

Serverless Architecture

>Overview

What does serverless mean?

- Serverless doesn't mean we've figured out how to run code without computers.

This is simply to write the code and bring the code. This is the natural extension of the cloud. Everything else is handled by AWS.

Serverless means you don't worry about provisioning, configuring, or scaling servers.

Going to serverless is simple to do. Below are the benefits

- Ease of use
 - There isn't much for us to do besides bringing our code. AWS handles almost everything for us.
- Event based
 - Serverless compute resources can be brought online in response to an event happening.
- Billing Model
 - "Pay as you go" in its purest form. You only pay for your provisioned resources and the length of runtime.

On the test, we want to focus on more managed tools. We want to get away from EC2 as much as possible. The more we can offload to AWS the better. Things like Lambda, Fargate, containers over the traditional EC2 architecture.

> Lambda

Serverless services like AWS Lambda allow you access to AWS compute power for up to 15 minutes for a single function. This lets you operate code in response to real-time event triggers. Lambda lets you run code in response to network-based triggers without managing servers like EC2 and having to worry about provisioning any underlying server infrastructure.

Lambda functions run only when triggered by a preset event. It could be a call from your mobile application, a change to a separate AWS resources (like an S3 bucket), or a log based alert.

If an hour or a week passes without a trigger, Lambda won't launch a function (and you won't be billed anything). If there are a thousand concurrent executions, Lambda will scale automatically to meet the demand. Lambda functions are short-lived: they'll time out after 15 minutes.

Roles should be used for Lambda to talk to other AWS APIs

Deploying to Lambda

Lambda is Amazon's serverless computing platform that runs functions that can be written in a variety of languages.

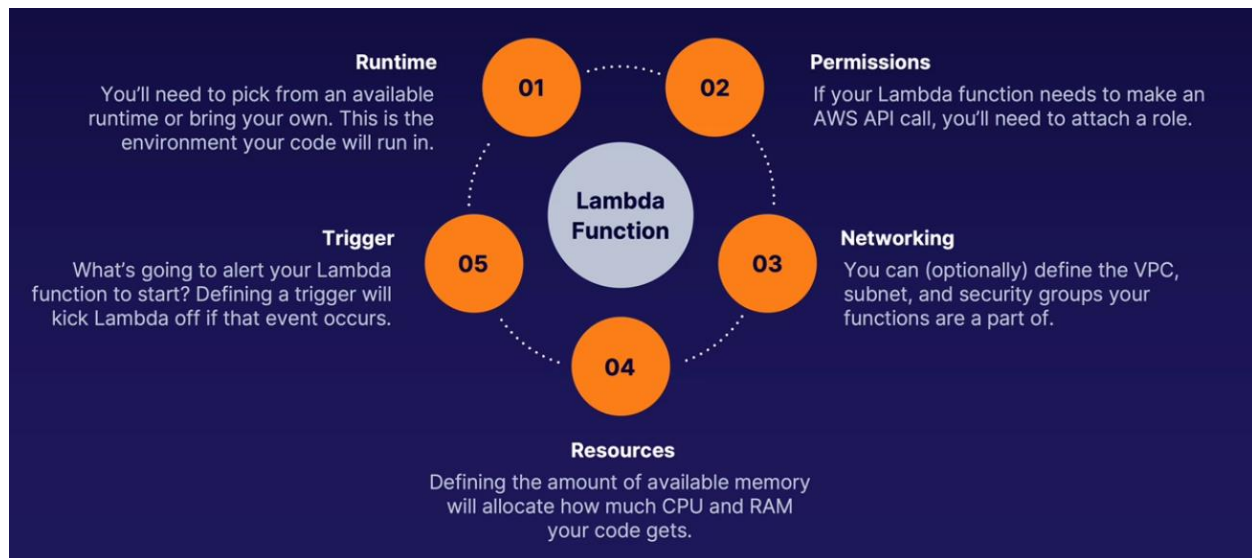
Because you don't have to even think about servers when using Lambda, deploying a new Lambda application with CodeDeploy simply involves creating a new Lambda function. If you need to update an existing function, CodeDeploy just creates a new version of that function. You can then

choose how CodeDeploy handles the switchover to the new version. You can have CodeDeploy shift traffic slowly from one version to the other, or you can do an immediate, full cutover to the new version.

- Lambda allows developers to focus on core business logic for the apps they are developing instead of worrying about managing servers.
- Supports programming languages like Java, Go, Powershell, Node.js, Python, and ruby.

If your Lambda function makes an API call, you need to attach a rule.

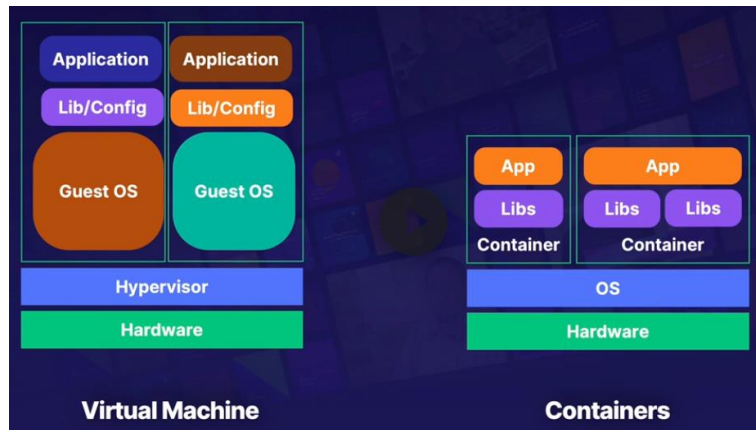
With Lambda, of course you can make API calls. But we can build in our own features and make our own remediation for problems or troubleshooting issues that pop up using Lambda functions.



>Container Overview

What is a container... Well you may think a container stores thing. In the IT world it is the same idea applies, which means you can put everything into. We can bundle up config files, apps, code, etc into the container. You can then move the container anywhere you want. Basically one computing environment to another, the contents doesn't change between environments.

Containers vs VM



Terminology

- Dockerfile
 - Amazon ECS uses Docker images in task definitions to launch containers. Docker is a technology that provides the tools for you to build, run, test, and deploy distributed applications in containers.
 - The image contains the code, libraries, dependencies, and config files, which it can be stored in a registry, similar to GitHub but for images.
 - Using Linux like commands that define what our containers include.

From the general flow you run a Docker file > From that docker file you build an image > From that image you upload it to a registry > And then from the registry you download the image and run it in a container

>Running containers in ECS or EKS

When running into some problems with containers. Having 1 container is easy to manage. But having 10, 100 or even 1,000 containers, that is where ECS comes in play.

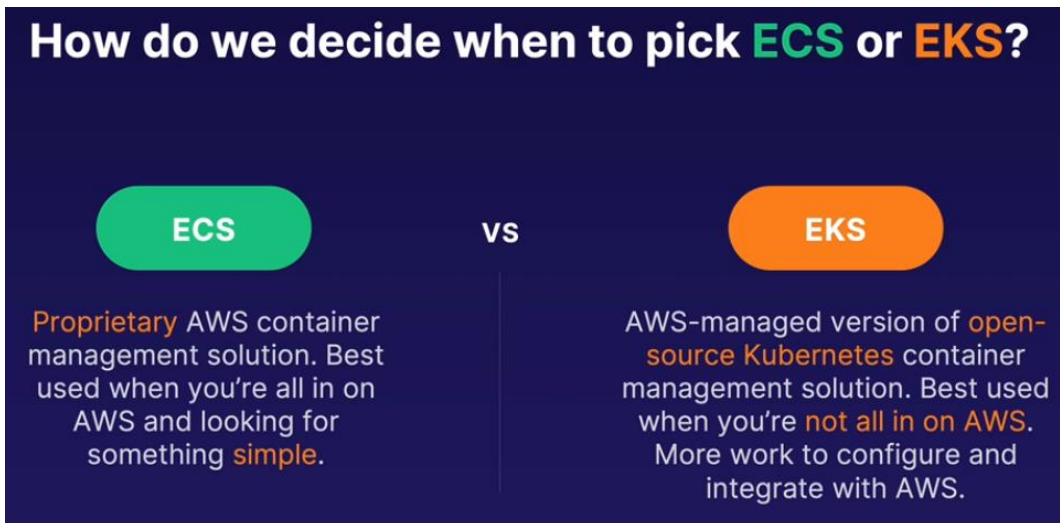
Amazon ECS is a fully managed container orchestration service that makes it easy for you to deploy, manage, and scale containerized applications.

Elastic Load Balancers (ELB) and roles are intergraded into containers that are appropriately registered with the load balancers as they come online and go offline.

Kubernetes

- Is an open-source container management and orchestration platform
 - Originally built by Google, open source for a while now

This isn't the easiest platform to get going with. There is a lot of lifting that needs to get done before we can start running our first container. But that is where EKS comes in play. Which is Elastic Kubernetes Service.



>Removing Servers with Fargate

- Fargate
 - Fargate is a serverless compute engine for containers
 - Fargate allows you to manage containers, like Docker.
 - Scales automatically
 - This is for ECS or EKS... lets clear this up
 - If I am trying to host traditional applications and want full access
 - For long running containers
 - Responsible for underlying OS
 - Use EC2
 - If I want to run short running functions, service-oriented applications and event driven apps, and don't manage servers
 - Look into AWS Lambda
 - If you want to run docker, you gotta choice either ECS or EKS
 - No OS access
 - Only pay for the exact amount of compute time such as CPU and memory
 - You would use fargate
- Fargate allows us to chuck the EC2 instances (Such as OS and hardware) and run the containers in the ether. With Fargate you no longer have to provision and scale cluster or patch/update each sever to meet the applications needs.
- When running task on Fargate, you no longer need to think about the infrastructure as this is operated, owned and controlled by AWS.
- All task launched within Fargate run inside the VPC (Of course as the user you select which VPC/subnets/security groups)
- Fargate supports ALB and NLB, which ELB is not supported with Fargate
- Security with Fargate
 - Fargate service owns infrastructure
 - Customer owns and manages their task
 - No SSH access to the infra
 - Cluster-level isolation for containers



>Amazon EventBridge (CloudWatch Events)

EventBridge (Formerly known as CloudWatch Events)

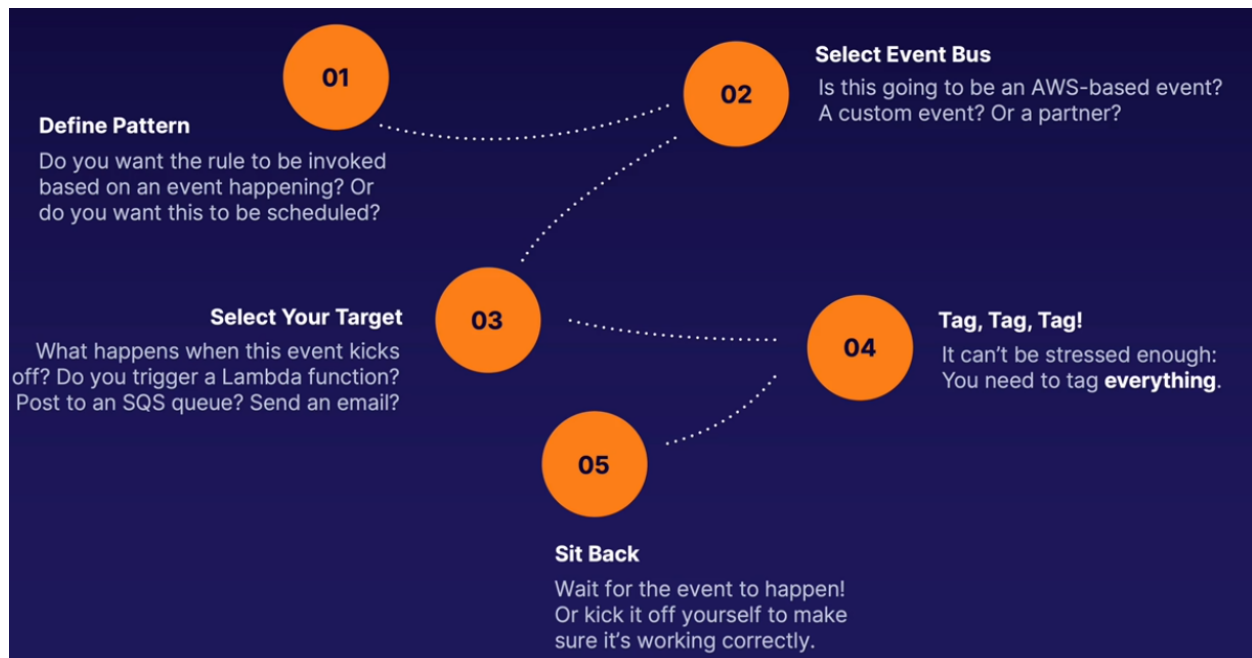
- Cloud watch
 - Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization.
 - CloudWatch Alarms
 - A CloudWatch alarm watches over the value of a single metric.
 - CloudWatch Logs
 - CloudWatch Logs collects and stores log files from AWS and non-AWS sources and makes it easy to view, search, and extract custom metrics from them.
 - CloudWatch Metrics
 - CloudWatch Metrics is a feature that collects numeric performance metrics from both AWS and non-AWS resources such as on-premises servers. A metric is a variable that contains a time ordered set of data points.
 - CloudWatch Events
 - The CloudWatch Events feature lets you continuously monitor for specific events that represent a change in your AWS resources—particularly write-only API operations—and take an action when they occur.
 - CloudWatch responds to events as they occur, in real time. Unlike CloudWatch alarms, which take action when a metric crosses and remains crossing a numeric threshold, CloudWatch events trigger immediately. For example, you can create a CloudWatch event to send an SNS notification whenever an EC2 instance terminates. Or you could trigger a Lambda function to process an image file as soon as it hits an S3 bucket.
- CloudTrail
 - CloudTrail keeps detailed event logs of every action that occurs against your AWS resources. Each event that CloudTrail logs includes the following parameters:

- AWS CloudTrail monitors and records account activity across your AWS infrastructure, giving you control over storage, analysis, and remediation actions.
- Protect your organization from penalties using CloudTrail logs to prove compliance with regulations such as SOC, PCI, and HIPAA.

CloudWatch - Don't forget you can use CloudWatch to monitor your EC2 instances and notify you when certain events occur.

Cloud Trail - Don't forget the things you can track with CloudTrail: username, event time and name, IP address, access key, Region, and error code.

Creating a rule



Exam tips

1. Lambda

- a. If there are questions about how to we add features or enforce things inside of AWS... Think of Lambda
- b. If you want an instance to automatically conforming to the rules that are set, Lambda can kick that off.
- c. Limitations
 - i. 10 gigs of RAM and 15 mins is the function can run
 - ii. You would want to use this for lightweight, temp code
- d. It starts from either S3, CloudWatch events, or EventBridge, ALBs, these support kicking off Lambda functions
- e. Microservices
 - i. Lambda excels in running small and lightweight functions
- f. Networking
 - i. Lambda can run inside or outside a VPC, which they don't have to run inside this networking space like EC2 does.

2. Containers

- a. Containers are generally seen as more flexible on the exam
- b. You can run on-site and move around to different environments
- c. Why containerizes applications?
 - i. For portability, ease of use, flexibility, run on prem or on AWS
- d. Dev vs prod.
 - i. Dev is prod, and prod is dev
 - ii. This is good, because the testing that is being conducted in dev, is going to carry through to production,
 - 1. That way you don't have to worry about things disappearing between different environments
- e. Open source = Kubernetes. Any questions about container management solution that run in AWS and on prem... think EKS
- f. Favor containers rather than EC2 on the exam

3. ECS/EKS

- a. ECS is going to be the preferred method when it comes about containers
- b. Anything dealing with on-prem or open source solution think Kubernetes or EKS
- c. ECS or EKS handles the management, placement, running of our containers
- d. Length
 - i. Great for one-off or long running applications

4. Fargate/Lambda

- a. Lambda = Lightweight functions (Run quick, and easily integrated)
- b. Fargate = Containers that don't need to run all the time
- c. EC2 = Needs to run all the time
- d. Fargate is a serverless tool and doesn't work by itself. ECS or EKS is a requirement
- e. Use EC2 when it comes about cost since Fargate is more expensive
- f. Fargate runs great for containers

- g. Lambda loves roles. Whenever you're talking about credentials and Lambda, ensure you're attaching a role to the function
 - h. Things like S3, API call, Kinesis and EventBridge are common triggers for Lambda
 - i. Lambda can allocate up to 10 GB of RAM and 15 mins of runtime
5. EventBridge
- a. This kicks off the serverless architecture
 - b. Any API calls that happens, can kick off EventBridge/CloudWatch events (Triggering Lambda functions)
 - c. EventBridge is the new name but might see it as CloudWatch on the exam
 - d. Fastest way to respond to an API call
 - e.