AWS Containers: The Basics and How to Secure Containers on Amazon

Learn about Amazon container features, container services including ECS, Fargate and EKS, and how to secure your containers in the AWS cloud

How Can You Deploy Containers on AWS?

AWS provides several services that can help you deploy containerized workloads in the cloud:

- **Elastic Container Service (ECS)**—Amazon's native container management service. You can use it to deploy containerized applications from an on-premises Docker environment.
- **AWS Fargate**—is based on Amazon ECS technology. This service lets you run serverless containers (without managing infrastructure).
- **Elastic Container Service (EKS)**—is Amazon's managed Kubernetes service. You can leverage EKS to easily run Kubernetes clusters on AWS.

In this article, you will learn:

AWS Container Features

AWS provides several options for running containerized workloads. These are some of the key features common to all Amazon container services.

Security

AWS offers 210 security, compliance and governance features. It provides strong security isolation between containers, ensures you are using the latest security updates, and allows you to set granular permissions for each container. AWS provides a shared responsibility model where the container control plane is under the control of AWS, while worker nodes and workloads are the responsibility of the organization.

Reliability

Amazon container services are based on the world's largest public cloud infrastructure, with 77 availability zones (AZ) across 24 regions. All container services (ECS, EKS, Fargate) have service level contracts to ensure availability using Amazon's extensive high availability architecture.

Integrations

DevSecOps

Cloud Native Applications

Serverless Architecture

Container Platforms

Container Platforms

Azure OpenShift

AWS Containers

Container as a Service

Kubernetes in Production

Docker Container

Kubernetes

Containerized Architecture

Vulnerability Management

Cloud Security

AWS Containers Services

Below we briefly review Amazon's key services for containerized workloads.

Amazon Elastic Container Service (ECS)

ECS is a scalable container management system, which is easy to use and highly integrated with other AWS services, but does not support Kubernetes, the de-facto standard for container orchestration.

You can use ECS to easily start, stop, and manage containers in a cluster. ECS lets you configure containers in task definitions, which you can use to run separate tasks or services. You can run services and tasks on the serverless infrastructure supported by AWS Fargate. Alternatively, you can run them on a cluster of Amazon EC2 instances, which you can directly manage.

Amazon ECS lets you start and stop a containerized application with a simple API call. You can identify cluster status from a central service, and if you are familiar with Amazon EC2, leverage all existing EC2 functionality for the instances running your containers.

You can schedule containers' placement in a cluster according to resource and availability requirements, as well as the isolation strategy you chose. With Amazon ECS, you don't need to run your own cluster and configuration management system, and you don't need to worry about scaling your management infrastructure.

ECS is a regional service, which makes it easy to run containers across multiple AZs in the same region. You can create an Amazon ECS cluster in a new or existing VPC. After the cluster is up and running, you can create task definitions to specify which images to run and in which quantities. Container images can be pulled from any container registry, including Amazon Elastic Container Registry (ECR).

Amazon Elastic Kubernetes Service (EKS)

EKS is Amazon's Kubernetes-based container management service. The service simplifies the process of running Kubernetes on AWS, letting you deploy without having to install or manage the architecture or nodes. Kubernetes is an open source system used to automate the management and deployment of containerized applications.

To ensure high availability, EKS runs Kubernetes control planes in multiple availability zones (AZ). It automatically detects and replaces failed instances, and provides automatic versioning and patching.

Amazon EKS uses an updated version of open source Kubernetes, to ensure users can leverage the plugins and tools offered by the Kubernetes community. EKS applications are fully compatible with those running on a standard Kubernetes environment for both on-premise deployments and various clouds. This means you can easily port your existing Kubernetes applications to EKS without changing the code.

AWS Fargate

AWS Fargate lets you deploy containers without managing the underlying servers on Amazon EC2. Fargate works well with ECS and EKS, and does not require configuring or scaling clusters of virtual machines (VMs) to run your containers. This eliminates the need to select instance types, decide on scaling policies, or optimize the distribution of containers across instances.

To use Fargate with ECS, you run a regular ECS task or service, selecting a Fargate launch type or capacity provider. All you need to do is pack the application into a container, specify memory and CPU requirements, set up IAM and network policies, and launch.

To use Fargate with EKS, create a Fargate profile matching your Kubernetes namespace and labels. You'll then need to delete existing pods and re-create them, and they will automatically be scheduled to run on Fargate.

Each Fargate task comes with individual isolation limits. It does not share resources with other tasks, and has its own kernel cores, processor and memory resources, as well as elastic network interfaces. You see a visualization of this principle in the diagram below.

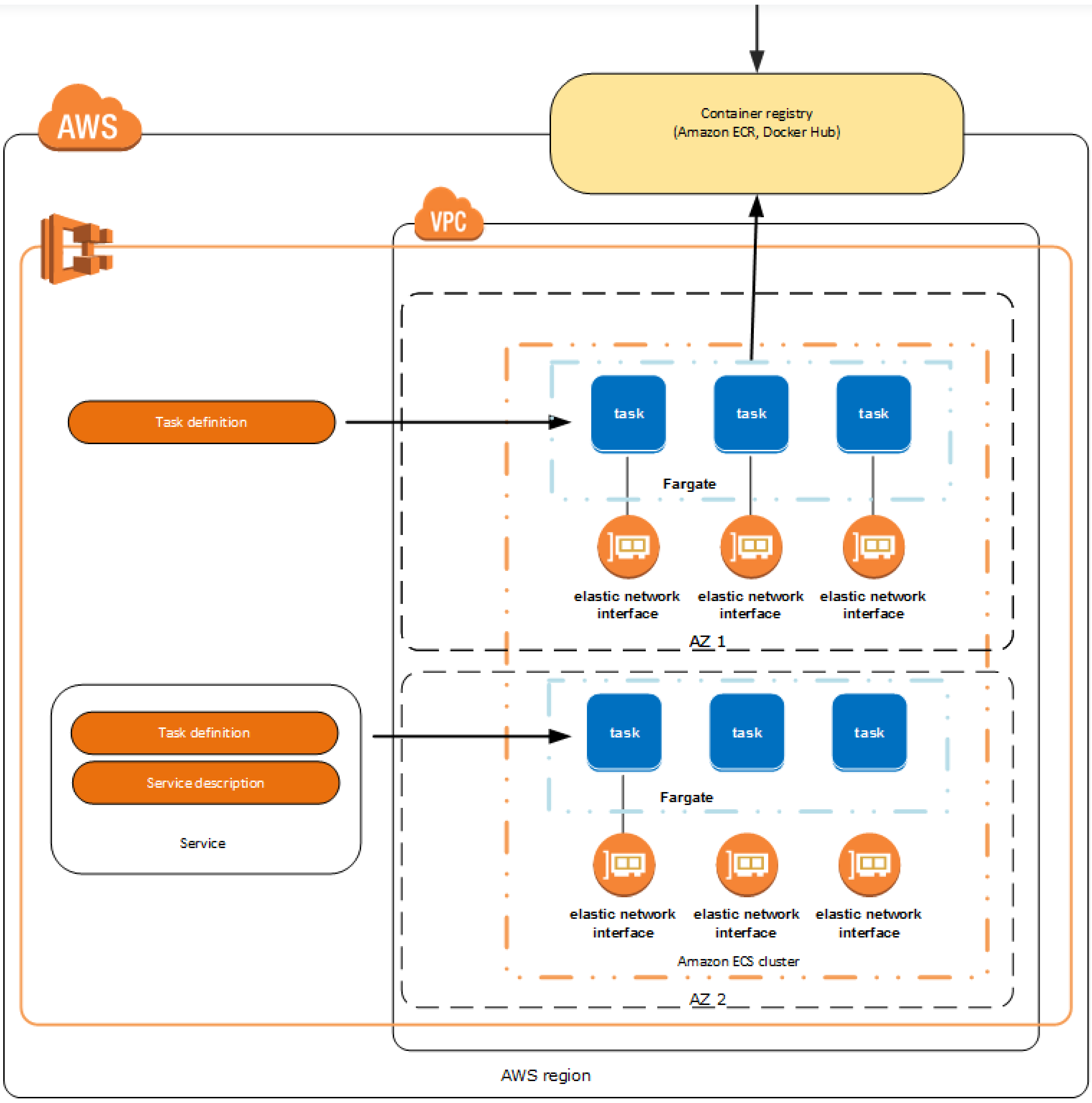

*Image source: **AWS***

In addition to these services, in 2020 Amazon launched support for container images in AWS Lambda. You can now package Lambda functions as container images weighing up to 10 GB, making it possible to run more complex workloads with dependencies using the serverless platform.

Amazon Elastic Container Registry (ECR)

ECR is Amazon's managed service for container image registry. The service provides security, scalability, and reliability. ECR uses resource-based permissions to maintain a dedicated container image repository, managed through AWS IAM.

This allows a single user or Amazon EC2 instance to access the repository and images in accordance with IAM policies. Developers can download, discover, and manage Docker images, Open Container Initiative (OCI) images, and OCI-compliant artifacts using their preferred command line interface.

AWS Container Security

As the development world moves to containers, the security challenges of a cloud native environment become apparent. Amazon provides built in mechanisms you can use to secure cloud-based containers.

AWS ECS Security

Identity and Access Management (IAM)

IAM is key to securing your containers on ECS. There are three main IAM roles used by ECS:

DevSecOps

Cloud Native Applications

Serverless Architecture

Container Platforms

Container Platforms

Azure OpenShift

AWS Containers

Container as a Service

Kubernetes in Production

Docker Container

Kubernetes

Containerized Architecture

Vulnerability Management

Cloud Security

should have full access to Amazon ECS. You can then determine the Amazon ECS resources and roles that your employees need access to.

- **IAM administrator—**the IAM administrator in your organization should be aware of Amazon ECS and learn how to define policies to manage cross-organizational access to ECS resources.

Infrastructure security

ECS is protected by the same security policies and procedures as other Amazon managed services. Any access to ECS is performed using AWS published API calls. The client must support TLS 1.0 or later (Amazon recommends TLS 1.2 or newer), as well as cipher suites with perfect forward secrecy (PFS), such as DHE or ECDHE.

All requests must be signed with the access key identifier and the secret key belonging to an IAM principal. Alternatively, access can be managed using the AWS Security Token Service (STS), creating temporary security credentials to sign a request.

AWS EKS Security

Shared responsibility model

In Amazon EKS, the Kubernetes control plane is under the control and responsibility of AWS. The master node is locked and secured by Amazon, while users are responsible for securing worker nodes and workloads running on them. This is important to understand when planning your EKS security strategy.

Deploy workers onto private subnets

To minimize your attack surface, providing a subnet dedicated to Kubernetes nodes. The subnet controls the assignment of public IP addresses to hosts. It is preferred to block public access to an EKS subnet. However, if you need to assign a public IP, apply AWS Security Group restrictions to limit access and visibility.

Avoiding privilege escalation

In the context of EKS, privilege escalation is a way for one user to run a file with the privileges of another user or group. The Linux sudo command is a good example of this. You can avoid privilege escalation by implementing a pod security policy, setting `allowPriviledEscalation` to false or adding `securityContext.allowPrivilegedEscalation` to the `podSpec` .

Use a third party solution for runtime protection

While Linux provides the Seccomp and Apparmor configuration files, these can be difficult to create and manage. If you don't have Linux security experts on your team, a good idea is to use a commercial solution for runtime protection of containers on EKS. Commercial solutions help you enforce security standards by default, and typically use machine learning to detect and block suspicious or anomalous activity.

EKS Distro

Amazon released a special Kubernetes distribution called EKS Distro, which allows users to deploy the same version of Kubernetes that is used to run Amazon EKS. Essentially, it's an open source distribution of Kubernetes, fully compatible with EKS, that you can download for free and deploy in your local data center. Using EKS Distro means securing it is entirely the customer's responsibility.

AWS ECR Security

Encryption at Rest

Amazon ECR stores images in special Amazon S3 buckets. By default, Amazon ECR uses server-side encryption and an encryption key managed by Amazon S3, which automatically encrypts your data with AES-256 encryption. Additionally, you can use server-side encryption with the customer master keys (CMKs) that are stored in the AWS Key Management Service (AWS KMS).

Monitoring

activity.

Additional AWS Container Materials

AWS Bottlerocket—Linux-Based OS Purpose-Built for Containers

AWS Bottlerocket is an open-source operating system (OS) based on Linux, specifically designed by Amazon Web Services to run containers on VMs or bare metal hosts. BottleRocket only contains the essential software required to run the container, increasing resource utilization and reducing the attack surface.

Today, most AWS users run containerized applications on general-purpose operating systems updated with various software packages, making it difficult to update the operating system automatically.

Bottlerocket is updated in one step, with no need to manage individual packages. This one-step upgrade process makes it easy to automate operating system updates using container orchestration services, like Amazon EKS and ECS. This can dramatically reduce administrative overhead. Additionally, one-step updates improves availability by reducing update errors and enabling rollback of updates.

Bottlerocket is available for free on the Amazon Elastic Compute Cloud (EC2) in the form of an Amazon Machine Image (AMI).

Container Security Tools On the AWS Marketplace

Following are three popular container tools available on the AWS marketplace. See all container services available on the marketplace.

Datadog Enterprise Container Agent

Datadog is a SaaS monitoring platform that lets you inspect the entire application stack, and provides customizable alerts and views. It integrates with over 250 systems, provides end-to-end tracing of logs, servers, containers, databases, and applications, and supports predictive analytics and anomaly detection.

Datadog fully supports containerized workloads on AWS, letting you monitor and alert on individual containers, Kubernetes clusters, and the Kubernetes control plane.

Bitnami Prometheus Container Solution

Bitnami provides Prometheus as a pre-packaged, tested container solution on AWS. Prometheus is an open source monitoring and warning system, supported by the CNCF alongside Kubernetes. It allows administrators to monitor container infrastructure by collecting metrics at specific intervals from designated targets.

Prometheus lets you monitor four types of metrics:

- Counters—cumulative metrics
- Gauges—point-time time metrics
- Histogram—categorizes observations into groups
- Summary—samples data and provides an overview with statistical significance

AWS Container Security with Aqua

Aqua provides the most complete security across the application lifecycle, from development to production, protecting all cloud native applications running on AWS including, Amazon ECS for contair orchestration, Amazon EKS for Kubernetes-based deployments, AWS Fargate for on-demand contain

DevSecOps
Cloud Native Applications
Serverless Architecture
Container Platforms
Container Platforms
Azure OpenShift
AWS Containers
Container as a Service
Kubernetes in Production
Docker Container
Kubernetes
Containerized Architecture
Vulnerability Management
Cloud Security

- **Image vulnerability scanning & assurance**
  Preventing unauthorized images from running in the AWS environment Aqua Continuously scan images stored in Amazon ECR to ensure that no vulnerabilities, bad configurations, or secrets are introduced into container images.
- **Protecting workloads running on Amazon EKS**
  Prevent unvetted containers from running on Amazon ECS, EKS and Fargate environments. Automatically create security policies based on container behavior and ensure that containers only do what they are supposed to do in the application context. Detect and prevent activities that violate policy, and defend against container-specific attack vectors.
- **Securing applications on AWS Fargate**
  Aqua embeds the MicroEnforcer into your containers to ensure that workloads are only performing their intended function, while detecting vulnerable or compromised containers.
- **Protecting AWS Lambda Functions**
  Discovering over-provisioned permissions and roles, vulnerabilities, and embedded credentials and keys. Monitoring functions at runtime, preventing code injection and malicious activity.
- **Cloud VM Security and Compliance**
  Protect workloads running on Amazon EC2 instances and ensure they are properly hardened. Scan for vulnerabilities and malware, apply File Integrity Monitoring (FIM), check configuration against the CIS Benchmark for Linux, and monitor user access and activity. Create command-level audit trail for compliance and forensics.

The full-featured Aqua platform is available for on-demand consumption on the AWS Marketplace **Get Aqua from the AWS Marketplace ›**

Learn more about Aqua for AWS security ›

Subscribe to updates

Your email*

douglas@data-defense.net

Subscribe

DevSecOps

Cloud Native Applications

Serverless Architecture

Container Platforms

Container Platforms

Azure OpenShift

AWS Containers

Container as a Service

Kubernetes in Production

Docker Container

Kubernetes

Containerized Architecture

Vulnerability Management

Cloud Security

Aqua Blog All about cloud native and security

DevSecOps

Cloud Native Applications

Serverless Architecture

Container Platforms

Container Platforms

Azure OpenShift

AWS Containers

Container as a Service

Kubernetes in Production

Docker Container

Kubernetes

Containerized Architecture

Vulnerability Management

Cloud Security