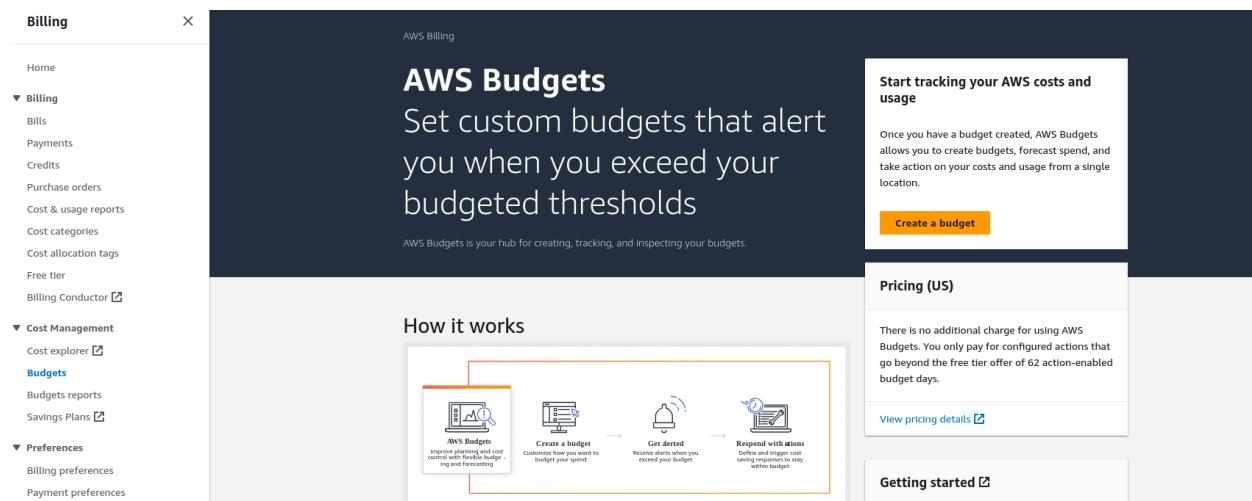


EC2 - Elastic Cloud Compute

Setting up billing alerts

- This is done to prevent from incurring unexpected charges on services accidentally.
- You can choose from multiple existing templates such as zero-cost or monthly budget, etc.



Choose budget type Info

Budget setup

Use a template (simplified)

Use the recommended configurations. You can change some configuration options after the budget is created.

Customize (advanced)

Customize a budget to set parameters specific to your use case. You can customize the time period, the start month, and specific accounts.

Templates - new

Choose a template that best matches your use case.

Zero spend budget

Create a budget that notifies you once your spending exceeds \$0.01 which is above the AWS Free Tier limits.

Monthly cost budget

Create a monthly budget that notifies you if you exceed, or are forecasted to exceed, the budget amount.

Daily Savings Plans coverage budget

Create a coverage budget for your Savings Plans that notifies you when you fall below the defined target.

Daily reservation utilization budget

Create a utilization budget for your reservations that notifies you when you fall below the defined target.

Zero spend budget - Template

Budget name

Provide a descriptive name for this budget.

Tanisha's Zero-Spend Budget

Names must be between 1-100 characters.

Email recipients

Specify the email recipients you want to notify when the threshold has exceeded.

Zero spend budget - Template

Budget name

Provide a descriptive name for this budget.

Tanisha's Zero-Spend Budget

Names must be between 1-100 characters.

Email recipients

Specify the email recipients you want to notify when the threshold has exceeded.

tanishaa.work@gmail.com , tanishachandani119@gmail.com

Maximum number of email recipients is 10.

Scope

All AWS services are in scope in this budget.

 You will be notified via email when any spend above \$0.01 is incurred.

▼ Template settings

This template has default configurations that can be changed later. To change any of these settings, see [Custom](#). You can also download this template in [JSON](#).

Cancel

Create budget

⌚ Your budget Tanisha's Zero-Spend Budget has been created successfully. After creating a budget, it can take up to 24 hours to populate all of your spend data.

AWS Billing > Budgets > Overview

Overview

Budgets (1) [Info](#)

Name	Thresholds	Budget	Amount used	Forecasted amount	Current vs. budgeted	Forecasted vs. bud...
Tanisha's Zero-Spend Budget	OK	\$1.00	\$0.00	-	0.00%	-

[Download CSV](#) [Actions](#) [Create budget](#)

EC2 - Introduction

- Elastic Cloud Compute
 - EC2 is one of the services which enable Infrastructure as a Service model for organizations.
 - EC2 famously offers virtual servers but that's not all, it also offers much more, such as:
 - Renting virtual machines (EC2)
 - Storing data on virtual drives (EBS)
 - Distributing load / traffic across machines (ELB)
 - Scaling the services of an application using auto-scaling groups (ASG)
 - The cloud is to be able to rent these compute whenever you need, on demand, and EC2 is just that.
-
- **To configure or create an ec2 instance, you will need to choose or decide the following things:**

EC2 sizing & configuration options

- Operating System (OS): Linux, Windows or Mac OS
- How much compute power & cores (CPU)
- How much random-access memory (RAM)
- How much storage space:
 - Network-attached (EBS & EFS)
 - hardware (EC2 Instance Store)
- Network card: speed of the card, Public IP address
- Firewall rules: security group
- Bootstrap script (configure at first launch): EC2 User Data

- Diving into the last point - bootstrap script: **what exactly is bootstrapping?**
 - Bootstrapping is basically the set of launching commands that are run when a machine starts. Basically, it's only run once, on initialization and never again.
 - We can bootstrap our instances using the **EC2 user data script**.
 - EC2 user data is used to automate boot tasks such as:
 - installing updates
 - installing software
 - downloading commonly used files from the internet
 - anything you wanna add

** it is worth noting that, the more you add to your ec2 user data script, the more your instance will have to do at boot time.
- The EC2 User Data Script runs with the root user privileges. The "root" user is the superuser with the highest level of administrative privileges, which will allow the script to perform actions that require elevated permissions, such as installing software, modifying system configuration, and creating or deleting files in system directories.

EC2 instance types: example

Instance	vCPU	Mem (GiB)	Storage	Network Performance	EBS Bandwidth (Mbps)
t2.micro	1	1	EBS-Only	Low to Moderate	
t2.xlarge	4	16	EBS-Only	Moderate	
c5d.4xlarge	16	32	1 x 400 NVMe SSD	Up to 10 Gbps	4,750
r5.16xlarge	64*	512	EBS Only	20 Gbps	13,600
m5.8xlarge	32	128	EBS Only	10 Gbps	6,800

EC2 Hands-on

- Aim: to launch an ec2 running instance linux
- We'll be launching our first virtual server using the AWS Console
- We'll get a first high-level approach to the various parameters
- We'll see that our web server is launched using EC2 user data
- We'll learn how to start / stop / terminate our instance.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Name and tags'. A 'Name' field contains 'CCP Demo Instance'. Below it is a link 'Add additional tags'.

[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

CCP Demo Instance

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



Search our full catalog including 1000s of application and OS images

Quick Start



[Browse more AMIs](#)

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Free tier eligible

ami-02e94b011299ef128 (64-bit (x86)) / ami-0a31c2002672717b6 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.2.20231030.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86) ▼

AMI ID

ami-
02e94b011299ef128

Verified provider

▼ **Instance type** [Info](#)

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory
Current generation: true	
On-Demand Linux base pricing: 0.0124 USD per Hour	
On-Demand Windows base pricing: 0.017 USD per Hour	
On-Demand RHEL base pricing: 0.0724 USD per Hour	
On-Demand SUSE base pricing: 0.0124 USD per Hour	

All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

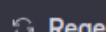
- **The whole key-pair raita (imp to undersand)**
 1. To be able to use SSH for accessing our instance, we need to choose a key pair.

SSH stands for "Secure Shell." It is a network protocol and a method for securely connecting to remote computers over an unsecured network, like the internet. SSH provides a secure way to access and manage a remote computer or server, allowing you to execute commands, transfer files, and perform various administrative tasks.

In terms of AWS (Amazon Web Services), SSH is a crucial tool for managing the virtual machines (EC2 instances) that you run in the cloud. Here's a simple explanation of how it works in AWS:

1. EC2 Instances: In AWS, you can create virtual servers called EC2 instances to run your applications and services.
2. Secure Access: To access these EC2 instances and perform tasks like configuration, software installation, and troubleshooting, you use SSH to establish a secure connection between your local computer and the remote EC2 instance.
3. Key Pairs: AWS uses a key pair system to ensure security. When you create an EC2 instance, you specify a key pair (a public key and a private key). The public key is stored on the EC2 instance, and the private key is kept on your local computer.
4. Authentication: When you want to connect to your EC2 instance using SSH, you provide your private key as authentication. The EC2 instance verifies the private key, and if it matches the public key stored on the instance, the connection is established.
5. Remote Control: Once the SSH connection is established, you can remotely control your EC2 instance by running commands in a terminal on your local computer. This allows you to manage and configure your server as needed.

Overall, SSH in AWS ensures that your interactions with your cloud servers are secure and private, protecting your data and preventing unauthorized access. It's an essential tool for managing your cloud infrastructure.



2. If already exists, choose. Else, create new key pair.

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

 [Create new key pair](#)

3. We need to create one.

Create key pair X

Key pair name

Key pairs allow you to connect to your instance securely.

CCP Demo

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

4. .pem or .ppk

- if you have mac or linux or windows 10, you can use .pem format
- else, go with .ppk and proceed with the puTTy software for SSH authentication.

Private key file format

.pem

For use with OpenSSH

.ppk

For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#) 

Cancel

Create key pair

5. As you create, it will automatically download.
- Leave the n/w settings as is for now.

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Select existing security group

We'll create a new security group called '**launch-wizard-7**' with the following rules:

Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

X

▼ Configure storage Info

[Advanced](#)

1x

8

GiB

gp2

▼

Root volume (Not encrypted)

[Add new volume](#)

0 x File systems

[Edit](#)

- click on advanced

▼ Storage (volumes) [Info](#)

Simple

EBS Volumes

[Hide details](#)

▼ Volume 1 (AMI Root) (Custom)

Storage type [Info](#)

EBS

Snapshot [Info](#)

snap-010813fab9701569d

Device name - *required* [Info](#)

/dev/xvda

Size (GiB) [Info](#)

8

Volume type [Info](#)

gp2

IOPS [Info](#)

100 / 3000

Delete on termination [Info](#)

Yes

Encrypted [Info](#)

Not encrypted

KMS key [Info](#)

Select

KMS keys are only applicable when encryption is set on this volume.

[Add new volume](#)

- the delete on termination is set to **yes** which implies that once you terminate your EC2 instance, your EBS volume will be deleted too. this basically helps ensure you don't incur any unexpected charges.
- Explain advanced details and let every setting be as it is till you get to the user data section. This is the **user data script** we discussed earlier which is for bootstrapping your instance. So, in here, you write your code or upload your code file.

User data - *optional* | [Info](#)

Upload a file with your user data or enter it in the field.

 [Choose file](#)

```
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hiii I am ${hostname} -f} today we are leaning about AWS EC2</h1>" >
/var/www/html/index.html
```

User data has already been base64 encoded

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.2.2...[read more](#)

ami-02e94b011299ef128

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

[Cancel](#)

[Launch instance](#)

[Review commands](#)

- After launching, wait for the pending state to change to running and for the 2 status checks to pass
- On selecting the instance, you can check out its details below



Demo CCP Inst...

i-0f2ca281a7b9716ec

Running



t2.micro

2/2 checks passed

Instances (1/6) [Info](#)

[C](#) [Connect](#) Instance state ▾ Actions ▾ **Launch instances** ▾

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status
<input checked="" type="checkbox"/> Demo CCP Inst...	i-0f2ca281a7b9716ec	Running	t2.micro	2/2 c

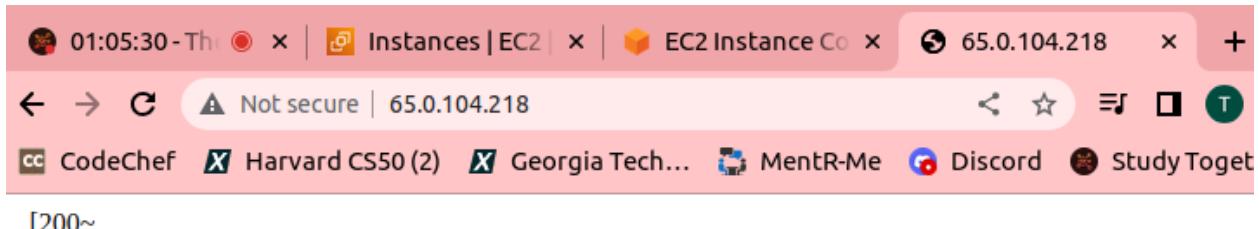
Instance: i-0f2ca281a7b9716ec (Demo CCP Instance)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status checks](#) [Monitoring](#) [Tags](#)

Instance summary [Info](#)

Instance ID i-0f2ca281a7b9716ec (Demo CCP Instance)	Public IPv4 address 65.0.104.218 open address	Private IPv4 addresses 172.31.37.95
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-65-0-104-218.ap-south-1.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-37-95.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-37-95.ap-south-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding
Auto-assigned IP address	VPC ID	

- Now, to check the web server running on our instance, copy the public IPv4 address of the instance and paste it in your browser. It won't work yet.
- It isn't working because in the URL, you need to make sure that you're using the HTTP protocol.



Hello worldHello world

- **Note:** every time you stop an instance and start it again, the public IP changes.

EC2 Instance Types

- Different types of EC2 instances are optimized for different use cases.
- <https://aws.amazon.com/ec2/instance-types/>
- In here, you will discover different types.
- Each type has its own family. Members of the family are different configurations to choose from, each configuration has its own purpose to serve.

General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

[General Purpose](#)

[Compute Optimized](#)

[Memory Optimized](#)

[Accelerated Computing](#)

[Storage Optimized](#)

[HPC Optimized](#)

[Instance Features](#)

[Measuring Instance Performance](#)

M7g	M7i	M7i-flex	M7a	Mac	M6g	M6i	M6in	M6a
M5	M5n	M5zn	M5a	M4	T4g	T3	T3a	T2

[Amazon EC2 M7g Instances](#) are powered by Arm-based AWS Graviton3 processors. They deliver the best price performance in Amazon EC2 for general purpose applications.

Features:

- Powered by custom-built AWS Graviton3 processors
- Features the latest DDR5 memory that offers 50% more bandwidth compared to DDR4
- 20% higher enhanced networking bandwidth compared to M6g instances
- EBS-optimized by default
- Instance storage offered via EBS or NVMe SSDs that are physically attached to the host server
- With M7gd instances, local NVMe-based SSDs are physically connected to the host server and provide block-level storage that is coupled to the lifetime of the instance
- Supports [Elastic Fabric Adapter \(EFA\)](#) on m7g.16xlarge, m7g.metal, and m7gd.16xlarge
- Powered by the [AWS Nitro System](#), a combination of dedicated hardware and lightweight hypervisor

- How to read into a configuration of an instance? You can get an overview like this:

Here, m instance class is for general purpose registers. At present, the generation has reached till 7.

m5.2xlarge

- m: instance class
- 5: generation (AWS improves them over time)
- 2xlarge: size within the instance class
- **Note: t2.micro** is a general purpose configuration of EC2 instance included with AWS free-tier.

- **Let's dive into different types:**

1. **General Purpose**

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads.

Typically used to web servers or code repositories.

2. **Compute Optimized**

- Great for compute-intensive tasks that require high performance processors:
 - Batch processing workloads
 - Media transcoding
 - High performance web servers
 - High performance computing (HPC)
 - Scientific modeling & machine learning
 - Dedicated gaming servers

3. **Memory Optimized**

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

- Fast performance for workloads that process large data sets in memory
- Use cases:
 - High performance, relational/non-relational databases
 - Distributed web scale cache stores
 - In-memory databases optimized for BI (business intelligence)
 - Applications performing real-time processing of big unstructured data

4. **Storage Optimized**

- Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage
- Use cases:
 - High frequency online transaction processing (OLTP) systems
 - Relational & NoSQL databases
 - Cache for in-memory databases (for example, Redis)
 - Data warehousing applications
 - Distributed file systems

5. Accelerated Computing

Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

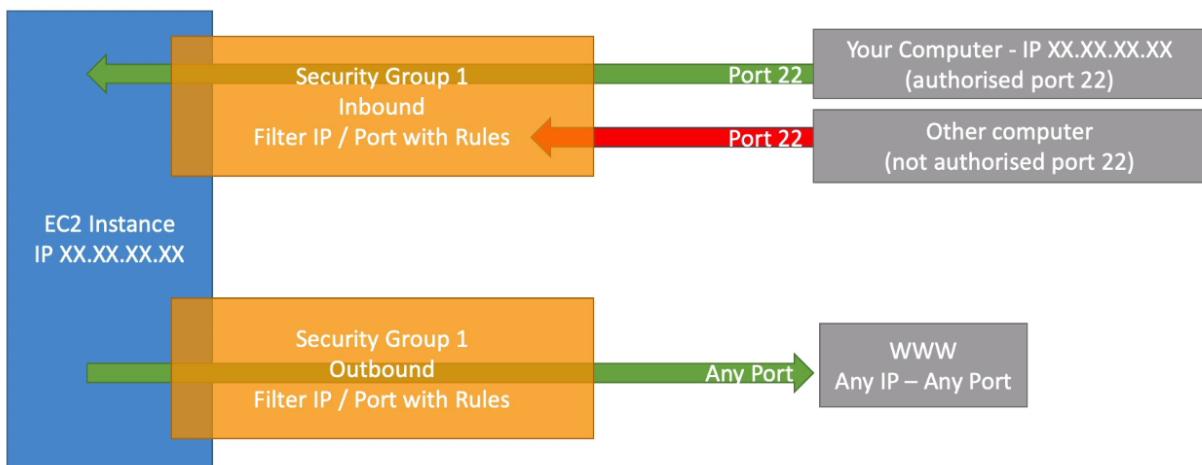
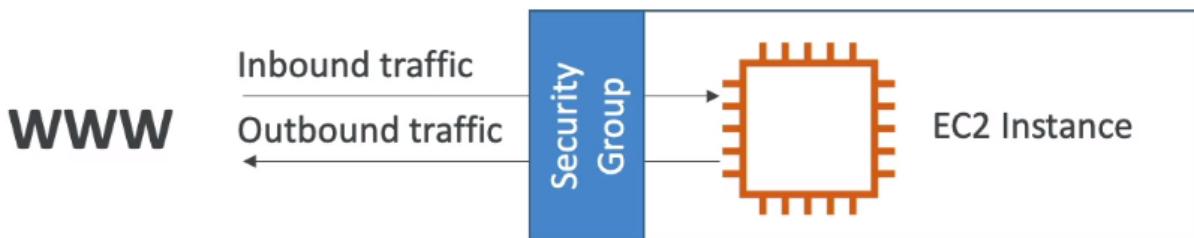
6. HPC Optimized (High Performance Computing)

HPC instances are ideal for applications that benefit from high-performance processors such as large, complex simulations and deep learning workloads.

Security Groups

- Security groups act as firewalls for our EC2 instances.
- They control what traffic is allowed into or out of our EC2 instance.
- Security groups are easy to maintain as they only contain **allow** rules.
- These rules can be applied to certain IP addresses or ports or even another security group.

- They regulate:
 - Access to Ports
 - Authorised IP ranges – IPv4 and IPv6
 - Control of inbound network (from other to the instance)
 - Control of outbound network (from the instance to other)



- Inbound and Outbound rules:

Certainly! Think of inbound and outbound security group rules for an EC2 instance as a set of rules that control who can visit your house (EC2 instance) and where you can go outside.

1. **Inbound Rules (People coming to your house):** These rules decide who is allowed to visit your EC2 instance from the outside. You can set rules to allow specific "visitors" (IP addresses) and say which "doors" (ports) they can use. For example, you might allow your friends to come in through the front door (port 80) but not strangers.
2. **Outbound Rules (You going outside):** These rules determine where you are allowed to go from your EC2 instance. You can set rules to specify which "destinations" (IP addresses) you can visit and which "doors" (ports) you can use. For instance, you might be allowed to go to the park (a specific IP address) through the main gate (a specific port), but not anywhere else.

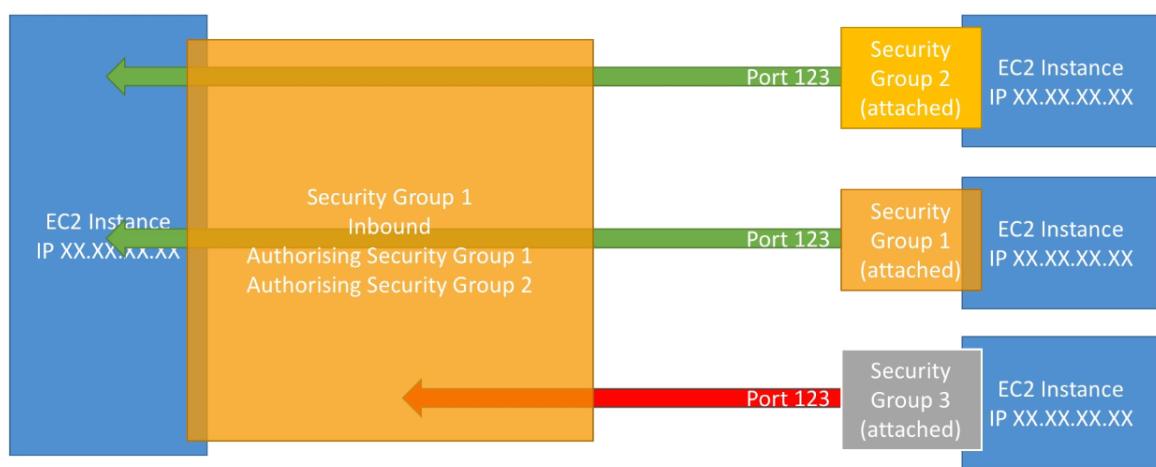
In both cases, these rules help ensure that your EC2 instance only talks to the right people and goes to the right places, keeping it secure. It's like having a doorman for your virtual house to make sure only approved guests get in and that you don't wander into risky areas.

- Security groups have a many-to-many relationship — a security group can be attached to multiple instances and one instance can have multiple security groups.
- By default, all inbound traffic is blocked and all outbound traffic is authorized. This will stay as is unless changed by you.
- Locked down to a region /VPC combination
- Does live “outside” the EC2 – if traffic is blocked the EC2 instance won’t see it
- It’s good to maintain one separate security group for SSH access
- If your application is not accessible (time out), then it’s a security group issue
- If your application gives a “connection refused” error, then it’s an application error or it’s not launched
- **How to reference security groups from other security groups?**

Basically how this works is:

Imagine you want to set inbound rules for a specific EC2 instance (say A) and you set it to **authorize security group 1 and security group 2** so then what happens is if another EC2 instance (say B) located anywhere, has security group 1 or 2 attached to it then instance B will directly be able to connect to instance A and then you don't have to worry about IP addresses.

Referencing other security groups Diagram



- Common security ports to know for the exam

Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance
- 21 = FTP (File Transfer Protocol) – upload files into a file share
- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH
- 80 = HTTP – access unsecured websites
- 443 = HTTPS – access secured websites
- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

Security Groups Hands-on

Networking and security >> Security Groups

- A tip: whenever you're trying to access your application with EC2 instance (either through SSH or HTTP query or anything), if it's giving a time out (infinite loading and then **err_time_out**) it means this has something to do with the security groups.

Possibly with port 80 HTTP rule. Look into it to fix this issue.

- Easily add or delete your inbound and outbound rules.

The screenshot shows the 'Inbound rule 3' configuration page. It includes fields for Security group rule ID (empty), Type (set to HTTPS), Protocol (TCP), Port range (443), and a Source field (empty). An 'Add rule' button is available. A note at the bottom left says: '⚠ Rules with source of 0.0.0.0/0 or recommend setting security group'. A note at the bottom right says: 'Your instance. We IP addresses only.' with a close button. Buttons for 'review changes' and 'Save rules' are at the bottom right.

Inbound rule 3	
Security group rule ID	Type info
-	HTTPS
Port range info	Protocol info
443	TCP
Description - optional info	Source info
<input type="text"/>	<input type="text"/>
<button>Add rule</button>	
<p>⚠ Rules with source of 0.0.0.0/0 or recommend setting security group</p>	
<p>Your instance. We IP addresses only.</p>	
<button>review changes</button>	<button>Save rules</button>

Inbound rule 3

Delete

Security group rule ID

-

Type Info

HTTPS

Protocol Info

TCP

Port range Info

443

Source type Info

Custom

Source Info

Search icon

Description - optional Info

- Custom (▲)
- Custom (✓)
- Anywhere-IPv4
- Anywhere-IPv6
- My IP

Add rule



Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

X

Cancel

Preview changes

Save rules

Let's say you choose to go with the My IP option, keep in mind that you will get a time out if your IP changes.

- Outbound traffic rule, by default, allows all traffic on IPv4 to anywhere. This allows the EC2 instance to get full internet connectivity anywhere.

sg-08b3080f32b0d6e27 - launch-wizard-7

The screenshot shows the AWS Management Console interface for managing security group rules. The top navigation bar includes 'Details', 'Inbound rules', 'Outbound rules' (which is the active tab), and 'Tags'. Below this, the 'Outbound rules' section displays one rule: 'sgr-05b31fcb404033fba' (Security group rule ID), 'IPv4' (IP version), and 'All traffic' (Type). A 'Filter security group rules' search bar and pagination controls (1 of 1) are also visible.

Name	Security group rule...	IP version	Type
-	sgr-05b31fcb404033fba	IPv4	All traffic

SSH-ing into your EC2 instance

- This basically means to connect to inside of your servers to perform some maintenance or action.
- SSH is one of the most important functions in the cloud as it lets you control your remote instance or server all using your CLI.
- Once are able to access our ec2 instance or SSH into it, our command line is going to be just as if we were inside that instance machine.

There are different ways of doing so based on your OS:



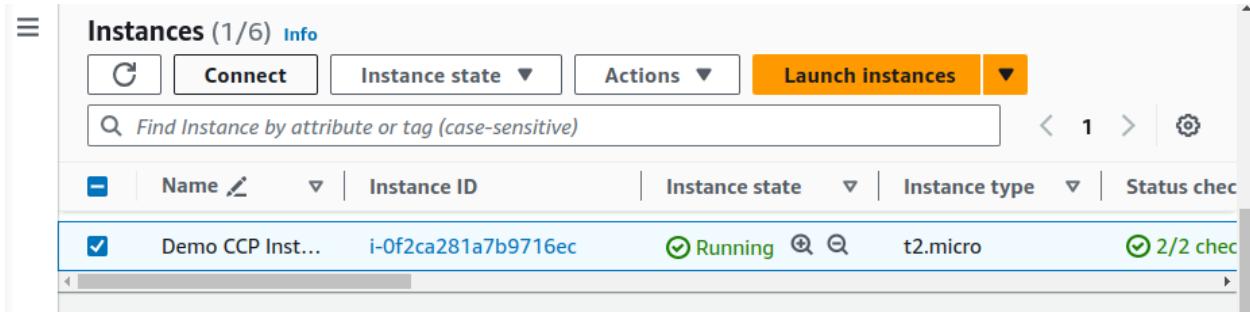
- Was able to ssh into my ec2 instance by following the commands in the tutorial, didn't get it though.

```
meowmeow@tanishas-penguin:~/Desktop/codewithtanisha/AWS Certs/AWS CCP CLF-C01$ ssh -i CCPDemo.pem ec2-user@3.111.168.216
, #_
~\_\_ #####_ Amazon Linux 2023
~~ \_\#####\
~~ \###|
~~ \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~' '-'>
~~ / /
~~ .-.- / /
~/m/
Last login: Mon Nov  6 17:20:09 2023 from 13.233.177.5
[ec2-user@ip-172-31-37-95 ~]$ 
```

- Command to close the connection: ctrl + G
- To get back into it: ssh -i <security_key_pair_file_name (ppk / pem)> ec2-user@<instances_public_IPv4>
ex: ssh -i CCPDemo.pem ec2-user@3.111.168.216
- These commands will work only if you're in the directory where your ppk or pem file is stored.

EC2 Instance Connect (easier alt to ssh)

- Allows us to do a browser based SSH session into our instance.



The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for 'Instances (1/6)', 'Info', 'Connect' (which is highlighted in blue), 'Instance state', 'Actions', and 'Launch instances'. Below this is a search bar with the placeholder 'Find Instance by attribute or tag (case-sensitive)'. The main table has columns for 'Name' (with a pencil icon), 'Instance ID', 'Instance state' (showing 'Running'), 'Instance type' (showing 't2.micro'), and 'Status check' (showing '2/2 check'). A single row is selected, showing 'Demo CCP Inst...' as the name, 'i-0f2ca281a7b9716ec' as the instance ID, 'Running' as the state, 't2.micro' as the type, and '2/2 check' as the status. The table includes navigation arrows and a settings gear icon at the bottom right.

- Once we connect to it, it's going to upload a temporary SSH key for us and establish a connection. This way we don't need to manage SSH keys.

Connect to instance [Info](#)

Connect to your instance i-0f2ca281a7b9716ec (Demo CCP Instance) using any of these options

[EC2 Instance Connect](#)

[Session Manager](#)

[SSH client](#)

[EC2 serial console](#)

Instance ID

[i-0f2ca281a7b9716ec \(Demo CCP Instance\)](#)

Connection Type

Connect using EC2 Instance Connect

Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.

Connect using EC2 Instance Connect Endpoint

Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address

[3.111.168.216](#)

User name

Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.

ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

[Cancel](#)

[Connect](#)

```

'      #
~\_ ####_      Amazon Linux 2023
~~ \#####\
~~ \###|
~~  \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
~~   V-' '-'>
~~   /
~~_. /_
~/ /_/
/_m/' 

Last login: Tue Nov  7 09:53:53 2023 from 122.168.54.167
[ec2-user@ip-172-31-37-95 ~]$ ping amazon.in
PING amazon.in (54.239.33.92) 56(84) bytes of data.
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=1 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=2 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=3 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=4 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=5 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=6 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=7 ttl=221 time=121 ms
64 bytes from 54.239.33.92 (54.239.33.92): icmp_seq=8 ttl=221 time=121 ms
^C
--- amazon.in ping statistics ---
3 packets transmitted, 8 received, 0% packet loss, time 7010ms
rtt min/avg/max/mdev = 120.837/120.864/120.900/0.019 ms
[ec2-user@ip-172-31-37-95 ~]$ []

```

- Note: this will only work if you have allowed port 22 (SSH) in your inbound rules. Sometimes you need to add both IPv4 and IPv6 from anywhere for it to work.

Inbound rules Info					
Security group rule ID	Type Info	Protocol Info	Port range	Source Info	Description - optional Info
sgr-06b77ae827058fb19	HTTP	TCP	80	Custom ▾ <input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>
sgr-0d3e6dcdd8ab1b41a	SSH	TCP	22	Custom ▾ <input type="text" value="0.0.0.0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>
-	SSH	TCP	22	Anywh... ▾ <input "::="" 0"="" type="text" value=""/> <input type="button" value="X"/>	<input type="button" value="Delete"/>

EC2 Instance Purchasing Options

- On-Demand Instances – short workload, predictable pricing, pay by second
- Reserved (1 & 3 years)
 - Reserved Instances – long workloads
 - Convertible Reserved Instances – long workloads with flexible instances
- Savings Plans (1 & 3 years) – commitment to an amount of usage, long workload
- Spot Instances – short workloads, cheap, can lose instances (less reliable)
- Dedicated Hosts – book an entire physical server, control instance placement
- Dedicated Instances – no other customers will share your hardware
- Capacity Reservations – reserve capacity in a specific AZ for any duration

EC2 On Demand

- Pay for what you use:
 - Linux or Windows - billing per second, after the first minute
 - All other operating systems - billing per hour
- Has the highest cost but no upfront payment
- No long-term commitment
- Recommended for short-term and un-interrupted workloads, where you can't predict how the application will behave

EC2 Reserved Instances

- Up to **72%** discount compared to On-demand
 - You reserve a specific instance attributes (Instance Type, Region, Tenancy, OS)
 - Reservation Period – 1 year (+discount) or 3 years (+++discount)
 - Payment Options – No Upfront (+), Partial Upfront (++) , All Upfront (+++)
 - Reserved Instance's Scope – Regional or Zonal (reserve capacity in an AZ)
 - Recommended for steady-state usage applications (think database)
 - You can buy and sell in the Reserved Instance Marketplace
-
- Convertible Reserved Instance
 - Can change the EC2 instance type, instance family, OS, scope and tenancy
 - Up to **66%** discount

Note: the % discounts are different from the video as AWS change them over time – the exact numbers are not needed for the exam. This is just for illustrative purposes ☺

EC2 Savings Plans

- Get a discount based on long-term usage (up to 72% - same as RIs)
 - Commit to a certain type of usage (\$10/hour for 1 or 3 years)
 - Usage beyond EC2 Savings Plans is billed at the On-Demand price
-
- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)
 - Flexible across:
 - Instance Size (e.g., m5.xlarge, m5.2xlarge)
 - OS (e.g., Linux, Windows)
 - Tenancy (Host, Dedicated, Default)



EC2 Spot Instances

- Can get a **discount** of up to 90% compared to On-demand
- Instances that you can “lose” at any point of time if your max price is less than the current spot price
- The **MOST** cost-efficient instances in AWS
- Useful for workloads that are resilient to failure
 - Batch jobs
 - Data analysis
 - Image processing
 - Any distributed workloads
 - Workloads with a flexible start and end time
- Not suitable for critical jobs or databases

EC2 Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use
- Allows you address **compliance requirements** and **use your existing server-bound software licenses** (per-socket, per-core, per-VM software licenses)
- Purchasing Options:
 - On-demand – pay per second for active Dedicated Host
 - Reserved - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)
- The most expensive option
- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)
- Or for companies that have strong regulatory or compliance needs

EC2 Dedicated Instances

- Instances run on hardware that's dedicated to you
- May share hardware with other instances in same account
- No control over instance placement (can move hardware after Stop / Start)
- **The key difference between dedicated instances and dedicated hosts is that**
—

With EC2 dedicated hosts, you have complete physical isolation on a physical server (host). This means you have exclusive access to the entire server, and no other AWS customers' instances can run on the same host.

When you use dedicated instances, it means that the physical hardware your virtual machines (EC2 instances) run on is reserved exclusively for your AWS account. While the physical hardware may be shared with other AWS customers' dedicated instances, there is a virtual barrier that separates your instances from theirs. Your instances don't mix or interact with instances from other AWS accounts on the same hardware.

** dedicated instance is like owning a floor in a building while dedicated host is like owning the whole damn building.

EC2 Capacity Reservations

- Reserve On-Demand instances capacity in a specific AZ for any duration
- You always have access to EC2 capacity when you need it
- No time commitment (create/cancel anytime), no billing discounts
- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts
- You're charged at On-Demand rate whether you run instances or not
- Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ
- **How to choose a pricing option? (imp for exam)**

Which purchasing option is right for me?



- On demand: coming and staying in resort whenever we like, we pay the full price
- Reserved: like planning ahead and if we plan to stay for a long time, we may get a good discount.
- Savings Plans: pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, ...)
- Spot instances: the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- Dedicated Hosts: We book an entire building of the resort
- Capacity Reservations: you book a room for a period with full price even you don't stay in it

Price Comparison

Example – m4.large – us-east-1

Price Type	Price (per hour)
On-Demand	\$0.10
Spot Instance (Spot Price)	\$0.038 - \$0.039 (up to 61% off)
Reserved Instance (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved Instance (3 years)	\$0.043 (No Upfront) - \$0.037 (All Upfront)
EC2 Savings Plan (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved Convertible Instance (1 year)	\$0.071 (No Upfront) - \$0.066 (All Upfront)
Dedicated Host	On-Demand Price
Dedicated Host Reservation	Up to 70% off
Capacity Reservations	On-Demand Price

Shared Responsibility Model for EC2

Shared Responsibility Model for EC2



- Infrastructure (global network security)
- Isolation on physical hosts
- Replacing faulty hardware
- Compliance validation
- Security Groups rules
- Operating-system patches and updates
- Software and utilities installed on the EC2 instance
- IAM Roles assigned to EC2 & IAM user access management
- Data security on your instance

Concluding

EC2 Section – Summary



- **EC2 Instance:** AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- **Security Groups:** Firewall attached to the EC2 instance
- **EC2 User Data:** Script launched at the first start of an instance
- **SSH:** start a terminal into our EC2 Instances (port 22)
- **EC2 Instance Role:** link to IAM roles
- **Purchasing Options:** On-Demand, Spot, Reserved (Standard + Convertible), Dedicated Host, Dedicated Instance



Good job!

Compute Optimized EC2 instances are great for compute-intensive workloads requiring high performance processors, such as batch processing, media transcoding, high performance web servers, high performance computing, scientific modeling & machine learning, and dedicated gaming servers.

Question 5:

A company would like to deploy a high-performance computing (HPC) application on EC2. Which EC2 instance type should it choose?



Compute Optimized



Storage Optimized



Memory Optimized



General Purpose

**Good job!**

Reserved Instances are good for long workloads. You can reserve instances for 1 or 3 years.

Question 7:

Which EC2 Purchasing Option should you use for an application you plan on running on a server continuously for 1 year?

Reserved Instances

Spot Instances

On-demand Instances

Convertible Instances