

## AWS Cloud foundation Report

Ruojia Zhang

## Module summary

### **Module1**

This is the first module of the training, and it briefly introduced the cloud computing, Advantages of cloud computing, Amazon Web Services and AWS Cloud Adoption Framework to me. I have got the idea of how cloud computing works and how it charges me for using it. Also I learned how to migrate current project to cloud computing in a very easy way.

### **Module2**

This is the second module of the training. It mainly discussed the Fundamentals of pricing, Total Cost of Ownership, AWS Organizations, AWS Billing and Cost Management and How Technical Support works. Also it introduced the AWS Pricing Calculator and Support plans scavenger hunt to me. I'm now clear how to calculate precise price I'll need to pay for a workload.

### **Module3**

In this third module of the training, the narrator mainly discussed the AWS Global Infrastructure and AWS service and service category overview to me. Also, he has shown the AWS Management Console and introduced the basic component of it to me. Now I have the idea of how AWS arranges its resources in a global perspective. And I have certain knowledge of how to plan my own cloud distribution.

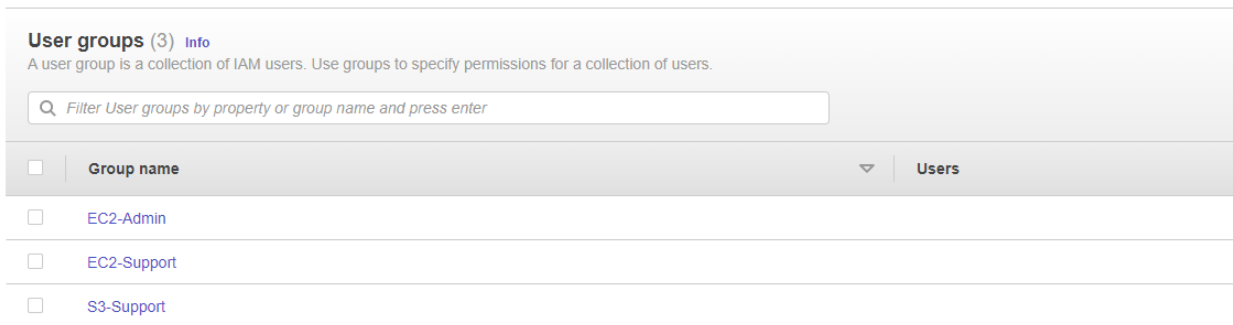
### **Module4**

This is the fourth module of the training and it mainly discussed AWS Cloud Security including the topics of AWS shared responsibility model, AWS IAM, Securing a new AWS account, Securing account, Securing data on AWS and Working to ensure compliance. The

narrator also shown a demonstration of IAM to me. Now I have the idea of how security on cloud works and what are the best practices in different scenarios.

### LAB1

Through LAB1, I have obtained a basic understanding of how IAM works. And how to add/drop/modify different IAM user using the IAM admin account. Also, I've learned about different roles in IAM and policies that enables IAM to work. For example, I can add s3 read/write access explicitly using the master account to other accounts or account groups.



*Figure 1.* IAM creation of LAB1. This has shown all three groups that will need to be operated.

Users

Permissions

Access Advisor

### Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

☐ Policy name [↗](#)

☐ [EC2-Admin-Policy](#)

#### EC2-Admin-Policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Action": [
6         "ec2:Describe*",
7         "ec2:StartInstances",
8         "ec2:StopInstances"
9       ],
10      "Resource": [
11        "*"
12      ],
13      "Effect": "Allow"
14    }
15  ]
16 }
```

*Figure 2.* The policy of EC2 admin that enables the user in this group to list/start instance/ stop instance for ec2.

## **Module5**

In this fifth module of the training, the narrator mainly discussed Networking Basics, Amazon VPC, VPC Networking, VPC security, Route 53 and CloudFront. After learning this module, I have a basic understanding for how to set up a VPC for a software distribution and have learnt the best practice for content delivery using the AWS services.

### **LAB2**

The Lab2 focuses on building my VPC and launch a Web server inside the EC2 Using the settings provided by VPC. Through this lab, I have a deep understanding of how VPC works and what I should do to make a functioning VPC for website hosting. Also, I have got the skill to launch a EC2 instance and use it as a web server.

EC2 > Instances > Launch an instance

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags [Info](#)

Name

[Add additional tags](#)

### ▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

**Quick Start**

Amazon Linux  
aws

macOS  
Mac

Ubuntu  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

S

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-026b57f3c383c2eec (64-bit (x86)) / ami-0636eac5d73e0e5d7 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220912.1 x86\_64 HVM gp2

Architecture

64-bit (x86)

AMI ID

ami-026b57f3c383c2eec Verified provider

### ▼ Summary

Number of instances [Info](#)

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)  
ami-026b57f3c383c2eec

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#)

Figure 3. The setup page in the lab to launch a ec2 instance for the web server.

```

[ec2-user@ip-172-31-22-92 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-22-92 ~]$ systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-10-09 02:27:28 UTC; 2s ago
     Docs: man:httpd.service(8)
  Main PID: 3803 (httpd)
    Status: "Processing requests..."
    CGroup: /system.slice/httpd.service
            └─3803 /usr/sbin/httpd -DFOREGROUND
              └─3804 /usr/sbin/httpd -DFOREGROUND
                └─3805 /usr/sbin/httpd -DFOREGROUND
                  └─3806 /usr/sbin/httpd -DFOREGROUND
                    └─3807 /usr/sbin/httpd -DFOREGROUND
                      └─3808 /usr/sbin/httpd -DFOREGROUND

Oct 09 02:27:28 ip-172-31-22-92.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 09 02:27:28 ip-172-31-22-92.ec2.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-22-92 ~]$

```

*Figure 4.* The httpd service is up and running in the EC2 instance and is able to provide web hosting services.

## Module6

In this module 6, the narrator mainly discussed the topics of different compute engines on AWS including AWS EC2, AWS Lambda, AWS Beanstalk and their best use case and why they suit for certain scenarios. Also I've learnt how to optimize the cost of an existing AWS system by migrating constant running EC2 instances to AWS Lambda or spot instances for a lower cost.

## Lab3

In the Lab3, I need to setup an EC2 server using the given instance type, key pair and then launch the instance with these configuration. Secondly, I'll need to monitor the instance and update the security group and then access the web server afterwards. Finally, I've learnt to reduce the instance type and resize the EBS Volume after that. The server runs normal after resizing.

Instance ID i-038b1c4442b0ce99c (WEEB server)	Public IPv4 address 52.90.126.115   <a href="#">open address</a>
IPv6 address -	Instance state ⌚ Pending
Hostname type IP name: ip-172-31-27-35.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-27-35.ec2.internal
Answer private resource DNS name IPv4 (A)	Instance type t2.small
Auto-assigned IP address 52.90.126.115 [Public IP]	VPC ID vpc-081e97e04716e7ad2
IAM Role -	Subnet ID subnet-0a163d75318a0c21b

*Figure 5.* The server is being resized into t2.small and restarting after resizing.

```

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-27-35 ~]$ ls
[ec2-user@ip-172-31-27-35 ~]$ df

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
devtmpfs	1000640	0	1000640	0%	/dev
tmpfs	1009792	0	1009792	0%	/dev/shm
tmpfs	1009792	408	1009384	1%	/run
tmpfs	1009792	0	1009792	0%	/sys/fs/cgroup
/dev/xvda1	8376300	1620428	6755872	20%	/
tmpfs	201960	0	201960	0%	/run/user/1000

```

[ec2-user@ip-172-31-27-35 ~]$ █

```

*Figure 6.* The server is up and running and the disk size has changed. All other functions normal.

### Module7

In the module7, the narrator mainly discussed storage in AWS. Topics including AWS EBS, Working with EBS, AWS S3, AWS EFS, AWS S3 Glacier are being discussed. After this module, I have the understanding of different tier of storage inside the AWS. And have an understanding of migrating data from faster layer into colder layer in the aim for cost saving.

### Lab4

In this lab, I'm to set up an SSD as the block device for an EC2 instance. Then attach and mount the volume to my EC2 instance. And create the snapshot of my volume. After that I have learnt to restore the data from the snapshot and restore them into a new volume. Finally, I'm able to mount the volume on to the EC2 instance and the data is the same after restoration.



## Create volume [Info](#)

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

### Volume settings

Volume type [Info](#)

General Purpose SSD (gp2) ▼

Size (GiB) [Info](#)

100

Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS [Info](#)

300 / 3000

Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS.

Throughput (MiB/s) [Info](#)

Not applicable

Availability Zone [Info](#)

us-east-1a ▼

Snapshot ID - optional [Info](#)

Don't create volume from a snapshot ▼



Encryption [Info](#)

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

☐ Encrypt this volume

Figure 7. The panel to set up a general purpose SSD with the size of 100 GiB in us-east-1a zone.

```
[ec2-user@ip-172-31-27-35 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        978M   0  978M   0% /dev
tmpfs           987M   0  987M   0% /dev/shm
tmpfs           987M 408K  986M   1% /run
tmpfs           987M   0  987M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs           198M   0  198M   0% /run/user/1000
[ec2-user@ip-172-31-27-35 ~]$
```

*Figure 8.* The provisioned disk is being attached to the running EC2 instance. And the space is increased.

### **Module8**

This module mainly focuses on the topic of database. The narrator covered topics including Amazon RDS, Amazon DynamoDB, Amazon Redshift, Amazon Aurora. And deeply discussed their own best practices and best fit scenarios. Their own advantage and disadvantage are being discussed. After this module, I have an understanding of SaaS and how Amazon manage their database product line.

### **Lab5**

In this Lab, I'm required to first launch an Amazon RDS DB instance with HA and then configure the DB instance to permit connections from my web server and lastly I'll need to open a web application and interact with my pre-provisioned DB. After this lab, I have learnt to setup a basic server-database application. And know that are the possible issue around this system and how to fix them.

## Create database

### Choose a database creation method [Info](#)

☒ **Standard create**

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

☐ **Easy create**

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

### Engine options

#### Engine type [Info](#)

☐ Amazon Aurora



☒ MySQL



☐ MariaDB



☐ PostgreSQL



☐ Oracle



☐ Microsoft SQL Server



#### Edition

☒ MySQL Community



**Known issues/limitations**

Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

#### Version

MySQL 8.0.28



Figure 9. The panel for creating the MySQL database for this LAB

**Security Groups (5)** [Info](#)

Q Filter security groups

<input type="checkbox"/>	Name ▾	Security group ID ▾	Security group name ▾	VPC ID ▾	Description ▾
<input type="checkbox"/>	-	sg-08382f6144ecaa3e9	default	vpc-04b57af9274676794 <a href="#">🔗</a>	default VPC security gr...
<input type="checkbox"/>	-	sg-01d38fb8cd9157a70	WorkEc2SecurityGroup	vpc-0ececdfd6880a57ba <a href="#">🔗</a>	VPC Security Group
<input type="checkbox"/>	Web Security Group	sg-093f38b102ff8aeaa	Web Security Group	vpc-0291bff25cbd2a02a <a href="#">🔗</a>	Enable HTTP access
<input type="checkbox"/>	-	sg-049e1f841d72c5bff	default	vpc-0291bff25cbd2a02a <a href="#">🔗</a>	default VPC security gr...
<input type="checkbox"/>	-	sg-08bffa8089964aa93	default	vpc-0ececdfd6880a57ba <a href="#">🔗</a>	default VPC security gr...

*Figure 10.* The security groups that are defined to run the DB and server

## Module9

The module 9 is talking about Cloud Architecture. The narrator introduces the concept of 5 pillar for best practice. And has covered topics including AWS well-architected frameworks design principles. Operational Excellence, Security, Reliability, Performance efficiency, Cost Optimization, and AWS trusted advisor. I have a deep understanding of how to optimize an existing cloud computing project after this module.

## Module10

The last module of this course is about Auto Scaling and monitoring. The narrator has covered the topics including Elastic Load Balancing, Amazon CloudWatch, Amazon EC2 Auto Scaling. Also, a lab is included in this module to try out Scale and Load balance on my own. After this module, I know how ELB can work with CloudWatch and triggers automatically scaling.

## Lab6

In this lab, I'm required to create an AMI from the currently running instance and then create a load balancer, create a launch configuration and an Auto Scaling group. Then Automatically scale new instances with in a private subnet. Finally, I will create Amazon

CloudWatch alarms and monitor performance of my infrastructure. After this lab, I know how to auto scale and monitor my own cloud computing project.

EC2 > Instances > i-09ab3f8e2fbc583dd > Create image

### Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID  
i-09ab3f8e2fbc583dd (Web Server 1)

Image name  
WebServerAMI  
Maximum 127 characters. Can't be modified after creation.

Image description - *optional*  
Lab AMI for Web Server  
Maximum 255 characters

No reboot  
☐ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - *optional*  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

No tags associated with the resource.

[Add new tag](#)  
You can add up to 50 more tags.

[Cancel](#) [Create image](#)

Figure 11. The create image panel for creating an image using the existing running server

## Specify group details


Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

### Basic configuration

Settings in this section cannot be changed after the target group is created.

#### Choose a target type

☒ Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#)  to manage and scale your EC2 capacity.

☐ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

#### Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

#### Protocol

#### Port

#### VPC

Select the VPC with the instances that you want to include in the target group.

-  
vpc-0e44ec46237bea786  
IPv4: 172.31.0.0/16

#### Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2

Figure 12. The create Load Balancer panel for creating an Load Balancer

## Course Completion Screen shot

