



Modernize today with containers on AWS

Companies worldwide are undergoing digital transformations. By modernizing their applications, they can deliver better service to customers and keep pace in a competitive landscape. One way AWS helps companies modernize is by implementing containers and initiating cultural shifts to streamline development. In this eBook, we discuss best practices in containerization and how you can get started today with containers on AWS.

It's about survival

Over the past few years, worldwide digital transformation has caught fire. Companies of all sizes are finding new ways to leverage technology to boost their agility and better respond to demands from customers. Fueling this fire is the need to survive in a changing environment. These days, it's digital or bust—you don't have to look too far to find examples of cloud-native companies disrupting industries while leaving legacy businesses behind in the dust. For many companies, an initial step toward digital transformation is modernizing their applications and taking advantage of automated environments in the cloud. Modernization empowers companies with:

Elasticity: The ability to respond to spikes in customer demand

Availability: The ability to serve customers' requests wherever and whenever

Agility: The ability to quickly fix a problem or deploy new functionality that customers want



Containers can help get you there

A digital transformation takes time, but in the end, the gains in productivity make the business case. While there are many tools out there to help companies modernize, containers continue to gain momentum as the go-to solution for developers to more efficiently package and deploy applications. According to the Cloud Native Computing Foundation (CNCF) Survey 2020, the use of containers in production has increased by 300 percent since 2016. This is evident from the adoption of two different container technologies broadly used today—Docker and Kubernetes. Of the cloud-hosted Kubernetes workloads, 82 percent run on AWS.¹

Containers streamline development

Containers provide a portable, consistent, and lightweight software environment for applications to easily run and scale anywhere. Throughout its lifecycle, an application will operate in many different environments, whether it's moving from test and production early on or from on-premises virtual machines to the cloud during a migration. Before containers, IT teams had to consider the compatibility

restrictions of each new environment and write additional code to ensure the application would function. Containers were developed to package the application with its dependencies, configuration files, and interfaces, allowing developers to use a single image that moved seamlessly between different hosts, thus removing the undifferentiated heavy lifting of infrastructure management from development teams.

For developers, containers allow them to focus on building the application—like adding new features or the latest security—instead of spending time managing the compatibility requirements of different environments.

Containers are also integral to breaking down traditional monolithic application architectures, and they enable a transition to microservices for easier scale. With microservices, each application component runs as its own service, allowing developers to work independently on different aspects.



¹ Nuclear Research report 2019

Containers drive sustainable culture shifts in development practices

Containers are not only a tool to modernize your applications, but they're also the catalyst for improvements to your development practices. Containers disrupt the traditional development cycle by driving developers to assume ownership of quality control for the apps and code developed. Developers used to only focus on building the applications, and they had little oversight on the success of packaging and deployments. With containers, developers experience the phenomenon known as "shift," which ensures they handle quality control upfront rather than further down in the development process.

With developers empowered to own the implications of their work, the next step is to create a culture that fosters failing fast and learning from mistakes. Containerization enables this by automating code integration/deployment, thus improving the company's overall agility.

Legacy companies can modernize

A large, 100-year-old corporation seized an opportunity to modernize its applications in one of its international markets. By implementing this cloud-based approach, the company's development team achieved greater agility and could iterate much quicker. Utilizing containers and an AWS-automated environment, they pushed their new platform from concept to prototype to deployment in just six weeks.



Get started with containers on AWS today

AWS provides all the solutions needed to make containerization seamless. Proven tools for infrastructure provisioning, orchestration, security, networking, automation, and monitoring are readily available today for you to get started with containers.

Provisioning

Seamlessly provision the underlying infrastructure and resources

To run containers, the underlying infrastructure needs to be provisioned. AWS offers two different solutions based on the extent of management or automation desired:

- Use AWS Fargate to automate the provisioning of the underlying infrastructure
- Use Amazon Elastic Compute Cloud (Amazon EC2) instances to manually define compute, storage, and network capabilities of the infrastructure

Orchestration

Scale and manage Docker or Kubernetes containers

- Leverage Amazon Elastic Container Registry (Amazon ECR) to store and manage Docker images with Amazon Elastic Container Service (Amazon ECS) as your orchestrator
- Adopt Amazon Elastic Kubernetes Service (Amazon EKS) to orchestrate your Kubernetes containers

Security

Secure, scan, and detect vulnerabilities in containers

- AWS Identity and Access Management (IAM) and tagging, security groups for Amazon EC2 instances, and Amazon Virtual Private Cloud (VPC) enable securing your containers
- Image scanning solutions detect vulnerabilities of Docker container images

Networking and connectivity

Distribute application traffic across containers

- Distribute application traffic across containers and serverless environments with AWS Elastic Load Balancing
- Route traffic for globally distributed applications running on containers with AWS Global Accelerator and AWS Elastic Load Balancing
- Improve application performance with AWS Global Accelerator and AWS Elastic Load Balancing
- Manage service-to-service communication and security with AWS App Mesh

Automation

Deploy code automatically with CI/CD

- Create a source code repository using AWS CodeCommit
- Configure a CI/CD pipeline using AWS CodePipeline
- Deploy AWS CodeBuild to build your container image
- Build, deploy, and run containerized web applications with AWS App Runner

Observation and monitoring

Ensure services running on containers are healthy and communicate with each other as expected

- Deploy AWS App Mesh to provide visibility into logging, metrics, and tracing, as well as to enable load balancing and traffic shaping
- Run health check of Docker container images to confirm your containers are running and your app is working
- Use Amazon CloudWatch Application Insights to monitor health and wellness of applications running in containers deployed in Amazon ECS, Amazon EKS, or Kubernetes on Amazon EC2



Provisioning

AWS gives you options to provision your infrastructure with Amazon EC2 and AWS Fargate. The main difference between the two is the amount of management and control you want over the underlying infrastructure that runs your container applications.

- ✓ **AWS Fargate:** With Fargate, you can run your containers without having to manage servers or clusters. All you have to do is package your application in containers, specify the CPU and memory requirements, define networking and IAM policies, and launch the application.
- ✓ **Amazon EC2:** The more traditional Amazon EC2 launch type allows you to bring instances to run your containers and provides server-level, granular control over the infrastructure that runs your container applications.



Orchestration

Once your application has been containerized, the next step is to run the containers in production. To scale your architecture, you'll want an orchestration tool. AWS offers orchestration platforms to fit your needs, whether on-premises or in the cloud.

Amazon ECS and Amazon ECS Anywhere:

Amazon Elastic Container Service (Amazon ECS) allows you to easily deploy containerized workloads on AWS. The powerful simplicity of Amazon ECS enables you to grow from a single Docker container to managing your entire enterprise application portfolio. Run and scale your container workloads across availability zones, in the cloud, and on-premises, without the complexity of managing a control plane or nodes. Amazon ECS provides an infinitely scalable control plane that's managed for you. This kind of orchestration tool works well for companies that use a proprietary operating system or want the ability to control their own infrastructure. AWS Copilot is a command line interface (CLI) that enables customers to quickly launch and easily manage containerized applications on AWS. AWS Copilot provides a simple declarative set of commands, including examples and guided experiences built in to help customers deploy quickly. All you need to spin up production-ready services is AWS Copilot, an AWS account, and your code.

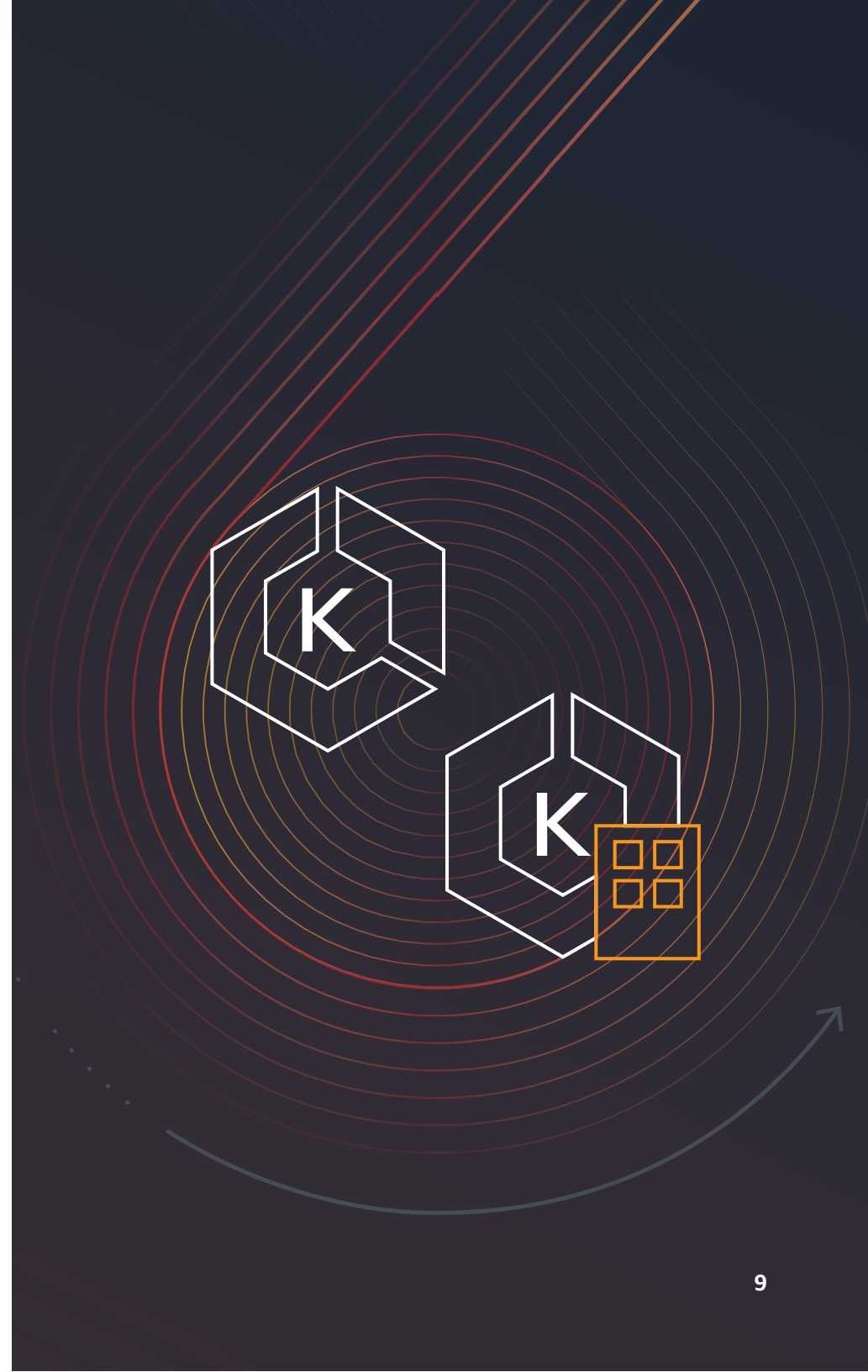
With **Amazon ECS Anywhere**, you can use the same familiar Amazon ECS console and operator tools to manage your on-premises container workloads for a consistent experience across your container-based applications. The AWS Systems Manager (formerly known as SSM) integration automatically and securely establishes trust between your on-premises hardware and the AWS control plane.

Amazon EKS and Amazon EKS Anywhere:

Kubernetes is a rapidly growing open-source container management platform built to help you run your containers at scale. With Amazon EKS, the Kubernetes management infrastructure is run for you across multiple AWS availability zones.

Our customers believe there are tremendous advantages to running Kubernetes on AWS. Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane nodes that are responsible for scheduling containers, managing the availability of applications, storing cluster data, and other key tasks. You can run your Kubernetes applications on both Amazon EC2 or AWS Fargate, which provide serverless compute for containers.

Amazon EKS Anywhere enables you to easily create and operate Kubernetes clusters (building with the software in Amazon EKS Distro) on-premises, including on your own virtual machines (VMs) and bare-metal servers. EKS Anywhere saves you the complexity of building and supporting your own tooling to manage Kubernetes clusters. EKS Anywhere provides automation tooling that simplifies cluster creation, administration, and operations on infrastructure such as bare metal, vSphere, and cloud virtual machines with default configurations for logging, monitoring, networking, and storage but brings opinionated tooling and additional components you would need to run Kubernetes in production, such as cluster installation and lifecycle management, observability, cluster backup, and policy management. Amazon EKS Distro packages up the same open-source Kubernetes software distribution used in EKS on AWS for use on your own infrastructure on-premises. EKS Distro clusters can be managed with your own tooling or with Amazon EKS Anywhere.



Amazon ECR:

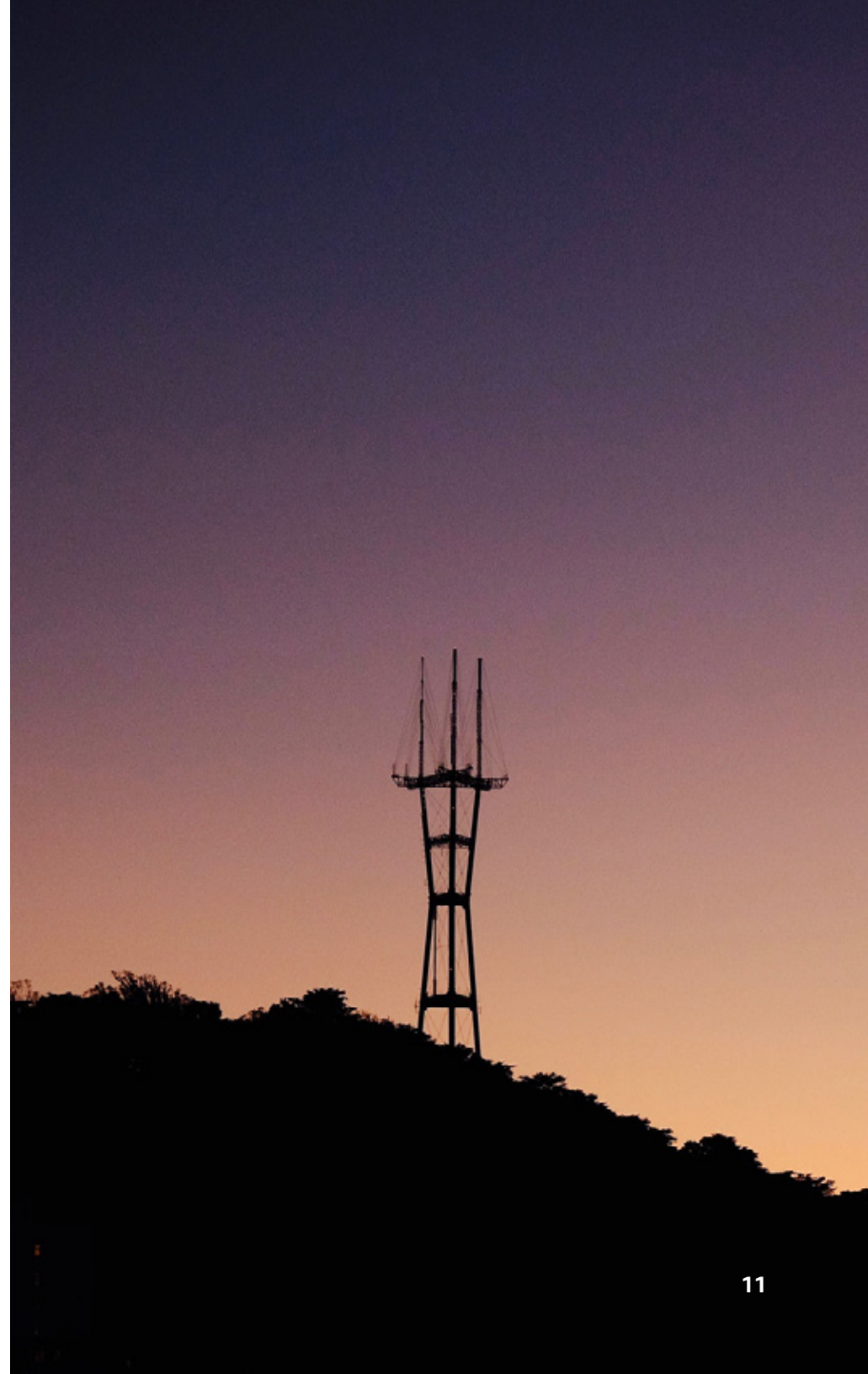
To use Docker containers, the first thing you need is a Docker image. This image acts as a blueprint for creating instances of containers. Amazon ECR is a fully managed Docker container registry that makes it easy for you to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon ECS, simplifying your development-to-production workflow. Amazon ECR eliminates the need to operate your own container repositories or worry about scaling the underlying infrastructure.

Security

Like quality control, security considerations have also shifted into the earlier stages of the development cycle. With more autonomy, developers can more readily adapt their code to address the latest security threats. However, security must be a priority across the organization. One way to support this cultural change is to make security efforts as transparent as possible and define an architecture upfront that takes security best practices and tools into account.

AWS provides multiple tools to control who has access to your containers. You can use AWS IAM to determine who is authenticated (signed in) and authorized (has permission) to use resources. Amazon VPC lets you logically isolate container tasks (Docker) or pods (Kubernetes) in a virtual network that you define. You can define security groups to create a virtual firewall between EC2 instances.

Using an image scanning solution, you can detect vulnerabilities of container images or image dependencies. Security teams can perform scans on these containers or image dependencies and then publish pre-approved resources that developers can consume with confidence.



Networking and connectivity

Once your application is running, you want to ensure that traffic is always distributed across your containers, enabling your end users to use your application uninterrupted.

- ✓ **Elastic Load Balancing** helps your users easily access your applications, using sophisticated load balancing algorithms that are natively integrated with AWS container services.
- ✓ **Amazon Global Accelerator** also helps ensure that your application is highly performant and can be accessed by users around the world, and served from AWS Regions closest to them.
- ✓ To manage communication between container services, **AWS App Mesh** enables fine grained security and deep visibility.



Automation

From now through 2024, there will be an increased demand for hybrid integration capabilities that will go hand in hand with increased consolidation of the competitive landscape of most forms of PaaS offerings.² With Amazon EKS Anywhere and Amazon ECS Anywhere, satisfy compliance, data gravity, and other business requirements by running your workloads on infrastructure you own while enjoying simple and familiar tooling. Rather than installing and operating a local control plane, you can use the same hyperscale, trusted, and fully managed control plane for your on-premises container workloads. Run containerized data processing workloads at edge locations on your own hardware so that you can stay close to your end customers and maintain reduced latency.

The automated environment removes the task of deploying code manually. When your infrastructure includes hundreds or thousands of containers, automating with continuous integration and continuous delivery (CI/CD) allows you to scale and reach faster while minimizing the risk of human error. AWS CodeCommit, AWS CodePipeline, and AWS CodeBuild enable you to create a source code repository, configure a CI/CD pipeline, and build your container image.

AWS App Runner is a fully managed application service that makes it easy for your developers to deploy and run containerized web applications and APIs at scale with just a few clicks. App Runner eliminates the need to configure and manage infrastructure. You can simply provide your source code, container image, or deployment pipeline, and App Runner builds and deploys the web application, load balances network traffic, scales capacity up or down based on demand, monitors application health, and encrypts traffic by default. You also don't need prior containers experience to use App Runner and take advantage of the portability, efficiency, and cost savings of containers.

² Gartner Forecast analysis, Feb 2021



Observability and monitoring

Once the microservices are deployed using containers, you'll need to monitor the health of the containers and ensure that the services are communicating with each other as expected.

- ✓ **AWS App Mesh** makes it easy to run services by providing consistent visibility and network traffic controls for services built across multiple types of compute infrastructure. App Mesh removes the need to update application code to change how monitoring data is collected or traffic is routed between services. App Mesh configures each service to export monitoring data and implements consistent communications control logic across your application. This makes it easy to quickly pinpoint the exact location of errors and automatically reroute network traffic when there are failures or when code changes need to be deployed. App Mesh uses the open-source Envoy Proxy, making it compatible with a wide range of AWS partners and open-source tools.
- ✓ Docker container images allow you to verify if the container and application are healthy. With a simple health check command, a Docker file will check a container to confirm that it's still working. This process can detect when a web server is stuck in an infinite loop and unable to handle new connections, even though the server process is still running.
- ✓ **Amazon CloudWatch Application Insights** supports container monitoring: You can now easily set up monitoring, alarms, and dashboards for your applications deployed in **Amazon ECS**, **Amazon EKS**, and **Kubernetes on EC2** containers running on AWS. Monitoring tier options are now available for capturing the metrics, telemetry, and logs for monitoring the health and wellness of applications running in containers on AWS.



Ready to get the most out of containers?

AWS and our vast partner ecosystem will provide you with the tools you need to get started—no matter what stage of your modernization journey you're in.

You can always reach out to an AWS sales associate or talk with your preferred APN Partner »

To learn more, visit:
aws.amazon.com/containers