Cheatography

# Module 2.1-AWS Core Services - Compute Cheat Sheet

by Vetrikumar Manugaran via cheatography.com/69473/cs/17535/

### Part 1 : Compute Services Overview (copy)

Running server on premises is expensive and hardware needs to be procured based on the project plans offered rather than in the reality usage

AWS offers flexilibilty and cost effectiveness by scale up your workload

Demand Increases-Easily Scale up

Demand Drops - can scale down to save money and easily scale down to save money and resources

You don't need to pay for what you're not using

### Compute Services Overview

Amazon Elastic Compute Cloud (Amazon EC2)- Webservice that provide resizable compute capacity

Can select a variety of OS and Resource Configurations(memory,CPU, Storage)

Benefits-Can cost effectively scale resources up and down based on your needs

### Compute Services Overview (copy)

AWS Lambda- lets you run code without provisioning or managing servers

There is no charge when your code's are not running

Can run code virtually in - mobile, internet of Things(IoT), streaming service

### Auto Scaling

Allows Resources to scale in and out to match workload demand

### Elastic Load Balancer

Automatically distributes incoming application traffic across multiple applications, microservices and containers hosted on Amazon EC2 instances

### AWS Elastic Beanstalk

Platform as a Service-facilitates quick deployement of your applications by providing all the application services

### AWS Lightsail

Jump start for your project- a virtual machine, SSD-based storage, data transfer, DNS Management and a static IP Address

### Amazon Elastic Container Service

Highly Scalable, high-performance container management service

Eliminates need to manage cluster management infrastructure

### AWS Elastic Container Service for Kubernetes

makes it simple to run Kubernetes on AWS without needing to install/operate your own kubernatus clusters

### Amazon EC2

Amazon EC2-a computer in the cloud

Refers - the amount of computational power required to fullfill your workload

If workload small-website that receive few visitors, - then your compute needs of workload is very small

EC2-Elastic Computional Cloud

Elastic-You can decrease the amount of servers required by an application automatically based on the current demand

Compute-refers to the compute or server, resources that are being presented

Cloud-refers to the fact that these are cloud-hosted compute resources

OS-Supported Windows and Linux such as Redhat, SUSE, Ubuntu

### Amazon EC2 (cont)

Amazon Machine Images-(AMI)-create images of your server at any time and can be reused to launch instances in the future

offers native support for the IPV6 Protocol

Can be enabled through AWS Management Console, Software Development Kit, and Command Line interface(CLI)

### Benefits of AWS EC2

Scalable: Add more instances when you need them; terminate them when you do not.

Optimizable: Choose from instance types optimized for compute, memory, storage, accelerated computing and general purpose. Each instance type has a variety of sizes. To learn more about instance types see

### Amazon Machine Image(AMI)

An AMI is a template that contains a software configuration. For example, an operating system, an application server, and applications.

From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

You must specify a source AMI when you launch an Amazon EC2 instance.

You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.

You can use different AMIs to launch instances when you need instances with different configurations.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

### Amazon EC2 Pricing

There are four ways to pay for Amazon EC2 instances: On-Demand, Reserved Instances, Spot Instances, and Dedicated Hosts.

Amazon EC2 usage of Linux- and Ubuntu-based instances that are launched in On-Demand, Spot, and Reserved are billed on one-second increments, with a minimum of 60 seconds.

All other types of OS are billed by the hour.

The minimum unite of time that will be charged is a minute, but after your first minute of time, we can account for seconds.

However, if you start then stop an instance in 10 seconds, you will be charged the 60 seconds not 10.

Dedicated Hosts provide you with Amazon EC2 instance capacity on physical servers dedicated for your use.

### Broad set of Compute Instance Types

Instance types comprise varying combinations of CPU, memory, storage, and networking capacity

give you the flexibility to choose the appropriate mix of resources for your applications.

Each instance type includes one or more instance sizes, which allows you to scale your resources to the requirements of your target workload.

### Choosing the Right Amazon EC2 Instance

The Considerations of choosing instance -Core count, Memory Size, storage size & type, network performance, CPU Technologies

Dimensions of new instances that are controlled by the instance type and Amazon Machine Image (AMI) for images launched on AWS: 1. Amount of virtual hardware dedicated to the instance 2. Software loaded on the instance

### Choosing the Right Amazon EC2 Instance (cont)

there are multiple sizes of Instances:- Large, XLarge, 2XLarge, etc.

Amazon EC2 provides different instance types to enable you to choose the CPU, memory, storage, and networking capacity that you need to run your applications.

### Per second Billing

With per-second billing, you pay for only what you use.

It takes cost of unused minutes and seconds in an hour off of the bill, so you can focus on improving your applications instead of maximizing usage to the hour.

Amazon EC2 usage are billed on one second increments, with a minimum of 60 seconds.

Similarly, provisioned storage for EBS volumes will be billed per-second increments, with a 60-second minimum.

Per-second billing is available for instances launched in:

• On-Demand, Reserved and Spot forms • All AWS Regions and Availability Zones • Amazon Linux and Ubuntu

### AWS Lambda - Introduction

an event-driven serverless compute service.

Lambda lets you run code without provisioning or managing servers.

You pay only for the compute time you consume - there is no charge when your code is not running.

### How does AWS Lambda Works ?

Lambda executes your code only when needed and scales automatically to thousands of requests per second

### How does AWS Lambda Works ? (cont)

With Lambda, you can run code for virtually any type of application or backend service - all with zero administration

Just upload your code and Lambda takes care of everything required to run and scale your code with high availability

You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

It should be noted that function execution on Lambda is limited to a maximum of 5 minutes.

### Lambda Key Benefits

You only pay for the compute you use. You don’t pay for compute time when your code is not running.This makes AWS Lambda ideal for variable in intermittent work loads.

You can run code for virtually any application or backend service, all with zero administration, including server and operating system maintenance. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability.

You can set up your code to automatically trigger from other AWS services, or call it directly from any web or mobile app.

Lambda supports a variety of different programming languages including Go, NodeJS, Java, C#, and Python.

### Getting started with Lambda

Can Read again and write down by your own

### Lambda Use Cases

Run code in response to an events

You can use it for event driven computing. For example:

You can run code in response to events, including changes to an S3 bucket or an Amazon DynamoDB table.

You can respond to HTTP requests using Amazon API Gateway.

You can invoke your code using API calls made using the AWS SDKs.

You can build serverless applications that are triggered by Lambda functions.

You can automatically deploy them using AWS CodePipeline and AWS CodeDeploy. AWS Academy Module 2.1: AWS Core Services - Compute ©

### Lambda Example

Get read and write by our own

### AWS Elastic Beanstalk

You can automatically deploy them using AWS CodePipeline and AWS CodeDeploy.

AWS Academy Module 2.1It’s is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

### How does AWS Elastic Beanstalk

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning and load balancing to automatic scaling and application health monitoring

you retain full control over the AWS resources powering your application and can access the underlying resources at any time

### AWS Elastic Beanstalk

Platform as a Service(PaaS)

Quickly deploys, scales, and manages web apps

Reduces management complexity

Keeps control in your hands :-

Choose your instance type

Choose your database

Set ant adjust Auto Scaling

Update your application

Access server log files

Enable HTTPS on load balancer

Elastic Beanstalk is a Platform as a Service. With the entire platform already built, you simply upload your code. This facilitates quick deployment of your applications.

### Benefits of Elastic Beanstalk

Packer Builder

Single Container, Multi Container, or Pre-configured Docker

Go, Java SE, Java with Tomcat, .Net with on Windows Server with IIS, Node.js, PHP, Python, Ruby

No charge for Elastic Beanstalk,

pay only for the underlaying services used.

### Elastic Beanstalk Components

provides all the applications services that you need for your application

The only thing you need to create your code, deploy it according to your needs. This makes is very quick and easy to deploy your application.

Updates to your application are easy when you deploy it. You simply upload the new code.

Write down and the Elastic Beanstalk components

### Elastic Beanstalk key benefits

is the fastest and simplest way to deploy your application on AWS

the AWS Management Console, a Git repository, or an Integrated Development Environment (IDE), such as Eclipse or Visual Studio to upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, automatic scaling, and application health monitoring

You can improve your productivity by focusing on writing code rather than spending time managing and configuring servers, databases, load balancers, firewalls, and networks.

Elastic Beanstalk provisions and operates the infrastructure and manages the application stack (platform) for you, so you do not have to spend the time or develop the expertise.

It keeps the underlying platform running your application up-to-date with the latest patches and updates.

your application can handle peaks in workload or traffic while minimizing your costs.

It automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings

You can use CPU utilization metrics to trigger Auto Scaling actions

You have the freedom to select the AWS resources, such as Amazon EC2 instance type, that are optimal for your application

Elastic Beanstalk lets you "open the hood" and retain full control over the AWS resources powering your application.