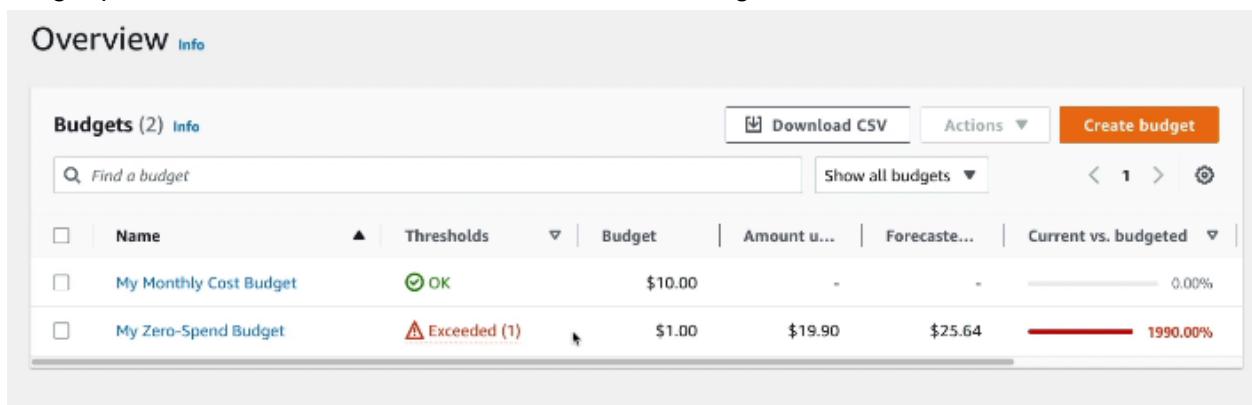


AWS Budget Setup

1. We have to setup billing details(budget) to make sure to spend the least amount of money or no money,
2. Billing budget will gives us alert ..if there we are spending some money
3. IAM user by default will not have billing access...We have to get it activated from the root account
4. After that we can access billing section of the IAM user
5. We can go to bills section and see for which services we are getting charged for
6. This way we can debug our own cose expenses
7. If we go to freetier section...we can see how much service is left for us to use
8. Then going to budget section...we can set our monthly budget plan and also zero budget plan...which notifies us if we exceed 0.1\$ of charges



The screenshot shows the AWS Budgets Overview page. At the top, there are buttons for 'Download CSV', 'Actions', and 'Create budget'. Below that is a search bar with 'Find a budget' placeholder and a 'Show all budgets' dropdown. A navigation bar with icons for back, forward, and search is at the bottom. The main table has columns: Name, Thresholds, Budget, Amount used, Forecasted, and Current vs. budgeted. Two rows are listed:

Name	Thresholds	Budget	Amount used	Forecasted	Current vs. budgeted
My Monthly Cost Budget	OK	\$10.00	-	-	0.00%
My Zero-Spend Budget	Exceeded (1)	\$1.00	\$19.90	\$25.64	1990.00%

9.

EC2 Basics:



The screenshot shows the Amazon EC2 landing page. The title 'Amazon EC2' is prominently displayed. Below it is a large call-to-action button labeled 'Get Started'. To the right of the button is a 'Launch' button. Below the title, there is a brief introduction and a bulleted list of facts:

- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service

- 1.
2. Well, it stands for Elastic Compute Cloud and this is the way to do Infrastructure as a Service
3. Also It is used everywhere

- It mainly consists in the capability of :
 - Renting virtual machines (EC2)
 - Storing data on virtual drives (EBS)
- 4.
- Distributing load across machines (ELB)
 - Scaling the services using an auto-scaling group (ASG)
- 5.
6. We can rent virtual machine on EC2 which are called ec2 instances and we can also store data on EBS or virtual drives
- Knowing EC2 is fundamental to understand how the Cloud works
- 7.
8. EC2 sizing and configuration :
9. First we have choose an OS ...linux is most used ..but we can use windows or mac
10. We need to choose how much power and cores(CPU) and how much RAM..how much storage space
11. If we want storage that is going to be attached through the network (thru EBS and EFS) or do you want it to be hardware attached?
- 12.

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)
13. We have to attach type of network to our EC2 instance(fast type etc) , the kind of public ip address
14. We have to also setup firewall rules

- Network card: speed of the card, Public IP address
- Firewall rules: **security group**
- Bootstrap script (configure at first launch): EC2 User Data

15. _____

16. EC2 user data:

EC2 User Data

- It is possible to bootstrap our instances using an [EC2 User data](#) script.
- [bootstrapping](#) means launching commands when a machine starts
- That script is [only run once](#) at the instance [first start](#)
- EC2 user data is used to automate boot tasks such as:
 - Installing updates
 - Installing software
 - Downloading common files from the internet
 - Anything you can think of
- The EC2 User Data Script runs with the root user

17.

18. the more you add into your User Data script, the more your instant has to do at boot time.

19. Simple, right?

20. By the way, the EC2 User Data scripts runs with a root user. [So any command you have will have the pseudo rights.](#)

21. EC2 instance types: example

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

22.

23. Here c5d..has hardware storage attached...we can see all the ec2 configs for each instance here..

t2.micro is part of the AWS free tier (up to 750 hours per month)

24.

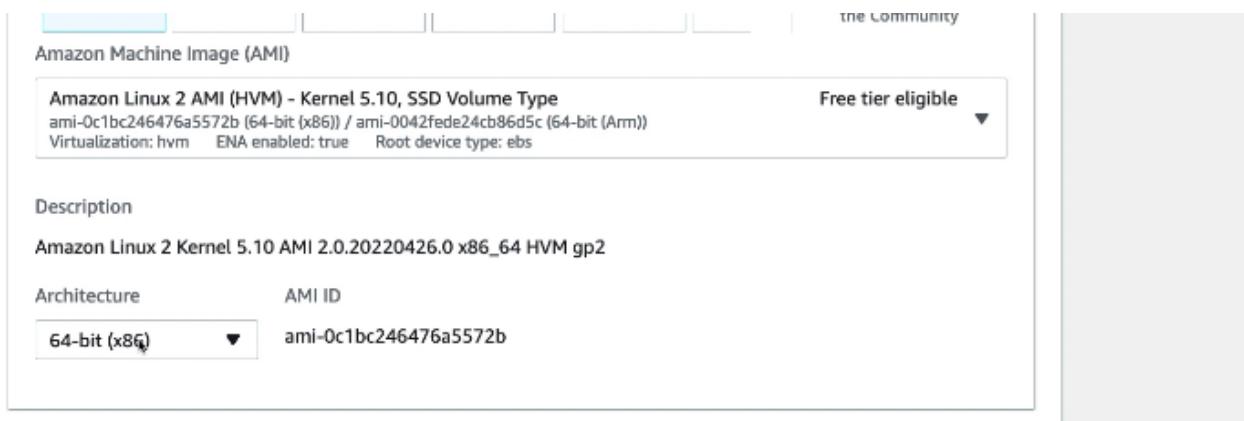
EC2 Hands on:

Hands-On: Launching an EC2 Instance running 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.

1.

2. We can create ec2 instance by going into its console and clicking create imstace ..we have to set name and we have to choose an image



3.

4. Next we have to choose instance type ..we are choosing t2 micro as it is free tier and better than t1
5. Then we have to ccreate key pair..here we have created Ec2 key pair and downloaded it
6. Then we set network settings and security firewall(to allow traffic)
7. Next we have to configure storage details..if we go to advanced settings we'll have more options
8. And then next in advanced settings ..in user data we have pass commands to our ec2 instance to execute on its first launch of instance..we have pasted ec2-user-data.ssh commands ..in commands we have created index.html file

User data Info

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello World from $(hostname -f)</h1>" > /var/www/html/index.html
```

9.

10. We can go to instances section..to see our current ec2 instances..it will take 2 minutes to run from pending state
11. If we click on our instance we can get the summary of our instance
12. In There is a public IPv4 address,
13. this is what we're going to use to access our EC2 instance,
14. or there is a private IPv4 address which is how to access that instance internally [on the AWS network, which is private.](#)
15. If we go to public ip address which is "<http://18.234.102.157/>" we get hello world msg from index.html file

16. We can also stop our instance from instance state...If we stop instance we dont get charged
17. If we stop and start instance later on..public ip address will be changed..private ip will always stays same

EC2 Instance Types Basics

EC2 Instance Types - Overview

- You can use different types of EC2 instances that are optimised for different use cases (<https://aws.amazon.com/ec2/instance-types/>)
- AWS has the following naming convention:

m5.2xlarge

- **m**: instance class
- **5**: generation (AWS improves them over time)
- **2xlarge**: size within the instance class

General Purpose
Compute Optimized
Memory Optimized
Accelerated Computing
Storage Optimized
Instance Features
Measuring Instance Performance

1. _____
2. General purpose instance are great for....

EC2 Instance Types – General Purpose

- Great for a diversity of workloads such as web servers or code repositories
- Balance between:
 - Compute
 - Memory
 - Networking
- In the course, we will be using the t2.micro which is a General Purpose EC2 instance

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.

Mac	T4g	T3	T3a	T2	M6g	M5	M5a	M5n	M5zn	M4	A1
-----	-----	----	-----	----	-----	----	-----	-----	------	----	----

* this list will evolve over time. please check the AWS website for the latest information

3. _____
4. Compute optimised

EC2 Instance Types – 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

Compute Optimized

Compute Optimized Instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modelling, dedicated gaming servers and ad server engines, machine learning Inference and other compute intensive applications.

C6g	C6gn	C5	C5a	C5n	C4
-----	------	----	-----	-----	----

- 5.
6. Here compute optimised instances starts with c2,c5 like names
7. Memory optimised

EC2 Instance Types – Memory Optimized

- 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

Memory Optimized

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

R6g R5 R5a R5b R5n R4 X1e X1 High Memory z1d

8.

9. Their names start with R(means RAM)

10. but there's also going to be X one high memory and Z one, but again, you don't have to remember the name of the instances, but good to know at a high level.

11. Storage optimised

EC2 Instance Types – 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

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

I3 I3en D2 D3 D3en H1

12.

13. We can compare all this here

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

14.

t2.micro is part of the AWS free tier (up to 750 hours per month)

Security Groups & Classic Ports Overview

1. Introduction to security groups:

Introduction to Security Groups

- Security Groups are the fundamental of network security in AWS
 - They control how traffic is allowed into or out of our EC2 Instances.
-
- Security groups only contain **allow** rules
 - Security groups rules can reference by IP or by security group

2. —
3. and security groups can have rules that reference either by IP addresses, so, where your computer is from **or by other security groups**.
4. Security groups acts as a firewall to our ec2 instance ..it allows inbound(from internet to ec2) and outbound(from ec2 to internet) traffic for our instance
5. Security groups deep dive:

Security Groups Deeper Dive

- Security groups are acting as a “firewall” on EC2 instances
- 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)

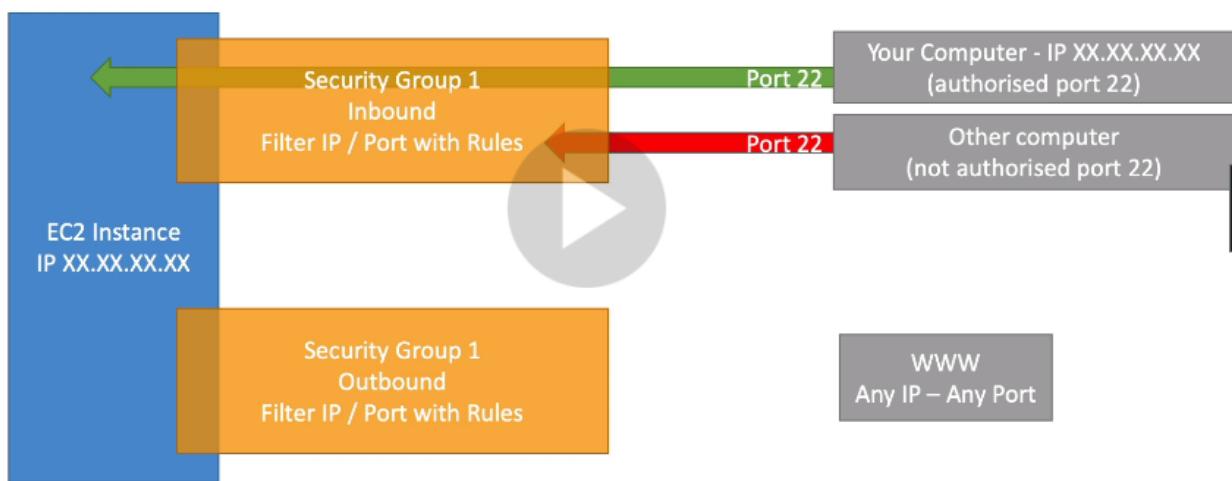
6.

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80 ↗	0.0.0.0/0	test http page
SSH	TCP	22	122.149.196.85/32	
Custom TCP Rule	TCP	4567	0.0.0.0/0	java app

7.

8. Security group diagram:

Security Groups Diagram



9.

10. Inbound traffic have to be authorised and outbound can send traffic to any source/port

Security Groups

Good to know

- Can be attached to multiple instances
- 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
- All inbound traffic is **blocked** by default
- All outbound traffic is **authorised** by default

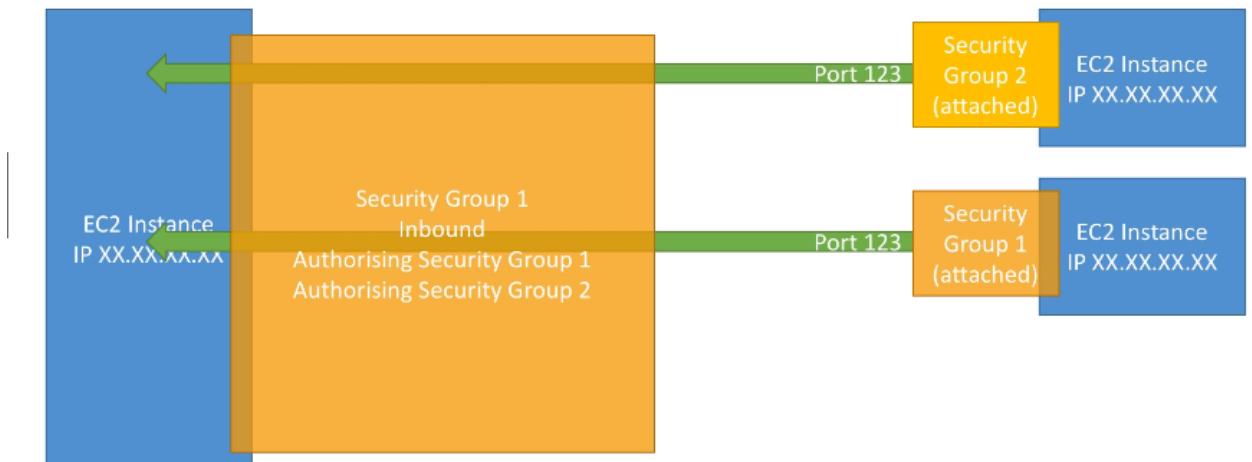
11.

12. If we get connection refused it means that it went thru security and theres some issue with our application

13. Referencing other security groups

Referencing other security groups

Diagram



14.

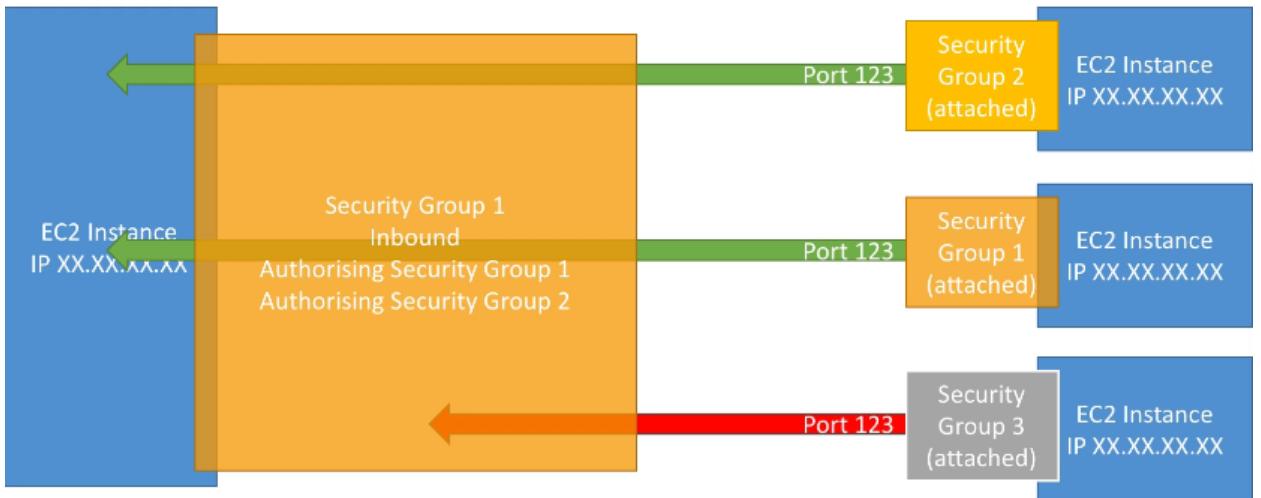
15. Here as other instance have same security group it has access to the main ec2 instance

16. And so regardless of the IP of our EC2 instances because they have the right

security group attached to them they're able to communicate straight through to other instances.

Referencing other security groups

Diagram



17.

18. Classic ports to know:

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

19.

20. Port 22 allows us to login to an ec2 instance on linux

21. And 3389 is for loggin into windows instance

Security Group Hands on:

1. If we click on our instance we get some details...then we can go to its security section to views some security group
2. To get more details we can go to security groups under network security ..in here we can see there is launchwizard1 which created for our ec2 instance
3. By selecting it we can get more details about inbound and outbound traffic

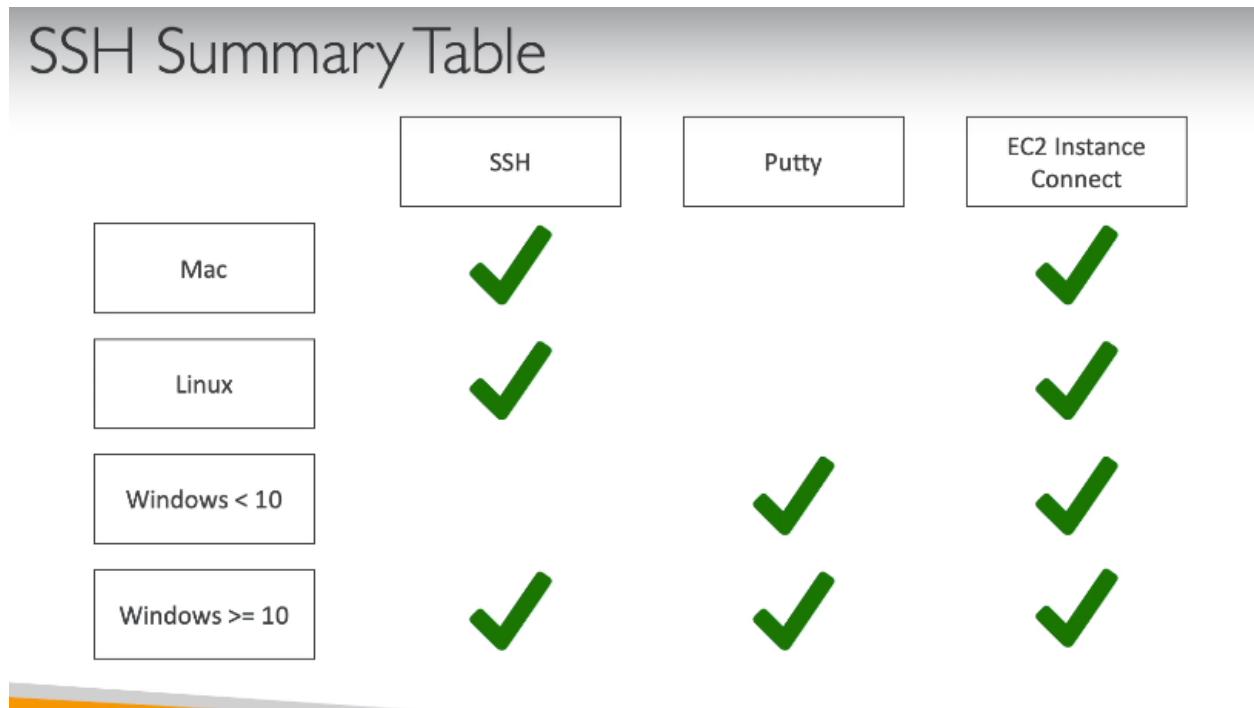
4. Inbound traffic must be from port 22 of SSH ...source = 0.0.0...means it accepts from anywhere
5. If we go to edit inbound we get this

Inbound rules Info						
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-014c028be257f5c2	SSH	TCP	22	Custom ▾ <input type="text" value="0.0.0.0"/> <input type="button" value="X"/>		Delete
sgr-0d0ebef26009f47e6	HTTP	TCP	80	Custom ▾ <input type="text" value="0.0.0.0"/> <input type="button" value="X"/>		Delete

6. [Add rule](#)
7. With 2nd inbound rule we were able to access instance from server ..by going to public ip address
8. If we get timeout issue..then it is for sure due to ec2 instance security groups
9. We can also delete and add security group rules
10. When creating security grp rule ..if in source we select my IP...it only allows our ip address
11. We can add multiple security groups to our instance

SSH Summary Table:

1. How do we connect inside of our servers to perform some maintenance or action



- 2.

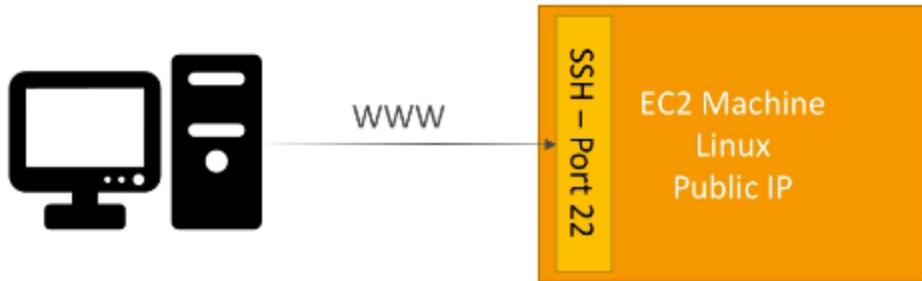
SSH troubleshooting

- Students have the most problems with SSH
 - If things don't work...
 1. Re-watch the lecture. You may have missed something
 2. Read the troubleshooting guide
 3. Try EC2 Instance Connect
 - If one method works (SSH, Putty or EC2 Instance Connect) you're good
- 3.

How to use SSH in windows?

How to SSH into your EC2 Instance Windows

- We'll learn how to SSH into your EC2 instance using [Windows](#)
 - SSH is one of the most important function. It allows you to control a remote machine, all using the command line.
1. SSH is used control a remote machine(on our case EC2 instance)..using the command line
 2. [if you remember,](#) we had an SSH security group on EC2 instance
 3. And basically we allowed SSH on port 22 to any IP, which basically allows our Windows machine to connect over the internet directly into the machine and control it using the command line.



- 5.
6. In previous version of windows less than 10...to use SSH we have to use putty
7. But in windows 10 and 11 we can use it in command line...just type ssh in cmd ..to see if its their
8. In the cmd..we have to go the directory where our Ec2tutorial.pem is present...in our case it was desktop
9. Then we have to enter "ssh -i .\Ec2Tutorial.pem ec2-user@public ip address"
10. This command says that please enter this IP using this user, the EC2 user which is the one we have using the key(.pem file)

```
PS C:\Users\stephanemaarek\Desktop> ssh -i .\Ec2Tutorial.pem ec2-user@3.250.26.200
The authenticity of host '3.250.26.200 (3.250.26.200)' can't be established.
ECDSA key fingerprint is SHA256:INQMVOPTzV+fRKl+EeoRtTp+pWI4koar4F8a6QTmPgE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.250.26.200' (ECDSA) to the list of known hosts.
Last login: Mon May 16 17:28:25 2022 from bl14-117-32.dsl.telepac.pt

[ec2-user@ip-172-31-33-135 ~]$ e_
```

12. [ec2-user@ip-172-31-33-135 ~]\$ e_

13. If we getting some permission issues..we have to change some security settings for ec2tutorial.pem file..see online

EC2 Instance connect:

1. To use Ec2 instance connect...first we to go to our instances and click on instance then on connect

Connect to instance Info

Connect to your instance i-034466697feb9ef80 (My First Instance) using any of these options

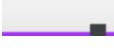
EC2 Instance Connect Session Manager SSH client EC2 Serial Console

Instance ID
Copy [i-034466697feb9ef80 \(My First Instance\)](#)

Public IP address
Copy 3.250.26.200

User name Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

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

2. 
3. Username is provided by default
4. as you see there's no SSH key option, when we click connect to it, it's going to upload for us a temporary SSH key and establish a connection
5. To use SSH ...we can use either putty or cmd or instance connect
6. If theres any issue accessing SSH..it might be from inbound security group rules
7. We have to allow port 22 for SSH

sgr-0d3e6dcdd8ab1b41a	SSH	TCP	22	Custom	<input type="text" value="0.0.0.0"/> X	<input type="button" value="Delete"/>
<input type="text" value="0.0.0.0"/> X						<input type="button" value="Delete"/>

8. <https://chat.openai.com/share/b53f4ba9-39f6-48ba-9046-389cf8d928fe>

```
ec2-user@ip-172-31-33-135 ~]$ ping google.com
PING google.com (172.253.116.100) 56(84) bytes of data.
64 bytes from dj-in-f100.1e100.net (172.253.116.100): icmp_seq=1 ttl=103 time=1.44 ms
64 bytes from dj-in-f100.1e100.net (172.253.116.100): icmp_seq=2 ttl=103 time=1.44 ms
64 bytes from dj-in-f100.1e100.net (172.253.116.100): icmp_seq=3 ttl=103 time=1.48 ms
< google.com ping statistics ---
6 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt avg/max/mdev = 1.442/1.457/1.481/0.035 ms
```

- 9.

EC2 Instance Roles Demo

1. Here first we have to connect to our ec2 instance using instance connect
2. Now we have linux terminal(ec2 Instance connect) available to use via cloud
3. If we entered aws iam list-users...we get “unable to locate credentials.....” But we should not use aws configure in the terminal
4. As a rule of thumb ...never enter any kind of secret keys or passwords into instance connect..our credentials may get stealed
5. instead we have to use IAM roles..
6. We have created a demorole for ec2...it has one policy called IAMReadOnlyAccess
7. Now we will attach this role to ec2 instance to provide it with credentials
8. Currently we dont have any IAMroles for our ec2 instance(we can check in security tab)
9. We will give it demoec2 role..which we have created and it has IAMReadOnlyAccess
10. Then if we go to our instance connect and pressed “aws iam list-users” we get the results now...because of the role

```
[ec2-user@ip-172-31-17-250 ~]$ aws iam list-users
{
    "Users": [
        {
            "UserName": "stephane",
            "PasswordLastUsed": "2021-04-19T20:41:59Z",
            "CreateDate": "2021-04-19T20:38:18Z",
            "UserId": "AIDATCOXNUAWA5KFW6MM7",
            "Path": "/",
            "Arn": "arn:aws:iam::211442049068:user/stephane"
        }
    ]
}
```

- 11.
12. So the role is really linked now to the EC2 Instance.
13. And this is how we provide AWS credentials to our EC2 Instances only, only through IAM roles.

EC2 Instances Purchasing Options

1. So far we've been using on demand instances
 - On-Demand Instances – short workload, predictable pricing, pay by second

EC2 Instances 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

2.

3. Ec2 On Demand:

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

4.

5. Ec2 Reserved instances:

6. In here we can also buy and sell our reserved instance in the marketplace if we dont need them anymore

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 ☺

7.

8. Ec2 Savings plans:

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)

9.

10. EC2 spot instances:

11. Suitable for workloads which easily recover from failures



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

12.

13. EC2 Dedicated Hosts:

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

14.

Hello Joel,

Dedicated Host

As soon as you 'allocate' a Dedicated Host, **you start paying for that whole host.**

A host computer is very big. In fact, it is the size of the largest instance of the selected family, but can be divided up into smaller instances of the same family. ("You can run any number of instances up to the core capacity associated with the host.")

Any instances that run on that Host are not charged, since you are already being billed for the Host.

That is why a Dedicated Host is more expensive than a Dedicated Instance -- the charge is for the *whole host*.

Dedicated Instance

"Dedicated Instances are Amazon EC2 instances that run in a virtual private cloud (VPC) on hardware that's dedicated to a single customer... Dedicated Instances may share hardware with other instances from the same AWS account that is not Dedicated Instances."

This means that **no other AWS Account will run an instance on the same Host**, but other instances (both dedicated and non-dedicated) from the same AWS Account might run on the same Host.

Billing is per instance, with a cost of approximately 10% more than the normal instance charge (but no extra charge if it is the largest instance in the family since it requires the whole host anyway).

I hope this helps.

Regards,
Manish

15.

16. Ec2 Dedicated Instances:

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)

Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a \$2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

17.

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

18.

19. Which purchasing option is right? Explained with resort example

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

20.

21. Price comparison

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

22.

Shared responsibility model for Ec2

1. AWS is responsible for the entire infrastructure, isolation on physical hosts, ...

Shared Responsibility Model for EC2



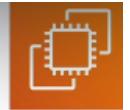
- Infrastructure (global network security)
- Isolation on physical hosts
- Replacing faulty hardware
- Compliance validation
- Security Groups rules
- Operating-system patches and updates
- Software and utilities installed on the EC2 instance
- IAM Roles assigned to EC2 & IAM user access management
-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

2.

EC2 Summary

1. At first we created an EC2 instance : by specifying OS, Instance size, Storage, Security grps(firewalls), Ec2 userdata
2. Security grps : We will attach a firewall to our Ec2 instance and it allows which address to access our instance ...So the security groups are attached to EC2 Instances.
3. And they are a firewall outside of your instance. And you can define rules to allow which ports and which API can access your EC2 instance.

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 + Scheduled), Dedicated Host, Dedicated Instance

4.

Extras:

1. 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.