

# Cloud Computing Theory and Practice

INSY 5345 & INSY 4307

DR. SANTOSO BUDIMAN

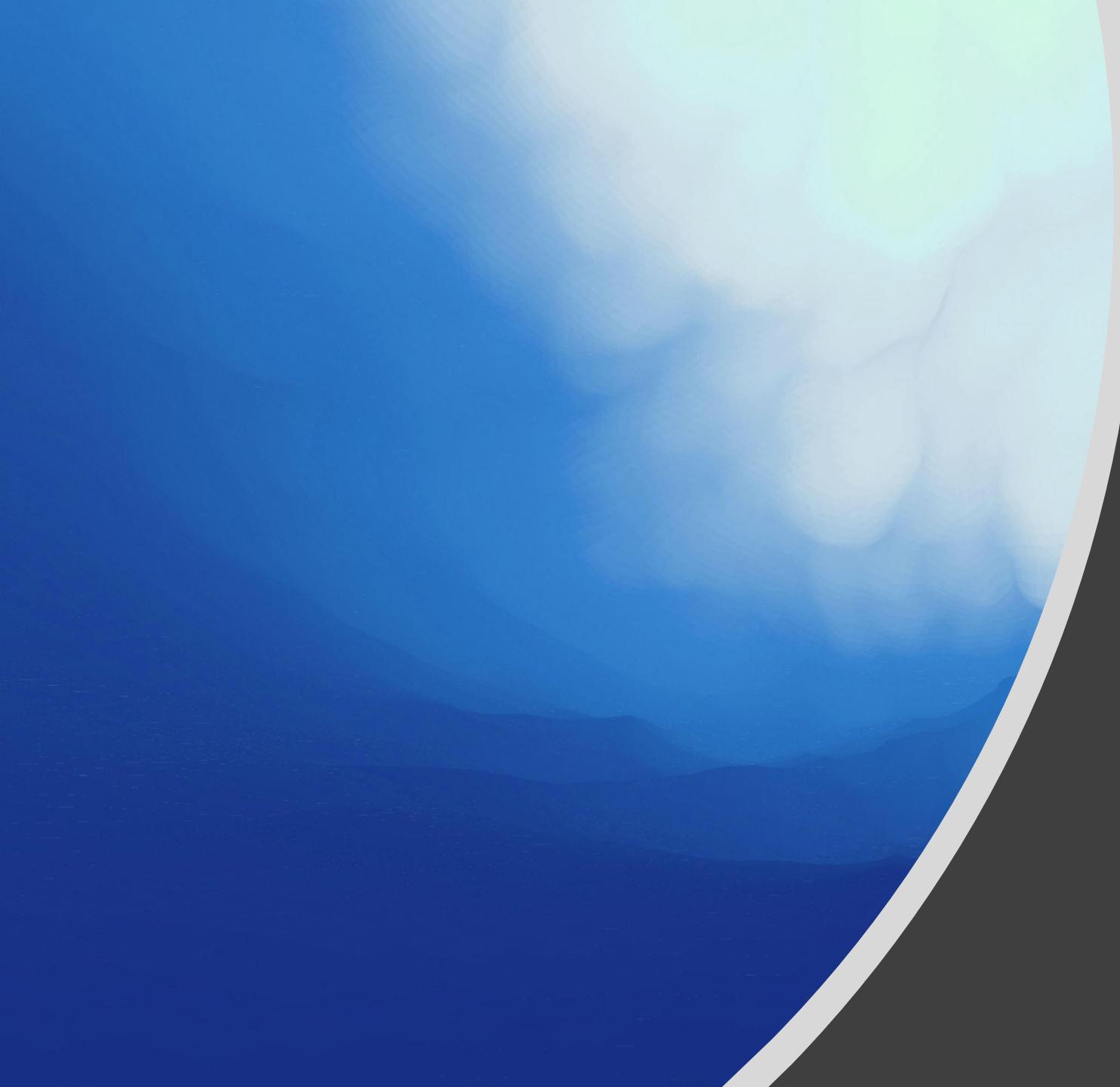


# Virtual Machines

SECTION 3

# Agenda

- What is an EC2 (AWS Virtual Machine)?
  - Instance Type
  - AMI
  - Security Groups
  - SSH and ICMP
- Launching EC2
- Ping test and SSH connection (access to EC2 for admin purposes)
- Installing a Web Server in EC2
- Bastion Host for security



# AWS Virtual Machine

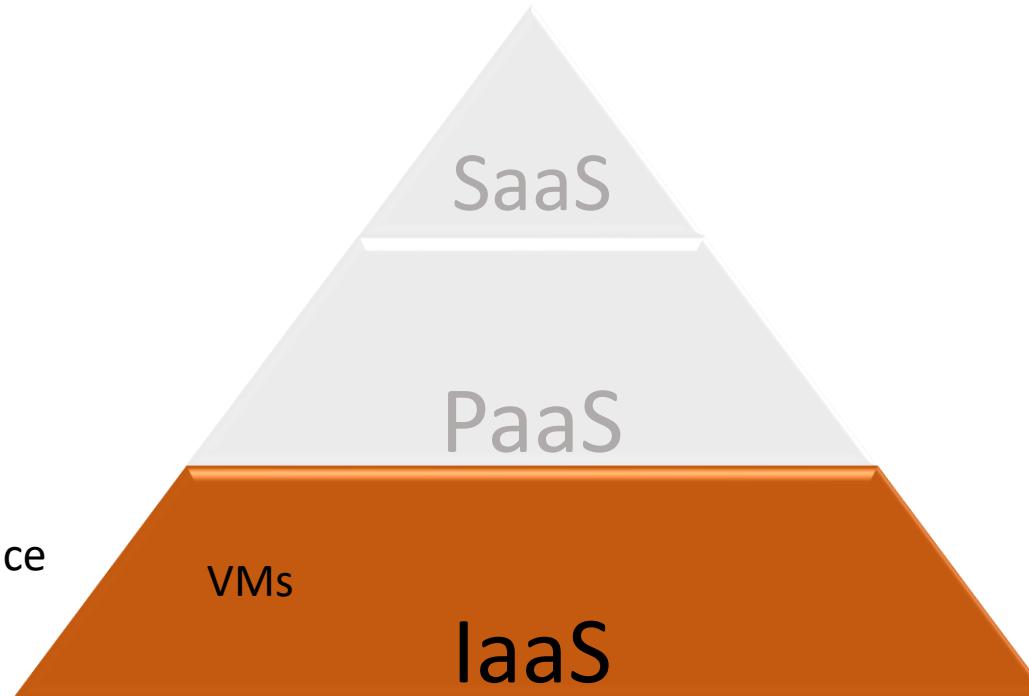
Elastic Compute Cloud (EC2)

# Cloud Computing – Service Model

SaaS = Software as a Service

PaaS = Platform as a Service

IaaS = Infrastructure as a Service





What is an  
EC2?

# AWS runtime compute choices

Virtual Machines (VMs)	Containers	Platform as a Service (PaaS)	Serverless	Specialized Solutions
 Amazon Elastic Compute Cloud (Amazon EC2)	 Amazon Elastic Container Service (Amazon ECS)	 AWS Elastic Beanstalk	 AWS Lambda	 AWS Outposts
 Amazon Lightsail			 AWS Fargate	 AWS Batch

Higher infrastructure control and customization

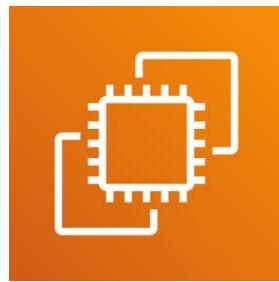
Faster application deployment

Fully managed services

Different compute services are available to meet the needs of different use cases.

This module will discuss Amazon EC2.

# Amazon EC2



Amazon Elastic  
Compute Cloud  
(Amazon EC2)

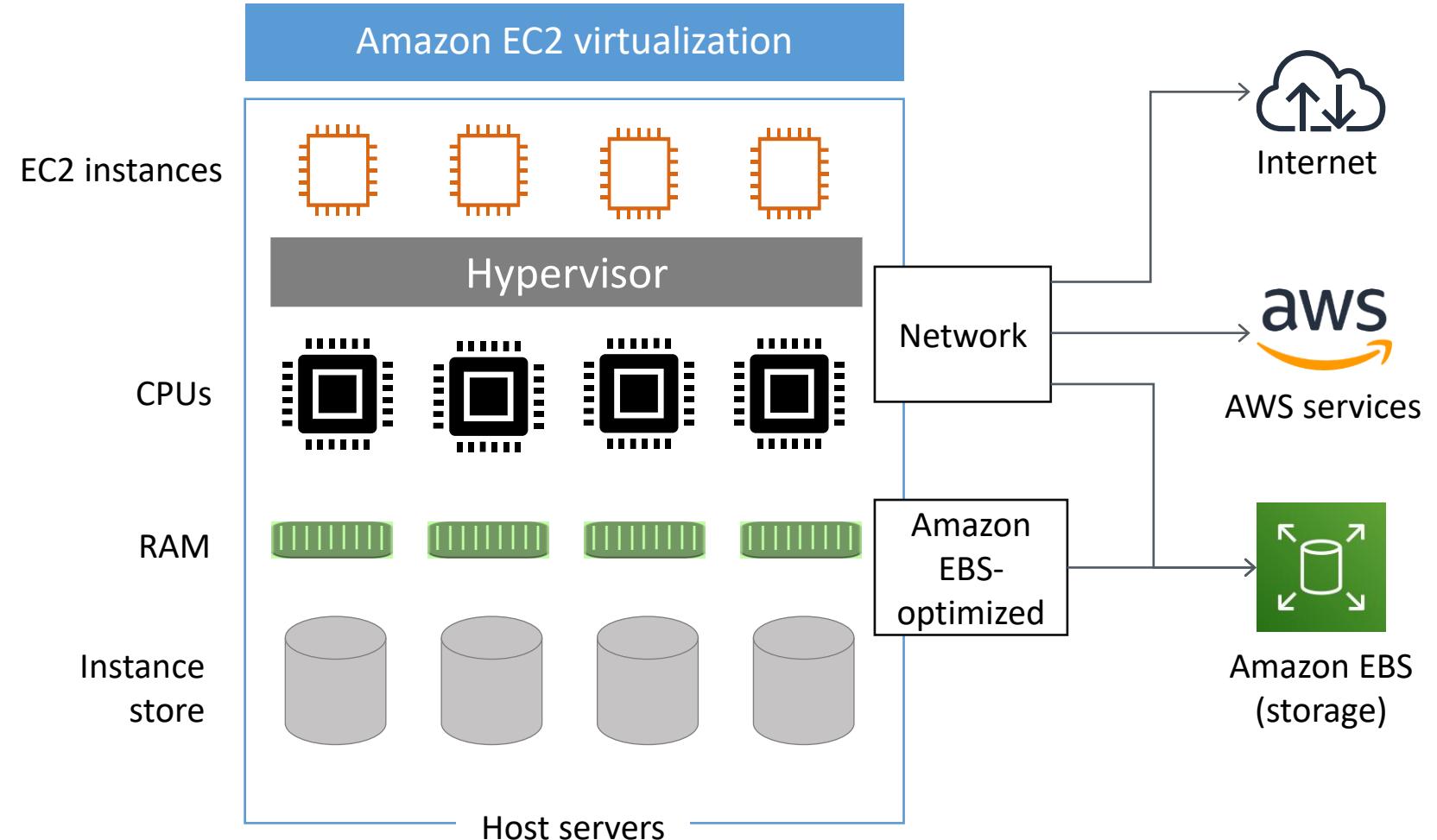
Amazon EC2 provides resizable compute capacity in the cloud.

- Provides virtual machines (servers)
- Provisions servers in minutes
- Can automatically scale capacity in or out as needed
- Pay only for the capacity used
- OS supported by EC2:
  - Microsoft Windows
  - Linux (various distributions) – we use mostly Ubuntu
- 400+ Instance types (CPU + Memory + Storage)

# EC2 instances

An EC2 instance is a **virtual machine** that runs on a physical host.

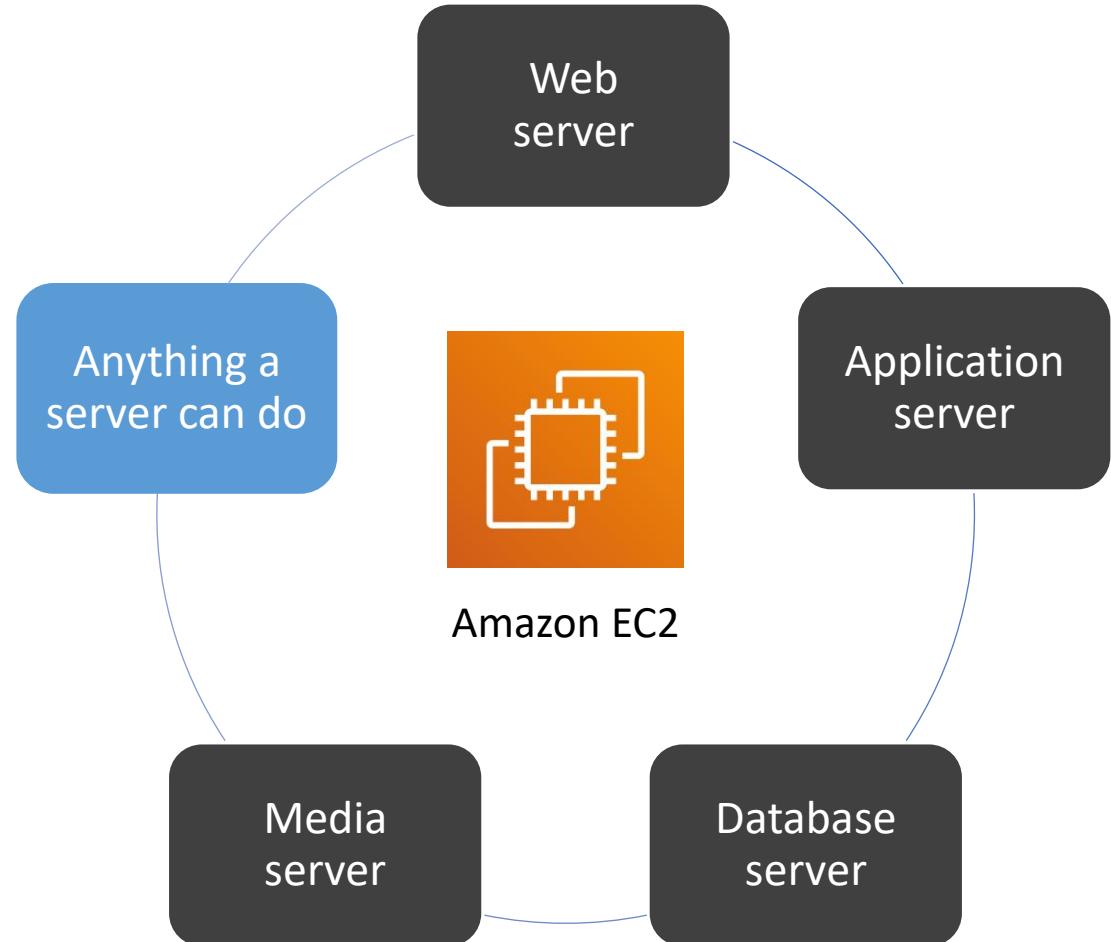
- You can choose different configurations of CPU and memory capacity
- Supports different storage options
  - [Instance store](#)
  - [Amazon Elastic Block Store \(Amazon EBS\)](#)
- Provides network connectivity



# Amazon EC2 use cases

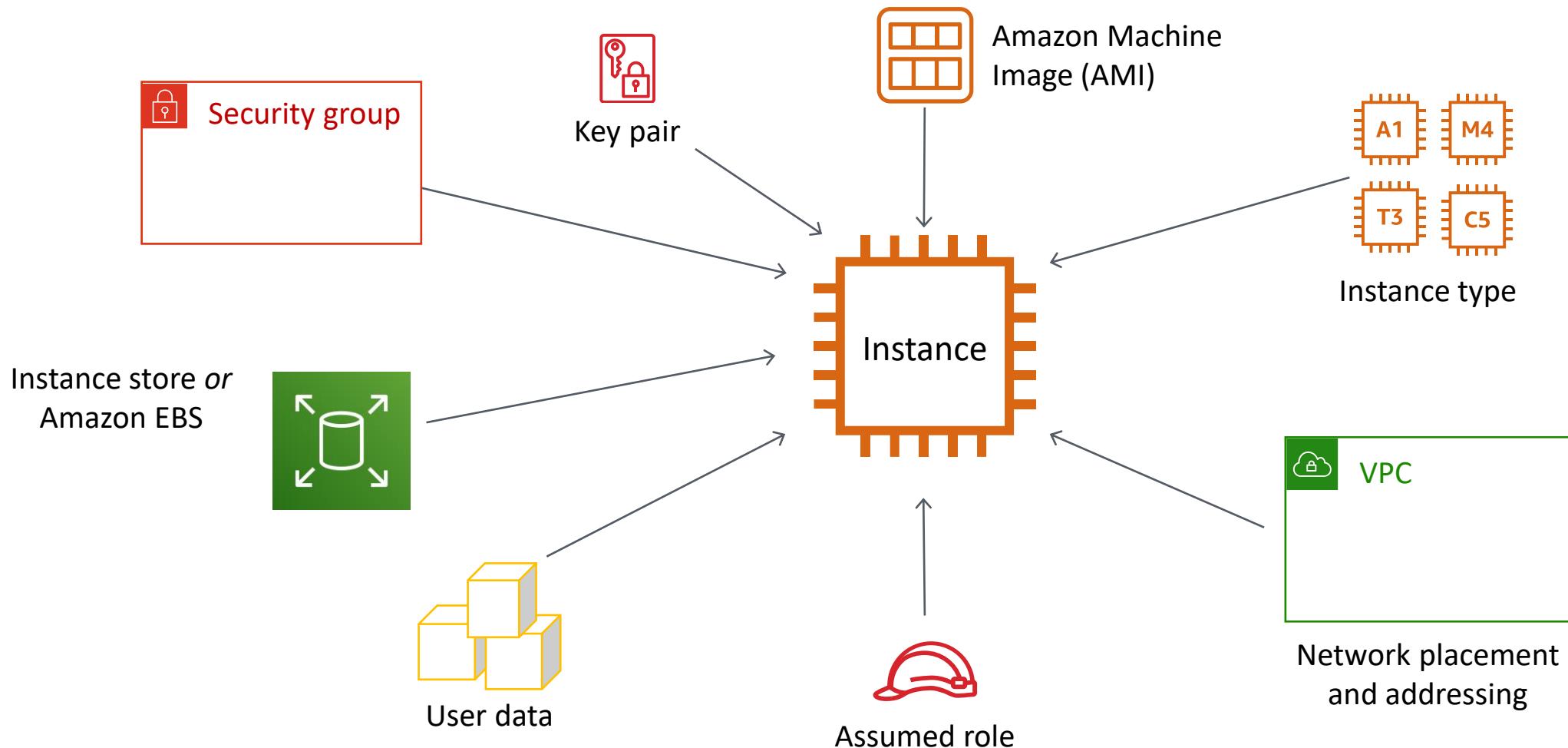
Use Amazon EC2 when you need:

- Complete control of your computing resources, including *operating system* and *processor type*
- Options for optimizing your compute costs –
  - *On-Demand Instances, Reserved Instances, and Spot Instances (spare EC2 capacity)*
  - *Savings Plans*
- Ability to run **any type of workload**, for example –
  - Simple websites
  - Enterprise applications
  - High performance computing (HPC) applications



# Provisioning an EC2 instance

Essential instance launch configuration parameters



---

# What is an AMI?

---

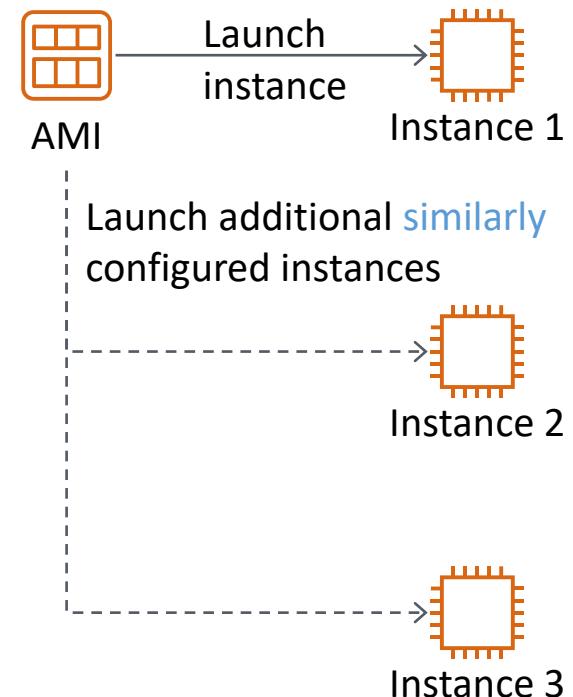
# Amazon Machine Image (AMI)



An **AMI** provides the information that is needed to launch an instance, including:

- A **template for the root volume**
  - Contains the guest operating system (OS) and perhaps other installed software
- **Launch permissions**
  - Control which AWS accounts can access the AMI
- **Block device mappings**
  - Specifies any storage volumes to attach to the instance

Create multiple instances from the same AMI



# AMI benefits

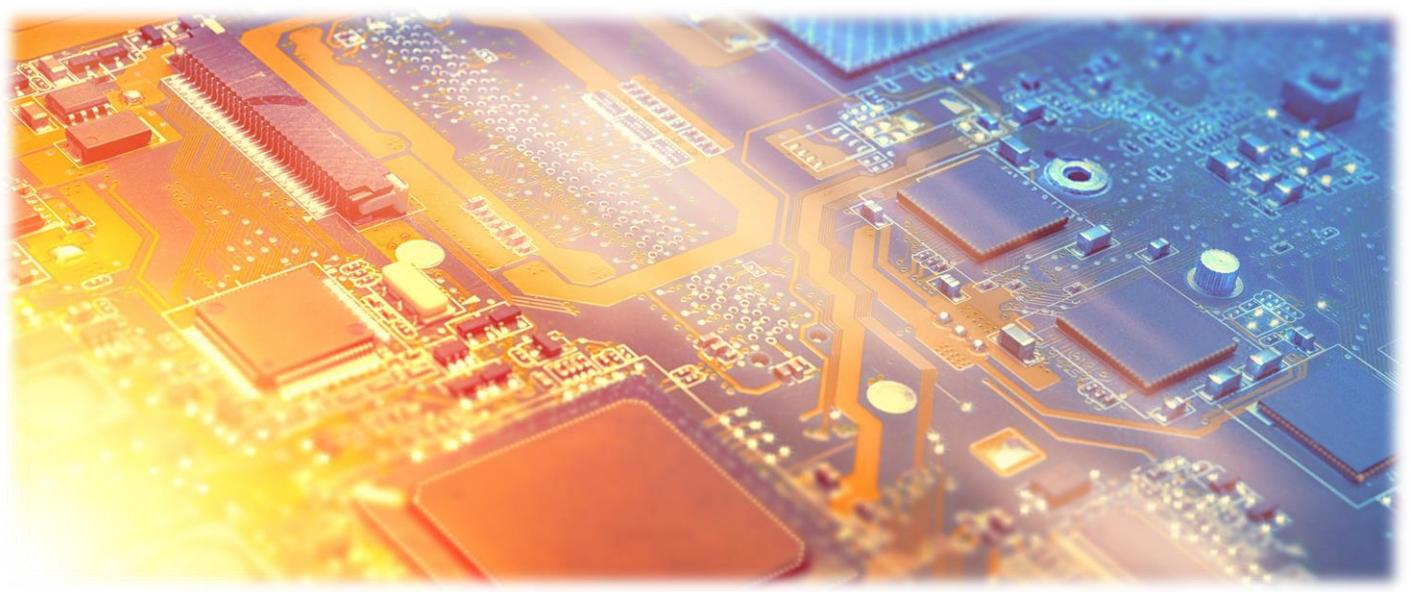
- Repeatability
  - An AMI can be used repeatedly to launch instances with efficiency and precision
- Reusability
  - Instances launched from the same AMI are identically configured
- Recoverability
  - You can create an AMI from a configured instance as a restorable backup
  - You can replace a failed instance by launching a new instance from the same AMI



# Choosing an AMI

Choose an AMI based on:

- Region
- Operating system
  - Microsoft Windows or Linux
- Storage type of the root device
- Architecture
- Virtualization type



AMI sources:

- [Quick Start](#) – *Linux and Microsoft Windows AMIs that are provided by AWS.*
- [My AMIs](#) – *Any AMIs that you create.*
- [AWS Marketplace](#) – *Pre-configured templates from third parties.*
- [Community AMIs](#) – *AMIs shared by others. Use at your own risk.*

# Amazon AMI - example

The screenshot shows the AWS Cloud9 'Create New Instance' wizard, specifically Step 1: Choose an Amazon Machine Image (AMI). The Red Hat option is selected, indicated by a yellow background and bold text. Other options listed are SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type and Ubuntu Server 18.04 LTS (HVM), SSD Volume Type. Each option includes details like Root device type (ebs), Virtualization type (hvm), and ENA Enabled status (Yes). On the right side, there are 'Select' buttons for each row and radio buttons for choosing 64-bit (x86) or 64-bit (Arm) architecture.

Step 1: Choose an Amazon Machine Image (AMI)

AMI Name	Description	Root device type	Virtualization type	ENA Enabled	Architecture
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type	ebs	hvm	Yes	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type	SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	ebs	hvm	Yes	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type	Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	ebs	hvm	Yes	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)

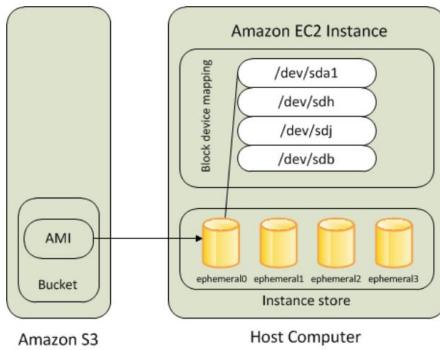
Elastic Network Adapter (ENA) - next generation network interface

Hardware Virtual Machine (HVM) –Type 1 Hypervisor

X86/ARM – Computer Architecture/CPU

EBS backed

# Instance Root Volume

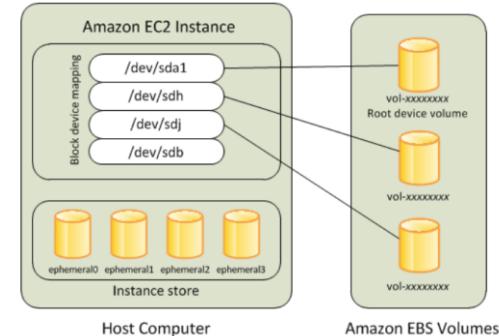


Note: Instance store is within the EC2 (RAM), and hence memory is erased when stopped.

## instance store-backed AMI

When an instance is launched, the *root device volume* contains the image used to boot the instance.

- ❑ When EC2 was introduced, all AMIs were backed by EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.
- ❑ After EBS is introduced, AMIs can be backed by EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
- Users can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS. But AMIs backed by EBS is recommended, because they launch faster and use persistent storage (can stop and start the instance).
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/RootDeviceStorage.html>

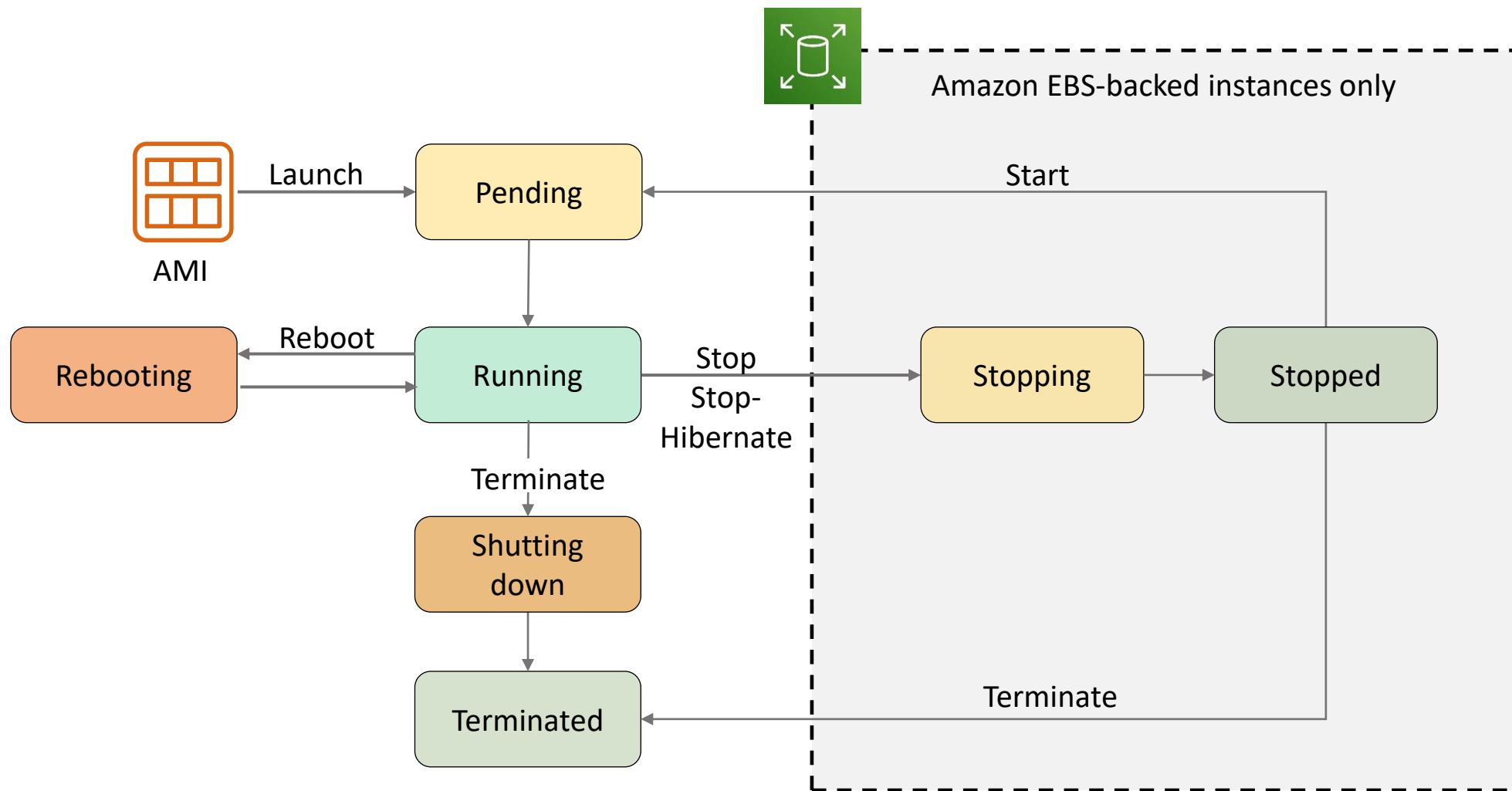


## EBS-backed AMI

# Instance store-backed versus Amazon EBS-backed AMI

Characteristic	Amazon EBS-Backed Instance	Instance Store-Backed Instance
Boot time for the instance	Boots faster	Takes longer to boot
Maximum size of root device	16 TiB	10 GiB
Ability to stop the instance	Can stop the instance	Can't stop the instance, only reboot or terminate it
Ability to change the instance type	Can change the instance type by stopping instance	Can't change the instance type because the instance can't be stopped
Instance charges	You are charged for instance usage, EBS volume usage, and storing your AMI as an EBS snapshot	You are charged for instance usage and storing your AMI in Amazon S3

# Amazon EC2 instance lifecycle



# Stop vs Stop-Hibernate

---

- When you stop your instance, it enters the stopping state, and then the stopped state. AWS does not charge usage or data transfer fees for your instance after you stop it, but storage for any Amazon EBS volumes is still charged. While your instance is in the stopped state, you can modify some attributes, like the instance type. When you stop your instance, the data stored in memory (RAM) is lost.
- When you stop-hibernate your instance, AWS signals the operating system to perform hibernation (suspend-to-disk), which saves the contents from the instance memory (RAM) to the Amazon EBS root volume. EC2 persists the instance's EBS root volume and any attached EBS data volumes. When the instance is started:
  - The EBS root volume is restored to its previous state
  - The RAM contents are reloaded
  - The processes that were previously running on the instance are resumed
  - Previously attached data volumes are re-attached and the instance retains its instance ID
  - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Hibernate.html>

# Stop-Hibernation option

1. Choose AMI    2. Choose Instance Type    3. Configure Instance    4. Add Storage    5. Add Tags    6. Configure Security Group    7. Review

## Step 3: Configure Instance Details

Domain join directory

 Create new directory

IAM role

 Create new IAM role

Shutdown behavior

Hibernation stops your instance and saves the contents of the instance's RAM to the root volume. You cannot enable hibernation after launch. [Learn more](#)

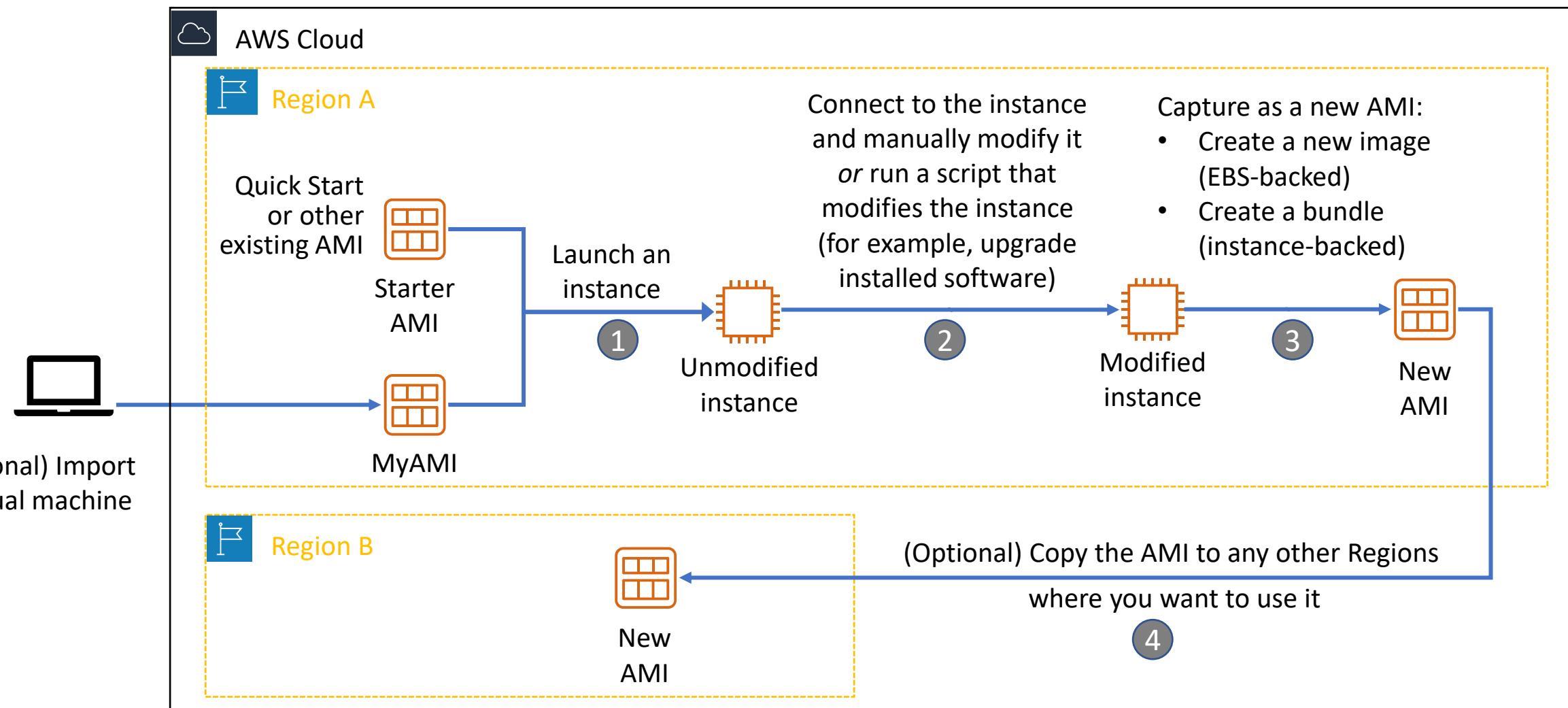
Stop - Hibernate behavior

  Enable hibernation as an additional stop behavior

Enable termination protection

  Protect against accidental termination

# Creating a new AMI



# EC2 Image Builder

EC2 Image Builder

EC2 Image Builder automates the *creation*, *management*, and *deployment* of up-to-date and compliant *golden VM images*.

- Provides a graphical interface to create image-building pipelines
- Creates and maintains *Amazon EC2 AMIs* and on-premises VM images
- Produces secure, validated, and up-to-date images
- Enforces version control

# EC2 Image Builder

The screenshot shows the AWS EC2 Image Builder console interface. On the left, a sidebar menu includes options like 'Image pipelines' (which is selected), 'Images', 'Saved configurations', 'Components', 'Image recipes', 'Container recipes', 'Infrastructure configuration', and 'Distribution settings'. Below the sidebar is a 'Documentation' link. The main content area has a header 'Getting started' under 'Image pipeline'. It explains that an image pipeline is an automation configuration for building Amazon Machine Images (AMIs) or Docker images on AWS. The configuration steps are numbered 1 through 3: 1. Create recipe (Base image and build/test components), 2. Define infrastructure configuration (optional) (Instance, VPC, IAM role, and other settings), and 3. Define distribution settings (optional) (Regions for output AMI and Docker image distribution). At the bottom of this section, it says 'No image pipelines available' and has a 'Create image pipeline' button. To the right of the main content area, there is a vertical sidebar with numbered items 6 through 11, likely representing steps or sections related to the pipeline creation process.

us-west-2.console.aws.amazon.com/imagebuilder/home?region=us-west-2#/

File 6

Services 7

Search for services, blogs, docs, and more 8

[Alt+S] 9

Oregon 10

UTASantosoAWS 11

EC2 Image Builder 12

Image pipelines 13

Images 14

Saved configurations 15

Components 16

Image recipes 17

Container recipes 18

Infrastructure configuration 19

Distribution settings 20

Documentation 21

Getting started 22

Image pipeline 23

An Image pipeline is an automation configuration for building Amazon Machine Images (AMIs) or Docker images on AWS. The configuration steps include:

1 Create recipe  
Base image and build/test components  
A recipe is a document that defines components to be applied to the base image to produce the desired configuration for the output image.

2 Define infrastructure configuration (optional)  
Instance, VPC, IAM role, and other settings  
Specify the infrastructure configurations for the instances that will run in your AWS account.

3 Define distribution settings (optional)  
Regions for output AMI and Docker image distribution  
Enter the AWS Regions for output AMI and Docker image distribution, and the AWS accounts that can launch the AMIs.

Image pipelines (0)

Find pipelines by name

Any status

Any type

View details

Actions

Create image pipeline

No image pipelines available

Create image pipeline

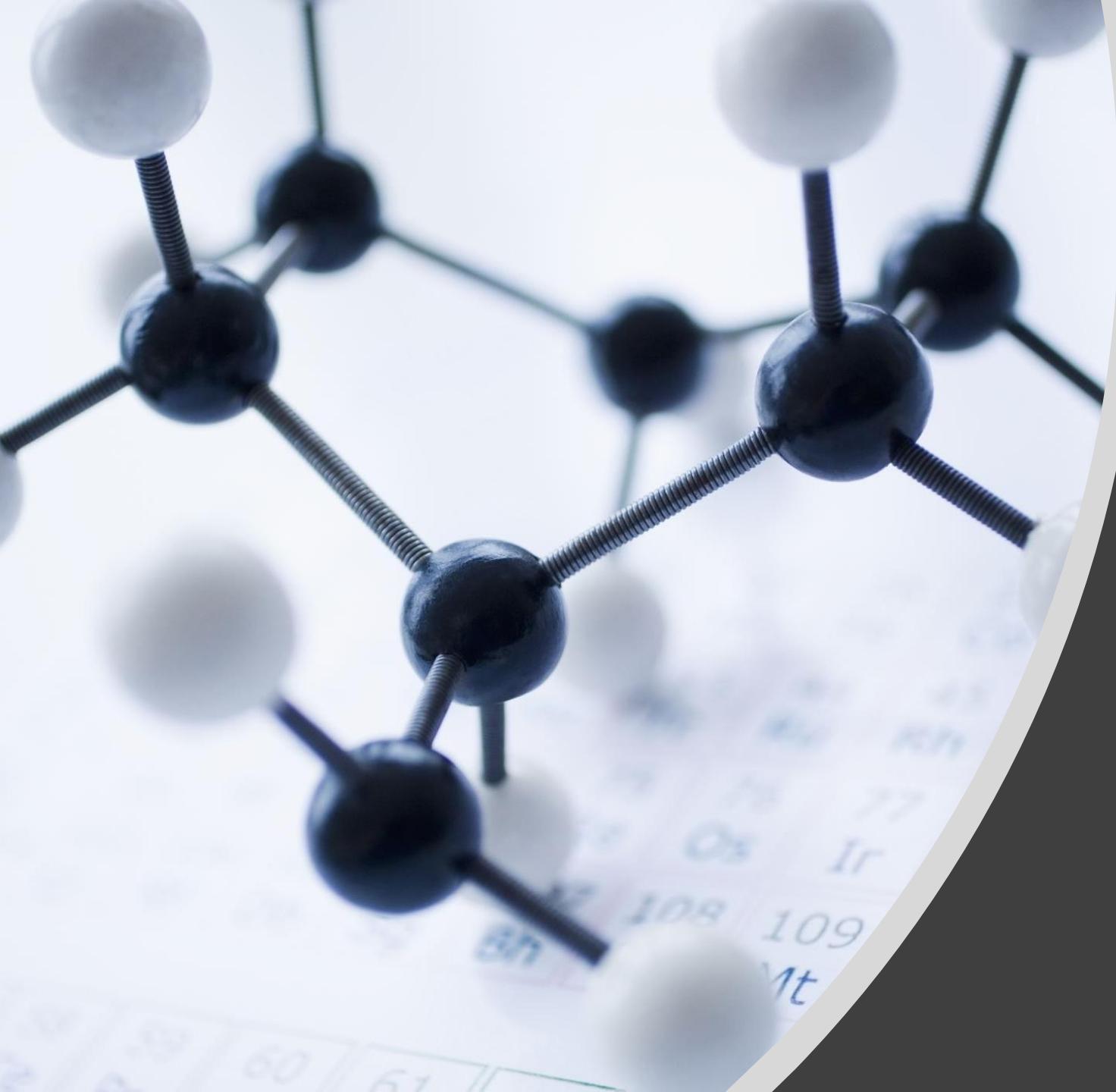
# AMI key takeaways

An AMI provides the information that is needed to launch an EC2 instance

For best performance, use an AMI with HVM virtualization type

Only an instance launched from an Amazon EBS-backed AMI can be stopped and started

An AMI is available in a Region

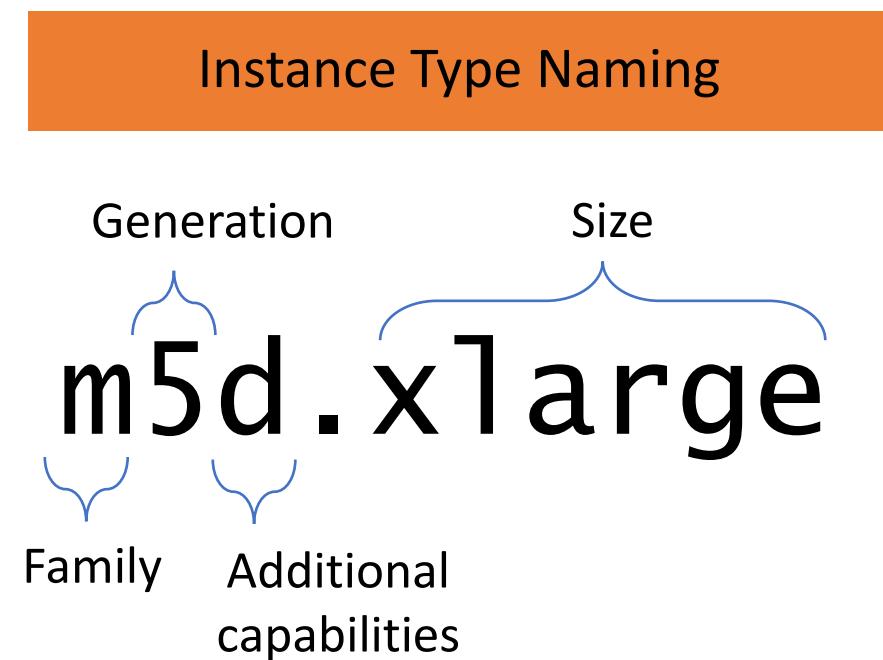


# Selecting an EC2 instance type

# EC2 instance type

An [EC2 instance type](#) defines a configuration of CPU, memory, storage, and network performance characteristics that provides a given level of compute performance.

vCPU	 4
Memory	 16 GiB
Storage	 1 x 150 NVMe SSD
Network performance	 Up to 10 Gbps



# AWS Instance types

The screenshot shows the AWS EC2 instance creation wizard at Step 2: Choose an Instance Type. The user has selected the 'All instance types' filter. The table lists various instance types, including t2.nano, t2.micro (highlighted with a red box and labeled 'Free tier eligible'), t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge, t3a.nano, t3a.micro, t3a.small, t3a.medium, and t3a.large. The t2.micro row is highlighted with a red box and labeled 'Free tier eligible'. The table includes columns for Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support.

Currently	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	<b>t2.micro</b> Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
	t2.small	1	2	EBS only	-	Low to Moderate	Yes
	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Filter by: All instance types Current generation Show/Hide Columns

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Cancel Previous Review and Launch Next: Configure Instance Details

GiB = Gibibytes (base 1024)  
GB = Gigabytes (base 1000)

# AWS vCPU

- Amazon EC2 instances support multithreading.
- **Each thread** is represented as a virtual CPU (**vCPU**) on the instance.
- An instance type has a default number of CPU cores, can be adjusted in some instance types.
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-optimize-cpu.html#cpu-options-supported-instances-values>

Instance type	Default vCPUs	Default CPU cores	Default threads per core	Valid number of CPU cores	Valid number of threads per core
m5n.16xlarge	64	32	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
m5n.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
t3.nano	2	1	2	1	1, 2
t3.micro	2	1	2	1	1, 2
t3.small	2	1	2	1	1, 2
t3.medium	2	1	2	1	1, 2
t3.large	2	1	2	1	1, 2
t3.xlarge	4	2	2	2	1, 2
t3.2xlarge	8	4	2	2, 4	1, 2

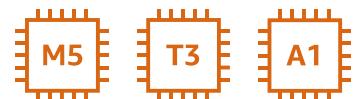
# Suitability of instance types for workloads (1 of 2)



## General purpose instance types

- Web or application servers
- Enterprise applications
- Gaming servers
- Caching fleets
- Analytics applications
- Development or test environments

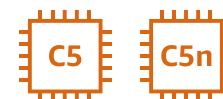
Example instance types:



## Compute optimized instance types

- Batch processing
- Distributed analytics
- High performance computing (HPC)
- Ad server engines
- Multiplayer gaming
- Video encoding

Example instance types:



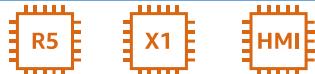
# Suitability of instance types for workloads (2 of 2)



## Memory optimized instance types

- In-memory caches
- High-performance databases
- Big data analytics

Example instance types:



## Accelerated computing instance types

- Machine learning, artificial intelligence (AI)
- HPC
- Graphics

Example instance types:



## Storage optimized instance types

- High-performance databases<sup>1</sup>
- Real-time analytics<sup>1</sup>
- Transactional workloads<sup>1</sup>
- NoSQL databases<sup>1</sup>
- Big data<sup>2</sup>
- Data warehouse<sup>2</sup>
- Log processing<sup>2</sup>

<sup>1</sup>High I/O example instance type:



<sup>2</sup>Dense Storage example instance types:



# Graphics Processing Units (GPU)

---

Graphics Processing Units (GPUs)

Can process many pieces of data simultaneously

GPU manufacturers: Nvidia, AMD, and Intel

Mostly used in gaming and artificial intelligence

Many deep learning technologies rely on GPUs working in conjunction with CPUs.

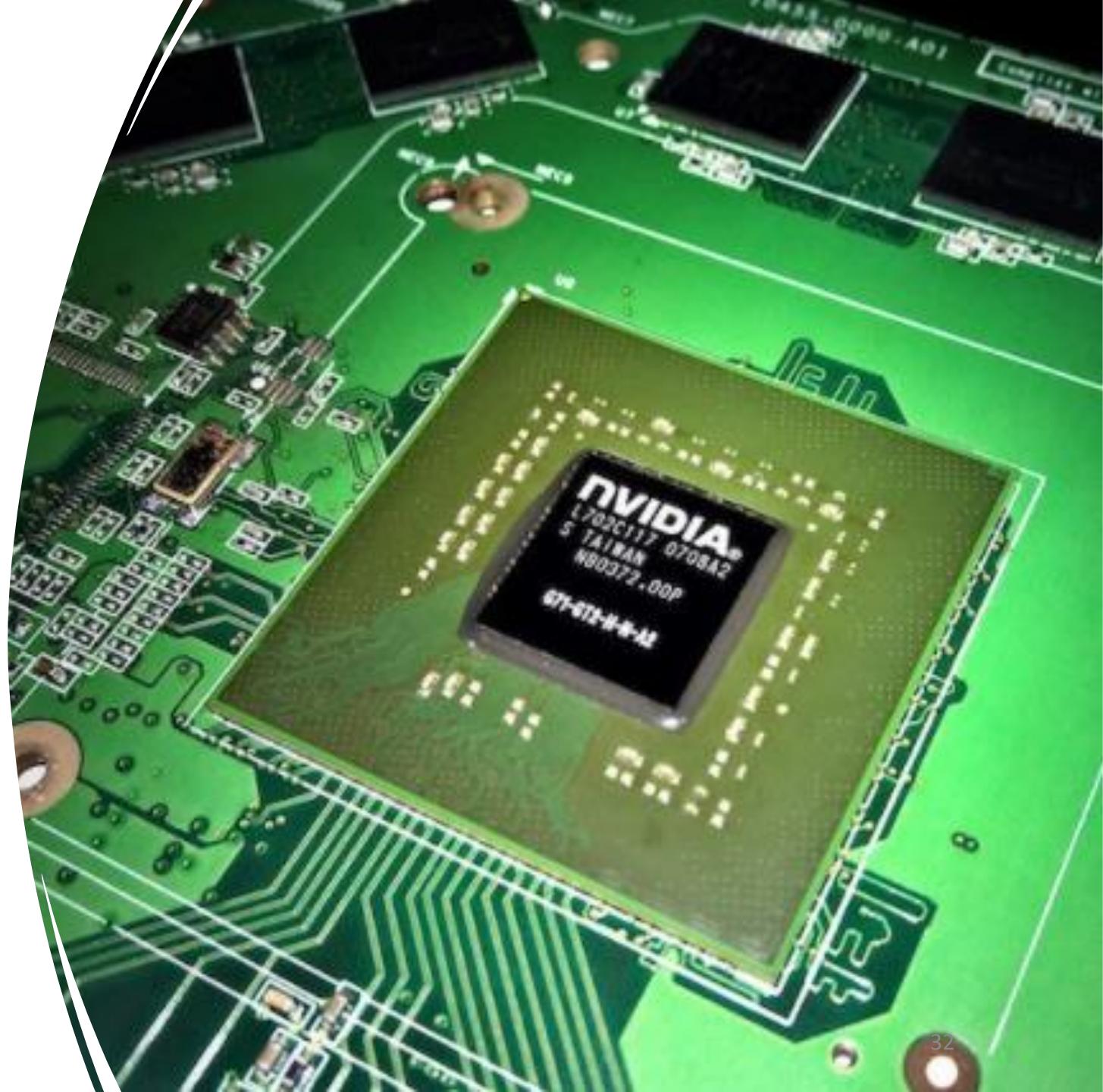
In some cases, **GPU** is 4-5 times **faster than CPU**

A GPU typically has more *logical* cores (arithmetic logic units or ALUs, control units and memory cache) than a CPU.

A **GPU** is **typically** not a standalone platform, but a **co-processor hosted by a CPU**.

**Training phase** of the **deep learning model** is the most resource-intensive task. GPU can make this phase faster.

- Small Deep Learning will be fine with CPU.



← → ⌂ 🔒 aws.amazon.com/ec2/instance-types/ 🔍 ⭐ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

**aws**

Contact Sales Support English My Account Sign In to the Console

re:Invent Products Solutions Pricing Documentation Learn Partner Network AWS Marketplace Customer Enablement Explore More Q

Amazon EC2 Overview Features Pricing Instance Types ▾ FAQs Getting Started Resources ▾

## Accelerated Computing

**General Purpose**

**Compute Optimized**

**Memory Optimized**

**Accelerated Computing**

**Storage Optimized**

**Instance Features**

**Measuring Instance Performance**

**P4** **P3** **P2** **Inf1** **G4dn** **G4ad** **G3** **F1**

Amazon EC2 P4 instances are the latest generation of GPU-based instances and provide highest performance for machine learning training and high performance computing in the cloud.

**Features:**

- Up to 8 NVIDIA A100 Tensor Core GPUs
- 400 Gbps instance networking with support for Elastic Fabric Adapter (EFA) and NVIDIA GPUDirect RDMA (remote direct memory access)
- 600 GB/s peer to peer GPU communication with NVIDIA NVSwitch
- Deployed in EC2 UltraClusters consisting of more than 4,000 NVIDIA A100 Tensor Core GPUs, Petabit-scale networking, and scalable low latency storage with Amazon FSx for Lustre
- 3.0 GHz 2nd Generation Intel Xeon Scalable (Cascade Lake) processors

Instance	GPUs	vCPUs	Mem(GiB)	Network Bandwidth	GPUDirect RDMA	GPU Peer to Peer	Storage	EBS Bandwidth
p4d.24xlarge	8	96	1152	400 Gbps ENA and EFA	Yes	600GB/s NVSwitch	8 x 1 TB NVMe SSD	19 Gbps

# Amazon EC2 Instance Type and Features

- Various combinations of **CPU, memory, networking, storage (400+ types)**
- CPU (Intel, AMD, AWS Graviton)
- Accelerators (GPU, FPGA (Field Programmable Gate Arrays)): NVIDIA, XILINX, etc.
- 2 types of instances within EC2 ecosystem:
  - Current-generation instances
  - Previous-generation instances (kept for backward compatibility)
- As a recommendation, choose new generation instance types in a family because they generally have better price-to-performance ratios
- Each instance type supports either (depending on the (AMI) used to launch it):
  - **paravirtual (PV) – older technology** used in previous AWS instance types, **no longer supported** in the current instance types.
  - **hardware virtual machine (HVM)** – supported by **all current instance types**.
    - Use **bare-metal (Type I) Hypervisor**
- [https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization\\_types.html](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/virtualization_types.html)

# Choosing an instance type

- Choose the instance type that meets –
  - The [performance needs](#) of your application
  - Your [cost requirements](#)
- When you create a *new* instance –
  - In the EC2 console, use the [Instance Types](#) page to filter by characteristics that you choose
  - Recommendation: The latest generation in an instance family typically has a better price-to-performance ratio
- If you have an *already existing* instance –
  - You can get recommendations for optimizing the instance type by using the [AWS Compute Optimizer](#)
  - You can evaluate recommendations and modify the instance accordingly

[how do you choose the correct one?](#)



# AWS Compute Optimizer

AWS Compute Optimizer

- Recommends *optimal instance type*, *instance size*, and Auto Scaling group configuration
- Analyzes workload patterns and makes recommendations
- Classifies instance findings as Under-provisioned, Over-provisioned, Optimized, or None

The screenshot shows the AWS Compute Optimizer dashboard under the 'Recommendations for EC2 instances' section. It displays 8 recommendations for over-provisioned instances across the US East (N. Virginia) region. The table includes columns for Instance ID, Instance name, Finding, Current instance type, Current On-Demand price, and Recommended instance type. The findings are all categorized as 'Over-provisioned'. The recommended instance types range from t3.large to r5.large.

Instance ID	Instance name	Finding	Current instance type	Current On-Demand price	Recommended instance type
i-0218a45abd8b53658	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
i-069f6e837890db127	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
i-07084b94d1bcf391b	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
i-0af9322ff627d7e8f	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
i-0ceb95ed248026d24	-	Over-provisioned	m5.xlarge	\$0.192 per hour	r5.large
i-0f277818dfef522e9	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large
i-0f4f4c06ad8afe81a	-	Over-provisioned	m5.2xlarge	\$0.384 per hour	r5.large
i-0fb9323080785de1e	-	Over-provisioned	c5.xlarge	\$0.17 per hour	t3.large

# Using user data to configure an EC2 instance



# User Data Field

As a note: When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

Linux commands can be written in the User Data field.

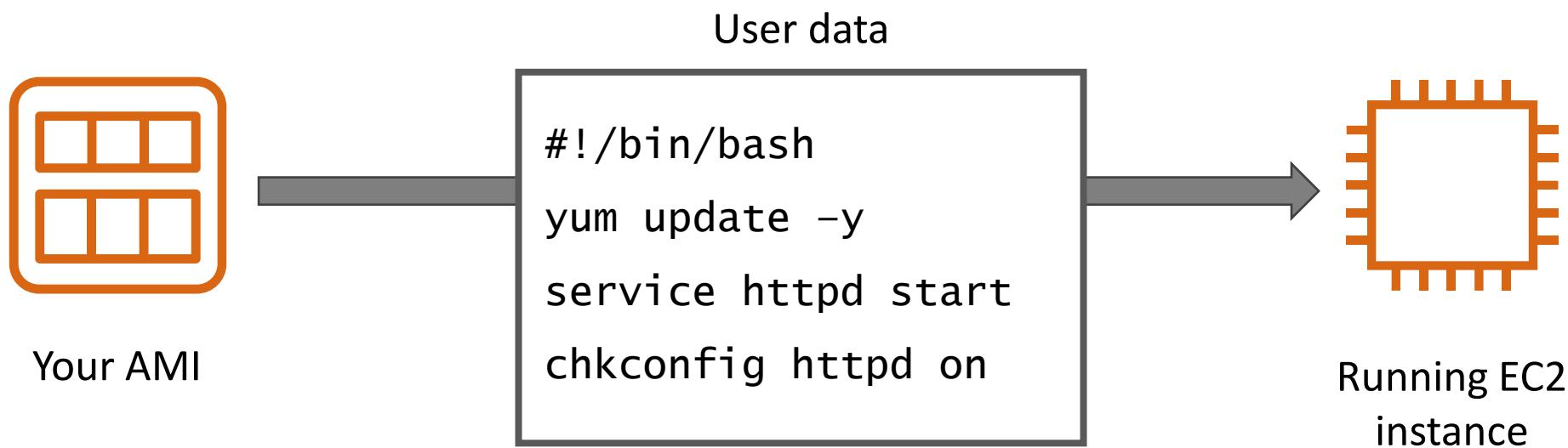
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

These set of linux commands can also be part of AMI (we will do later).

Note: when creating a Launch Template for automatic scaling, you can either create an AMI which include executed linux commands or use User Data.

# EC2 instance user data

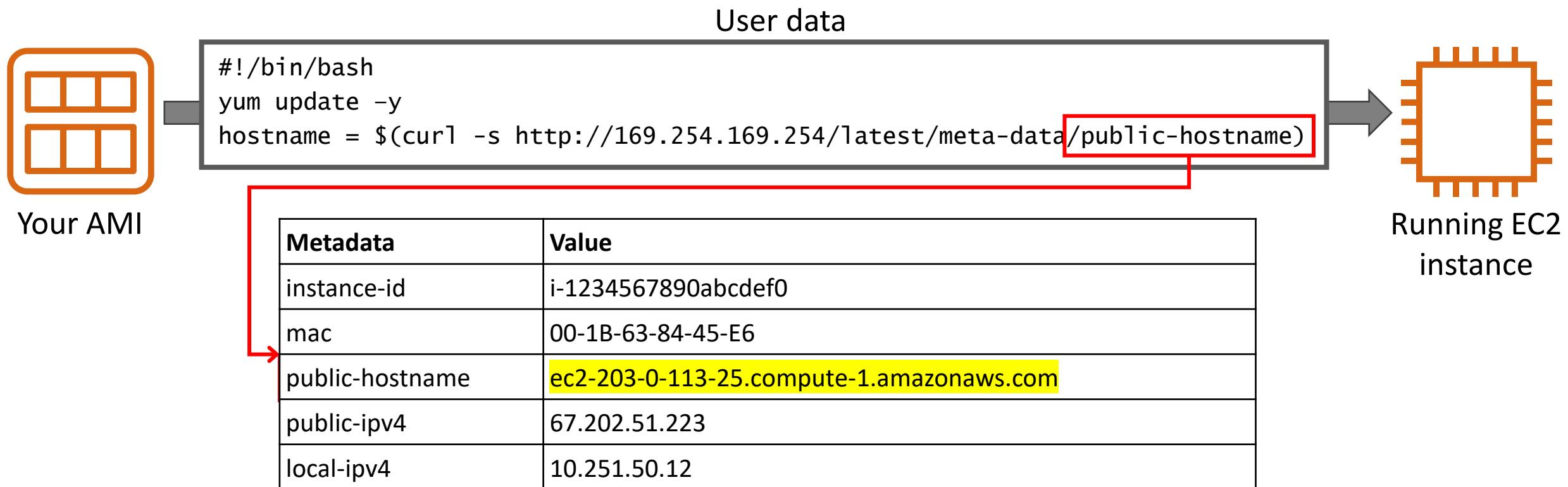
When you launch an EC2 instance, specify [user data](#) to run an initialization script (shell script or *cloud-init* directive).



# Retrieving instance metadata

Instance metadata is **information about your instance**.

- Is accessible from your instance at URL: `http://169.254.169.254/latest/meta-data/`
- Can be retrieved from a user data script

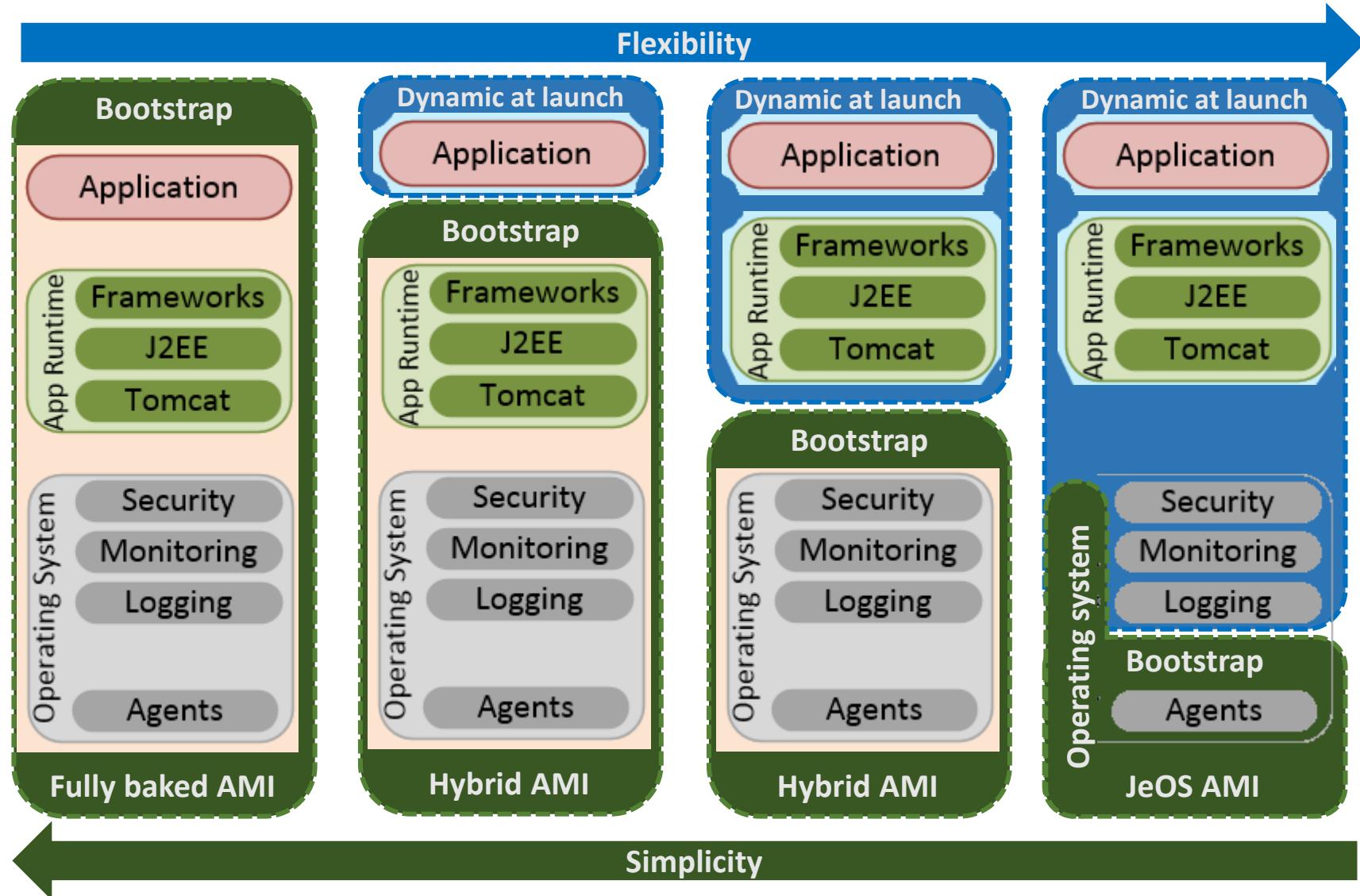


# 3 ways to install application

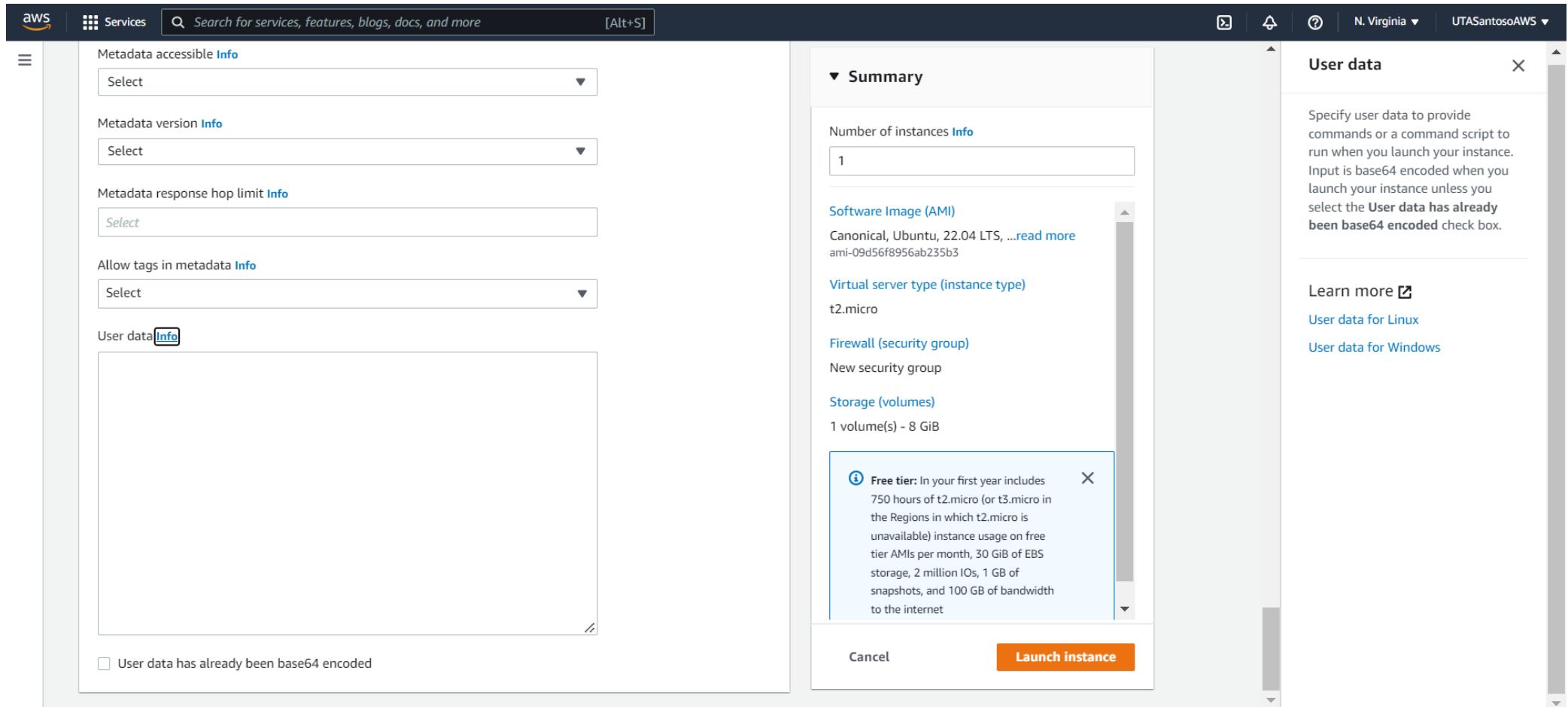
Let's say you want to install an application in an EC2, you have 3 ways to do it:

1. Launch an EC2 with the basic AMI. After it is up, install the application.
2. Launch an EC2 with the basic AMI and add the installation commands in the User Data field.
3. Create an AMI with the application in it, then launch an EC2 with this AMI.

# Configuring an EC2 instance: AMI versus user data



# User Data field in “Advanced details”



# Adding storage to an Amazon EC2 instance

# Amazon EC2 storage overview

## Root volume

This volume always contains the guest OS



Instance store



Amazon EBS  
(SSD-backed only)



An EC2 instance will *always* have a [root volume](#),  
and can *optionally* have one or more [data volumes](#).

## Data volumes

For data accessed by a single instance



Instance store



Amazon EBS

For data accessible from multiple instances



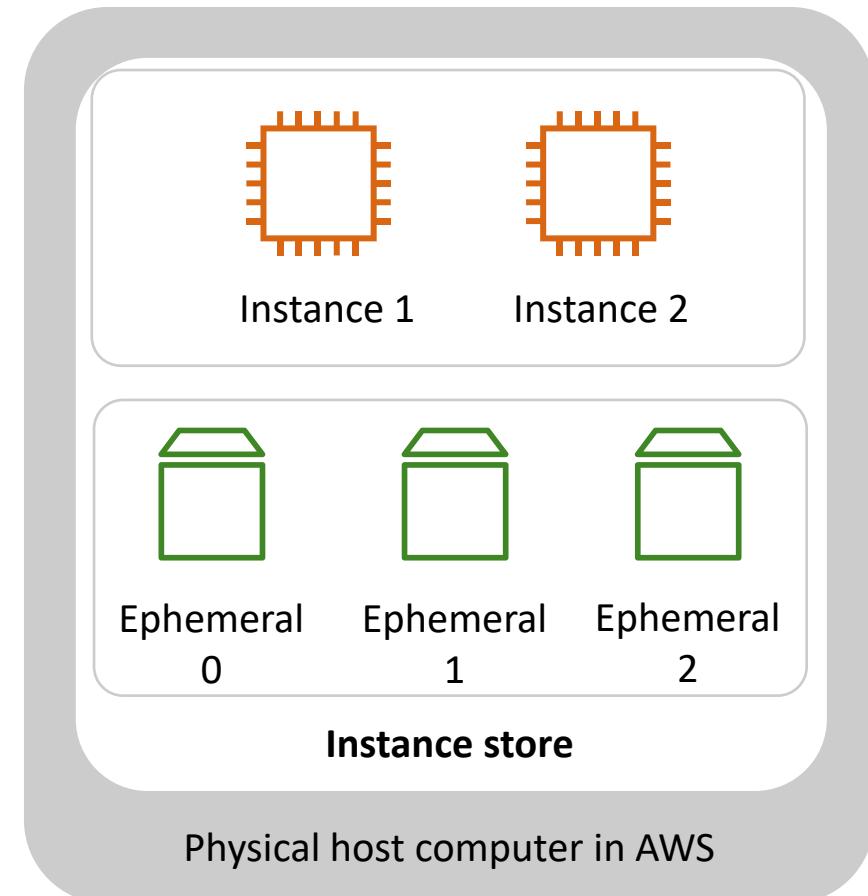
Amazon Elastic File System  
(Amazon EFS) [Linux]



Amazon FSx for Windows  
File Server

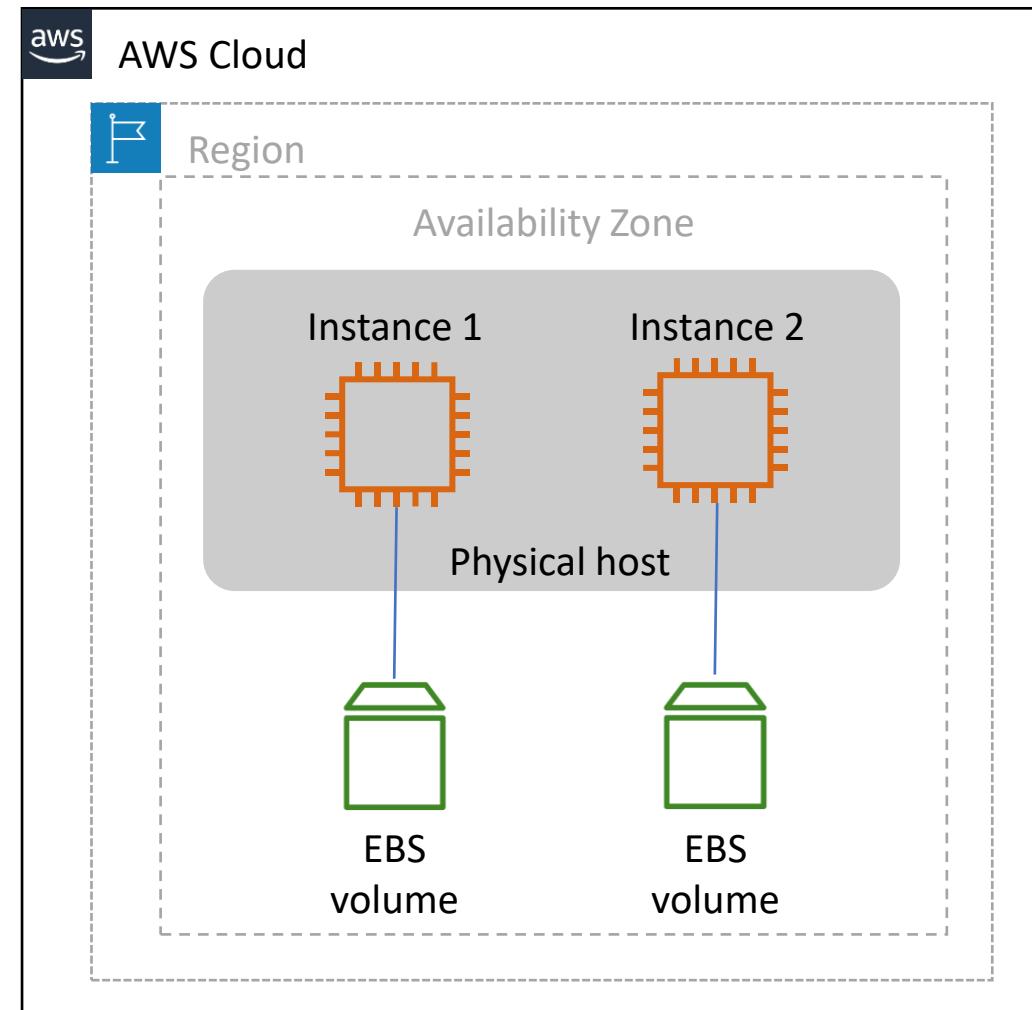
# Instance store

- An instance store provides **non-persistent storage** to an instance –
  - The data is stored on the *same physical server* where the instance runs
- Characteristics –
  - Temporary block-level storage
  - Uses HDD or SSD
  - **Instance store data is lost when the instance is *stopped* or *terminated***
- Example use cases –
  - Buffers
  - Cache
  - Scratch data



# Amazon EBS

- Amazon EBS volumes provide **network-attached persistent storage** to an EC2 instance.
- Characteristics –
  - Is persistent block-level storage
  - Can attach to any instance in the same Availability Zone
  - Uses HDD or SSD
  - Can be encrypted
  - Supports snapshots that are persisted to S3
  - Data persists independently from the life of the instance
- Example use cases –
  - Stand-alone database
  - General application data storage



# Amazon EBS SSD-backed volume types

Amazon EBS SSD-backed volumes are suited for use cases where the performance focus is on IOPS.

	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)
Description	Balances price and performance for a wide variety of workloads	<ul style="list-style-type: none"><li>Highest-performance SSD volume</li><li>Good for mission-critical, low-latency, or high-throughput workloads</li></ul>
Use Cases	<ul style="list-style-type: none"><li>Recommended for most workloads</li><li>Can be a boot volume</li></ul>	<ul style="list-style-type: none"><li>Critical business applications that require sustained IOPS performance</li><li>Large database workloads</li><li>Transactional workloads</li><li>It can be a boot volume</li></ul>

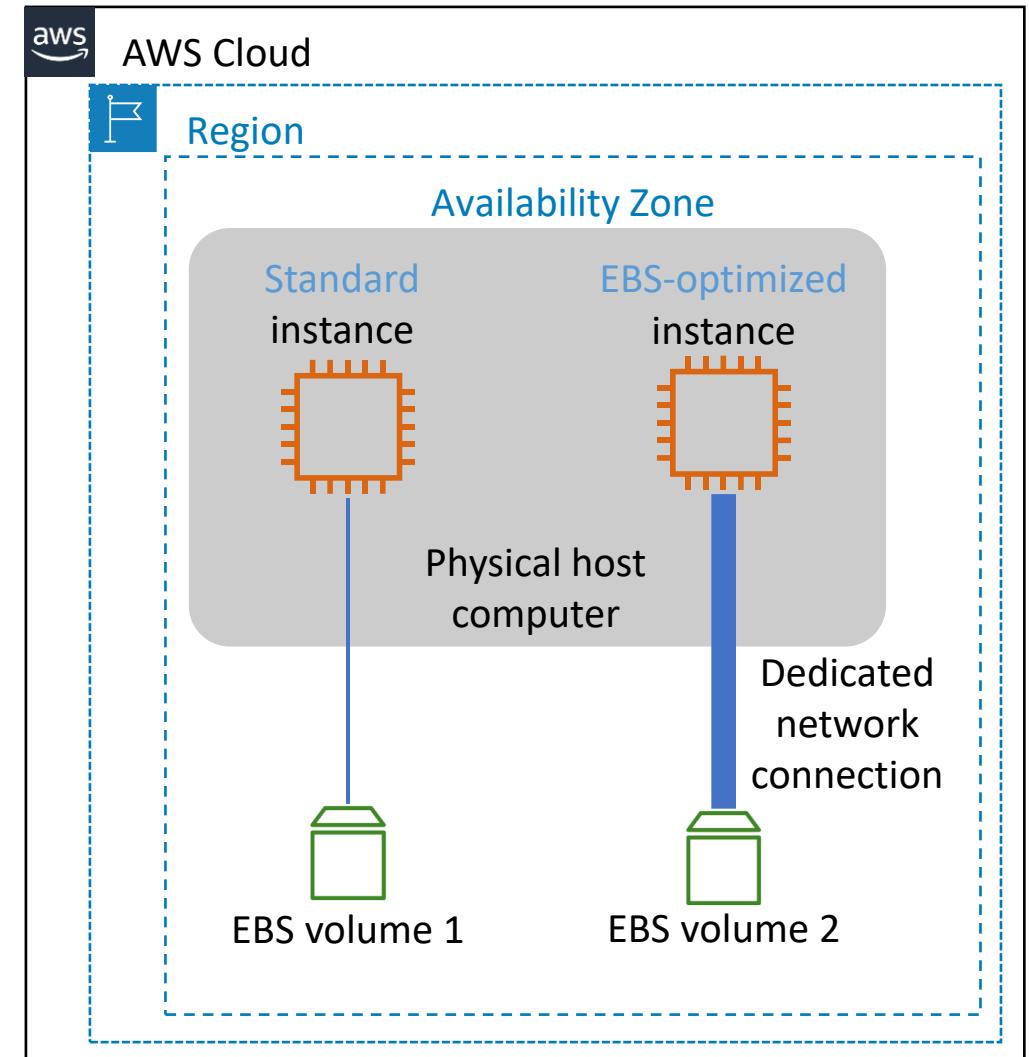
# Amazon EBS HDD-backed volume types

Amazon EBS HDD-backed volumes work well when the focus is on throughput.

	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	<ul style="list-style-type: none"><li>• Low-cost volume type</li><li>• Designed for frequently accessed, throughput-intensive workloads</li></ul>	<ul style="list-style-type: none"><li>• Lowest-cost HDD volume</li><li>• Designed for less frequently accessed workloads</li></ul>
Use Cases	<ul style="list-style-type: none"><li>• Streaming workloads</li><li>• Big data</li><li>• Data warehouses</li><li>• Log processing</li><li>• It cannot be a boot volume</li></ul>	<ul style="list-style-type: none"><li>• Throughput-oriented storage for large volumes of infrequently accessed data</li><li>• Use cases where the lowest storage cost is important</li><li>• It cannot be a boot volume</li></ul>

# Amazon EBS-optimized instances

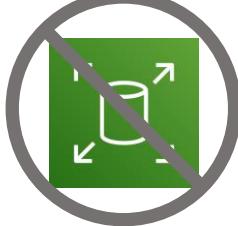
- Certain EC2 instance types can be [EBS-optimized](#)
- Benefits –
  - Provides a [dedicated network connection](#) to attached EBS volumes
  - Increases I/O performance
  - Additional performance is achieved if using an Amazon EC2 [Nitro System-based instance type](#)
- Usage –
  - For EBS-optimized instance types, optimization is enabled by default
  - For other instance types that support it, optimization must be manually enabled



# Shared file systems for EC2 instances

What if you have **multiple instances** that must use the **same storage**?

Amazon EBS: Attaches only to one instance



Amazon EBS

Amazon S3: Is an option, but is not ideal



Amazon S3

Amazon EFS and  
Amazon FSx for Windows  
File Server: Both satisfy  
the requirement



Amazon EFS  
(Linux)



Amazon FSx for  
Windows File  
Server (Windows)

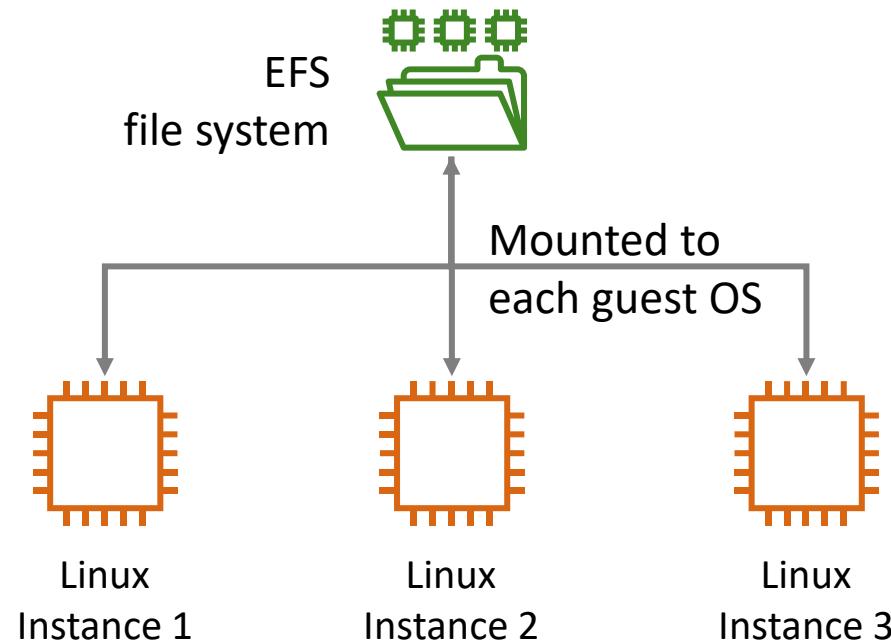


Amazon  
Elastic File System  
(Amazon EFS)

Amazon EFS provides file system storage for  
**Linux-based** workloads.

- Fully managed elastic file system
- Scales automatically up or down as files are added and removed
- Petabytes of capacity
- Supports Network File System (NFS) protocols
  - Mount the file system to the EC2 instance
- Compatible with all Linux-based AMIs for Amazon EC2

# Amazon EFS use cases



## Common workloads and applications:

- Home directories
- File system for enterprise applications
- Application testing and development
- Database backups
- Web serving and content management
- Media workflows
- Big data analytics

Example command to mount the file system to each guest OS:

```
$ sudo mount -t nfs4 mount-target-DNS:/ ~/efs-mount-point
```



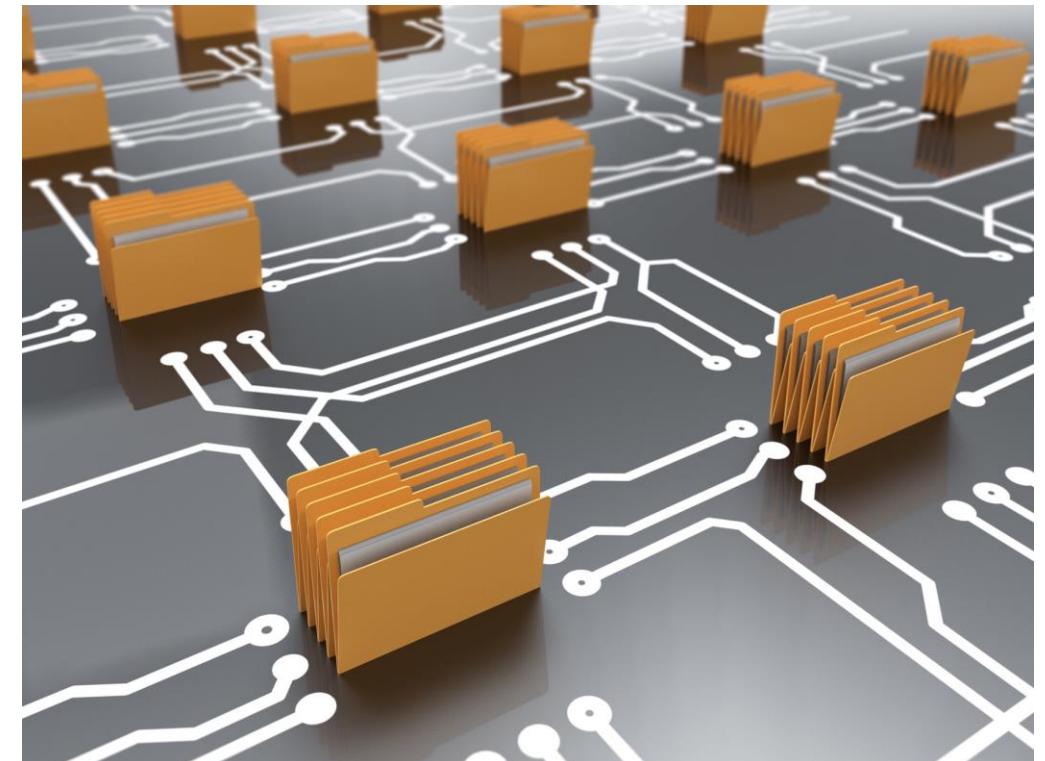
Amazon FSx for  
Windows File  
Server

Provides fully managed shared file system storage for Microsoft Windows EC2 instances.

- Native Microsoft Windows compatibility
- New Technology File System (NTFS)
- Native Server Message Block (SMB) protocol version 2.0 to 3.1.1
- Distributed File System (DFS) Namespaces and DFS Replication
- Integrates with Microsoft Active Directory and supports Windows access control lists (ACLs)
- Backed by high-performance SSD storage

Amazon FSx for Windows File Server supports a broad set of Microsoft Windows workloads.

- Home directories
- Lift-and-shift application workloads
- Media and entertainment workflows
- Data analytics
- Web serving and content management
- Software development environments



## key takeaways

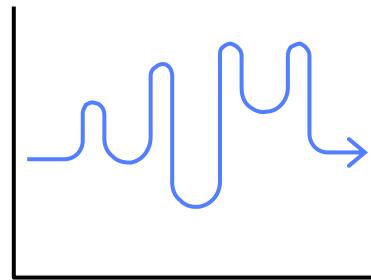
- Storage options for EC2 instances include instance store, Amazon EBS, Amazon EFS, and Amazon FSx for Windows File Server
- For a root volume, use instance store or SSD-backed Amazon EBS
- For a data volume that serves only one instance, use instance store or Amazon EBS storage
- For a data volume that serves multiple Linux instances, use Amazon EFS
- For a data volume that serves multiple Microsoft Windows instances, use Amazon FSx for Windows File Server

# Amazon EC2 pricing options

# Amazon EC2 pricing options (1 of 2)

## On-Demand Instances

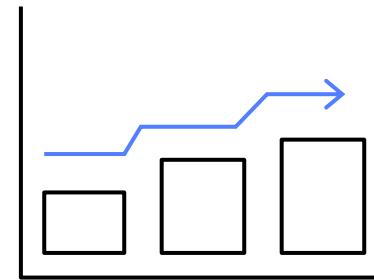
Pay for compute capacity by [the second or by the hour](#) with [no long-term commitments](#).



Spiky workloads,  
workload experimentation

## Reserved Instances

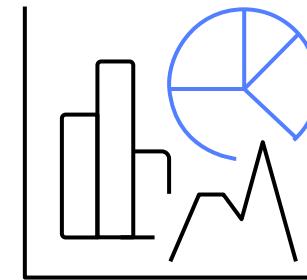
Make a [1-year or 3-year commitment](#) and receive a [significant discount](#) off on-demand prices.



Committed and  
steady-state workloads

## Savings Plans

Same discounts as Reserved Instances with [more flexibility](#).



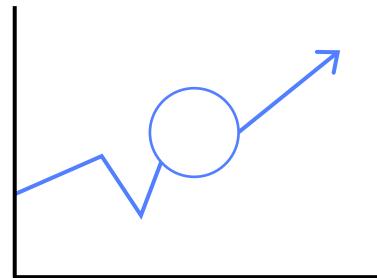
All Amazon EC2,  
AWS Fargate, and  
AWS Lambda workloads

# Amazon EC2 pricing options (2 of 2)

## Spot Instances

Spare Amazon EC2 capacity at **substantial savings** off  
On-Demand Instance prices.

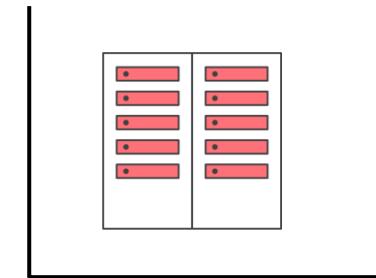
bid on unused EC2 instances



Fault-tolerant, flexible, stateless workloads

## Dedicated Hosts

**Physical server** with Amazon EC2 instance capacity  
fully dedicated for your use.



Workloads that require the use of your own  
software licenses or single tenancy to meet  
compliance requirements

# Amazon EC2 dedicated options

Amazon EC2 dedicated options provide EC2 instance capacity on physical servers that are dedicated for your use (single-tenant hardware).

## Dedicated Instances

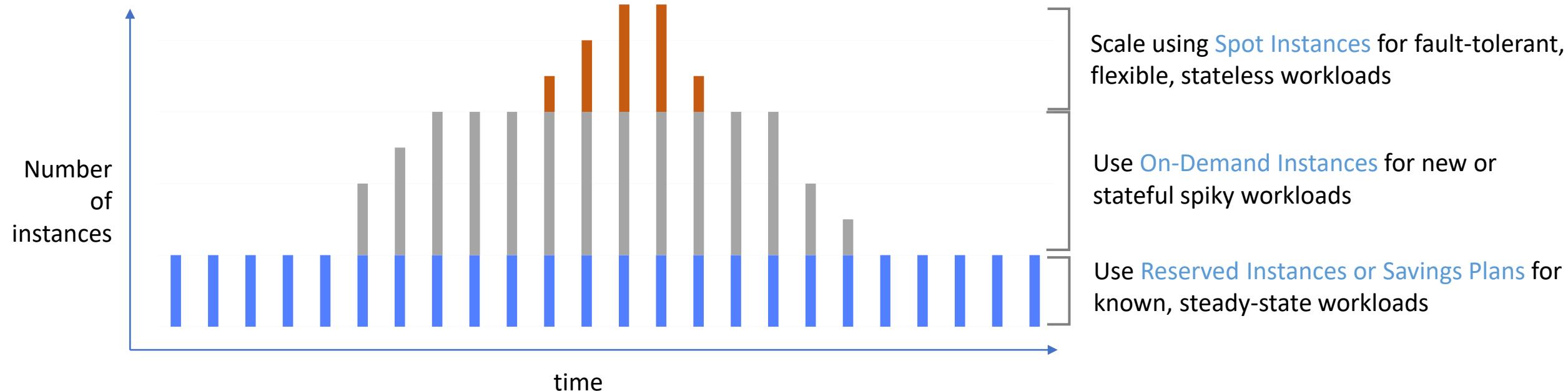
- Per-instance billing
- Automatic instance placement
- Benefit – Isolates the hosts that run your instances

## Dedicated Hosts

- Per-host billing
- Visibility of sockets, cores, and host ID
- Affinity between a host and an instance
- Targeted instance placement
- Add capacity by using an allocation request
- Benefit – Enables you to use your server-bound software licenses and address compliance requirements

# Amazon EC2 cost optimization guideline

To [optimize](#) the cost of Amazon EC2 instances, [combine](#) the available purchase options.



# Amazon EC2 considerations

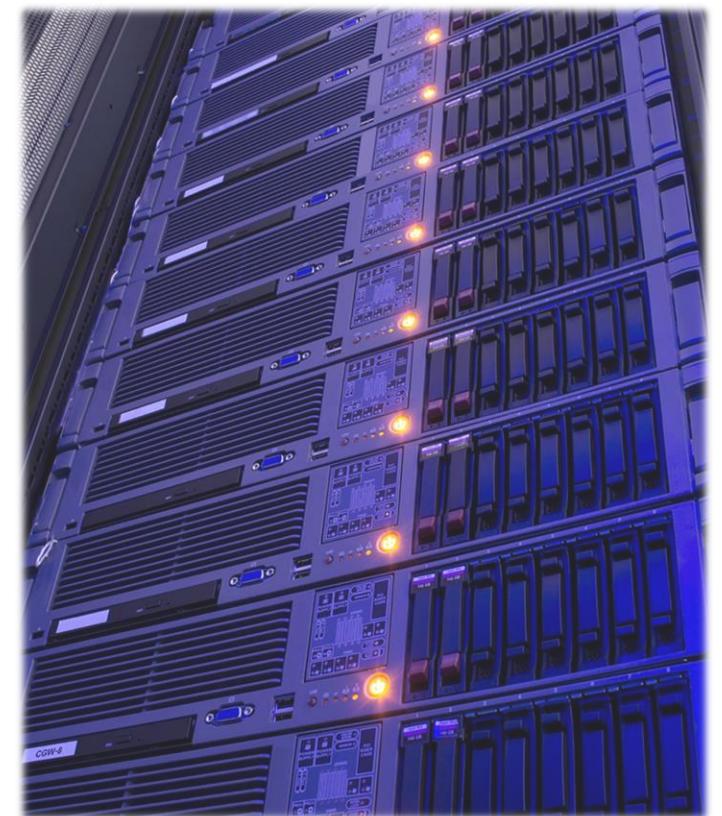
# Network Features

- EC2-Classic
  - Original EC2 release
  - Instances run in a single, flat network that is shared with other customers
  - **Does not apply with new accounts**
- Amazon VPC
  - New instances are launched in Amazon VPC (**Default or Customized**)
  - An instance must be in a VPC
  - Allows customers with control of the IP address space, the ability to segment with subnets, the ability to provide network-level security, etc.

# Placement groups

Placement groups enable you to [control where instances run](#) in an Availability Zone.

- They influence where a group of [interdependent instances](#) run –
  - Increase network performance between them
  - Reduce correlated or simultaneous failure
- Placement strategies –
  - Cluster - Packs instances close together inside an Availability Zone to achieve low-latency network performance.
  - Partition - Spreads instances across logical partitions so that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions.
  - Spread- Strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.
- Limitations –
  - An instance can be launched in only one placement group at a time
  - Instances with a tenancy of *host* cannot be launched in a placement group



# Placement Group

The screenshot shows the AWS Management Console interface for managing Placement Groups. The left sidebar navigation includes links for Scheduled Instances, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Services (Search bar, Actions dropdown, Create placement group button). The main content area is titled "Placement groups" and displays a table with columns: Group name, Group Id, Strategy, State, and Partition. A message at the bottom states, "You do not have any placement groups in this region."

Scheduled Instances  
Capacity Reservations  
Images  
AMIs New  
AMI Catalog  
Elastic Block Store  
Volumes New  
Snapshots New  
Lifecycle Manager New  
Network & Security  
Security Groups  
Elastic IPs  
Placement Groups  
Key Pairs  
Network Interfaces

Search for services, features, blogs, docs, and more [Alt+S]

Placement groups

Filter placement groups

Actions ▾ Create placement group

Group name	Group Id	Strategy	State	Partition
You do not have any placement groups in this region.				

The screenshot shows the AWS Management Console interface for creating a placement group. At the top, there's a navigation bar with the AWS logo, a search bar containing "Search for services, features, blogs, docs, and more" and a keyboard shortcut "[Alt+S]", and account information for "N. Virginia" and "UTASantosoAWS". Below the navigation bar, the breadcrumb trail indicates the user is in the EC2 service under Placement groups, specifically on the "Create placement group" page. The main content area has a title "Create placement group" and a section titled "Placement group settings". This section includes fields for "Name" (an empty input field), "Placement strategy" (a dropdown menu currently set to "Choose strategy" with options "Cluster", "Spread", and "Partition"), and a "Add tag" button. A note below the tag button states "You can add up to 50 more tags." At the bottom of the form are two buttons: "Cancel" and a prominent orange "Create group" button.

## Create placement group

### Placement group settings

Name

Placement strategy

Determines how the instances are placed on the underlying hardware.

Choose strategy ▾

Cluster

Spread

Partition

Add tag

You can add up to 50 more tags.

Cancel Create group

When you spin-up an EC2, you can go to “Advance Detail” and select the Placement Group you created.

# Security Groups (Firewall)

# AWS Security Group Basic

- **Security group** – acts as a virtual **firewall (inbound and outbound)** that controls the traffic for one or more instances.
  - act at the **instance level, not the subnet level**.
  - **each instance** in a subnet could be assigned to a **different set of security groups**.
  - is **region specific** and **VPC specific**.
  - If **SG not specified at EC2 launch time**, the EC2 is assigned to the **default security group for the VPC**.
  - For **each security group**, add rules that control the inbound and outbound traffic.
  - An instance can be associated with up-to **5 security groups**.
    - if there is conflict, **most permissible will prevail**
  - SGs are **stateful** — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. For VPC SG, responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
  - There are quotas on the number of security groups per VPC, the number of rules per security group, and the number of security groups associated with a network interface. See [Amazon VPC quotas](#).

# AWS Security Group Basic- Rules

**Security group rules** filter inbound and outbound traffic based on protocols and port numbers.

Can add rules to each security group.

The rules of a security group control

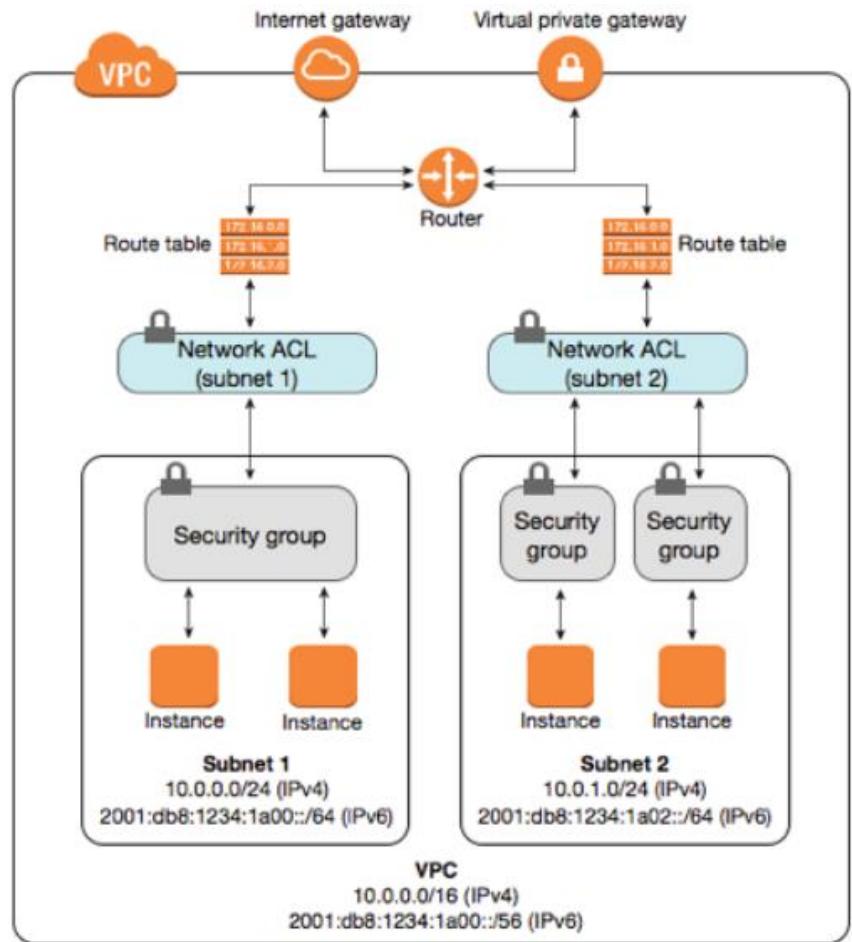
- the inbound traffic that is allowed to reach the instances that are associated with the security group
- The outbound traffic that's allowed to leave them

[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html)

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/security-group-rules.html>

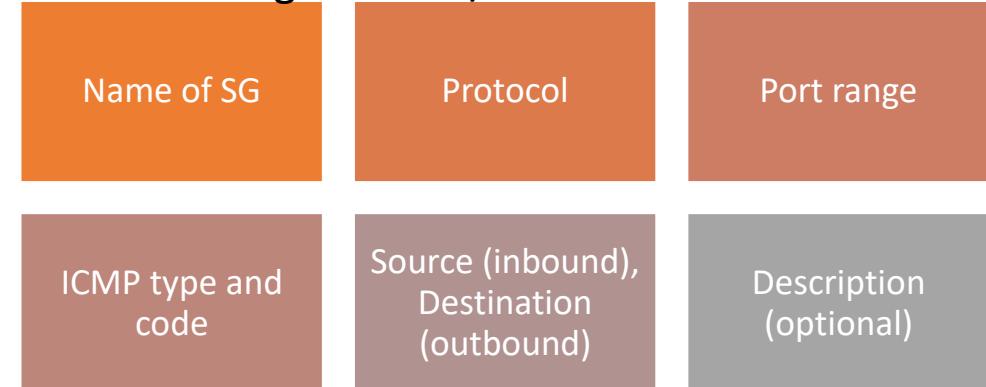
- Only **allow rules**, but **not deny rules**.
- Rules for a security group can be **modified at any time**.
- **You can't change the outbound rules for an EC2-Classic security group (old instances) – We don't use Old instances**
- Instances associated with a security group can't talk to each other unless add rules allowing the traffic (exception: the default security group has these rules by default).

# Security Group



- In+ Out rules in the same SG.
- EC2 can have with multiple SGs.
- **Rules only allow, no deny rule.**
- All rules are added.
- **if conflict, most permissible.**
- **SGs can be changed anytime.**

When adding inbound/outbound rule:



ICMP types and codes. You can select “All ICMP – ipv4”  
<https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>

# New Created Security Group Default rules

When a new security group is created, by default:

- it has no inbound rules. **No inbound traffic is allowed until inbound rules are added.**
  - Including internal traffics
- **There is an outbound rule that allows all outbound traffic.** This rule can be removed and new ones added.
  - Note if no outbound rule, no outbound traffic is allowed.

Remember a rule can **only allow but can not deny.**

New rules (inbound and outbound) can be added anytime.

New Security Group default	
No inbound rule	Allow all outbound traffic rule

# Default Security Group for VPC

VPC automatically comes with a default security group.

If a security group is not specified when an instance is launched, it is associated with the default security group (may happen when launching using API or CLI).

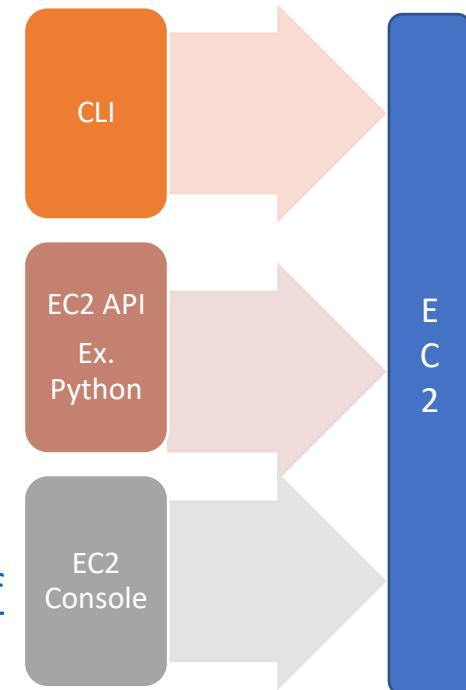
- If an instance is **launched in EC2 console**, the launch instance wizard automatically defines a "launch-wizard-xx" security group, which is associated with the instance.
- However, if an instance is launched using the **EC2 API or CLI** and a security group is not defined, the instance will be associated with the VPC default security group.

The rules for the default security group can be changed.

Default security group can not be deleted.

Note: We use EC2 console in this class to spin up EC2, we will always have at least 1 security group assigned to EC2s.

[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html#DefaultSecurityGroup](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html#DefaultSecurityGroup)



# Default Security Group

Note:  
0.0.0.0/0 – all IP address  
x.x.x.x/32 – specific IP address

## Default Security Group for Your VPC

Your VPC automatically comes with a default security group. If you don't specify a different security group when you launch the instance, we associate the default security group with your instance.



If you launch an instance in the Amazon EC2 console, the launch instance wizard automatically defines a "launch-wizard-xx" security group, which you can associate with the instance instead of the default security group.

The following table describes the default rules for a default security group.

Inbound			
Source	Protocol	Port Range	Description
The security group ID (sg-xxxxxxxx)	All	All	Allow inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group.
Outbound			
Destination	Protocol	Port Range	Description
0.0.0.0/0	All	All	Allow all outbound IPv4 traffic.
::/0	All	All	Allow all outbound IPv6 traffic. This rule is added by default if you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC.

You can change the rules for the default security group.

You can't delete a default security group. If you try to delete the default security group, you get the following error: Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user.



If you've modified the outbound rules for your security group, we do not automatically add an outbound rule for IPv6 traffic when you associate an IPv6 block with your VPC.

# Default VPC Security Group

The screenshot shows the AWS VPC Security Groups page. The left sidebar includes links for New VPC Experience, VPC Dashboard, Filter by VPC (with a dropdown for Select a VPC), and sections for VIRTUAL PRIVATE CLOUD (Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists). The main content area has a search bar at the top right and a table titled "Security Groups (1/11) Info". The table has columns: Name, Security group ID, Security group name, VPC ID, Description, Owner, Inbound rules count, and Outbound rules count. A checkbox column is also present. The table lists 11 security groups, with the last one, "sg-dbdf8cff" (Name: default), selected, indicated by a checked checkbox.

	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
<input type="checkbox"/>	-	sg-005036ae670ccb71c	lecture9kubernetes	vpc-472bcd3a	launch-wizard-6 create...	183451715204	4 Permission entries	1 Permission entry
<input type="checkbox"/>	-	sg-0168f61317caabd77	launch-wizard-3	vpc-472bcd3a	launch-wizard-3 create...	183451715204	1 Permission entry	1 Permission entry
<input type="checkbox"/>	-	sg-0310685be33623e6e	ContainerDemo	vpc-472bcd3a	launch-wizard-6 create...	183451715204	1 Permission entry	1 Permission entry
<input type="checkbox"/>	-	sg-06a4194b4af37184	lecture9kubernetesB	vpc-472bcd3a	launch-wizard-6 create...	183451715204	0 Permission entries	1 Permission entry
<input type="checkbox"/>	-	sg-07880d19912889f58	lecture9demo	vpc-472bcd3a	launch-wizard-6 create...	183451715204	4 Permission entries	1 Permission entry
<input type="checkbox"/>	-	sg-08b4f374e230f18b0	demolecture8	vpc-472bcd3a	launch-wizard-5 create...	183451715204	4 Permission entries	1 Permission entry
<input type="checkbox"/>	-	sg-0a2455bd1a12ac471	launch-wizard-4	vpc-472bcd3a	launch-wizard-4 create...	183451715204	2 Permission entries	1 Permission entry
<input type="checkbox"/>	-	sg-0b7f9b46c7ed877ba	launch-wizard-1	vpc-472bcd3a	launch-wizard-1 create...	183451715204	1 Permission entry	1 Permission entry
<input type="checkbox"/>	-	sg-0e5acf18234b6e57	launch-wizard-5	vpc-472bcd3a	launch-wizard-5 create...	183451715204	1 Permission entry	1 Permission entry
<input type="checkbox"/>	-	sg-0e5f7236552827def	launch-wizard-2	vpc-472bcd3a	launch-wizard-2 create...	183451715204	1 Permission entry	1 Permission entry
<input checked="" type="checkbox"/>	-	sg-dbdf8cff	default	vpc-472bcd3a	default VPC security gr...	183451715204	1 Permission entry	1 Permission entry

# Default VPC Security Group - Inbound

The screenshot shows the AWS VPC Security Groups console. The top navigation bar includes the AWS logo, services dropdown, search bar ('Search for services, features, marketplace products, and docs [Alt+S]'), and account information. The left sidebar lists VPC-related services: New VPC Experience, VPC Dashboard, Filter by VPC (with a 'Select a VPC' button), VIRTUAL PRIVATE CLOUD (Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, and NAT Gateways). The main content area shows the details for the security group 'sg-dbdf8cff - default'. The 'Inbound rules' tab is selected, displaying one rule: 'sgr-08e10cc2312b119ac' allowing all traffic from the source 'sg-dbdf8cff'. The 'Actions' dropdown menu is visible in the top right.

**Details**

Security group name default	Security group ID sg-dbdf8cff	Description default VPC security group	VPC ID vpc-472bcd3a
Owner 183451715204	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

**Inbound rules (1/1)**

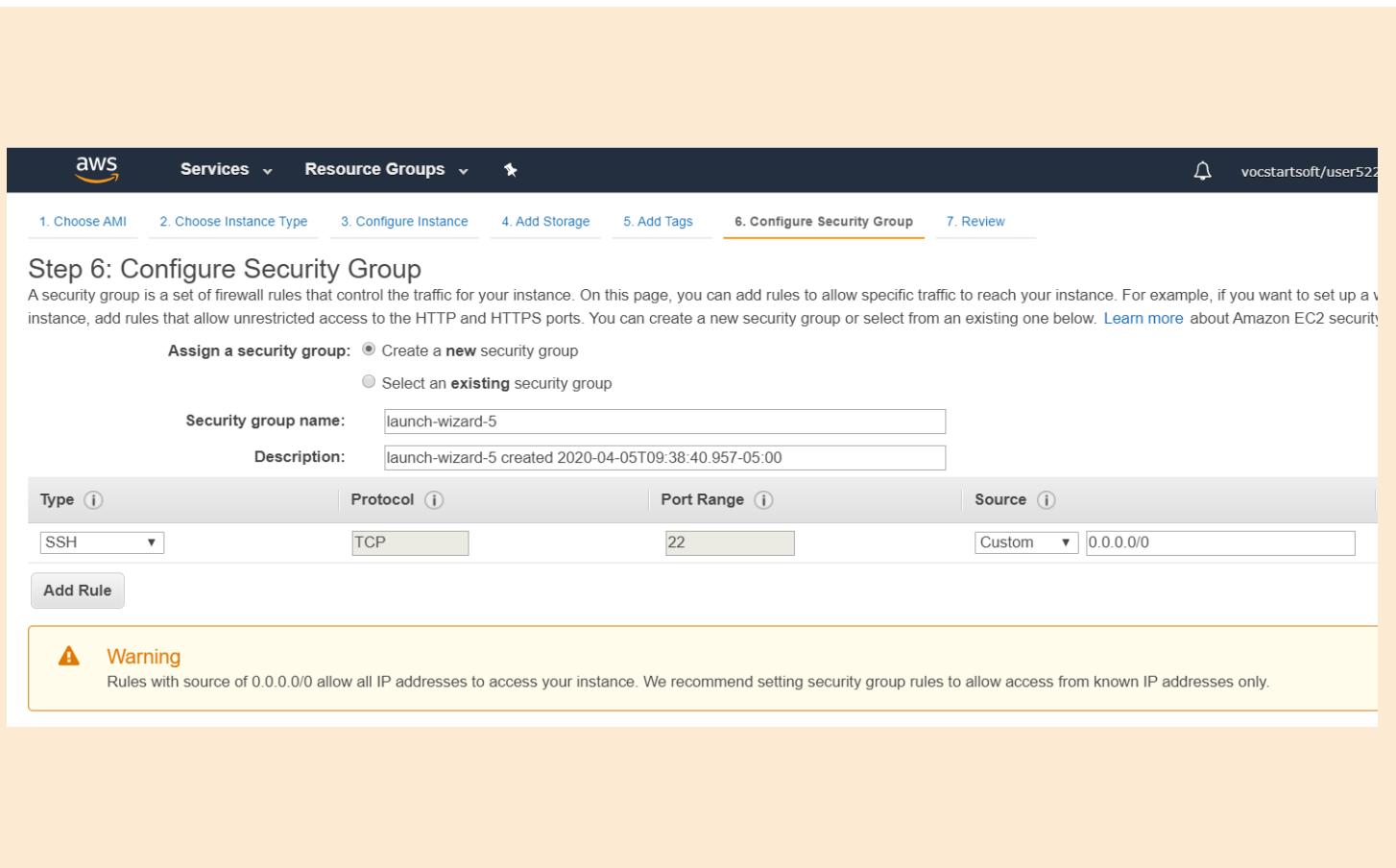
Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-08e10cc2312b119ac	-	All traffic	All	All	sg-dbdf8cff	-

# Default VPC Security Group - outbound

The screenshot shows the AWS VPC Security Groups console. The left sidebar is titled "VIRTUAL PRIVATE CLOUD" and includes links for Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, and NAT Gateways. The main content area shows the details for the security group "sg-dbdf8cff - default". The "Outbound rules" tab is selected, displaying one rule:

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination	Description
sgr-01f0c520bdacc8a03	IPv4	All traffic	All	All	0.0.0.0/0	-	

# Security Group – Instance Creation



The screenshot shows the AWS EC2 instance creation wizard at Step 6: Configure Security Group. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and a user icon 'vocstartsoft/user522'. Below the navigation, a progress bar shows steps 1 through 7, with '6. Configure Security Group' highlighted. The main content area is titled 'Step 6: Configure Security Group' and contains the following information:

- Assign a security group:**  Create a **new** security group  
 Select an **existing** security group
- Security group name:** launch-wizard-5
- Description:** launch-wizard-5 created 2020-04-05T09:38:40.957-05:00

Below these fields is a table with four columns: Type, Protocol, Port Range, and Source. The first row shows 'SSH' selected in the Type dropdown, 'TCP' in the Protocol dropdown, '22' in the Port Range input, and 'Custom' with '0.0.0.0/0' in the Source input.

**Add Rule** button

**Warning**: 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.

- If you launch an instance using the **Amazon EC2 API** or a **command line tool** and you **don't specify** a security group, the instance is automatically assigned to the **default security group for the VPC**.
- If you launch an instance using the **Amazon EC2 console**, you have an option to create a new security group for the instance. (see example below).

# Source/Destination for a Rule

## Source or Destination Options

An individual IPv4 address must use the /32 prefix length; for example, 203.0.113.1/32

An individual IPv6 address must use the /128 prefix length; for example, 2001:db8:1234:1a00::123/128.

A range of IPv4 addresses in CIDR block notation; for example, 203.0.113.0/24.

A range of IPv6 addresses in CIDR block notation; for example, 2001:db8:1234:1a00::/64.

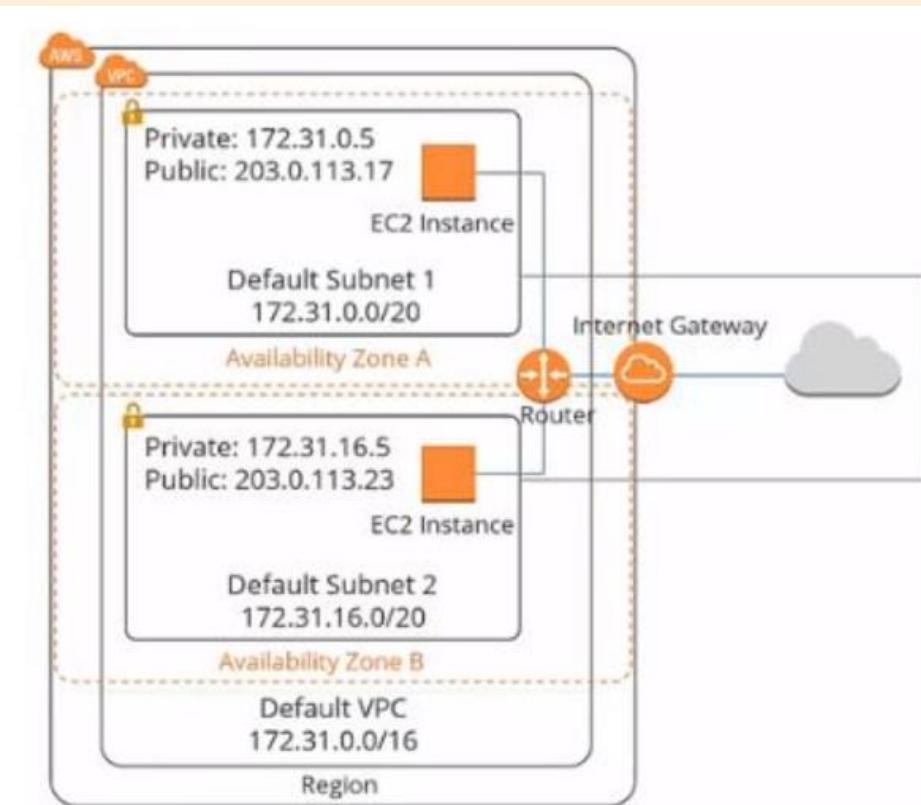
A prefix list ID The prefix list is a collection of one or more CIDR blocks. For example, pl-1234abc1234abc123. For more information, see [Prefix lists](#) in the Amazon VPC User Guide.

Another security group This allows instances that are associated with the specified security group to access instances associated with this security group. Choosing this option does not add rules from the source security group to this security group.

- The current security group
- A different security group for the same VPC
- A different security group for a peer VPC in a VPC peering connection

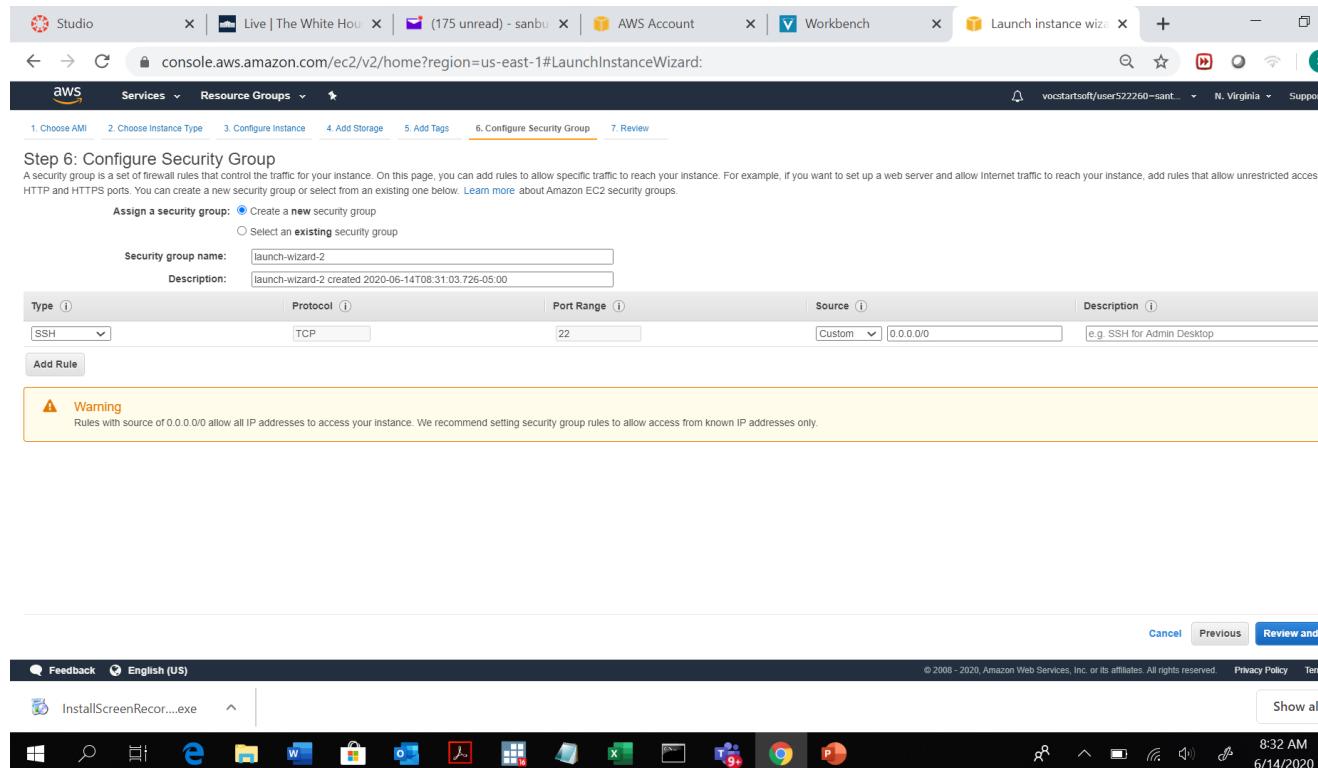
- Source or destination: The **source (inbound rules)** or **destination (outbound rules)** for the traffic.
- [https://docs.amazonaws.cn/en\\_us/AWSEC2/latest/WindowsGuide/security-group-rules.html](https://docs.amazonaws.cn/en_us/AWSEC2/latest/WindowsGuide/security-group-rules.html)

# EC2 in a VPC



- An EC2 instance must reside in a VPC.
- VPC can be default or customized (you create).
  - A **default VPC** is a logically isolated virtual network in the **AWS** cloud that is automatically created for your **AWS** account the first time you provision Amazon **EC2** resources.
- In this section we will use the default VPC.

# EC2 in Default VPC



- When creating EC2 (even in default VPC) using console, you have options to create new security group. Unless set to allowed, no traffic can get in including from other internal EC2

AWS Services Resource Groups

New EC2 Experience Learn more

Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts New Scheduled Instances Capacity Reservations

AMIs AMIs Bundle Tasks

ELASTIC BLOCK STORE Volumes Snapshots Lifecycle Manager

NETWORK & SECURITY Security Groups New Elastic IPs New Placement Groups New Key Pairs New Network Interfaces

LOAD BALANCING Load Balancers

EC2 > Security Groups > sg-08f63b0a3a8ee9e61 - launch-wizard-2

## sg-08f63b0a3a8ee9e61 - launch-wizard-2

Delete security group Copy to new security group

Details	
Security group name	Security group ID
launch-wizard-2	sg-08f63b0a3a8ee9e61
Owner	Description
608910113900	launch-wizard-2 created 2020-06-14T08:31:03.726-05:00
Inbound rules count	VPC ID
1 Permission entry	vpc-04230f7e
Outbound rules count	
1 Permission entry	

Inbound rules Outbound rules Tags

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-

<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#>

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

AWS Services Resource Groups N. Virginia Support

New EC2 Experience Learn more

Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts New Scheduled Instances Capacity Reservations

AMIs Bundle Tasks

Volumes Snapshots Lifecycle Manager

Security Groups New Elastic IPs New Placement Groups New Key Pairs New Network Interfaces

Load Balancers

EC2 > Security Groups > sg-08f63b0a3a8ee9e61 - launch-wizard-2

## sg-08f63b0a3a8ee9e61 - launch-wizard-2

Delete security group Copy to new security group

Details			
Security group name	Security group ID	Description	VPC ID
launch-wizard-2	sg-08f63b0a3a8ee9e61	launch-wizard-2 created 2020-06-14T08:31:03.726-05:00	vpc-04230f7e
Owner	Inbound rules count	Outbound rules count	
608910113900	1 Permission entry	1 Permission entry	

Inbound rules Outbound rules Tags

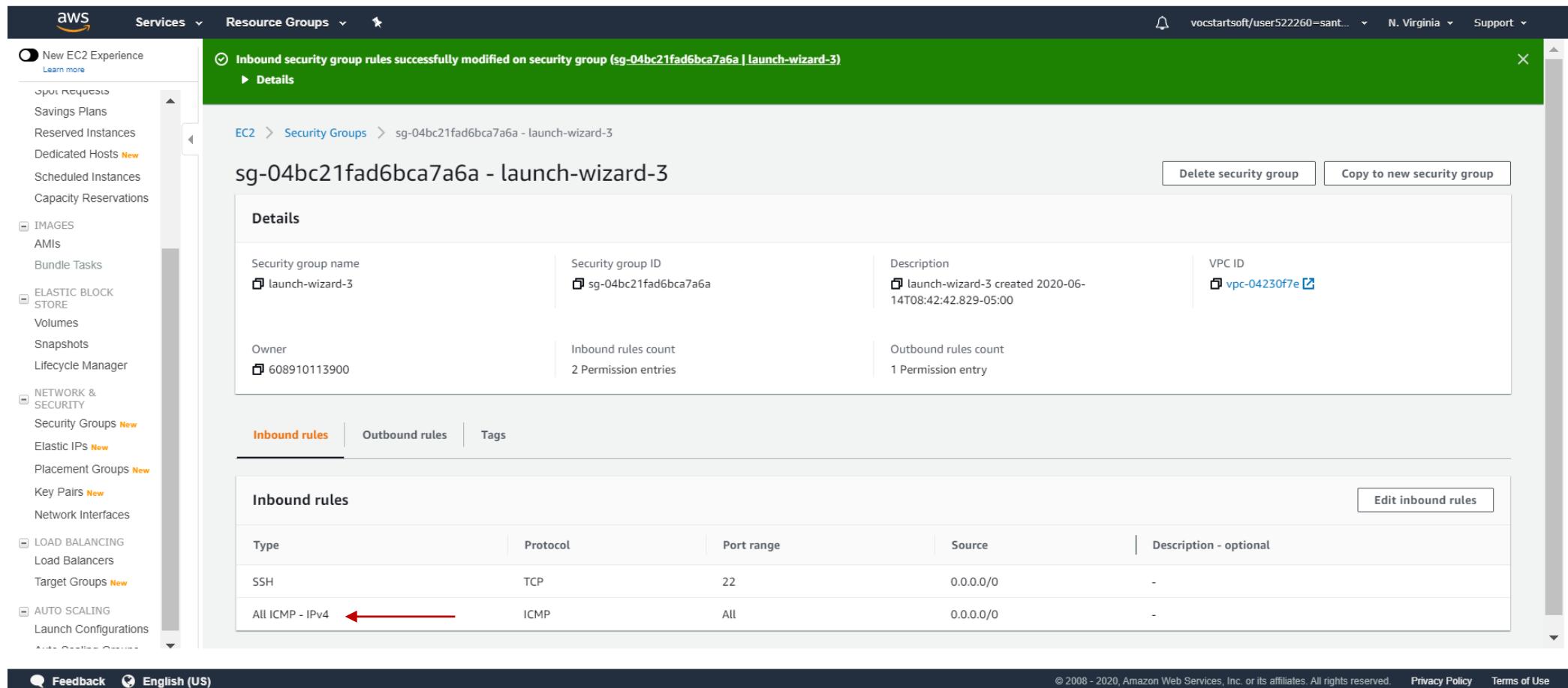
Edit outbound rules

Type	Protocol	Port range	Destination	Description - optional
All traffic	All	All	0.0.0.0/0	-

<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#>

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

# To allow to receive Ping command



The screenshot shows the AWS Management Console interface for managing security groups. The left sidebar lists various services like EC2, Lambda, and S3. The main content area is titled "sg-04bc21fad6bca7a6a - launch-wizard-3". A green banner at the top indicates that inbound security group rules were successfully modified. The "Inbound rules" tab is selected, showing two entries:

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-
All ICMP - IPv4	ICMP	All	0.0.0.0/0	-

A red arrow points to the "All ICMP - IPv4" rule.

At the bottom of the page, there are links for Feedback, English (US), and a footer with copyright information: © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use.

# Ping command to check connectivity

- **Ping** is a command used to test the reachability (connectivity) of a host on an Internet Protocol (IP) network. It is available for virtually all operating systems that have networking capability, including most embedded network administration software.
  - In linux you can use ping ipaddress or ping –c3 ipaddress (-c3 if you want to limit to 3 times).
  - Note: Command Prompt does not recognize –c3 (ping ipaddress)
- **Ping** uses the Internet Control Message Protocol (**ICMP**). The tool sends **ICMP** Echo Request packets to the destination host and waits for **ICMP** Echo Replies.
- ICMP is part of the Network layer and sits on top of the IP protocol.

A photograph of a rocket launching from a dark launch pad. The rocket is white with blue stripes and is angled upwards towards the top left of the frame. A bright, orange-yellow flame and a large plume of white smoke and steam rise from its base. The background is a dark, cloudy sky.

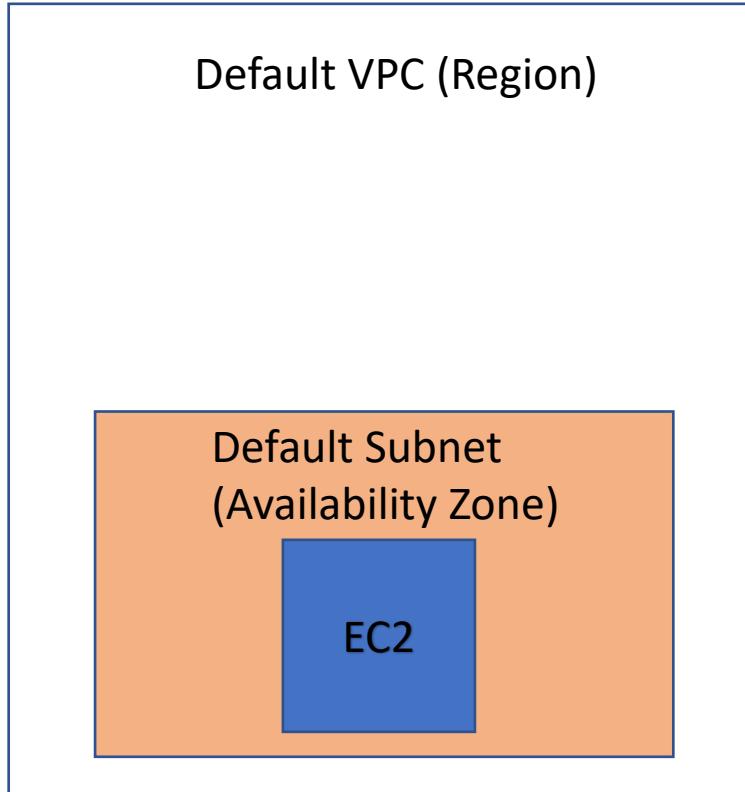
Launching  
EC2

# Launching EC2 Steps in Console



<https://aws.amazon.com/ec2/features/>

# EC2 in Default VPC



For each AWS account

- A **default VPC** per each **Region**
- A **default subnet** per each **Availability Zone**
- Unless defined, an EC2 will be in a default Subnet in a default VPC when created.
- <https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html#launching-into>

# EC2 Console

The screenshot shows the AWS EC2 console interface. On the left, there's a navigation sidebar with various services like EC2 Dashboard, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main area has sections for Resources (listing Running instances, Elastic IPs, Dedicated Hosts, etc.), Launch instance (with a prominent orange 'Launch instance' button), Scheduled events, and Migrate a machine. To the right, there's a 'Service health' section showing the status of the service as operating normally, and a 'Zone status' table for the US East (N. Virginia) region, which also shows all zones operating normally. The top right corner shows account attributes, including a 'Default VPC' entry set to 'vpc-472bcd3a'. A red arrow points to this entry. The bottom right corner has an 'Explore AWS' section with links to save costs, enable Graviton2, and migrate instances.

Welcome to the new EC2 console!

We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle.

**Resources**

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Running instances	0	Elastic IPs	0	Dedicated Hosts	0	Snapshots	0
Volumes	0	Load balancers	2	Key pairs	1	Security groups	8
Placement groups	0						

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#)

**Account attributes**

Supported platforms [\[?\]](#)

- VPC

Default VPC [\[?\]](#)  
vpc-472bcd3a

Settings

EBS encryption

Zones

Default credit specification

Console experiments

**Explore AWS**

Save Up to 25% on EC2

AWS Compute Optimizer identifies optimal AWS Compute resources to reduce costs and improve performance for your workloads. [Get started](#)

Enable Best Price-Performance with AWS Graviton2

AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. [Learn more](#)

Save 10% with AMD EPYC-Powered Instances

Lower cost on compute and memory with AMD EPYC processors. [Learn more](#)

**Additional information**

Getting started guide

Documentation

All EC2 resources

Forums

# Default VPC and its Subnets

The screenshot shows two views of the AWS VPC Dashboard. The top view displays 'Your VPCs (1/1)' with one entry: VPC ID vpc-472bcd3a, State Available, IPv4 CIDR 172.31.0.0/16, and other details like DHCP options set dopt-ebbe3d91 and Main route table rtb-27abc659. The bottom view shows 'Create subnet' with a table of subnets for VPC vpc-472bcd3a across six Availability Zones (us-east-1a to us-east-1f). Each subnet has an IPv4 CIDR of 172.31.x.x/20 and 4091 available IPv4 addresses. A red arrow points to the IPv4 CIDR of the last subnet.

Name	Subnet ID	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Network Border Group	Route table	Network ACL	Default subnet	Auto-assign pub. IP	Audited
subnet-460e1778	available	vpc-472bcd3a	172.31.48.0/20	4091	-	us-east-1e	use1-az3	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No
subnet-82d101a3	available	vpc-472bcd3a	172.31.80.0/20	4091	-	us-east-1b	use1-az2	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No
subnet-8c6cbbd3	available	vpc-472bcd3a	172.31.32.0/20	4091	-	us-east-1d	use1-az6	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No
subnet-8f2a4fc2	available	vpc-472bcd3a	172.31.16.0/20	4091	-	us-east-1c	use1-az4	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No
subnet-b50cc6d3	available	vpc-472bcd3a	172.31.0.0/20	4091	-	us-east-1a	use1-az1	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No
subnet-f7ca0f1	available	vpc-472bcd3a	172.31.64.0/20	4091	-	us-east-1f	use1-az5	us-east-1	rtb-27abc659	acl-2291d85f	Yes	Yes	No

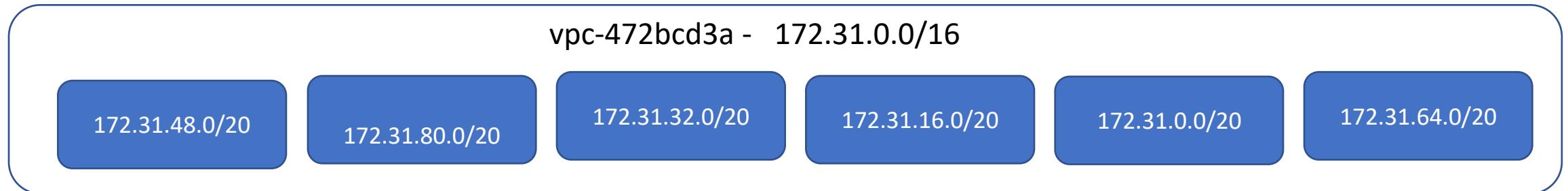
Notice the CIDR notations for the VPC and its subnets (all default):

- VPC – 172.31.0.0/16
- There are 6 subnets (1 per each AZ)
  - CIDR – 172.31.xx.xx/20
  - Available IPv4 for each subnet =  $(2^{12}) - 5 = 4091$  (number of possible hosts)

# Default VPC

- VPC with a size /16 IPv4 CIDR block (172.31.0.0/16). This provides up to 65,536 ( $2^{16}$ ) private IPv4 addresses.
- A size /20 default subnet in each Availability Zone. This provides up to 4,096 addresses per subnet, 5 are reserved for aws use, so 4,091.
- An internet gateway – when we create a customized VPC later, we need to connect this.
- Add a route to the main route table that points all traffic (0.0.0.0/0) to the internet gateway (internet access).
- A default security group and associate it with your default VPC.
- A default network access control list (ACL) and associate it with your default VPC.
- Associate the default DHCP options.
- DNS is enabled by default for default VPCs. <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html#vpc-dns-support>
  - DNS hostname is a name that uniquely and absolutely names a computer
- instance in a VPC has public and private DNS hostnames that correspond to the public IPv4 and private IPv4 addresses for the instance. No DNS hostnames for IPv6 addresses.
  - A public (external) DNS hostname takes the form ec2-public-ipv4-address.compute-1.amazonaws.com for the us-east-1 Region, and ec2-public-ipv4-address.region.compute.amazonaws.com for other Regions.
  - The private DNS hostname takes the form ip-private-ipv4-address.ec2.internal for the us-east-1 Region, and ip-private-ipv4-address.region.compute.internal for other Regions
  - <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-dns.html>
- <https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html>

# Default VPC – Region N. Virginia



	Bit value	Octet 4								Octet 3								Octet 2								Octet 1							
		128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
VPC	172.31.0.0/16	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Subnet	172.31.48.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0		
Subnet	172.31.80.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0		
Subnet	172.31.32.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0		
Subnet	172.31.16.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
Subnet	172.31.0.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
Subnet	172.31.64.0/20	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0		

- Defined by AWS
- 16 bits from the left to identify the network (/16)
- Bit 17-Bit20 from the left to identify the subnet within that network (between /16 of VPC and /20 of subnets)
  - You could have  $2^4 = 16$  subnets
  - Default- 1 subnet per AZ
- The rest 12 bits (32-12) to identify the hosts in each subnet
  - Total host per subnet =  $2^{12} - 5 = 4091$  (AWS reserves 5)

# What if we change region?

The screenshot shows the AWS EC2 console interface for the US West (Oregon) Region. The top navigation bar includes the AWS logo, services dropdown, resource groups, notifications, account information (UTASantosoAWS), and support links. A blue banner at the top reads: "Welcome to the new EC2 console! We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle." The left sidebar contains navigation links for New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances, Images, Elastic Block Store, Network & Security, and Key Pairs. The main content area is divided into several sections: "Resources" (Shows 0 Running instances, 0 Elastic IPs, 0 Dedicated Hosts, 0 Snapshots, 0 Volumes, 0 Load balancers, 0 Key pairs, and 1 Security group), "Launch instance" (Shows a "Launch Instance" button and a note about launching in the US West (Oregon) Region), "Scheduled events" (Shows "US West (Oregon)" and "No scheduled events"), "Service health" (Shows the Region as "US West (Oregon)" and the status as "This service is operating normally"), "Zone status" (Shows four availability zones: us-west-2a, us-west-2b, us-west-2c, and us-west-2d, all listed as "Zone is operating normally"), and "Explore AWS" (Promotions for AWS Graviton2, AMD EPYC-Powered Instances, and GPU Powered ML Inference with G4). A red arrow points from the top right towards the "Oregon" link in the top right corner of the header.

**Welcome to the new EC2 console!**  
We're redesigning the EC2 console to make it easier to use and improve performance. We'll release new screens periodically. We encourage you to try them and let us know where we can make improvements. To switch between the old console and the new console, use the New EC2 Experience toggle.

**Resources**

You are using the following Amazon EC2 resources in the US West (Oregon) Region:

Running instances	0	Elastic IPs	0	Dedicated Hosts	0	Snapshots	0
Volumes	0	Load balancers	0	Key pairs	0	Security groups	1
Placement groups	0						

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. [Learn more](#)

**Launch instance**

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

**Service health**

Region	Status
US West (Oregon)	This service is operating normally

**Zone status**

Zone	Status
us-west-2a (usw2-az1)	Zone is operating normally
us-west-2b (usw2-az2)	Zone is operating normally
us-west-2c (usw2-az3)	Zone is operating normally
us-west-2d (usw2-az4)	Zone is operating normally

**Explore AWS**

- Enable Best Price-Performance with AWS Graviton2
- Save 10% with AMD EPYC-Powered Instances
- GPU Powered ML Inference with G4

**Additional information**

# Default Subnets

The screenshot shows two views of the AWS VPC Dashboard.

**Top View:** Shows the main VPC dashboard with one VPC listed. The VPC details are as follows:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR (Network Border Group)	IPv6 pool	DHCP options set	Main route table	Main network ACL
-	vpc-fe85f586	Available	172.31.0.0/16	-	-	dopt-8884edf0	rtb-8f7e23f4	acl-9c8d46e7

**Bottom View:** Shows the Subnets page, listing four available subnets under the VPC vpc-fe85f586. The subnet details are as follows:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID	Network Border Group	Route table	Network ACL	Default subnet	Auto-assign publ	Au
subnet-2cfa3754	available	vpc-fe85f586	172.31.16.0/20	4091	-	-	us-west-2a	usw2-az1	us-west-2	rtb-8f7e23f4	acl-9c8d46e7	Yes	Yes	No
subnet-75bb553f	available	vpc-fe85f586	172.31.32.0/20	4091	-	-	us-west-2b	usw2-az2	us-west-2	rtb-8f7e23f4	acl-9c8d46e7	Yes	Yes	No
subnet-843166af	available	vpc-fe85f586	172.31.48.0/20	4091	-	-	us-west-2d	usw2-az4	us-west-2	rtb-8f7e23f4	acl-9c8d46e7	Yes	Yes	No
subnet-ff4b97a2	available	vpc-fe85f586	172.31.0.0/20	4091	-	-	us-west-2c	usw2-az3	us-west-2	rtb-8f7e23f4	acl-9c8d46e7	Yes	Yes	No

# Example

- Spin up an EC2
- Access the EC2 using SSH
- Install web-server software (nginx) into the EC2
- Update the webpage
- Call the URL

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

selection at the end -add
_modifier_ob.select= 1
Modifier_ob.select=1
bpy.context.scene.objects.active
("Selected" + str(modifier))
modifier.select = 0
bpy.context.selected_objects
data.objects[one.name].sele
print("please select exactly one
- OPERATOR CLASSES -->
types.Operator):
X mirror to the selected ob
ject.mirror_mirror_x"
or X"
```

```
context):
next.active_object is not
```

# Spin Up an EC2

The screenshot shows the AWS EC2 Dashboard. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, the URL 'vocstartsoft/user522260=sant...', a region selector for 'N. Virginia', and 'Support' dropdown. On the left, a sidebar menu lists 'New EC2 Experience' (with a 'Learn more' link), 'EC2 Dashboard' (marked as 'New'), 'Events', 'Tags', 'Reports', 'Limits', and a 'INSTANCES' section with 'Instances' (which is selected, indicated by an orange border) and other options like 'Instance Types', 'Launch Templates' (marked as 'New'), 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', and 'Capacity Reservations'. The main content area features a 'Launch Instance' button at the top, followed by a search bar with the placeholder 'Filter by tags and attributes or search by keyword'. Below the search bar, a message states 'You do not have any running instances in this region.' and 'First time using EC2? Check out the [Getting Started Guide](#). Click the Launch Instance button to start your own server.' A large blue arrow points from the bottom right towards the 'Launch Instance' button.

# Select an AMI

The screenshot shows the AWS Step 1: Choose an Amazon Machine Image (AMI) interface. At the top, there's a navigation bar with the AWS logo, Services dropdown, search bar ('Search for services, features, marketplace products, and docs'), and [Alt+S] key shortcut. Below the navigation is a breadcrumb trail: 1. Choose AMI (highlighted in orange), 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review.

**Step 1: Choose an Amazon Machine Image (AMI)**

**SUSE Linux Enterprise Server 15 SP2 (HVM), SSD Volume Type** - ami-0de50fcabcd46f2f (64-bit x86) / ami-05f2fbf6d89313bd (64-bit Arm)  
SUSE Linux Enterprise Server 15 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
**Select**  
 64-bit (x86)  
 64-bit (Arm)

**Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-09e67e426f25ce0d7 (64-bit x86) / ami-00d1ab6b335f217cf (64-bit Arm)  
Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
**Select**  
 64-bit (x86)  
 64-bit (Arm)

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-0747bdcabd34c712a (64-bit x86) / ami-08353a25e80beeaa3e (64-bit Arm)  
Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
**Select**  
 64-bit (x86)  
 64-bit (Arm)

**Microsoft Windows Server 2019 Base** - ami-03295ec1641924349  
Microsoft Windows 2019 Datacenter edition. [English]  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
**Select**  
64-bit (x86)

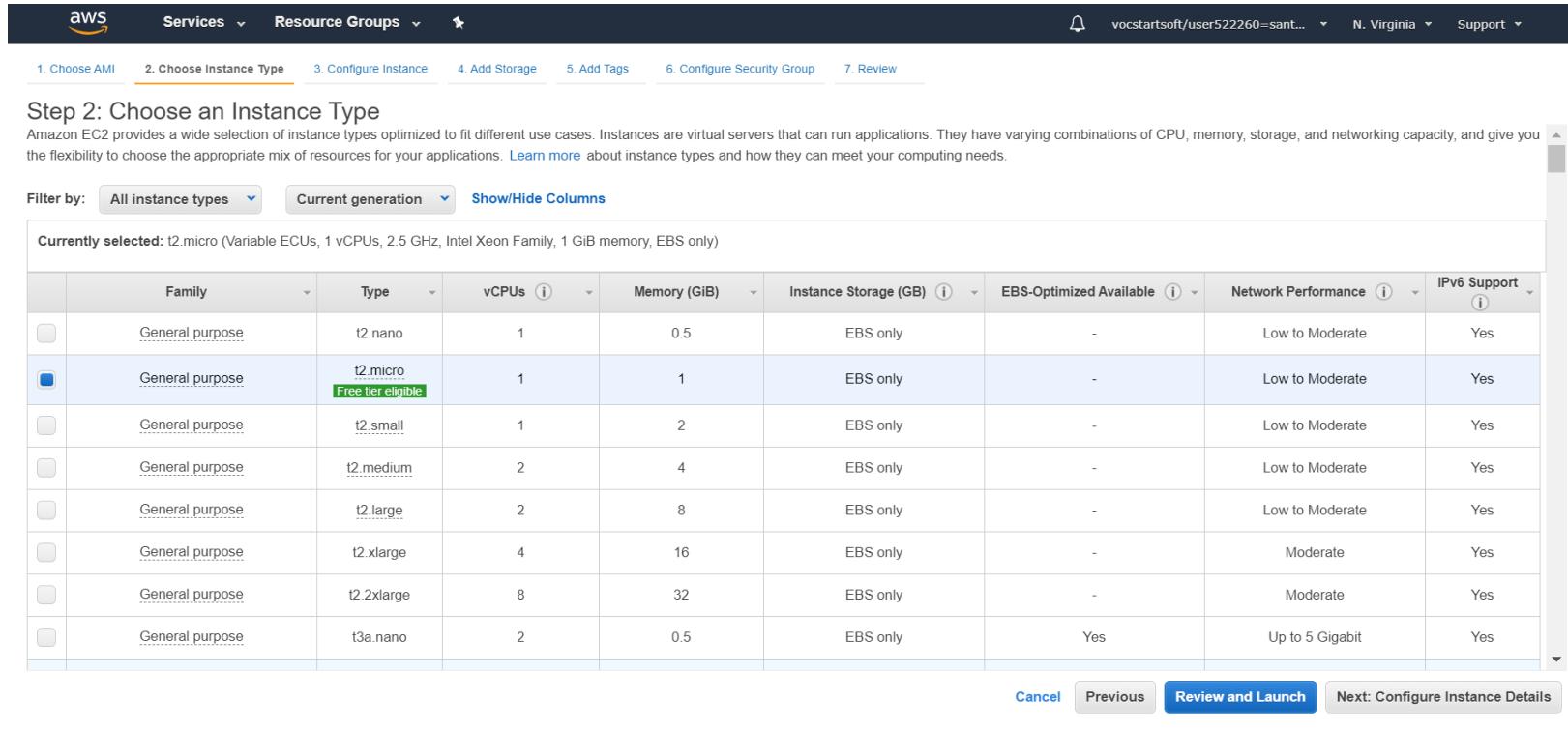
**Deep Learning AMI (Ubuntu 18.04) Version 48.0** - ami-0b70285e5215b80eb  
MXNet-1.8.0 & 1.7.0, TensorFlow-2.4.2, 2.3.3 & 1.15.5, PyTorch-1.4.0 & 1.8.1, Neuron, & others. NVIDIA CUDA, cuDNN, NCCL, Intel MKL-DNN, Docker, NVIDIA-Docker & EFA support. For fully managed experience, check: <https://aws.amazon.com/sagemaker>  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
**Select**  
64-bit (x86)

I choose **Ubuntu 20.04** (or the latest) so I can use certain commands to install nginx web server.

**Ubuntu** is a **linux distro** based on **Debian**, most software **is** usable on both distros.

Note Amazon Linux is not compatible with ubuntu.

# Choose Instance Type



The screenshot shows the AWS EC2 instance creation wizard at Step 2: Choose an Instance Type. The 't2.micro' instance is selected, highlighted with a blue border and labeled 'Free tier eligible'. The table lists various General purpose instance types with their details like vCPUs, Memory, and Network Performance.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	<b>t2.micro</b> Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Buttons at the bottom include: Cancel, Previous, Review and Launch (highlighted in blue), and Next: Configure Instance Details.

t2.micro (free tier) is enough for our purpose.

Note: it is possible to change instance type afterwards (after observing the application needs).

# EC2 in Default VPC (N.Virginia)

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1

Purchasing option: Request Spot instances

Network: vpc-04230f7e (default) | Create new VPC

Subnet: No preference (default subnet in any Availability Zone) | Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open | Create new Capacity Reservation

IAM role: None | Create new IAM role

Shutdown behavior: Stop

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring

Additional charges apply.

You can attach role here if EC2 needs to access other aws services

I spin up EC2 in default subnet and VPC.  
Allow public IP creation

Cancel Previous Review and Launch Next: Add Storage

# Add Storage

The screenshot shows the AWS EC2 instance creation wizard at Step 4: Add Storage. The top navigation bar includes 'Services', 'Resource Groups', and tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage' (which is highlighted), '5. Add Tags', '6. Configure Security Group', and '7. Review'. The main content area is titled 'Step 4: Add Storage' with the sub-instruction: 'Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.' A table lists the current storage configuration:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e078112eedeec9db	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

An 'Add New Volume' button is located below the table. A callout box provides information about free tier usage: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.'

At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Tags'.

Leave everything default, I don't need anything extra

# Add Tags (Optional)

The screenshot shows the AWS EC2 instance creation wizard at Step 5: Add Tags. The top navigation bar includes 'Services' and 'Resource Groups'. The breadcrumb path is 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (highlighted in orange), 6. Configure Security Group, and 7. Review. A user profile 'vocstartsoft/user522260=sant...' is visible. The main content area has tabs for 'Instances' and 'Volumes'. Below them, a message states 'This resource currently has no tags'. It includes instructions to 'Choose the Add tag button or click to add a Name tag.' and to ensure the 'IAM policy' includes tag creation permissions. An 'Add Tag' button is present, with a note '(Up to 50 tags maximum)'. The overall interface is clean with a dark header and light body.

Optional- you may want to add “**Name**” for the EC2.

This screenshot is identical to the one above, but it shows a single tag named 'Santoso1' has been added. The 'Name' field in the 'Add Tag' input row is filled with 'Santoso1'. The 'Instances', 'Volumes', and 'Network Interfaces' checkboxes are checked, indicating the tag will apply to all three. The 'Add another tag' button is visible below.

The screenshot shows the AWS EC2 instance creation wizard at Step 6: Configure Security Group. The top navigation bar includes links for Choose AMI, Choose Instance Type, Configure Instance, Add Storage, Add Tags, Configure Security Group (which is highlighted), and Review. A search bar and a keyboard shortcut [Alt+S] are also present.

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

**Assign a security group:**

- Create a **new** security group
- Select an **existing** security group

**Security group name:** Santoso1

**Description:** launch-wizard-6 created 2021-07-25T08:32:24.187-05:00

Type	Protocol	Port Range	Source	Description	Actions
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	X
All ICMP - IPv4	ICMP	0 - 65535	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	X

**Add Rule**

# Create a New Security Group

- Here you can set inbound firewall per instance basis.
- I create a new one and name it santoso1
- I add ICMP so we can ping to this EC2 from everywhere (0.0.0.0/0)
- Note: if we want to use http (web server), it also has to be allowed)

# Review

AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S]

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

**Step 7: Review Instance Launch**

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

**AMI Details** Edit AMI

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-09e67e426f25ce0d7

Free tier eligible Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebs Virtualization type: hvm

**Instance Type** Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

**Security Groups** Edit security groups

Security group name: Santos01  
Description: launch-wizard-6 created 2021-07-25T08:32:24.187-05:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
All ICMP - IPv4	All	N/A	0.0.0.0/0	

**Instance Details** Edit instance details

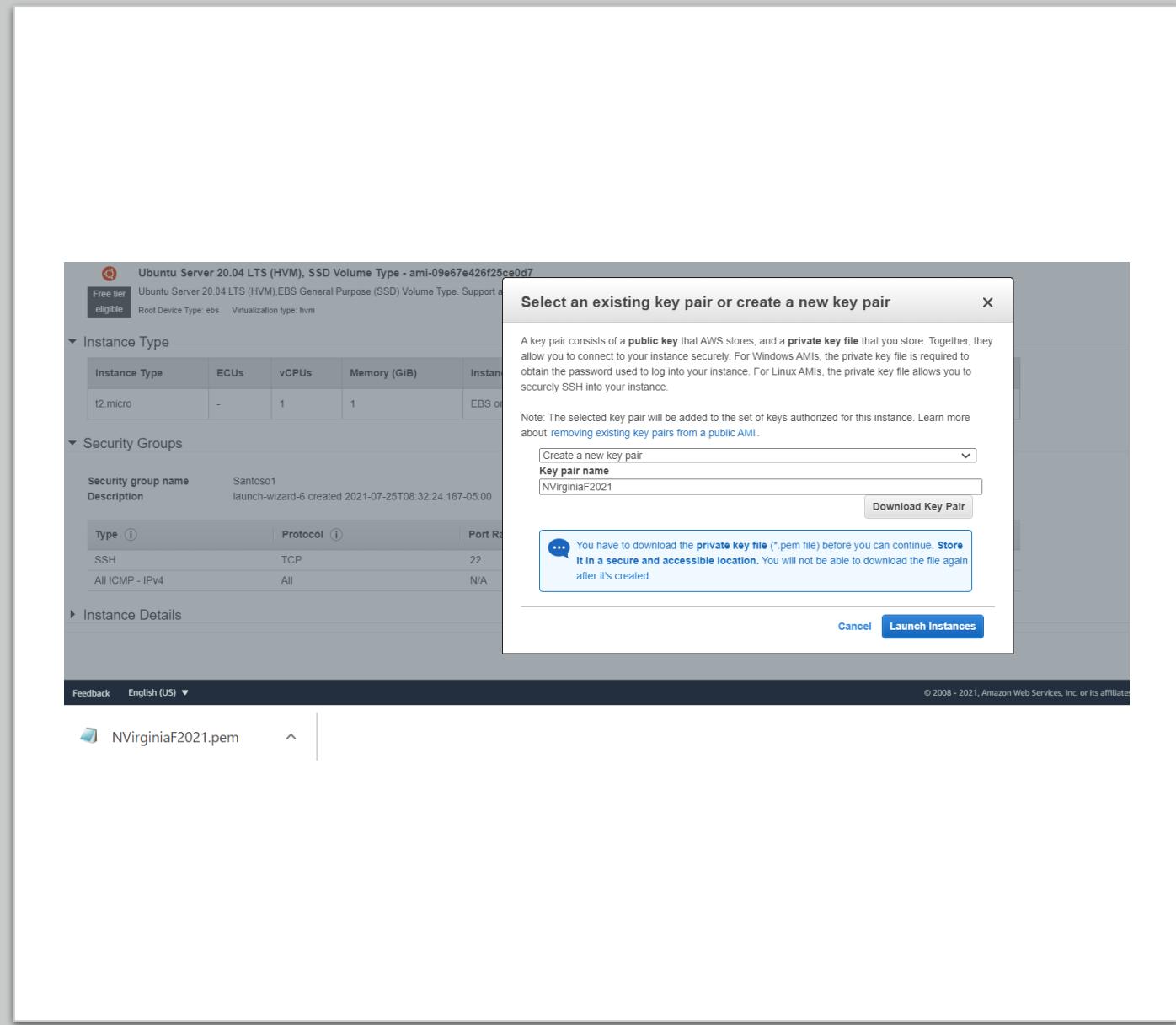
**Storage** Edit storage

**Tags** Edit tags

Cancel Previous Launch

# Key Pair

- Create, name, and download a key pair
- Can use existing one (same key pair can be used for multiple instances)
- **Key pair is tied to a region.**
- .pem must be saved in your local laptop drive.
- Convert .pem to .ppk to be used in Putty (this is your private key)
- Launch the instance



AWS Services ▾ Search for services, features, marketplace products, and docs [Alt+S] UTASantosoAWS N. Virginia Support

New EC2 Experience Tell us what you think

EC2 Dashboard Events Tags Limits

**Instances (1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Santoso1	i-09182853c68ae0dae	Running	t2.micro	-	No alarms	us-east-1a	ec2-44-192-120-27.compute-1.amazonaws.com	44.192.120.27

**Instances (1) Info**

Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name	Launch time
	ec2-44-192-120-27.compute-1.amazonaws.com	44.192.120.27	-	-	disabled	Santoso1	NVirginiaF2021	2021/07/25 08:56 GMT-5

# EC2 Detail

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area has a search bar at the top. Below it, a table lists one instance: Santoso1 (i-09182853c68ae0dae). The instance is running, t2.micro type, in us-east-1a, with a public IPv4 of 44.192.120.27. The bottom section provides detailed information for this instance.

**Instances (1/1) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Santoso1	i-09182853c68ae0dae	Running	t2.micro	-	No alarms	us-east-1a	ec2-44-192-120-27.compute-1.amazonaws.com	44.192.120.27

**Instance: i-09182853c68ae0dae (Santoso1)**

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
<b>Instance summary</b>						
Instance ID i-09182853c68ae0dae (Santoso1)	Public IPv4 address 44.192.120.27   <a href="#">open address</a>	Private IPv4 addresses 172.31.0.108				
Instance state Running	Public IPv4 DNS ec2-44-192-120-27.compute-1.amazonaws.com   <a href="#">open address</a>	Private IPv4 DNS ip-172-31-0-108.ec2.internal				
Instance type t2.micro	Elastic IP addresses -	VPC ID <a href="#">vpc-472bcd3a</a>				
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.   <a href="#">Learn more</a>	IAM Role -	Subnet ID <a href="#">subnet-b50cc6d3</a>				
<b>Instance details</b>						

# EC2 Security Group

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs), and Elastic Block Store (Volumes). The main content area has a header with 'Instances (1/1) Info' and a search bar. Below the header is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. One row is selected, showing 'Santoso1' with Instance ID 'i-09182853c68ae0dae', State 'Running', Type 't2.micro', and other details. Below the table, a section titled 'Instance: i-09182853c68ae0dae (Santoso1)' shows tabs for Details, Security (which is selected), Networking, Storage, Status checks, Monitoring, and Tags. Under the Security tab, there's a 'Security details' section with fields for IAM Role (empty), Owner ID (183451715204), and Launch time (Sun Jul 25 2021 08:56:00 GMT-0500 (Central Daylight Time)). There's also a 'Security groups' field containing 'sg-03c716697c2877685 (Santoso1)'. A 'Inbound rules' section follows, with a table showing Port range, Protocol, Source, and Security groups. Two entries are listed: one for port 22/TCP from 0.0.0.0/0 to 'Santoso1', and another for port -1/ICMP from 0.0.0.0/0 to 'Santoso1'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Santoso1	i-09182853c68ae0dae	Running	t2.micro	-	No alarms	us-east-1a	ec2-44-192-120-27.compute-1.amazonaws.com

Port range	Protocol	Source	Security groups
22	TCP	0.0.0.0/0	Santoso1
-1	ICMP	0.0.0.0/0	Santoso1

# EC2 Security Group

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links for New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs), and Elastic Block Store (Volumes). The main content area has a search bar at the top. Below it, a table lists one instance: Santoso1 (i-09182853c68ae0dae). The instance is running, of type t2.micro, in us-east-1a, with a public IPv4 DNS of ec2-44-192-120-27.compute-1.amazonaws.com. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Below the table, the details for the Santoso1 instance are expanded. The Inbound rules section shows two entries: port range 22 (TCP) and port range -1 (ICMP), both from source 0.0.0.0/0 and assigned to the security group Santoso1. The Outbound rules section shows a single entry: port range All (All) to destination 0.0.0.0/0, also assigned to the security group Santoso1.

Port range	Protocol	Source	Security groups
22	TCP	0.0.0.0/0	Santoso1
-1	ICMP	0.0.0.0/0	Santoso1

Port range	Protocol	Destination	Security groups
All	All	0.0.0.0/0	Santoso1

# Check Connectivity Using Ping command from Command Prompt

## Using Public IPv4 address

```
Pinging 44.192.120.27 with 32 bytes of data:  
Reply from 44.192.120.27: bytes=32 time=43ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=42ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=40ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=42ms TTL=47  
  
Ping statistics for 44.192.120.27:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 40ms, Maximum = 43ms, Average = 41ms
```

## Using Public IPv4 DNS name

```
Pinging ec2-44-192-120-27.compute-1.amazonaws.com [44.192.120.27] with 32 bytes of data:  
Reply from 44.192.120.27: bytes=32 time=39ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=42ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=41ms TTL=47  
Reply from 44.192.120.27: bytes=32 time=42ms TTL=47  
  
Ping statistics for 44.192.120.27:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 39ms, Maximum = 42ms, Average = 41ms
```

## Using Private IPv4 address

```
Pinging 172.31.0.108 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
  
Ping statistics for 172.31.0.108:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

## Using Private IPv4 DNS name

```
C:\Users\budimans>ping ip-172-31-0-108.ec2.internal  
Ping request could not find host ip-172-31-0-108.ec2.internal. Please check the name and try again.
```

The screenshot shows the AWS EC2 Instances page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('UTASantosoAWS', 'N. Virginia', 'Support'). On the left, a sidebar has 'New EC2 Experience' (with a 'Tell us what you think' link), 'EC2 Dashboard' (marked as 'New'), 'Events' (marked as 'New'), 'Tags', and 'Limits'. The main area has a 'Launch Instance' button and a search bar. A table lists two instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time	Security Groups	Owner
Santoso2	i-0d53c3d8142375eb4	t2.micro	us-east-1e	running	Initializing	None	ec2-54-234-98-148.co...	54.234.98.148	-	AWSSantoso	disabled	September 4, 2020 at 11:39...	Santoso2	183451715204
Santoso1	i-0ec22594f624ec1d8	t2.micro	us-east-1e	running	2/2 checks ...	None	ec2-100-25-142-102.co...	100.25.142.102	-	AWSSantoso	disabled	September 4, 2020 at 11:26...	Santoso1	183451715204

# Create another EC2

Notes:

- You can create a separate Security Group for each EC2
- An instance always has private IP address, it may have public IP address if set previously

# Connecting to the EC2 Instance

There are several ways to connect to an EC2 instance for administration purposes.

The 3 below are discussed in this chapter (ssh connection):

- ❑ EC2 Instance Connect

- ❑ can easily connect to instances from browser-based SSH experience in the EC2 console.
- ❑ US East (Ohio and N. Virginia), US West (N. California and Oregon), Asia Pacific (Mumbai, Seoul, Singapore, Sydney, and Tokyo), Canada (Central), EU (Frankfurt, Ireland, London, and Paris), and South America (São Paulo)
- ❑ Connect from API, CLI, or the EC2 console.
- ❑ control SSH access using AWS IAM policies

- ❑ SSH Client

- ❑ Can connect from your laptop command prompt
- ❑ Need Private Key (.pem) in laptop

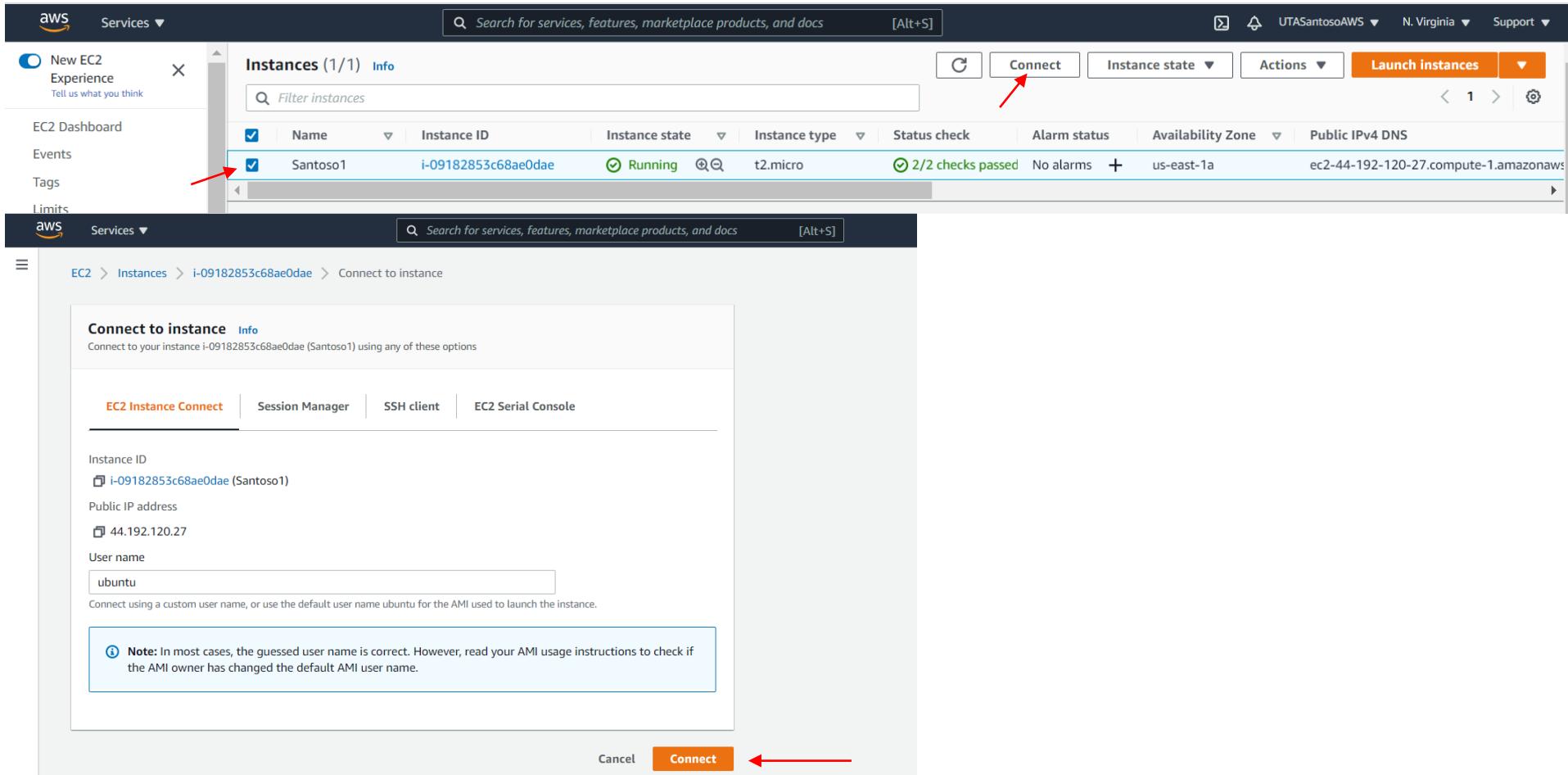
- ❑ Putty

- ❑ PuTTY is an SSH and telnet client
- ❑ Must convert private key from .pem to .ppk

You can also use AWS Systems Manager (Session Manager). You need to install SSM Agent

# SSH Connection – EC2 Instance Connect

# SSH Connection – EC2 Instance Connect (EC2 Console – browser based)



← → ⌂ [console.aws.amazon.com/ec2/v2/connect/ubuntu/i-09182853c68ae0dae](https://console.aws.amazon.com/ec2/v2/connect/ubuntu/i-09182853c68ae0dae)

```
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Mon Jul 26 00:17:04 UTC 2021

System load: 0.0          Processes: 105
Usage of /: 18.6% of 7.69GB Users logged in: 1
Memory usage: 24%          IPv4 address for eth0: 172.31.0.108
Swap usage: 0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

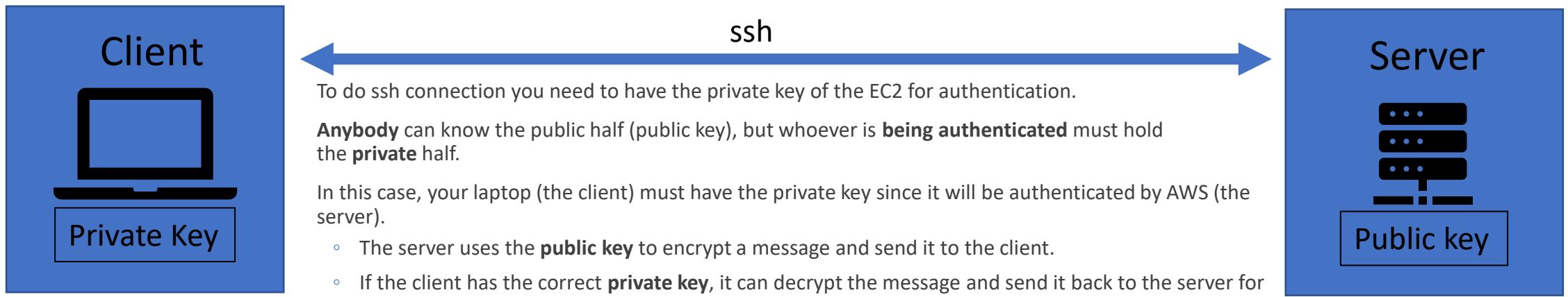
Last login: Sun Jul 25 23:58:53 2021 from 18.206.107.24
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-0-108:~$ █
```

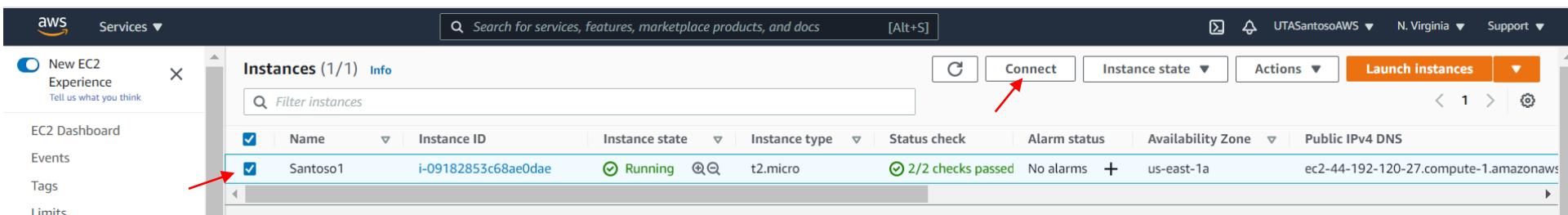
May need to refresh when timed out/hang

# SSH Connection – ssh client (through laptop)

# ssh Key-Pair for Authentication



Run ssh command from your laptop's command prompt



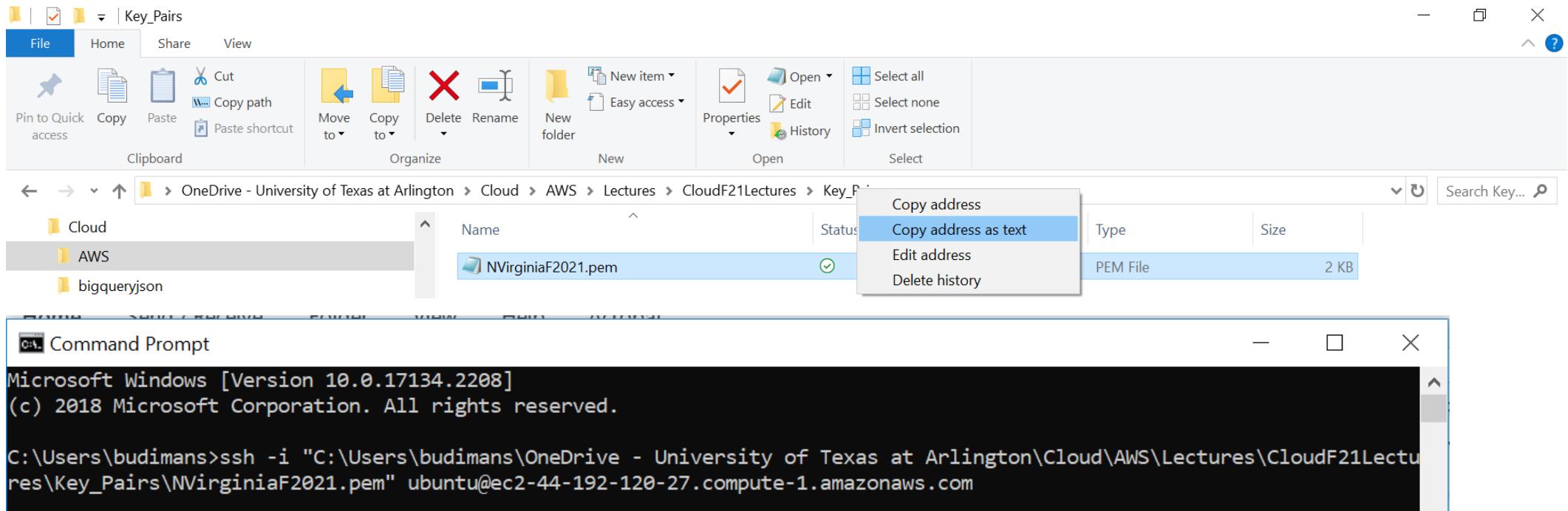
The screenshot shows the AWS EC2 Connect to instance page. At the top, there's a navigation bar with the AWS logo, 'Services ▾', a search bar containing 'Search for services, features, marketplace products, and docs [Alt+S]', and a breadcrumb trail: EC2 > Instances > i-09182853c68ae0dae > Connect to instance.

The main content area has a title 'Connect to instance' with a 'Info' link. Below it, a sub-instruction says 'Connect to your instance i-09182853c68ae0dae (Santoso1) using any of these options'. There are four tabs: 'EC2 Instance Connect', 'Session Manager', 'SSH client' (which is selected and highlighted in orange), and 'EC2 Serial Console'. Under the 'SSH client' tab, the 'Instance ID' is listed as 'i-09182853c68ae0dae (Santoso1)'. A numbered list of steps follows:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is NVirginiaF2021.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
     chmod 400 NVirginiaF2021.pem    **You may need to do this if you use Mac**
4. Connect to your instance using its Public DNS:  
     ec2-44-192-120-27.compute-1.amazonaws.com

Below this, there's an 'Example:' section with a copy link:  ssh -i "NVirginiaF2021.pem" ubuntu@ec2-44-192-120-27.compute-1.amazonaws.com. To the right, there are instructions for Windows users: 'For Windows users, copy this and paste on Command Prompt' and 'Add directory path to the .pem file'.

In a callout box at the bottom left, there's a note: **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.



- Goto laptop's command prompt
- run the ssh command (cut and paste from example in the previous page).
- Make sure to include path to the .pem file.
- If using mac, may need to do chmod 400 ...
- Sometimes use right click to paste

```
ubuntu@ip-172-31-0-108: ~
Microsoft Windows [Version 10.0.17134.2208]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\budimans>ssh -i "C:\Users\budimans\OneDrive - University of Texas at Arlington\Cloud\AWS\Lectures\CloudF21Lectures\Key_Pairs\NVirginiaF2021.pem" ubuntu@ec2-44-192-120-27.compute-1.amazonaws.com
The authenticity of host 'ec2-44-192-120-27.compute-1.amazonaws.com (44.192.120.27)' can't be established.
ECDSA key fingerprint is SHA256:J9N5b5Y4BsX5BcR+u2JrE4r4Amts9Atv0oRBfEZ8G0w.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-44-192-120-27.compute-1.amazonaws.com,44.192.120.27' (ECDSA) to the list of known hosts.

Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1045-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Jul 26 00:34:41 UTC 2021

System load:  0.0          Processes:           105
Usage of /:   18.6% of 7.69GB  Users logged in:      1
Memory usage: 24%          IPv4 address for eth0: 172.31.0.108
Swap usage:   0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

  https://ubuntu.com/blog/microk8s-memory-optimisation

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jul 26 00:22:20 2021 from 18.206.107.26
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-0-108:~$
```

```
ubuntu@ip-172-31-58-62:~  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-58-62:~$ ping -c3 172.31.51.128  
PING 172.31.51.128 (172.31.51.128) 56(84) bytes of data.  
64 bytes from 172.31.51.128: icmp_seq=1 ttl=64 time=0.569 ms  
64 bytes from 172.31.51.128: icmp_seq=2 ttl=64 time=0.432 ms  
64 bytes from 172.31.51.128: icmp_seq=3 ttl=64 time=0.396 ms  
  
--- 172.31.51.128 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2020ms  
rtt min/avg/max/mdev = 0.396/0.465/0.569/0.078 ms  
ubuntu@ip-172-31-58-62:~$ ping -c3 54.234.96.148  
  
Command 'ing' not found, but there are 20 similar ones.  
  
ubuntu@ip-172-31-58-62:~$ ping -c3 54.234.96.148  
PING 54.234.96.148 (54.234.96.148) 56(84) bytes of data.  
64 bytes from 54.234.96.148: icmp_seq=1 ttl=63 time=0.432 ms  
64 bytes from 54.234.96.148: icmp_seq=2 ttl=63 time=0.522 ms  
64 bytes from 54.234.96.148: icmp_seq=3 ttl=63 time=0.478 ms  
  
--- 54.234.96.148 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2026ms  
rtt min/avg/max/mdev = 0.432/0.477/0.522/0.040 ms  
ubuntu@ip-172-31-58-62:~$
```

- You can ping another EC2 within the same VPC using its private IP or its public IP (depending on your Security Group)

# Some Linux commands

FILE SYSTEM	SYSTEM	PROCESS MANAGEMENT	PERMISSIONS	NETWORKING
<p><code>ls</code> — list items in current directory</p> <p><code>ls -l</code> — list items in current directory and show in long format to see permissions, size, and modification date</p> <p><code>ls -a</code> — list all items in current directory, including hidden files</p> <p><code>ls -F</code> — list all items in current directory and show directories with a slash and executables with a star</p> <p><code>ls dir</code> — list all items in directory dir</p> <p><code>cd dir</code> — change directory to dir</p> <p><code>cd ..</code> — go up one directory</p> <p><code>cd /</code> — go to the root directory</p> <p><code>cd ~</code> — go to your home directory</p> <p><code>cd -</code> — go to the last directory you were just in</p> <p><code>pwd</code> — show present working directory</p> <p><code>mkdir dir</code> — make directory dir</p> <p><code>rm file</code> — remove file</p> <p><code>rm -r dir</code> — remove directory dir recursively</p> <p><code>cp file1 file2</code> — copy file1 to file2</p> <p><code>cp -r dir1 dir2</code> — copy directory dir1 to dir2 recursively</p> <p><code>mv file1 file2</code> — move (rename) file1 to file2</p> <p><code>ln -s file link</code> — create symbolic link to file</p> <p><code>touch file</code> — create or update file</p> <p><code>cat file</code> — output the contents of file</p> <p><code>less file</code> — view file with page navigation</p> <p><code>head file</code> — output the first 10 lines of file</p> <p><code>tail file</code> — output the last 10 lines of file</p> <p><code>tail -f file</code> — output the contents of file as it grows, starting with the last 10 lines</p> <p><code>vim file</code> — edit file</p> <p><code>alias name 'command'</code> — create an alias for a command</p>	<p><code>shutdown</code> — shut down machine</p> <p><code>reboot</code> — restart machine</p> <p><code>date</code> — show the current date and time</p> <p><code>whoami</code> — who you are logged in as</p> <p><code>finger user</code> — display information about user</p> <p><code>man command</code> — show the manual for command</p> <p><code>df</code> — show disk usage</p> <p><code>du</code> — show directory space usage</p> <p><code>free</code> — show memory and swap usage</p> <p><code>whereis app</code> — show possible locations of app</p> <p><code>which app</code> — show which app will be run by default</p>	<p><code>ps</code> — display your currently active processes</p> <p><code>top</code> — display all running processes</p> <p><code>kill pid</code> — kill process id pid</p> <p><code>kill -9 pid</code> — force kill process id pid</p>	<p><code>ls -l</code> — list items in current directory and show permissions</p> <p><code>chmod ugo file</code> — change permissions of file to ugo - u is the user's permissions, g is the group's permissions, and o is everyone else's permissions. The values of u, g, and o can be any number between 0 and 7.</p> <p>7 — full permissions 6 — read and write only 5 — read and execute only 4 — read only 3 — write and execute only 2 — write only 1 — execute only 0 — no permissions</p> <p><code>chmod 600 file</code> — you can read and write - good for files</p> <p><code>chmod 700 file</code> — you can read, write, and execute - good for scripts</p> <p><code>chmod 644 file</code> — you can read and write, and everyone else can only read - good for web pages</p> <p><code>chmod 755 file</code> — you can read, write, and execute, and everyone else can read and execute - good for programs that you want to share</p>	<p><code>wget file</code> — download a file</p> <p><code>curl file</code> — download a file</p> <p><code>scp user@host:file dir</code> — secure copy a file from remote server to the dir directory on your machine</p> <p><code>scp file user@host:dir</code> — secure copy a file from your machine to the dir directory on a remote server</p> <p><code>scp -r user@host:dir dir</code> — secure copy the directory dir from remote server to the directory dir on your machine</p> <p><code>ssh user@host</code> — connect to host as user</p> <p><code>ssh -p port user@host</code> — connect to host on port as user</p> <p><code>ssh-copy-id user@host</code> — add your key to host for user to enable a keyed or passwordless login</p> <p><code>ping host</code> — ping host and output results</p> <p><code>whois domain</code> — get information for domain</p> <p><code>dig domain</code> — get DNS information for domain</p> <p><code>dig -x host</code> — reverse lookup host</p> <p><code>lsof -i tcp:1337</code> — list all processes running on port 1337</p>

<http://cheatsheetworld.com/programming/unix-linux-cheat-sheet/>

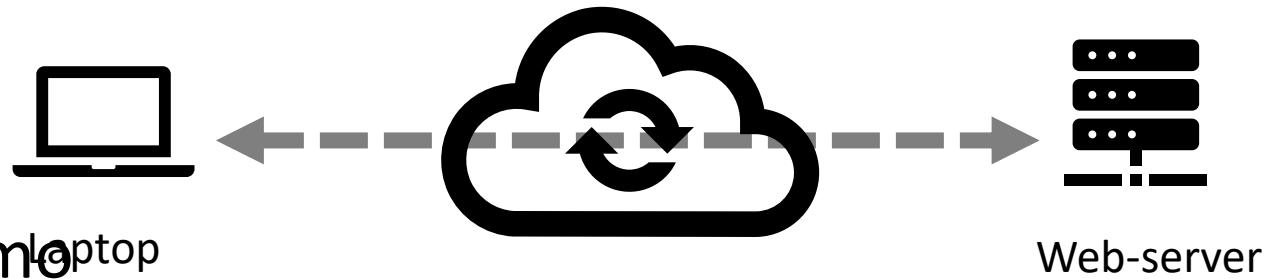
Command Prompt

```
ubuntu@ip-172-31-0-108:~$ #clear to clear the screen
ubuntu@ip-172-31-0-108:~$ #ls -l is the long listing of files including permission and size
ubuntu@ip-172-31-0-108:~$ #ls -a listing all including hidden files
ubuntu@ip-172-31-0-108:~$ ls
ubuntu@ip-172-31-0-108:~$ ls -l
total 0
ubuntu@ip-172-31-0-108:~$ ls -a
. .. .bash_history .bash_logout .bashrc .cache .profile .ssh
ubuntu@ip-172-31-0-108:~$ ls -al
total 32
drwxr-xr-x 4 ubuntu ubuntu 4096 Jul 25 22:45 .
drwxr-xr-x 3 root root 4096 Jul 25 13:56 ..
-rw----- 1 ubuntu ubuntu 10 Jul 26 00:23 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Feb 25 2020 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Jul 25 20:45 .cache
-rw-r--r-- 1 ubuntu ubuntu 807 Feb 25 2020 .profile
drwx----- 2 ubuntu ubuntu 4096 Jul 25 13:56 .ssh
ubuntu@ip-172-31-0-108:~$ whoami
ubuntu
ubuntu@ip-172-31-0-108:~$ #print working directory pwd
ubuntu@ip-172-31-0-108:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-0-108:~$ #cd / to go to root directory
ubuntu@ip-172-31-0-108:~$ cd /
ubuntu@ip-172-31-0-108:/$ ls
bin dev home lib32 libx32 media opt root sbin srv tmp var
boot etc lib lib64 lost+found mnt proc run snap sys usr
ubuntu@ip-172-31-0-108:/$ cd
ubuntu@ip-172-31-0-108:~$ ls
ubuntu@ip-172-31-0-108:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-0-108:~$ #cd to go to home directory
ubuntu@ip-172-31-0-108:~$ # man for manual
ubuntu@ip-172-31-0-108:~$ man ls
ubuntu@ip-172-31-0-108:~$ exit
logout
Connection to ec2-44-192-120-27.compute-1.amazonaws.com closed.
```

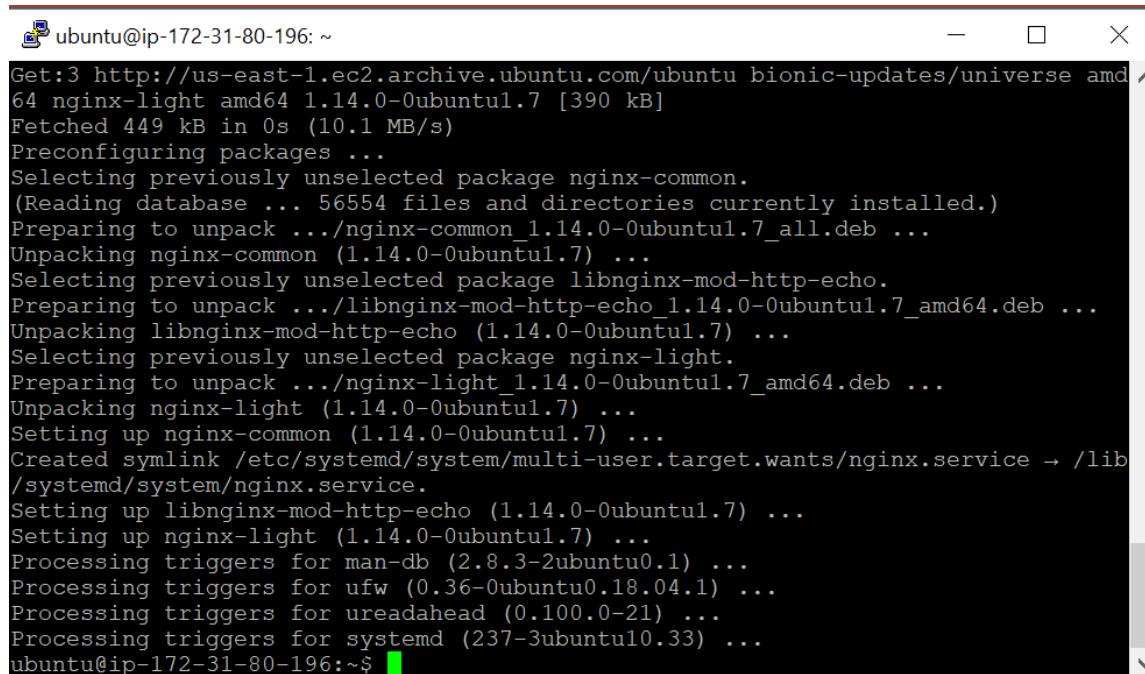
# Installing a Web-Server in EC2

# Web-Server

- Web-server is a server that delivers requested web pages
  - It has an IP address (and a domain name).
- Any computer can be a web-server as long as it has the necessary Web-server software, such as:
  - Nginx - an open-source Web server software
  - Apache – together with Nginx are the two most common open-source web server
  - Tornado
  - XAMPP
- We will use **Nginx** for demo



Nginx is an open-source web server. We will use this to do our homework.



A terminal window titled "ubuntu@ip-172-31-80-196: ~" showing the output of an apt-get install command. The command installed nginx-light, nginx-common, libnginx-mod-http-echo, and their dependencies. The terminal shows the progress of the download, unpacking, and configuration of these packages.

```
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 nginx-light amd64 1.14.0-0ubuntu1.7 [390 kB]
Fetched 449 kB in 0s (10.1 MB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 56554 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.14.0-0ubuntu1.7_all.deb ...
Unpacking nginx-common (1.14.0-0ubuntu1.7) ...
Selecting previously unselected package libnginx-mod-http-echo.
Preparing to unpack .../libnginx-mod-http-echo_1.14.0-0ubuntu1.7_amd64.deb ...
Unpacking libnginx-mod-http-echo (1.14.0-0ubuntu1.7) ...
Selecting previously unselected package nginx-light.
Preparing to unpack .../nginx-light_1.14.0-0ubuntu1.7_amd64.deb ...
Unpacking nginx-light (1.14.0-0ubuntu1.7) ...
Setting up nginx-common (1.14.0-0ubuntu1.7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Setting up libnginx-mod-http-echo (1.14.0-0ubuntu1.7) ...
Setting up nginx-light (1.14.0-0ubuntu1.7) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu0.33) ...
ubuntu@ip-172-31-80-196:~$
```

The below commands work on **ubuntu but not on Amazon Linux**

- sudo apt-get update
- sudo apt-get install nginx-light -y

Notes:

sudo- run commands as a superuser

apt-get is a command-line tool which helps in handling packages in Linux

# Edit nginx webserver

---

HyperText Markup Language (HTML) is the basic scripting language that is used to structure a web page and its content. When you install the nginx software in an EC2, an html file is stored in a certain directory. You can edit this file to change what will appear in the webpage.

Edit using nano editor:

```
sudo nano /var/www/html/index.nginx-debian.html
```

- add above <h1>Welcome nginx...:</h1>
- <h1>This is the 2<sup>nd</sup> machine – your name </h1>
- **Press Ctrl O, Enter, Ctrl X**

← → ⌂ 🔒 console.aws.amazon.com/ec2/v2/connect/ubuntu/i-09182853c68ae0dae

GNU nano 4.8 /var/www/html/index.nginx-debian.html Modified

```
<!DOCTYPE html> <html> <head> <title>Welcome to nginx!</title> <style> body { width: 35em; margin: 0 auto; font-family: Tahoma, Verdana, Arial, sans-serif; } </style> </head> <body> <h1>Santoso Budiman<h1> <h1>Welcome to nginx!</h1> <p>If you see this page, the nginx web server is successfully installed and working. Further configuration is required.</p> <p>For online documentation and support please refer to <a href="http://nginx.org/">nginx.org</a>. <br/> Commercial support is available at <a href="http://nginx.com/">nginx.com</a>.</p> <p><em>Thank you for using nginx.</em></p> </body> </html>
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text M-] To Bracket M-Q Previous  
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^\_ Go To Line M-E Redo M-6 Copy Text ^Q Where Was M-W Next

# Update Security group on the EC2 to allow http

Inbound rules [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
SSH	TCP	22	Custom <a href="#">▼</a>	0.0.0.0/0 <a href="#">X</a>
All ICMP - IPv4	ICMP	All	Custom <a href="#">▼</a>	0.0.0.0/0 <a href="#">X</a>
HTTP	TCP	80	Custom <a href="#">▼</a>	0.0.0.0/0 <a href="#">X</a>

**Add rule**

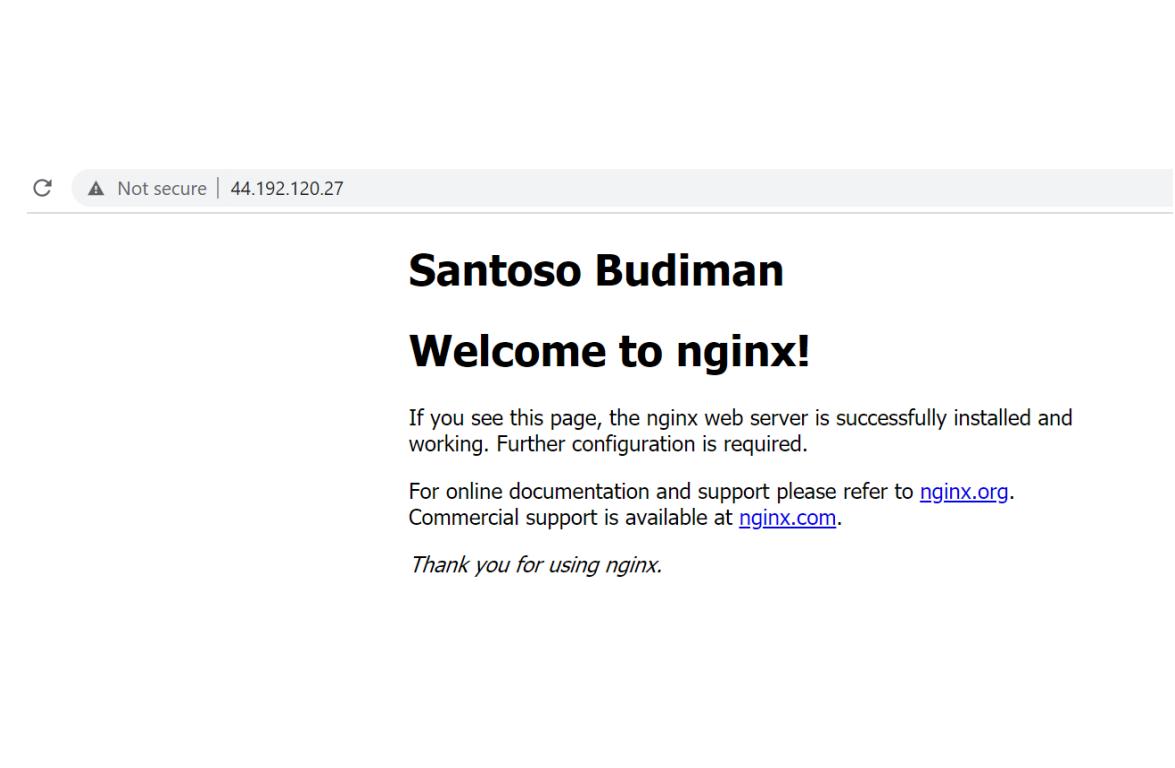
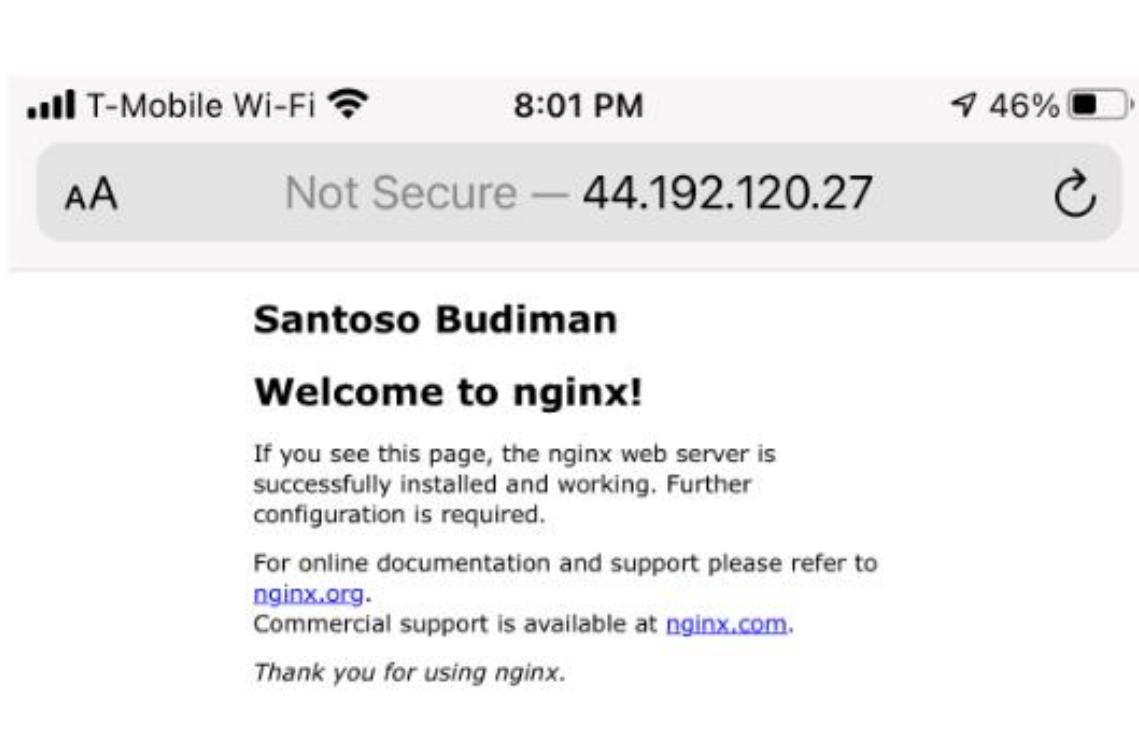
**0.0.0.0/0 is from everywhere**

**Example From specific IP address only 47.185.208.61/32**

**NOTE:** Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

# Test the website

---





# Bastion Host

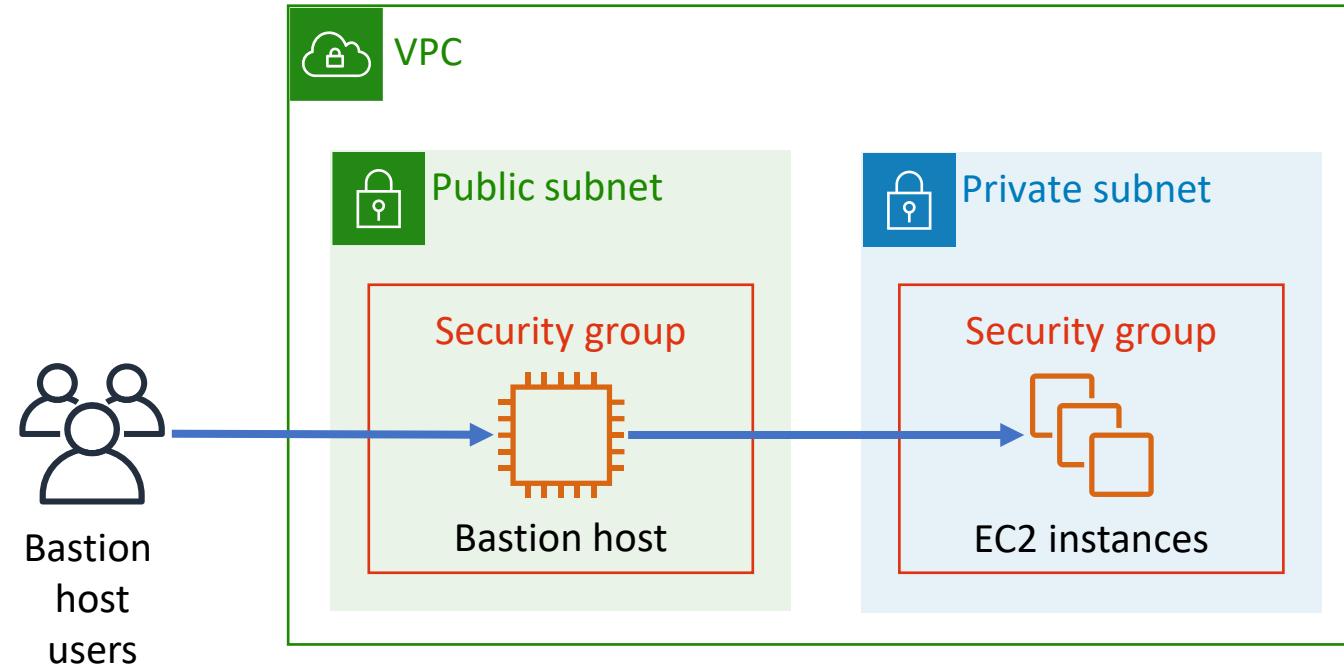


# Bastion

- Bastion - A **bastion** or **bulwark** is a structure projecting outward from the curtain wall of a fortification.
- Provide front-end protection.
- A **bastion host** is a **server** whose purpose is to provide access to a private network from an external network, such as the Internet.
  - minimizes the chances of penetration (exposure to potential attack).

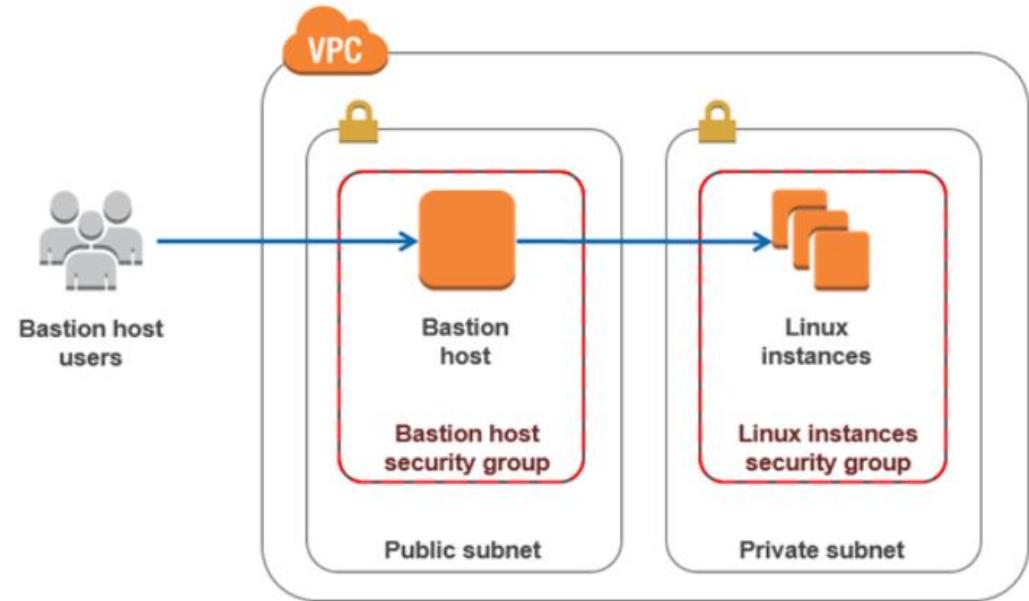
# Bastion hosts

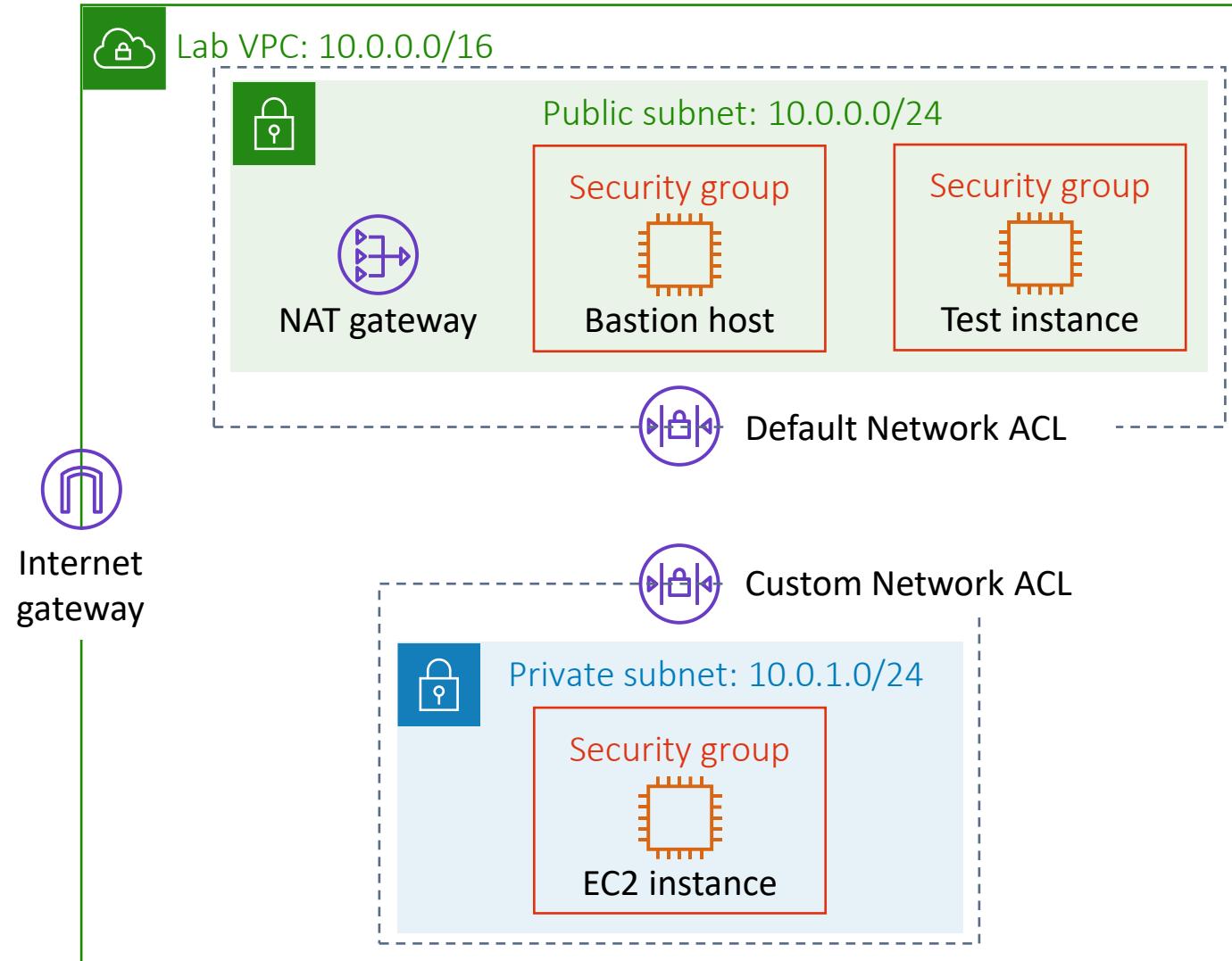
- A server whose purpose is to provide access to a private network from an external network
- Must minimize the chances of penetration



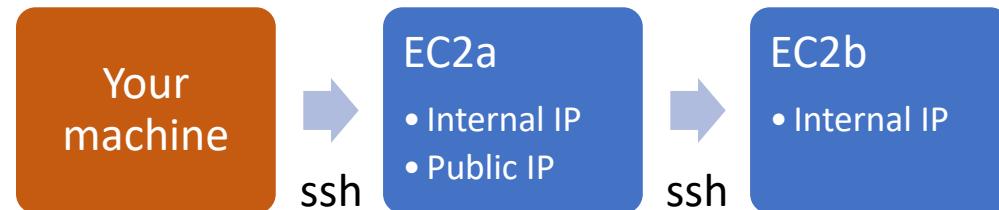
# AWS Bastion Host

- Bastion host for security.
- Bastion host can be used to mitigate the risk of allowing SSH connections from an external network to the Linux instances launched in a private subnet of an AWS VPC.
- The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of the VPC.
- Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host.
- **Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.**  
<https://aws.amazon.com/blogs/security/how-to-record-ssh-sessions-established-through-a-bastion-host/>





# SSH from EC2 to EC2 (if the 2<sup>nd</sup> EC2 does not have a public IP)



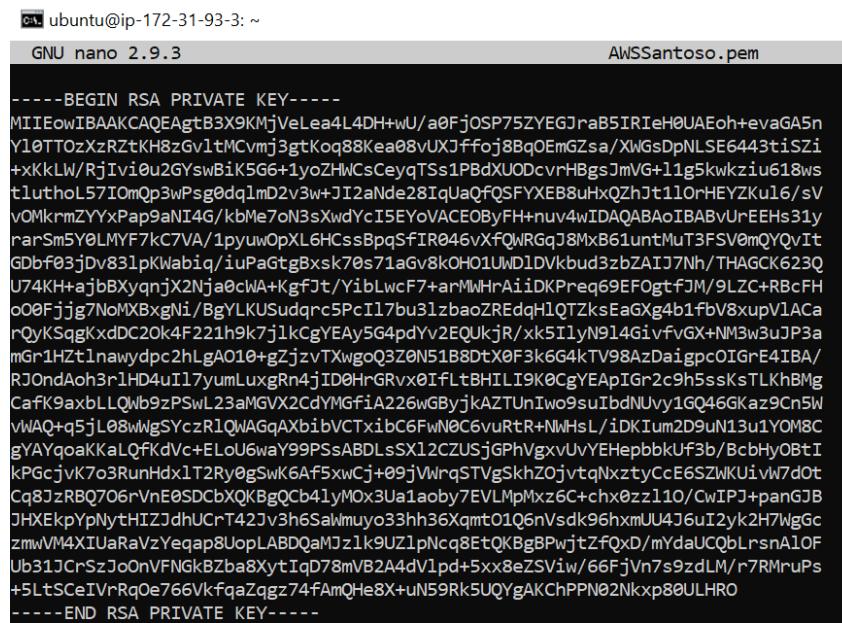
- Use PuTTy/other method to ssh to EC2a using its public IP
- Copy .pem (key pair) to EC2a
  - Use nano editor to cut and paste/WinSCP
  - make sure r for owner (do chmod 400 “the.pem file”)
  - <https://www.linux.com/training-tutorials/understanding-linux-file-permissions/>
- Do: ssh –i “the.pem” **xxx**@the internal ip address of EC2b
  - **xxx is** ec2-user for Amazon Linux or ubuntu for ubuntu

# SSH to an EC2a which has a Public IP

After SSH to EC2a:

- Copy the .pem to EC2a (either cut and paste or copy using WinSCP).
- Below is copy and paste using nano editor
- Copy from .pem from your laptop and paste to a document with the same name in EC2a (this is your private key)

```
ubuntu@ip-172-31-93-3:~$ nano AwSSantoso.pem
```



```
-----BEGIN RSA PRIVATE KEY-----  
MIIEowIBAAKCAQEAgkB3X9KMjVeLea4L4DH+wU/a0Fj0SP75ZYEGJraB5IRIeH0UAEoh+evaGA5n  
Y10TT0zXrZtKH8zGv1tMCVmj3gtKoq88Kea08vUX3ffojBbqOEmGZsa/XlVgsDpNLSE6443tiSzI  
+xKkLW/RjIvi0u2GYswBiK5G6+1yoZHWsCeyqTSs1PBdXUODcvrHebsJmVG+1ig5kwkzui618ws  
tluthoL57IOmQp3wPsg0dq1mD2v3w+JI2aMde28IqJa0fQSFYXE8uHxQZhjt11orHEYZKu16/sV  
vOMkrmZYYxPap9aNi4G/kbMe7ON3sXwdYcI5Ey0VACE0ByFH+nuv4wIDAQABAoIBABvUrEEHs31y  
rarSm5Y0LMyF7kC7VA/1pyuwOpXL6HCssBpqSfIR046vXfqWRGqj8MxB61untMuT3FSV0mQYQvIt  
GDbf03jDv83lpKwabiq/iuPaGtgBxsk70s71aGv8k0HO1UwD1Dvkbud3zbZAIJ7Nh/THAGCK623Q  
U74KHHajbBXynqjX2Nja0cwA+kgfJt/YibLwcf7+arMWHRaiiDKPreq69EFogtfJM/9LZC+RbcFH  
o00Fjjg7NoMXBxgNi/BgYLKUSudqrcPcIl7bu3lzbaoZREddqhlQTZksEaXgxg4b1fbv8xupv1ACa  
rQyKSqgKxdDC20k4F221h7j1kCgYEAy5G4pdYv2EQUkjr/xk51lyN914GivfvGX+NMBw3uJP3a  
mGr1Hzt1nawydpch2hLgAO10+gjzvTXwgoQ3Z0N518Dt0xF3kG4kTV98AzDaigpcOIGrE4IBA/  
RJOnDAoh3r1HD4uI17yumLuxgRn4jID0HrGRvx0IfLtBHIL19K0CgYEApIGr2c9h5ssKsTLKhBmg  
CaFk9axbLLQwb9zPSwL23aMGVX2CdYMgfIA226wGByjkAZTUuIwo9suIbdnUvy1GQ46GKaz9Cn5W  
vWAQ+j5jL08wlgSYczR1QwAGqAXbibvCTxiBc6fwN0C6vurR+NMhsL/iDKIum2D9uN13u1Y0M8C  
gYAYqoaKKaLQfKdVc+ELoU6waY99PSsABDLsSX12CZUSjGPPhVgxvUvYEHepbbkUF3b/BcbHyOBtI  
kPGcjvK7o3RunHdx1T2Ry0gSwk6Af5xwCj+09jWlrqSTVgSkhZ0jvtqNxztyCcE6SZWKUivw7dOt  
Cq8JzRBQ706rVnE0SDCbXQkBqC64lyMoX3ua1aoby7EVLMpMxz6C+chx0zz110/CwIPJ+panGJB  
JHxEkpYpNy0tHIZJdhUCrT42Jv3h6SaWmuyo3hh36Xqmt01Q6nVsdk96hxmUU4J6uI2yk2H7WgGc  
zmwVh4XIUaRaVzYeqap8UopLABDQaMJz1k9UZ1pNcq8EtQKBgBPwjtZfQxD/mYdaUCQbLrsnAlOF  
Ub31JCrSzJo0nVFNGkBZba8XytIqD78mVB2A4dVlpd+5xx8eZSVi/66FjVn7s9zdLM/r7RMruPs  
+5LtSCeIVrRq0e766VkfqaZqgz74fAmQHe8X+uN59Rk5UQyAKChPPN02Nkxp80ULHRO  
-----END RSA PRIVATE KEY-----
```

Save and close: ctrl o,enter, ctrl x

Check the file permissions

```
ubuntu@ip-172-31-93-3:~$ ls -l  
total 4  
-rw----- 1 ubuntu ubuntu 1671 Sep  4 22:39 AwSSantoso.pem  
ubuntu@ip-172-31-93-3:~$
```

# SSh from EC2a to EC2b

```
ssh -i "AWSSantoso.pem" ubuntu@172.31.35.127
```

```
ubuntu@ip-172-31-93-3:~$ ssh -i "AWSSantoso.pem" ubuntu@172.31.35.127
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.3.0-1034-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri Sep  4 22:42:12 UTC 2020

System load:  0.0          Processes:      90
Usage of /:   14.6% of 7.69GB  Users logged in:  0
Memory usage: 17%          IP address for eth0: 172.31.35.127
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

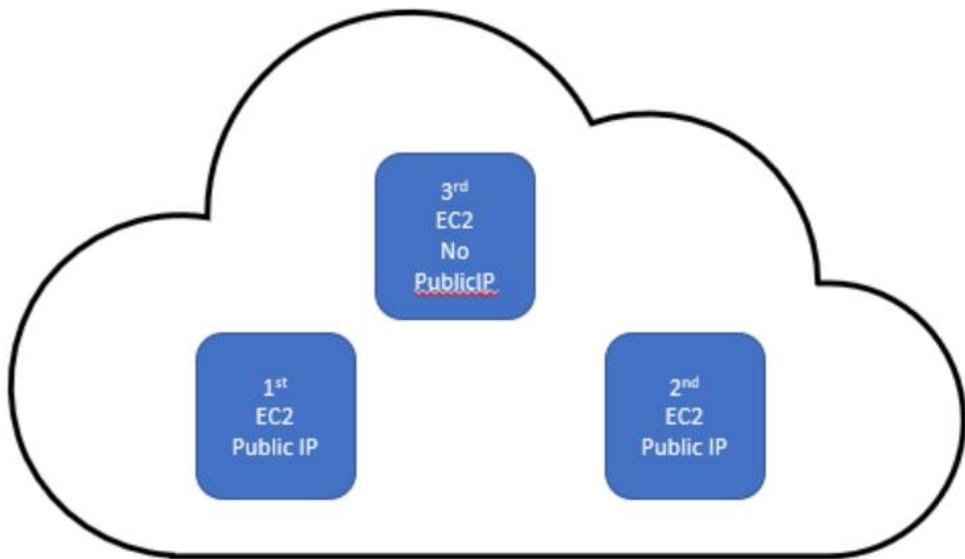
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Sep  4 21:23:16 2020 from 172.31.93.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-35-127:~$
```

You may need to manually type the command, sometimes copy paste does not work

# Demo (similar to the homework but NOT the same)



Note:

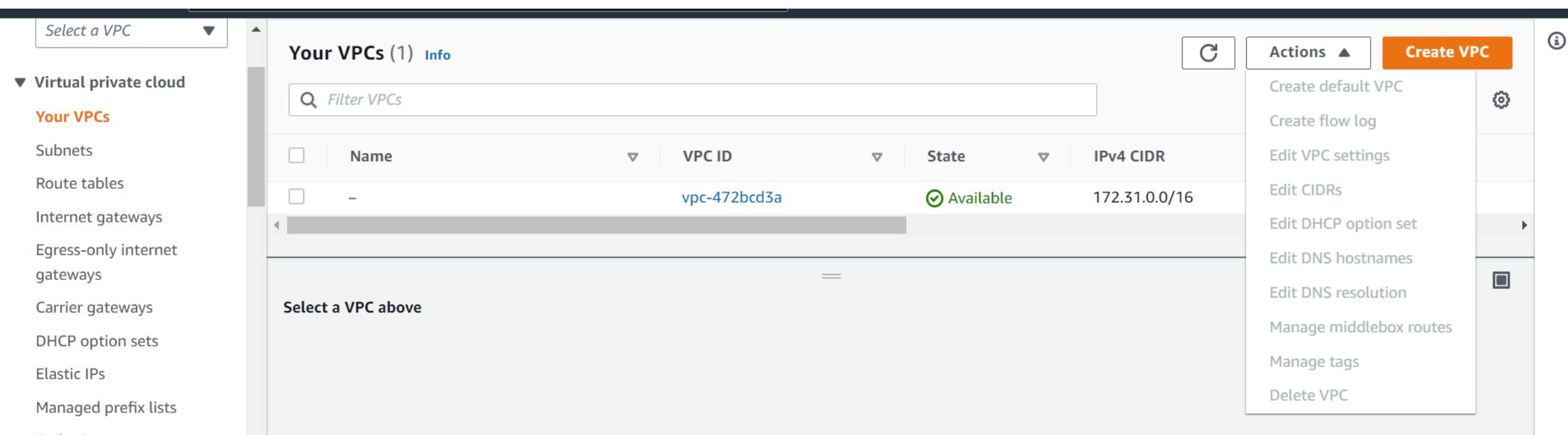
1. Each EC2 has an internal IP address but not always has public IP.
2. You can ping with internal/external IP address depending on the Security Group.
3. Ping and ssh must use public IP address if sent from different network.
4. How do you ssh to the 3<sup>rd</sup> EC2 (no public IP)?
5. Show the 3 EC2 instances & their SGs.

Spin up 3EC2s:

- 1<sup>st</sup> EC2 – web server
  - Has public IP
  - Allow SSH from everywhere
  - Allow ping from everywhere
  - Allow http call from everywhere
  - Bastion host for 3<sup>rd</sup> EC2
  - Install nginx and edit the html file
- 2<sup>nd</sup> EC2- general purpose EC2
  - Has public ip
  - Allow SSH from everywhere
  - Allow ping from everywhere
- 3<sup>rd</sup> EC2 – internal use only
  - No public IP
  - Only allow ssh from 1<sup>st</sup> EC2
  - Only allow ping from 1<sup>st</sup> EC2
  - Remember, this EC2 does not have a public IP. It only recognize internal IP address.

Note: in the HW, you were asked to only allow traffic from your laptop in some cases. You may use <https://ip4.me/> to find your laptop's IP.

# If Default VPC is deleted by Accident (I have not tried this)



The screenshot shows the AWS VPC console interface. On the left, a sidebar menu under 'Virtual private cloud' includes 'Your VPCs' (which is selected), 'Subnets', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', and 'Managed prefix lists'. The main area is titled 'Your VPCs (1)' and contains a table with one row. The table columns are 'Name' (with a dropdown arrow), 'VPC ID' (containing 'vpc-472bcd3a'), 'State' (containing 'Available' with a green checkmark), and 'IPv4 CIDR' (containing '172.31.0.0/16'). Below the table, a message says 'Select a VPC above'. To the right of the table is a 'Actions' menu with options: 'Create default VPC' (with a gear icon), 'Create flow log', 'Edit VPC settings', 'Edit CIDRs', 'Edit DHCP option set' (with a right-pointing arrow), 'Edit DNS hostnames', 'Edit DNS resolution' (with a square icon), 'Manage middlebox routes', 'Manage tags', and 'Delete VPC'.

Name	VPC ID	State	IPv4 CIDR
-	vpc-472bcd3a	Available	172.31.0.0/16

Select a VPC above

Actions ▲ Create VPC

- Create default VPC
- Create flow log
- Edit VPC settings
- Edit CIDRs
- Edit DHCP option set
- Edit DNS hostnames
- Edit DNS resolution
- Manage middlebox routes
- Manage tags
- Delete VPC

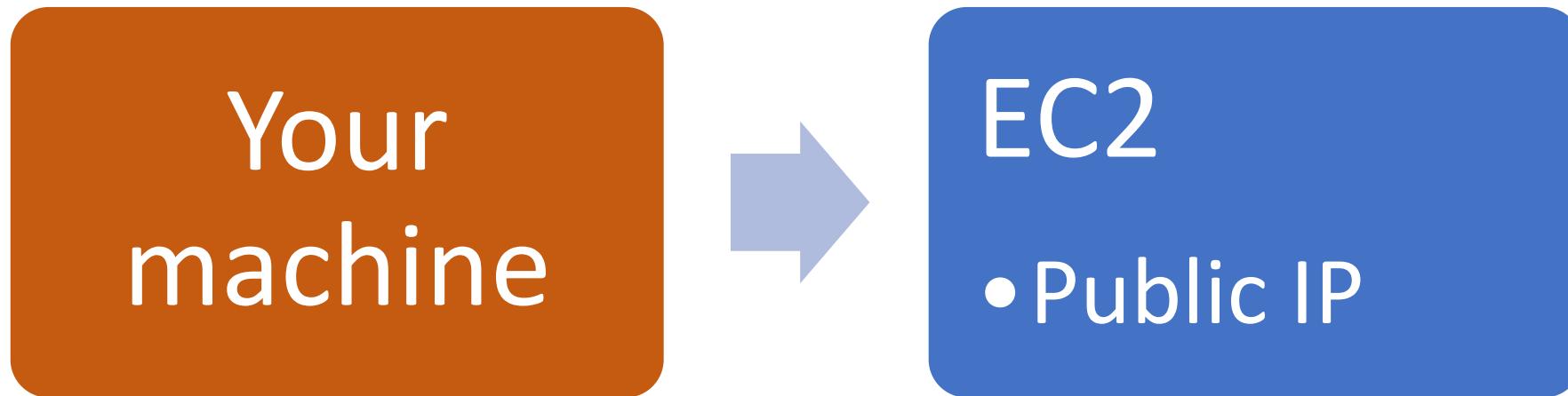
DO NOT DELETE DEFAULT VPC BUT THERE IS A "CREATE DEFAULT VPC" UNDER ACTION. I NEVER TRIED THIS AND HOPEFULLY YOU DON'T NEED TO DO THIS EITHER.

# End Of Lecture

# WINSCP

Transferring Files

# Connection and File Transfer

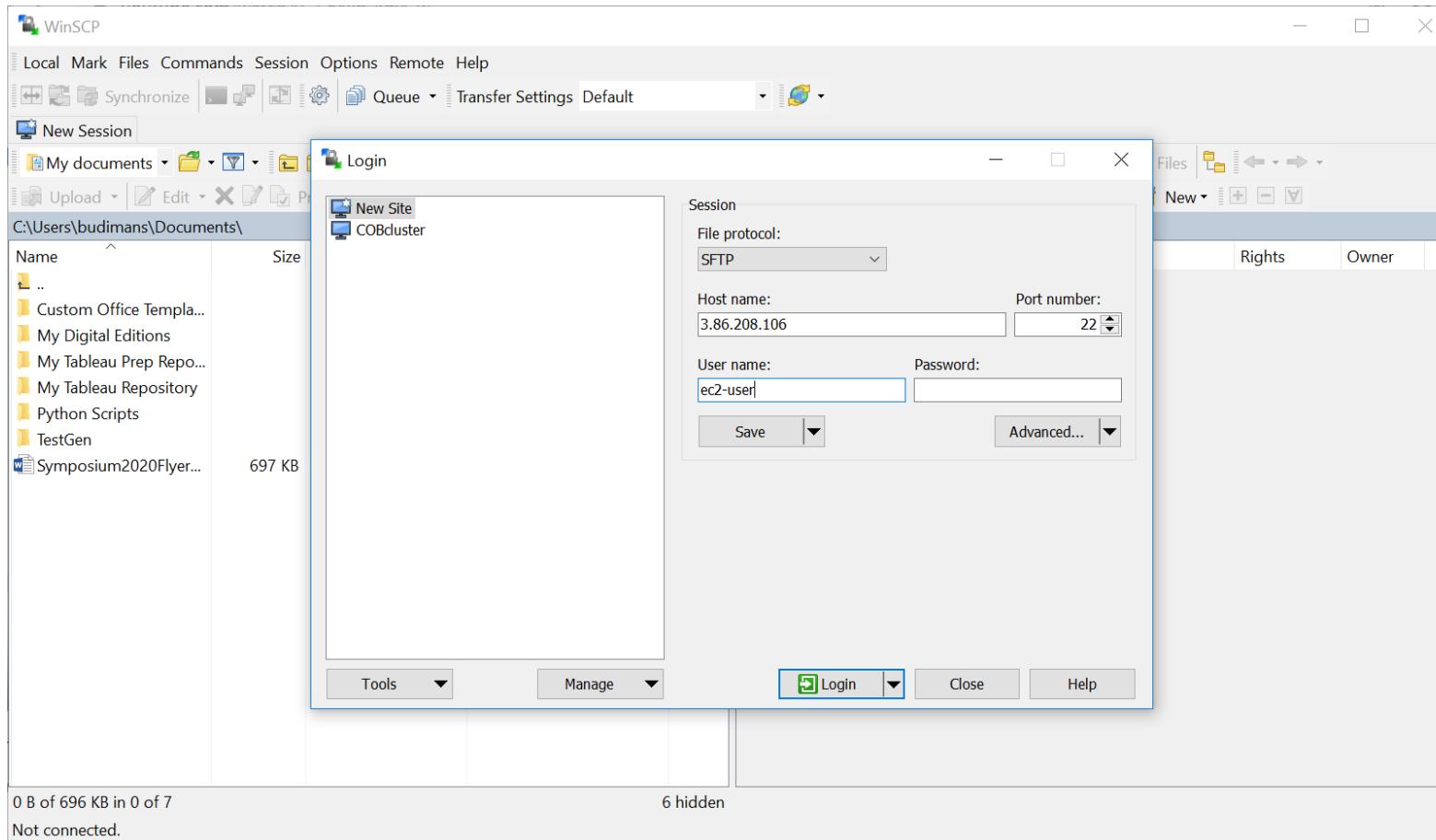


Putty - ssh  
WinSCP – transfer files

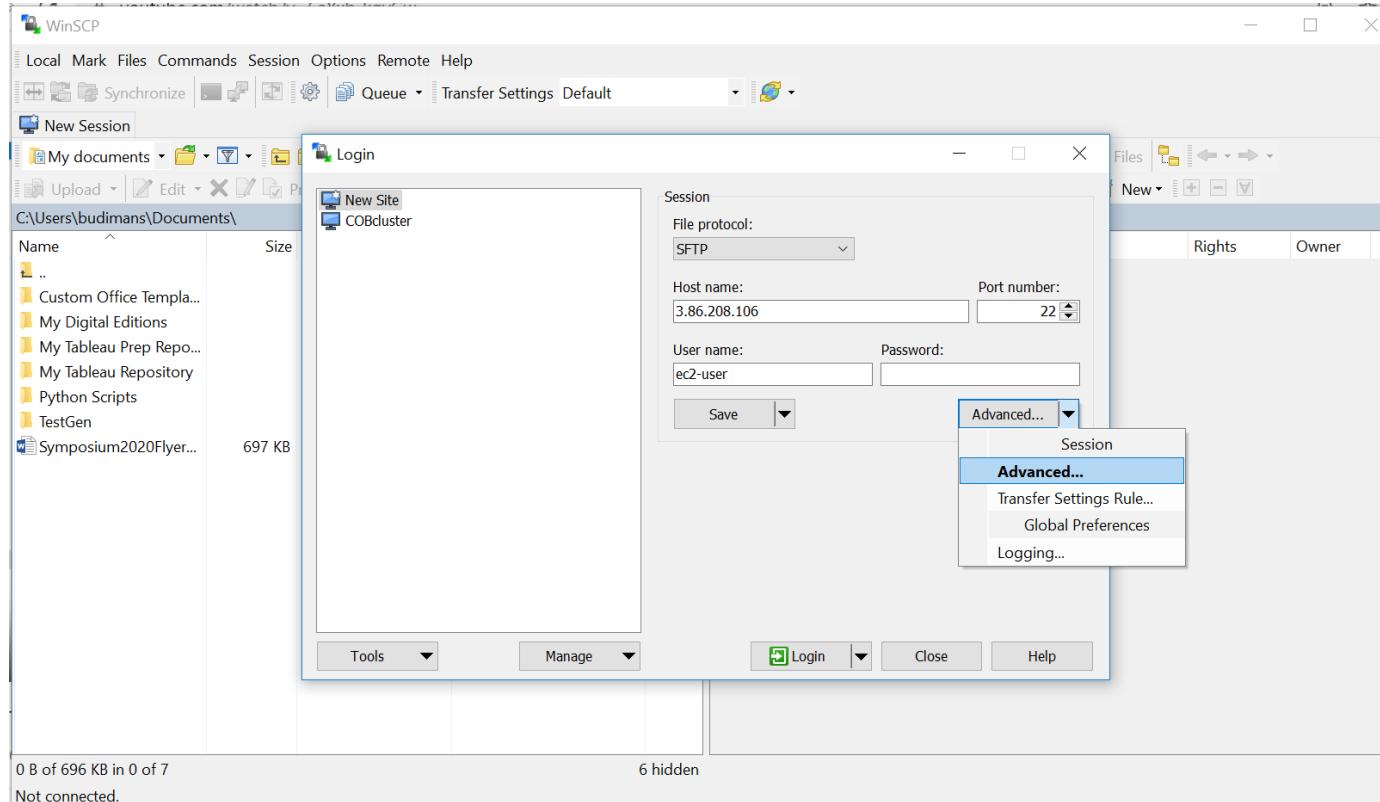
# WinSCP vs PuTTY

- **WinSCP** (*Windows Secure Copy*) is a free and open-source.
- **Secure Copy Protocol (SCP)** is for secured data transfer and is based on **SSH** protocol.
- **WinSCP** is for file transfer; **PuTTY** is to interact with the server directly.
- Download WinSCP from the following website
  - <https://winscp.net/eng/index.php>

# File Transfer using WinSCP



# File Transfer using WinSCP

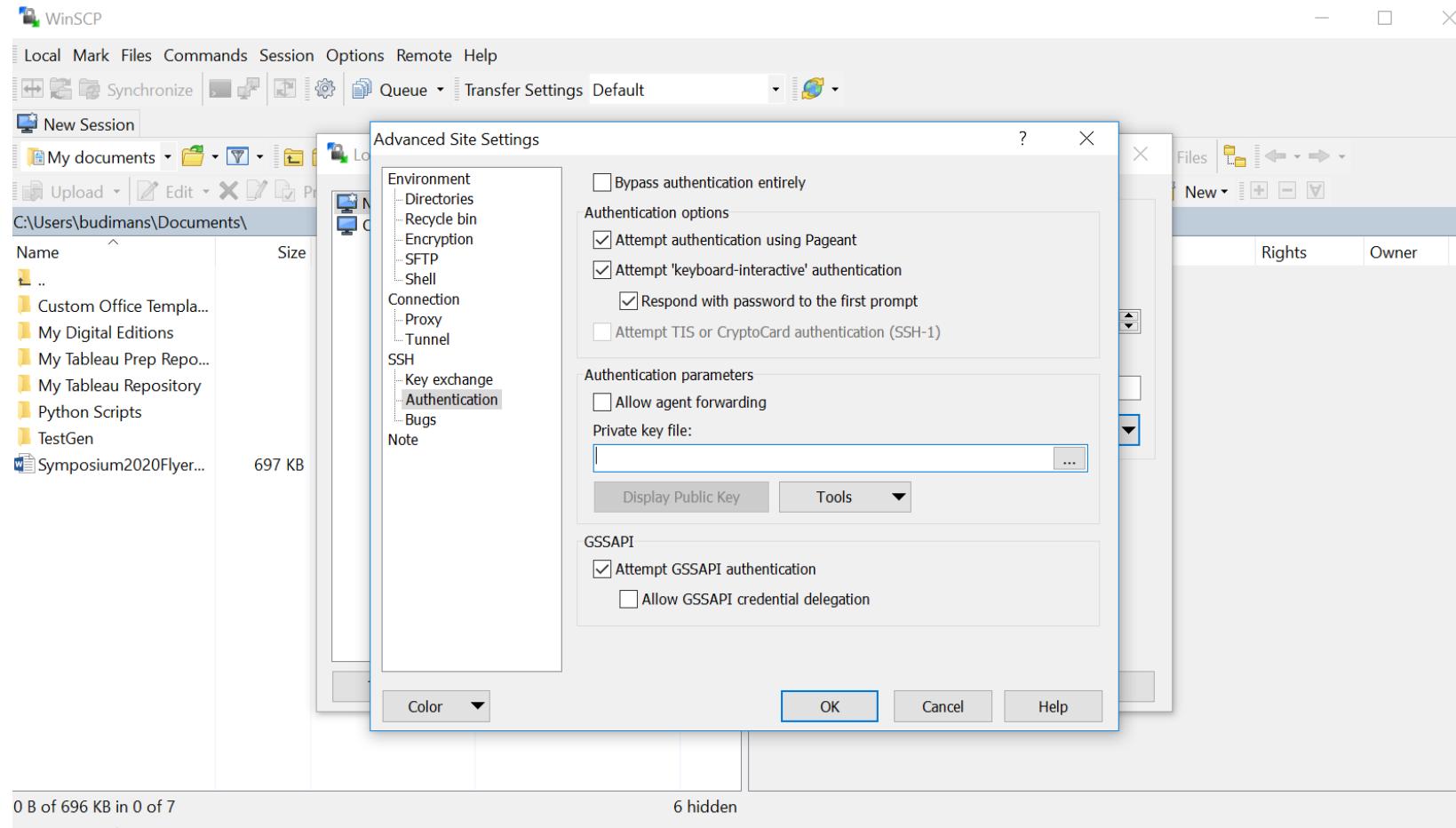


For Amazon Linux, the user name is `ec2-user`.

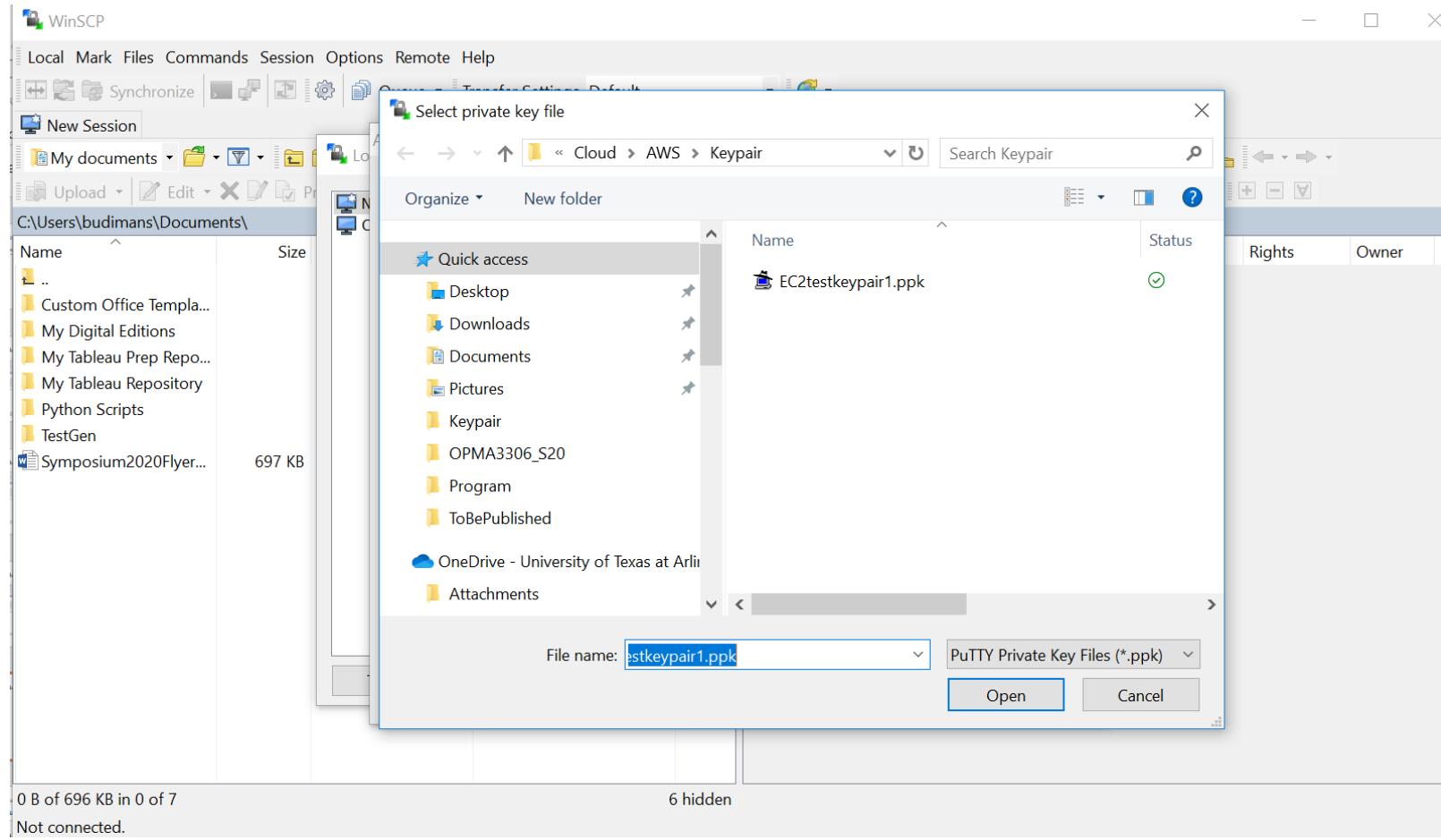
For RHEL 5, the user name is either `root` or `ec2-user`.

For Ubuntu, the user name is `ubuntu`

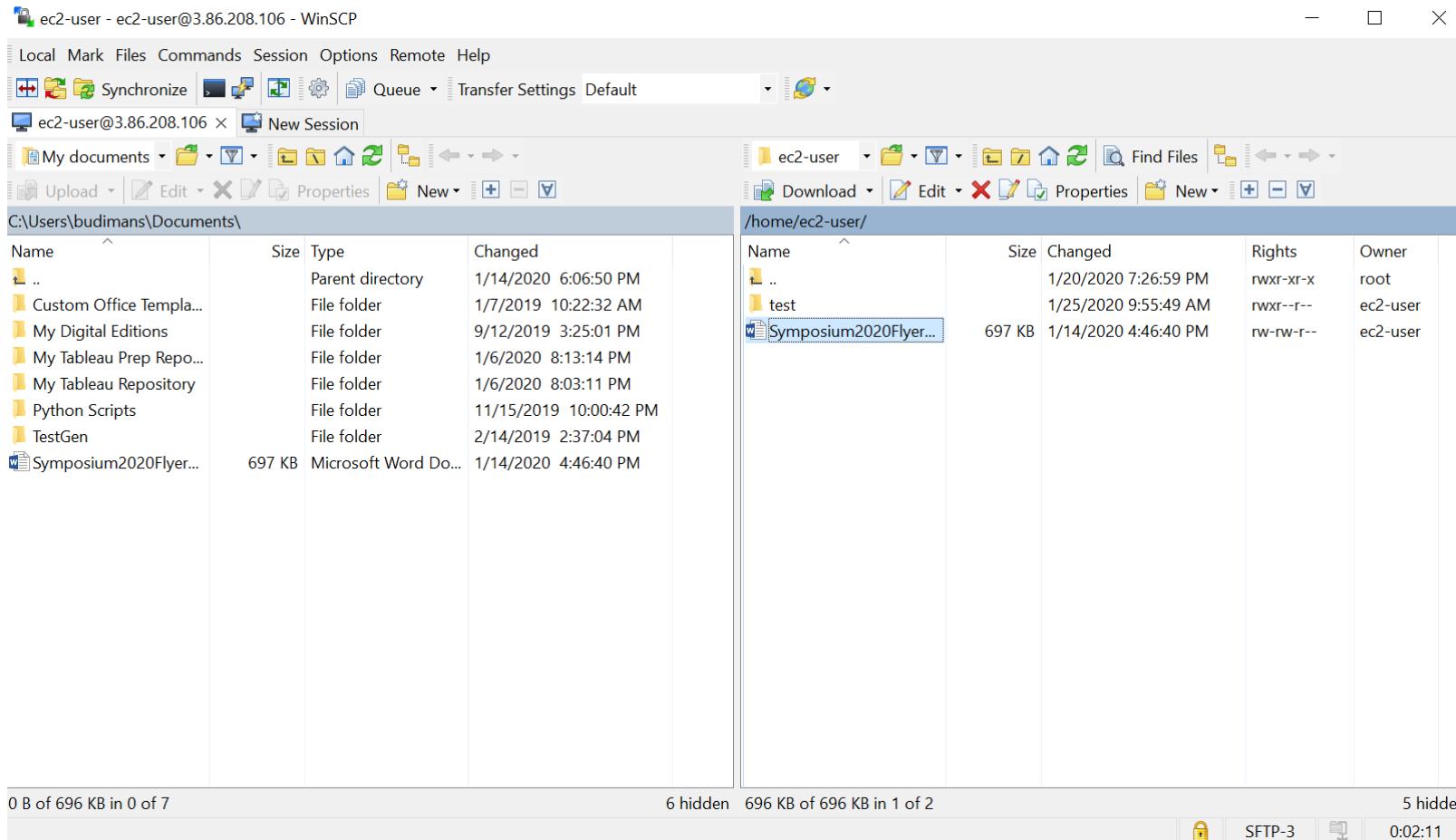
# File Transfer using WinSCP



# File Transfer using WinSCP



# File Transfer using WinSCP



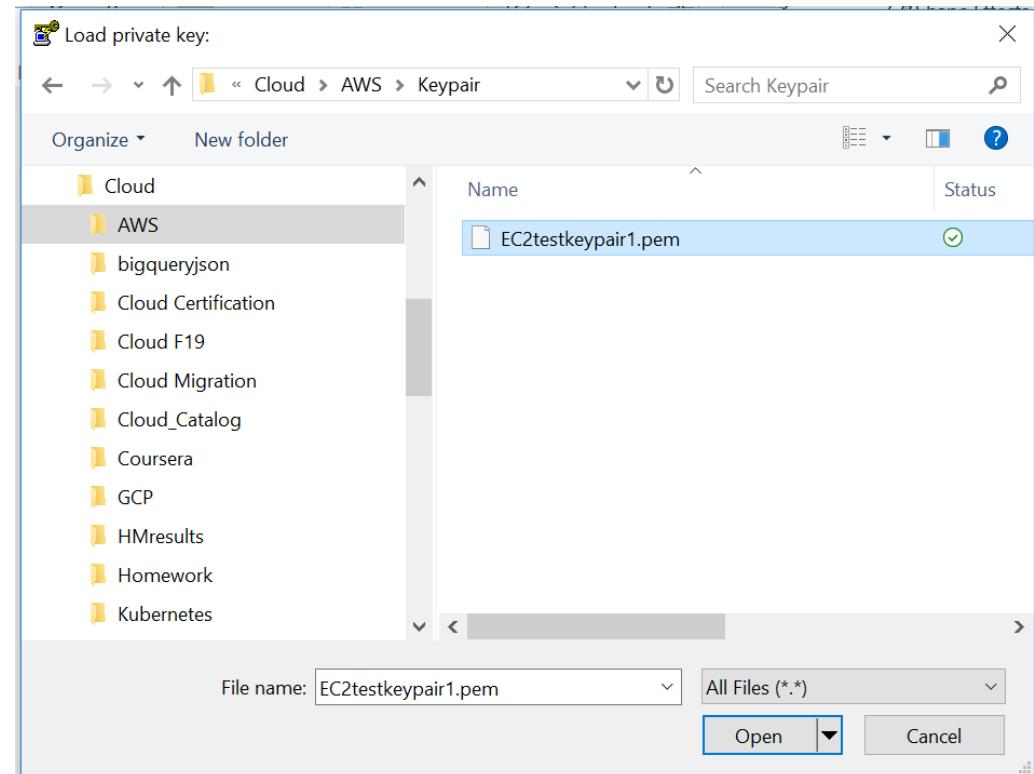
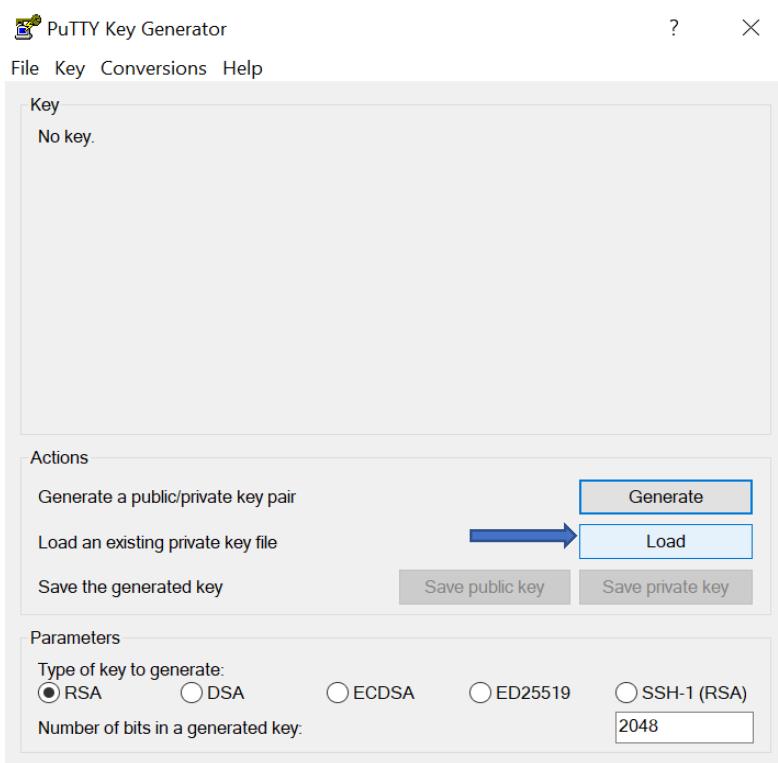
# SSH using Putty

# PuTTY Information



- PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform.
- PuTTY is an open-source software that is available with source code and is developed and supported by a group of volunteers.
- PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems.
- <https://www.putty.org/>
- [https://uta.service-now.com/selfservice2?id=utassp01\\_kb\\_article&sys\\_id=1cc0119ddbebf800ed7dbf0ce961911&pageid=utassp01\\_search](https://uta.service-now.com/selfservice2?id=utassp01_kb_article&sys_id=1cc0119ddbebf800ed7dbf0ce961911&pageid=utassp01_search)

# SSH - Convert PEM to PPK file (Key Pair)

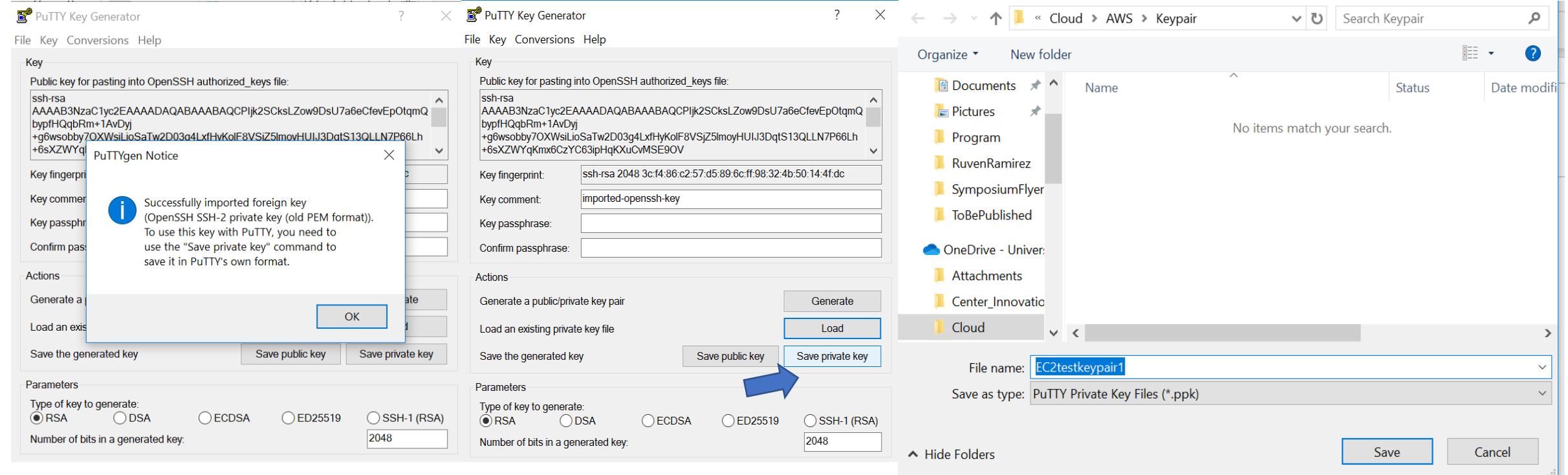


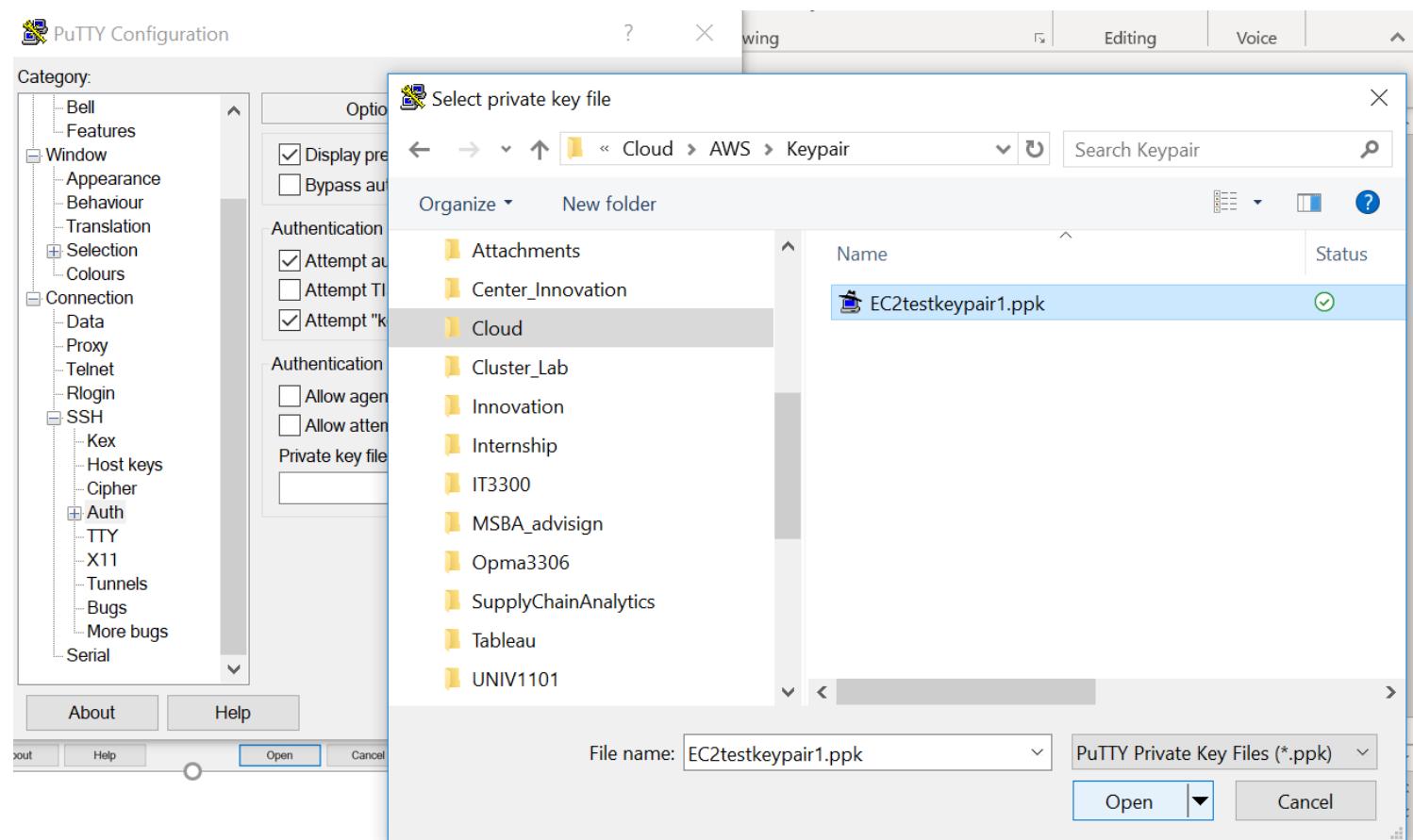
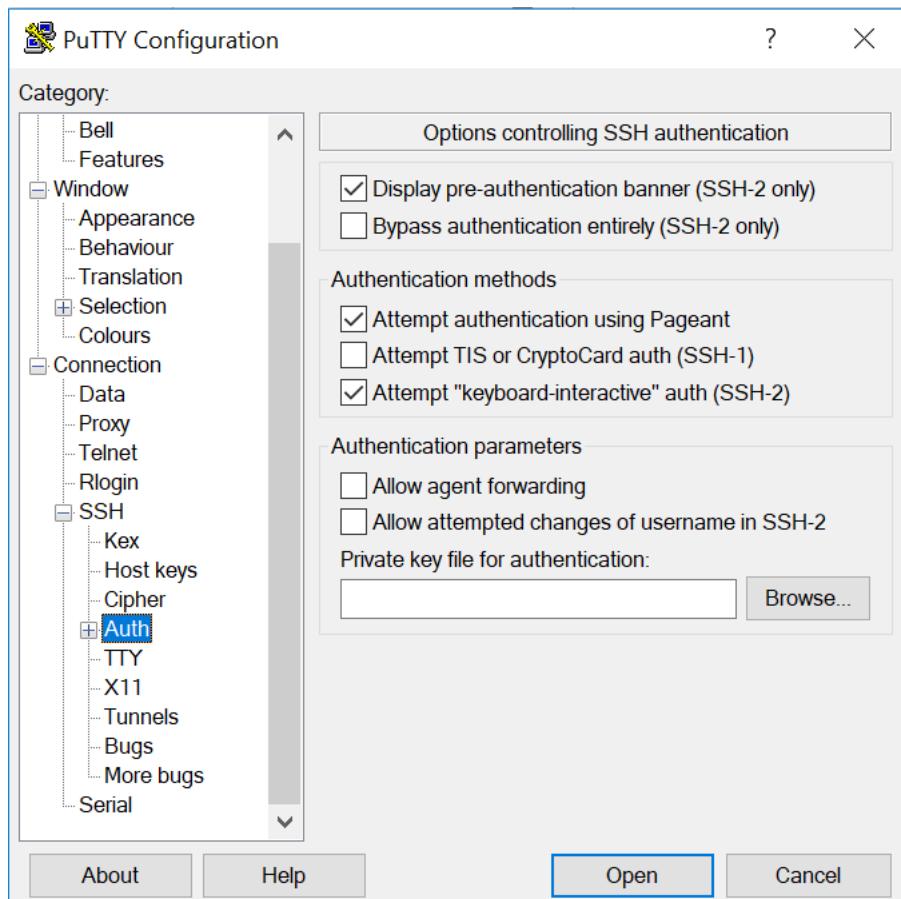
.pem file created in AWS must be saved in a directory  
The .pem created in AWS needs to be converted to .ppk

The SAME Key Pair can be used for all instances

# .pem vs .ppk

- PEM (Privacy Enhanced Mail) is base64 container format for encoding keys and certificates.
  - .pem download from AWS when you created your key-pair.
  - Only a one-time download
- PPK (Putty Private Key) is a windows ssh client
  - Does not support .pem format
  - .pem must be converted to .ppk format using PuTTyGen
- **Anybody** can know the public half (public key), whoever is **being authenticated** must hold the **private** half. In this case, your PC (the client) must have the private key since it is being authenticated by AWS (the server).
  - The server uses the **public key** to encrypt a message and send it to the client.
  - If the client has the correct **private key**, they can decrypt the message and send it back to the server for verification.





The image shows two windows related to SSH configuration and execution.

**PuTTY Configuration Window:**

- Category:** SSH
- Auth:** Private key file for authentication: C:\Users\budimans\OneDrive - University (Browse...)

**Terminal Window:**

- Title: ec2-54-88-59-3.compute-1.amazonaws.com - PuTTY
- Text: login as: ubuntu

**Caption:**

Amazon linux – login as: ec2-user  
Ubuntu –login as:ubuntu

# Note if using aws academy sandbox

Create security group     Select existing security group

Security group name - *required*

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .\_-:/()#@[]+=&;{}!\$\*

Description - *required* [Info](#)

- If EC2 creation failed due to SG, in addition to fixing the inbound/outbound rules, you may have to change the **Description (for whatever reason it doesn't like the same one)**.