**ELASTIC COMPUTE CLOUD**

EC2 = Elastic Compute Cloud. It is an Infrastructure As A Service (IaaS). It mainly consists of the capability such as:

* Renting Virtual Machines (EC2)
* Storing Data on Virtual Drives (EBS)
* Distributing Load across machines (ELB)
* Scaling the services using an auto-scaling groups (ASG)

EC2 is the fundamental unit of Cloud Computing.

**EC2 User Data**

* It is possible to bootstrap our instances using and EC2 User Data Script.
* Bootstrapping means launching a set of commands when a machine starts.
* The script is *only run once* at the *first start*.
* EC2 user data is used to automate boot tasks such as: - Installing updates, installing software, Downloading common files from the internet.
* EC2 User Data Script runs with the root user.

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 400NVMe SSD | Up to 10 Gbps | 4750 |
| R5.16xlarge | 64 | 512 | EBS Only | 20 Gbps | 13600 |
| M5.8xlarge | 32 | 128 | EBS Only | 10 Gbps | 6800 |

**NOTE:** t2.micro is a part of AWS free tier (up to 750hours/month)

**LAUNCHING AN EC2 INSTANCE:**

1. Login to AWS Console -> Instances -> Launch instance
2. Give the required details, such as – the name and tag, AMI (AWS Linux is the default one) and the instance type.
3. Generate a new key pair to that will be used to login to the EC2 instance. Give the key pair name, select the key pair type (RSA), and select the Private key file format (.pem (SSH) for mac, Linux or windows 10+ ; .ppk (putty) for windows OS version less than 10.
4. You can give the list of commands in the “user Data” section (probably a web page) that will launch at the time of launching the instance.
5. If you stop and start an instance again, the public IP might change.
6. The private IP will always remain the same.

**EC2 INSTANCE TYPES – OVERVIEW**

EC2 Instances types: <https://aws.amazon.com/ec2/instance-types/>

AWS has the following naming convention:  
  
For ex: m5.2xlarge  
 **m:** Instance class  
**5:** generation (AWS improves them over time)  
**2xlarge:** size within the instance class.

1. **General Purpose Instance Type:**Great for a diversity of workloads such as web servers or code repositories.   
   A good balance between compute, memory and networking
2. **Compute Optimized:**

Great for compute intensive tasks that require high performance processors:  
 Batch processing workloads  
 Media transcoding  
 High performing web servers  
 High performing Computing (HPC)  
 Scientific Modeling and machine learning  
 Dedicated gaming.

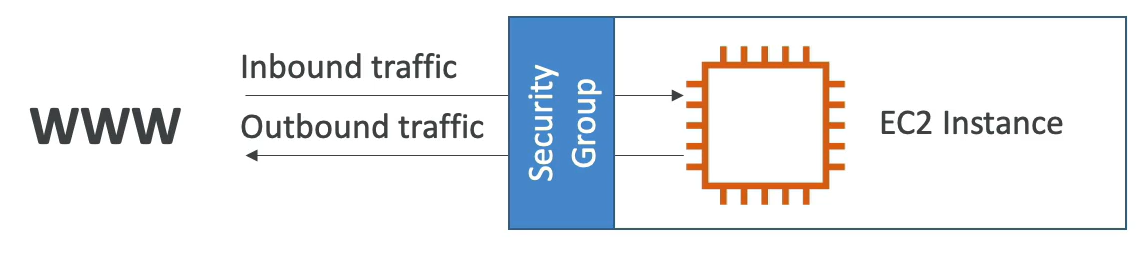
1. **Memory Optimized:**

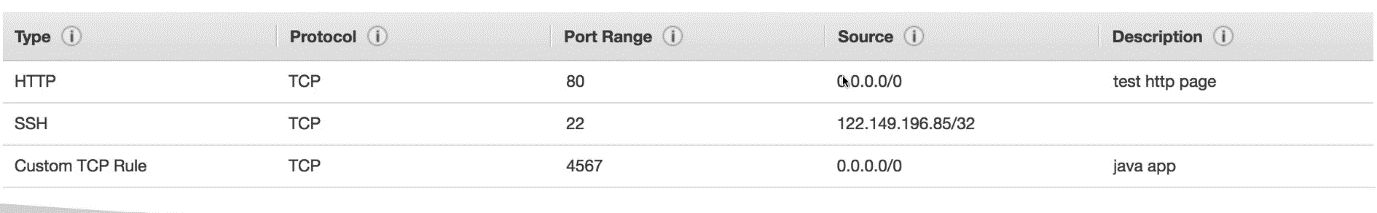
Fast performance for workloads that process large data sets in memory.  
Use cases:  
 High performance, relational, non-relational databases.  
 Distributed web scale cache store  
 In-memory databases optimized for BI (business intelligence)  
 Applications performing real-time processing of big unstructured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

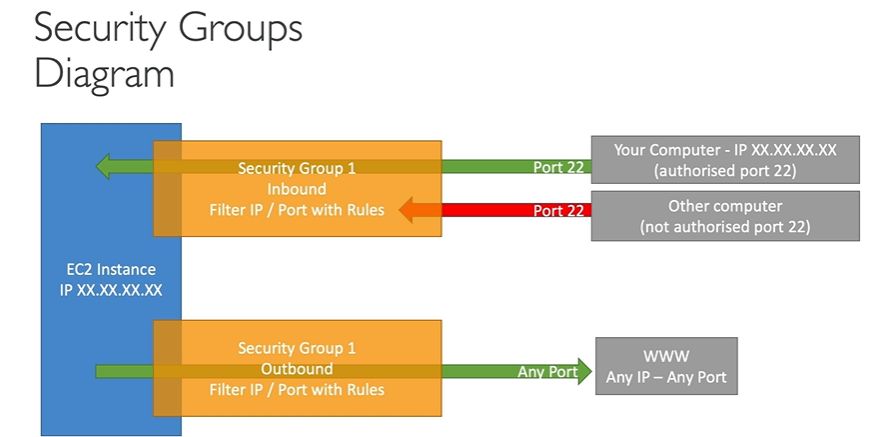
1. **Storage Optimized:**

Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage, like:  
  
High frequency Online transaction Processing (OLTP) systems  
Relational and NoSQL databases  
Cache for in-memory databases (Redis)  
Data warehousing applications  
Distributed File systems.

**SECURITY GROUPS**

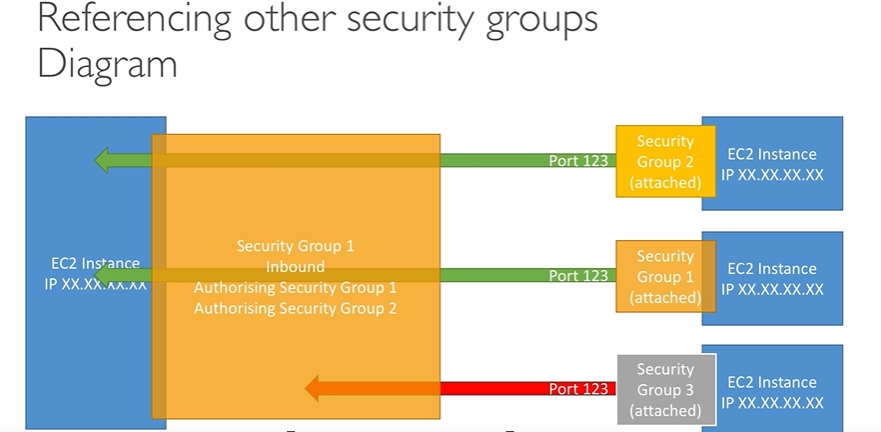
* Security groups (firewalls) are the fundamental of network security in AWS
* They control how traffic is allowed into or out of our EC2 instances  
    
  ****
* Security groups only contain allow rules.
* Security groups rules can reference by IP or by security groups.
* Security groups are acting as a “firewall” on EC2 instances.
* They regulate:
  + - Access to ports
    - Authorized IP ranges – IPv4 or IPv6
    - Control the inbound network (from other to the instance)
    - Control of outbound network (from the instance to other)





**NOTES:**

* SGs can be attached to multiple instances and similarly, an instance can have multiple SGs too.
* SGs are locked down to a region / VPC combination. So, if we switch from R1 to R2, we have to create new SGs
* SGs live “outside” the EC2 – if the traffic is blocked, the EC2 will not see it.
* It is good to maintain one separate SG for SSH access.
* If an application is not accessible (timed out), it is a security group issue.
* If your application gives a “connection refused” error, then it is an application error or it is not launched.
* By default, all inbound traffic is blocked
* By default, all outbound traffic is allowed



**CLASSIC PORTS TO KNOW**

1. 22: SSH (Secure Shell) – log into a Linux Instance
2. 21: FTP (File transfer Protocol) – upload files into a file share
3. 22: SFTP (Secure File Transfer Protocol) – upload file using SSH
4. 80: HTTP – access unsecured websites
5. 443: HTTPS – access secured websites
6. 3389: RDP (Remote Desktop Protocol) – log into a Windows instance

To access Security Group configuration, go to *Network and Security* -> *Security Groups.* It will show the list of all SGs created and the details following.

**NOTE:** 0.0.0.0/0 = Allows everything

**SSH** is a tool like Putty. SSH can be used on Mac, Linux OS or Windows10 and above. Putty can be used on all versions of Windows OS.

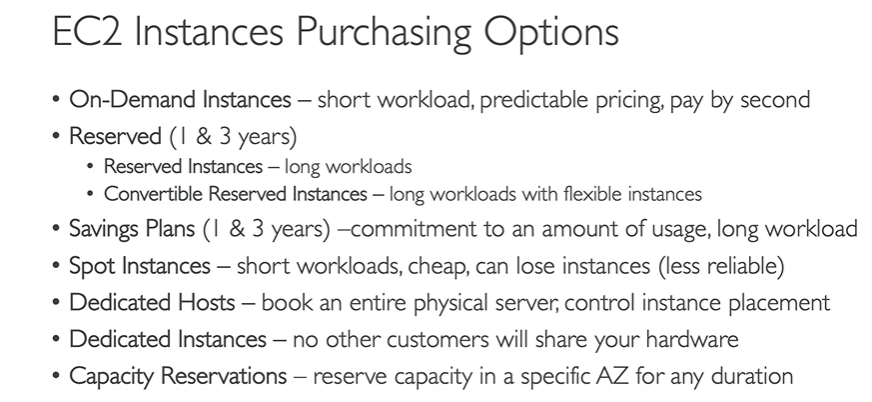
**EC2 Instance Connect** used a web browser to connect to the EC2 instances. It is valid for Mac, Linux, and Windows all versions.

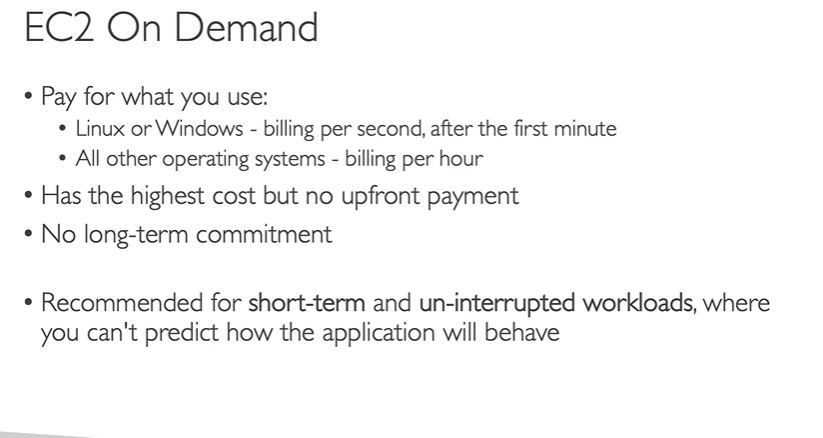
To use EC2 Instance Connect, go to your instance, and click on *Connect.* Here we could see multiple options – EC2 Instance Connect, Session manager, SSH Client, EC2 Serial Console

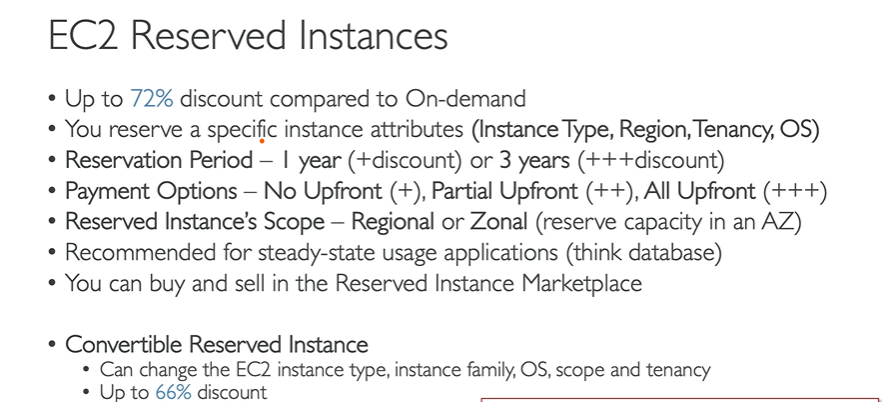
Under EC2 Instance Connect, choose the correct username and click on *Connect.*

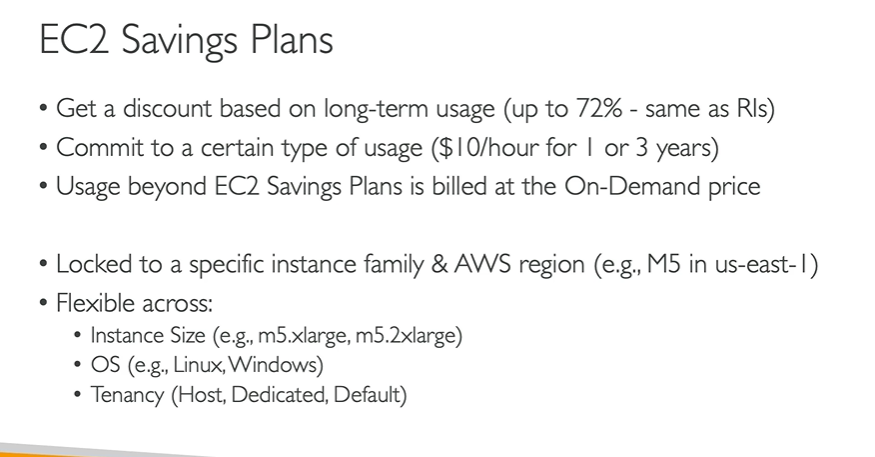
**IAM Roles Commands:**

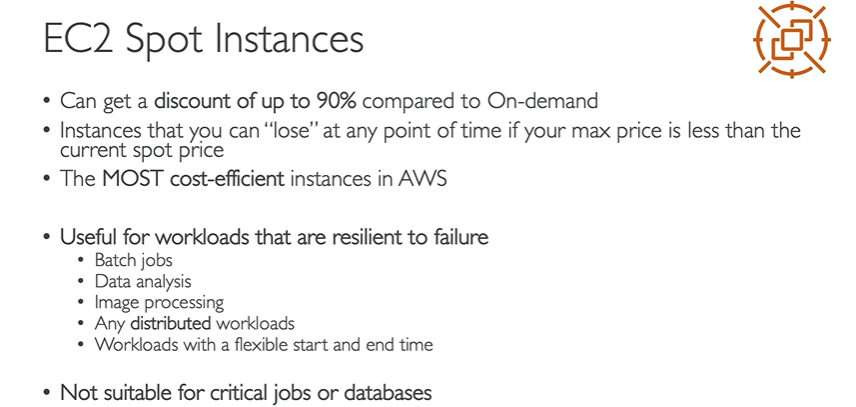
1. **aws --version:** shows current AWS version
2. **aws iam list-users:** displays the list of all the users
3. **aws configure:** To create credentials. However, creating creds using this command is highly discouraged as this can show the personal API and secret keys and anyone can then login.
4. Using IAM Roles in the AWS console is the best practice to create credentials.
5. IAM roles attached to an instance can be seen under *Security* tab
6. Search for IAM in the console and Roles. We need to attach the IAM roles to the instances as required.
7. For that, go to the instances -> Actions -> Security -> Modify IAM Role. Now choose the IAM roles from the dropdown.

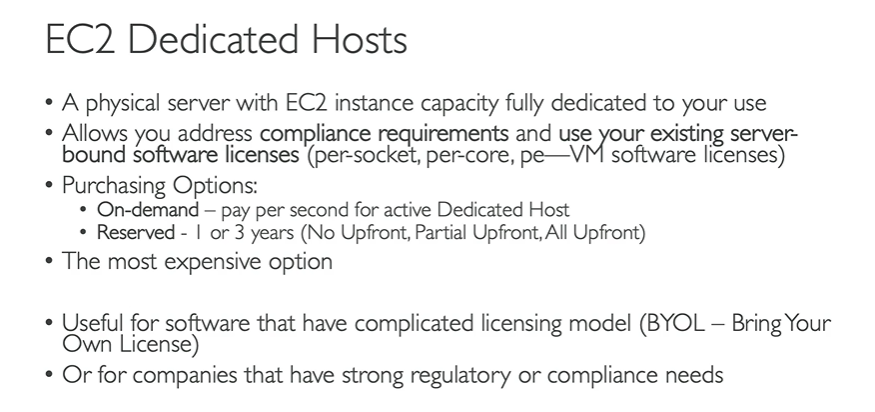


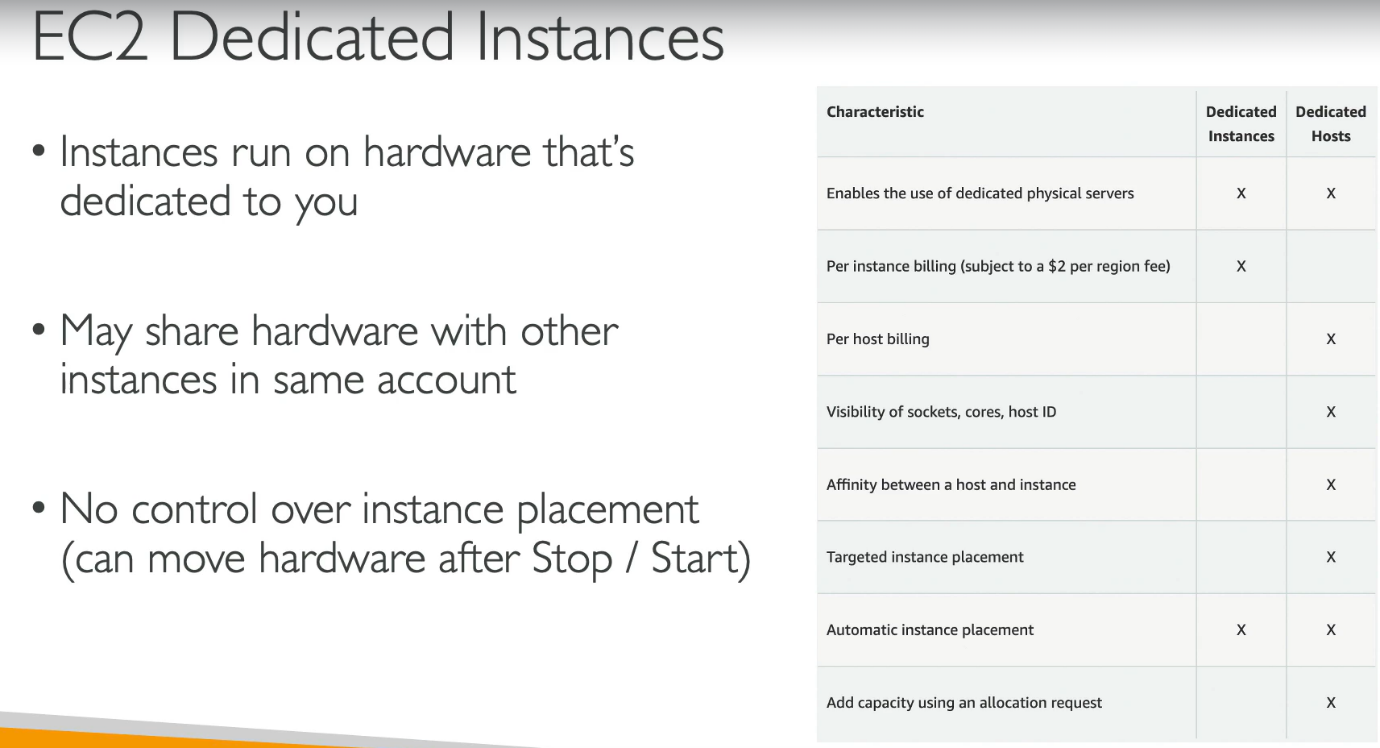


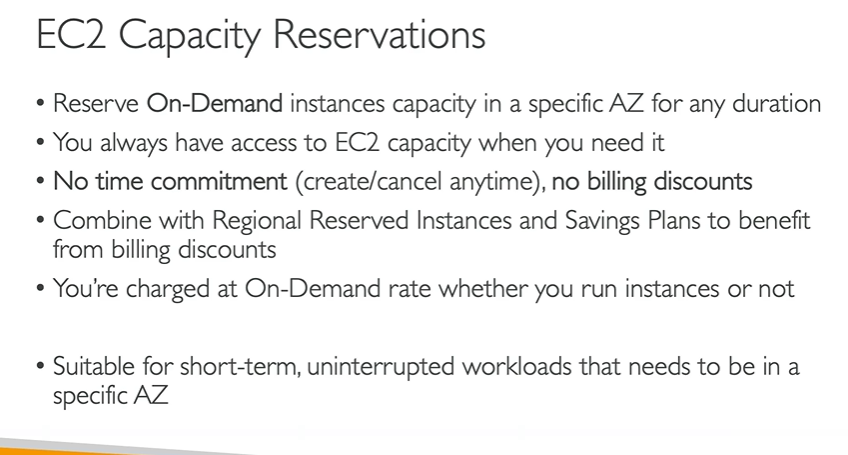


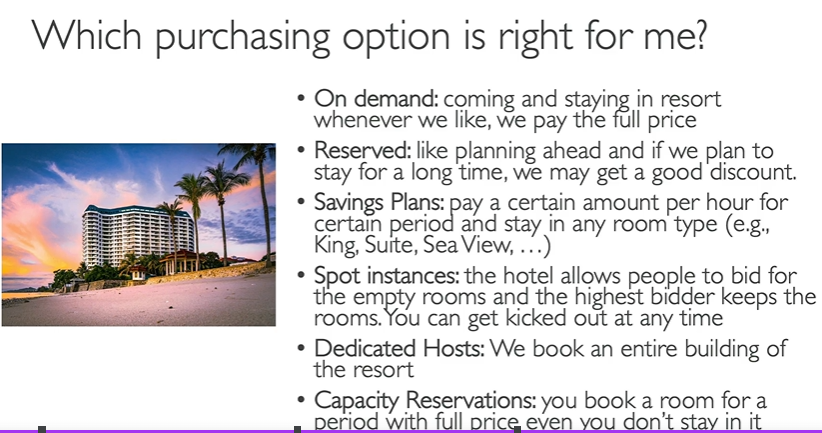














**EC2 SUMMARY:**

* **EC2 Instance:** AMI (The OS) + Instance Size (CPU + RAM) + Storage + Security Groups (firewalls) + EC2 User Data
* **Security Groups**: Firewalls 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 Hosts, Dedicated Instance