN attachments in Amazon VPC Transit Gateways in the *Amazon VPC Transit Gateway User Guide*.

Note

Amazon EVS does not support connectivity via an AWS Direct Connect private virtual interface (VIF), or via an AWS Site-to-Site VPN connection that terminates directly into the underlay VPC.

Set up a VPC Route Server instance with endpoints and peers

Amazon EVS uses Amazon VPC Route Server to to enable BGP-based dynamic routing to your VPC underlay network. You must specify a route server that shares routes to at least two route server endpoints in the service access subnet. The peer ASN configured on the route server peers must match, and the peer IP addresses must be unique.

⚠ **Important**

When enabling Route Server propagation, ensure that all route tables being propagated have at least one explicit subnet association. BGP route advertisement fails if the route table does have an explicit subnet association.

For more information about setting up VPC Route Server, see the Route Server get started tutorial.

ⓘ **Note**

For Route Server peer liveness detection, Amazon EVS only support the default BGP keepalive mechanism. Amazon EVS does not support multi-hop Bidirectional Forwarding Detection (BFD).

ⓘ **Note**

We recommend that you enable persistent routes for the route server instance with a persist duration between 1-5 minutes. If enabled, routes will be preserved in the route server's routing database even if all BGP sessions end. For more information, see Create a route server in the *Amazon VPC User Guide*.

ⓘ **Note**

If you are using a NAT gateway or a transit gateway, ensure that your route server is configured correctly to propagate NSX routes to the VPC route table(s).

Create an Amazon EVS environment

Important

To get started as simply and quickly as possible, this topic includes steps to create an Amazon EVS environment with default settings. Before creating an environment, we recommend that you familiarize yourself with all settings and deploy an environment with the settings that meet your requirements. Environments can only be configured during initial environment creation. Environments cannot be modified after you've created them. For an overview of all possible Amazon EVS environment settings, see the Amazon EVS API Reference Guide.

Note

You environment ID will be available to Amazon EVS across all AWS Regions for VCF license compliance needs.

Note

Amazon EVS environments must be deployed into the same Region and Availability Zone as the VPC and VPC subnets.

Complete this step to create an Amazon EVS environment with hosts and VLAN subnets.

Example

Amazon EVS console

1. Go to the Amazon EVS console.

   Note

   Ensure that the AWS Region shown in the upper right of your console is the AWS Region that you want to create your environment in. If it's not, choose the dropdown next to the AWS Region name and choose the AWS Region that you want to use.

> **Note**
>
> Amazon EVS operations triggered from the Amazon EVS console will not generate CloudTrail events.

2. In the navigation pane, choose **Environments**.
3. Choose **Create environment**.
4. On the **Validate Amazon EVS requirements** page, do the following.

   a. Check that the AWS Support requirement and the service quota requirements are met. For more information about Amazon EVS support requirements, see the section called "Sign up for an AWS Business, AWS Enterprise On-Ramp, or AWS Enterprise Support plan". For more information about Amazon EVS quota requirements, see the section called "Service quotas".

   b. (Optional) For **Name**, enter an environment name.

   c. For **Environment version**, choose your VCF version. Amazon EVS currently only supports version 5.2.1.x.

   d. For **Site ID**, enter your Broadcom Site ID.

   e. For **VCF Solution key**, enter a VCF solution key. This license key cannot be in use by an existing environment.

   > **Note**
   >
   > The VCF solution key must have at least 256 cores.

   > **Note**
   >
   > Your VCF license will be available to Amazon EVS across all AWS Regions for license compliance. Amazon EVS does not validate license keys. To validate license keys, visit Broadcom support.

> **Note**
>
> Amazon EVS requires that you maintain a valid VCF solution key in SDDC Manager for the service to function properly. If you manage the VCF solution key using the vSphere Client post-deployment, you must ensure that the keys also appears in the licensing screen of the SDDC Manager user interface.

f. For **vSAN license key**, enter a vSAN license key. This license key cannot be in use by an existing environment.

> **Note**
>
> The vSAN license key must have at least 110 TiB of vSAN capacity.

> **Note**
>
> Your VCF license will be available to Amazon EVS across all AWS Regions for license compliance. Amazon EVS does not validate license keys. To validate license keys, visit Broadcom support.

> **Note**
>
> Amazon EVS requires that you maintain a valid vSAN license key in SDDC Manager for the service to function properly. If you manage the vSAN license key using the vSphere Client post-deployment, you must ensure that the keys also appears in the licensing screen of the SDDC Manager user interface.

g. For **VCF license terms**, check the box to confirm that you have purchased and will continue to maintain the required number of VCF software licenses to cover all physical processor cores in the Amazon EVS environment. Information about your VCF Software in Amazon EVS will be shared with Broadcom to verify license compliance.

h. Choose **Next**.

5. On the **Specify host details** page, complete the following steps 4 times to add 4 hosts to the environment. Amazon EVS environments require 4 hosts for initial deployment.

   a. Choose **Add host details**.

   b. For **DNS hostname**, enter the host name for the host.

   c. For **instance type**, choose the EC2 instance type.

      > ⚠ **Important**
      >
      > Do not stop or terminate EC2 instances that Amazon EVS deploys. This action results in data loss.

      > ⓘ **Note**
      >
      > Amazon EVS only supports i4i.metal EC2 instances at this time.

   d. For **SSH key pair**, choose an SSH key pair for SSH access into the host.

   e. Choose **Add host**.

6. On the **Configure networks and connectivity** page, do the following.

   a. For **VPC**, choose the VPC that you previously created.

   b. For **Service access subnet**, choose the private subnet that was created when you created the VPC.

   c. For **Security group -*optional*** , you can choose up to 2 security groups that control communication between the Amazon EVS control plane and VPC. Amazon EVS uses the default security group if no security group is chosen.

      > ⓘ **Note**
      >
      > Ensure that the security groups that you choose provide connectivity to your DNS servers and Amazon EVS VLAN subnets.

   d. Under **Management connectivity**, enter the CIDR blocks to be used for the Amazon EVS VLAN subnets.

> **Important**
>
> Amazon EVS VLAN subnets can only be created during Amazon EVS environment creation, and cannot be modified after the environment is created. You must ensure that the VLAN subnet CIDR blocks are properly sized before creating the environment. You will not be able to add VLAN subnets after the environment is deployed. For more information, see the section called "Amazon EVS networking considerations".

e. Under **Expansion VLANs**, enter the CIDR blocks for additional Amazon EVS VLAN subnets that can be used to expand VCF capabilities within Amazon EVS, such as enabling NSX Federation.

> **Note**
>
> Ensure that the VLAN CIDR blocks that you provide are properly sized within the VPC. For more information, see the section called "Amazon EVS networking considerations".

f. Under **Workload/VCF connectivity**, enter the CIDR block for the NSX uplink VLAN, and choose 2 VPC Route Server peer IDs that peer to Route Server endpoints over the NSX uplink.

> **Note**
>
> Amazon EVS requires a VPC Route Server instance that is associated with 2 Route Server endpoints and 2 Route Server peers. This configuration enables dynamic BGP-based routing over the NSX uplink. For more information, see the section called "Set up a VPC Route Server instance with endpoints and peers".

g. Choose **Next**.

7. On the **Specify Management DNS hostnames** page, do the following.

   a. Under **Management appliance DNS hostnames**, enter the DNS hostnames for the virtual machines to host VCF management appliances. If using Route 53 as your DNS provider, also choose the hosted zone that contains your DNS records.

b. Under **Credentials**, choose whether you'd like to use the AWS managed KMS key for Secrets Manager or a customer managed KMS key that you provide. This key is used to encrypt the VCF credentials that are required to use SDDC Manager, NSX Manager, and vCenter appliances.

> **Note**
>
> There are usage costs associated with customer managed KMS keys. For more information, see the AWS KMS pricing page.

c. Choose **Next**.

8. (Optional) On the **Add tags** page, add any tags that you would like to be assigned to this environment and choose **Next**.

> **Note**
>
> Hosts created as part of this environment will receive the following tag: `DoNotDelete-EVS-environmentid-hostname`.

> **Note**
>
> Tags that are associated with the Amazon EVS environment do not propagate to underlying AWS resources such as EC2 instances. You can create tags on underlying AWS resources using the respective service console or the AWS CLI.

9. On the **Review and create** page, review your configuration and choose **Create environment**.

> **Important**
>
> During environment deployment, Amazon EVS creates the EVS VLAN subnets and implicitly associates them with the main route table. After the deployment completes, you must explicitly associate the Amazon EVS VLAN subnets with a route table for NSX connectivity purposes. For more information, see the section called "Explicitly associate Amazon EVS VLAN subnets to a VPC route table".

Note

Amazon EVS deploys a recent bundled version of VMware Cloud Foundation which may not include individual product updates, known as async patches. Upon completion of this deployment, we strongly recommend that you review and update individual products using Broadcom’s Async Patch Tool (AP Tool) or SDDC Manager in-product LCM automation. NSX upgrades must be done outside of SDDC Manager.

Note

Environment creation can take several hours.

AWS CLI

1. Open a terminal session.
2. Create an Amazon EVS environment. Below is a sample `aws evs create-environment` request.

Important

Before running the `aws evs create-environment` command, check that all Amazon EVS prerequisites have been met. Environment deployment fails if prerequisites have not been met. For more information about Amazon EVS support requirements, see the section called “Sign up for an AWS Business, AWS Enterprise On-Ramp, or AWS Enterprise Support plan”. For more information about Amazon EVS quota requirements, see the section called “Service quotas”.

Important

During environment deployment, Amazon EVS creates the EVS VLAN subnets and implicitly associates them with the main route table. After the deployment completes, you must explicitly associate the Amazon EVS VLAN subnets with a route table for NSX connectivity purposes. For more information, see the section called “Explicitly associate Amazon EVS VLAN subnets to a VPC route table”.

Note

Amazon EVS deploys a recent bundled version of VMware Cloud Foundation which may not include individual product updates, known as async patches. Upon completion of this deployment, we strongly recommend you review and update individual products using Broadcom’s Async Patch Tool (AP Tool) or SDDC Manager in-product LCM automation. NSX upgrades must be done outside of SDDC Manager.

Note

Environment deployment can take several hours.

- For `--vpc-id`, specify the VPC that you previously created with a minimum IPv4 CIDR range of /22.
- For `--service-access-subnet-id`, specify the unique ID of the private subnet that was created when you created the VPC.
- For `--vcf-version`, Amazon EVS currently only supports VCF 5.2.1.x.
- With `--terms-accepted`, you confirm that you have purchased and will continue to maintain the required number of VCF software licenses to cover all physical processor cores in the Amazon EVS environment. Information about your VCF software in Amazon EVS will be shared with Broadcom to verify license compliance.
- For `--license-info`, enter your VCF solution key and vSAN license key.

  Note

  The VCF solution key must have at least 256 cores. The vSAN license key must have at least 110 TiB of vSAN capacity.

Note

Amazon EVS requires that you maintain a valid VCF solution key and vSAN license key in SDDC Manager for the service to function properly. If you manage these license keys using the vSphere Client post-deployment, you must ensure that they also appear in the licensing screen of the SDDC Manager user interface.

Note

The VCF solution key and vSAN license key cannot be in use by an existing Amazon EVS environment.

- For `--initial-vlans` specify the CIDR ranges for the Amazon EVS VLAN subnets that Amazon EVS creates on your behalf. These VLANs are used to deploy VCF management appliances.

Important

Amazon EVS VLAN subnets can only be created during Amazon EVS environment creation, and cannot be modified after the environment is created. You must ensure that the VLAN subnet CIDR blocks are properly sized before creating the environment. You will not be able to add VLAN subnets after the environment is deployed. For more information, see the section called “Amazon EVS networking considerations”.

- For `--hosts`, specify host details for the hosts that Amazon EVS requires for environment deployment. Include DNS hostname, EC2 SSH key name, and EC2 instance type for each host.

Important

Do not stop or terminate EC2 instances that Amazon EVS deploys. This action results in data loss.

Note

Amazon EVS only supports i4i.metal EC2 instances at this time.

- For `--connectivity-info`, specify the 2 VPC Route Server peer IDs that you created in the previous step.

Note

Amazon EVS requires a VPC Route Server instance that is associated with 2 Route Server endpoints and 2 Route Server peers. This configuration enables dynamic BGP-based routing over the NSX uplink. For more information, see the section called "Set up a VPC Route Server instance with endpoints and peers".

- For `--vcf-hostnames`, enter the DNS hostnames for the virtual machines to host VCF management appliances.
- For `--site-id`, enter your unique Broadcom site ID. This ID allows access to the Broadcom portal, and is provided to you by Broadcom at the close of your software contract or contract renewal.
- (Optional) For `--region`, enter the Region that your environment will be deployed into. If the Region isn't specified, your default Region is used.

```
aws evs create-environment \
--environment-name testEnv \
--vpc-id vpc-1234567890abcdef0 \
--service-access-subnet-id subnet-01234a1b2cde1234f \
--vcf-version VCF-5.2.1 \
--terms-accepted \
--license-info "{
      \"solutionKey\": \"00000-00000-00000-abcde-11111\",
      \"vsanKey\": \"00000-00000-00000-abcde-22222\"
    }" \
   --initial-vlans "{
      \"vmkManagement\": {
        \"cidr\": \"10.10.0.0/24\"
      },
      \"vmManagement\": {
        \"cidr\": \"10.10.1.0/24\"
      },
      \"vMotion\": {
        \"cidr\": \"10.10.2.0/24\"
      },
      \"vSan\": {
        \"cidr\": \"10.10.3.0/24\"
      },
      \"vTep\": {
        \"cidr\": \"10.10.4.0/24\"
      },
      \"edgeVTep\": {
        \"cidr\": \"10.10.5.0/24\"
      },
      \"nsxUplink\": {
        \"cidr\": \"10.10.6.0/24\"
      },
      \"hcx\": {
        \"cidr\": \"10.10.7.0/24\"
      },
      \"expansionVlan1\": {
        \"cidr\": \"10.10.8.0/24\"
      },
      \"expansionVlan2\": {
          \"cidr\": \"10.10.9.0/24\"
      }
    }" \
--hosts "[
    {
      \"hostName\": \"esx01\",
      \"keyName\": \"sshKey-04-05-45\”,
      \"instanceType\": \"i4i.metal\"
    },
    {
      \"hostName\": \"esx02\",
      \"keyName\": \"sshKey-04-05-45\",
      \"instanceType\": \"i4i.metal\"
    },
    {
      \"hostName\": \"esx03\",
      \"keyName\": \"sshKey-04-05-45\",
      \"instanceType\": \"i4i.metal\"
    },
    {
      \"hostName\": \"esx04\",
      \"keyName\": \"sshKey-04-05-45\",
      \"instanceType\": \"i4i.metal\"
    }
  ]" \
--connectivity-info "{
    \"privateRouteServerPeerings\": [\"rsp-1234567890abcdef0\",\"rsp-
abcdef01234567890\"]
  }" \
  --vcf-hostnames "{
    \"vCenter\": \"vcf-vc01\",
    \"nsx\": \"vcf-nsx\",
    \"nsxManager1\": \"vcf-nsxm01\",
    \"nsxManager2\": \"vcf-nsxm02\",
    \"nsxManager3\": \"vcf-nsxm03\",
    \"nsxEdge1\": \"vcf-edge01\",
    \"nsxEdge2\": \"vcf-edge02\",
    \"sddcManager\": \"vcf-sddcm01\",
    \"cloudBuilder\": \"vcf-cb01\"
  }" \
--site-id my-site-id \
--region us-east-2
```

The following is a sample response.

```
{
    "environment": {
        "environmentId": "env-abcde12345",
        "environmentState": "CREATING",
        "stateDetails": "The environment is being initialized, this operation
 may take some time to complete.",
        "createdAt": "2025-04-13T12:03:39.718000+00:00",
        "modifiedAt": "2025-04-13T12:03:39.718000+00:00",
        "environmentArn": "arn:aws:evs:us-east-2:111122223333:environment/env-
abcde12345",
        "environmentName": "testEnv",
        "vpcId": "vpc-1234567890abcdef0",
        "serviceAccessSubnetId": "subnet-01234a1b2cde1234f",
        "vcfVersion": "VCF-5.2.1",
        "termsAccepted": true,
        "licenseInfo": [
            {
                "solutionKey": "00000-00000-00000-abcde-11111",
                "vsanKey": "00000-00000-00000-abcde-22222"
            }
        ],
        "siteId": "my-site-id",
        "connectivityInfo": {
            "privateRouteServerPeerings": [
                "rsp-1234567890abcdef0",
                "rsp-abcdef01234567890"
            ]
        },
        "vcfHostnames": {
            "vCenter": "vcf-vc01",
            "nsx": "vcf-nsx",
            "nsxManager1": "vcf-nsxm01",
            "nsxManager2": "vcf-nsxm02",
            "nsxManager3": "vcf-nsxm03",
            "nsxEdge1": "vcf-edge01",
            "nsxEdge2": "vcf-edge02",
            "sddcManager": "vcf-sddcm01",
            "cloudBuilder": "vcf-cb01"
        }
    }
}
```

Verify Amazon EVS environment creation

Example

Amazon EVS console

1. Go to the Amazon EVS console.
2. In the navigation pane, choose **Environments**.
3. Select the environment.
4. Select the **Details** tab.
5. Check that the **Environment status** is **Passed** and the **Environment state** is **Created**. This lets you know that the environment is ready to use.

Note

Environment creation can take several hours. If the **Environment state** still shows **Creating**, refresh the page.

AWS CLI

1. Open a terminal session.
2. Run the following command, using the environment ID for your environment and the Region name that contains your resources. The environment is ready to use when the `environmentState` is `CREATED`.

   Note

   Environment creation can take several hours. If the `environmentState` still shows `CREATING`, run the command again to refresh the output.

   ```
   aws evs get-environment  --environment-id env-abcde12345
   ```

   The following is a sample response.

   ```
   {
       "environment": {
           "environmentId": "env-abcde12345",
           "environmentState": "CREATED",
           "createdAt": "2025-04-13T13:39:49.546000+00:00",
           "modifiedAt": "2025-04-13T13:40:39.355000+00:00",
           "environmentArn": "arn:aws:evs:us-east-2:111122223333:environment/env-
   abcde12345",
           "environmentName": "testEnv",
           "vpcId": "vpc-0c6def5b7b61c9f41",
           "serviceAccessSubnetId": "subnet-06a3c3b74d36b7d5e",
           "vcfVersion": "VCF-5.2.1",
           "termsAccepted": true,
           "licenseInfo": [
               {
   ```

```
                "solutionKey": "00000-00000-00000-abcde-11111",
                "vsanKey": "00000-00000-00000-abcde-22222"
            }
        ],
        "siteId": "my-site-id",
        "checks": [],
        "connectivityInfo": {
            "privateRouteServerPeerings": [
                "rsp-056b2b1727a51e956",
                "rsp-07f636c5150f171c3"
            ]
        },
        "vcfHostnames": {
            "vCenter": "vcf-vc01",
            "nsx": "vcf-nsx",
            "nsxManager1": "vcf-nsxm01",
            "nsxManager2": "vcf-nsxm02",
            "nsxManager3": "vcf-nsxm03",
            "nsxEdge1": "vcf-edge01",
            "nsxEdge2": "vcf-edge02",
            "sddcManager": "vcf-sddcm01",
            "cloudBuilder": "vcf-cb01"
        },
        "credentials": []
    }
}
```

Explicitly associate Amazon EVS VLAN subnets to a VPC route table

Explicitly associate each of the Amazon EVS VLAN subnets with a route table in your VPC. This route table is used to allow AWS resources to communicate with virtual machines on NSX network segments, running with Amazon EVS.

Example

Amazon VPC console

1. Go to the VPC console.
2. In the navigation pane, choose **Route tables**.

3. Choose the route table that you want to associate with Amazon EVS VLAN subnets.
4. Select the **Subnet associations** tab.
5. Under **Explicit subnet associations**, select **Edit subnet associations**.
6. Select all of the Amazon EVS VLAN subnets.
7. Choose **Save associations**.

AWS CLI

1. Open a terminal session.
2. Identify the Amazon EVS VLAN subnet IDs.

```
aws ec2 describe-subnets
```

3. Associate your Amazon EVS VLAN subnets with a route table in your VPC.

```
aws ec2 associate-route-table \
--route-table-id rtb-0123456789abcdef0 \
--subnet-id subnet-01234a1b2cde1234f
```

(Optional) Configure transit gateway route tables and Direct Connect prefixes for on-premises connectivity

If you are configuring on-premises network connectivity using AWS Direct Connect or AWS Site-to-Site VPN with a transit gateway, you must update the transit gateway route tables with the VPC CIDRs created within the Amazon EVS environment. For more information, see Transit gateway route tables in Amazon VPC Transit Gateways.

If you are using AWS Direct Connect, you may need to also update your Direct Connect prefixes to send and receive updated routes from the VPC. For more information, see Allows prefixes interactions for AWS Direct Connect gateways.

Create a network ACL to control Amazon EVS VLAN subnet traffic

Amazon EVS uses a network access control list (ACL) to control traffic to and from Amazon EVS VLAN subnets. You can use the default network ACL for your VPC, or you can create a custom network ACL for your VPC with rules that are similar to the rules for your security groups to add a layer of security to your VPC. For more information, see Create a network ACL for your VPC in the *Amazon VPC User Guide*.

⚠ **Important**

EC2 security groups do not function on elastic network interfaces that are attached to Amazon EVS VLAN subnets. To control traffic to and from Amazon EVS VLAN subnets, you must use a network access control list.

Retrieve VCF credentials and access VCF management appliances

Amazon EVS uses AWS Secrets Manager to create, encrypt, and store managed secrets in your account. These secrets contain the VCF credentials needed to install and access VCF management appliances such as vCenter Server, NSX, and SDDC Manager. For more information about retrieving secrets, see Get secrets from AWS Secrets Manager.

ⓘ **Note**

Amazon EVS does not provide managed rotation of your secrets. We recommend that you rotate your secrets regularly on a set rotation window to ensure that secrets are not long-lived.

After you have retrieved your VCF credentials from AWS Secrets Manager, you can use them to log into your VCF management appliances. For more information, see Log in to the SDDC Manager User Interface and How to Use and Configure Your vSphere Client in the VMware product documentation.

Configure the EC2 Serial Console

By default, Amazon EVS enables the ESXi Shell on newly deployed Amazon EVS hosts. This configuration allows access to the Amazon EC2 instance’s serial port through the EC2 serial console, which you can use to troubleshoot boot, network configuration, and other issues. The serial console does not require your instance to have any networking capabilities. With the serial console, you can enter commands to a running EC2 instance as if your keyboard and monitor are directly attached to the instance’s serial port.

The EC2 serial console can be accessed using the EC2 console or the AWS CLI. For more information, see EC2 Serial Console for instances in the *Amazon EC2 User Guide*.

> **Note**
>
> The EC2 serial console is the only Amazon EVS supported mechanism to access the Direct Console User Interface (DCUI) to interact with an ESXi host locally.

> **Note**
>
> Amazon EVS disables remote SSH by default. For more information about enabling SSH to access the remote ESXi Shell, see Remote ESXi Shell Access with SSH in the VMware vSphere product documentation.

Connect to the EC2 Serial Console

To connect to the EC2 serial console and use your chosen tool for troubleshooting, certain prerequisite tasks must be completed. For more information, see Prerequisites for the EC2 Serial Console and Connect to the EC2 Serial Console in the *Amazon EC2 User Guide*.

> **Note**
>
> To connect to the EC2 serial console, your EC2 instance state must be `running`. You can’t connect to the serial console if the instance is in the `pending`, `stopping`, `stopped`, `shutting-down`, or `terminated` state. For more information about instance state changes, see Amazon EC2 instance state change in the *Amazon EC2 User Guide*.

Configure access to the EC2 Serial Console

To configure access to the EC2 serial console, you or your administrator must grant serial console access at the account level and then configure IAM policies to grant access to your users. For Linux instances, you must also configure a password-based user on every instance so that your users can use the serial console for troubleshooting. For more information, see Configure access to the EC2 Serial Console in the *Amazon EC2 User Guide*.

Clean up

Follow these steps to delete the AWS resources that were created.

Delete the Amazon EVS hosts and environment

Follow these steps to delete the Amazon EVS hosts and environment. This action deletes the VMware VCF installation that runs in your Amazon EVS environment.

> **Note**
>
> To delete an Amazon EVS environment, you must first delete all hosts within the environment. An environment cannot be deleted if there are hosts associated with the environment.

Example

SDDC UI and Amazon EVS console

1. Go the to SDDC Manager user interface.
2. Remove the hosts from the vSphere cluster. This will unassign the hosts from the SDDC domain. Repeat this step for each host in the cluster. For more information, see Remove a Host from a vSphere Cluster in a Workload Domain in the VCF product documentation.
3. Decommission the unassigned hosts. For more information, see Decommission Hosts in the VCF product documentation.
4. Go to the Amazon EVS console.

> **Note**
>
> Amazon EVS operations triggered from the Amazon EVS console will not generate CloudTrail events.

5. In the navigation pane, choose **Environment**.
6. Select the environment that contains the hosts to delete.
7. Select the **Hosts** tab.
8. Select the host and choose **Delete** within the **Hosts** tab. Repeat this step for each host in the environment.
9. At the top of the **Environments** page, choose **Delete** and then **Delete environment**.

> **Note**
>
> Environment deletion also deletes the Amazon EVS VLAN subnets and AWS Secrets Manager secrets that Amazon EVS created. AWS resources that you create are not deleted. These resources may continue to incur costs.

10If you have Amazon EC2 Capacity Reservations in place that you no longer require, ensure that you've canceled them. For more information, see Cancel a Capacity Reservation in the *Amazon EC2 User Guide*.

SDDC UI and AWS CLI

1. Open a terminal session.
2. Identify the environment that contains the host to delete.

```
aws evs list-environments
```

The following is a sample response.

```
{
    "environmentSummaries": [
        {
            "environmentId": "env-abcde12345",
            "environmentName": "testEnv",
            "vcfVersion": "VCF-5.2.1",
            "environmentState": "CREATED",
            "createdAt": "2025-04-13T14:42:41.430000+00:00",
            "modifiedAt": "2025-04-13T14:43:33.412000+00:00",
            "environmentArn": "arn:aws:evs:us-east-2:111122223333:environment/env-abcde12345"
        },
        {
            "environmentId": "env-edcba54321",
            "environmentName": "testEnv2",
            "vcfVersion": "VCF-5.2.1",
            "environmentState": "CREATED",
            "createdAt": "2025-04-13T13:39:49.546000+00:00",
            "modifiedAt": "2025-04-13T13:52:13.342000+00:00",
            "environmentArn": "arn:aws:evs:us-east-2:111122223333:environment/env-edcba54321"
        }
    ]
}
```

3. Go the to SDDC Manager user interface.
4. Remove the hosts from the vSphere cluster. This will unassign the hosts from the SDDC domain. Repeat this step for each host in the cluster. For more information, see Remove a Host from a vSphere Cluster in a Workload Domain in the VCF product documentation.
5. Decommission the unassigned hosts. For more information, see Decommission Hosts in the VCF product documentation.
6. Delete the hosts from the environment. Below is a sample `aws evs delete-environment-host` request.

   > **Note**
   >
   > To be able to delete an environment, you must first delete all of the hosts that are contained in the environment.

   ```
   aws evs delete-environment-host \
   --environment-id env-abcde12345 \
   --host esx01
   ```

7. Repeat the previous steps to delete the remaining hosts in your environment.

8. Delete the environment.

```
aws evs delete-environment --environment-id env-abcde12345
```

> **Note**
>
> Environment deletion also deletes the Amazon EVS VLAN subnets and AWS Secrets Manager secrets that Amazon EVS created. Other AWS resources that you create are not deleted. These resources may continue to incur costs.

9. If you have Amazon EC2 Capacity Reservations in place that you no longer require, ensure that you've canceled them. For more information, see Cancel a Capacity Reservation in the *Amazon EC2 User Guide*.

Delete the VPC Route Server components

For steps to delete the Amazon VPC Route Server components that you created, see Route Server cleanup in the *Amazon VPC User Guide*.

Delete the network access control list (ACL)

For steps to delete a network access control list, see Delete a network ACL for your VPC in the *Amazon VPC User Guide*.

Delete elastic network interfaces

For steps to delete elastic network interfaces, see Delete a network interface in the *Amazon EC2 User Guide*.

Disassociate and delete subnet route tables

For steps to disassociate and delete subnet route tables, see Subnet route tables in the *Amazon VPC User Guide*.

Delete subnets

Delete the VPC subnets, including the service access subnet. For steps to delete VPC subnets, see Delete a subnet in the *Amazon VPC User Guide*.

Note

If you’re using Route 53 for DNS, remove the inbound endpoints before you attempt to delete the service access subnet. Otherwise, you will not be able to delete the service access subnet.

Note

Amazon EVS deletes the VLAN subnets on your behalf when the environment is deleted. Amazon EVS VLAN subnets can only be deleted when the environment is deleted.

Delete the VPC

For steps to delete the VPC, see Delete your VPC in the *Amazon VPC User Guide*.

Next steps

Migrate your workloads to Amazon EVS using VMware Hybrid Cloud Extension (VMware HCX). For more information, see *Migration*.

Migrate workloads to Amazon EVS using VMware Hybrid Cloud Extension (VMware HCX)

Note

Amazon EVS is in public preview release and is subject to change.

After you have created an Amazon EVS environment, you can migrate your existing VMware-based workloads to Amazon Elastic VMware Service (Amazon EVS) using VMware Hybrid Cloud Extension (VMware HCX). For more information about VMware HCX migration, see VMware HCX Migration Types in the *VMware HCX User Guide*.

The following tutorial describes how to use VMware HCX to migrate a VMware workload to Amazon EVS.

You can use VMware HCX to migrate workloads over a private connection using AWS Direct Connect with an associated transit gateway, or using an AWS Site-to-Site VPN attachment to a transit gateway.

Note

Amazon EVS does not support connectivity via an AWS Direct Connect private virtual interface (VIF), or via an AWS Site-to-Site VPN connection that terminates directly into the underlay VPC.

For more information about setting up an AWS Direct Connect connection, see AWS Direct Connect gateways and transit gateway associations in the *AWS Direct Connect User Guide*. For more information about using AWS Site-to-Site VPN with AWS Transit Gateway, see AWS Site-to-Site VPN attachments in Amazon VPC Transit Gateways in the *Amazon VPC Transit Gateway User Guide*.

Prerequisites

Before using VMware HCX with Amazon EVS, ensure that HCX prerequisites have been met and an Amazon EVS environment has been created and connected to your on-premises network using either AWS Direct Connect with a transit gateway or AWS Site-to-Site VPN with a transit gateway.

For steps to create an Amazon EVS environment, see *Getting started*. For more information about VMware HCX prerequisites, see the section called “VMware HCX prerequisites”.

Check the status of the HCX VLAN subnet

Follow these steps to check that the HCX VLAN subnet is properly configured.

Example

Amazon EVS console

1. Go to the Amazon EVS console.
2. In the navigation pane, choose **Environments**.
3. Select the Amazon EVS environment.
4. Select the **Networks and connectivity** tab.
5. Under **VLANs**, identify the HCX VLAN and check that the **State** is **Created**.
6. Copy the HCX `vlan` ID for later use.

AWS CLI

1. Run the following command, using the environment ID for your environment and the Region name that contains your resources.

   ```
   aws evs list-environment-vlans  --region <region-name> --environment-id env-
   abcde12345
   ```

   The following is a sample response.

   ```
   {
    "environmentVlans": [
           {
               "vlan": 80,
               "cidr": "10.10.7.0/24",
               "availabilityZone": "us-east-2c",
               "functionName": "hcx",
               "createdAt": "2025-04-13T13:39:58.845000+00:00",
               "modifiedAt": "2025-04-13T13:47:57.067000+00:00",
               "vlanState": "CREATED",
   ```

```
            "stateDetails": ""
        },
        {
            "vlan": 20,
            "cidr": "10.10.1.0/24",
            "availabilityZone": "us-east-2c",
            "functionName": "vmManagement",
            "createdAt": "2025-04-13T13:39:58.456000+00:00",
            "modifiedAt": "2025-04-13T13:47:57.524000+00:00",
            "vlanState": "CREATED",
            "stateDetails": ""
        }
 ]
}
```

2. Identify the VLAN with a `functionName` of `hcx` and check that the `vlanState` is `CREATED`.
3. Copy the HCX `vlan` ID for later use.

Check that the HCX VLAN subnet is associated with a network ACL

Follow these steps to check that the HCX VLAN subnet is associated with a network ACL. For more information about network ACL association, see the section called “Create a network ACL to control Amazon EVS VLAN subnet traffic”.

Example

Amazon VPC console

1. Go to the Amazon VPC console.
2. In the navigation pane, choose **Network ACLs**.
3. Select the network ACL that your VLAN subnets are associated with.
4. Select the **Subnet associations** tab.
5. Check that the HCX VLAN subnet is listed among the associated subnets.

AWS CLI

1. Run the following command, using the HCX VLAN subnet ID in the `Values` filter.

```
aws ec2 describe-network-acls --filters "Name=subnet-id,Values=subnet-
abcdefg9876543210"
```

2. Check that the correct network ACL is returned in the response.

Create a distributed port group with the HCX public uplink VLAN ID

Go to the vSphere Client interface and follow the steps in Add a Distributed Port Group to add a distributed port group to a vSphere Distributed Switch.

When configuring failback within the vSphere Client interface, ensure that uplink1 is an active uplink and uplink2 is a standby uplink to enable Active/Standby failover. For the VLAN setting in the vSphere Client interface, enter the HCX VLAN ID that you previously identified.

(Optional) Set up HCX WAN Optimization

The HCX WAN Optimization service (HCX-WAN-OPT) improves the performance characteristics of private lines or internet path by applying WAN optimization techniques like data reduction and WAN path conditioning. The HCX WAN Optimization service is recommended on deployments that are not able to dedicate 10Gbit paths for migrations. In 10Gbit, low latency deployments, using WAN Optimization may not yield improved migration performance. For more information, see VMware HCX Deployment Considerations and Best Practices.

The HCX WAN Optimization service is deployed in conjunction with the HCX WAN Interconnect service appliance (HCX-WAN-IX). HCX-WAN-IX is responsible for data replication between the enterprise environment and the Amazon EVS environment.

To use the HCX WAN Optimization service with Amazon EVS, you need to use a distributed port group on the HCX VLAN subnet. Use the distributed port group that was created in the earlier step.

(Optional) Enable HCX Mobility Optimized Networking

HCX Mobility Optimized Networking (MON) is a feature of the HCX Network Extension Service. MON-enabled network extensions improve traffic flows for migrated virtual machines by enabling selective routing within your Amazon EVS environment. MON allows you to configure the optimal path for migrating workload traffic to Amazon EVS, avoiding a long round-trip network path through the source gateway. This feature is available for all Amazon EVS deployments. For more information, see Configuring Mobility Optimized Networking in the VMware HCX User Guide.

⚠ **Important**

Before your enable HCX MON, read the following limitations and unsupported configurations for HCX Network Extension.
Restrictions and Limitations for Network Extension
Restrictions and Limitations for Mobility Optimized Networking Topologies

⚠ **Important**

Before you enable HCX MON, make sure that in the NSX interface you've configured route redistribution for the destination network CIDR. For more information, see Configure BGP and Route Redistribution in the VMware NSX documentation.

Verify HCX connectivity

VMware HCX includes built-in diagnostic tools that can be used to test connectivity. For more information, see VMware HCX Troubleshooting in the *VMware HCX User Guide*.

Security in Amazon Elastic VMware Service

Note

Amazon EVS is in public preview release and is subject to change.

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon Elastic VMware Service, see AWS services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations

This documentation helps you understand how to apply the shared responsibility model when using Amazon Elastic VMware Service. It shows you how to configure Amazon Elastic VMware Service to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Elastic VMware Service resources.

Contents

Identity and access management for Amazon Elastic VMware Service

Note

Amazon EVS is in public preview release and is subject to change.

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Amazon Elastic VMware Service resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work you do in Amazon Elastic VMware Service.

Service user – If you use the Amazon Elastic VMware Service service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Amazon Elastic VMware Service features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator.

Service administrator - If you're in charge of Amazon Elastic VMware Service resources at your company, you probably have full access to Amazon Elastic VMware Service. It's your job to determine which Amazon Elastic VMware Service features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Amazon Elastic VMware Service, see the section called “How Amazon Elastic VMware Service works with IAM”.

IAM administrator - If you’re an IAM administrator, you might want to learn details about how you can write policies to manage access to Amazon Elastic VMware Service. To view example Amazon Elastic VMware Service identity-based policies that you can use in IAM, see Amazon Elastic VMware Service identity-based policy examples.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company’s single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests using your credentials. If you don’t use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signature Version 4 signing process in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *Account Management Reference Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A federated identity is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An IAM group is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access –** To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center (successor to AWS Single Sign-On) User Guide*.
- **Temporary IAM user permissions –** An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access –** You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.
- **Cross-service access –** Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Principal permissions –** When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services,

you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions.

  - **Service role –** A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role –** A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an Amazon EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the Amazon EC2 instance. To assign an AWS role to an Amazon EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the Amazon EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. By default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket. Service administrators can use these policies to define what actions a specified principal (account member, user, or role) can perform on that resource and under what conditions. Resource-based policies are inline policies. There are no managed resource-based policies.

Access control lists (ACLs)

Access control lists (ACLs) are a type of policy that controls which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format. Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access Control List (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries –** A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs) –** SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies –** Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Amazon Elastic VMware Service works with IAM

> **Note**
>
> Amazon EVS is in public preview release and is subject to change.

Before you use IAM to manage access to Amazon Elastic VMware Service, learn what IAM features are available to use with Amazon Elastic VMware Service.

IAM feature	Amazon EVS support
the section called “Identity-based policies for Amazon EVS”	Yes
the section called “Resource-based policies within Amazon EVS”	No
the section called “Policy actions for Amazon EVS”	Yes
the section called “Policy resources for Amazon EVS”	Partial
the section called “Policy condition keys for Amazon EVS”	Yes
the section called “Access control lists (ACLs) in Amazon EVS”	No
the section called “Attribute-based access control (ABAC) with Amazon EVS”	Yes
the section called “Using temporary credentia ls with Amazon EVS”	Yes
the section called “Forward access sessions for Amazon EVS”	Yes
the section called “Service roles for Amazon EVS”	No
the section called “Service-linked roles for Amazon EVS”	Yes

To get a high-level view of how Amazon Elastic VMware Service and other AWS services work with IAM, see AWS services that work with IAM in the *IAM User Guide*.

Topics

Identity-based policies for Amazon EVS

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Amazon EVS

To view examples of Amazon Elastic VMware Service identity-based policies, see Amazon Elastic VMware Service identity-based policy examples.

Resource-based policies within Amazon EVS

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the IAM User Guide.

Policy actions for Amazon EVS

Supports actions Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used in a policy to grant permissions to perform the associated operation.

Policy actions in Amazon Elastic VMware Service use the following prefix before the action: `evs:`. For example, to grant someone permission to create an environment with the Amazon EVS `CreateEnvironment` API operation, you include the `evs:CreateEnvironment` action in their policy. Policy statements must include either an `Action` or `NotAction` element. Amazon Elastic VMware Service defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
      "evs:action1",
      "evs:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "evs:List*"
```

To see a list of Amazon Elastic VMware Service actions, see Actions Defined by Amazon Elastic VMware Service in the *Service Authorization Reference*.

Policy resources for Amazon EVS

Supports policy resources: Partial

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Amazon EVS resource types and their ARNs, see Resources defined by Amazon Elastic VMware Service in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Amazon Elastic VMware Service.

Some Amazon EVS API actions support multiple resources. For example, multiple environments can be referenced when calling the `ListEnvironments` API action. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

For example, the Amazon EVS environment resource has the following ARN:

```
arn:${Partition}:evs:${Region}:${Account}:environment/${EnvironmentId}
```

To specify the environments `my-environment-1` and `my-environment-2` in your statement, use the following example ARNs:

```
"Resource": [
         "arn:aws:evs:us-east-1:123456789012:environment/my-environment-1",
         "arn:aws:evs:us-east-1:123456789012:environment/my-environment-2"
```

To specify all environments that belong to a specific account, use the wildcard (*):

```
"Resource": "arn:aws:evs:us-east-1:123456789012:environment/*"
```

Policy condition keys for Amazon EVS

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` block) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

Amazon Elastic VMware Service defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

All Amazon EC2 actions support the `aws:RequestedRegion` and `ec2:Region` condition keys. For more information, see Example: Restricting access to a specific region.

To see a list of Amazon Elastic VMware Service condition keys, see Condition Keys for Amazon Elastic VMware Service in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by Amazon Elastic VMware Service.

Access control lists (ACLs) in Amazon EVS

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Attribute-based access control (ABAC) with Amazon EVS

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called tags. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

You can attach tags to Amazon Elastic VMware Service resources or pass tags in a request to Amazon Elastic VMware Service. To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/<key-name>`, `aws:RequestTag/<key-name>`, or `aws:TagKeys` condition keys. For more information about which actions that you can use tags in condition keys with, see Actions defined by Amazon EVS in the *Service Authorization Reference*.

Using temporary credentials with Amazon EVS

Supports temporary credentials: Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company’s single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switch from a user to an IAM role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Forward access sessions for Amazon EVS

Supports forward access sessions (FAS): Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Amazon EVS

Supports service roles: No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

Service-linked roles for Amazon EVS

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing Amazon Elastic VMware Service service-linked roles, see the section called “Using service-linked roles”.

Amazon EVS identity-based policy examples

Note

Amazon EVS is in public preview release and is subject to change.

By default, IAM users and roles don't have permission to create or modify Amazon Elastic VMware Service resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies using the JSON editor in the *IAM User Guide*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Amazon Elastic VMware Service resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the AWS managed policies that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on

specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or root users in your account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

Using the Amazon Elastic VMware Service console

To access the Amazon Elastic VMware Service console, an IAM principal must have a minimum set of permissions. These permissions must allow the principal to list and view details about the Amazon Elastic VMware Service resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for principals with that policy attached to them.

To ensure that your IAM principals can still use the Amazon Elastic VMware Service console, create a policy with your own unique name, such as `AmazonEVSAdminPolicy`. Attach the policy to the principals. For more information, see Adding permissions to a user in the *IAM User Guide*:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evs:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EVSServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/evs.amazonaws.com/
AWSServiceRoleForEVS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "evs.amazonaws.com"
                }
            }
        }
    ]
}
```

You don’t need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you’re trying to perform.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Create and manage an Amazon EVS environment

This example policy includes the permissions required to create and delete an Amazon EVS environment, and add or delete hosts after the environment has been created.

You can replace the AWS Region with the AWS Region that you want to create an environment in. If your account already has the `AWSServiceRoleForAmazonEVS` role, you can remove the `iam:CreateServiceLinkedRole` action from the policy. If you've ever created an Amazon EVS environment in your account, a role with these permissions already exists, unless you deleted it.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadOnlyDescribeActions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeHosts",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeAddresses",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeSubnets",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeInstances",
                "ec2:DescribeRouteServers",
                "ec2:DescribeRouteServerEndpoints",
                "ec2:DescribeRouteServerPeers",
                "ec2:DescribePlacementGroups",
                "ec2:DescribeVolumes",
                "ec2:DescribeSecurityGroups",
                "support:DescribeServices",
                "support:DescribeSupportLevel",
                "servicequotas:GetServiceQuota",
                "servicequotas:ListServiceQuotas"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ModifyNetworkInterfaceStatement",
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "ModifyNetworkInterfaceStatementForSubnetAssociation",
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyNetworkInterfaceAttribute"
            ],
            "Resource": "arn:aws:ec2:*:*:subnet/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "CreateNetworkInterfaceWithTag",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "CreateNetworkInterfaceAdditionalResources",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "TagOnCreateEC2Resources",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:subnet/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "CreateNetworkInterface",
                        "RunInstances",
                        "CreateSubnet",
                        "CreateVolume"
                    ]
                },
                "Null": {
                    "aws:RequestTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "DetachNetworkInterface",
            "Effect": "Allow",
            "Action": [
                "ec2:DetachNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "RunInstancesWithTag",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "RunInstancesWithTagResource",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "RunInstancesWithoutTag",
            "Effect": "Allow",
            "Action": [
                "ec2:RunInstances"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:image/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:placement-group/*"
            ]
        },
        {
            "Sid": "TerminateInstancesWithTag",
            "Effect": "Allow",
            "Action": [
                "ec2:TerminateInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "CreateSubnetWithTag",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSubnet"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "CreateSubnetWithoutTagForExistingVPC",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSubnet"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*"
            ]
        },
        {
            "Sid": "DeleteSubnetWithTag",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSubnet"
            ],
            "Resource": "arn:aws:ec2:*:*:subnet/*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "VolumeDeletion",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteVolume"
            ],
            "Resource": "arn:aws:ec2:*:*:volume/*",
             "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "VolumeDetachment",
            "Effect": "Allow",
            "Action": [
                "ec2:DetachVolume"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "RouteServerAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:GetRouteServerAssociations"
            ],
            "Resource": "arn:aws:ec2:*:*:route-server/*"

        },
        {
            "Sid": "EVSServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/evs.amazonaws.com/
AWSServiceRoleForEVS",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "evs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "SecretsManagerCreateWithTag",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonEVSManaged": "true"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "AmazonEVSManaged"
                    ]
                }
            }
        },
        {
            "Sid": "SecretsManagerTagging",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:TagResource"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AmazonEVSManaged": "true",
                    "aws:ResourceTag/AmazonEVSManaged": "true"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "AmazonEVSManaged"
                    ]
                }
            }
        },
        {
            "Sid": "SecretsManagerOps",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:UpdateSecret"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonEVSManaged": "false"
                }
            }
        },
        {
            "Sid": "SecretsManagerRandomPassword",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EVSPermissions",
            "Effect": "Allow",
            "Action": [
                "evs:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "KMSKeyAccessInConsole",
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },

        {
            "Sid": "KMSKeyAliasAccess",
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        }
    ]
}
```

Get and list Amazon EVS environments, hosts, and VLANs

This example policy includes the minimum permissions required for an administrator to get and list all Amazon EVS environments, hosts, and VLANs within a given account in the us-east-2 AWS Region.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "evs:Get*",
        "evs:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Troubleshooting Amazon Elastic VMware Service identity and access

Note

Amazon EVS is in public preview release and is subject to change.

Use the following information to help you diagnose and fix common issues that you might encounter when working with Amazon Elastic VMware Service and IAM.

Topics

- AccessDeniedException
- I want to allow people outside of my AWS account to access my Amazon Elastic VMware Service resources

AccessDeniedException

If you receive an `AccessDeniedException` when calling an AWS API operation, then the IAM principal credentials that you’re using don’t have the required permissions to make that call.

```
An error occurred (AccessDeniedException) when calling the CreateEnvironment operation:
User: arn:aws:iam::111122223333:user/user_name is not authorized to perform:
evs:CreateEnvironment on resource: arn:aws:evs:region:111122223333:environment/my-env
```

In the previous example message, the user does not have permissions to call the Amazon EVS `CreateEnvironment` API operation. To provide Amazon EVS admin permissions to an IAM principal, see the section called "Amazon EVS identity-based policy examples".

For more general information about IAM, see Control access to AWS resources using policies in the *IAM User Guide*.

I want to allow people outside of my AWS account to access my Amazon Elastic VMware Service resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Amazon Elastic VMware Service supports these features, see the section called "How Amazon Elastic VMware Service works with IAM".
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing Access to Externally Authenticated Users (Identity Federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the *IAM User Guide*.

AWS managed policies for Amazon EVS

Note

Amazon EVS is in public preview release and is subject to change.

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services. For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AmazonEVSServiceRolePolicy

You can't attach `AmazonEVSServiceRolePolicy` to your IAM entities. This policy is attached to a service-linked role that allows Amazon EVS to perform actions on your behalf. For more information, see the section called "Using service-linked roles". When you create an environment using an IAM principal that has the `iam:CreateServiceLinkedRole` permission, the `AWSServiceRoleforAmazonEVS` service-linked role is automatically created for you with this policy attached to it.

This policy allows the service-linked role to call AWS services on your behalf.

Permissions details

This policy includes the following permissions that allow Amazon EVS to complete the following tasks.

- `ec2` - Create, modify, tag, and delete an elastic network interface that is used to establish a persistent connection between Amazon EVS and a VMware Virtual Cloud Foundation (VCF) SDDC Manager appliance in the customer's VPC subnet. This connectivity is required for Amazon EVS to be able to deploy, manage, and monitor the VCF deployment.

To view the latest version of the JSON policy document, see AmazonEVSServiceRolePolicy in the *AWS Managed Policy Reference Guide*.

Amazon EVS updates to AWS managed policies

View details about updates to AWS managed policies for Amazon EVS since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the *Document history* page.

Change	Description	Date
AmazonEVSServiceRolePolicy — New policy added	Amazon EVS added a new policy that allow the service to connect to a VPC subnet in the customer account. This connection is required for service functionality. To learn more, see the section called "AWS managed policy: AmazonEVSServiceRolePolicy".	June 09, 2025
Amazon EVS started tracking changes	Amazon EVS started tracking changes for its AWS managed policies.	June 09, 2025

Using service-linked roles for Amazon EVS

Note

Amazon EVS is in public preview release and is subject to change.

Amazon Elastic VMware Service uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon EVS. Service-linked roles are predefined by Amazon EVS and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon EVS easier because you don't have to manually add the necessary permissions. Amazon EVS defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon EVS can assume its roles. The defined permissions include the trust policy and the permissions policy. The permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your Amazon EVS resources because you can’t inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that have **Yes** in the **Service-linked role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Amazon EVS

Amazon EVS uses the service-linked role named `AWSServiceRoleForAmazonEVS`. The role allows Amazon EVS to manage environments in your account. The attached policy allows the role to manage the following resources: EVS elastic network interfaces, EVS VLAN subnets, and VPCs.

The `AWSServiceRoleForAmazonEVS` service-linked role trusts the following services to assume the role:

- `evs.amazonaws.com`

The role permissions policy allows Amazon EVS to complete the following actions on the specified resources:

- AmazonEVSServiceRolePolicy

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Creating a service-linked role for Amazon EVS

You don’t need to manually create a service-linked role. When you create an environment in the AWS Management Console, the AWS CLI, or the AWS API, Amazon EVS creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create an environment, Amazon EVS creates the service-linked role for you again.

Editing a service-linked role for Amazon EVS

Amazon EVS does not allow you to edit the `AWSServiceRoleForAmazonEVS` service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for Amazon EVS

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up your service-linked role before you can manually delete it.

Cleaning up a service-linked role

Before you can use IAM to delete a service-linked role, you must first delete any resources used by the role. For steps to delete an Amazon EVS environment with hosts, see the section called "Delete the Amazon EVS hosts and environment".

> ⓘ **Note**
>
> If the Amazon EVS service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

Manually delete the service-linked role

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForAmazonEVS` service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Supported regions for Amazon EVS service-linked roles

Amazon EVS supports using service-linked roles in all of the regions where the service is available. For more information, see *Endpoints and quotas*.

Using Amazon EVS with other AWS services

Note

Amazon EVS is in public preview release and is subject to change.

Amazon EVS is integrated with other AWS services to provide additional solutions. This topic identifies some of the services that Amazon EVS works with to add functionality.

Topics

Create Amazon EVS resources with AWS CloudFormation

Note

Amazon EVS is in public preview release and is subject to change.

Amazon EVS is integrated with AWS CloudFormation, a service that helps you model and set up your AWS resources so that you can spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want, an Amazon EVS environment for example, and AWS CloudFormation takes care of provisioning and configuring those resources for you.

When you use AWS CloudFormation, you can reuse your template to set up your Amazon EVS resources consistently and repeatedly. Just describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

Amazon EVS and AWS CloudFormation templates

To provision and configure resources for Amazon EVS and related services, you must understand AWS CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your AWS CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with AWS CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

Amazon EVS supports creating environments in AWS CloudFormation. For more information, including examples of JSON and YAML templates for your environments, see Amazon EVS resource type reference in the *AWS CloudFormation User Guide*.

Learn more about AWS CloudFormation

To learn more about AWS CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation Command Line Interface User Guide

Run high-performance workloads with Amazon FSx for NetApp ONTAP

> **Note**
>
> Amazon EVS is in public preview release and is subject to change.

Amazon FSx for NetApp ONTAP is a storage service that allows you to launch and run fully managed ONTAP file systems in the cloud. ONTAP is NetApp's file system technology that provides a widely adopted set of data access and data management capabilities. FSx for ONTAP provides the features, performance, and APIs of on-premises NetApp file systems with the agility, scalability, and simplicity of a fully managed AWS service. For more information, see the FSx for ONTAP User Guide.

Amazon EVS supports the use of Amazon FSx for NetApp ONTAP as an NFS/iSCSI datastore and as guest-connected storage for VMware virtual machines running on Amazon EVS.

Configure FSx for NetApp ONTAP as an NFS datastore

> **Note**
>
> Amazon EVS is in public preview release and is subject to change.

The following procedure details the minimum steps required to configure FSx for NetApp ONTAP as an NFS datastore for Amazon EVS using the FSx console and the VMware vSphere client interface that runs on Amazon EVS.

Prerequisites

Before you use Amazon EVS with Amazon FSx for NetApp ONTAP, make sure that the following prerequisite tasks have been completed.

- An Amazon EVS environment is deployed in your Virtual Private Cloud (VPC). For more information, see *Getting started*.
- You have access to your vSphere client running on Amazon EVS.
- You or your storage admin must have necessary permissions to create and manage FSx for ONTAP file systems in your VPC. For more information, see Identity and access management for Amazon FSx for NetApp ONTAP.

Your IAM principal has appropriate permissions to create and manage FSx for ONTAP file systems in your VPC. For more information, see the section called “Create and manage an Amazon EVS environment”.

Create an FSx for NetApp ONTAP file system

1. Go to the Amazon FSx console.
2. Choose **Create file system**.
3. Select **Amazon FSx for NetApp ONTAP**.
4. Choose **Next**.
5. Select **Standard create**.
6. For **Deployment type**, select a Single-AZ deployment option.

Note

Amazon EVS only supports Single-AZ deployments at this time.

7. For **SSD storage capacity**, specify 1024 GiB.
8. For **Throughput capacity**, choose **Specify throughput capacity**. Choose at least 512 MB/s for Single-AZ 1 or at least 768 MB/s for Single-AZ 2.
9. Select the Amazon EVS VPC that has connectivity to your Amazon EVS VLAN subnets.
10. Select a security group that permits all required FSx for ONTAP NFS traffic to the Amazon EVS host VMkernel management VLAN subnet.
11. Select the Amazon EVS service access subnet that your file system will be deployed in. For more information, see the section called “Service access subnet”.
12. For **Junction path**, specify a meaningful name such as `/vol1` to identify this volume in vSphere.
13. Within **Default volume configuration**, set **Storage efficiency** to **Enabled**.
14. Leave the remaining setting at their default values and choose **Next**.
15. Review the file system attributes and choose **Create file system**.

Retrieve the NFS DNS name for the storage virtual machine

1. Go to the Amazon FSx console.
2. On the left menu, select **File systems**.
3. Choose the newly created file system.
4. Select the **Storage virtual machines** tab.
5. Choose the storage virtual machine.
6. Select the **Endpoints** tab.
7. Copy the network file system (NFS) DNS name for later use in VMware Vsphere.

Create an NFS datastore in vSphere using the FSx for ONTAP volume

Follow the instructions in Create an NFS Datastore in vSphere Environmentto configure Amazon FSx for NetApp ONTAP as external storage for VMware vSphere. For the Server setting in the vSphere client interface, use the storage virtual machine (SVM) NFS DNS name that you copied in the previous step.

Configure FSx for NetApp ONTAP FSx as an iSCSI datastore

> **Note**
>
> Amazon EVS is in public preview release and is subject to change.

The following procedure details the minimum steps required to configure FSx for NetApp ONTAP as an iSCSI datastore for Amazon EVS using the FSx console and VMware vSphere client interface that runs on Amazon EVS.

Prerequisites

Before you use Amazon EVS with Amazon FSx for NetApp ONTAP, make sure that the following prerequisite tasks have been completed.

- An Amazon EVS environment is deployed in your Virtual Private Cloud (VPC). For more information, see *Getting started*.
- You have access to your vSphere client running on Amazon EVS.
- You or your storage admin must have necessary permissions to create and manage FSx for ONTAP file systems in your VPC. For more information, see Identity and access management for Amazon FSx for NetApp ONTAP.

Create an FSx for NetApp ONTAP file system

1. Go to the Amazon FSx console.
2. Choose **Create file system**.
3. Select **Amazon FSx for NetApp ONTAP**.
4. Choose **Next**.
5. Select **Standard create**.
6. For **Deployment type**, select a Single-AZ deployment option.

   > **Note**
   >
   > Amazon EVS only supports Single-AZ deployments at this time.

7. For **SSD storage capacity**, specify 1024 GiB.
8. For **Throughput capacity**, choose **Specify throughput capacity**. Choose at least 512 MB/s for Single-AZ 1 or at least 768 MB/s for Single-AZ 2.
9. Select the Amazon EVS VPC that has connectivity to your Amazon EVS VLAN subnets.
10. Select a security group that permits all required FSx for ONTAP iSCSI traffic to the Amazon EVS host VMkernel management VLAN subnet.
11. Select the Amazon EVS service access subnet that your file system will be deployed in. For more information, see the section called “Service access subnet”.
12. Within **Default volume configuration**, set **Storage efficiency** to **Enabled**.
13. Leave the remaining setting at their default values and choose **Next**.
14. Review the file system attributes and choose **Create file system**.

Configure a software iSCSI adapter in vSphere for ESXi host storage

For each ESXi host, you must configure the software iSCSI adapter so that your ESXi hosts can use it to access iSCSI storage. For instruction to configure the software iSCSI adapter for ESXi hosts in vSphere, see Add or Remove the Software iSCSI Adapter in the VMware vSphere product documentation.

After you configure the software iSCSI adapter, copy the iSCSI Qualified Name (IQN) associated with an iSCSI adapter. These values will be used later.

Create an iSCSI LUN

FSx for ONTAP allows you to create Logical Unit Numbers (LUNs) that are specifically intended for iSCSI access, providing shared block storage to your ESXi hosts. You use the NetApp ONTAP CLI to create a LUN.

Below is a sample command.

> **Note**
>
> It is recommended to configure the LUN size to 90% of the volume size.

```
lun create -vserver <your_svm_name> \
-path /vol/<your_volume_name>/<lun_name> \
-size <required_datastore_capacity> \
-ostype vmware
```

For more information, see Creating an iSCSI LUN in the *FSx for ONTAP User Guide*.

Configure and map an initiator group to the iSCSI LUN

Now that you have created an iSCSI LUN, the next step in the process is to create an initiator group (`igroup`) to connect the volume to the cluster and map the LUN to the initiator group. You use the NetApp ONTAP CLI to perform these actions.

1. Configure the initiator group.

   Below is a sample command. For `--initiator`, use the iSCSI adapter IQNs that you copied in the previous step.

   ```
   igroup create <svm_name> \
   -igroup <initiator_group_name> \
   -protocol iscsi \
   -ostype vmware \
   -initiator <esxi_iqn_1>,<esxi_iqn_2>,<esxi_iqn_3>,<esxi_iqn_4>
   ```

2. Confirm that the `igroup` exists.

   ```
   lun igroup show
   ```

3. Map the LUN to the initiator group. Below is a sample command.

   ```
   lun mapping create -vserver <svm_name> \
   -path /vol/<vol_name>/<lun_name> \
   -igroup <initiator_group_name> \
   -lun-id <scsi_lun_number_for this_datastore>
   ```

4. Use the `lun show -path` command to confirm that the LUN is created, online, and mapped.

   ```
   lun show -path /vol/<vol_name>/<lun_name> -fields state,mapped,serial-hex
   ```

For more information, see Provisioning iSCSI for Linux or Provisioning iSCSI for Windows in the *FSx for ONTAP User Guide*.

Configure dynamic discovery of the iSCSI LUN in vSphere

To allow the ESXi hosts to see the iSCSI LUN, you must configure dynamic discovery for each host in the vSphere client interface. For the iSCSI server field, enter the (NFS) DNS name that you copied in the previous step. For more information, see Configure Dynamic or Static Discovery for iSCSI and iSER on ESXi Host in the VMware vSphere product documentation.

Create a VMFS Datastore in VMware vSphere using the iSCSI LUN

Virtual Machine File System (VMFS) datastores serve as repositories for VMware virtual machines. Follow the instruction in Create a vSphere VMFS Datastore to set up the VMFS datastore in VMware vSphere using the iSCSI LUN that you previously configured.

Troubleshooting

Note

Amazon EVS is in public preview release and is subject to change.

This chapter details some common issues encountered while creating or managing Amazon EVS environments.

Troubleshoot failed environment status checks

Amazon EVS performs automated checks on your environment to identify issues. You can view the status of your environment to identify specific and detectable problems.

Review environment status check information

To investigate impaired environments using the Amazon EVS console

1. Open the Amazon EVS console.
2. In the navigation pane, choose **Environments**, and then select your environment.
3. Select the **Details** tab to see an overview of the environment.
4. Check the **Environment status**. Hover on this field to expand a popover with individual results for each environment status check.

Reachability check failed

The reachability check verifies that Amazon EVS has a persistent connection to SDDC Manager. If Amazon EVS cannot reach the environment, this check fails.

If this check fails, Amazon EVS can no longer reach SDDC Manager to validate the environment status, and hosts can no longer be added to the environment. Reachability failure will also cause the license key re-use and key coverage checks to fail, and the host count check to return an **Unknown** response.

Reachabilty failures indicates that there may be an issue with SDDC Manager, firewall configuration, or a missing certificate. You can attempt to resolve these issues, or reach out to AWS Support for further assistance.

Host count check failed

This check verifies that your environment has a minimum of four hosts, which is a requirement for VCF 5.2.1.

If this check fails, you will need to add hosts so that your environment meets this minimum requirement. Amazon EVS only supports environments with 4 to 16 hosts.

Key re-use check failed

This check verifies that the VCF license key is not in use by another Amazon EVS environment. VCF licenses can be used for only one Amazon EVS environment. This check fails if a used license is added to the environment.

If this check fails, you receive an error response that the Amazon EVS environment could not be created. To address the issue, review your license settings in SDDC Manager and replace any previously used licenses with unused licenses.

> ⚠ **Important**
>
> Use the SDDC Manager user interface to manage VCF component license keys. Amazon EVS requires that you maintain valid component license keys in SDDC Manager for the service to function properly. If you manage component license keys using the vSphere Client, you must ensure that those keys also appear in the licensing screen of the SDDC Manager user interface to prevent license key check failure.

Key coverage check failed

This check verifies that your VCF license key assigned to vCenter Server allocates sufficient vCPU cores and vSAN storage capacity (TiB) for all deployed hosts.

If this check fails, you receive an error response that the Amazon EVS environment could not be created, or an Amazon EVS host could not be added to the environment. Key coverage failure may indicate one of the following issues:

- You've exceeded the supported host count for Amazon EVS. Amazon EVS supports 4 to 16 hosts per environment. If this is the issue, remove or add hosts until your environment is in the supported host range.
- VCF licenses are not properly assigned to vCenter Server. You must assign a license to vCenter Server before its evaluation period expires or the currently assigned license expires. If this is the issue, review license assignments in SDDC Manager.
- Current VCF licenses don't cover vCPU core and vSAN storage capacity needs. The VCF solution key must have at least 256 cores. The vSAN license key must have at least 110 TiB of vSAN capacity. If this is the issue, add vSAN licenses in SDDC Manager until your usage needs are met.

If the above actions don't resolve the issue, reach out to AWS Support for further assistance.

> ⚠ **Important**
>
> Use the SDDC Manager user interface to manage VCF component license keys. Amazon EVS requires that you maintain valid component license keys in SDDC Manager for the service to function properly. If you manage component license keys using the vSphere Client, you must ensure that those keys also appear in the licensing screen of the SDDC Manager user interface to prevent license key check failure.

vSphere HA agent on this host could not reach isolation address

In the vCenter user interface, with the ESXi host selected, you see the message "vSphere HA agent on this host could not reach isolation address <IPv6 address>".

This error message indicates that the vSphere HA agent on a host is unable to reach the default IPv6 isolation address that vSphere HA uses for heartbeat checks. The error message is not indicative of a problem, and only occurs because Amazon EVS does not support IPv6 at this time. The absence of IPV6 support for Amazon EVS does not affect the core functionality of vSphere HA.

To remove the vSphere HA error message, you must disable vSphere HA. For steps to disable vSphere HA in the vSphere client, see the Broadcom article Disabling and enabling VMware High Availability (HA).

VSAN upgrade prechecks fail for ESXi host cluster

When attempting to upgrade the ESXi host cluster using SDDC Manager, vSAN disk-related prechecks may fail. This is because Amazon EVS uses vSAN Express Storage Architecture (ESA), and the upgrade prechecks do not apply to vSAN ESA. For more information, see the Broadcom knowledge base article on this topic.

Amazon Elastic VMware Service endpoints and quotas

Note

Amazon EVS is in public preview release and is subject to change.

The following are the service endpoints and service quotas for this service. To connect programmatically to an AWS service, you use an endpoint. In addition to the standard AWS endpoints, some AWS services offer FIPS endpoints in selected Regions. For more information, see AWS service endpoints. Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account. For more information, see AWS service quotas.

Service endpoints

The Amazon EVS API provides regional and dual-stack endpoints, as well as FIPS endpoints for US Regions. To use the dual-stack endpoints with the AWS CLI, see the Dual-stack and FIPS endpoints configuration in the *AWS SDKs and Tools Reference Guide*.

Region Name	Region	Endpoint	Protocol
US East (North Virginia)	us-east-1	evs.us-east-1.amazonaws.com evs-fips.us-east-1.amazonaws.com evs.us-east-1.api.aws evs-fips.us-east-1.api.aws	HTTPS
US East (Ohio)	us-east-2	evs.us-east-2.amazonaws.com evs-fips.us-east-2.amazonaws.com evs.us-east-2.api.aws evs-fips.us-east-2.api.aws	HTTPS

Region Name	Region	Endpoint	Protocol
US West (Oregon)	us-west-2	evs.us-west-2.amazonaws.com evs-fips.us-west-2.amazonaws.com evs.us-west-2.api.aws evs-fips.us-west-2.api.aws	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	evs.ap-northeast-1.amazonaws.com evs.ap-northeast-1.api.aws	HTTPS
Europe (Frankfurt)	eu-central-1	evs.eu-central-1.amazonaws.com evs.eu-central-1.api.aws	HTTPS
Europe (Ireland)	eu-west-1	evs.eu-west-1.amazonaws.com evs.eu-west-1.api.aws	HTTPS

Service quotas

Amazon EVS has integrated with Service Quotas, an AWS service that you can use to view and manage your quotas from a central location. For more information, see What Is Service Quotas? in the *Service Quotas User Guide*.

With Service Quotas integration, you can use the AWS Management Console or AWS CLI to look up the value of your Amazon EVS quotas and request a quota increase for adjustable quotas. For more information, see Requesting a quota increase in the *Service Quotas User Guide* and request-service-quota-increase in the *AWS CLI Command Reference*.

⚠ **Important**

To enable Amazon EVS environment creation, your host count per EVS environment quota must be at least 4. The default quota is 0. To increase this quota, go to the Service Quotas console and request a quota increase.

⚠ **Important**

Ensure that your EC2 Running On-Demand Standard Instance quota reflects the number of vCPUs that you need for all of the EC2 instances that you will use on Amazon EVS. Each i4i.metal instance uses 128 vCPUs. For information about increasing EC2 service quotas, see Request an increase in the *Amazon EC2 User Guide*.

ⓘ **Note**

If you plan to use EC2 Dedicated Hosts for your Amazon EVS environment, ensure that your EC2 Dedicated i4i Hosts quota reflects the number of Dedicated Hosts that you intend to use for a desired Region. For information about increasing EC2 service quotas, see Request an increase in the *Amazon EC2 User Guide*.

Name	Default	Adjustable	Description
Host count per EVS environment	0	Yes	Maximum number of hosts that can be provisioned within a single Amazon EVS environment.
Environment count per AWS account	1	Yes	The maximum number of EVS environments that can be created in this account in the current Region.

Document history for the Amazon Elastic VMware Service User Guide

Note

Amazon EVS is in public preview release and is subject to change.

The following table describes the documentation releases for Amazon Elastic VMware Service.

Change	Description	Date
Released the environment count per AWS account quota	Amazon EVS released environment count per AWS account quota. The environment count per AWS account quota represent s the maximum number of Amazon EVS environments that can be created in a given account and Region.	July 8, 2025
Amazon EVS released in the Europe (Ireland) Region	Amazon EVS was released in the Europe (Ireland) Region.	June 18, 2025
Released AmazonEVS ServiceRolePolicy	The AWS managed policy AmazonEVSServiceRolePolicy was released.	June 9, 2025
Initial User Guide release	The Amazon Elastic VMware Service User Guide was released. The Amazon EVS User Guide describes all Amazon EVS concepts and provides	June 9, 2025

instructions on using the various features with both the console and the command line interface.