

# Running Hybrid Container workloads with Amazon EKS Anywhere

First published March 22, 2023



# Notices

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

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved.



# Contents

Abstract and introduction.....	1
Abstract.....	1
Introduction.....	2
Use cases for Amazon EKS Anywhere .....	4
Data sovereignty .....	4
Using existing investments in hardware and data center space .....	4
Low latency requirements .....	5
Disconnected hybrid and edge locations.....	5
Local data processing and filtering .....	6
Amazon EKS deployment options .....	6
Amazon EKS on AWS Outposts.....	7
Local clusters for Amazon EKS on AWS Outposts .....	8
Amazon EKS on AWS Local Zones .....	8
Amazon EKS on AWS Wavelength .....	8
Amazon EKS Anywhere deployment options.....	9
Amazon EKS Anywhere architecture.....	11
Amazon EKS Anywhere control plane.....	11
Amazon EKS Anywhere data plane .....	12
Well architected best practices .....	22
Security.....	22
Cost optimization.....	27
Performance efficiency.....	27
Reliability .....	29
Day 2 operations and management .....	33
Amazon EKS Anywhere partners and integrations .....	37
Core .....	37

Platform .....	38
Independent software vendors (ISV) .....	38
Services .....	38
Amazon EKS Anywhere AWS support .....	39
Conclusion .....	39
Contributors .....	39
Further reading .....	40
Document revisions .....	40

# Abstract and introduction

## Abstract

Many organizations adopt cloud-native technologies to build and operate applications with increased agility, uncompromising security, and inherent resiliency. Cloud-native software approaches such as containers and Kubernetes have spearheaded a wave of change that help organizations run highly available, scalable, and self-healing workloads with reduced operational overhead and costs.

Most Amazon Web Services (AWS) customers choose to operate in the cloud. A subset of customers operates applications that can't be moved to the cloud due to low-latency, local data processing, high data transfer costs, or data residency requirements. This leads many organizations to operate in a hybrid environment.

A hybrid environment combines AWS Cloud services with infrastructure in a private customer-managed environment, such as a data centers and co-location facilities. Hybrid deployments enable organizations to extend cloud services into on-premises environments. [AWS hybrid cloud services](#) are designed to deliver a consistent AWS experience in the cloud, on-premises, and at the [edge](#).

[Amazon EKS Anywhere](#) is an AWS hybrid cloud service that allows customers to create and operate Kubernetes clusters on customer-managed infrastructure.

This whitepaper provides cloud engineers and architects best practices for operating Amazon EKS Anywhere on customer-managed infrastructure. The guidance provided in this paper is a subset of best practices provided in the [EKS Best Practices Guides](#), and focuses on building well-architected Amazon EKS Anywhere clusters.

Amazon Elastic Kubernetes Service ([Amazon EKS](#)) deployment option for [AWS Local Zones](#), [AWS Outposts](#), [AWS Snow Family devices](#), and [AWS Wavelength](#) are also discussed in this paper but detailed best practice guidance is out of scope for these deployment options.

# Introduction

Organizations are increasingly choosing Kubernetes as their container orchestration because of its extensibility, open-source support, and portability. [Amazon Elastic Kubernetes Service](#) (Amazon EKS) makes it easier to run Kubernetes clusters on AWS by providing a production-grade managed Kubernetes control plane. Amazon EKS runs an upstream and certified conformant version of Kubernetes, which is designed to be performant, reliable, and secure.

Amazon EKS provides the most trusted way to start, run, and scale Kubernetes. Amazon EKS provides a spectrum of Kubernetes deployment options ranging from AWS managed to customer-managed infrastructure:

- Amazon EKS
- Amazon EKS on AWS Outposts
- Amazon EKS in AWS Local Zones
- Amazon EKS in AWS Wavelength Zones
- Amazon EKS Anywhere on Snowball Edge
- Amazon EKS Anywhere on customer managed hardware
- Amazon EKS Distro



[Amazon EKS](#) uses [Amazon EKS Distro](#) (EKS-D), which is an open-source Kubernetes distribution designed to create reliable and secure self-managed Kubernetes clusters. Amazon EKS Distro includes binaries and containers of open-source Kubernetes, etcd (i.e., cluster configuration database), networking, and storage plugins, and tested for compatibility. The project includes the latest upstream updates and extended security patching support. Customers can access EKS-D releases on [GitHub](#) or within AWS via Amazon Simple Storage Service ([Amazon S3](#)) and Amazon Elastic Container Registry ([Amazon ECR](#)) for a common source of releases and updates.

For customers that want to run Kubernetes clusters on-premises, [Amazon EKS Anywhere](#) is a deployment option. Amazon EKS Anywhere allows customers to create and operate Kubernetes clusters on-premises on customer hardware. Just like Amazon EKS, Amazon EKS Anywhere also uses EKS-D. It provides cluster management Command Line Interface (CLI) tooling to simplify create, delete, and upgrade of Kubernetes clusters on-premises. Customers can get support for Amazon EKS Anywhere clusters by purchasing an Amazon [EKS Anywhere Enterprise Subscription](#).

- Amazon EKS Anywhere has the following benefits:
- Simplifies on-premises Kubernetes management with default component configurations and automated cluster management tools.
- AWS Support for customers with an Amazon EKS Anywhere Enterprise Subscription.
- Flexible deployment options.
- Can be deployed in disconnected (*air-gapped*) environments.
- Upstream Kubernetes conformant with open-source Kubernetes projects such as GitOps, FluxCD and opinionated set of curated packages including (Harbor, MetalLB, and more).
- AWS-packaged software reducing effort spent testing and managing self-managed Kubernetes versions.

# Use cases for Amazon EKS Anywhere

Many organizations use Kubernetes as a uniform run-time and management layer across multiple environments. When operating in the cloud, AWS recommends Amazon EKS as the preferred deployment option for Kubernetes clusters. While cloud remains the most popular destination for running Kubernetes applications, there are specific use cases that require running workloads on-premises. Below are a few reasons why customers choose Amazon EKS Anywhere.

## Data sovereignty

AWS customers within the global customer base, have to comply with data protection regulations of applicable jurisdictions, which may limit egress of their customers' personally identifiable information outside of the borders of a country or region. Organizations can use AWS Regions to run workloads within borders and limit cross-border data transfer.

Customers can consider Amazon EKS Anywhere when operating in countries without an AWS Region, and when [AWS Outposts](#), [AWS Local Zones](#), or [AWS Wavelength](#) don't meet the customer's use-case requirements. Amazon EKS Anywhere deploys Kubernetes cluster on customer-managed infrastructure, which allows customers to control data storage and help them comply with applicable regulations.

## Using existing investments in hardware and data center space

Cloud migration is a multi-step journey for most organizations. Most cloud migrations don't happen overnight. Previously, IT departments must support cloud and on-premises simultaneously. As they move workloads from on-premises to cloud, customers are faced with questions like, "what to do with the existing hardware?" and "can we modernize applications before moving to the cloud?".

With Amazon EKS Anywhere, customers can use existing investments in hardware and data center space. They can use Kubernetes as an abstraction to manage on-premises and cloud deployments consistently.

Customers can also containerize and validate applications on-premises. Amazon EKS Anywhere and Amazon EKS Distro provide customers with a conformant Kubernetes cluster that they can use to develop and test applications prior to moving to the cloud.





## Low latency requirements

Many customers in industries like online gaming, media & entertainment, healthcare, financial services, and the public sector have applications that require low latency. When an AWS Region isn't close enough or within a geopolitical boundary to meet these requirements, customers can use [AWS Local Zones](#) and [AWS Wavelength](#).

Amazon EKS supports low-latency workloads that take advantage of Local Zones and Wavelength Zones. When customers cannot use an AWS Region, Local Zones, or AWS Wavelength, they can use Amazon EKS Anywhere to run workloads in self-managed data centers. As the AWS global footprint is constantly expanding, Amazon EKS Anywhere customers have a simplified migration path to the cloud when a Region, Local Zone, or Wavelength Zone becomes available.



The [AWS Global Infrastructure](#) map as of 30/01/2023.

## Disconnected hybrid and edge locations

Some customers have deployments in environments with inconsistent connectivity to the internet. Workloads deployed in these locations are designed to operate even while resources outside of the local network are inaccessible.

Customers can architect Amazon EKS Anywhere clusters for secure applications in disconnected environments. When cloud connectivity is interrupted, customers can still access Amazon EKS Anywhere clusters from the local network. The cluster and applications continue to run during network disconnects. Customers can use the following mechanisms to make Kubernetes clusters resilient to intermittent network connectivity:

- Host container images, Helm charts, and related artifacts in a local registry.
- Maintain a local copy of data required for the workloads.
- If cluster authentication is configured to use remote identity providers, such as AWS Identity and Access Manager [\(AWS IAM\) Authenticator](#), configure backup authentication for disconnected state (x509 certs).
- If observability solution relies on aggregation of metrics and logs in a remote location, establish backup troubleshooting mechanisms based on local Kubernetes-native and metrics HTTP endpoints and access to local logs.
- Commercial products deployed with [AWS Marketplace for Containers Anywhere](#) and integrated with [AWS License Manager](#) (or metering Application Programming Interface [API]) may not work in fully disconnected mode.

## Local data processing and filtering

Many industries such as manufacturing, media, and telecommunications have applications that produce data in large volumes. There are cases in which customers have to maintain raw data sets in large quantities on-premises. When uploading data to the cloud, customers use tools such as Apache Airflow and Apache Spark to process and filter data. Sending a subset reduces data transfer time and cost.

Kubernetes makes it easier to deploy and operate data processing pipelines. Customers can deploy Amazon EKS Anywhere in data centers or edge locations to process data locally before sending it to the cloud.

## Amazon EKS deployment options

This section presents an overview of the different Amazon EKS deployment options.

















	EKS Distro	EKS Anywhere	EKS-A on Snow	EKS on Outposts	EKS
					
Hardware	Customer	Customer			
Location	On-prem	On-prem	Edge	On-prem	
Control plane location	On-prem	On-prem	Edge	Region or Outpost	
Data plane location	On-prem	On-prem	Edge	On-prem	
Support	Community				
Required Region Connectivity	No	No	No	Yes	Yes
					
	Customer Managed				AWS Managed

Figure: The different deployment options in Amazon EKS portfolio across hardware, physical location, control plane location, data plane location, support and region connectivity.

## Amazon EKS on AWS Outposts

AWS Outpost Rack is part of the [AWS Outposts family](#). AWS Outposts offers the same AWS infrastructure, AWS services, APIs, and tools to virtually any on-premises data center or co-location center for a truly consistent hybrid experience.

Amazon EKS supports deploying worker nodes on AWS Outpost Rack. In this deployment, the Kubernetes control plane runs in an AWS Region, while the nodes run on AWS Outposts in customer-managed data centers. The Amazon EKS service communicates through the network with the nodes running on the AWS Outposts machine.

In this scenario, if there is poor or intermittent connectivity to the AWS region running Amazon EKS, AWS recommends local clusters for Amazon EKS on AWS Outposts. During periods of disconnection, when a Kubernetes cluster is unable to communicate with its nodes, the cluster may consider the nodes unhealthy and schedule them for replacement. This may lead to application downtimes when connectivity is restored (see [Pod eviction in Kubernetes documentation](#)).

## Local clusters for Amazon EKS on AWS Outposts

While Amazon EKS on AWS Outpost Rack allows customers to run just the worker nodes on-premises, Amazon EKS local clusters run both the Kubernetes control plane and data plane on-premises on the AWS Outpost. This helps mitigate the risk of application downtime that might result from AWS Outpost temporary network disconnects to the cloud.

You can perform Kubernetes operations during network disconnects to the cloud. For more information, see [Preparing for network disconnects](#).

## Amazon EKS on AWS Local Zones

AWS Local Zones are a type of infrastructure deployment that places compute, storage, database, and other select AWS services close to large population and industry centers. An AWS Local Zone is an extension of an AWS Region.

Customers deploy applications in Local Zones to locate applications in geographic proximity to users. Resources created in a Local Zone can serve local users with low-latency communications. For more information, see [Local Zones](#).

Amazon EKS supports certain resources in Local Zones. This includes [self-managed Amazon Elastic Compute Cloud \(Amazon EC2\) nodes](#), Amazon Elastic Block Store ([Amazon EBS](#)) volumes, and Application Load Balancers (ALBs). Worker nodes deployed in Local Zones are part of an Amazon EKS cluster running in an AWS Region.

## Amazon EKS on AWS Wavelength

AWS Wavelength allows customers to run applications that deliver ultra-low latencies to mobile devices and end users. AWS Wavelength embeds AWS compute and storage services within 5 G networks, providing mobile edge computing infrastructure for developing, deploying, and scaling ultra-low-latency applications.

Customers with applications that require low latency or local data processing can deploy workloads on Amazon EC2 instances in Wavelength Zones. Amazon EKS clusters support Kubernetes worker nodes deployed in Wavelength Zones. For more information, see [AWS Wavelength](#).

# Amazon EKS Anywhere deployment options

This section presents an overview of the different Amazon EKS Anywhere deployment options:

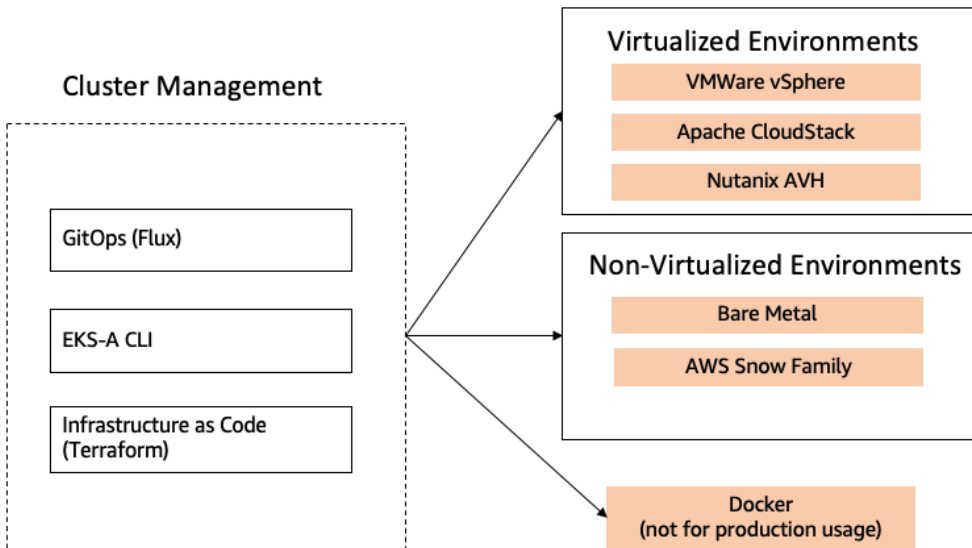


Figure: Amazon EKS Anywhere deployment options.

## Amazon EKS Anywhere on Docker

Amazon EKS Anywhere supports a Docker provider for *development and testing use cases only*. This allows customers to try Amazon EKS Anywhere on local system before deploying to a supported provider. To install the Amazon EKS Anywhere binaries and see system requirements please follow the [installation guide](#). This deployment type is not recommended for production.

## Virtualized Amazon EKS Anywhere deployment options

### Amazon EKS Anywhere on VMware

Customers using VMware vSphere to manage their virtualized infrastructure can deploy a production-grade Kubernetes cluster using Amazon EKS.

VMware vSphere provider for Amazon EKS Anywhere helps customers deploy a highly available Kubernetes control plane and an etcd cluster running in virtual machines.

## Amazon EKS Anywhere on Apache CloudStack

Amazon EKS Anywhere provides support for creating Kubernetes clusters on Apache CloudStack. Apache CloudStack, a project of the Apache Software Foundation, is open-source software designed to deploy and manage large networks of virtual machines.

Amazon EKS Anywhere uses the new [Cluster API provider for Apache CloudStack](#) (CAPC) to provide customers declarative, Kubernetes-style APIs for cluster creation, configuration, and management. AWS collaborated with the Apache CloudStack community to build CAPC, which is now part of the Kubernetes Special Interest Group (SIG).

## Amazon EKS Anywhere on Nutanix

Amazon EKS Anywhere provides support for creating Kubernetes clusters on [Nutanix Cloud Infrastructure](#) with Nutanix Acropolis Hypervisor (AVH). This enables customers to deploy on Nutanix and benefit from the Nutanix unified data services which provides file, block and Amazon S3 compatible object storage. Customers can use the Nutanix Container Storage Interface (CSI) driver to interface with file and block storage.

Amazon EKS Anywhere uses the [Cluster API provider for Nutanix](#) (CAPX) to provide Kubernetes style APIs for cluster creation, configuration, and management.

## Non-virtualized Amazon EKS Anywhere deployment options

### Amazon EKS Anywhere on bare metal

Amazon EKS Anywhere provides support for creating Kubernetes clusters on bare metal servers. This deployment type enables customers to run Kubernetes applications directly on hardware, without virtualization. Applications running in this mode have direct access to the physical hardware and compute accelerators, local storage, and native input/output (I/O) speeds.

Amazon EKS Anywhere uses [Tinkerbell](#) (a Cloud Native Computing Foundation [CNCF] sandbox project) for server bootstrapping and cluster size, networking, and software configuration. Please see Amazon EKS Anywhere [documentation](#) for requirements, hardware support, and installation guide.

### Amazon EKS Anywhere on AWS Snowball Edge

Amazon EKS Anywhere provides the ability to deploy and operate a fault-tolerant and highly available Kubernetes cluster [AWS Snow Family](#) devices. Snow devices are AWS-supported hardware devices that customers can deploy in datacenters and at the edge.



Customers can order a supported AWS Snow Family device with Amazon EKS Anywhere preinstalled. Amazon EKS Anywhere is designed to create Kubernetes clusters that use multiple co-located AWS Snow Family devices for high availability.

## Amazon EKS Anywhere architecture

### Amazon EKS Anywhere control plane

Amazon EKS Anywhere creates clusters based on Amazon EKS Distro. Amazon EKS Anywhere supports creating multiple control plane nodes to provide a highly available control plane. The Kubernetes control plane runs [the Kubernetes API server](#) and controllers such as [kube-scheduler](#), [kube-controller-manager](#), and uses `kube-vip` for distributing Kubernetes API server requests.

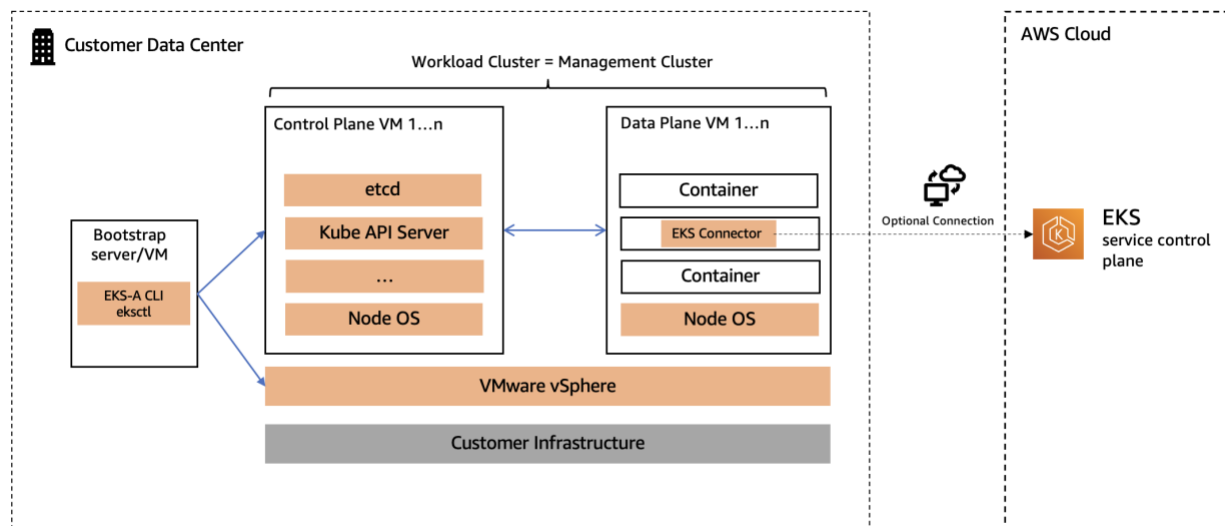


Figure: Amazon EKS Anywhere Control and Data Plane

Amazon EKS Anywhere supports two types of etcd topologies:

- **Stacked:** The etcd members and control plane components are co-located (run on the same node/machines)
- **Unstacked/External:** With the unstacked or external etcd topology, etcd members have dedicated machines and aren't co-located with control plane components



AWS recommends unstacked etcd topology [\(if supported\)](#) for a highly available (HA) cluster for the following reasons:

- External etcd topology decouples the control plane components and etcd member. This topology ensures that a Kubernetes control plane node or component (like `kube-apiserver`) failure won't directly impact an etcd member.
- etcd is a resource intensive process. Running it on dedicated nodes ensures that etcd resource consumption won't affect other Kubernetes control plane components.

## Cluster API Kubernetes (CAPI)

[Kubernetes Cluster API](#) or CAPI is a Kubernetes SIG (Special Interest Group) project that focuses on providing declarative APIs and tooling to simplify provisioning, upgrading, and operating multiple Kubernetes clusters. The two key objectives of CAPI project are:

- Manage the lifecycle (create, scale, upgrade, destroy) of Kubernetes-conformant clusters using a declarative API
- Support multiple environments such as on-premises and cloud.

Amazon EKS Anywhere uses an infrastructure provider model for creating, upgrading, and managing Kubernetes clusters that uses the [Kubernetes Cluster API](#) project.

Amazon EKS Anywhere provider for VMware vSphere is based on the Kubernetes Cluster API Provider vSphere (CAPV) specifications. Similarly, Amazon EKS Anywhere supports Cluster API for Docker Provider (CAPD) for creating development and test workload clusters. The Amazon EKS Anywhere project wraps Cluster API, various other CLIs and plugins (`eksctl cli`, `anywhere plugin`, `kubectrl`, `aws-iam-authenticator`), and bundles them in a single package to simplify the creation of workload clusters.

## Amazon EKS Anywhere data plane

The Kubernetes data plane is composed of nodes that provide the compute, storage, and network resources for running workloads as Kubernetes pods.

Amazon EKS Anywhere provides [Bottlerocket](#), a Linux-based open-source operating system built by AWS and is the default only fully supported for Amazon EKS Anywhere nodes. Bottlerocket OVAs and raw images are distributed by the Amazon EKS Anywhere project. Alternatively, Amazon EKS Anywhere has been tested with Ubuntu-



based and Red Hat Enterprise Linux (RHEL) based nodes. Amazon assists with troubleshooting and configuration guidance with Ubuntu-based and RHEL-based nodes under the Amazon [EKS Anywhere Enterprise Subscription](#). Please see the [documentation](#) for building Ubuntu-based and RHEL-based nodes.

## Amazon EKS Anywhere networking

Amazon EKS Anywhere uses an optimized bundle of Cilium as the default Container Network Interface (CNI) plugin. Cilium provides networking capabilities such as Pod-to-Pod connectivity, in-cluster routing, and Pod IP address management (IPAM).

Cilium agent run as DaemonSets on every Amazon EKS Anywhere node. Amazon EKS Anywhere clusters also include a Cilium operator deployment to handle certain cluster-wide operations.

In Amazon EKS Anywhere, the Cilium agent implements an overlay network using the Generic Network Virtualization Encapsulation (Geneve) network encapsulation protocol on User Datagram Protocol (UDP) port 6081. Firewalls must permit traffic on UDP port 6081 between nodes.

Cilium health check uses Internet Control Message Protocol Type 0/8, Code 0 or TCP port 4240 to validate node-to-node connectivity. Cilium health check monitoring is optional feature and does not affect Cilium functionality.

Amazon EKS Anywhere supports [Cilium Network Policies](#). Amazon EKS Anywhere doesn't support all the features as part of the wide Cilium project.

## Secure connectivity to AWS Cloud

Amazon EKS Anywhere clusters are designed to operate in environments with or without connectivity to the AWS Cloud. Amazon EKS Anywhere customers might operate a hybrid environment with a set of applications running on-premises while the primary environment runs in the cloud.

[Amazon Virtual Private Cloud \(Amazon VPC\)](#) provides multiple network connectivity options for connecting remote networks to customers environment. These options are useful for integrating AWS resources with Amazon EKS Anywhere (for example, monitoring, authentication, security, data, or other systems).

Customers can securely connect on-premises workloads with services running in their VPC by using [AWS Direct Connect](#) or VPN.



## Amazon EKS Connector and Amazon EKS console

The Amazon EKS Console gives customers a central dashboard to see the status of registered Kubernetes clusters, applications, and associated cloud resources. Amazon EKS Anywhere customers can install the [Amazon EKS Connector](#) to visualize Amazon EKS Anywhere clusters in the Amazon EKS Console.

The [Amazon EKS Connector is an open source tool](#) designed to register and connect any conformant Kubernetes cluster to AWS and visualize it in the Amazon EKS Console. Customers can see the status, configuration, and workloads for connected clusters in the Amazon EKS console.

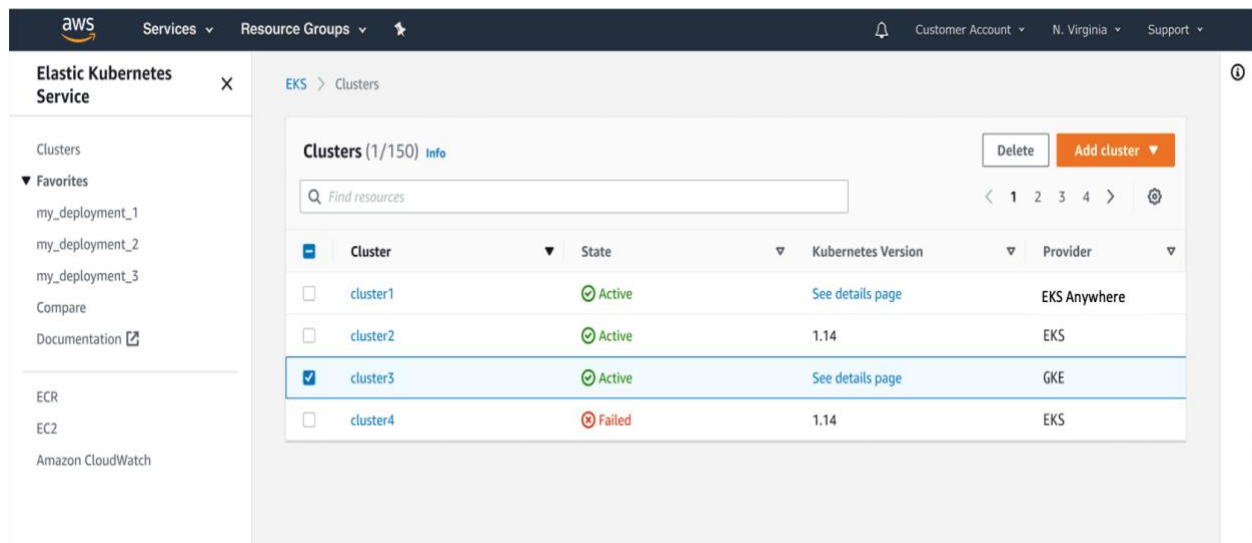


Figure: Visualize Amazon EKS Anywhere clusters on the Amazon EKS console

## Amazon EKS Anywhere curated packages

[Amazon EKS Anywhere curated packages](#) are Amazon curated software packages that extend the core functionalities of Kubernetes on Amazon EKS Anywhere clusters. Curated packages simplify the installation and management of common Kubernetes tools such as Harbor, MetalLB, and Emissary Ingress. The curated packages CLI installs packages and is designed to provide version compatibility.

Amazon EKS Anywhere curated packages are:

- **Amazon-built:** All container images of the packages are built from source code by Amazon. Open Source Software (OSS) package images are built from the open source upstream.

- **Amazon-scanned:** Amazon continuously scans the container images for security vulnerabilities and provides remediation.
- **Amazon-signed:** Amazon signs the package bundle manifest (a Kubernetes manifest) for the list of curated packages. The manifest is signed with AWS Key Management Service (AWS KMS) managed private keys. The curated packages are installed and managed by a package controller on the clusters. Amazon provides validation of signatures through an admission control webhook in the package controller and the public keys distributed in the bundle manifest file.
- **Amazon-tested:** Amazon tests the compatibility of all curated packages including the OSS packages with each new version of Amazon EKS Anywhere.
- **Amazon-supported:** All curated packages including the curated OSS packages are supported under the Amazon EKS Anywhere Enterprise Subscription.

Amazon EKS Anywhere curated packages are available to customers with the Amazon EKS Anywhere Enterprise Subscription.

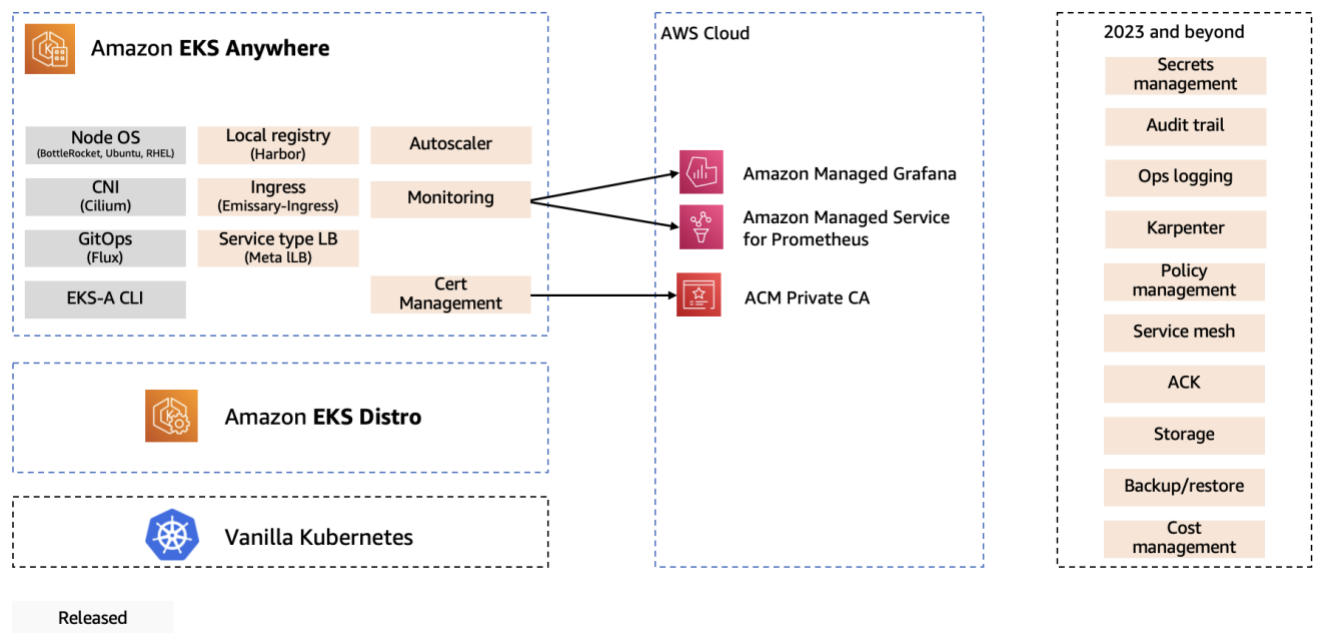


Figure: Amazon EKS Anywhere curated packages overview.

## Open-source components

Amazon EKS Anywhere integrates with a number of third-party vendor components, including Ubuntu LTS and Red Hat Enterprise Linux as supported operating systems,

Cilium (CNI), and FluxCD (GitOps for Cluster Management). It also provides flexibility for customers to integrate with their choice of tools in other areas.

As with any standard Kubernetes distribution, customers can count on the open platform that enables extensibility and replacement of the platform with other Kubernetes compliant components; for instance, customers can use [Calico](#) for CNI instead of Cilium if necessary.

It's important to note that AWS supports such third-party components on a best efforts basis. Customers may also purchase enterprise subscriptions from third party vendors to obtain additional support for these components. Please see [Amazon EKS Anywhere Partners](#) for the information on vendor supported integrations.

The following components and tools were validated with Amazon EKS Anywhere by AWS and AWS Partners:

	Amazon supported curated packages	Other supported options recommended by Amazon
Infrastructure as code		<a href="#">Terraform</a>
Ingress controller	<a href="#">Emissary</a>	<a href="#">NGINX ingress controller</a>
Service type load balancer	<a href="#">MetalLB</a>	<a href="#">KubeVip</a>
Local container image repository	<a href="#">Harbor</a>	
Monitoring		<a href="#">Prometheus</a> , <a href="#">Grafana</a> , and <a href="#">Pixie</a>
Logging		<a href="#">Fluent Bit</a>
Operating system		<a href="#">Bottlerocket</a> , <a href="#">Ubuntu</a>
Policy agent		<a href="#">Open Policy Agent (OPA)</a> , <a href="#">Kyverno</a>
Service mesh		<a href="#">Istio</a> , <a href="#">Linkerd</a>

	Amazon supported curated packages	Other supported options recommended by Amazon
Cluster backup and restore		<a href="#">Velero</a>
Networking	<a href="#">Cilium</a>	<a href="#">Calico</a>
Object storage		<a href="#">MinIO</a>
Secrets management		<a href="#">Hashicorp Vault</a>

## Amazon EKS Anywhere cluster lifecycle

Each Amazon EKS Anywhere cluster is built from a cluster specification file, with the structure of the configuration file based on the target provider for the cluster.

The following diagram provides a rough sequence of events for the lifecycle of an Amazon EKS Anywhere cluster for creating and upgrading a target Amazon EKS Anywhere cluster.

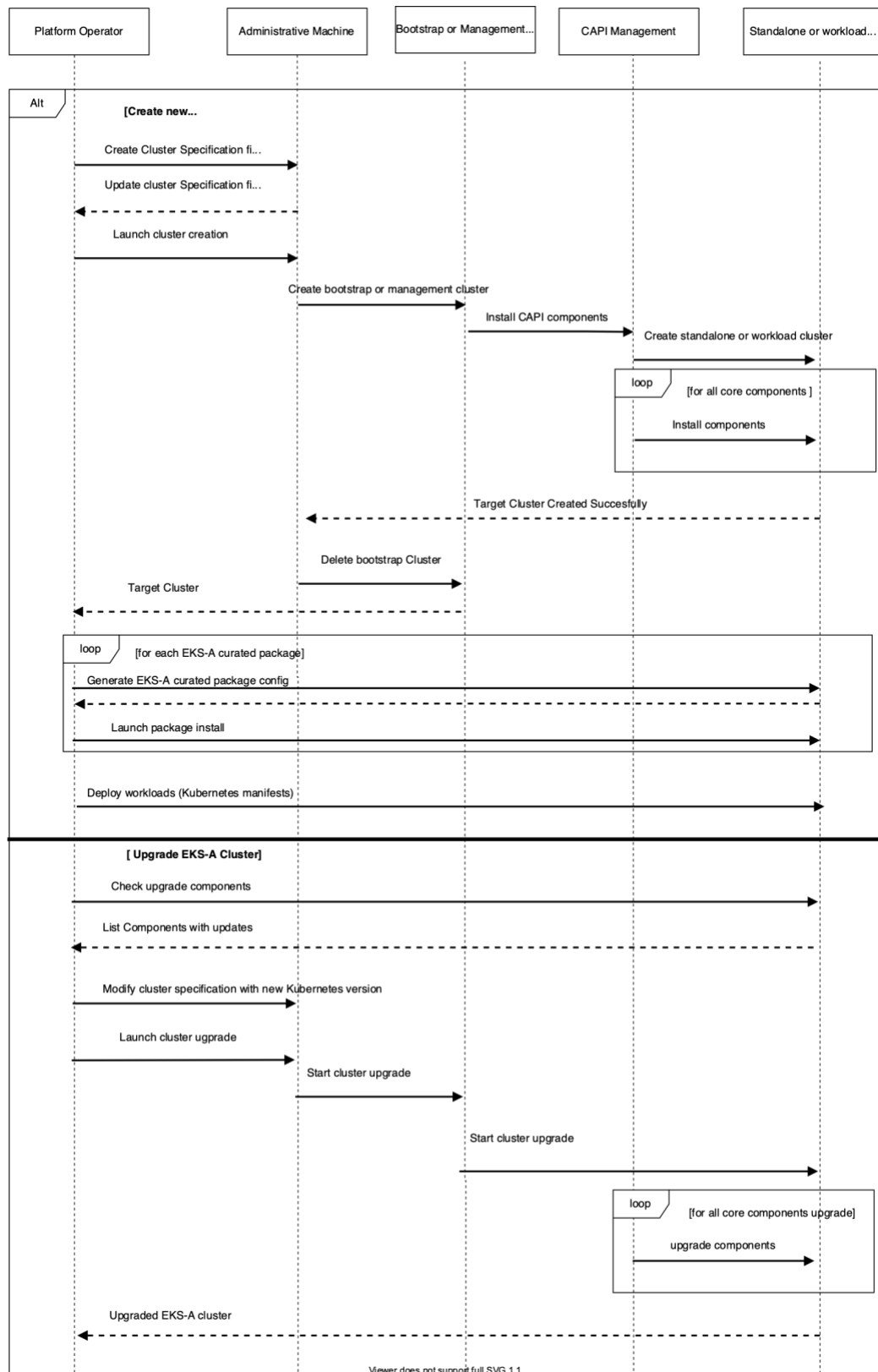


Figure: Amazon EKS Anywhere cluster lifecycle creation workflow

The first flow in the sequence diagram shows cluster creation workflow. In this process, the platform operator first generates a cluster specification file for their chosen Amazon EKS Anywhere deployment option using the `eksctl`. The platform operator modifies the generated cluster specification file (including Control Plane Data Plane sizing) to meet their requirements. For bare metal, the platform operator also needs to create an [hardware inventory file](#). Next, the platform operator can start the cluster creation process.

The cluster creation process starts by creating a temporary `Kind` Kubernetes bootstrap cluster. This typically runs on an administrative machine. Once the bootstrap cluster is created the Cluster API components are installed on the bootstrap cluster, these components orchestrate the creation of the target cluster. This includes setting up the control and data plane, installing core networking, and storage components. Finally, the bootstrap cluster installs CAPI on the target cluster and moves the CAPI objects over to the target cluster, so it can take over the management of itself. The process ends with the bootstrap cluster being deleted and `kubeconfig` created, which the platform operator can use to authenticate with the created Kubernetes cluster. Additionally, the platform operator can choose install Amazon EKS Anywhere [curated packages](#) (e.g., Emissary Ingress controller, Harbor container registry) to expand the functionality of their cluster.

The second flow in the sequence diagram shows the cluster upgrade workflow, in this process the platform operator first uses the `eksctl` to check the list of Amazon EKS Anywhere components (cluster-api, etcdadm-controller, etcdadm-bootstrap, vsphere, flux) for new releases for the specified target cluster. The platform operator then updates the Kubernetes version in the cluster specification file and starts the upgrade process. A new Amazon EKS Anywhere bootstrap cluster is created to trigger the upgrade process if there is no management cluster. On a successful control plane and data plane upgrade, all the core components are upgraded. See [Handling cluster upgrades](#) for details on the upgrade process.

For a more detailed view of the cluster creation workflow, for the different Amazon EKS Anywhere deployment options see the Amazon EKS Anywhere [documentation](#)

## Management and workload cluster deployment architecture

Amazon EKS Anywhere has two deployment topology options:

- A *standalone cluster* that contains Cluster API (CAPI) management components so it can self-manage (e.g., performing cluster update on itself).

- A *management-workload-cluster* topology where a management cluster creates and manages a fleet of workload clusters.

The diagram below provides a graphical representation of a management and workloads cluster architecture.

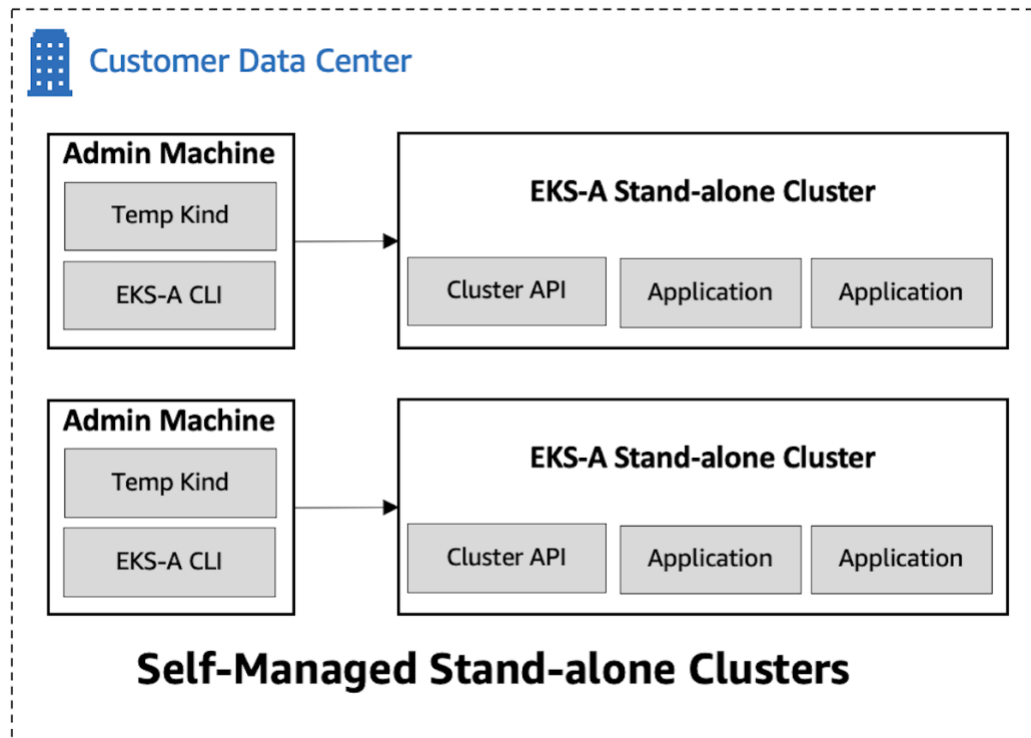


Figure: Amazon EKS Anywhere self-managed stand-alone clusters topology



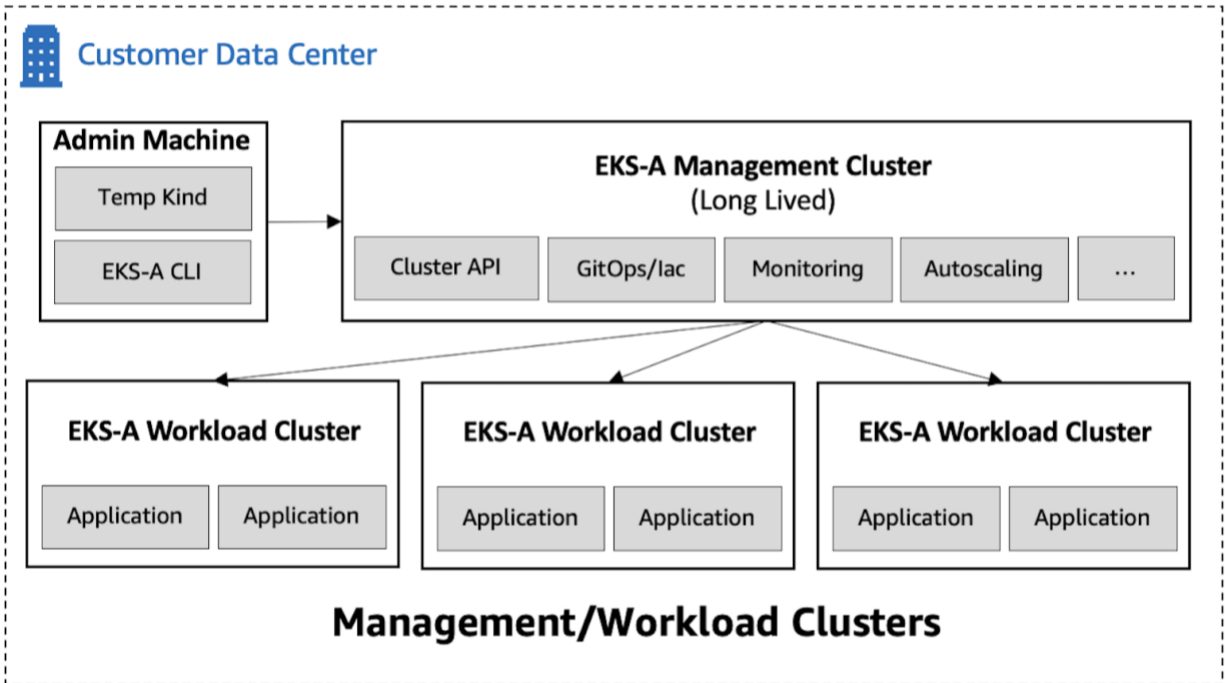


Figure: Amazon EKS Anywhere management/workload cluster topology

An Amazon EKS Anywhere management cluster is a long-lived, on-premises Kubernetes cluster that can create and manage a fleet of Amazon EKS Anywhere workload clusters where customers run their applications.

The management cluster can only be created and managed by `eksctl`. It runs on customer-managed hardware on-premises and does not require connectivity back to AWS. Customers are responsible for operating the Amazon EKS Anywhere management cluster including patching, upgrading, scaling, and monitoring the cluster control plane and data plane.

For customers who want to run three or more Amazon EKS Anywhere clusters, AWS recommends a management-workload-cluster deployment topology because of the advantages listed in the table below. The Amazon EKS Anywhere curated packages feature will also recommend deploying certain packages such as the container registry package or monitoring packages on the management cluster to avoid circular dependency.

	Standalone cluster topology	Management-workload-cluster topology
<b>Pros</b>	<p>Optimizing hardware resource allocation.</p> <p>Reduced operational overhead of maintaining a separate management cluster</p>	<p>Isolation of secrets.</p> <p>Resource isolation between different teams.</p> <p>Reduced noisy-neighbor effect. Isolation between development and production workloads. Isolation between applications and fleet management services (e.g., monitoring server or container registry. Provides a central control plane and API to automate cluster lifecycles)</p>
<b>Cons</b>	<p>Shared secrets such as SSH credentials or VMware credentials across all teams who share the cluster.</p> <p>Without a central control plane (e.g., a parent management cluster), not possible to automate cluster creation/deletion with advanced methods like GitOps or IaC.</p> <p>Circular dependency if the cluster has to host monitoring server or local container registry</p>	<p>Consumes extra resources. The creation/deletion of the management cluster itself can't be automated through GitOps or IaC.</p> <p>Not able to manage multiple workload clusters across different datacenters/vSphere clusters.</p>

## Well architected best practices

### Security

AWS Cloud Services follow the [Shared Responsibility Model](#), where AWS is responsible for security of the cloud, while the customer is responsible for security in the cloud. However, Amazon EKS Anywhere is an open-source tool and the distribution of responsibility differs from that of a managed cloud service like Amazon EKS.

Management of Amazon EKS Anywhere is a shared responsibility. AWS provides cluster management tooling; however, the underlying infrastructure is a customer responsibility. Customers are responsible for securing Amazon EKS Anywhere clusters and workloads running in it.

## **AWS responsibilities**

AWS is responsible for vetting and securely sourcing the services and tools packaged with Amazon EKS Anywhere (such as Bottlerocket, CoreDNS, Cilium, Flux, CAPI, and [govc](#)).

The Amazon EKS Anywhere supply chain as well as build and delivery infrastructure are secured to the standard of any AWS service.

AWS is responsible for the secure development and testing of the Amazon EKS Anywhere controller, its associated custom resource definitions, and the Amazon EKS Anywhere CLI.

## **Customer responsibilities**

Securing, updating, and maintaining Amazon EKS Anywhere clusters and the underlying infrastructure are customer responsibilities. AWS recommends securing access to Kubernetes control plane and data plane as described in [Kubernetes security best practices](#).

Amazon EKS Anywhere cluster control plane audit logs can be found at `/var/log/kubernetes/api-audit.log` on the Kubernetes control plane nodes. Customers can obtain the logs by connecting to the nodes. Customers can establish remote access to the nodes via single sign-on (SSH) using the SSH private-key pair specified during Amazon EKS Anywhere cluster creation or the keys generated during the Amazon EKS Anywhere cluster creation process.

Most recommendations documented in the [Amazon EKS Best Practices Guide for Security](#) also apply to Amazon EKS Anywhere clusters. The Amazon [EKS Anywhere documentation](#) includes security best practices that specifically applicable to Amazon EKS Anywhere clusters.

*eksctl anywhere creates a kubeconfig file at cluster creation. This file grants administrative privileges to the bearer. Following the principle of least privilege, cluster administrators should use scoped roles for operating and performing maintenance, and avoid using administrative roles for routine operations.*

Amazon EKS Anywhere provides controls that customers can use to create Kubernetes clusters to help them comply with regulatory requirements.

## Cluster authentication

Amazon EKS Anywhere clusters support OpenID Connect (OIDC) integration, including [Azure Active Directory \(Azure AD\)](#), [Active Directory Federated Services \(ADFS\)](#). This enables customers to federate through ADFS, Azure AD or other Identify Providers (IDP) that support OIDC standard. First, the user authenticates with the OIDC compatible identity provider and fetches JSON Web Token (JWT). Next, the user then passes this JWT bearer token to the Kubernetes control plane API server and assumes a Kubernetes role.

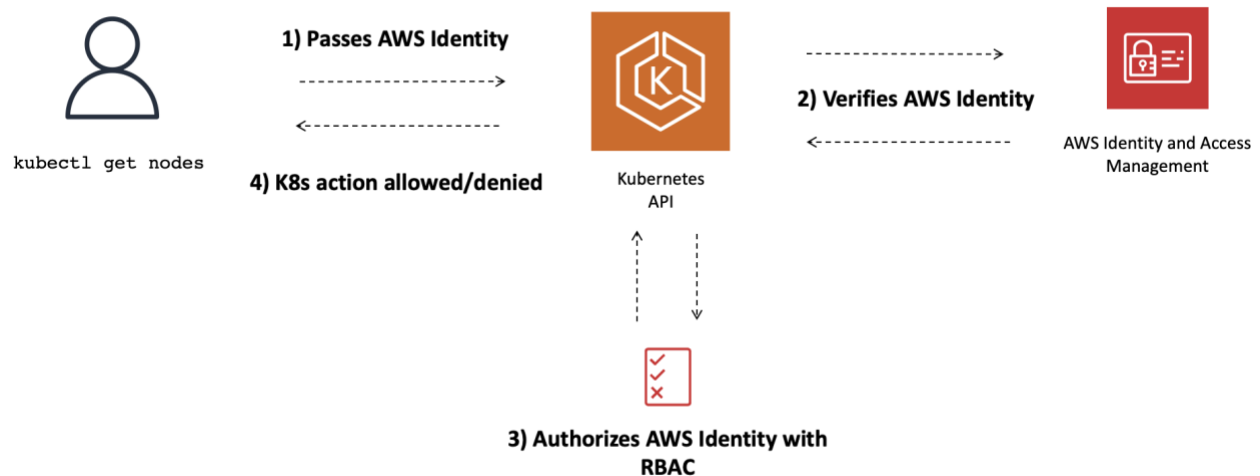


Figure: Overview of Kubernetes API server authentication and authorization flow.

In addition to external OIDC support for authentication, Amazon EKS Anywhere enables configuring [AWS IAM authenticator](#) to authenticate users to the cluster via AWS Identity and Access Management (AWS IAM), users. Follow this [guide](#) to get started.

## Network security

Amazon EKS Anywhere runs on customer-managed infrastructure, which puts customers in-charge for restricting network access to Kubernetes nodes, Pods, and load balancers. AWS recommends customers allow Amazon EKS Anywhere clusters to connect to the AWS Cloud. Customers can deploy Amazon EKS Anywhere in *air-gapped* environments that don't allow connectivity to resources outside of the private network; however, AWS recommends permitting connectivity to an AWS Region. Restricting connectivity to an AWS Region prevents cluster management using the Amazon EKS Console and integration with AWS services (e.g., Amazon IAM, Amazon Managed Service for Prometheus, etc). Users on private network can still manage Amazon EKS Anywhere clusters using tools like kubectl when cloud connectivity is unavailable.

Any Pod in a Kubernetes cluster can connect to another Pod by default. Security best practices recommend restricting network access to sensitive workloads within a cluster. Customers can use Kubernetes network policies that are designed to implement traffic restriction between Pods and other network entities (such as services, endpoints, and IP addresses). Alternatively, customers can also isolate workloads by deploying them in dedicated clusters.

Amazon [EKS Anywhere uses Cilium CNI](#) implements the Kubernetes Network Policy specification for L3 and L4 level. Please see [Cilium documentation](#) for more details.

Many [advanced features](#) of Cilium are not yet enabled as part of Amazon EKS Anywhere, including Hubble observability, DNS-aware and HTTP-Aware Network Policy, Multi-cluster Routing, Transparent Encryption, and Advanced Load-balancing. However, these features are supported by Cilium Enterprise and Cilium Enterprise is supported on Amazon EKS Anywhere. Customers can purchase Cilium Enterprise from Isovalent through AWS Marketplace for Container Anywhere.

## AWS IAM roles for pods

Applications running in Amazon EKS Anywhere clusters can use an [AWS SDK](#) or the AWS Command Line Interface ([AWS CLI](#)) to make API requests to AWS services using AWS Identity and Access Management ([AWS IAM](#)) permissions. AWS IAM roles for service accounts provide the ability to manage credentials for applications without distributing AWS credentials.

AWS IAM roles for service account (IRSA) enables applications running in Amazon EKS and Amazon EKS Anywhere clusters to authenticate with AWS services using AWS IAM roles.

In order to enable IRSA for Amazon EKS Anywhere cluster, customers should create an OIDC provider for the cluster and host cluster's public service account signing key. Please see AWS [IAM for Pods configuration](#) for more information on setting up IRSA on Amazon EKS Anywhere cluster.

## **Bottlerocket**

Bottlerocket is an open-source Linux-based operating system from Amazon that was purpose built for running containers with a strong emphasis on security. Amazon EKS Anywhere uses Bottlerocket as the default operating system for Kubernetes nodes.

Bottlerocket improves security posture by removing all shells, interpreters, and package managers from the Bottlerocket image. It provides access controls such as Linux capabilities and SELinux, and integrity checks such as dm-verity. Customers can enable kernel lockdown to prevent unsigned kernel modules from being loaded or prevent userspace applications from reading kernel memory. Please see [Bottlerocket security guidance](#) to learn more about securing Bottlerocket.

## **Cluster upgrades**

Amazon EKS Anywhere CLI makes it easy to update cluster and core components. Please see [Handling cluster upgrades](#) for details.

Kubernetes has minor releases [three times per year](#) and Amazon EKS Distro follows the cadence. Upgrading Amazon EKS Anywhere clusters are a customer responsibility. To improve security, customers should avoid running outdated clusters and unsupported versions.

Customers can learn more about AWS version support policy and releases on Amazon [EKS Anywhere documentation](#).

Amazon EKS Anywhere is based on a number of components, including hardened open-source components, components authored by AWS as well as by AWS Partners. If any of the Amazon EKS Anywhere components that require a security patch or an urgent bug fix, then AWS will release a patch version. If the external components such as Cilium or Bottlerocket have a normal release (as opposed to an urgent bug fix), AWS will include such updates within the regular minor release cycle, covering full quality assurance activities.



Amazon EKS Anywhere [Release Alerts](#) that can be used both as a source of up-to-date notifications and as an event source to trigger custom upgrade workflows.

## Cost optimization

Amazon EKS Anywhere is available as open-source software. There are no upfront commitments or fees to use Amazon EKS Anywhere. Customers can optionally purchase an Amazon EKS Anywhere Enterprise Subscription to get support for Amazon EKS Anywhere cluster components as well as all bundled open-source tooling. [AWS Enterprise Support](#) or [Enterprise On-Ramp Support](#) is a prerequisite for purchasing an Amazon EKS Anywhere Enterprise Subscription. Please see [Amazon EKS Anywhere pricing](#) for details.

Customers can optimize Amazon EKS Anywhere clusters for cost by right-sizing the underlying VMs and bare metal servers. Monitoring systems like [Prometheus](#) provide resource utilization statistics that are helpful in sizing nodes. Amazon EKS Anywhere control plane and data plane can be scaled horizontally and vertically.

Amazon EKS Anywhere default operating system Bottlerocket provides a minimal operating system that's optimized for running containers. Customers can use Bottlerocket to reduce operating system overhead on nodes.

Amazon EKS Anywhere supports the [Kubernetes Cluster Autoscaler](#) for data plane scaling and Customers can use Horizontal Pod Autoscaler to scale workloads in the cluster. The Kubernetes Metrics Server and Kubernetes Cluster Autoscaler can be installed by Amazon EKS Anywhere curated packages.

For cost visibility, customers can use tools such as [Kubecost](#). Regardless of the selected cost management tool, customers should be aware that translation of the collected utilization data to the actual cost is unique to their on-premises environment. When applying a cost management add-on, such as Kubecost, it's important to specify the pricing model that maps Central processing unit (CPU), Graphics processing unit (GPU), Random-access memory (RAM), Egress and other resource utilization metrics to its corresponding price. Here is an [example](#) of such a mapping provided by Kubecost that can be configured through `helm` parameters (see `defaultModelPricing`).

## Performance efficiency

### Autoscaling

Kubernetes supports scaling workloads based on demands. Customers can use [Horizontal Pod Autoscaler](#) (HPA) and [Vertical Pod Autoscaler](#) to autoscale applications





running in Amazon EKS Anywhere clusters. For HPA, customers must install the Kubernetes Metrics Server, which can be installed by Amazon EKS Anywhere curated package.

Data plane scaling mechanisms such as [Cluster Autoscaler](#) can be installed via Amazon EKS Anywhere curated packages; however, [Karpenter](#) (which scales nodes based on resource requirements) isn't supported. For more information on scaling Amazon EKS Anywhere data plane, see [here](#).

## **Persistent storage**

Stateful Kubernetes workloads persist data using external storage platforms. Kubernetes [Container Storage Interface](#) specification abstracts underlying storage subsystem and standardizes usage of persistent storage for workloads.

Customers can choose to use different Container Storage Interface (CSI) depending on their Amazon EKS Anywhere deployment models. Customers running Amazon EKS Anywhere clusters on VMware vSphere can use vSphere storage Container Storage Interface (CSI) plugin for provisioning persistent volumes on vSphere storage. Likewise customers running Amazon EKS Anywhere clusters on Snowball Edge can use Amazon Elastic Block Store ([Amazon EBS](#)) Container Storage Interface (CSI) driver for provisioning persistent volumes on Amazon EBS block storage.

AWS Partners, such as NetApp and Purestorage, also provide CSI drivers for provisioning persistent volumes. Kubernetes documentation includes a [list of CSI drivers](#) compatible with the upstream Kubernetes version.

## **Network connectivity to AWS**

Organizations establish connectivity between their data centers and VPCs when operating in hybrid environments (on-premises and AWS Cloud). [AWS Direct Connect](#) and VPN connections are two ways to provide network connectivity between applications running on Amazon EKS Anywhere and customer VPCs.

AWS recommends provisioning performant connections to provide consistent network experience. The [AWS Hybrid Connectivity whitepaper](#) describes design considerations for selecting hybrid connectivity.

## **Kubernetes control plane and data plane connectivity**

A stable and consistent network connection is required from the Kubernetes data plane to the Kubernetes control plane. The underlying host platform (e.g., vSphere) might provide the option to deploy across multiple physical data centers on the same network



segment. It isn't recommended to deploy the data plane or control plane across different data centers unless there is a flat network and stable connectivity.

When a worker node loses connection to the control plane, existing containers continue running until connectivity is regained. If Kubernetes control plane is unable to connect to a worker node for more than five minutes, then the cluster marks the node as unhealthy. When the node regains connection to the control plane, the cluster evicts all the Pods on the node, which can cause application unavailability. Please see [Kubernetes documentation](#) for more information about node disconnection behavior.

### **Multi-homed Kubernetes pods for high network throughput**

There are applications that require Pods with multiple network interfaces, referred to as multi-homed Pods. These applications typically use specialized hardware such as [Single Root I/O Virtualization \(SR-IOV\)](#) and [Data Plane Development Kit \(DPDK\)](#), which bypass the operating system kernel for increased bandwidth and network performance. Other use cases for multi-homed Pods include:

- Traffic Splitting - Running network functions (NF) that require separation of control/management, and data/user plane network traffic to meet low latency Quality of Service (QoS) requirements.
- Security - Supporting multi-tenant networks with strict traffic isolation requirements. Connecting multiple subnets to Pods to meet compliance requirements.

The Multus CNI plugin, works with Amazon EKS Anywhere on bare metal and allows Pods to have multiple network interfaces, the cluster machines where these Pods run must have multiple network interfaces configured. For more information checkout the Multus CNI configuration Amazon EKS Anywhere [documentation](#).

## **Reliability**

### **Amazon EKS Anywhere control plane**

The following section lists recommendations to improve the reliability of the Amazon EKS Anywhere control plane.

#### **Kubernetes control plane and etcd sizing**

The Kubernetes control plane can be scaled either horizontally (i.e., adding more nodes) or vertically (i.e., provide nodes with more resources). Not all Amazon EKS Anywhere providers support vertical scaling, so review the documentation for more [information](#). For high availability, it's recommended to deploy the Kubernetes control plane with an external etcd (unstacked/external topology).

AWS recommends having a minimum of two Kubernetes control plane nodes to support the Kubernetes API server components for high availability. If etcd is also running on the control plane nodes (see [stacked etcd topology](#)), which is the default Amazon EKS Anywhere configuration, then the minimum recommended number of Kubernetes control plane nodes is three. Consequently, customers should scale the control plane in odd numbers (3,5, ...). See etcd [documentation](#) for more information.

From an experimentation standpoint, single-node Amazon EKS Anywhere cluster can be built using a [custom built binary](#) for Amazon EKS Anywhere. At the time of this whitepaper being published, Amazon EKS Anywhere on bare metal is the only provider supporting [single-node clusters](#).

## **Monitoring Kubernetes control plane**

The Kubernetes control plane comprises components responsible for critical cluster operations. When control plane processes are starved of resources or are unresponsive, critical cluster operations are impaired.

It's a best practice to create production clusters with highly available control plane to eliminate single points of failure. Customers can monitor the health of Kubernetes processes using performance monitoring tools such as Prometheus or AWS Partner solutions.

The Amazon EKS Best Practices guide includes guidance for [monitoring control plane metrics](#). Please see [Troubleshooting Amazon EKS API servers with Prometheus](#) for advanced Kubernetes control plane monitoring and troubleshooting tips.

## **Backing up etcd**

Kubernetes stores all objects and their associated state in etcd. AWS recommends customers create snapshots of etcd periodically for recovery purposes. For more information, see [etcd backup and restore](#) in the Amazon EKS Anywhere documentation

## **Network connectivity to the cloud**

Using Amazon EKS Anywhere in conjunction with services running in AWS Cloud requires a resilient network connection. AWS recommends adding redundancy to network connection to eliminate hardware and network circuits as a single point of failure.

It's a best practice to use dynamically routed, active/active connections for automatic load balancing and failover across redundant network connections. Connections should have capacity to ensure that the failure of one network connection does not overwhelm and degrade redundant connections.

Customers can connect data centers with their Amazon VPC using one or more AWS Direct Connect, AWS Managed Site-to-Site VPN, or Customer-managed Site-to-Site VPN. See the [AWS Hybrid Connectivity](#) whitepaper for more details.

## GitOps

GitOps is a continuous delivery pattern that stores Kubernetes cluster objects in a Git repository. Instead of making imperative changes to a cluster, users create and modify Kubernetes resources by creating and modifying Kubernetes manifests in version-controlled Git repository. A GitOps operator monitors the Git repository continually and applies changes to the cluster automatically. To learn more about GitOps, see [Automating Amazon EKS with GitOps](#).

Amazon EKS Anywhere on VMware bundles the FluxCD GitOps controller. Customers can also use other Kubernetes GitOps tools such as [ArgoCD](#) with Amazon EKS Anywhere. At the time of this whitepaper being published, GitOps enabled Amazon EKS Anywhere cluster is generally available only for Amazon EKS Anywhere on VMWare.

## Applications / data plane

To operate applications customers should design for a highly available and reliable Amazon EKS Anywhere data plane. To run highly available and resilient application and data plane on Amazon EKS Anywhere we recommend the following:

### Spread workloads across multiple data plane nodes

It's a best practice to protect critical workloads by using Kubernetes [Pod Topology Spread Constraints](#) or [Pod affinity / anti-affinity](#) to ensure Kubernetes deployments replicas are spread across data plane nodes and not deployed on a single-node. For more information see the [Kubernetes documentation](#) on Assigning Pods to nodes.

## Load balancers

AWS Application Load Balancers and Network Load Balancers are not available on Amazon EKS Anywhere. A Curated Package is available for MetalLB, an open-source utility that provides a network load-balancer implementation to create load balancers when a user creates a service of type `LoadBalancer`.

For a highly available on-premises load balancer set-up, MetalLB is the recommended solution. Metal LB allocates an IP address to Kubernetes service by advertising via Layer 2 Address Resolution Protocol (ARP) or Border Gateway Protocol (BGP). A limitation of MetalLB ARP deployment configuration is single-node bottlenecking. Only a single leader node is elected for MetalLB which limits Kubernetes service ingress

bandwidth to the single-node. Customers can also consider [kube-vip](#) as alternative to MetalLB.

## **Ingress**

Amazon EKS Anywhere provides Emissary as a Curated Package for ingress. Customers can also use other Kubernetes ingress controller that implements the ingress specification, such as [nginx ingress controller](#). It's a best practice to run multiple replicas of the ingress controller, preferably spread across multiple nodes.

## **Container image registry**

Container images are stored in a container registry. It's important the registry is highly available and fault tolerant to ensure that when Kubernetes Pods scale, or during deployment this component does not become a single point of failure.

Customers can create an on-premises container registry using Harbor using Amazon EKS Anywhere curated packages. Harbor is an open-source trusted cloud native registry project that stores, signs, and scans content. Harbor extends the open-source Docker Distribution by adding the functionalities usually required by users such as security, identity, and management. Having a registry closer to the build and run environment can improve the image transfer efficiency. Harbor supports replication of images between registries, and also offers advanced security features such as user management, access control and activity auditing.

This requires a highly available persistent storage system.

## **Use Amazon EKS Anywhere curated packages**

Customers can use Amazon EKS Anywhere curated packages to install Kubernetes add-ons with more confidence. Curated packages provide trusted, up-to-date, and compatible software that are supported by Amazon, reducing the need for multiple vendor support agreements. An Amazon EKS Anywhere subscription is required to make use of curated packages. See [here](#) for the list of curated packages.

## **Network reliability**

### **Plan for network CIDR growth**

By default, the Cilium CNI bundles with Amazon EKS Anywhere assigns Pod's IP address from the allocated Pod CIDR and Service CIDR on cluster creation. It's recommended to size these CIDR for growth. Only one customer specified CIDR block

specification is permitted for Pod and service CIDR IPs and these CIDR blocks shouldn't conflict with the network subnet range selected for the VMs or hosts.

## CoreDNS

CoreDNS provides name resolution in Kubernetes clusters. It's recommended to consider monitoring CoreDNS latency

(`coredns_dns_request_duration_seconds_sum`), errors

(`coredns_dns_response_rcode_count_total`, NXDOMAIN, SERVFAIL,

FormErr) and CoreDNS Pod's memory consumption. Customers can also improve

Cluster DNS performance by running [NodeLocal DNS Cache](#). [Cluster-proportional auto-scaler](#) for CoreDNS can be installed to automatically horizontally CoreDNS based on the number of CPUs or nodes in the cluster.

## Operational excellence pillar

### Amazon EKS Anywhere release cycle

Starting from Amazon EKS Anywhere version 0.11, Amazon EKS Anywhere supports at least four recent versions of Kubernetes. The end of support date of a Kubernetes version aligns with Amazon EKS in AWS as documented on the [Amazon EKS Kubernetes release calendar](#).

Common vulnerabilities and exposures (CVE) patches and bug fixes, including those for the supported Kubernetes versions, are back-ported to the latest Amazon EKS Anywhere version. For more information on version support see the Amazon EKS Anywhere [documentation](#). Customers can get notified of new releases as documented on the [Release Alerts](#) page.

## Day 2 operations and management

### Monitoring hybrid container workloads

Customers running Amazon EKS Anywhere might use Prometheus for monitoring their Kubernetes cluster and would like to use Grafana as one of the options to visualize the metrics. Prometheus is a popular open-source monitoring tool that provides powerful querying features and has wide support for a variety of workloads. Amazon Managed Service for Prometheus is a fully managed Prometheus-compatible AWS service that makes it easier to monitor environments, such as Amazon EKS, [Amazon Elastic Container Service \(Amazon ECS\)](#), and [Amazon EC2](#), securely and reliably. [Amazon Managed Grafana](#) is a fully managed and secure data visualization service for open-source Grafana that enables customers to instantly query, correlate, and visualize

operational metrics, logs, and traces for their applications from multiple data sources. Amazon Managed Grafana integrates with multiple AWS security services and supports [AWS IAM Identity Center \(Successor to AWS Single Sign-On\)](#) to offer single sign-on for accessing the Grafana console in Amazon Managed Grafana workspace, manage access control, search data, and build visualizations. Reference [this](#) post for monitoring Amazon EKS Anywhere using Amazon Managed Service for Prometheus and Amazon Managed Grafana.

Customers running Amazon EKS Anywhere on air-gapped environments should be running their own version of self-managed monitoring solutions such Prometheus to continuously monitor their Amazon EKS Anywhere clusters.

### Monitor IP address inventory

It's recommended to monitor the available IP addresses left for the Pod, Service CIDR, and host DHCP pool. There needs to be headroom to scale the workload and support in-place rolling cluster deployments, or blue/green cluster upgrades if that is the preferred upgrade mechanism.

The IP address and CIDR range used for the control plane load balancer endpoint and any Kubernetes services of type Load Balancer should be excluded from the Dynamic Host Configuration Protocol (DHCP) pool to avoid IP allocation conflicts.

### Amazon EKS Anywhere Shared Support Model for Support

AWS Supports	Third-party Software support for which AWS will be the point of contact	Customers Responsibility
Kubernetes Distribution (EKS-D)	Cilium and related network policies	Cluster upgrades (operations)
Bottlerocket	Flux Controller	Application and code
Container Runtime		Physical server, storage, and network

AWS Supports	Third-party Software support for which AWS will be the point of contact	Customers Responsibility
<b>Cluster create/update/delete</b>		External DNS, DHCP, and identity systems
<b>Cluster upgrades (tooling)</b>		Third-party tooling other than those bundled with Amazon EKS Anywhere
		Load Balancers (software and hardware)

Enterprise support for Amazon EKS Anywhere is available to Amazon customers who pay for the [Amazon EKS Anywhere Enterprise subscription](#). For customers with Amazon EKS Anywhere Enterprise Subscription, AWS is the single point of contact for:

- Amazon EKS Distro
- Amazon EKS Anywhere CLI
- Bundled components (Bottlerocket host OS, Cilium CNI, GitOps [Flux]) curated packages

The customer is responsible for:

- Applications and code
- Operations (cluster creation, upgrades, maintenance)
- External load balancer
- Third-party tooling other than those bundles with Amazon EKS Anywhere
- External DNS, DHCP and identity systems
- Physical servers, storage, and networking

AWS recommends if customers plan to use an alternative CNI plugin (other than the version bundled with Amazon EKS Anywhere) in production, they should obtain third party support for the plugin, or have in-house expertise to troubleshoot.

Amazon EKS Anywhere has a Support Bundle command built-in for troubleshooting for sharing with AWS support. This allows customers to gather cluster information, save it to an administrative machine, and perform analysis of the results. Amazon EKS



Anywhere uses [troubleshoot.sh](#) to [collect](#) and [analyze](#) Kubernetes cluster logs, cluster resource information, and other relevant debugging information.

Don't add personally identifiable information (PII) or other confidential or sensitive information to a support bundle. Support bundles provided to AWS Support are accessible to other AWS services.

## Handling cluster upgrades

Amazon EKS Anywhere provides upgrade ability and runs a set of preflight checks to check if a cluster is ready to be upgraded. Amazon EKS Anywhere then performs the upgrade, modifying the cluster to match the updated specification. For a list of all the upgradeable attributes, other than Kubernetes version, in the Amazon EKS Anywhere Kubernetes cluster specification file see the Upgrade Cluster Amazon EKS Anywhere [documentation](#). The upgrade command also upgrades core components of Amazon EKS Anywhere and lets the user enjoy the latest features, bug fixes and security patches.

Cluster upgrades aren't handled automatically and require administrator actions to modify the cluster specification and perform an upgrade. It is advised to upgrade clusters in development environments first and verify that workloads and controllers are compatible with the new version.

Cluster upgrades are performed in place using a rolling process (similar to Kubernetes Deployments). Upgrades can only happen one minor version at a time (e.g., 1.20 → 1.21). Control plane components will be upgraded before worker nodes. A new VM is created with the new version and then an old VM is removed. This happens one at a time until all the control plane components have been upgraded.

Amazon EKS Anywhere upgrade process also upgrades the following components:

- Core Cluster API provider (CAPI)
- CAPI providers
- Cilium CNI plugin
- Cert-manager
- Etcdadm CAPI provider
- Amazon EKS Anywhere controllers and CRDs
- GitOps controllers (Flux) - this is an optional component, which will be upgraded only if specified

If customers have specific requirements, then they could also look at implementing an A/B approach using GitOps. Some workloads, for example, batch which typically exist



for a period of time and may be appropriate to delete the existing cluster and create new. When considering cluster upgrades via in-place or A/B consider plan upfront to ensure there is adequate compute capacity.

## Amazon EKS Anywhere partners and integrations

Partners are key to the success of EKS Amazon Anywhere. Customers who consider Amazon EKS Anywhere adoption expect partner and third-party products to be supported on Amazon EKS Anywhere.

The categories of third-party products, components, and integrations on Amazon EKS Anywhere include:

- **Core:** third-party software or tools that are components of Amazon EKS Anywhere.
- **Platform:** hardware and virtualization environments where Amazon EKS Anywhere was validated.
- **Independent software vendors (ISV):** products and solutions (add-ons) from independent software vendors that support Amazon EKS Anywhere.
- **Services:** services offerings that include Amazon EKS Anywhere.

### Core

[Isovalent](#) builds open-source cloud-native networking software that solves networking, security, and observability for modern infrastructure. Amazon EKS Anywhere [chose](#) Cilium for its Container Network Interface (CNI). This CNI add-on features eBPF-powered connectivity, observability, and network security and is installed with Amazon EKS Anywhere automatically. Cilium is also supported as an [alternate compatible CNI plugin](#) on Amazon EKS.

[Weaveworks](#) is the original GitOps company and is a founding member of the Cloud Native Computing Foundation (CNCF). Amazon EKS Anywhere comes pre-installed with FluxCD controller (an open-source project originated and sponsored by Weaveworks), which customers can use to manage cluster operations, add-ons and workloads. Weaveworks is also the author of the `eksctl` tool to create and manage Amazon EKS Clusters. This tool was [extended](#) with Amazon EKS Anywhere support and is now the primary approach for cluster provisioning. In addition to that, Weaveworks validated their [Weave GitOps offering on Amazon EKS Anywhere](#) for enterprise scale GitOps management.

## Platform

At the time of publication, Amazon EKS Anywhere has been validated on Dell (PowerFlex), Equinix, Lenovo (applicable to bare metal and VMWare installations), and Nutanix Cloud Infrastructure. Customers can get up-to-date information about currently supported platform partners on the [Amazon EKS Anywhere partners page](#).

## Independent software vendors (ISV)

Kubernetes provides a great foundation for customer compute and software delivery platform, but it requires third-party software (such as ISV products) to cover all the needs of the enterprise.

Here are some examples of ISV integrations for Amazon EKS Anywhere categorized based on the use case:

- **Backups:** Kasten by Veeam
- **Container security:** Aqua Security, Crowdstrike, Sysdig
- **Cost management:** Kubecost
- **Database:** CockroachLabs (CockroachDB)
- **DevOps:** Armory (Spinnaker), Harness, Nirmata, Rafay, SUSE Rancher
- **Ingress:** F5 (NGINX and BIG-IP with F5 IngressLink), Kong (API Gateway)
- **Monitoring and logging:** Datadog, Dynatrace, New Relic, Splunk
- **Policy engines:** Nirmata
- **Secret management:** Hashicorp Vault
- **Service mesh:** Solo.io, Tetrade, VMWare (TSM)
- **Storage:** NetApp (CSI driver), Purestorage (CSI driver), Nutanix (CSI Driver)

For a complete and current list of independent software vendors supported on Amazon EKS Anywhere please refer to the [Amazon EKS Anywhere Partners page](#).

## Services

Customers adopting Amazon EKS Anywhere may require assistance from experts to address a wide spectrum of requirements from cluster configuration and operations to regulatory compliance. For such cases, AWS Partners have included Amazon EKS Anywhere in their offering portfolio. These partners enabled their solutions architecture and engineering to become experts in solution delivery, whether those are isolated on-premise installation or hybrid solutions that require interactions with AWS services. Some of the services partners are also experts in specific industry domain such as telecommunications, financial services, etc.

For a complete and current list of services partners who are capable of delivering consulting services for Amazon EKS Anywhere, please refer to the [Amazon EKS Anywhere partners page](#).

## Amazon EKS Anywhere AWS support

Amazon EKS Anywhere is an open source project supported by the community. Customers can open an [issue on GitHub](#) for bug fixes and feature requests.

To get support and additional paid features for your Amazon EKS Anywhere clusters, you can purchase an Amazon EKS Anywhere Enterprise Subscription. AWS Enterprise Support or AWS Enterprise On-Ramp Support Plan is a pre-requisite for purchasing an Amazon EKS Anywhere Enterprise Subscription. To learn more about AWS Enterprise Support, click [here](#). To purchase an Amazon EKS Anywhere Enterprise Subscription, click [here](#). To request a free trial, talk to your Amazon representative or connect with one [here](#).

Amazon EKS Anywhere customers facing a potential security issue should notify AWS Security via [vulnerability reporting page](#). Please do not create a public GitHub issue for security problems.

## Conclusion

This whitepaper presented an overview of Amazon EKS Anywhere and deployment options to support Hybrid Container deployments. In addition, it discussed the different use cases and technical design considerations for a well-architected Amazon EKS Anywhere cluster from a security, performance, cost optimization, operational excellence and reliability perspective.

## Contributors

Contributors to this document include:

- Elamaran Shanmugam (Ela), Senior Containers Specialist Solutions Architect, AWS
- Re Alvarez Parmar, Principal Containers Specialist Solutions Architect, AWS
- Robert Northard, Senior Containers Specialist Solutions Architect, AWS
- Mikhail Shapiro, Principal Containers Partner Solutions Architect, AWS

## Further reading

For additional information, refer to:

- [Amazon EKS Best Practices Guide](#)
- [Amazon EKS Anywhere Frequently Asked Questions](#)
- [Amazon EKS Anywhere documentation](#)
- [Amazon EKS Anywhere Blogs](#)
- [Amazon EKS Anywhere GitHub Roadmap](#)
- [Amazon EKS Anywhere release changelog](#)

## Document revisions

Date	Description
March 22, 2023	First publication