AW S A c a d e m y C l o u d F o u n d a t i o n s Module 6: Compute 

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved. Welcome to Module 6: Compute



© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 2Module overview 

• Container services Topics

Activities • Amazon EC2 versus Managed Service

• Compute services overview

Section 2 includes a recorded Amazon EC2 demonstration. The end of this same section

• Amazon EC2

• Hands-on with AWS Lambda

includes a hands-on lab, where you will practice launching an EC2 instance by using the AWS 

• Hands-on with AWS Elastic Beanstalk

• Amazon EC2 cost optimization

• Container services

Demo

• Introduction to AWS Lambda

• Recorded demonstration of Amazon EC2

• Introduction to AWS Elastic Beanstalk

Lab

• Introduction to Amazon EC2

Finally, you will be asked to complete a knowledge check that will test your understanding of

the key concepts that are covered in this module.

Knowledge check

This module will address the following topics:

• Compute services overview

• Amazon EC2

• Amazon EC2 cost optimization

• Introduction to AWS Lambda

• Introduction to AWS Elastic Beanstalk

Management Console. There is also an activity in this section that has you compare the advantages and disadvantages of running a database deployment on Amazon EC2, versus running it on Amazon Relational Database Service (RDS).

Section 5 includes a hands-on AWS Lambda activity and section 6 includes a hands-on Elastic Beanstalk activity.

Module objectives 

After completing this module, you should be able to: • Provide an overview of different AWS compute services in the cloud • Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2) • Identify the functionality in the EC2 console

• Perform basic functions in Amazon EC2 to build a virtual computing environment • Identify Amazon EC2 cost optimization elements

• Demonstrate when to use AWS Elastic Beanstalk

• Demonstrate when to use AWS Lambda

• Identify how to run containerized applications in a cluster of managed servers

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 3

After completing this module, you should be able to:

• Provide an overview of different AWS compute services in the cloud • Demonstrate why to use Amazon Elastic Compute Cloud (Amazon EC2) • Identify the functionality in the EC2 console

• Perform basic functions in EC2 to build a virtual computing environment • Identify EC2 cost optimization elements

• Demonstrate when to use AWS Elastic Beanstalk • Demonstrate when to use AWS Lambda

• Identify how to run containerized applications in a cluster of managed servers

M o d u l e 6 : C o m p u t e

Section 1: Compute services overview

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. Introducing Section 1: Compute services overview.

AWS compute services 

• ~~Amazon Elasti~~c Co~~mpute Cloud~~ (Am~~azon EC2)~~ provides resizable virtual machines. 

Amazon Web Services (AWS) offers many compute services. This module will discuss the highlighted services. • ~~Amazon Elasti~~c Co~~ntainer Regist~~ry ~~(Amazon ECR)~~ is used to store and retrieve Docker

Amazon EC2 Amazon EC2

Amazon Elastic

VMware Cloud

Amazon Elastic

Auto Scaling Container Registry

on AWS

Container Service

(Amazon ECR)

(Amazon ECS)

AWS Lambda Amazon Elastic Amazon Lightsail

AWS Elastic

Kubernetes Service

AWS Batch

Beanstalk

(Amazon EKS)



AWS Fargate AWS Outposts

AWS Serverless

Application Repository

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

5

Amazon Web Services (AWS) offers many compute services. Here is a brief summary of what each compute service offers:

• Amazon EC2 Auto Scaling supports application availability by allowing you to define conditions that will automatically launch or terminate EC2 instances.

images.

• Amazon Elastic Container Service (Amazon ECS) is a container orchestration service that supports Docker.

• VMware Cloud on AWS enables you to provision a hybrid cloud without custom hardware.

• AWS Elastic Beanstalk provides a simple way to run and manage web applications. • AWS Lambda is a serverless compute solution. You pay only for the compute time that you use.

• Amazon Elastic Kubernetes Service (Amazon EKS) enables you to run managed Kubernetes on AWS.

• Amazon Lightsail provides a simple-to-use service for building an application or website. • AWS Batch provides a tool for running batch jobs at any scale. • AWS Fargate provides a way to run containers that reduce the need for you to manage servers or clusters.

• AWS Outposts provides a way to run select AWS services in your on-premises data center.• AWS Serverless Application Repository provides a way to discover, deploy, and publish serverless applications.

This module will discuss details of the services that are highlighted on the slide.

Key Concepts Characteristics Ease of Use 

Categorizing compute services

| Services |  |  |  |
| --- | --- | --- | --- |
| •Amazon EC2 | • Infrastructure as a service (IaaS)• Instance-based• Virtual machines | • Provision virtual machines that you  can manage as you choose | A familiar concept to many IT professionals. |
| •AWS Lambda | • Serverlesscomputing• Function-based • Low-cost | • Write and deploy code that executes  on a schedule or that can be  triggered by events    • Use when possible (architect for the cloud) | A relatively new concept for many IT staff members, but easy  to use after you learn how. |
| •Amazon ECS •Amazon EKS •AWS Fargate •Amazon ECR | • Container-based computing  • Instance-based | • Spin up and execute jobs more  quickly | AWS Fargate reduces  administrative overhead, but  you can use options that give you more control. |
| •AWS Elastic  Beanstalk | • Platform as a service (PaaS)  • For web applications | • Focus on your code (building your  application)  • Can easily tie into other services—databases, Domain Name System (DNS), etc. | Fast and easy to get started. |

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 6

You can think of each AWS compute service as belonging to one of four broad categories: virtual machines (VMs) that provide infrastructure as a service (IaaS), serverless, container based, and platform as a service (PaaS).

Amazon EC2 provides virtual machines, and you can think of it as infrastructure as a service (IaaS). IaaS services provide flexibility and leave many of the server management responsibilities to you. You choose the operating system, and you also choose the size and resource capabilities of the servers that you launch. For IT professionals who have experience using on-premises computing, virtual machines are a familiar concept. Amazon EC2 was one of the first AWS services, and it remains one of the most popular services.

AWS Lambda is a zero-administration compute platform. AWS Lambda enables you to run code without provisioning or managing servers. You pay only for the compute time that is consumed. This serverless technology concept is relatively new to many IT professionals. However, it is becoming more popular because it supports cloud-native architectures, which enable massive scalability at a lower cost than running servers 24/7 to support the same workloads.

Container-based services—including Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, AWS Fargate, and Amazon Elastic Container Registry—enable you to run multiple workloads on a single operating system (OS). Containers spin up more quickly than virtual machines, thus offering responsiveness. Container-based solutions continue to grow in popularity.

Finally, AWS Elastic Beanstalk provides a platform as a service (PaaS). It facilitates the quick deployment of applications that you create by providing all the application services that you need. AWS manages the OS, the application server, and the other infrastructure components so that you can focus on developing your application code.

compute environments. The optimal compute service or services that you use will depend on 

Choosing the optimal compute service

• The optimal compute service or services that you use will depend on your use case

• Some aspects to consider –

• What is your application design?

• What are your usage patterns?

• Which configuration settings will you want to manage? • Selecting the wrong compute solution for an architecture can lead to lower performance efficiency

• A good starting place—Understand the available compute options

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 7

AWS offers many compute services because different use cases benefit from different your use case.

Often, the compute architecture that you use is determined by legacy code. However, that does not mean that you cannot evolve the architecture to take advantage of proven cloud native designs.

Best practices include:

• Evaluate the available compute options

• Understand the available compute configuration options • Collect computer-related metrics

• Use the available elasticity of resources

• Re-evaluate compute needs based on metrics

Sometimes, a customer will start with one compute solution and decide to change the designbased on their analysis of metrics. If you are interested in seeing an example of how a customer modified their choice of compute services for a particular use case, view this Inventory Tracking solution video.

M o d u l e 6 : C o m p u t e

Section 2: Amazon EC2

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. Introducing Section 2: Amazon EC2.

EC2 instances Photo by Taylor Vick on Unsplash 

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved. ~~9~~Example uses of Amazon

Amazon Elastic Compute Cloud (Amazon EC2) Amazon E~~la~~st~~ic Compute Clou~~d (Amazon EC2) provides virtual mach~~ines where you can host~~

 Application server

 Web server

 Database server

 Game server

 Mail server

 Media server

 Catalog server

 File server

 Computing server

 Proxy server

Photo by panumas nikhomkhai from Pexels

Running servers on-premises is an expensive undertaking. Hardware must be procured, and this procurement can be based on project plans instead of the reality of how the servers are used. Data centers are expensive to build, staff, and maintain. Organizations also need to permanently provision a sufficient amount of hardware to handle traffic spikes and peak workloads. After traditional on-premises deployments are built, server capacity might be unused and idle for a significant portion of the time that the servers are running, which is wasteful.

the same kinds of applications that you might run on a traditional on-premises server. It provides secure, resizable compute capacity in the cloud. EC2 instances can support a variety of workloads. Common uses for EC2 instances include, but are not limited to:

• Application servers

• Web servers

• Database servers

• Game servers

• Mail servers

• Media servers

• Catalog servers • File servers • Computing servers • Proxy servers

Amazon EC2 overview

• Amazon Elastic Compute Cloud (Amazon EC2)

• Provides virtual machines—referred to as EC2

instances—in the cloud.

• Gives you full control over the guest operating system 

over the Windows or Linux operating system that runs on the instance. Most server operating

(Windows or Linux) on each instance.

• You can launch instances of any size into an

Availability Zone anywhere in the world.

• Launch instances from Amazon Machine Images (AMIs).

Amazon

• Launch instances with a few clicks or a line of code, and

EC2

they are ready in minutes.

• You can control traffic to and from instances.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 10

The EC2 in Amazon EC2 stands for Elastic Compute Cloud:

• Elastic refers to the fact that you can easily increase or decrease the number of servers you run to support an application automatically, and you can also increase or decrease the size of existing servers.

• Compute refers to reason why most users run servers in the first place, which is to host running applications or process data—actions that require compute resources, including processing power (CPU) and memory (RAM).

• Cloud refers to the fact that the EC2 instances that you run are hosted in the cloud. Amazon EC2 provides virtual machines in the cloud and gives you full administrative control

systems are supported, including: Windows 2008, 2012, 2016, and 2019, Red Hat, SuSE, Ubuntu, and Amazon Linux.

An operating system that runs on a virtual machine is often called a guest operating system to distinguish it from the host operating system. The host operating system is directly installed on any server hardware that hosts one or more virtual machines.

With Amazon EC2, you can launch any number of instances of any size into any Availability Zone anywhere in the world in a matter of minutes. Instances launch from Amazon Machine Images (AMIs), which are effectively virtual machine templates. AMIs are discussed in more detail later in this module.

You can control traffic to and from instances by using security groups. Also, because the servers run in the AWS Cloud, you can build solutions that take use multiple AWS services.

Console Launch Instance Wizard. You will have the opportunity to experience using the 

Launching an Amazon EC2 instance

This section of the module walks

through nine key decisions to

make when you create an EC2

instance by using the AWS

Management Console Launch

Instance Wizard.

launch an instance. The slides cover essential concepts that are good to know when you

Along the way, essential

make these choices. These concepts are described to help you understand the options that

Amazon EC2 concepts will be

are available, and the effects of the decisions that you will make.

explored.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 11

The first time you launch an Amazon EC2 instance, you will likely use the AWS Management Launch Wizard in the lab that is in this module.

The Launch Instance Wizard makes it easy to launch an instance. For example, if you choose to accept all the default settings, you can skip most of the steps that are provided by the wizard and launch an EC2 instance in as few as six clicks. An example of this process is shown in the demonstration at the end of this section.

However, for most deployments you will want to modify the default settings so that the servers you launch are deployed in a way that matches your specific needs.

The next series of slides introduce you to the essential choices that you must make when you

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data 

6. Storage options

7. Tags

8. Security group

• Quick Start – Linux and W~~indows AM~~Is that are provided by AWS

9. Key pairAMI Instance • My AMIs – Any AMIs that ~~you create~~d

• AWS Marketplace – Pre-c~~onfigured t~~emplates from third parties

• Community AMIs – AMIs ~~shared by~~ others; use at your own risk

1. Select an AMI

Choices made using the Launch Instance Wizard:

Launch instance

• Amazon Machine Image (AMI)

• Is a template that is used to create an EC2 instance (which is a virtual machine, or VM, that runs in the AWS Cloud)

• Contains a Windows or Linux operating system

• Often also has some software pre-installed • AMI choices:

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

12

An Amazon Machine Image (AMI) provides information that is required to launch an EC2

instance. You must specify a source AMI when you launch an instance. You can use different AMIs to launch different types of instances. For example, you can choose one AMI to launch an instance that will become a web server and another AMI to deploy an instance that will host an application server. You can also launch multiple instances from a single AMI.

An AMI includes the following components:

• A template for the root volume of the instance. A root volume typically contains an operating system (OS) and everything that was installed in that OS (applications, libraries, etc.). Amazon EC2 copies the template to the root volume of a new EC2 instance, and then starts it.

• Launch permissions that control which AWS accounts can use the AMI. • A block device mapping that specifies the volumes to attach to the instance (if any) when it is launched.

You can choose many AMIs:

• Quick Start – AWS offers a number of pre-built AMIs for launching your instances. These AMIs include many Linux and Windows options.

• My AMIs – These AMIs are AMIs that you created.

• AWS Marketplace – The AWS Marketplace offers a digital catalog that lists thousands of • Community AMIs – These AMIs are created by people all around the world. These AMIs

production or corporate environment.

software solutions. These AMIs can offer specific use cases to help you get started quickly.

are not checked by AWS, so use them at your own risk. Community AMIs can offer many different solutions to various problems, but use them with care. Avoid using them in any

a~~n EC2 instance, and then save the EC2 instance as an AMI. You c~~a~~n~~ t~~hen launch~~ an EC2 

Creating a new AMI: Example

AWS Cloud

AMI details

Region A

Connect to the instance

and manually modify it or

Quick Start or

run a script that modifies

Amazon EC2 stops the ~~instance, creates a snapsho~~t of its root volu~~m~~e, and finally registers

other

the instance (for

Capture as

example, upgrade

Launch an

existing AMI

a new AMI

Starter instance 

installed software)

AMI

1 2 3 Modified

Unmodified

InstanceNew

Instance

MyAMI

(Optional) Import

a virtual machine

Region B

AMI



New AMI

Copy the AMI to any other Regions where you want to use it 4

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 13

An AMI is created from an EC2 instance. You can import a virtual machine so that it becomes

instance from that AMI. Alternatively, you can start with an existing AMI—such as of the Quick Start AMIs provided by AWS—and create an EC2 instance from it.

Regardless of which options you chose (step 1), you will have what the diagram refers to as an unmodified instance. From that instance, you might then create a golden instance—that is, a virtual machine that you configured with the specific OS and application settings that you want (step 2)—and then capture that as a new AMI (step 3). When you create an AMI,

the snapshot as an AMI.

After an AMI is registered, the AMI can be used to launch new instances in the same AWS Region. The new AMI can now be thought of as a new starter AMI. You might want to also copy the AMI to other Regions (step 4), so that EC2 instances can also be launched in those locations.

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

Amazon EC2 provides a selection of instance types that optimized to fit different use cases.

2. Select an instance type

capacity. The different instance types give you the flexibility to choose the appropriate mix of

• Consider your use case

Choices made using the

Launch Instance Wizard:

• How will the EC2 instance you create be used?

• The instance type that you choose determines –

• Memory (RAM)

• Processing power (CPU)

• Disk space and disk type (Storage)

• Network performance

• Instance type categories –

• General purpose

• Compute optimized

• Memory optimized

• Storage optimized

• Accelerated computing

• Instance types offer family, generation, and size

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

14

After you choose the AMI for launching the instance, you must choose on an instance type.

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

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

Instance type categories include general purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instances. Each instance type category offers many instance types to choose from.

Name vCPU Memory (GB) Storage

t3.nano 2 0.5 EBS-Only

t3.micro 2 1 EBS-Only

t3.small 2 2 EBS-Only

t3.medium 2 4 EBS-Only

t3.large 2 8 EBS-Only

t3.xlarge 4 16 EBS-Only

t3.2xlarge 8 32 EBS-OnlyExample instance sizes

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 15W~~hen you look at an~~ EC2 instance type, you will see that its name has several parts. For

EC2 instance type naming and sizes

T is the family name, which is then followed by a number. Here, that number is 3. The number is the generation number of that type. So, a t3 instance is the third generation of the T family. In general, instance types that are of a higher generation are more powerful

Instance type details

Instance

Instance type naming

• Example: t3.large

• T is the family name

• 3 is the generation number

• Large is the size

example, consider the T type.

and provide a better value for the price.

The next part of the name is the size portion of the instance. When you compare sizes, it is important to look at the coefficient portion of the size category.

For example, a t3.2xlarge has twice the vCPU and memory of a t3.xlarge. The t3.xlarge has, in turn, twice the vCPU and memory of a t3.large.

It is also important to note that network bandwidth is also tied to the size of the Amazon EC2 instance. If you will run jobs that will be very network-intensive, you might be required to increase the instance specifications to meet your needs.

Optimized~~Instance Types~~ a1, m4, m5, © 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. ~~16~~Instance type details s~~torage amount, memory amount, and network performance. The chart provides a high-lev~~el 

Select instance type: Based on use case num~~bers fit into each category type. Consider a few of the instance types in more detail:~~

• T3 i~~nstances provide burstable performance~~ general purpose instances that provide a

General

Compute

Memory

Accelerated

Storage

Purpose

Optimized

Optimized

Computing

t2, t3 c4, c5 r4, r5,

f1, g3, g4,

high performance at a low price per compute ratio. Use cases include scientific modeling,

p2, p3 d2, h1, i3

x1, z1

• R5 instances are optimized for memory-intensive applications. Use cases include high

Use Case Broad High

In-memory

Machine

Distributed file

performance

databases

learning

systems

big data, Apache Hadoop or Apache Spark clusters, and other enterprise applications.

Instance types vary in several ways, including: CPU type, CPU or core count, storage type, view of the different instance categories, and which instance type families and generation

baseline level of CPU performance with the ability to burst above the baseline. Use cases for this type of instance include websites and web applications, development environments, build servers, code repositories, microservices, test and staging environments, and line-of-business applications.

• C5 instances are optimized for compute-intensive workloads, and deliver cost-effective batch processing, ad serving, highly scalable multiplayer gaming, and video encoding.

performance databases, data mining and analysis, in-memory databases, distributed web scale in-memory caches, applications that perform real-time processing of unstructured

To learn more about each instance type, see the Amazon EC2 Instance Types documentation.

Instance types: Networking features

• The network bandwidth (Gbps) varies by instance type.

• See Amazon EC2 Instance Types to compare.

• To maximize networking and bandwidth performance of your instance type: • If you have interdependent instances, launch them into a cluster placement group. • Enable enhanced networking.

instances to meet the needs of your workload. For example, you might specify that three

• Enhanced networking types are supported on most instance types. • See the Networking and Storage Features documentation for details.

• Enhanced networking types –

• Elastic Network Adapter (ENA): Supports network speeds of up to 100 Gbps. • Intel 82599 Virtual Function interface: Supports network speeds of up to 10 Gbps.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 17

In addition to considering the CPU, RAM, and storage needs of your workloads, it is also important to consider your network bandwidth requirements.

Each instance type provides a documented network performance level. For example, an a1.medium instance will provide up to 10 Gbps, but a p3dn.24xlarge instance provides up to 100 Gbps. Choose an instance type that meets your requirements.

When you launch multiple new EC2 instances, Amazon EC2 attempts to place the instances so that they are spread out across the underlying hardware by default. It does this to minimize correlated failures. However, if you want to specify specific placement criteria, you can use placement groups to influence the placement of a group of interdependent

instances should all be deployed in the same Availability Zone to ensure lower network latency and higher network throughput between instances. See the Placement Groupdocumentation for details.

Many instance types also enable you to configure enhanced networking to get significantly higher packet per second (PPS) performance, lower delay variation in the arrival of packets over the network (network jitter), and lower latencies. See the Elastic Network Adapter (ENA)documentation for details.

1. AMI 

2. Instance Type 

3. Network settings 

4. IAM role 

5. User data

6. Storage options 

7. Tags

8. Security group

9. Key pair

After you have choose an AMI and an instan~~ce type, you m~~ust spec~~ify the ne~~t~~w~~ork location

3. Specify network settings

When you launch an instance in a default VPC, AWS will assign it a ~~public IP add~~ressby

Choices made by using the

• Where should the instance be deployed?

Launch Instance Wizard:

• Identify the VPC and optionally the subnet

• Should a public IP address be automatically assigned?

• To make it internet-accessible

AWS Cloud

enabling or disabling the public IP addressing feature during launch ~~(which over~~rides the

Region

Availability Zone 1 Availability Zone 2

VPC

Public subnet

Example: specify

to deploy the

instance here

Instance

Private subnet

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

18

where the EC2 instance will be deployed. The choice of Region must be made before you start the Launch Instance Wizard. Verify that you are in the correct Region page of the Amazon EC2 console before you choose Launch Instance.

default. When you launch an instance into a nondefault VPC, the subnet has an attribute that determines whether instances launched into that subnet receive a public IP address from the public IPv4 address pool. By default, AWS will not assign a public IP address to instances that are launched in a nondefault subnet. You can control whether your instance receives a public IP address by either modifying the public IP addressing attribute of your subnet, or by

subnet's public IP addressing attribute).

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

4. Attach IAM role (optional)

Choices made by using the

• Will software on the EC2 instance need to interact with other AWS

Launch Instance Wizard:

services?

• If yes, attach an appropriate IAM Role.

• An AWS Identity and Access Management (IAM) role that is attached

to an EC2 instance is kept in an instance profile.

• You are not restricted to attaching a role only at instance launch.

• You can also attach a role to an instance that already exists.

Example:

Application on

In the example, you see that an IAM role is used to grant permissions to an application that 

attached to

instance can

access

Role that grants Amazon

S3 bucket

Instance

Simple Storage Service (Amazon S3) bucket access

with objects

permissions

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

19

It is common to use EC2 instances to run an application that must make secure API calls to other AWS services. To support these use cases, AWS enables you to attach an AWS Identity and Access Management (IAM) role to an EC2 instance. Without this feature, you might be tempted to place AWS credentials on an EC2 instance so an application that runs on that instance to use. However, you should never store AWS credentials on an EC2 instance. It is highly insecure. Instead, attach an IAM role to the EC2 instance. The IAM role then grants permission to make application programming interface (API) requests to the applications that run on the EC2 instance.

An instance profile is a container for an IAM role. If you use the AWS Management Console to create a role for Amazon EC2, the console automatically creates an instance profile and gives it the same name as the role. When you then use the Amazon EC2 console to launch an instance with an IAM role, you can select a role to associate with the instance. In the console, the list that displays is actually a list of instance profile names.

runs on an EC2 instance. The application must access a bucket in Amazon S3. You can attach an IAM role when you launch the instance, but you can also attach a role to an

already running EC2 instance. When you define a role that can be used by an EC2 instance, you define which accounts or AWS services can assume the role. You also define which API actions and resources the application can use after it assumes the role. If you change a role, the change is propagated to all instances that have the role attached to them.

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

• Use user data ~~scripts to customize the runtime en~~vironment of your

8. Security group

9. Key pair

5. User data script (optional)

User data

Choices made by using the

Launch Instance Wizard:

#!/bin/bash yum update –y yum install -y wget 

AMI Running EC2 instance

• Optionally specify a user data script at instance launch

instance

• Script executes the first time the instance starts • Can be used strategically

• For example, reduce the number of custom AMIs that you build and maintain

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

20

When you create your EC2 instances, you have the option of passing user data to the instance. User data can automate the completion of installations and configurations at instance launch. For example, a user data script might patch and update the instance's operating system, fetch and install software license keys, or install additional software.

In the example user data script, you see a simple three-line Linux Bash shell script. The first line indicates that the script should be run by the Bash shell. The second line invokes the Yellowdog Updater, Modified (YUM) utility, which is commonly used in many Linux distributions—such as Amazon Linux, CentOS, and Red Hat Linux—to retrieve software from an online repository and install it. In line two of the example, that command tells YUM to update all installed packages to the latest versions that are known to the software repository that it is configured to access. Line three of the script indicates that the Wgetutility should be installed. Wget is a common utility for downloading files from the web.

For a Windows instance, the user data script should be written in a format that is compatible with a Command Prompt window (batch commands) or with Windows PowerShell. See the Windows User Data Scripts documentation for details.

When the EC2 instance is created, the user data script will run with root privileges during the final phases of the boot process. On Linux instances, it is executed by the cloud-init

service. On Windows instances, it is executed by the EC2Config or EC2Launch utility. By default, user data only runs the first time that the instance starts up. However, if you would like your user data script to run every time the instance is booted, you can create a Multipurpose Internet Mail Extensions (MIME) multipart file user data script (this process is not commonly done).

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

6. Specify storage

configured to launch more than one storage volume by default to provide storage that is

• Configure the root volume

Choices made by using the

Launch Instance Wizard:

• Where the guest operating system is installed

• Attach additional storage volumes (optional)

• AMI might already include more than one volume

• For each volume, specify:

• The size of the disk (in GB)

• The volume type

• Different types of solid state drives (SSDs) and hard disk

drives (HDDs) are available

• If the volume will be deleted when the instance is

terminated

• If encryption should be used

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

21

When you launch an EC2 instance, you can configure storage options. For example, you can configure the size of the root volume where the guest operating system is installed. You can also attach additional storage volumes when you launch the instance. Some AMIs are also

separate from the root volume.

For each volume that your instance will have, you can specify the size of the disks, the volume types, and whether the storage will be retained if the instance is terminated. You can also specify if encryption should be used.

storage service that is designed to be used with Amazon EC2 for both throughput- and 

Amazon EC2 storage options

• Amazon Elastic Block Store (Amazon EBS) –

Amazon EC2 Instance Store provides temporary block-level storage for your instance. This

• Durable, block-level storage volumes.

• You can stop the instance and start it again, and the data will still be

there.

• Amazon EC2 Instance Store –

• Storage is provided on disks that are attached to the host computer

where the EC2 instance is running.

• If the instance stops, data stored here is deleted.

• Other options for storage (not for the root volume) – • Mount an Amazon Elastic File System (Amazon EFS) file system.

• Connect to Amazon Simple Storage Service (Amazon S3).

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 22

Amazon Elastic Block Store (Amazon EBS) is an easy-to-use, high-performance durable block

transaction-intensive workloads. With Amazon EBS, you can choose from four different volume types to balance the optimal price and performance. You can change volume types or increase volume size without disrupting your critical applications, so you can have cost effective storage when you need it.

storage is located on disks that are physically attached to the host computer. Instance Store works well when you must temporarily store information that changes frequently, such as buffers, caches, scratch data, and other temporary content. You can also use Instance Store for data that is replicated across a fleet of instances, such as a load balanced pool of web servers. If the instances are stopped—either because of user error or a malfunction—the data on the instance store will be deleted.

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic Network File System (NFS) file system for use with AWS Cloud services and on-premises resources. It is built to scale on-demand to petabytes without disrupting applications. It grows and shrinks automatically as you add and remove files, which reduces the need to provision and manage capacity to accommodate growth.

scalability, data availability, security, and performance. You can store and protect any

archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics.

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers amount of data for a variety of use cases, such as websites, mobile apps, backup and restore,

Example storage options 

Instance Store

• Instance 1 characteristics – Amazon Elastic Block

Host computer

Store (Amazon EBS)

• It has an Amazon EBS root volume type for the operating system.

• What will happen if the instance is

Attached as

stopped and then started again?

Attached as 

Root volume

Storage volume

20-GB volume

Ephemeral

• Instance 2 characteristics –

Instance 1 volume 1

Attached as

Storage volume

• It has an Instance Store root volume type for the operating 

Attached as

Amazon EC2 API call. It can only be terminated. However, it could be stopped from within 500-GB volume 

system.

• What will happen if the instance

Root volume

Ephemeral

Instance 2 volume 2

OS or disk failure—which would cause the instance to be terminated. If the instance was

stops (because of user error or a

system malfunction)?

terminated, all the data that was stored on Ephemeral volume 2 would be lost, including the

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 23

Here, you see two examples of how storage options could be configured for EC2 instances.

The Instance 1 example shows that the root volume—which contains the OS and possibly other data—is stored on Amazon EBS. This instance also has two attached volumes. One volume is a 500-GB Amazon EBS storage volume, and the other volume is an Instance Store volume. If this instance was stopped and then started again, the OS would survive and any data that was stored on either the 20-GB Amazon EBS volume or the 500-GB Amazon EBS volume would remain intact. However, any data that was stored on Ephemeral volume 1 would be permanently lost. Instance Store works well for temporarily storing information that changes frequently, such as buffers, caches, scratch data, and other temporary content.

The Instance 2 example shows that the root volume is on an instance store (Ephemeral volume 2). An instance with an Instance Store root volume cannot be stopped by an

the instance's OS (for example, by issuing a shutdown command)—or it could stop because of

OS. You would not be able to start the instance again. Therefore, do not rely on Instance Store for valuable, long-term data. Instead, use more durable data storage, such as Amazon EBS, Amazon EFS, or Amazon S3.

If an instance reboots (intentionally or unintentionally), data on the instance store root volume does persist.

1. AMI 

2. Instance Type

3. Network settings 

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

optional value, both of which you define. Tags enable you to categorize AWS resources, such

7. Add tags

• A tag is a label that you can assign to an AWS

Choices made by using the

Launch Instance Wizard:

resource.

instances is a tag key that is called Name and a tag value that describes the instance, such as

• Consists of a key and an optional value.

• Tagging is how you can attach metadata to an EC2

instance.

• Potential benefits of tagging—Filtering,

automation, cost allocation, and access control.

Example:

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

24

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an

as EC2 instances, in different ways. For example, you might tag instances by purpose, owner, or environment.

Tagging is how you can attach metadata to an EC2 instance.

Tag keys and tag values are case-sensitive. For example, a commonly used tag for EC2

My Web Server. The Name tag is exposed by default in the Amazon EC2 console Instancespage. However, if you create a key that is called name (with lower-case n), it will not appear in the Name column for the list of instances (though it will still appear in the instance details panel in the Tags tab).

It is a best practice to develop Tagging strategies. Using a consistent set of tag keys makes it easier for you to manage your resources. You can also search and filter the resources based on the tags that you add.

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

8. Security group settings

• A security group is a set of firewall rules that control

Choices made by using the

Launch Instance Wizard:

traffic to the instance.

• It exists outside of the instance's guest OS.

• Create rules that specify the source and which ports

that network communications can use.

• Specify the port number and the protocol, such as

Transmission Control Protocol (TCP), User Datagram

Protocol (UDP), or Internet Control Message Protocol

(ICMP).

• Specify the source (for example, an IP address or another

security group) that is allowed to use the rule.

Example rule:

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

25

A security group acts as a virtual firewall that controls network traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, the default security group is used.

You can add rules to each security group. Rules allow traffic to or from its associated instances. You can modify the rules for a security group at any time, and the new rules will be automatically applied to all instances that are associated with the security group. When AWS decides whether to allow traffic to reach an instance, all the rules from all the security groups that are associated with the instance are evaluated. When you launch an instance in a virtual private cloud (VPC), you must either create a new security group or use one that already exists in that VPC. After you launch an instance, you can change its security groups.

When you define a rule, you can specify the allowable source of the network communication (inbound rules) or destination (outbound rules). The source can be an IP address, an IP address range, another security group, a gateway VPC endpoint, or anywhere (which means that all sources will be allowed). By default, a security group includes an outbound rulethat allows all outbound traffic. You can remove the rule and add outbound rules that only allow specific outbound traffic. If your security group has no outbound rules, no outboundtraffic that originates from your instance is allowed.

In the example rule, the rule allows Secure Shell (SSH) traffic over Transmission Control Protocol (TCP) port 22 if the source of the request is My IP. The My IP IP address is calculated

by determining what IP address you are currently connected to the AWS Cloud from when you define the rule.

Network access control lists (network ACLs) can also be used are firewalls to protect subnets in a VPC.

1. AMI 

2. Instance Type

3. Network settings

4. IAM role

5. User data

6. Storage options

7. Tags

8. Security group

9. Key pair

9. Identify or create the key pair

• At instance launch, you specify an existing key

Choices made by using the

Launch Instance Wizard:

pair or create a new key pair.

• A key pair consists of –

mykey.pem

• A public key that AWS stores.

• A private key file that you store.

• It enables secure connections to the instance.

• For Windows AMIs –

• Use the private key to obtain the administrator

password that you need to log in to your instance.

• For Linux AMIs –

• Use the private key to use SSH to securely connect to

your instance.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

26

After you specify all the required configurations to launch an EC2 instance, and after you customize any optional EC2 launch wizard configuration settings, you are presented with a Review Instance Launch window. If you then choose Launch, a dialog asks you to choose an existing key pair, proceed without a key pair, or create a new key pair before you can choose Launch Instances and create the EC2 instance.

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. The technology uses a public key to encrypt a piece of data, and then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair. Public key cryptography enables you to securely access your instances by using a private key instead of a password.

When you launch an instance, you specify a key pair. You can specify an existing key pair or a new key pair that you create at launch. If you create a new key pair, download it and save it in a safe location. This opportunity is the only chance you get to save the private key file.

To connect to a Windows instance, use the private key to obtain the administrator password, and then log in to the EC2 instance's Windows Desktop by using Remote Desktop Protocol (RDP). To establish an SSH connection from a Windows machine to an Amazon EC2 instance, you can use a tool such as PuTTY, which will require the same private key.

With Linux instances, at boot time, the public key content is placed on the instance. An entry is created in within ~/.ssh/authorized\_keys. To log in to your Linux instance (for example, by using SSH), you must provide the private key when you establish the connection.

Amazon EC2 console view of a running EC2 

instance

Information about the available instance includes IP address and DNS address information, 

the instance type, the unique instance ID that was assigned to the instance, the AMI ID of the

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 27

After you choose Launch Instances and then choose View Instances, you will be presented with a screen that looks similar to the example.

Many of the settings that you specified during launch are visible in the Descriptionpanel.

AMI that you used to launch the instance, the VPC ID, the subnet ID, and more.

Many of these details provide hyperlinks that you can choose to learn more information about the resources that are relevant to the EC2 instance you launched.

Another option: Launch an EC2 instance with the AWS Command Line Interface 

• aws ~~– Specifies an in~~vocation of theaws command line utility. 

• EC2 instances can also be created

• ec2 ~~– Specifies an in~~vocation of theec2 service command.

programmatically.

AWS Command Line

• run-instances ~~–~~ Is the subcom~~mand~~ that is being invoked.

Interface (AWS CLI)

• This example shows how simple the

Example command:

• image-id – This parameter is followed by an AMI ID. All AMIs have a unique AMI ID.

command can be.

• count – You can specify more than one.

aws ec2 run-instances \

• This command assumes that the key pair and

--image-id ami-1a2b3c4d \

• instance-type – You can specify the instance type to create (for example) a c3.large

security group already exist.

--count 1 \ --instance-type c3.large \

• More options could be specified. See the AWS CLI

• key-name – In the example, assume that MyKeyPair already exists.

--key-name MyKeyPair \

Command Reference for details.

--security-groups MySecurityGroup \

• security-groups - In this example, assume that MySecurityGroup already exists.

--region us-east-1

• region - AMIs exist in an AWS Region, so you must specify the Region where the AWS

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 28

You can also launch EC2 instances programmatically, either by using the AWS Command Line Interface (AWS CLI) or one of the AWS software development kits (SDKs).

In the example AWS CLI command, you see a single command that specifies the minimal information that is needed to launch an instance. The command includes the following information:

The rest of the command specifies several parameters, including: instance

CLI will find the AMI and launch the EC2 instance.

The command should successfully create an EC2 instance if:

• The command is properly formed

• The resources that the command needs already exist

• You have sufficient permissions to run the command • You have sufficient capacity in the AWS account

If the command is successful, the API responds to the command with the instance ID and other relevant data for your application to use in subsequent API requests.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 29

Amazon EC2 instance lifecycle

• Pending ~~– When an instance is first launched from an AMI, or when you sta~~rt a stopped

Only instances backed by Amazon EBS

• Running – When the inst~~ance is full~~y booted and ready, it exits the pendingstate and

pending Launch

Start

AMI

• Rebooting – AWS recom~~mends you~~ reboot an instance by using the Amazon EC2 console,

Reboot

Stop

rebooting running

Stop

stopping stopped

Hibernate

Terminate

shutting 

down

• Shutting down – This state is an intermediary state between running and terminated. 

Terminate

terminated

• Terminated – A terminated instance remains visible in the Amazon EC2 console for a while

Here, you see the lifecycle of an instance. The arrows show actions that you can take and the boxes show the state the instance will enter after that action. An instance can be in one of the following states:

instance, it enters the pending state when the instance is booted and deployed to a host computer. The instance type that you specified at launch determines the hardware of the host computer for your instance.

enters the running state. You can connect over the internet to your running instance.

AWS CLI, or AWS SDKs instead of invoking a reboot from within the guest operating system (OS). A rebooted instance stays on the same physical host, maintains the same public DNS name and public IP address, and if it has instance store volumes, it retains the data on those volumes.

before the virtual machine is deleted. However, you can’t connect to or recover a terminated instance.

• Stopping – Instances that are backed by Amazon EBS can be stopped. They enter the

• Stopped – A stopped instance will not incur the same cost as a running instance. Starting a

stopping state before they attain the fully stopped state.

stopped instance puts it back into the pending state, which moves the instance to a new host machine.

Instance hibernation option • Benefits • It saves the contents from the instance memory (RAM).

• On instance restart, RAM contents are reloaded, previously running processes are resumed. • You can save on cost in a hibernated state versus a running state (costs are similar to a stopped instance).

• Prerequisites

• Only certain Linux AMIs (such as Amazon Linux 2) and only certain instance families support it. • Instance must have an encrypted Amazon EBS root volume and a maximum of150 GB RAM. • Hibernation must be enabled at instance launch.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 30

Some instances that are backed by Amazon EBS support hibernation. When you hibernate an instance, the guest OS saves the contents from the instance memory (RAM) to your Amazon EBS root volume. When you restart the instance, the root volume is restored to its previous state, the RAM contents are reloaded, and the processes that were previously running on the instance are resumed.

Only certain Linux AMIs that are backed by Amazon EBS and other certain instance types support hibernation. Hibernation also requires that you encrypt the root EBS volume. In addition, you must enable hibernation when the instance is first launched. You cannot enable hibernation on an existing instance that did not originally have hibernation enabled.

For further details about prerequisites and cost, see the Hibernate Your Linux InstanceAWS documentation page.

Consider using an Elastic IP address • If you require a persistent public IP

• Rebooting an instance will not change

address –

any IP addresses or DNS hostnames.

• Associate an Elastic IP address with the

instance.

• When an instance is stopped and then

started again –

• Elastic IP address characteristics –

• The public IPv4 address and external DNS

hostname will change.

• Can be associated with instances in the Region as needed.

• The private IPv4 address and internal DNS

address in the Region where the instance exists. After the Elastic IP address is allocated, you 

hostname do not change.

• Remains allocated to your account until you choose to release it.

Elastic IP

Address

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

31

A public IP address is an IPv4 address that is reachable from the internet. Each instance that receives a public IP address is also given an external DNS hostname. For example, if the public IP address assigned to the instance is 203.0.113.25, then the external DNS hostname might be ec2-203-0-113-25.compute-1.amazonaws.com.

If you specify that a public IP address should be assigned to your instance, it is assigned from the AWS pool of public IPv4 addresses. The public IP address is not associated with your AWS account. When a public IP address is disassociated from your instance, it is released back into

the public IPv4 address pool, and you will not be able to specify that you want to reuse it. AWS releases your instance's public IP address when the instance is stopped or terminated. Your stopped instance receives a new public IP address when it is restarted.

If you require a persistent public IP address, you might want to associate an Elastic IP address with the instance. To associate an Elastic IP address, you must first allocate a new Elastic IP

can associate the Elastic IP address with an EC2 instance.

By default, all AWS accounts are limited to five (5) Elastic IP addresses per Region because public (IPv4) internet addresses are a scarce public resource. However, this is a soft limit, and you can request a limit increase (which might be approved).

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 32the instance. To access it in a browser, go to the following URL: 

EC2 instance metadata

programmatically, such as from a terminal window that has the cURL utility. In the terminal window, run curl http://169.254.169.254/latest/meta-data/to retrieve

• Instance metadata is data about your instance.

• While you are connected to the instance, you can view it – • In a browser: http://169.254.169.254/latest/meta-data/ • In a terminal window: curl http://169.254.169.254/latest/meta-data/

• Example retrievable values –

• Public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone.

• Any user data specified at instance launch can also be accessed at:

http://169.254.169.254/latest/user-data/ • It can be used to configure or manage a running instance.

• For example, author a configuration script that reads the metadata and uses it to configure applications or OS settings.

Instance metadata is data about your instance. You can view it while you are connected to http://169.254.169.254/latest/meta-data/. The data can also be read

it. The IP address 169.254.169.254 is a link-local address and it is valid only from the instance.

Instance metadata provides much of the same information about the running instance that you can find in the AWS Management Console. For example, you can discover the public IP address, private IP address, public hostname, instance ID, security groups, Region, Availability Zone, and more.

Any user data that is specified at instance launch can also be accessed at the following URL: http://169.254.169.254/latest/user-data.

EC2 instance metadata can be used to configure or manage a running instance. For example, you can author a configuration script that accesses the metadata information and uses it to configure applications or OS settings.

Amazon CloudWatch for monitoring By default, Amazon EC2 provides basic monitoring, which sends metric data to CloudWatch 

• Use Amazon CloudWatch to monitor EC2 instances

• Provides near-real-time metrics

Amazon CloudWatch Instance with CloudWatch

• Provides charts in the Amazon EC2 console Monitoring

tab that you can view

• Maintains 15 months of historical data

• Basic monitoring

• Default, no additional cost

• Metric data sent to CloudWatch every 5 minutes

• Detailed monitoring

• Fixed monthly rate for seven pre-selected metrics

• Metric data delivered every 1 minute

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 33

You can monitor your instances by using Amazon CloudWatch, which collects and processes raw data from Amazon EC2 into readable, near-real-time metrics. These statistics are recorded for a period of 15 months, so you can access historical information and gain a better perspective on how your web application or service is performing.

in 5-minute periods. To send metric data for your instance to CloudWatch in 1-minute periods, you can enable detailed monitoring on the instance. For more information, see Enable or Disable Detailed Monitoring for Your Instances.

The Amazon EC2 console displays a series of graphs based on the raw data from Amazon CloudWatch. Depending on your needs, you might prefer to get data for your instances from Amazon CloudWatch instead of through the graphs in the console. By default, Amazon CloudWatch does not provide RAM metrics for EC2 instances, though that is an option that you can configure if you want to CloudWatch to collect that data.

Some key takeaways from this section of the module include:

Section 2 key takeaways

34

• Amazon EC2 enables you to run Windows and Linux virtual machines in the cloud.

• You launch EC2 instances from an AMI template into a VPC in your account.

• You can choose from many instance types. Each instance type offers different combinations of CPU, RAM, storage, and networking capabilities.

• You can configure security groups to control access to instances (specify allowed ports and source).

• User data enables you to specify a script to run the first time that an instance launches.

• Only instances that are backed by Amazon EBS can be stopped.

• You can use Amazon CloudWatch to capture and review metrics on EC2 instances.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

• Amazon EC2 enables you to run Windows and Linux virtual machines in the cloud. • You launch EC2 instances from an AMI template into a VPC in your account. • You can choose from many instance types. Each instance type offers different combinations of CPU, RAM, storage, and networking capabilities.

• You can configure security groups to control access to instances (specify allowed ports and source).

• User data enables you to specify a script to run the first time that an instance launches. • Only instances that are backed by Amazon EBS can be stopped.

• You can use Amazon CloudWatch to capture and review metrics on EC2 instances.



• How to use the AWS Management Console to launch an Amazon EC2 instance (with all the 

Recorded

Amazon EC2

• How to connect to the Windows ~~instance b~~y using a Remote Desktop client and the key demonstration

35

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Now, take a moment to watch the EC2 Demo. The recording runs just over 3 minutes and reinforces some of the concepts that were discussed in this section of the module.

The demonstration shows:

default instance settings accepted).

pair that was identified during instance launch to decrypt the Windows password for login. • How to terminate the instance after it is no longer needed.



Lab 3:

Introduction to 

Amazon EC2



36

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introducing Lab 3: Introduction to Amazon EC2. This lab provides hands-on practice with launching, resizing, managing, and monitoring an Amazon EC2 instance.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 37Public subnet 

Lab 3 scenario

In this lab, you will launch and configure your first virtual machine that runs on Amazon EC2.

AWS Cloud

Region

Availability Zone 1

Lab VPC

Web server

instance

Introducing Lab 3: Introduction to Amazon EC2.

In this lab, you will launch and configure a virtual machine that runs on Amazon EC2.

• Task 1 – Launch Your Amazon EC2 Instance • Task 2 – Monitor Your Instance 

• Task 3 – Update Your Security Group and Access the Web Server • Task 4 – Resize Your Instance: Instance Type and EBS Volume • Task 5 – Explore EC2 Limits

• Task 6 – Test Termination Protection

Lab 3: Tasks

• Resize Your Instance: Instance Type and EBS Volume

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 38

In this hands-on lab, you will:

• Launch Your Amazon EC2 Instance

• Monitor Your Instance

• Update Your Security Group and Access the Web Server

• Explore EC2 Limits

• Test Termination Protection

1. Launched an instance that is configured as a 

2. Viewed the instance system log

3. Reconfigured a security group

4. Modified the instance type and root volume

1. Launched an instance that is configure~~d as a web server~~

Lab 3: Final product

2. Viewed the instance system log

Amazon EC2

3. Reconfigured a security group

By the end of the lab, you will have:

4. Modified the instance type and root volume siz~~e~~

web server size

VPC 

AMI

Security

group

t2.micro

t2.small

instance

instance

Amazon Elastic 

Block Store

(Amazon EBS)

10-GB root

8-GB root

volume

volume

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 39By the end of the lab, you will have:

Begin Lab 1: Introduction to AWS IAM

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 40Begin Lab 3: Introduction to 

~ 35 minutes

Amazon EC2

It is now time to start the lab.



Lab debrief:

Key takeaways

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 41

The instructor will lead a conversation about the key takeaways from the lab after you have completed it.

Amazon EC2 versus using a managed service like Amazon Relational Database Service  

Activity: Amazon EC2

Photo by Pixabay from Pexels.

42

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

In this educator-led activity, you will discuss the advantages and disadvantages of using (Amazon RDS).

Activity: Gather information Amazon EC2 Amazon RDS

Amazon EC2 by using the AWS Quick Start – SQL Server R~~efere~~nce Architecturedeployment. 

AWS Cloud

Availability Zone 1 Availability Zone 2

MS SQL Server

MS SQL Server secondary

primary DB instance

DB instance

Always-on

mirroring

Volume Volume

Volume

Volume

replica

replica

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 43

The objective of this activity is to demonstrate that you understand the differences between building a deployment that uses Amazon EC2 and using a fully managed service, such as Amazon RDS, to deploy your solution. At the end of this activity, you should be prepared to discuss the advantages and disadvantages of deploying Microsoft SQL Server on Amazon EC2 versus deploying it on Amazon RDS.

The educator will ask you to:

1. Watch an 8-minute video that explains the benefits of deploying Microsoft SQL Server on You are encouraged to take notes.

2. Read a blog post about the benefits of running Microsoft SQL Server on Amazon RDS. You are again encouraged to take notes.

3. Participate in the class conversation about the questions posed on the next slide.

1. Between Amazon EC2 or Amazon RDS, which provides a managed service? What does managed service 2. Name at least one advantage of deploying Microsoft SQL Server on Amazon EC2 instead of Amazon RDS. 3. What advantage does the Quick Start provide over a manual installation on Amazon EC2? 4. Which deployment option offers the best approach for all use cases? 

5. Which approach costs more: using Amazon EC2 or using Amazon RDS?

Activity: Check your understanding

mean?

• ANSWER: Amazon RDS provides a managed service. Amazon RDS handles provisioning, installation and patching, • Instance – Standard (Single-AZ) instance

automated backups, restoring snapshots from points in time, high availability, and monitoring. • Instance size – db.m5.large

• Region – US East (Ohio)

• ANSWER: Amazon EC2 offers complete control over every configuration, the OS, and the software stack. • Pricing – On-Demand Instance

• ANSWER: The Quick Start is a reference architecture with proven best practices built into the design. • Instance – Windows instance

• ANSWER: Neither. The correct deployment option depends on your specific needs.

• Instance size – m5.large

• Region – US East (Ohio)

• ANSWER: It depends. Managing the database deployment on Amazon EC2 requires more customer oversight and time. If time is your priority, then Amazon RDS might be less expensive. If you have in-house expertise,

• Pricing – On-Demand Instance

Amazon EC2 might be more cost-effective.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 44

The educator will lead the class in a conversation as each question is revealed. Then, the educator will display the written suggested responses and you can discuss these points further.

Regarding question 5, the answer was based on the information that is listed on the AWS Pricing pages as of October, 2019.

• For Amazon RDS, you pay $0.977 per hour if you run Microsoft SQL Server based on these parameters:

• For Amazon EC2, you pay $0.668 per hour if you run Microsoft SQL Server based on these parameters:

As you consider cost, do not forget to include the cost of labor. For example, keep in mind that with a standard Single-AZ Amazon RDS deployment—which is the basis of the example price reference—automated backups are provided. With Amazon RDS, if a DB instance component failed and a user-initiated restore operation is required, you would have a

restorable backup that you could use. If you run the database on Amazon EC2, you could configure an equally robust backup procedure for Microsoft SQL Server. However, it would take time, knowledge, and technical skill to build the solution. You would also need to pre configure the solution before you encounter the situation where you need it. For these reasons, when you consider the needs of your deployments holistically, you might find that

could use Amazon EC2. In this case, you might find Amazon EC2 to be the more cost-effective

using Amazon RDS is less expensive than using Amazon EC2. However, if you have skilled database administrators on staff—and you also have very specific deployment requirements that make it preferable for you to have total control over all aspects of the deployment—you

solution.

M o d u l e 6 : C o m p u t e

Section 3: Amazon EC2 cost optimization

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. Introducing Section 3: Amazon EC2 cost optimization.

Amazon EC2 pricing models On-Demand Instances

Reserved Instances

Spot Instances

• Pay by the hour

• Full, partial, or no upfront payment

• Instances run as long as they are

for instance you reserve.

• No long-term commitments.

available and your bid is above the Spot Instance price.

• Discount on hourly charge for that

• Eligible for the AWS Free Tier.

instance.

• They can be interrupted by AWS with a 2-minute notification.

• 1-year or 3-year term.

Dedicated Hosts • A physical server with EC2 instance

• Interruption options include terminated, stopped or hibernated.

Scheduled Reserved

capacity fully dedicated to your use.

• Prices can be significantly less

Instances

expensive compared to On-Demand

Instances

Dedicated Instances

• Purchase a capacity reservation that is

• Good choice when you have flexibility

always available on a recurring

schedule you specify.

• Instances that run in a VPC on hardware

that is dedicated to a single customer.

• 1-year term.

in when your applications can run.

Per second billing available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

46

Amazon offers different pricing models to choose from when you want to run EC2 instances.

Per second billing is only available for On-Demand Instances, Reserved Instances, and Spot Instances that run Amazon Linux or Ubuntu.

On-Demand Instances are eligible for the AWS Free Tier. They have the lowest upfront cost and the most flexibility. There are no upfront commitments or long-term contracts. It is a good choice for applications with short-term, spiky, or unpredictable workloads.

Dedicated Hosts are physical servers with instance capacity that is dedicated to your use. They enable you to use your existing per-socket, per-core, or per-VM software licenses, such as for Microsoft Windows or Microsoft SQL Server.

Dedicated Instances are instances that run in a virtual private cloud (VPC) on hardware that’s dedicated to a single customer. They are physically isolated at the host hardware level from instances that belong to other AWS accounts.

Reserved Instance enable you to reserve computing capacity for 1-year or 3-year term with lower hourly running costs. The discounted usage price is fixed for as long as you own the Reserved Instance. If you expect consistent, heavy use, they can provide substantial savings

compared to On-Demand Instances.

Scheduled Reserved Instances enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified duration, for a 1-year term. You pay for the time that the instances are scheduled, even if you do not use them.

Spot Instances enable you to bid on unused EC2 instances, which can lower your costs. The hourly price for a Spot Instance fluctuates depending on supply and demand. Your Spot Instance runs whenever your bid exceeds the current market price.

Spot Instances Reserved Instances Dedicated HostsPredictability ensures  

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 47

Amazon EC2 pricing models: Benefits

| On-Demand Instances |  |  |  |
| --- | --- | --- | --- |
| •Low cost and flexibility | •Large scale, dynamic workload | •  compute capacity is  available when needed | •Save money on licensing  costs    •Help meet compliance and regulatory  requirements |

want to run on Amazon EC2, or when you have specific compliance or regulatory

Each Amazon EC2 pricing model provides a different set of benefits.

On-Demand Instances offer the most flexibility, with no long-term contract and low rates. Spot Instances provide large scale at a significantly discounted price.

Reserved Instances are a good choice if you have predictable or steady-state compute needs (for example, an instance that you know you want to keep running most or all of the time for months or years).

Dedicated Hosts are a good choice when you have licensing restrictions for the software you requirements that preclude you from using the other deployment options.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 48Spiky Workloads Steady-State Workloads Time-Insensitive 

Spot Instances Reserved Instances Dedicated Hosts

Amazon EC2 pricing models: Use cases an ~~application for a short time (for example, during application development or testing).~~

Highly Sensitive

Workloads

Workloads

| On-Demand Instances |  |  |  |
| --- | --- | --- | --- |
| •Short-term, spiky, or  unpredictable workloads  •Application development or testing | •Applications with flexible  start and end times    •Applications only feasible at  very low compute prices    •Users with urgent computing needs for large amounts of additional capacity | •Steady state or predictable  usage workloads    •Applications that require  reserved capacity, including  disaster recovery    •Users able to make upfront  payments to reduce total  computing costs even further | •Bring your own license    (BYOL)    •Compliance and regulatory  restrictions      •Usage and licensing tracking  •Control instance placement |

Here is a review of some use cases for the various pricing options.

On-Demand Instance pricing works well for spiky workloads or if you only need to test or run

Sometimes, your workloads are unpredictable, and On-Demand Instances are a good choice for these cases.

Spot Instances are a good choice if your applications can tolerate interruption with a 2- minute warning notification. By default, instances are terminated, but you can configure them to stop or hibernate instead. Common use cases include fault-tolerant applications such as web servers, API backends, and big data processing. Workloads that constantly save data to persistent storage (such as Amazon S3) are also good candidates.

Reserved Instances are a good choice when you have long-term workloads with predictable usage patterns, such as servers that you know you will want to run in a consistent way over many months.

Dedicated Hosts are a good choice when you have existing per-socket, per-core, or per-VMsoftware licenses, or when you must address specific corporate compliance and regulatory requirements.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 49Right size Optimal • Right-size ~~– Choose the right balance of instance types. Notice when servers can~~ be either 

The four pillars of cost optimization

• Increase elasticity ~~– Design your deployments to reduce the amount of server ca~~pacity

Cost Optimization

that is i~~dle by implementing deployments that are elastic, such as deployments t~~hat use automa~~tic scaling to handle peak loads.~~

• Optimal pricing model ~~– Recognize the available pricing options. Analyze your u~~sage

Increase

Optimize

pricing

elasticity

storage

• Optimize storage choices ~~– Analyze the storage requirements of your deployme~~nts.

model

choices

To optimize costs, you must consider four consistent, powerful drivers:

sized down or turned off, and still meet your performance requirements.

patterns so that you can run EC2 instances with the right mix of pricing options.

Reduce unused storage overhead when possible, and choose less expensive storage options if they can still meet your requirements for storage performance.

Pillar 1: Right size 

both a technical perspective and a cost perspective. Right-sizing is the process of reviewing

Pillars:

Provision instances to match the need



• CPU, memory, storage, and network throughput

1. Right size

• Select appropriate instance types for your use

2. Increase elasticity

3. Optimal pricing model

Use Amazon CloudWatch metrics

4. Optimize storage choices

environment, and then test your application on those different test deployments to 

• How idle are instances? When?

• Downsize instances

Best practice: Right size, then reserve

• Use Amazon CloudWatch metrics and set up custom metrics. A metric represents a time ordered set of values that are published to CloudWatch (for example, the CPU usage of a

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 50

First, consider right-sizing. AWS offers approximately 60 instance types and sizes. The wide choice of options enables customers to select the instance that best fits their workload. It can be difficult to know where to start and what instance choice will prove to be the best, from

deployed resources and looking for opportunities to downsize when possible. To right-size:

• Select the cheapest instance available that still meets your performance requirements.

• Review CPU, RAM, storage, and network utilization to identify instances that could be downsized. You might want to provision a variety of instance types and sizes in a test

identify which instances offer the best performance-to-cost ratio. For right-sizing, use techniques such as load testing to your advantage.

particular EC2 instance). Data points can come from any application or business activity for which you collect data.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 51Stop or hibernate Amazon EBS-backed instances One form of elasticity is to create, start, or use EC2 instances when they are needed, but then to turn them off when they are not in use. Elasticity is one of the central tenets of the 

Pillar 2: Increase elasticity

Pillars:

development workloads, or test workloads. For example, if you run development or test

that are not actively in use

1. Right-Size

• Example: non-production development or test instances



hours and thus reduce runtime costs by perhaps 65 percent. The concept is similar to why

2. Increase Elasticity

3. Optimal pricing model

Use automatic scaling to match needs based on

4. Optimize storage choices

usage

• Automated and time-based elasticity

As a rule of thumb, you should target 20–30 percent of your Amazon EC2 instances to run as On-Demand Instances or Spot Instances, and you should also actively look for ways to

cloud, but customers often go through a learning process to operationalize elasticity to drive cost savings.

The easiest way for large customers to embrace elasticity is to look for resources that look like good candidates for stopping or hibernating, such as non-production environments,

workloads in a single time zone, you can easily turn off those instances outside of business

there is a light switch next to the door, and why most offices encourage employees to turn off the lights on their way out of the office each night.

For production workloads, configuring more precise and granular automatic scaling policies can help you take advantage of horizontal scaling to meet peak capacity needs and to not pay for peak capacity all the time.

maximize elasticity.

• Use On-Demand Instance and Spot Instances for variable 

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 52Leverage the right pricing model for your use case• Consider your usage patterns

Pillar 3: Optimal pricing model

Pillars:

1. Right-Size

2. Increase Elasticity

Optimize and combine purchase types

4. Optimize storage choices

3. Optimal pricing model

Examples:

workloads

• Use Reserved Instances for predictable workloads

Consider serverless solutions (AWS Lambda)

AWS provides a number of pricing models for Amazon EC2 to help customers save money. The models available were discussed in detail earlier in this module. Customers can combine multiple purchase types to optimize pricing based on their current and forecast capacity needs.

Customers are also encouraged to consider their application architecture. For example, does the functionality provided by your application need to run on an EC2 virtual machine? Perhaps by making use of the AWS Lambda service instead, you could significantly decrease your costs.

AWS Lambda is discussed later in this module.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 53Reduce costs while maintaining storage performance and Pillar 4: Optimize storage choices 

Pillars:

availability

1. Right-Size

Resize EBS volumes

2. Increase Elasticity

3. Optimal pricing model

Change EBS volume types

4. Optimize storage choices

 Can you meet performance requirements with less expensive storage?

 Example: Amazon EBS Throughput Optimized HDD (st1) storage typically 

costs half as much as the default General Purpose SSD (gp2) storage option.

Customers often use EBS snapshots to create data backups. However, some customers forget

Delete EBS snapshots that are no longer needed

Identify the most appropriate destination for specific types of

data

 Does the application need the instance to reside on Amazon EBS?

 Amazon S3 storage options with lifecycle policies can reduce costs

reduce costs. For example, you might automate the migration of older infrequently accessed data to cheaper storage locations, such as Amazon Simple Storage Service Glacier.

Customers can also reduce storage costs. When you launch EC2 instances, different instance types offer different storage options. It is a best practice to try to reduce costs while also maintaining storage performance and availability.

One way you can accomplish this is by resizing EBS volumes. For example, if you originally provisioned a 500-GB volume for an EC2 instance that will only need a maximum of 20 GB of storage space, you can reduce the size of the volume and save on costs.

There are also a variety of EBS volume types. Choose the least expensive type that still meets your performance requirements. For example, Amazon EBS Throughput Optimized HDD (st1) storage typically costs half as much as the default General Purpose SSD (gp2) storage option. If an st1 drive will meet the needs of your workload, take advantage of the cost savings.

to delete snapshots that are no longer needed. Delete these unneeded snapshots to save on costs.

Finally, try to identify the most appropriate destination for specific types of data. Does your application need the data it uses to reside on Amazon EBS? Would the application run equally as well if it used Amazon S3 for storage instead? Configuring data lifecycle policies can also

Measure, monitor, and improve what purpose. You can activate cost allocation tags in the Billing and Cost Management 

• Cost optimization is an ongoing process.

• Recommendations –

• Define and enforce cost allocation tagging.

• Define metrics, set targets, and review regularly.

• Encourage teams to architect for cost.

Use AWS services such as AWS Trusted Advisor, which provides real-time guidance to help

• Assign the responsibility of optimization to an individual or to a

team.

Cost-optimization efforts are typically more successful when the responsibility for cost

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 54

If it is done correctly, cost optimization is not a one-time process that a customer completes. Instead, by routinely measuring and analyzing your systems, you can continually improve and adjust your costs.

Tagging helps provide information about what resources are being used by whomand for

console, and AWS can generate a cost allocation report with usage and costs grouped by your active tags. Apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services.

Encourage teams to architect for cost. AWS Cost Explorer is a free tool that you can use to view graphs of your costs. You can use Cost Explorer to see patterns in how much you spend on AWS resources over time, identify areas that need further inquiry, and see trends that you can use to understand your costs.

you provision resources that follow AWS best practices.

optimization is assigned to an individual or to a team.

Some key takeaways from this section of the module are: 

Section 3 key takeaways

• Amazon EC2 pricing models include On-Demand Instances, Reserved Instances, Spot Instances, Dedicated Instances, and Dedicated Hosts.

• Spot Instances can be interrupted with a 2-minute notification. However, they can offer 

• Spot Instances can be interrupted with a 2-minute

notification. However, they can offer significant cost

• The four pillars of cost optimization are –

savings over On-Demand Instances.

• The four pillars of cost optimizationare:

• Right size

• Increase elasticity

• Optimal pricing model

• Optimize storage choices

55

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

• Amazon EC2 pricing models include On-Demand Instances, Reserved Instances, Spot Instances, Dedicated Instances, and Dedicated Hosts. Per second billing is available for On Demand Instances, Reserved Instances, and Spot Instances that use only Amazon Linux and Ubuntu.

significant cost savings over On-Demand Instances.

• Right size

• Increase elasticity

• Optimal pricing model

• Optimize storage choices

M o d u l e 6 : C o m p u t e

Section 4: Container services

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. Introducing Section 4: Container services.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 57Container basics 

• Containers are a method of operating system virtualization.

• Benefits – • Repeatable.

Your Container Your application

• Self-contained execution environments. • Software runs the same in different environments. 

• Developer's laptop, test, production. • Faster to launch and stop or terminate than

Dependencies Configurations

machines. Spinning up a container happens in hundreds of milliseconds. Thus, by using

Hooks into OS

virtual machines

Containers are a method of operating system virtualization that enables you to run an application and its dependencies in resource-isolated processes. By using containers, you can easily package an application's code, configurations, and dependencies into easy-to-use building blocks that deliver environmental consistency, operational efficiency, developer productivity, and version control.

Containers are smaller than virtual machines, and do not contain an entire operating system. Instead, containers share a virtualized operating system and run as resource-isolated processes, which ensure quick, reliable, and consistent deployments. Containers hold everything that the software needs to run, such as libraries, system tools, code, and the runtime.

Containers deliver environmental consistency because the application’s code, configurations, and dependencies are packaged into a single object.

In terms of space, container images are usually an order of magnitude smaller than virtual containers, you can use a fast, portable, and infrastructure-agnostic execution environment.

Containers can help ensure that applications deploy quickly, reliably, and consistently, regardless of deployment environment. Containers also give you more granular control over resources, which gives your infrastructure improved efficiency.

Docker is a software platform that packages software (such as applications) into containers. 

What is Docker?

By using Docker, you can quickly deploy and scale applications into any environment.

• Docker is a software platform that

Docker is best used as a solution when you want to:

enables you to build, test, and deploy applications quickly.

• You run containers on Docker. • Containers are created from a template called an image.

• A container has everything a

Container

Containers have everything the software needs to run: 

Libraries System

software application needs to run.

tools Code Runtime

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 58

Docker is installed on each server that will host containers, and it provides simple commands that you can use to build, start, or stop containers.

• Standardize environments

• Reduce conflicts between language stacks and versions • Use containers as a service

• Run microservices using standardized code deployments • Require portability for data processing

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 59

Containers versus virtual machines

Three virtual machines on three EC2 instances

Example

VM 1 VM 2 VM 3

Th~~e right of the diagram has a virtual machine (VM)-based deployment~~. Each of the three

Three containers on one EC2 instance

Container

App 2

App 3

App 1

Container

Container

Container

Bins/Libs

Bins/Libs

Bins/Libs

instance 1

instance 2

instance 3

App 2

App 3

App 1

Docker

engine Bins/Libs

EC2

EC2

EC2

instance

instance

Bins/Libs

Bins/Libs

instance

guest OS

guest OS

guest OS

EC2 instance guest OS

in its own container (which provides process isolation), but all the containers run on a single

Hypervisor

Part of

Host operating system

AWS Global

Infrastructure

Physical server

Many people who are first introduced to the concept of a container think that containers are exactly like virtual machines. However, the differences are in the details. One significant difference is that virtual machines run directly on a hypervisor, but containers can run on any Linux OS if they have the appropriate kernel feature support and the Docker daemon is present. This makes containers very portable. Your laptop, your VM, your EC2 instance, and your bare metal server are all potential hosts where you can run a container.

EC2 instances runs directly on the hypervisor that is provided by the AWS Global Infrastructure. Each EC2 instance runs a virtual machine. In this VM-based deployment, each of the three apps runs on its own VM, which provides process isolation.

The left of the diagram has a container-based deployment. There is only one EC2 instance that runs a virtual machine. The Docker engine is installed on the Linux guest OS of the EC2 instance, and there are three containers. In this container-based deployment, each app runs

EC2 instance. The processes that run in the containers communicate directly to the kernel in the Linux guest OS and are largely unaware of their container silo. The Docker engine is present to manage how the containers run on the Linux guest OS, and it also provides essential management functions throughout the container lifecycle.

In an actual container-based deployment, a large EC2 instance could run hundreds of containers.

Container ServiceGiven what you now know about containers, you might think that you could launch one or 

Amazon Elastic Container Service (Amazon

ECS)

• Amazon Elastic Container Service (Amazon ECS) – container management service that supports Docker containers. Amazon ECS enables you to

• A highly scalable, fast, container management service

• Key benefits –

• Orchestrates the execution of Docker containers

Amazon Elastic

• Maintains and scales the fleet of nodes that run your containers

• Removes the complexity of standing up the infrastructure • Monitor container deployment

• Integrated with features that are familiar to Amazon EC2 service users – • Schedule containers by using a built-in scheduler or a third-party scheduler (for example,

• Elastic Load Balancing

Apache Mesos or Blox)

• Amazon EC2 security groups

• Amazon EBS volumes

• IAM roles

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 60

more Amazon EC2 instances, install Docker on each instance, and manage and run the Docker containers on those Amazon EC2 instances yourself. While that is an option, AWS provides a service called Amazon Elastic Container Service (Amazon ECS) that simplifies container management.

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance easily run applications on a managed cluster of Amazon EC2 instances. Essential Amazon ECS features include the ability to:

• Launch up to tens of thousands of Docker containers in seconds • Manage the state of the cluster that runs the containers

Amazon ECS clusters can also use Spot Instances and Reserved Instances.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 61ECS cluster 

Amazon ECS orchestrates containers

EC2 instance

Requests to run containers

x3 x2

Container A

EC2 instance

ECS cluster. The cluster (when you choose the EC2 launch type) consists of a group of EC2

Container B

Amazon Elastic Container Service

(Amazon ECS)

To prepare your application to run on Amazon ECS, you create a task definitionwhich is a text file that describes one or more containers, up to a maximum of ten, that form your application. It can be thought of as a blueprint for your application. Task definitions specify parameters for your application, for example which containers to use, which ports should be opened for your application, and what data volumes should be used with the containers in the task.

A task is the instantiation of a task definition within a cluster. You can specify the number of tasks that will run on your cluster. The Amazon ECS task scheduler is responsible for placing tasks within your cluster. A task will run anywhere from one to ten containers, depending on the task definition you defined.

When Amazon ECS runs the containers that make up your task, it places them on an instances each of which is running an Amazon ECS container agent.

Amazon ECS provides multiple scheduling strategies that will place containers across your clusters based on your resource needs (for example, CPU or RAM) and availability requirements.

• Key question: Do you want to manage the Amazon ECS cluster that runs the containers?

Amazon ECS cluster options

• An EC2 Windows + Networking cluster

• If yes, create an Amazon ECS cluster backed by Amazon EC2 (provides more granular control over infrastructure)

instance. In this way, t~~he EC2 launch type provides more granular con~~tr~~ol o~~ver the

• If no, create an Amazon ECS cluster backed by AWS Fargate (easier to maintain, focus on your applications)

infrastructure that run~~s your container applications because you man~~ag~~e t~~he EC2 instances

Containers

Amazon ECS cluster

backed by FargateAmazon ECS cluster Amazon ECS keeps tra~~ck of all the CPU, memory, and other resources~~ in ~~yo~~ur cluster. Amazon

Container instance

Container instance

Container instance

backed by Amazon

1

2

3

ECS also finds the bes~~t server for your container on based on your sp~~ec~~ifie~~d resource

EC2

App 1

App 2

App 3

Bins/Libs

Bins/Libs

You manage

Bins/Libs

If you choose the networking-only Fargate launch type, then the cluster that will run your

You manage

Docker engines (one per OS in the cluster)

AWSmanages

VM guest operating systems in the Amazon ECS cluster

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 62

When you create an Amazon ECS cluster, you have three options: • A Networking Only cluster (powered by AWS Fargate)

• An EC2 Linux + Networking cluster

If you choose one of the two EC2 launch type options, you will then be prompted to choose whether the cluster EC2 instances will run as On-Demand Instances or Spot Instances. In addition, you will need to specify many details about the EC2 instances that will make up your cluster—the same details that you must specify when you launch a stand lone EC2

that make up the cluster.

requirements.

containers will be managed by AWS. With this option, you only need to package your application in containers, specify the CPU and memory requirements, define networking and IAM policies, and launch the application. You do not need to provision, configure, or scale the cluster. It removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing. The Fargate option enables you to focus on designing and building your applications.

Kubernetes is open source software for container orchestration. Kubernetes can work with many containerization technologies, including Docker. Because it is a popular open source 

What is Kubernetes?

Kubernetes enables you to deploy and manage containerized applications at scale. With

• Kubernetes is open source software for container orchestration. Kubernetes, you can run any type of containerized application by using the same toolset in

• Deploy and manage containerized applications at scale.

• The same toolset can be used on premises and in the cloud.

instances (called nodes). It runs containers on the cluster, which are based on where

• Complements Docker.

• Docker enables you to run multiple containers on a single OS host.

Containers are run in logical groupings called pods. You can run and scale one or many

• Kubernetes orchestrates multiple Docker hosts (nodes).

• Automates –

• Container provisioning.

• Networking.

• Load distribution.

• Scaling.

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 63

project, a large community of developers and companies build extensions, integrations, and plugins that keep the software relevant, and new and in-demand features are added frequently.

both on-premises data centers and the cloud. Kubernetes manages a cluster of compute compute resources are available and the resource requirements of each container.

containers together as a pod. Each pod is given an IP address and a single Domain Name System (DNS) name, which Kubernetes uses to connect your services with each other and external traffic.

A key advantage of Kubernetes is that you can use it to run your containerized applications anywhere without needing to change your operational tooling. For example, applications can be moved from local on-premises development machines to production deployments in the cloud by using the same operational tooling.

Kubernetes ServiceYou might think that you could launch one or more Amazon EC2 instances, install Docker on 

Amazon Elastic Kubernetes Service (Amazon

EKS)

• Amazon Elastic Kubernetes Service (Amazon EKS)

• Enables you to run Kubernetes on AWS

• Certified Kubernetes conformant (supports easy migration)

• Supports Linux and Windows containers

Amazon Elastic

• Compatible with Kubernetes community tools and supports

popular Kubernetes add-ons

• Use Amazon EKS to –

scale, reliability, and availability of the AWS Cloud, which includes AWS networking and

• Manage clusters of Amazon EC2 compute instances

security services like Application Load Balancers for load distribution, IAM for role-based

• Run containers that are orchestrated by Kubernetes on those

instances

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 64

each instance, install Kubernetes on the cluster, and manage and run Kubernetes yourself. While that is an option, AWS provides a service called Amazon Elastic Kubernetes Service (Amazon EKS) that simplifies the management of Kubernetes clusters.

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service that makes it easy for you to run Kubernetes on AWS without needing to install, operate, and maintain your own Kubernetes control plane. It is certified Kubernetes conformant, so existing applications that run on upstream Kubernetes are compatible with Amazon EKS.

Amazon EKS automatically manages the availability and scalability of the cluster nodes that are responsible for starting and stopping containers, scheduling containers on virtual machines, storing cluster data, and other tasks. It automatically detects and replaces unhealthy control plane nodes for each cluster. You can take advantage of the performance,

access control, and VPC for pod networking.

You may be wondering why Amazon offers both Amazon ECS and Amazon EKS, since they are both capable of orchestrating Docker containers. The reason that both services exist is to provide customers with flexible options. You can decide which option best matches your needs.

Amazon Elastic Container Registry (Amazon ECR) is a fully managed Docker container 

Amazon Elastic Container Registry (Amazon

ECR)

Amazon E~~CR supports Doc~~k~~er Re~~gistry HTTP API version 2, which enables you to interact with

Amazon ECR is a fully managed Docker container registry that

makes it easy for developers to store, manage, and deploy

Docker container images.

Amazon ECS integration

environm~~ent—whether it is in th~~e cloud, on premises, or on your local machine. 

Docker support

Team collaboration

Amazon Elastic

Container Registry Image Registry

Access control

Third-party integrations

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 65© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

registry that makes it easy for developers to store, manage, and deploy Docker container images. It is integrated with Amazon ECS, so you can store, run, and manage container images for applications that run on Amazon ECS. Specify the Amazon ECR repository in your task definition, and Amazon ECS will retrieve the appropriate images for your applications.

Amazon ECR by using Docker CLI commands or your preferred Docker tools. Thus, you can maintain your existing development workflow and access Amazon ECR from any Docker

You can transfer your container images to and from Amazon ECS via HTTPS. Your images are also automatically encrypted at rest using Amazon S3 server-side encryption.

It is also possible to use Amazon ECR images with Amazon EKS. See the Using Amazon ECR Images with Amazon EKS documentation for details.

66 © 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. Some key takeaways from this section include:

Section 4 key takeaways

• Containers can hold everything that an application needs to run.

• Docker is a software platform that packages software into containers.

• A single application can span multiple containers.

• Amazon Elastic Container Service (Amazon ECS) orchestrates the execution of Docker containers.

• Kubernetes is open source software for container orchestration.

• Amazon Elastic Kubernetes Service (Amazon EKS) enables you to run Kubernetes on AWS

• Amazon Elastic Container Registry (Amazon ECR) enables you to store, manage, and deploy your Docker containers.

• Containers can hold everything that an application needs to run. • Docker is a software platform that packages software into containers.

• A single application can span multiple containers.

• Amazon Elastic Container Service (Amazon ECS) orchestrates the execution of Docker containers.

• Kubernetes is open source software for container orchestration.

• Amazon Elastic Kubernetes Service (Amazon EKS) enables you to run Kubernetes on AWS• Amazon Elastic Container Registry (Amazon ECR) enables you to store, manage, and deploy your Docker containers.

M o d u l e 6 : C o m p u t e

Section 5: Introduction to AWS Lambda

© 2019 Amazon Web Services, Inc. or its Affiliates. All rights reserved. 67Introducing Section 5: Introduction to AWS Lambda.