

Compute Fundamentals for AWS

HELP US IMPROVE

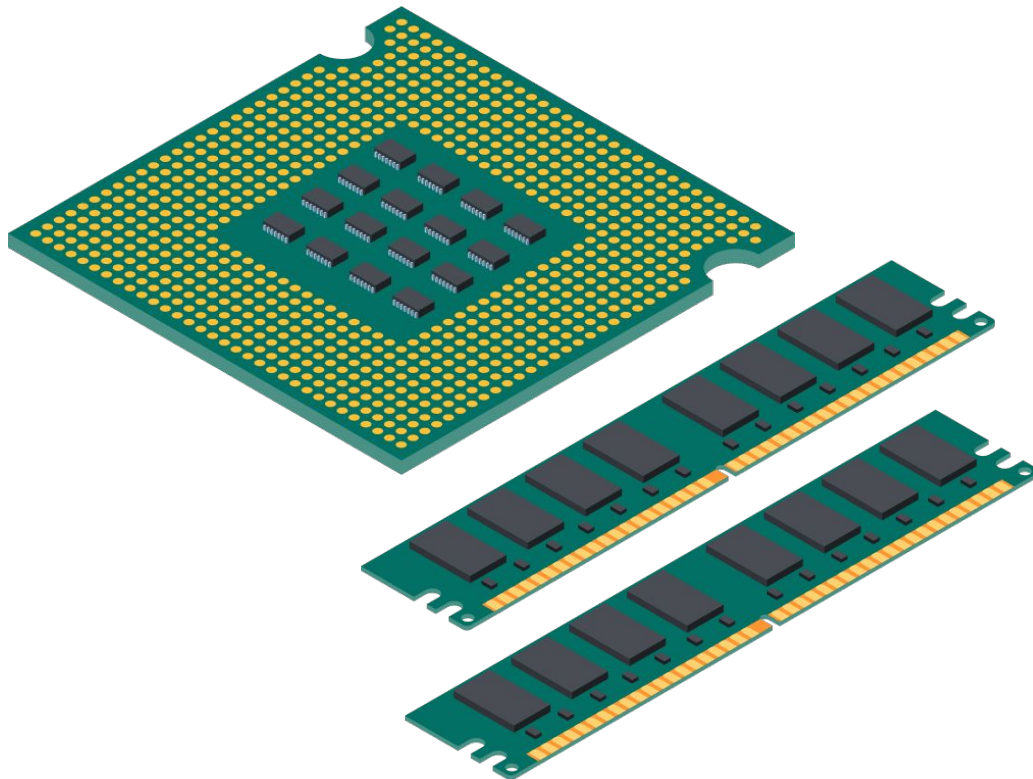
- Your feedback is important to us. Please consider taking a quick 3-question survey [here](#) to share your feedback on this deck.



Compute resources can be considered the brains and processing power required by applications and systems to carry out computational tasks via a series of instructions

WHAT IS COMPUTE?

Compute is closely related to common server components such as CPUs and RAM



ELASTIC COMPUTE CLOUD (EC2)

EC2 allows you to deploy virtual servers within your AWS environment. Most people will require an EC2 instance within their environment as a part of at least one of their solutions

The EC2 service can be broken down into the following components:

- Amazon Machine Images (AMIs)**

- Instance types**

- Instance Purchasing Options**

- Tenancy**

- User Data**

- Storage options**

- Security**



EC2 CONTAINER SERVICE (**ECS**)

This service allows you to run Docker-enabled applications packaged as containers across a cluster of EC2 instances without requiring you to manage a complex and administratively heavy cluster management system

The burden of managing your own cluster management system is abstracted with the ECS service by passing that responsibility over to AWS, specifically through the use of AWS Fargate



ELASTIC CONTAINER REGISTRY **(ECR)**

ECR provides a secure location to store and manage your docker images

This is a fully managed service, so you don't need to provision any infrastructure to allow you to create this registry of docker images

This service allows developers to push, pull and manage their library of docker images in a central and secure location



ELASTIC CONTAINER REGISTRY **(ECR)**

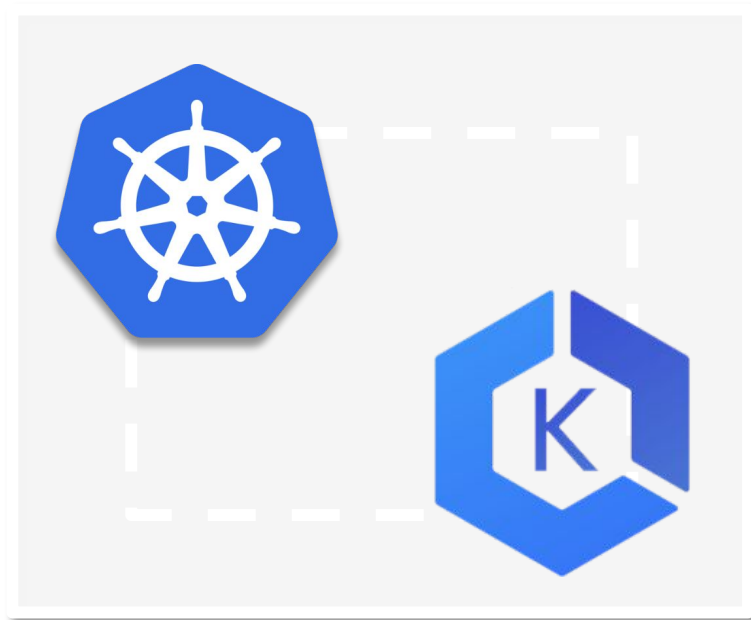
These are the main components used in ECR:



ECS FOR KUBERNETES (**EKS**)

EKS provides a managed service allowing you to **run Kubernetes across your AWS infrastructure** without having to take care of provisioning and running the Kubernetes management infrastructure in what's referred to as **the control plane**

You only need to provision and maintain the **worker nodes**.



KUBERNETES CONTROL PLANE

There are a number of different components that make up the control plane and these include **a number of different APIs, the kubelet processes and the Kubernetes Master**

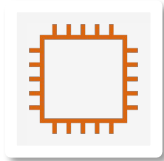
The control plane schedules containers onto nodes

The control plane also tracks the state of all Kubernetes objects by continually monitoring the objects

In EKS, AWS is responsible for provisioning, scaling and managing the control plane, and they do this by utilizing multiple availability zones for additional resilience

WORKER NODES

Kubernetes clusters are composed of nodes



A node is a worker machine in Kubernetes. It runs as an on-demand EC2 instance and includes software to run containers



For each node created, a specific AMI is used, which also ensures Docker and the kubelet is installed for security controls



Once the worker nodes are provisioned they can then connect to EKS using an endpoint

WORKING WITH EKS

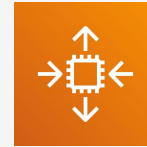
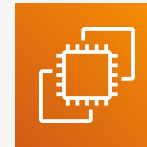
- Create an EKS Service Role
- Create an EKS Cluster VPC
- Install kubectl and the AWS-IAM-Authenticator
- Create your EKS Cluster
- Configure kubectl for EKS
- Provision and configure worker nodes
- Configure the worker nodes to join the EKS cluster



AWS ELASTIC BEANSTALK

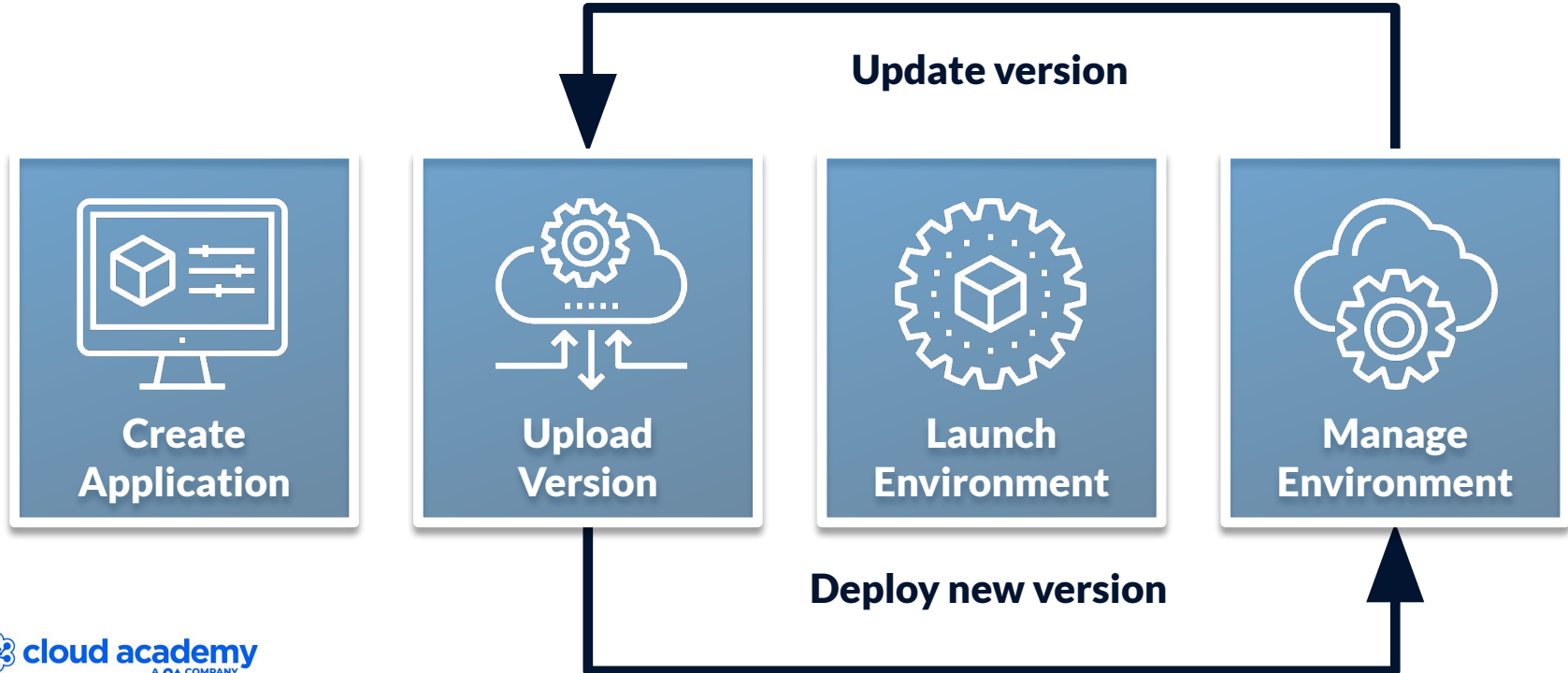
AWS Elastic Beanstalk is an AWS managed service that takes your uploaded code of your web application code and automatically provisions and deploys the required resources within AWS to make the web application operational

These resources include EC2, Auto Scaling, application health-monitoring and Elastic Load Balancing, in addition to capacity provisioning



ELASTIC BEANSTALK WORKFLOW

AWS Elastic Beanstalk operates a very simple workflow process for your application deployment and ongoing management



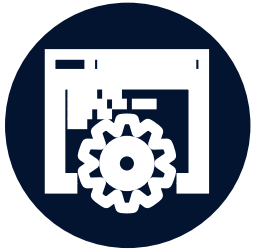
AWS LAMBDA



AWS Lambda is a serverless compute service that allows you to run your application code without having to manage EC2 instances



Serverless means that you do not need to worry about provisioning and managing your own compute resource to run your own code, instead this is managed and provisioned by AWS



The service does require compute power to carry out your code requests, but because the AWS user does not need to be concerned with what managing this compute power or where its provisioned from, it's considered 'serverless' from the user perspective

COMPONENTS OF AWS LAMBDA

The following elements form the key constructs of a Lambda Application



The **Lambda function** is compiled of your own code that you want Lambda to invoke as per defined triggers



A **Trigger** is essentially an operation from an event source that causes the function to invoke



Events sources are AWS services that can be used to trigger your Lambda functions



Downstream resources are resources that are required during the execution of your Lambda Function



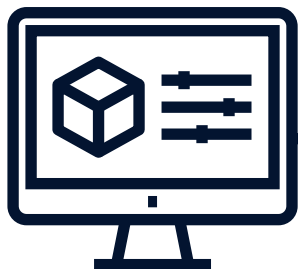
Log streams help to identity issues and troubleshoot issues with your Lambda function

These log streams would essentially be a sequence of events that all come from the same function and recorded in CloudWatch

CREATING LAMBDA FUNCTIONS

At a high level, the configuration steps for creating a Lambda function via the AWS Management Console could consist of:

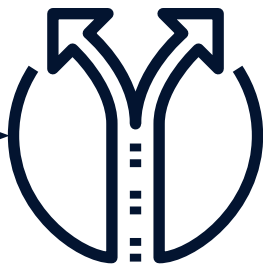
Selecting a Blueprint



Select a blueprint template provided by AWS Lambda

Ex: *S3-get-object* – an S3 trigger that retrieves metadata

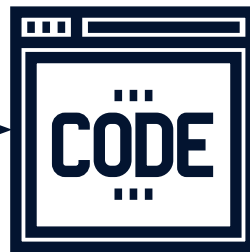
Configure Triggers



Define the trigger for your Lambda function

Ex: specifying the S3 bucket for your function

Configure Function



Upload code or edit it in-line

Define the required resources, maximum execution timeout, IAM Role and Handler Name

WHAT IS BATCH COMPUTING?



Batch Computing is primarily used in specialist use cases which require a vast amount of compute power across a cluster of compute resources to complete batch processing executing a series of tasks

With AWS Batch many of these constraints, administration activities and maintenance tasks are removed

AWS BATCH COMPONENTS

Jobs: A job is classed as the unit of work run by AWS Batch.



Job definitions: These define specific parameters for the jobs themselves.

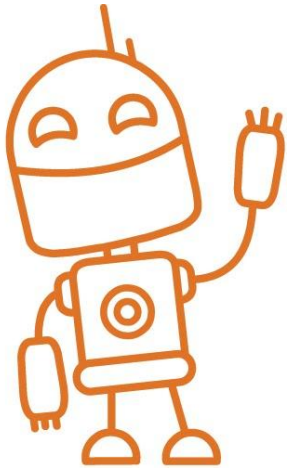
Job queues: Jobs that are scheduled are placed into a job queue until they run.

Job scheduling: This schedules when a job should be run and from which compute environment.

Compute environments: The environments that contain the compute resources to carry out the job.

AMAZON LIGHTSAIL

Amazon Lightsail provides a virtual private server (VPS). It has been designed to be simple, quick and very easy to use at a low cost point, for small scale use cases by small businesses or for single users.



It is commonly used to host simple websites, small applications and blogs

You can run multiple Lightsail instances together allowing them to communicate

It is possible to connect it to other AWS resources and to your existing VPC running within AWS via a peering connection

HELP US IMPROVE

- Your feedback is important to us. Please consider taking a quick 3-question survey [here](#) to share your feedback on this deck.

