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

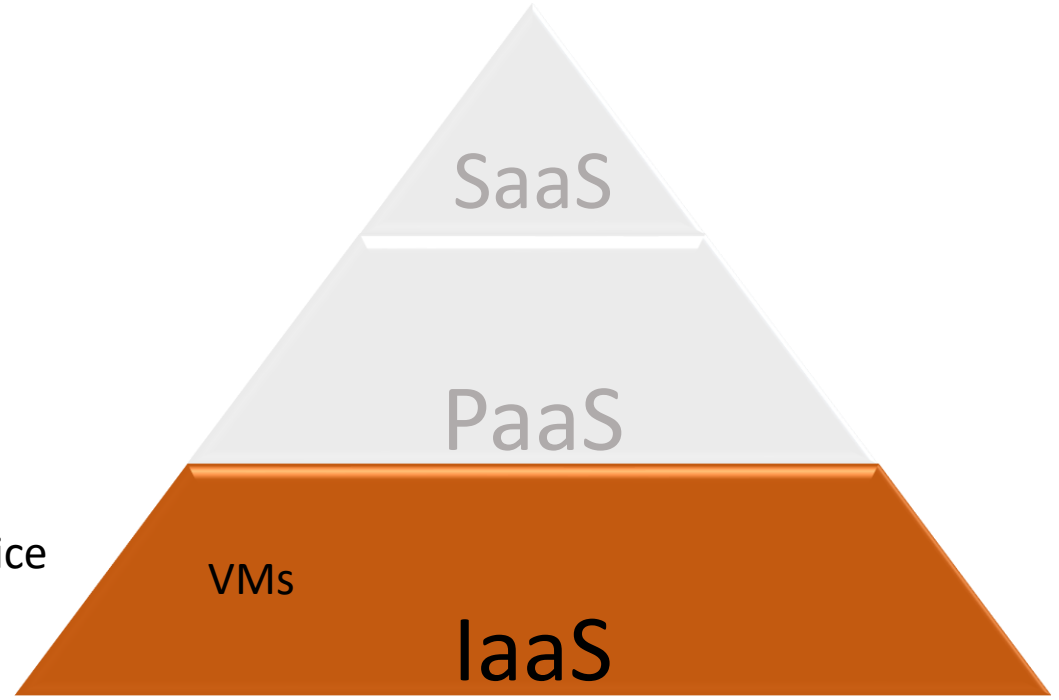
Elastic Compute Cloud (EC2)

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

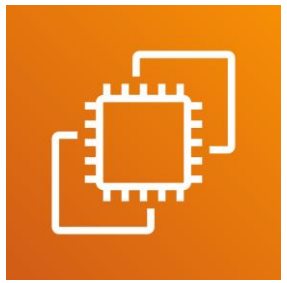
Faster application deployment

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)

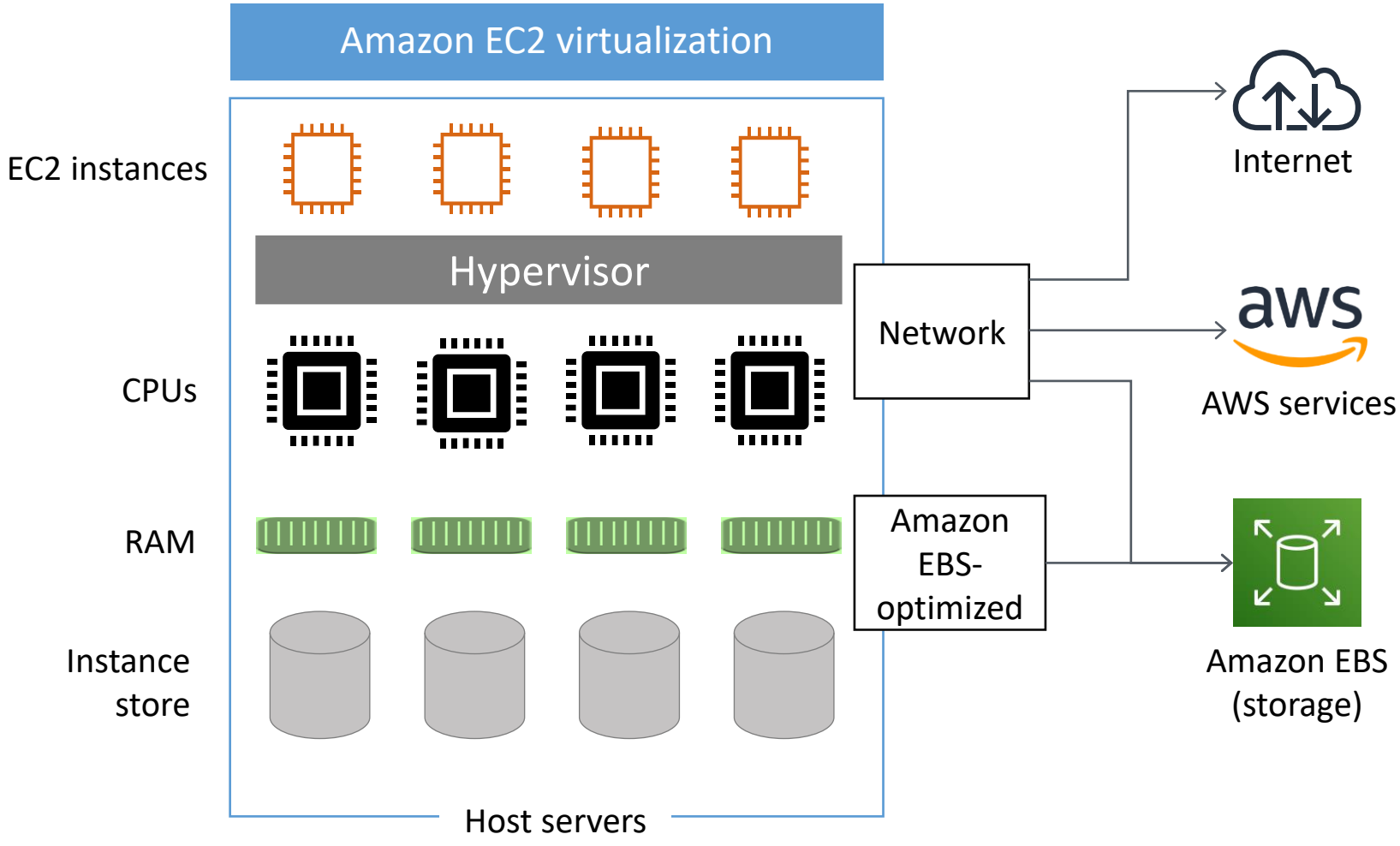
Amazon EC2 provides resizable compute capacity in the cloud.

- 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.

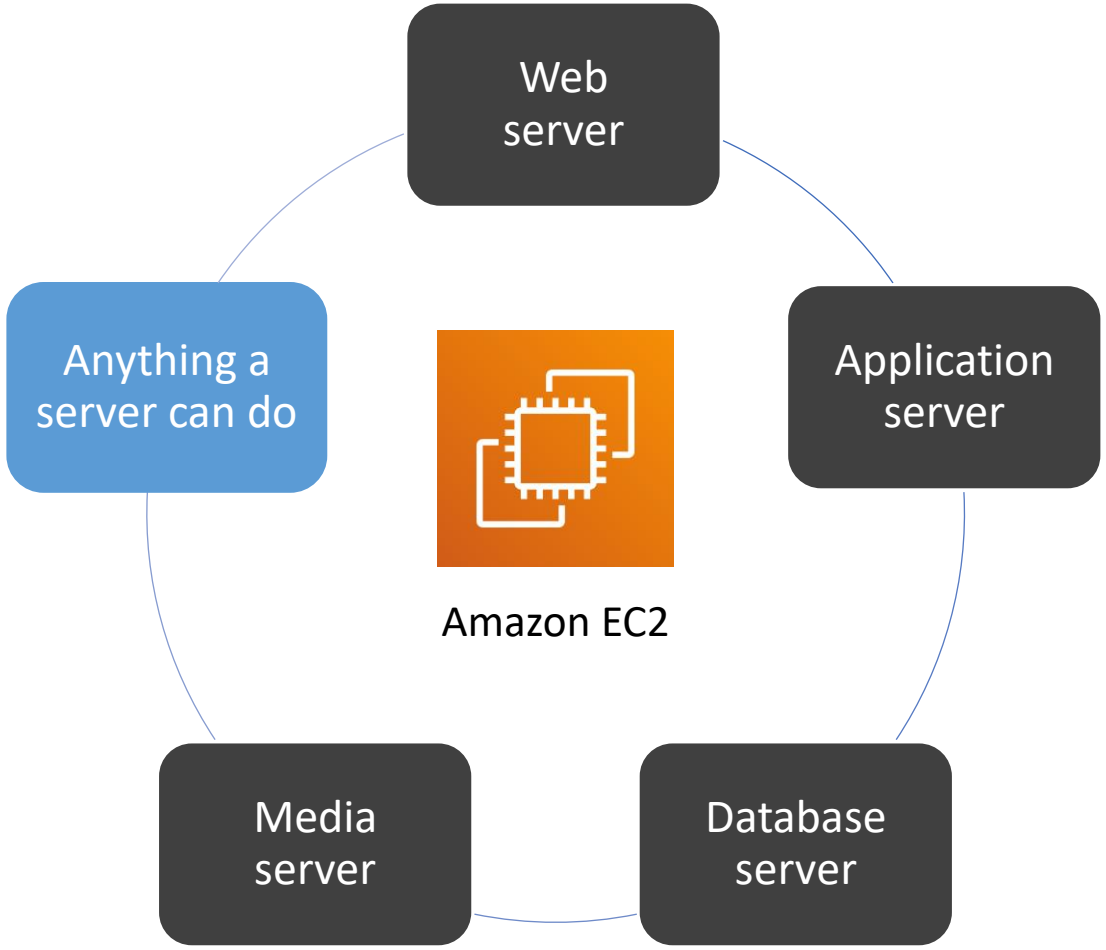
- You can choose different configurations of CPU and memory capacity
- Supports different storage options
  - Instance store
  - Amazon Elastic Block Store (Amazon EBS)
- Provides network connectivity



Amazon EC2 use cases

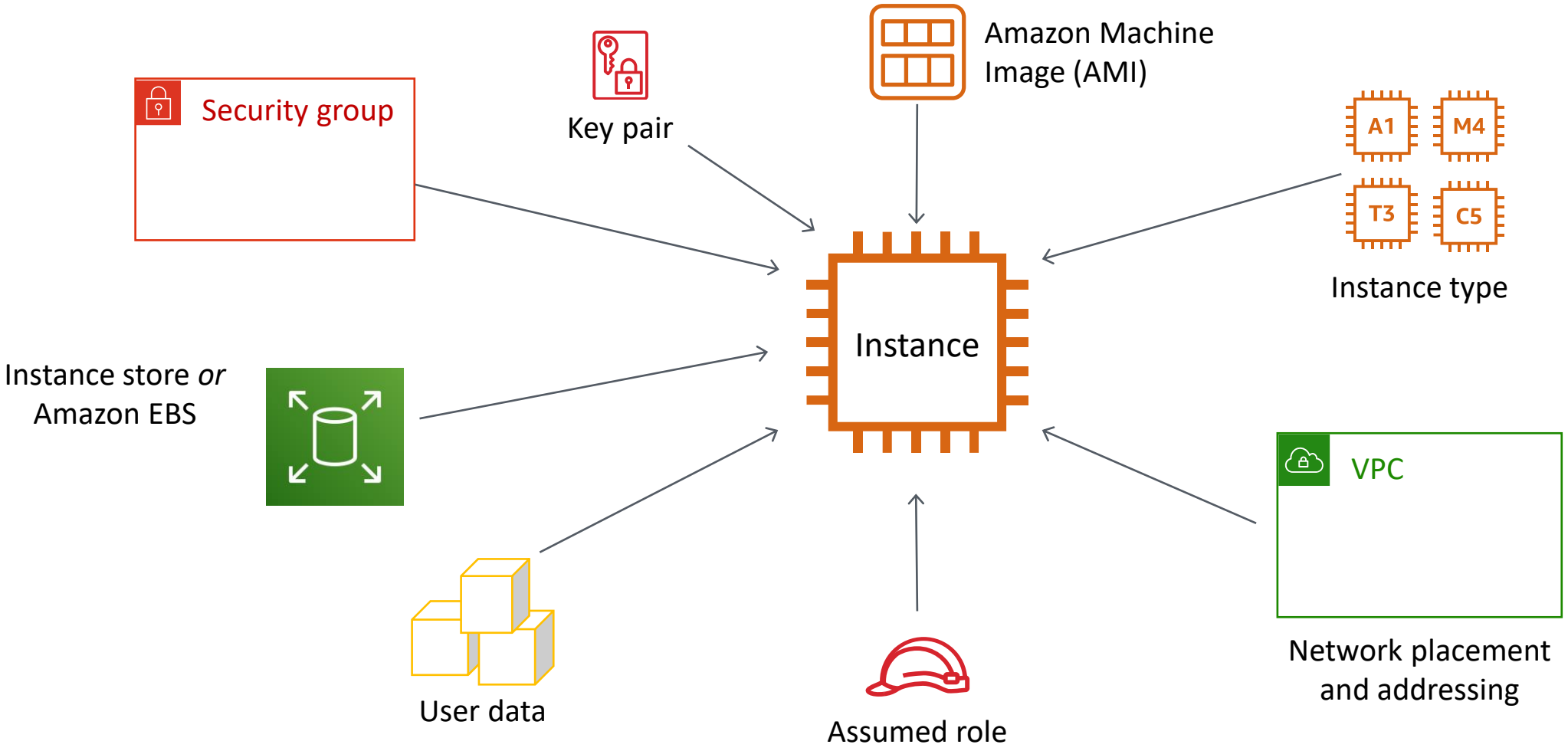
Use Amazon EC2 when you need:

- 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

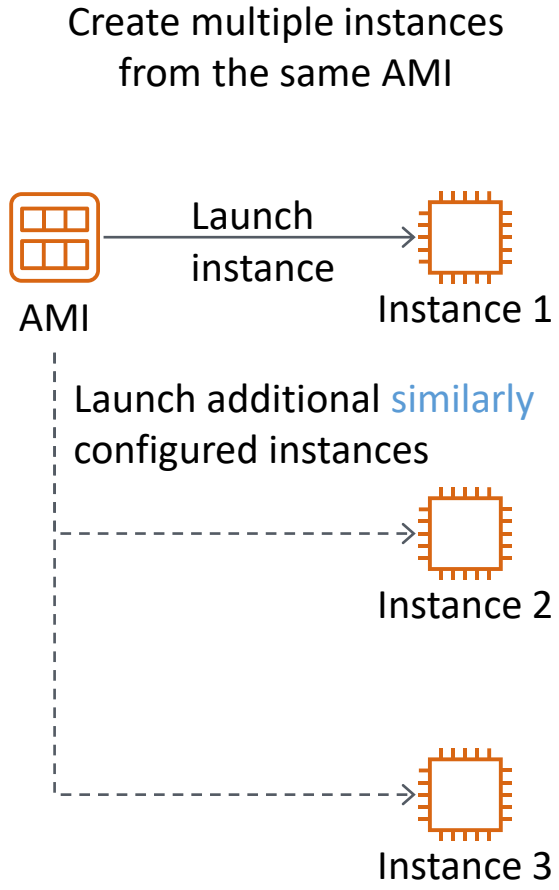


What is an AMI?

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



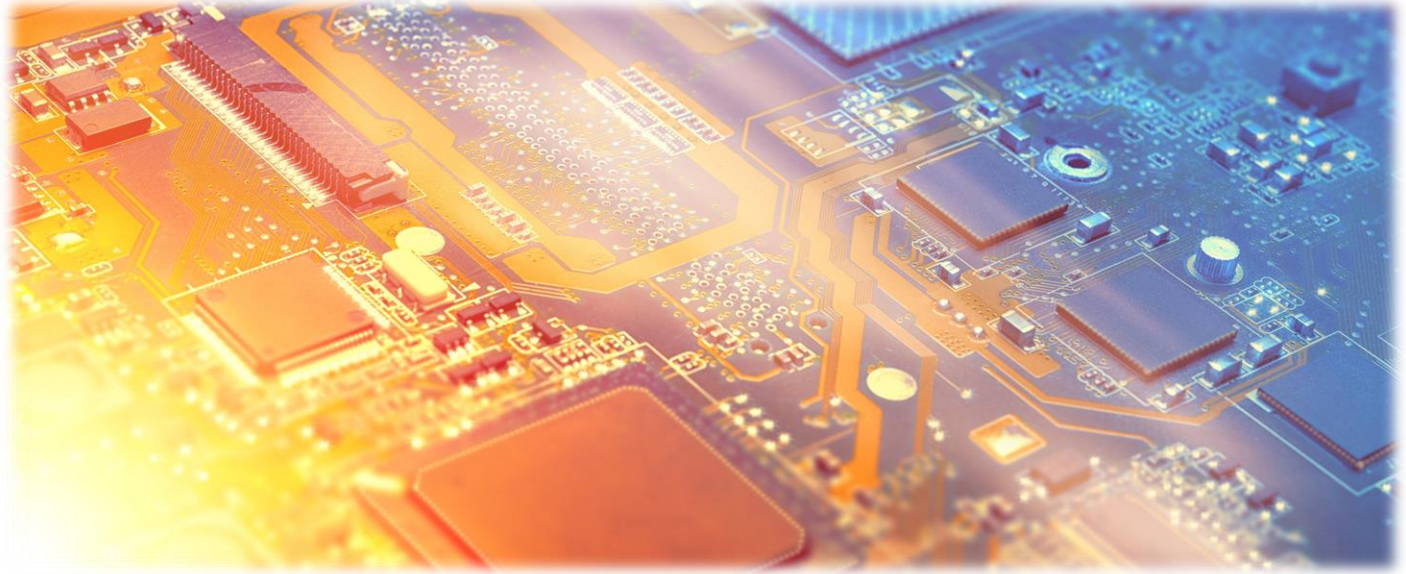
AMI benefits

- 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:

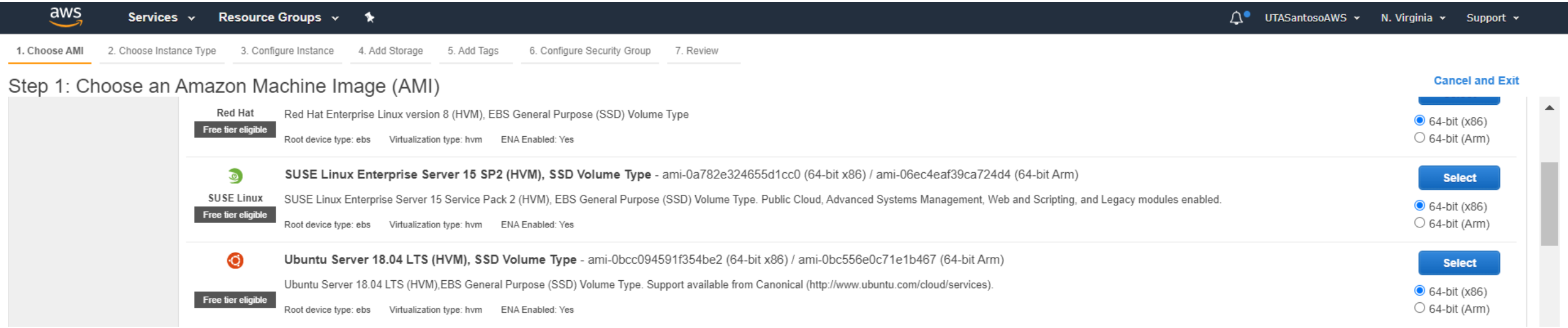
- Region
- Operating system
  - Microsoft Windows or Linux
- Storage type of the root device
- Architecture
- Virtualization type



AMI sources:

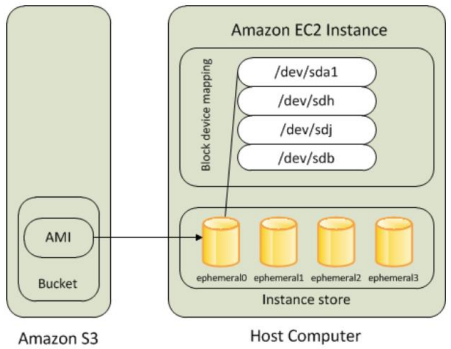
- Quick Start – *Linux and Microsoft Windows AMIs that are provided by AWS.*
- My AMIs – *Any AMIs that you create.*
- AWS Marketplace – *Pre-configured templates from third parties.*
- Community AMIs – *AMIs shared by others. Use at your own risk.*

Amazon AMI - example



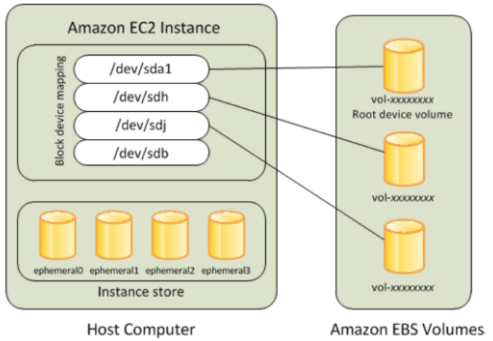
- Elastic Network Adapter (ENA) - next generation network interface
- Hardware Virtual Machine (HVM) –Type 1 Hypervisor
- X86/ARM – Computer Architecture/CPU
- EBS backed

Instance Root Volume



instance store-backed AMI

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



EBS-backed AMI

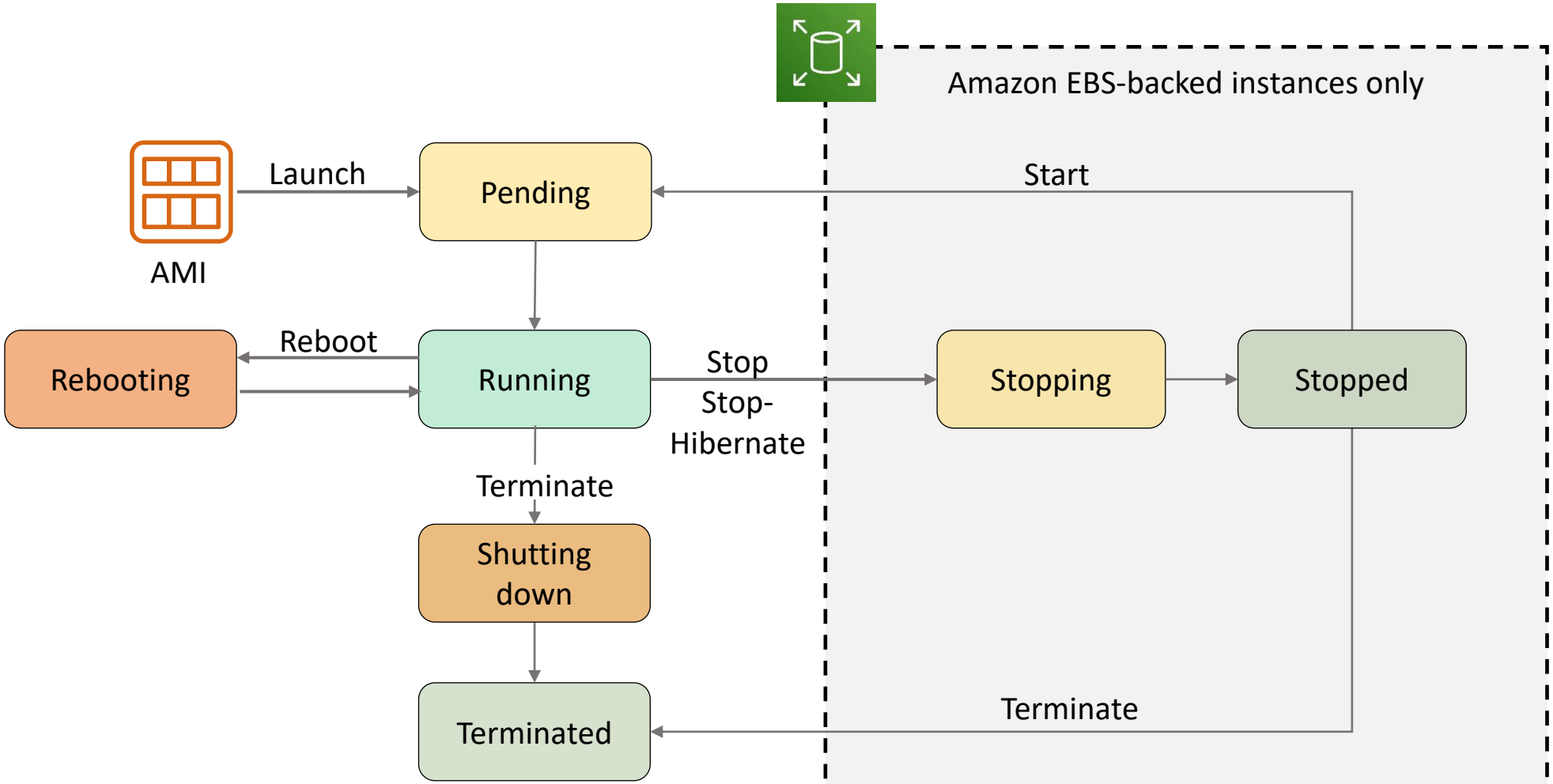
When an instance is launched, the *root device volume* contains the image used to boot the instance.

- ❑ 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

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

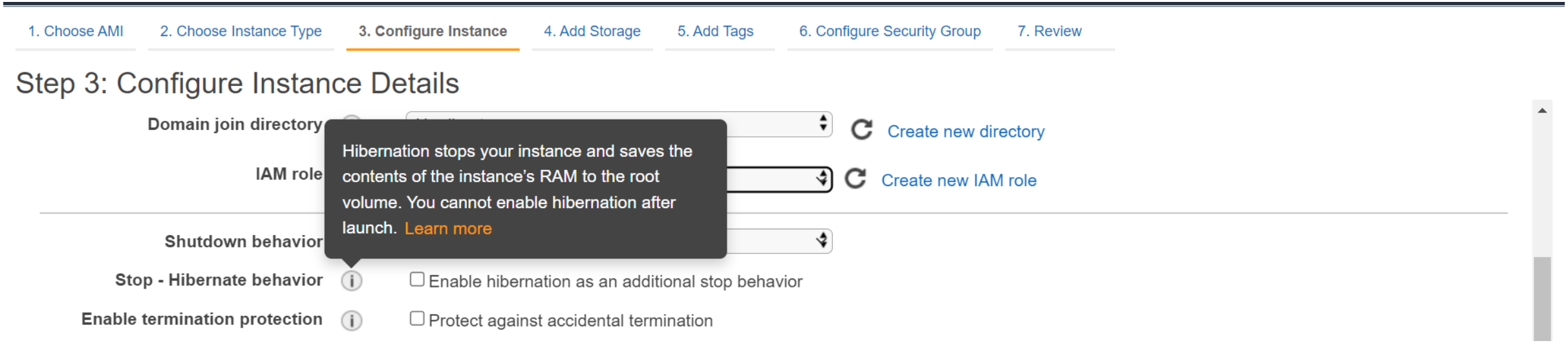
Amazon EC2 instance lifecycle



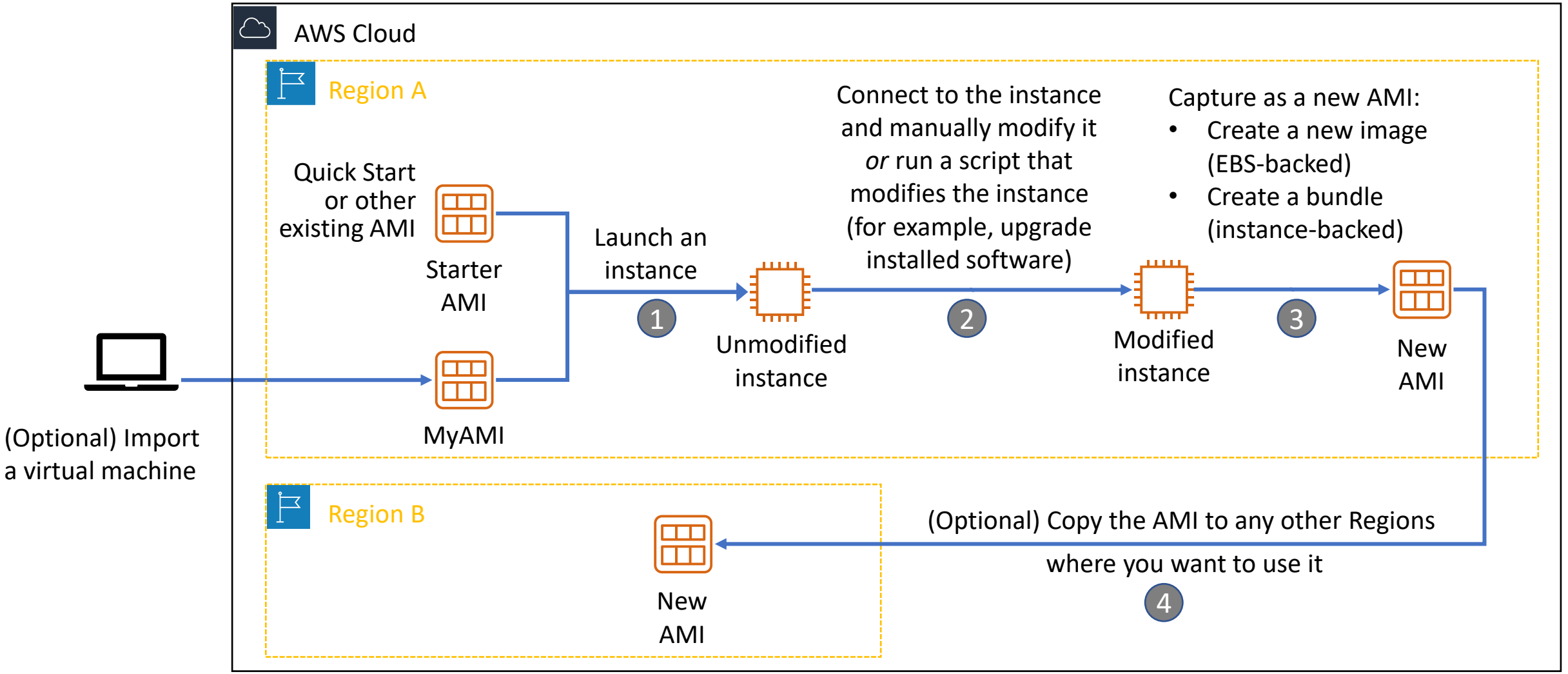
Stop vs Stop-Hibernate

- 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



Creating a new AMI

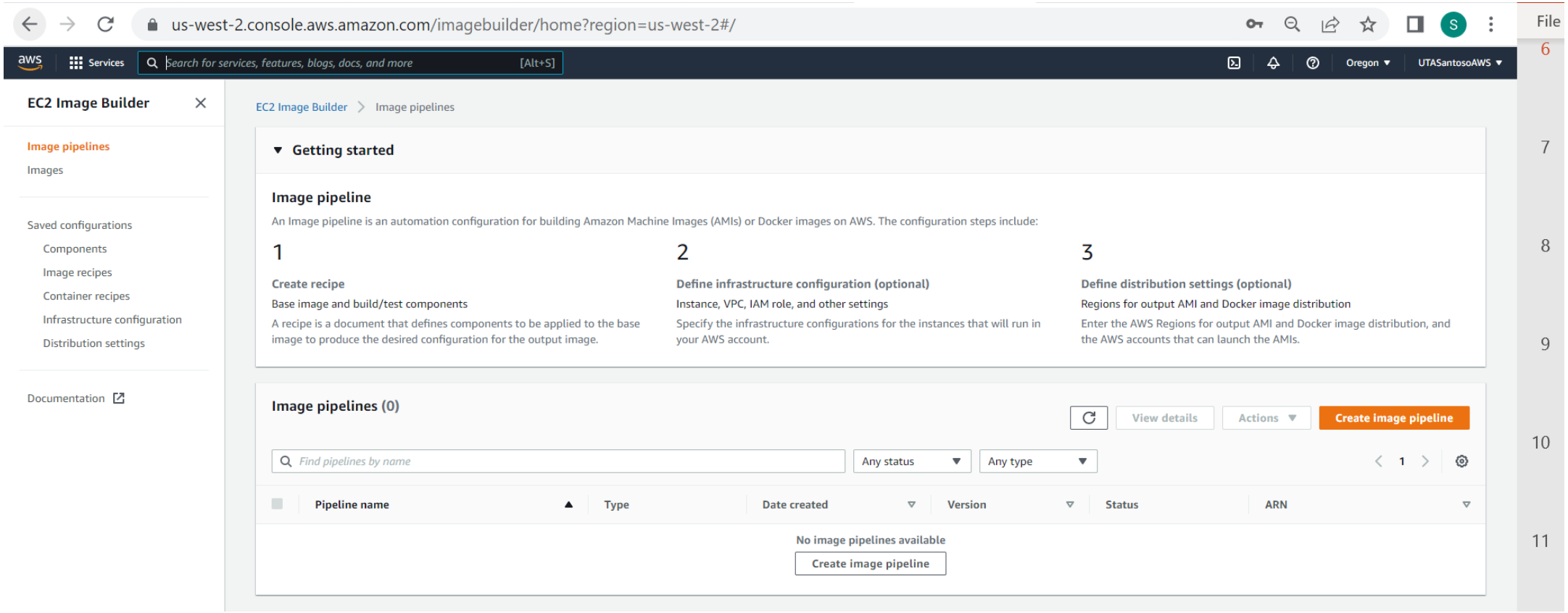


EC2 Image Builder

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

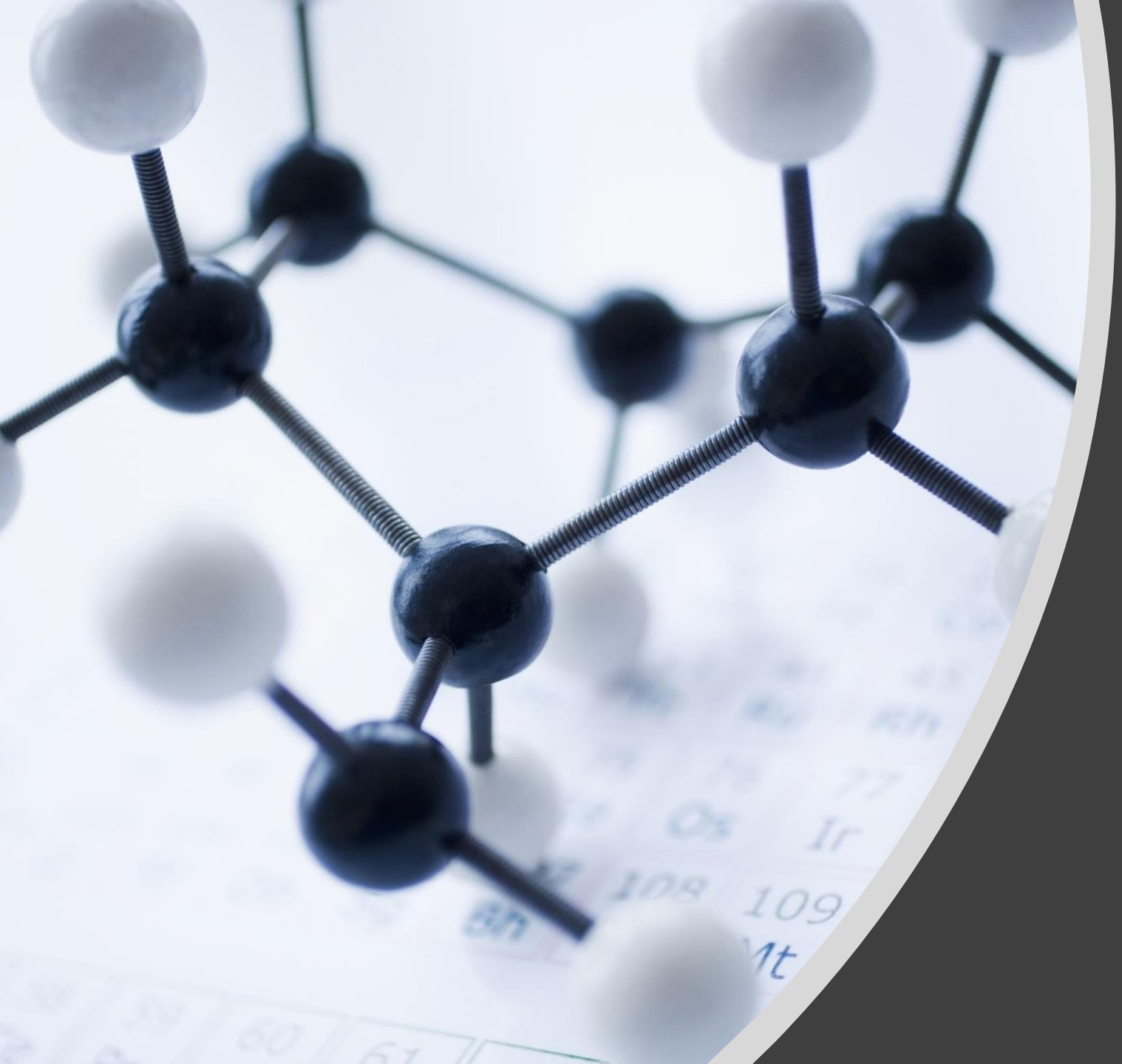
- Provides a graphical interface to create image-building pipelines
- Creates and maintains Amazon EC2 AMIs and on-premises VM images
- Produces secure, validated, and up-to-date images
- Enforces version control

EC2 Image Builder



AMI key takeaways

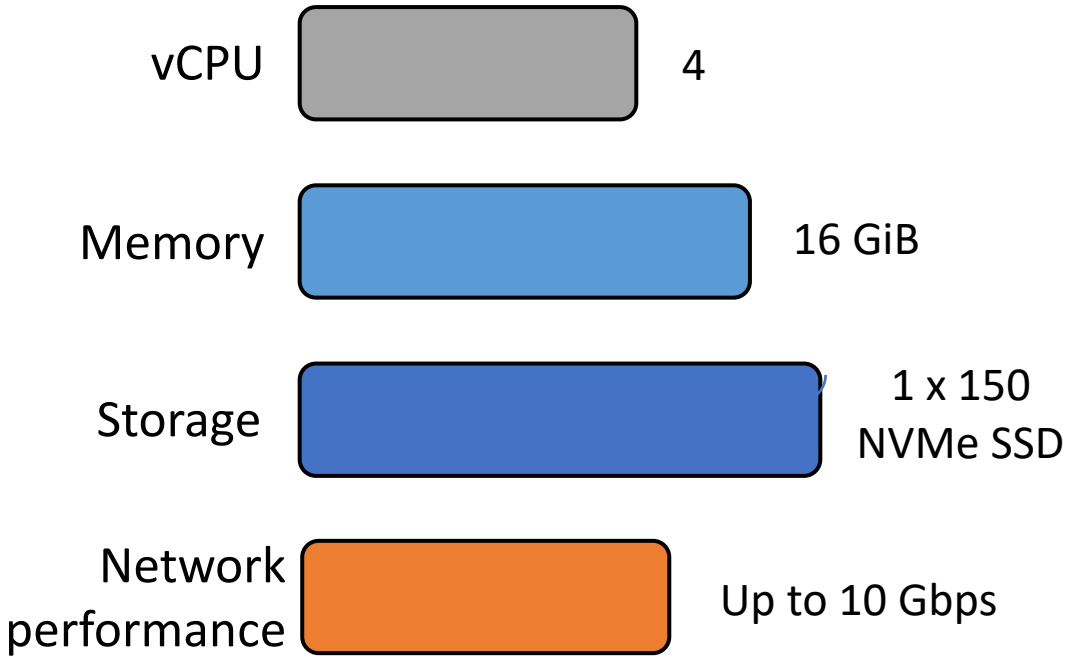
- An AMI provides the information that is needed to launch an EC2 instance
- For best performance, use an AMI with HVM virtualization type
- Only an instance launched from an Amazon EBS-backed AMI can be stopped and started
- An AMI is available in a Region

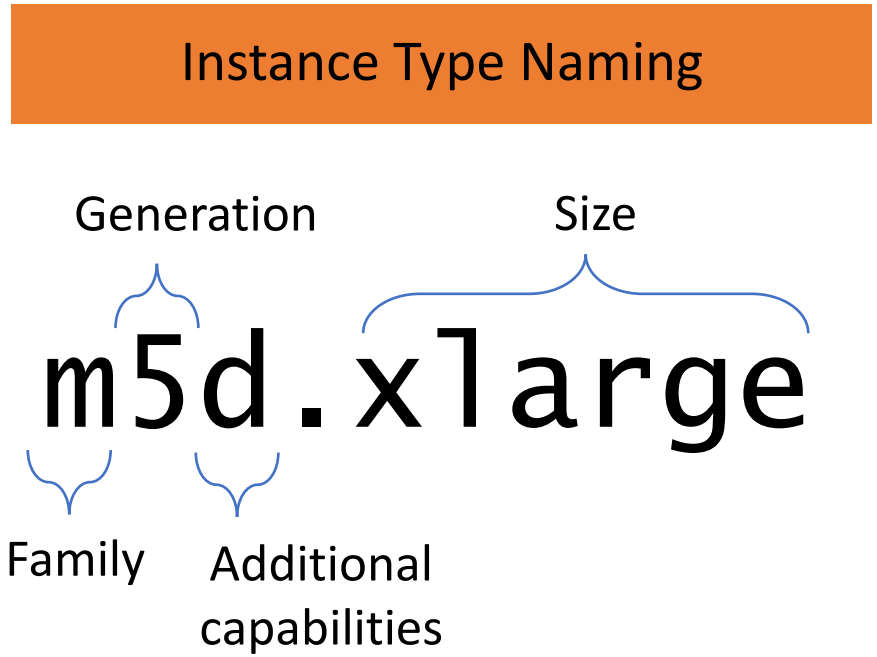


Selecting an EC2 instance type

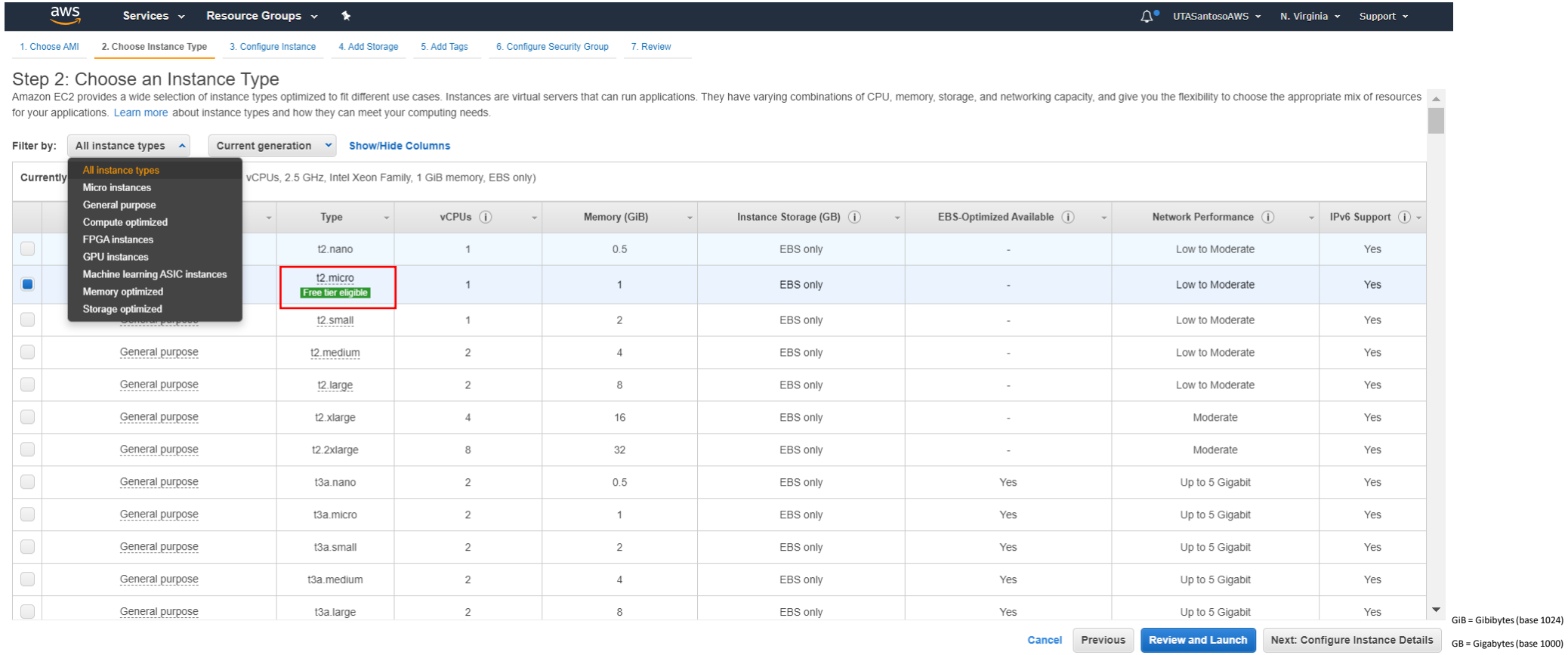
EC2 instance type

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





AWS Instance types



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)

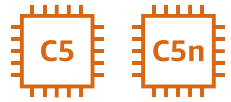
General purpose instance types

- 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: R5 X1 HMI

Accelerated computing instance types

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

Example instance types: P3 G4 F1

Storage optimized instance types

- High-performance databases[1]
- Real-time analytics[1]
- Transactional workloads[1]
- NoSQL databases[1]
- Big data[2]
- Data warehouse[2]
- Log processing[2]

[1]High I/O example instance type: I3

[2]Dense Storage example instance types: D2 H1

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.

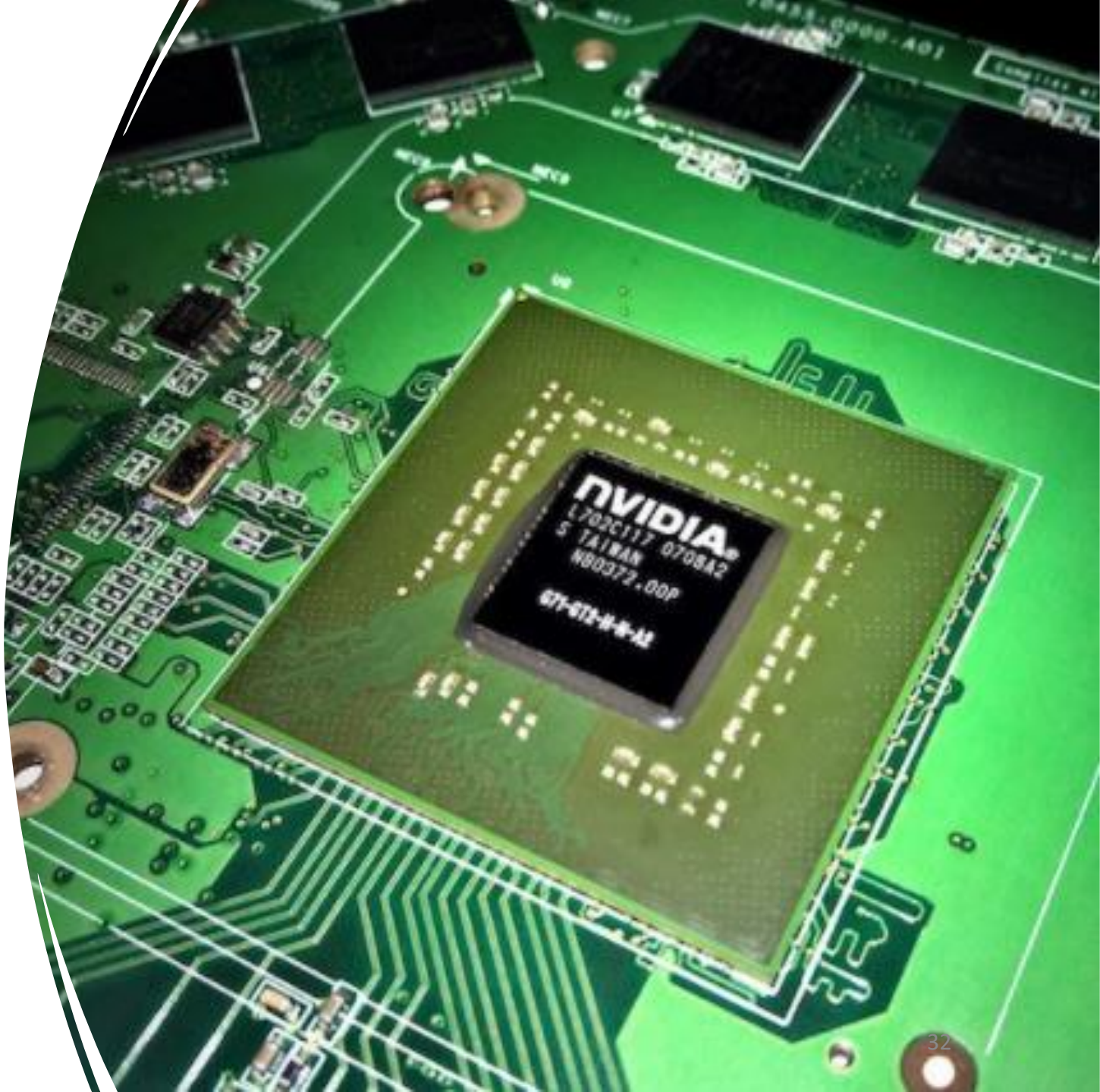
In some cases, **GPU** is 4-5 times **faster than CPU**

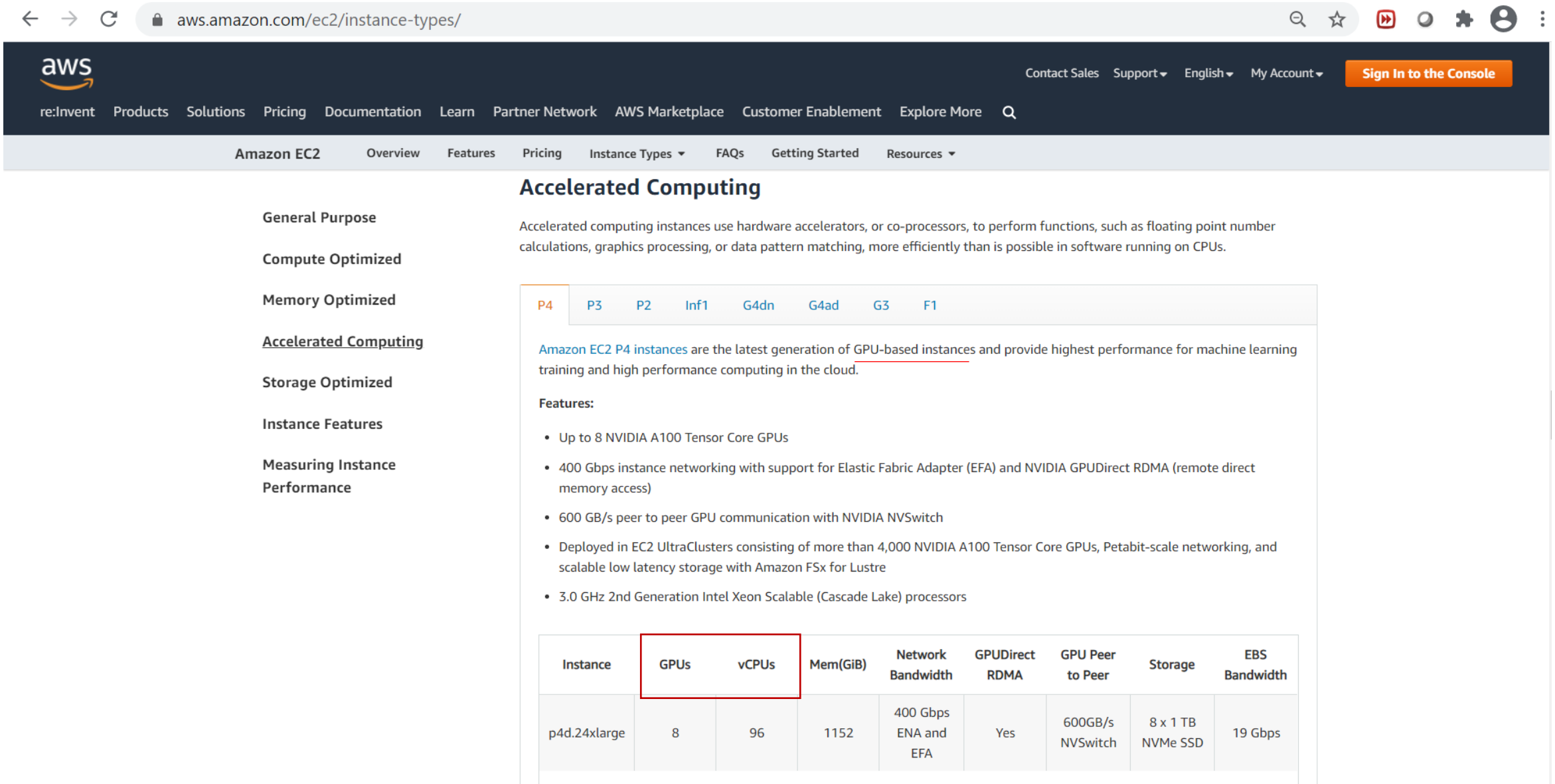
A GPU typically has more *logical* cores (arithmetic logic units or ALUs, control units and memory cache) than a CPU.

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
Amazon EC2
Overview
Features
Pricing
Instance Types
FAQs
Getting Started
Resources
General Purpose
Compute Optimized
Memory Optimized
Accelerated Computing
Storage Optimized
Instance Features
Measuring Instance Performance
Accelerated Computing
Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.
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

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

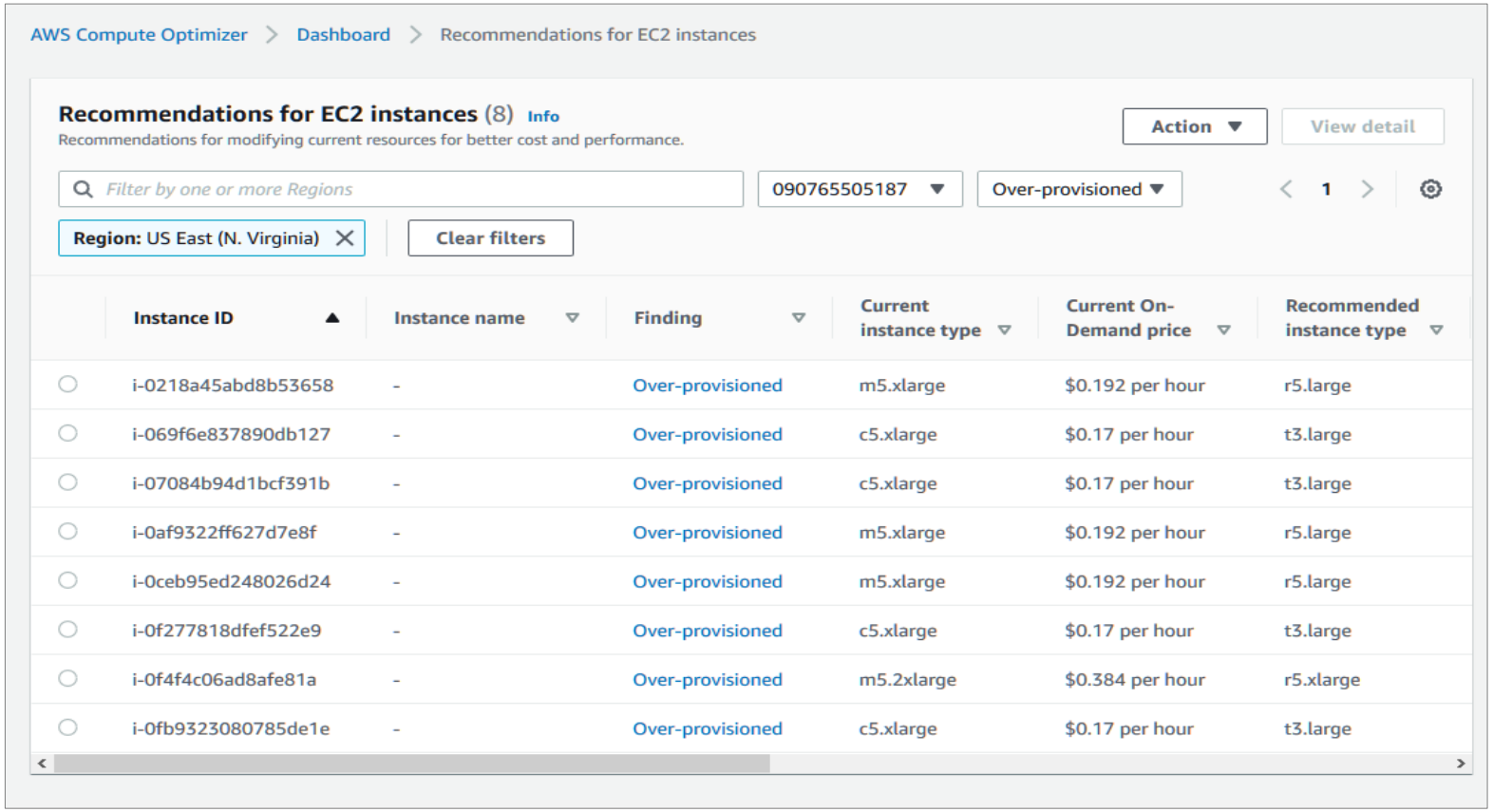
- If you have an *already existing* instance –
  - You can get recommendations for optimizing the instance type by using the AWS Compute Optimizer
  - You can evaluate recommendations and modify the instance accordingly

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



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.

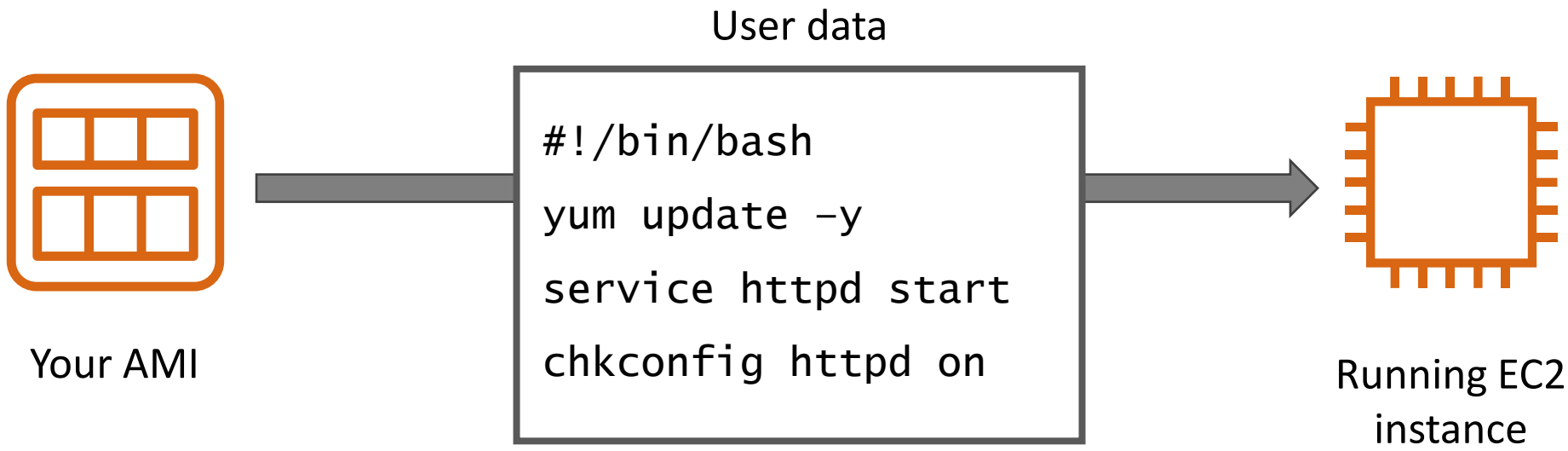
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html

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).



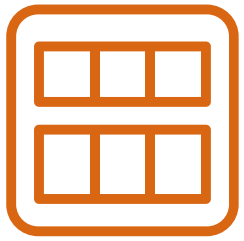
Retrieving instance metadata

Instance metadata is information about your instance.

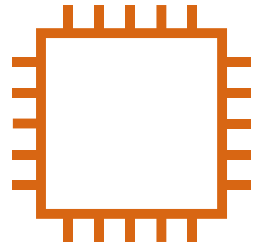
- Is accessible from your instance at URL: `http://169.254.169.254/latest/meta-data/`
- Can be retrieved from a user data script

User data

```
#!/bin/bash
yum update –y
hostname = $(curl -s http://169.254.169.254/latest/meta-data/public-hostname)
```



Your AMI



Running EC2 instance

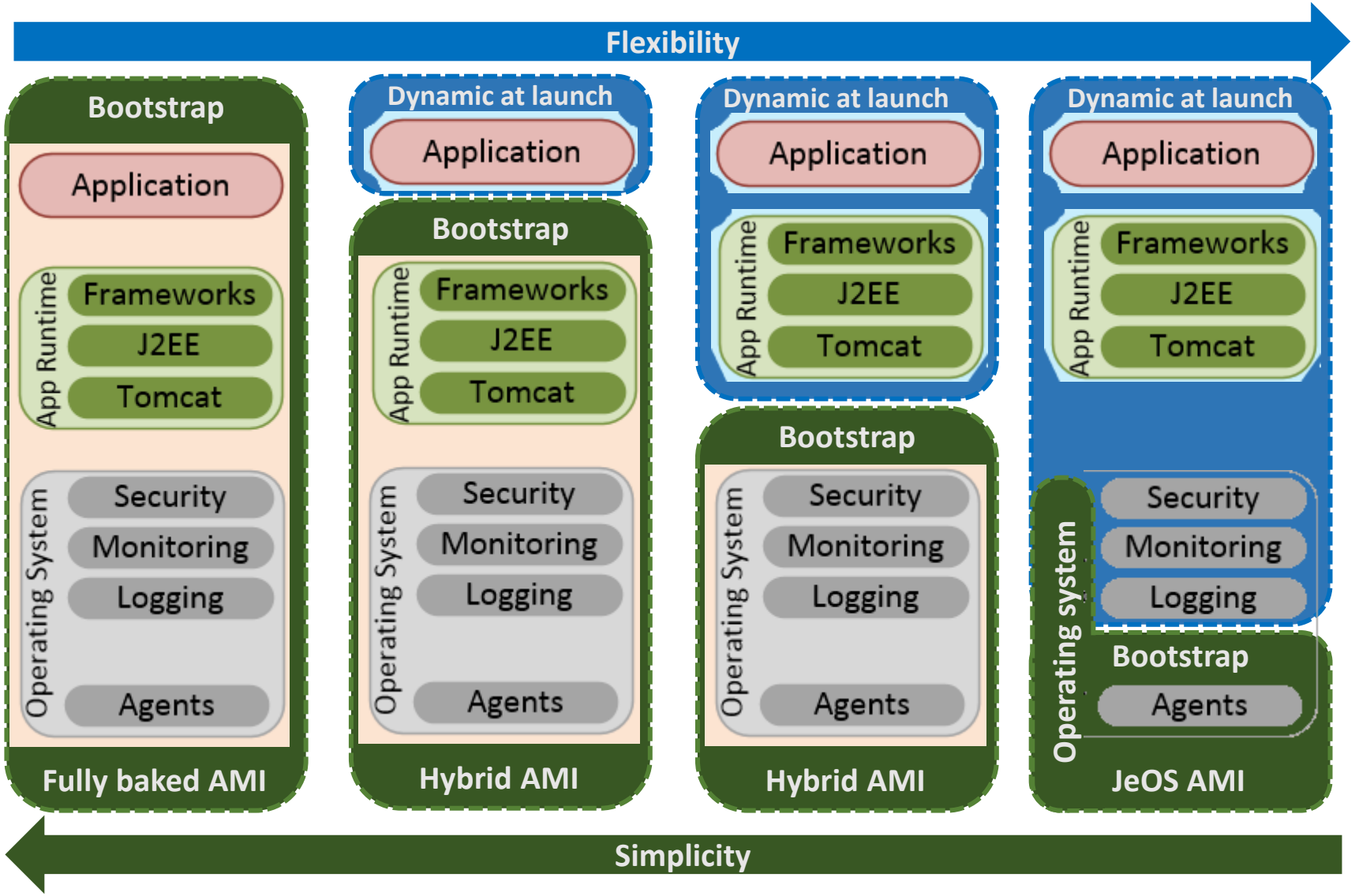
Metadata	Value
instance-id	i-1234567890abcdef0
mac	00-1B-63-84-45-E6
public-hostname	ec2-203-0-113-25.compute-1.amazonaws.com
public-ipv4	67.202.51.223
local-ipv4	10.251.50.12

3 ways to install application

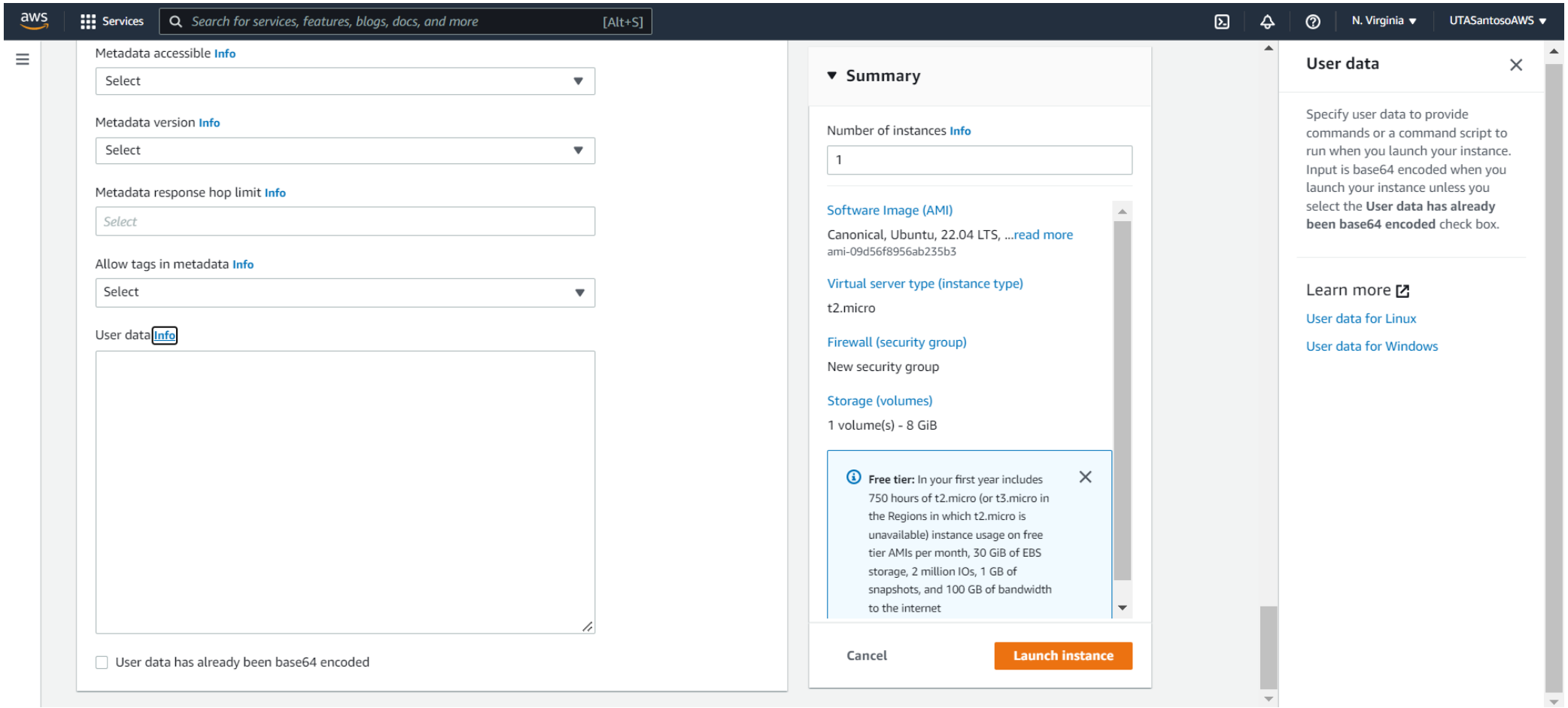
Let’s say you want to install an application in an EC2, you have 3 ways to do it:

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”

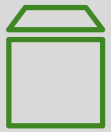


Adding storage to an Amazon EC2 instance

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

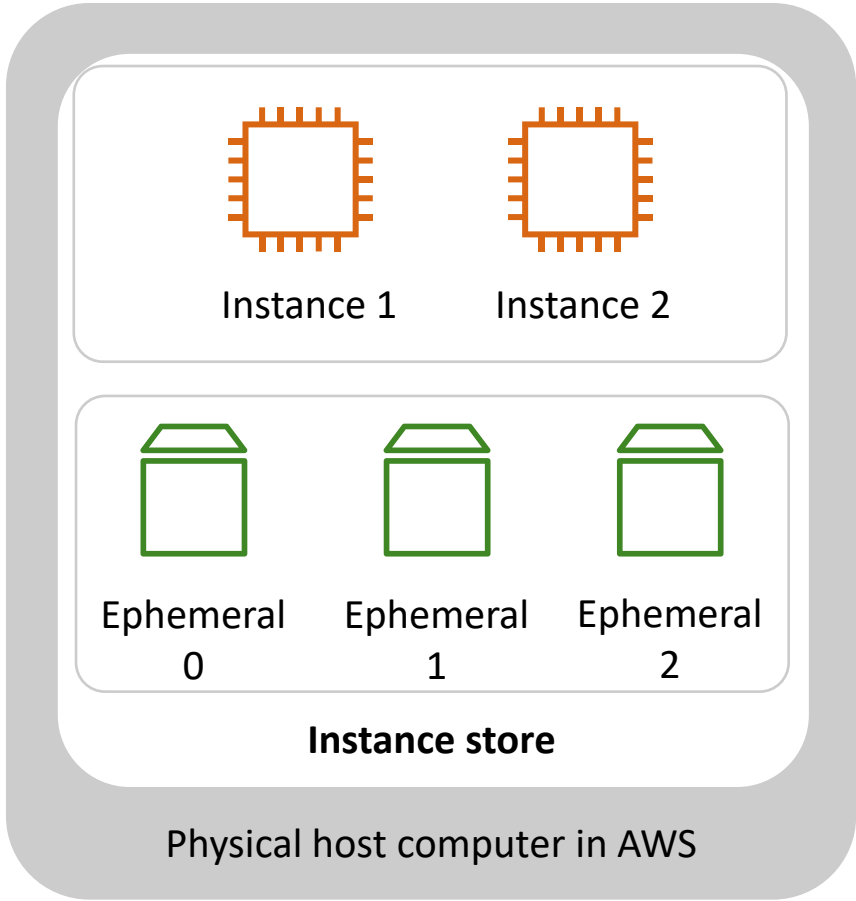
For data accessible from multiple instances

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

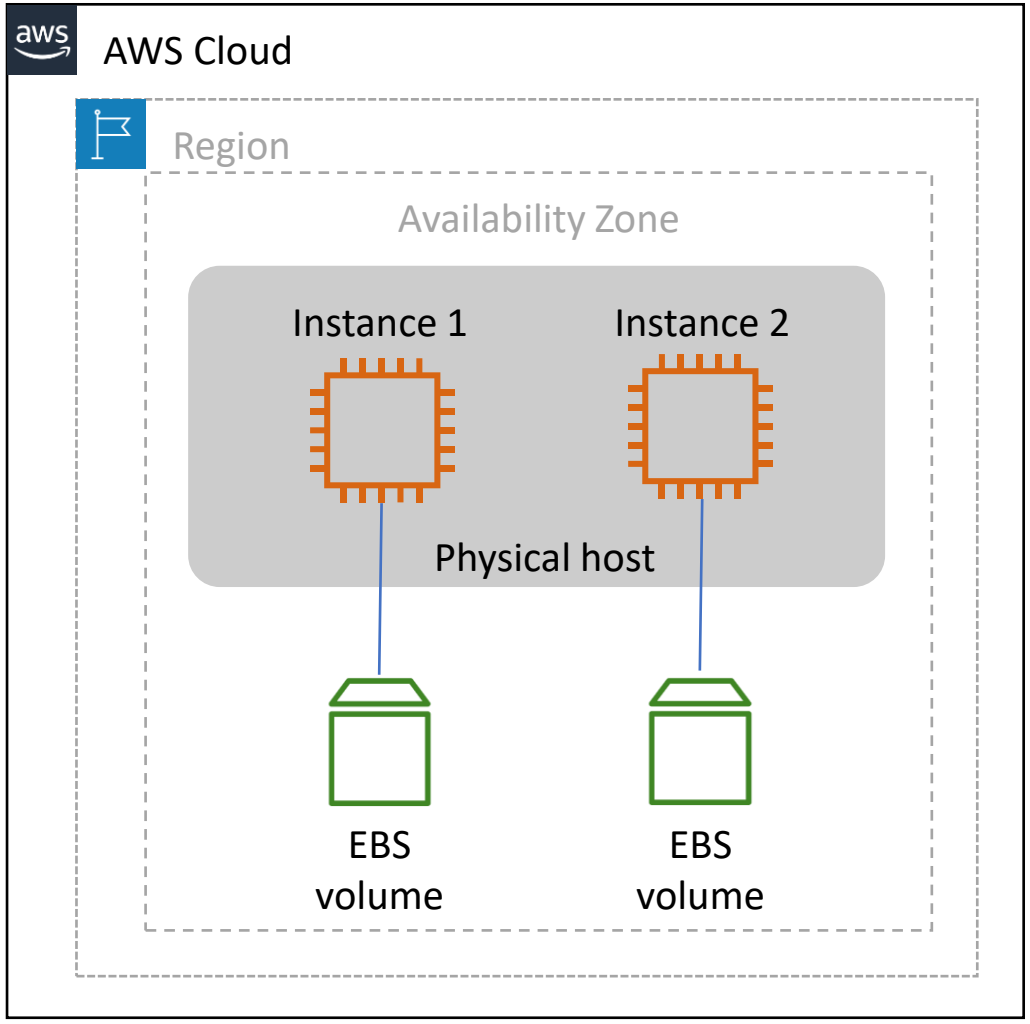


Amazon EBS

- 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	• Highest-performance SSD volume • Good for mission-critical, low-latency, or high-throughput workloads
Use Cases	• Recommended for most workloads • Can be a boot volume	• Critical business applications that require sustained IOPS performance • Large database workloads • Transactional workloads • It can be a boot volume

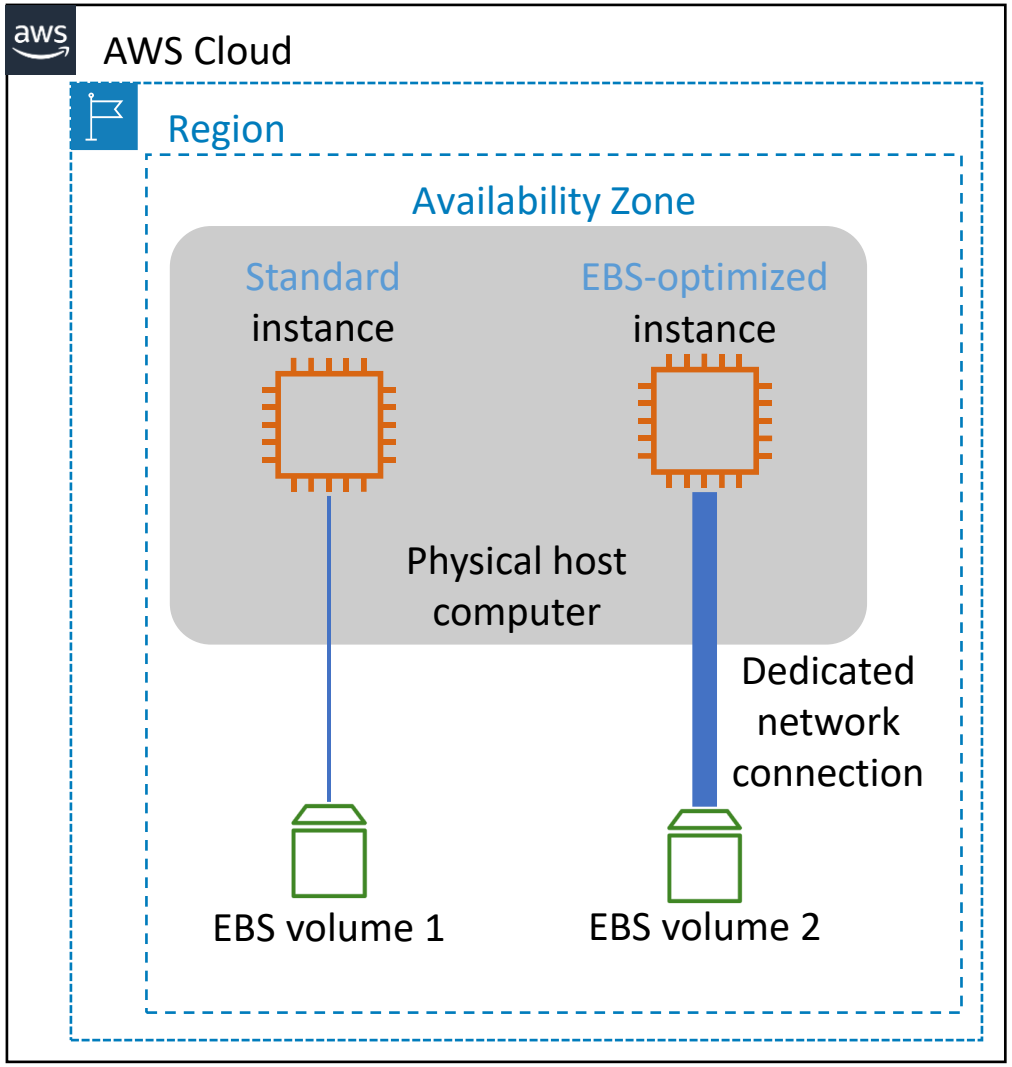
Amazon EBS HDD-backed volume types

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

	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	• Low-cost volume type • Designed for frequently accessed, throughput-intensive workloads	• Lowest-cost HDD volume • Designed for less frequently accessed workloads
Use Cases	• Streaming workloads • Big data • Data warehouses • Log processing • It cannot be a boot volume	• Throughput-oriented storage for large volumes of infrequently accessed data • Use cases where the lowest storage cost is important • It cannot be a boot volume

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 instances 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)

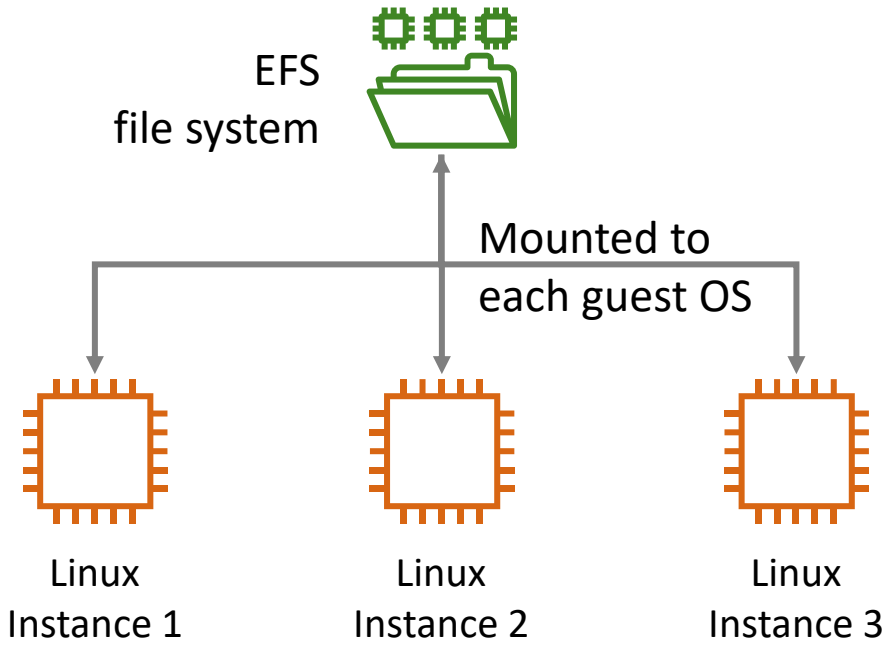
Amazon EFS

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

Amazon FSx for Windows File Server

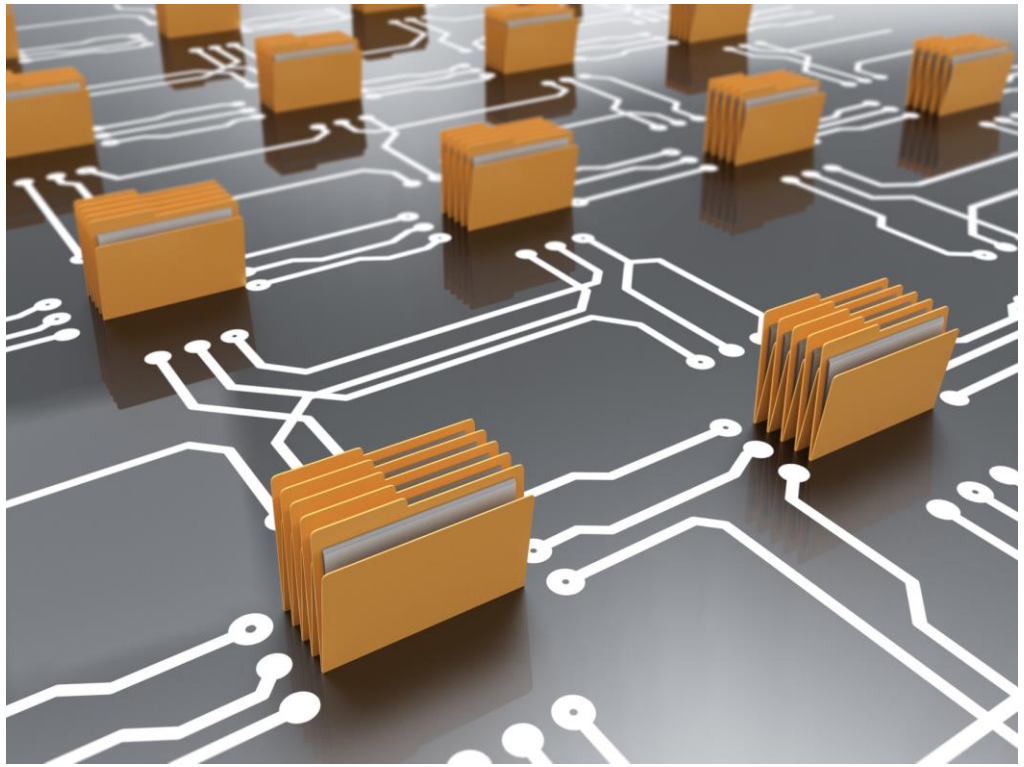
Provides fully managed shared file system storage for Microsoft Windows EC2 instances.

- 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 use cases

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

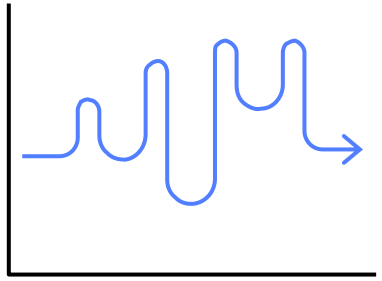
- Storage options for EC2 instances include instance store, Amazon EBS, Amazon EFS, and Amazon FSx for Windows File Server
- For a root volume, use instance store or SSD-backed Amazon EBS
- For a data volume that serves only one instance, use instance store or Amazon EBS storage
- For a data volume that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

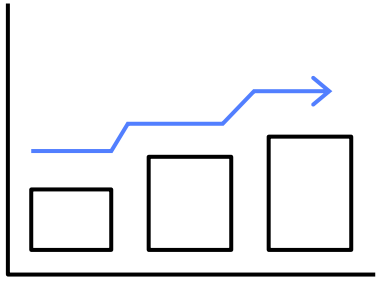
Pay for compute capacity by the second or by the hour with no long-term commitments.



Spiky workloads, workload experimentation

Reserved Instances

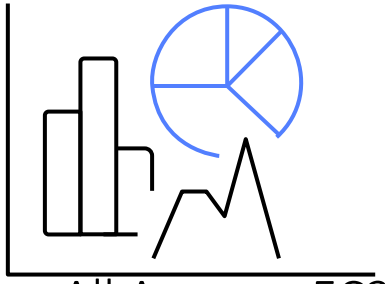
Make a 1-year or 3-year commitment and receive a significant discount off on-demand prices.



Committed and steady-state workloads

Savings Plans

Same discounts as Reserved Instances with more flexibility.



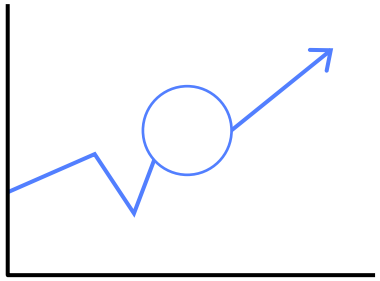
All Amazon EC2, AWS Fargate, and AWS Lambda workloads

Amazon EC2 pricing options (2 of 2)

Spot Instances

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

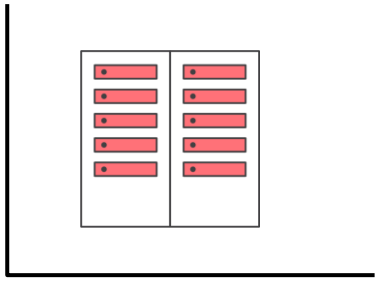
bid on unused EC2 instances



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

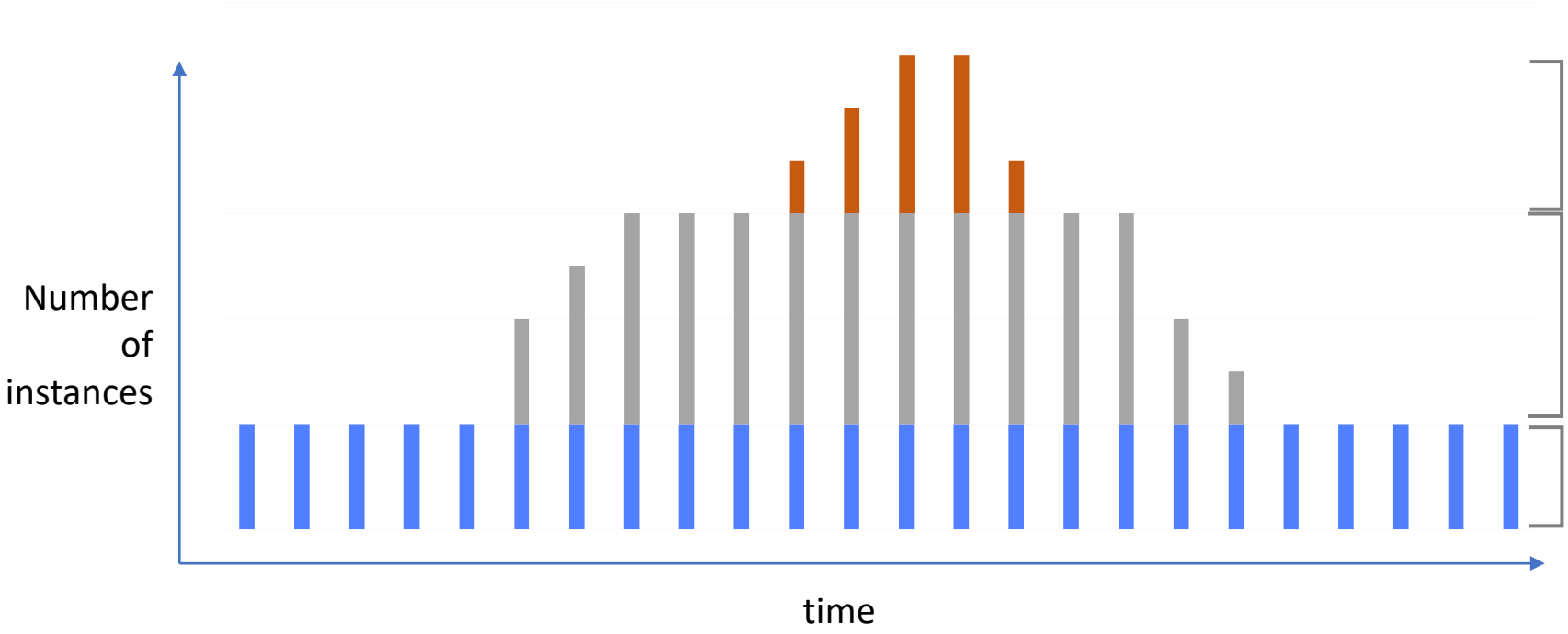
Dedicated Hosts

- 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

https://aws.amazon.com/ec2/dedicated-hosts/#:~:text=An%20important%20difference%20between%20a,same%20physical%20server%20over%20time.

Amazon EC2 cost optimization guideline

To optimize the cost of Amazon EC2 instances, combine the available purchase options.



Scale using Spot Instances for fault-tolerant, flexible, stateless workloads

Use On-Demand Instances for new or stateful spiky workloads

Use Reserved Instances or Savings Plans for known, steady-state workloads

Amazon EC2 considerations

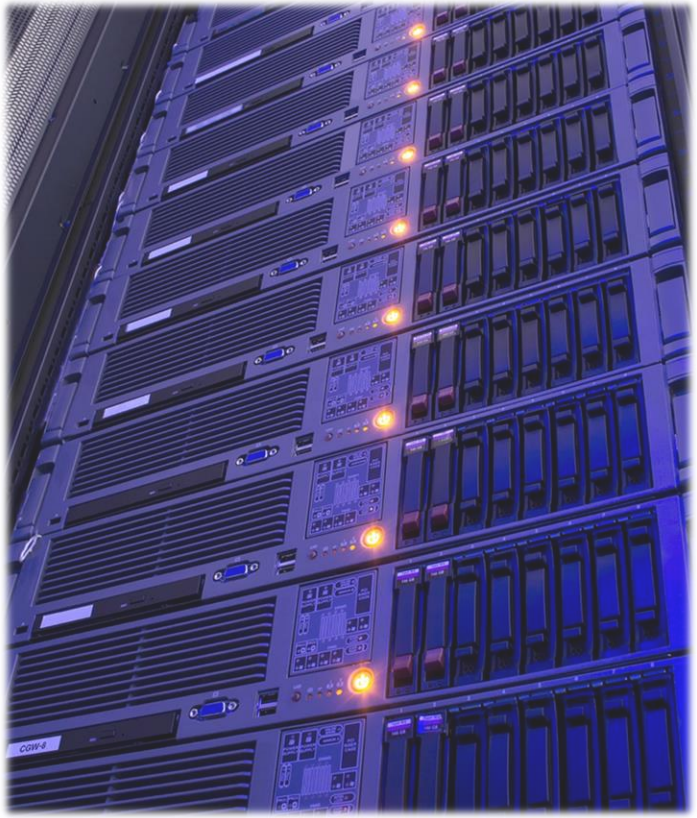
Network Features

- 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.

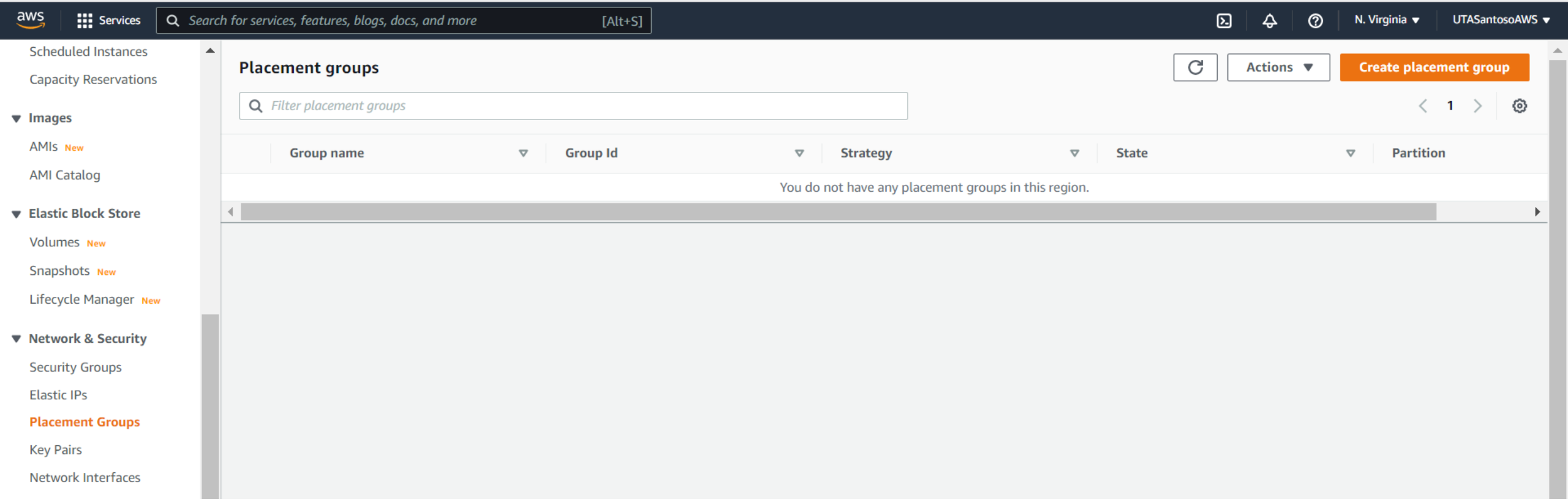
Placement groups

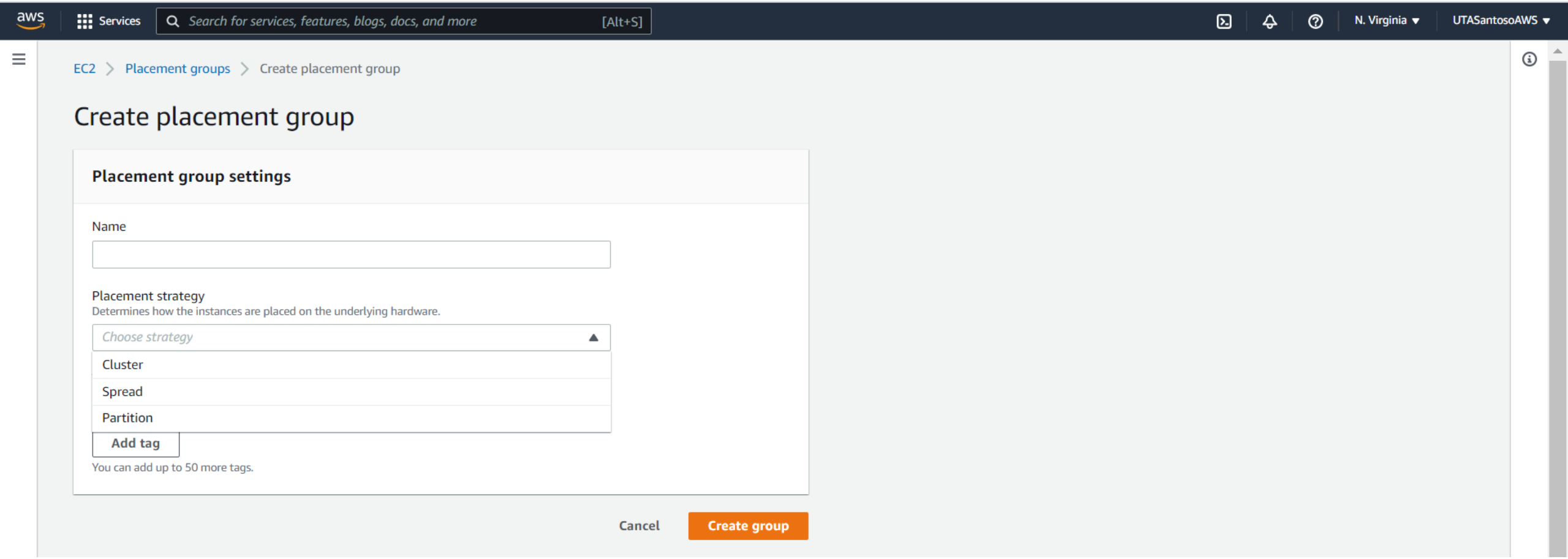
Placement groups enable you to control where instances run in an Availability Zone.

- 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





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

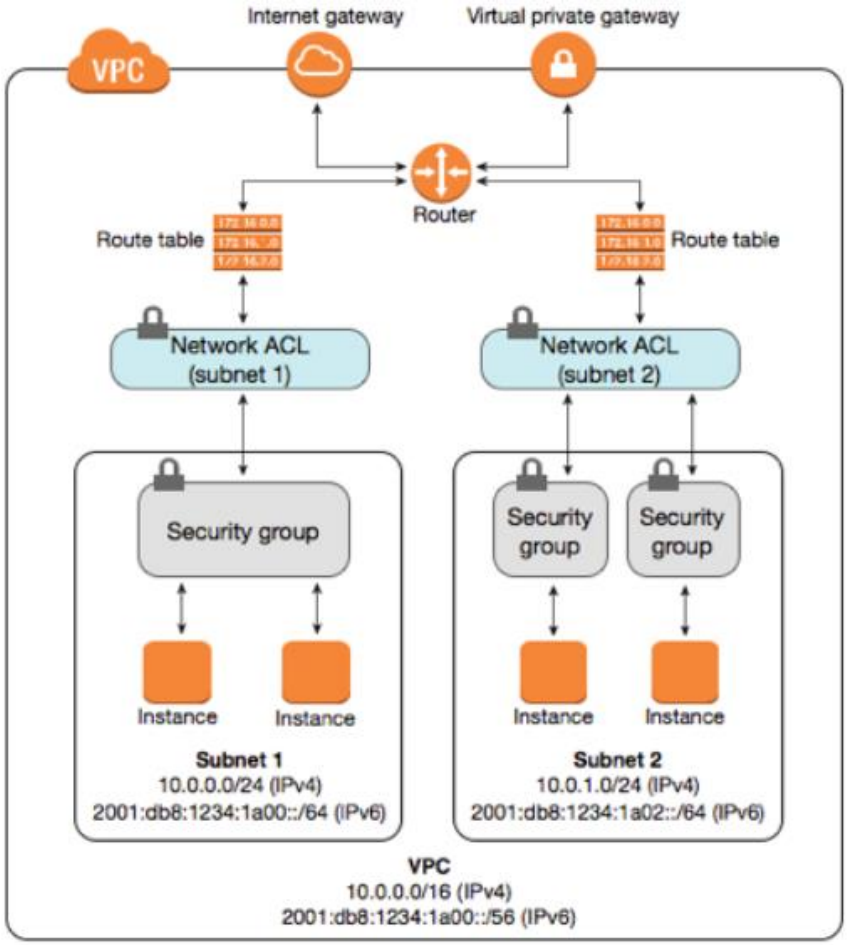
- the inbound traffic that is allowed to reach the instances that are associated with the security group
- The outbound traffic that's allowed to leave them

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html

https://docs.aws.amazon.com/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



Security Group	
Inbound	Outbound

- 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:

Name of SG	Protocol	Port range
ICMP type and code	Source (inbound), Destination (outbound)	Description (optional)

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

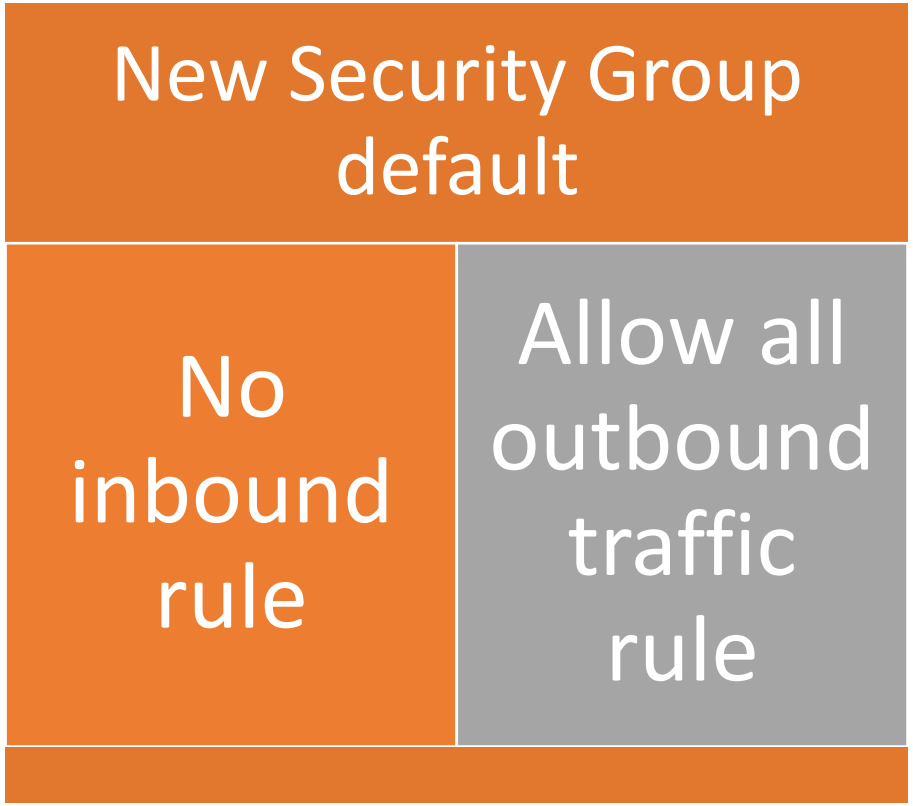
New Created Security Group Default rules

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.



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).

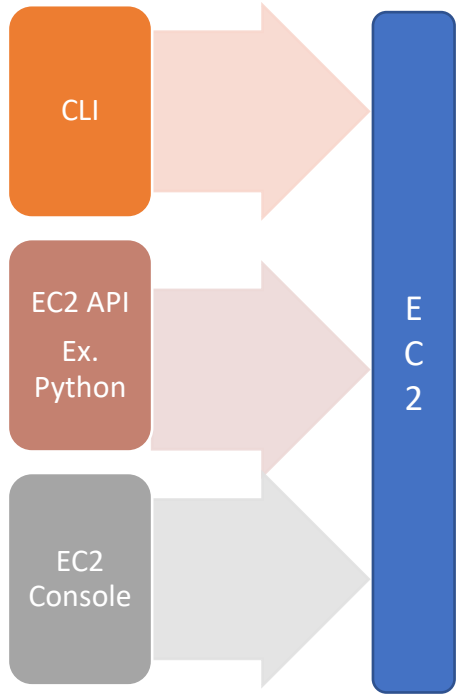
- ❑ If an instance is **launched in EC2 console**, the launch instance wizard automatically defines a "**launch-wizard-xx**" security group, which is associated with the instance.
- ❑ However, if an instance is launched using the **EC2 API or CLI** and a security group is not defined, the instance will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



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.

Note

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.

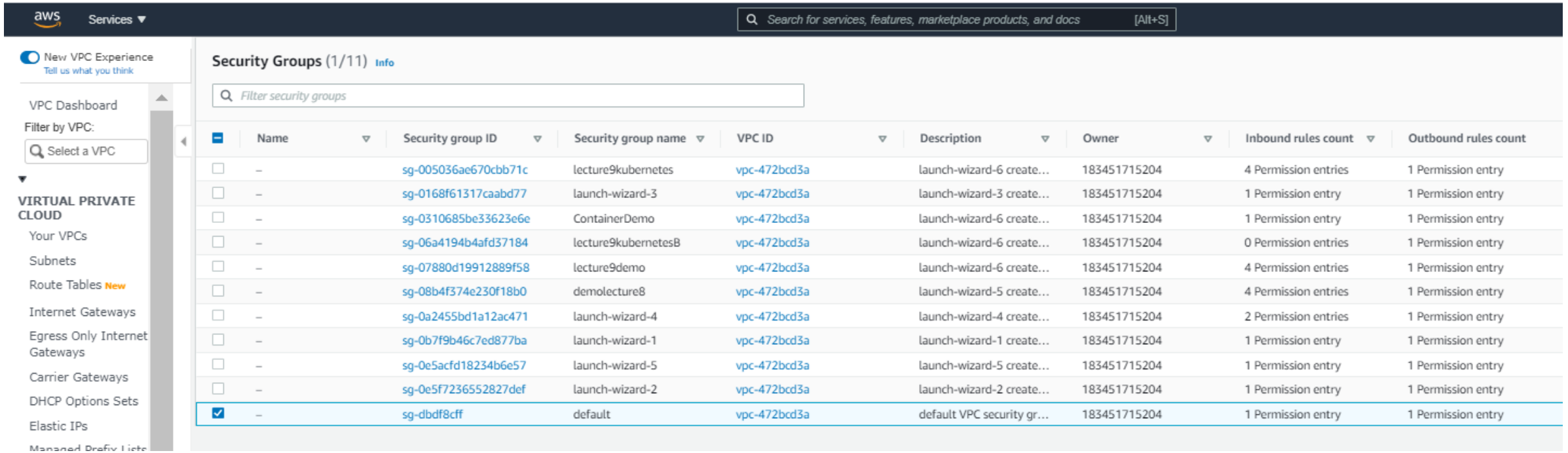
You can change the rules for the default security group.

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.`

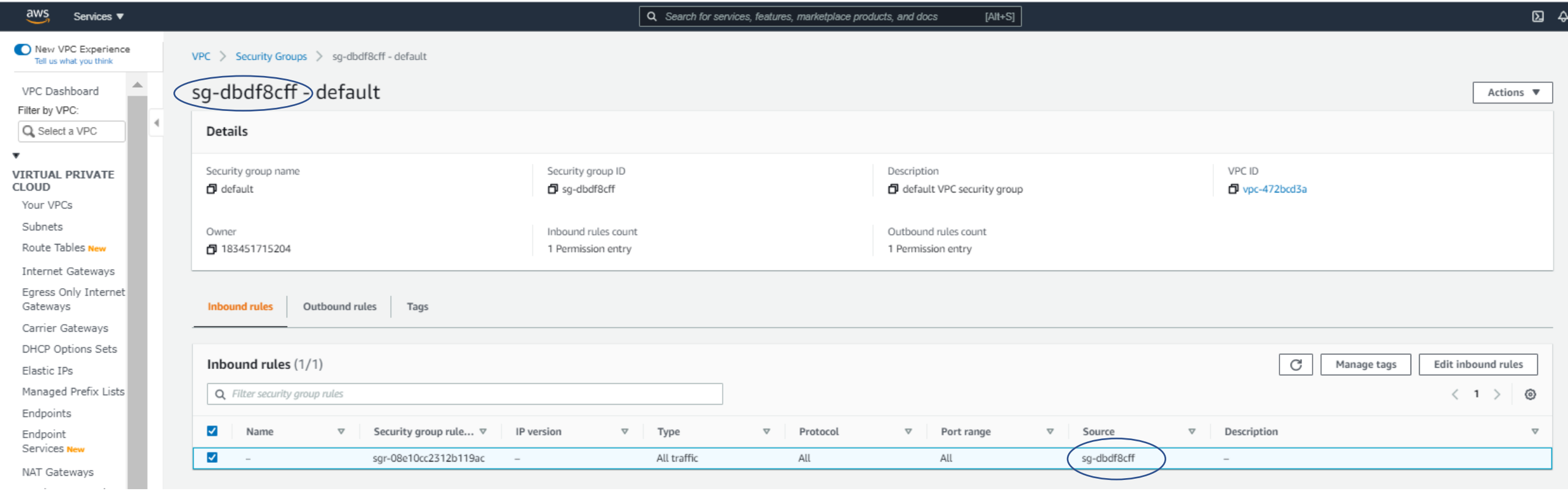
Note

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.

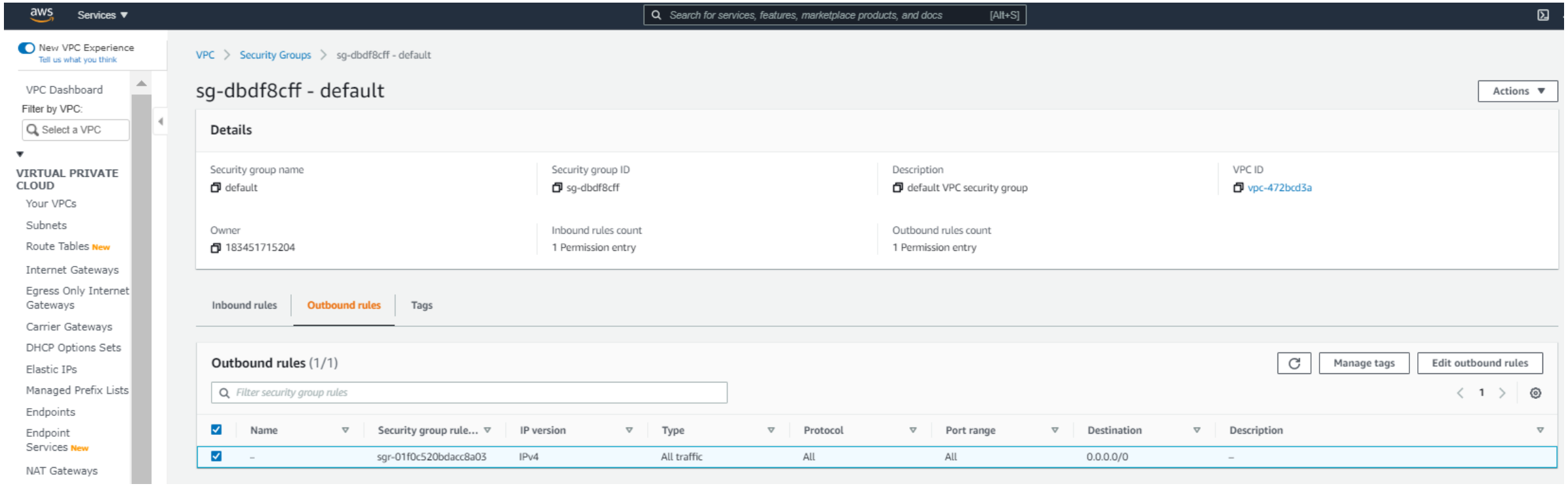
Default VPC Security Group



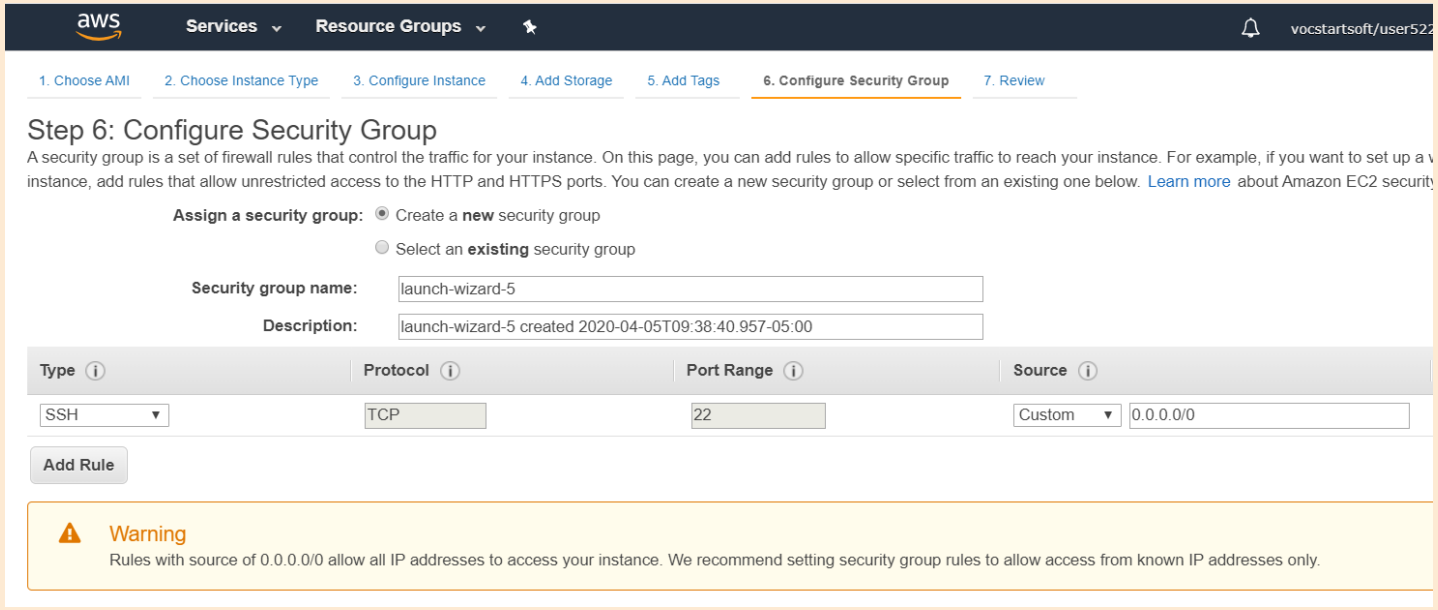
Default VPC Security Group - Inbound



Default VPC Security Group - outbound



Security Group – Instance Creation



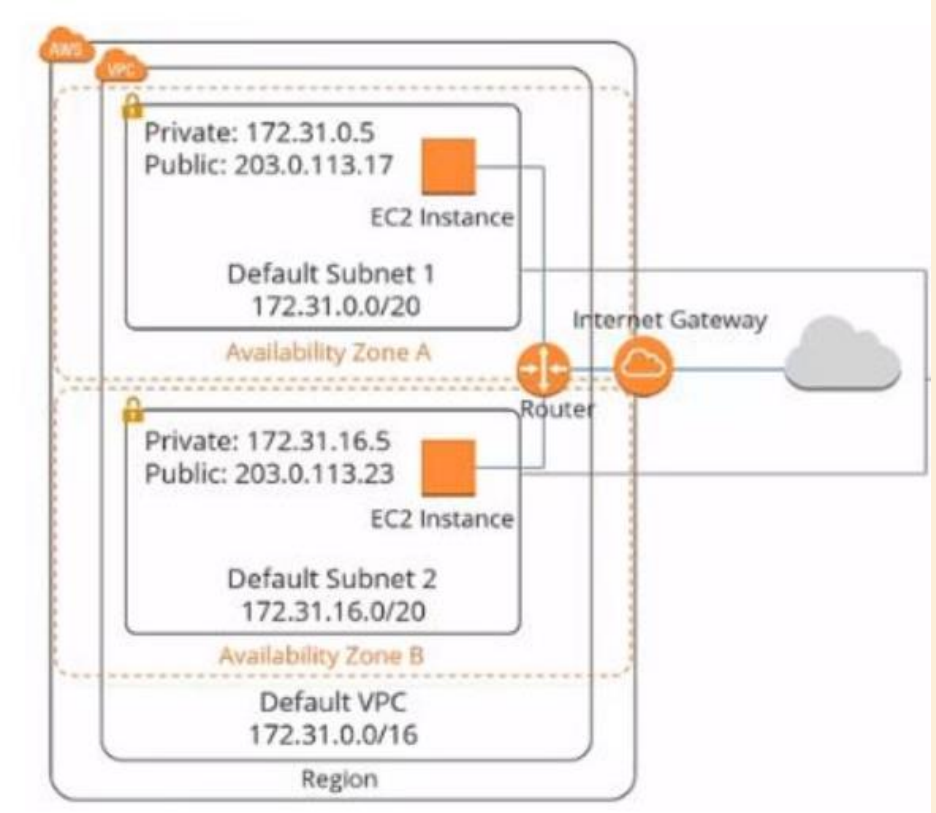
- If you launch an instance using the Amazon EC2 **API** or a **command line tool** and you **don't specify** a security group, the instance is automatically assigned to the **default security group for the VPC**.
- If you launch an instance using the **Amazon EC2 console**, you have an option to create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

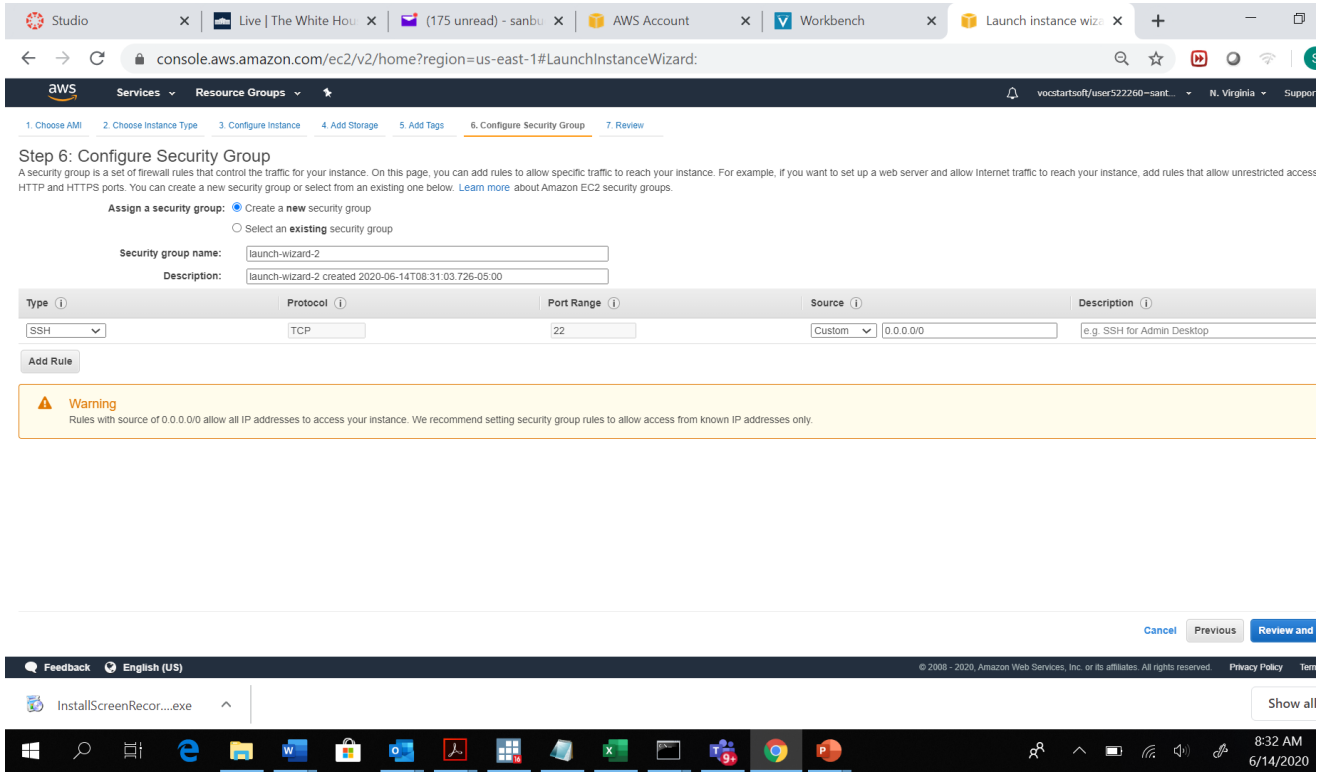
- Source or destination: The **source (inbound rules)** or **destination (outbound rules)** for the traffic.
- https://docs.amazonaws.cn/en_us/AWSEC2/latest/WindowsGuide/security-group-rules.html

EC2 in a VPC

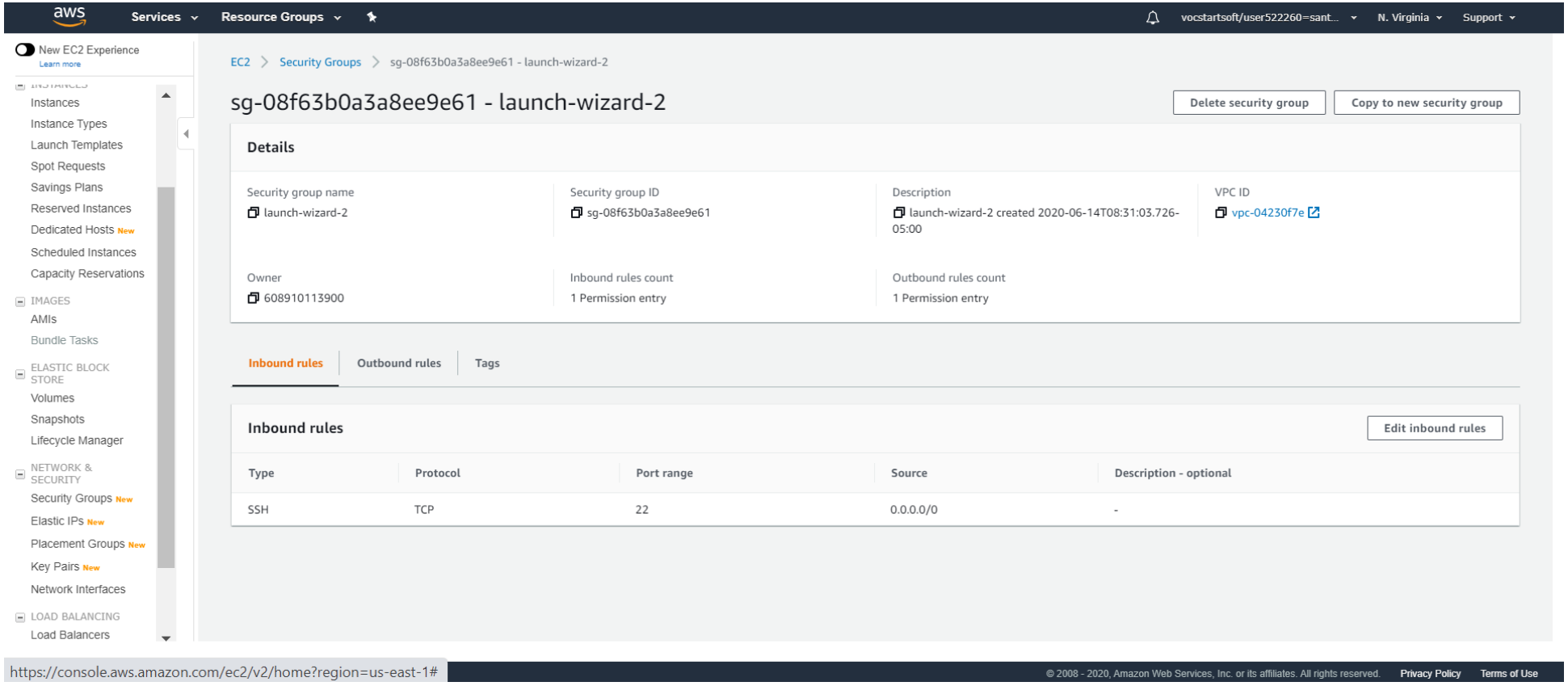


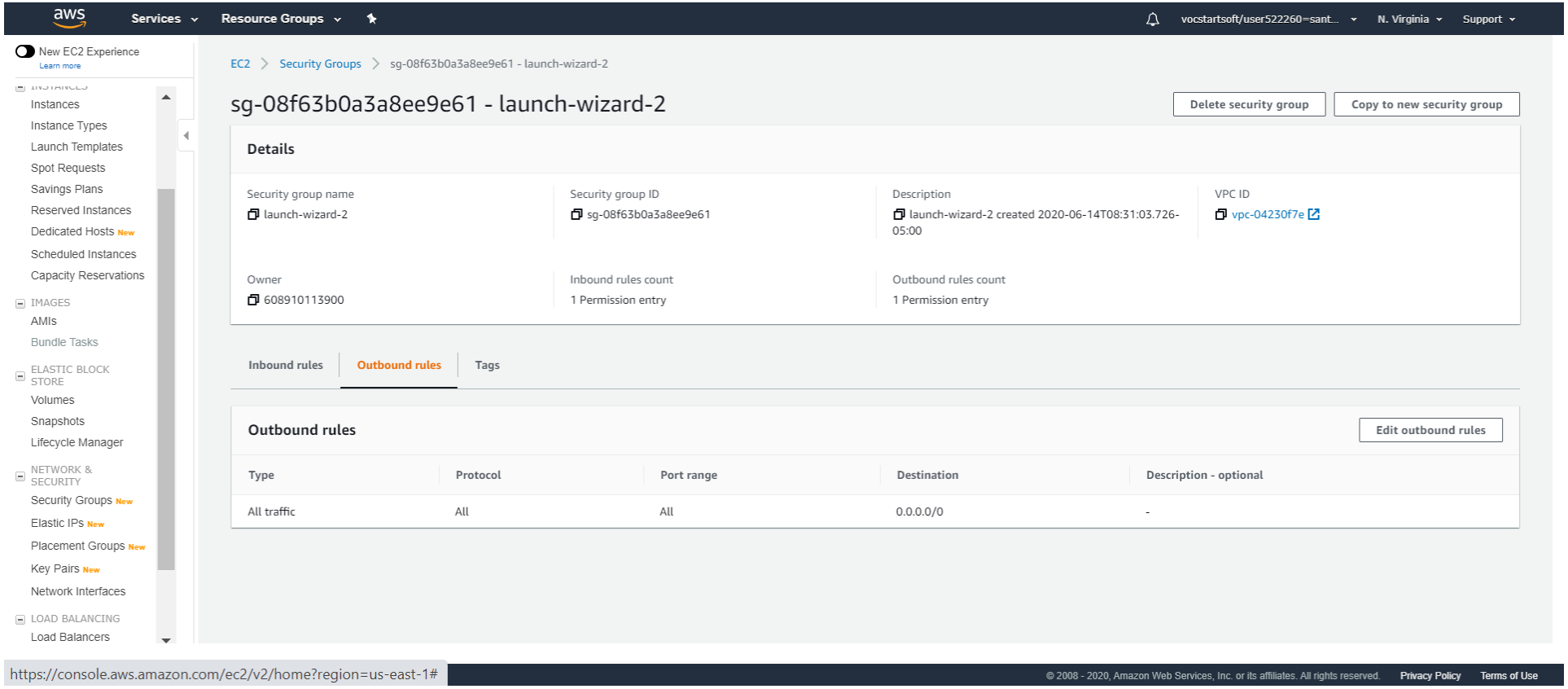
- An EC2 instance must reside in a VPC.
- VPC can be default or customized (you create).
  - A **default VPC** is a logically isolated virtual network in the **AWS** cloud that is automatically created for your **AWS** account the first time you provision Amazon **EC2** resources.
- In this section we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

EC2 in Default VPC

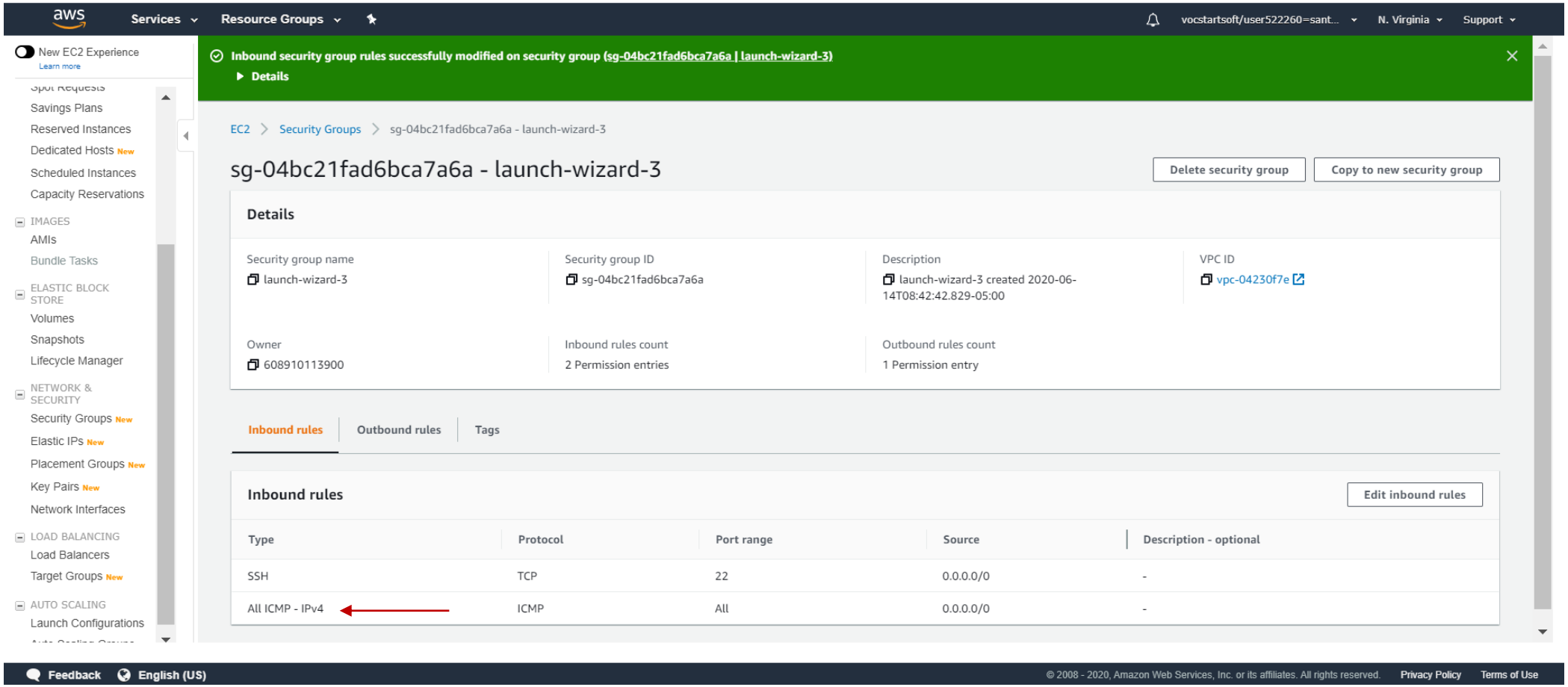


- When creating EC2 (even in default VPC) using console, you have options to create new security group. Unless set to allowed, no traffic can get in including from other internal EC2


aws
Services
Resource Groups
vocstartsoft/user522260=sant...
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
IMAGES
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
launch-wizard-2
Security group ID
sg-08f63b0a3a8ee9e61
Description
launch-wizard-2 created 2020-06-14T08:31:03.726-05:00
VPC ID
vpc-04230f7e
Owner
608910113900
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules
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
vocstartsoft/user522260=sant...
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
IMAGES
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
launch-wizard-2
Security group ID
sg-08f63b0a3a8ee9e61
Description
launch-wizard-2 created 2020-06-14T08:31:03.726-05:00
VPC ID
vpc-04230f7e
Owner
608910113900
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Outbound rules
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



Ping command to check connectivity

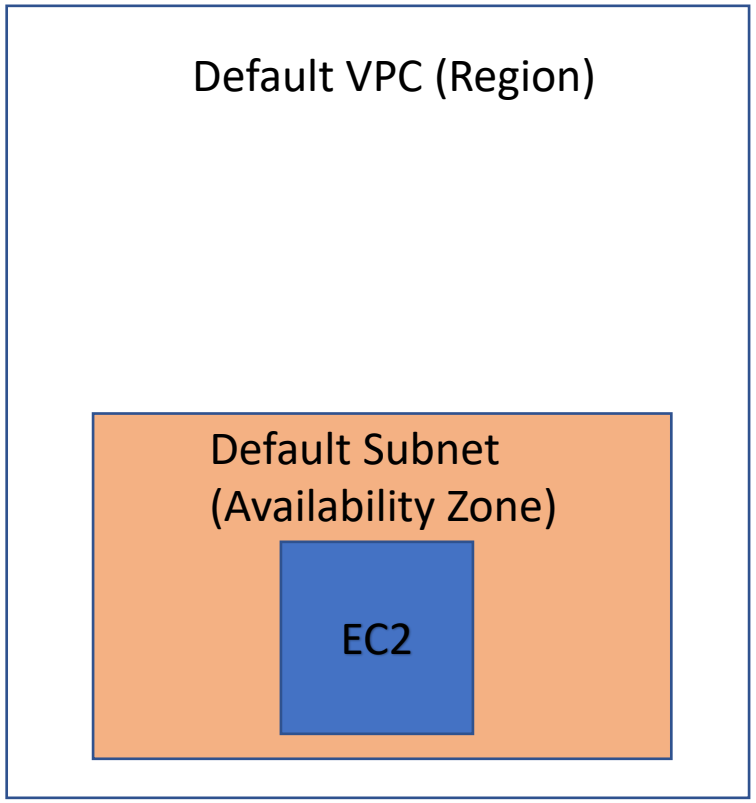
- **Ping** is a command used to test the reachability (connectivity) of a host on an Internet Protocol (IP) network. It is available for virtually all operating systems that have networking capability, including most embedded network administration software.
  - In linux you can use ping ipaddress or 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.

Launching EC2

Launching EC2 Steps in Console

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

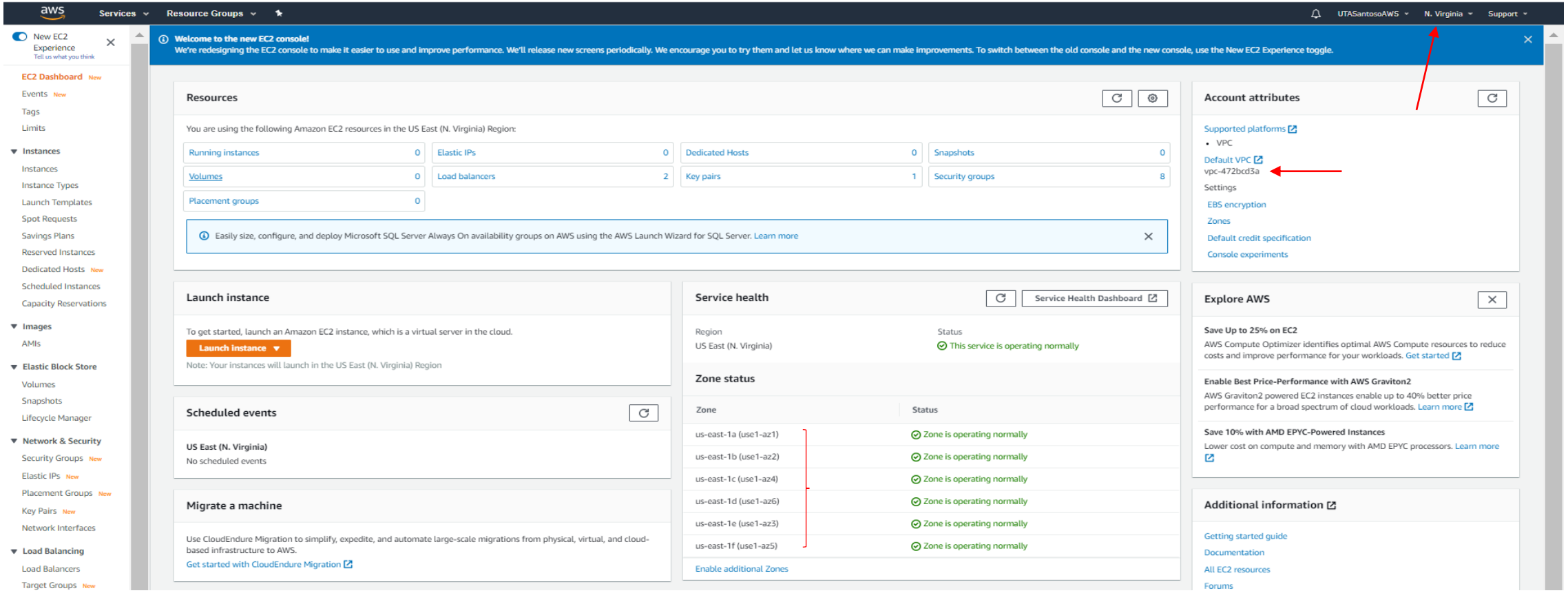
EC2 in Default VPC



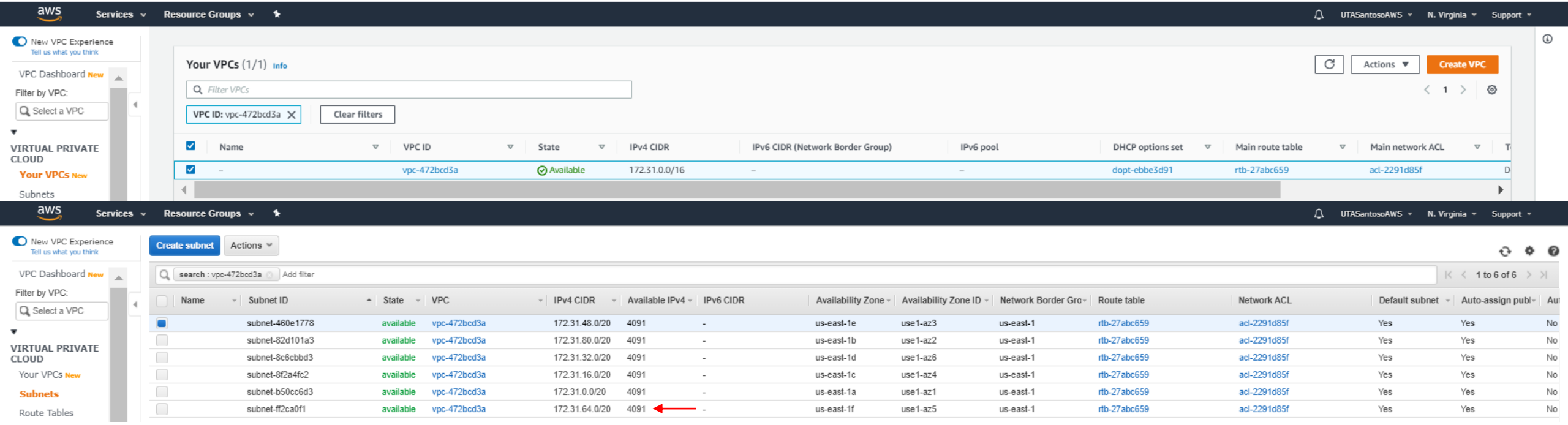
For each AWS account

- 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


aws
Services
Resource Groups
UTASantosoAWS
N. Virginia
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events New
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
Images
AMIs
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
Target Groups New
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
Launch instance
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance
Note: Your instances will launch in the US East (N. Virginia) Region
Service health
Service Health Dashboard
Region
US East (N. Virginia)
Status
This service is operating normally
Zone status
Zone
Status
us-east-1a (use1-az1) Zone is operating normally
us-east-1b (use1-az2) Zone is operating normally
us-east-1c (use1-az4) Zone is operating normally
us-east-1d (use1-az6) Zone is operating normally
us-east-1e (use1-az3) Zone is operating normally
us-east-1f (use1-az5) Zone is operating normally
Enable additional Zones
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
Scheduled events
US East (N. Virginia)
No scheduled events
Migrate a machine
Use CloudEndure Migration to simplify, expedite, and automate large-scale migrations from physical, virtual, and cloud-based infrastructure to AWS.
Get started with CloudEndure Migration
Additional information
Getting started guide
Documentation
All EC2 resources
Forums

Default VPC and its Subnets



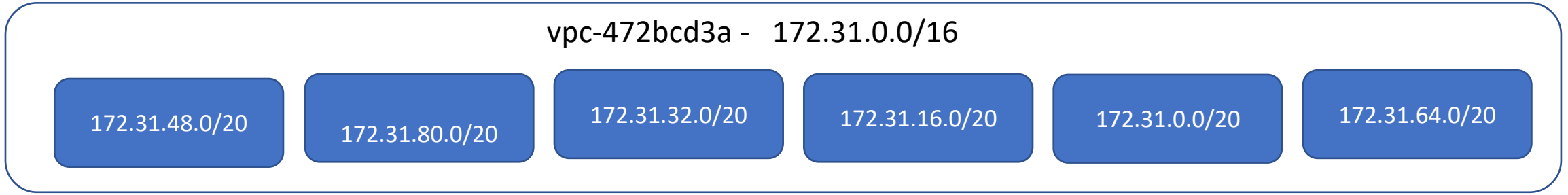
Notice the CIDR notations for the VPC and its subnets (all default):

- VPC – 172.31.0.0/16
- There are 6 subnets (1 per each AZ)
  - CIDR – 172.31.xx.xx/20
  - Availabe 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

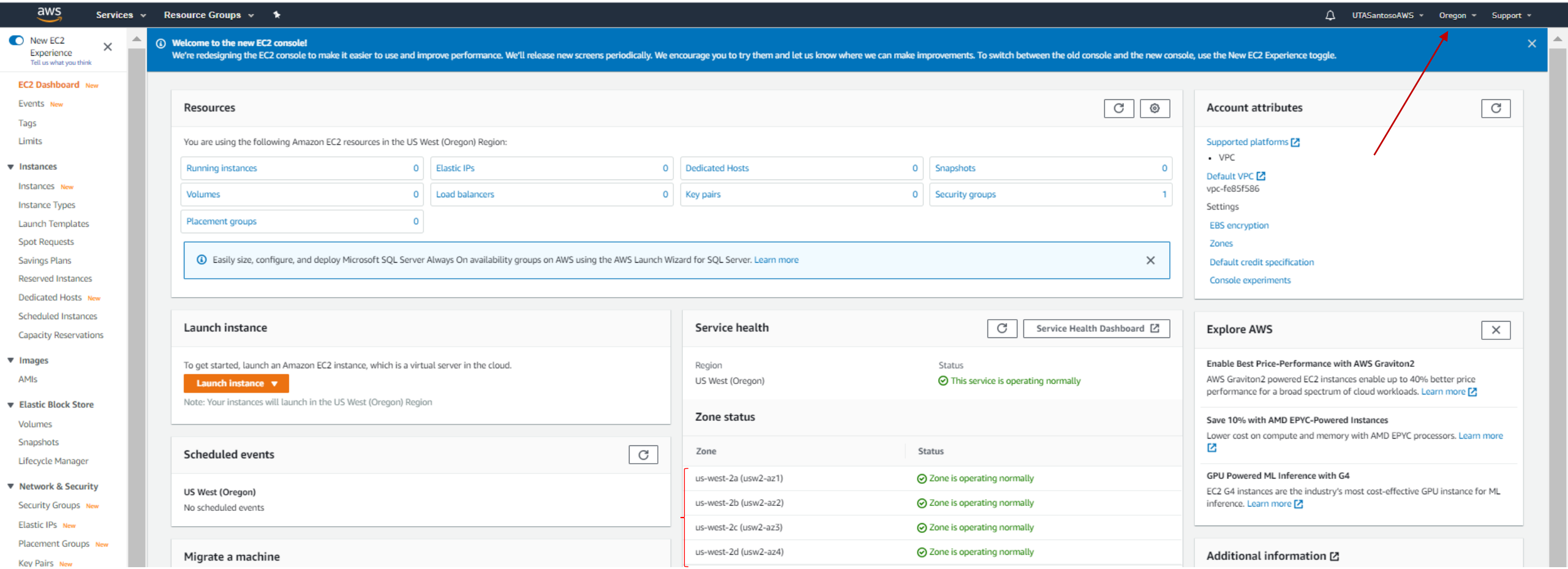
Default VPC – Region N. Virginia



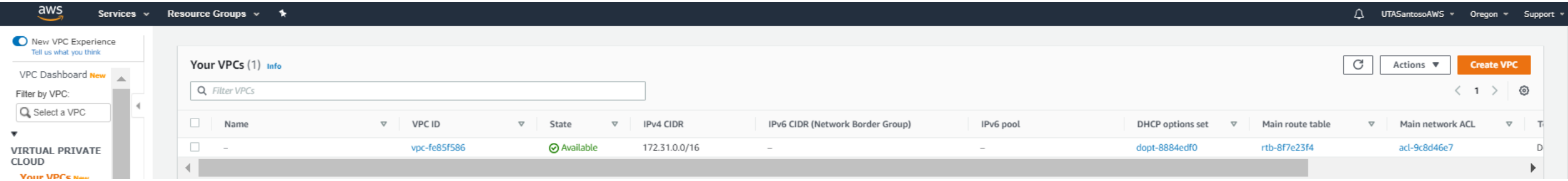
		Octet 4								Octet 3								Octet 2								Octet 1							
	Bit value	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
VPC	172.31.0.0/16	1	0	1	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Subnet	172.31.48.0/20	1	0	1	0	1	1	0	0	0	0	0	1	1	1	1	1	0	0	1	1	0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

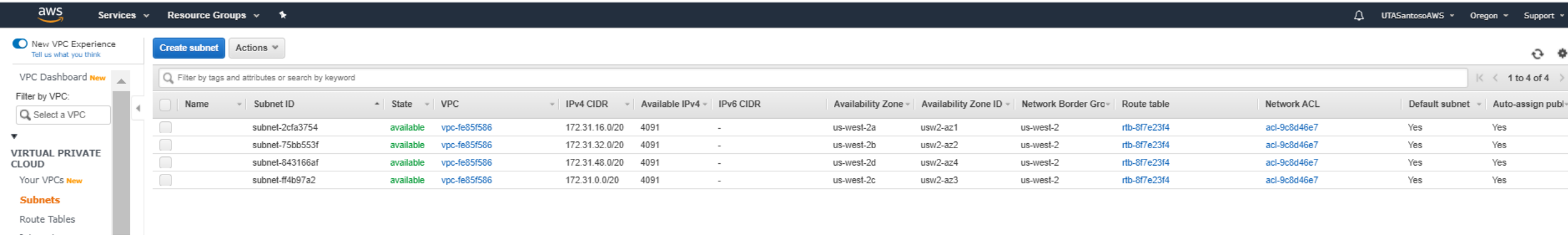
- Defined by AWS
- 16 bits from the left to identify the network (/16)
- Bit 17-Bit20 from the left to identify the subnet within that network (between /16 of VPC and /20 of subnets)
  - You could have 2^4 = 16 subnets
  - Default- 1 subnet per AZ
- The rest 12 bits (32-12) to identify the hosts in each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

What if we change region?



Default Subnets

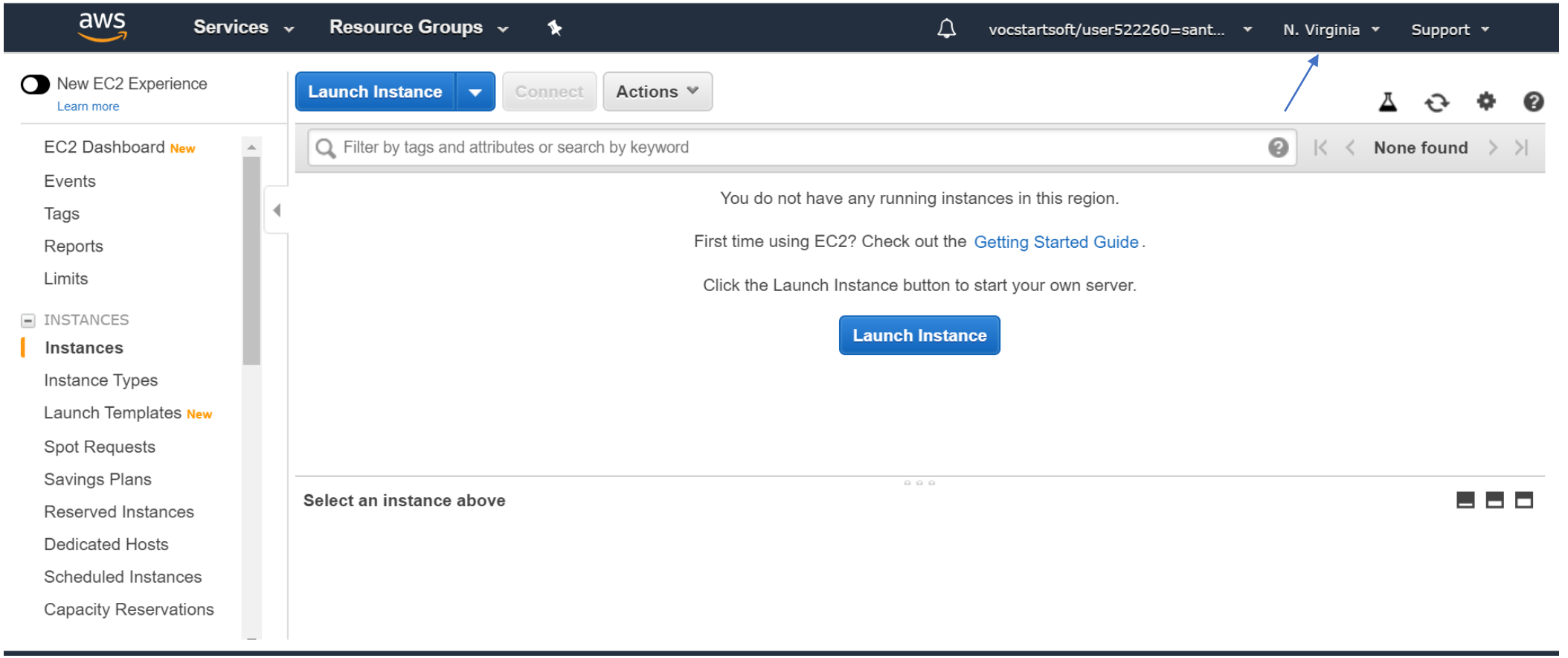




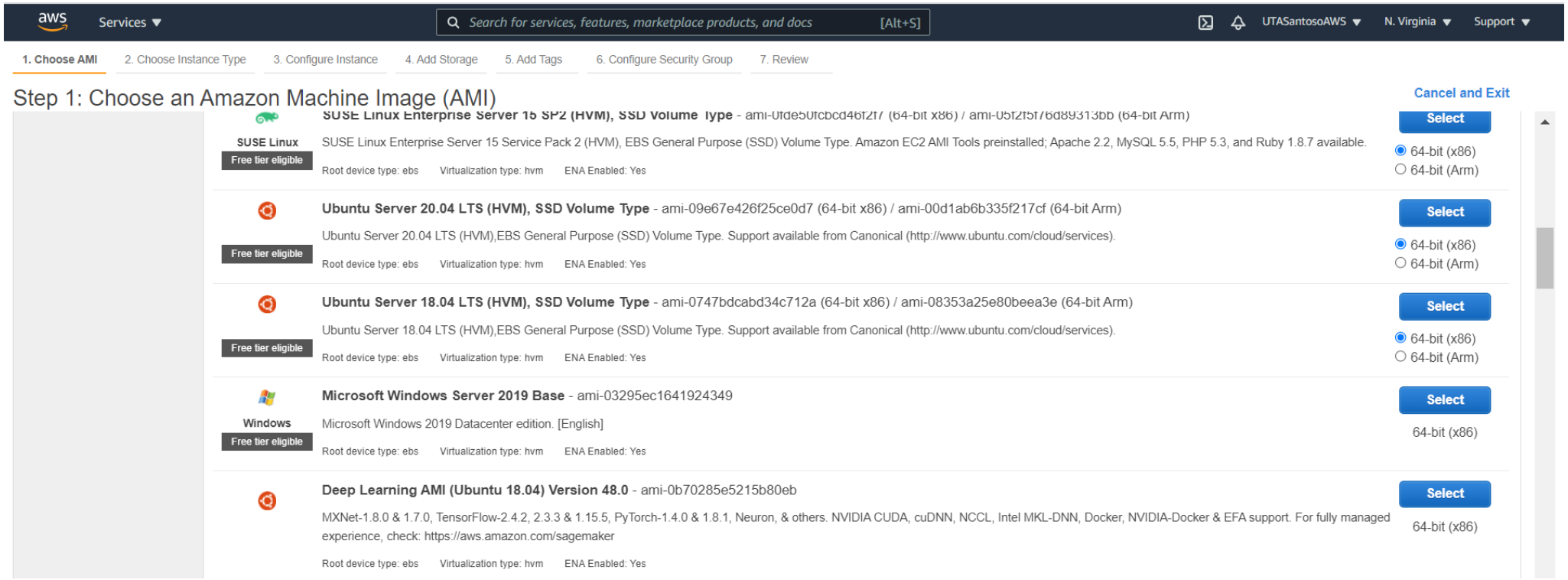
Example

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

Spin Up an EC2

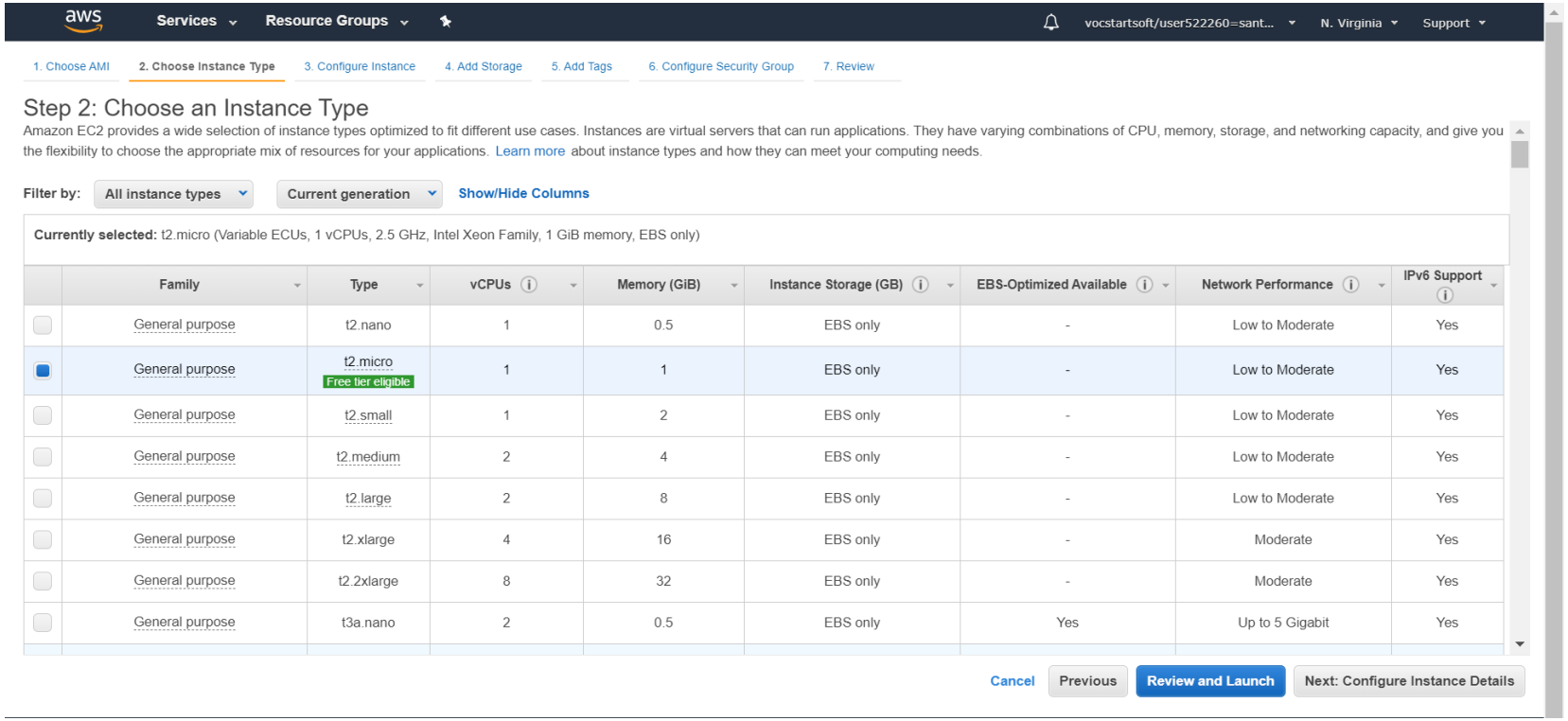


Select an AMI



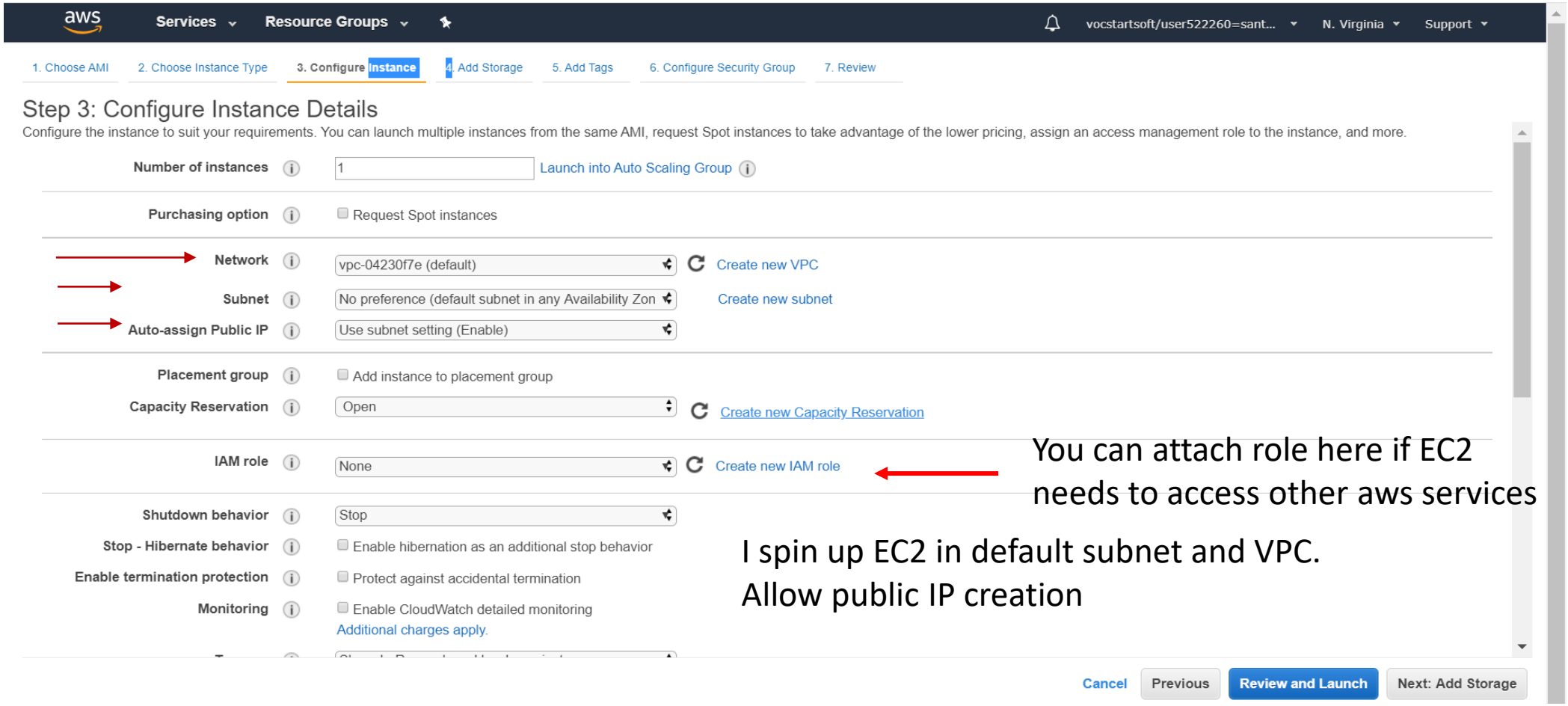
I choose **Ubuntu** 20.04 (or the latest) so I can use certain commands to install nginx web server.
Ubuntu is a linux distro based on **Debian**, most software **is** usable on both distros.
Note Amazon Linux is not compatible with ubuntu.

Choose Instance Type



t2.micro (free tier) is enough for our purpose.
Note: it is possible to change instance type afterwards (after observing the application needs).

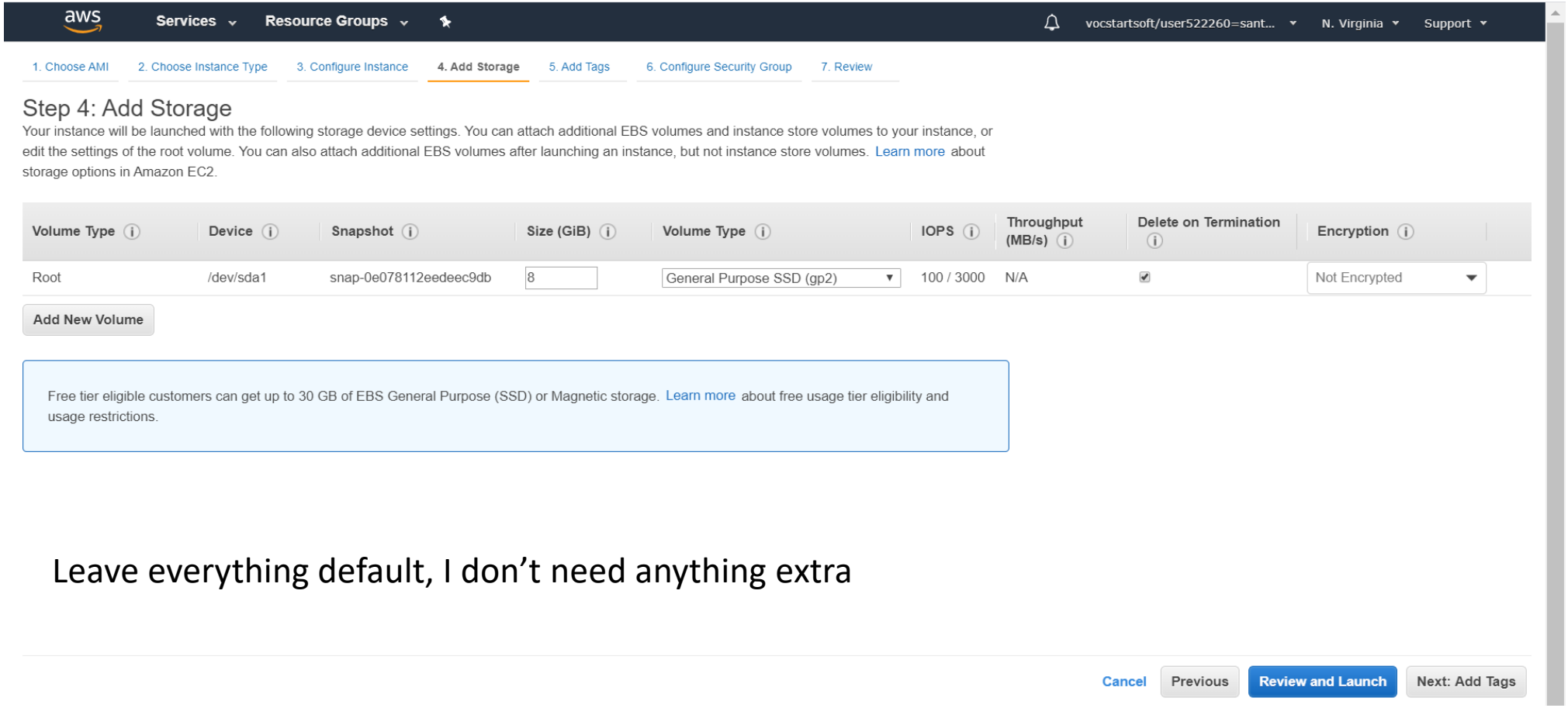
EC2 in Default VPC (N.Virginia)



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

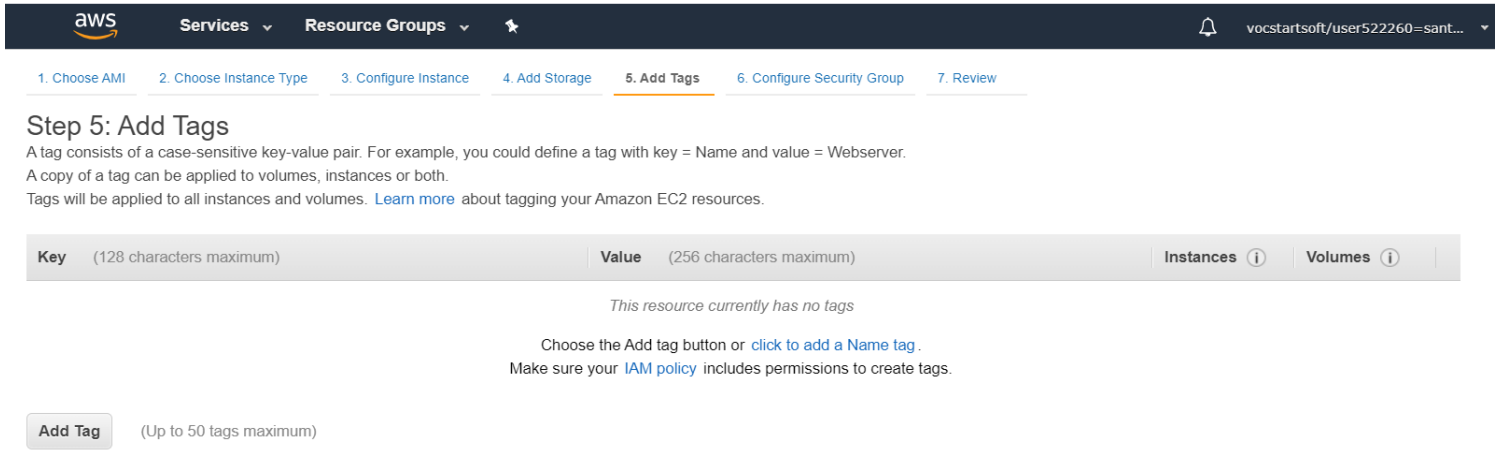
I spin up EC2 in default subnet and VPC.
Allow public IP creation

Add Storage

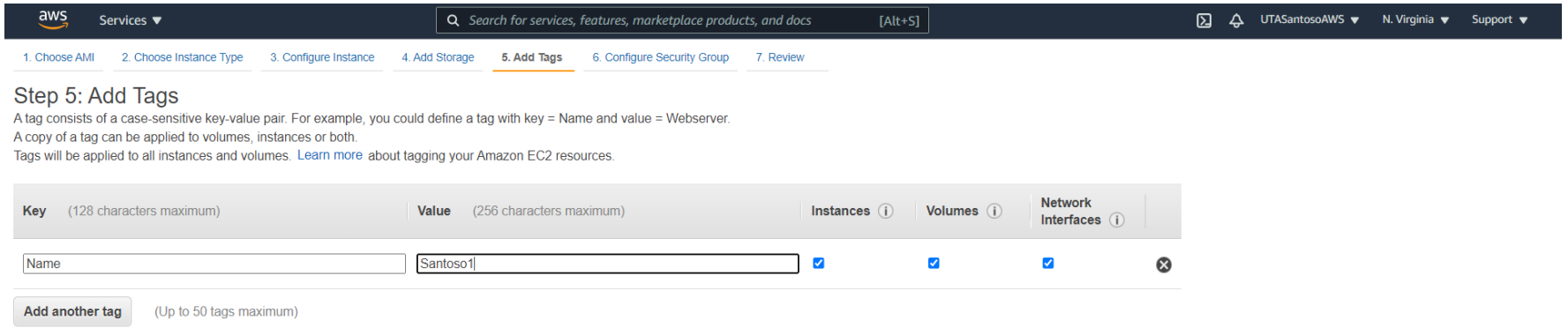


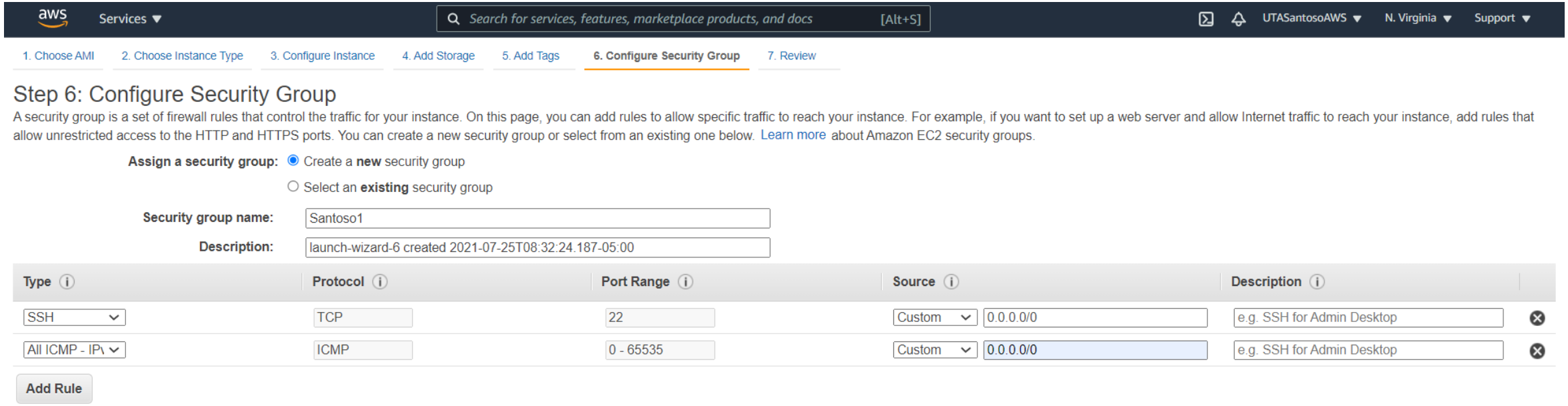
Leave everything default, I don't need anything extra

Add Tags (Optional)



Optional- you may want to add "Name" for the EC2.

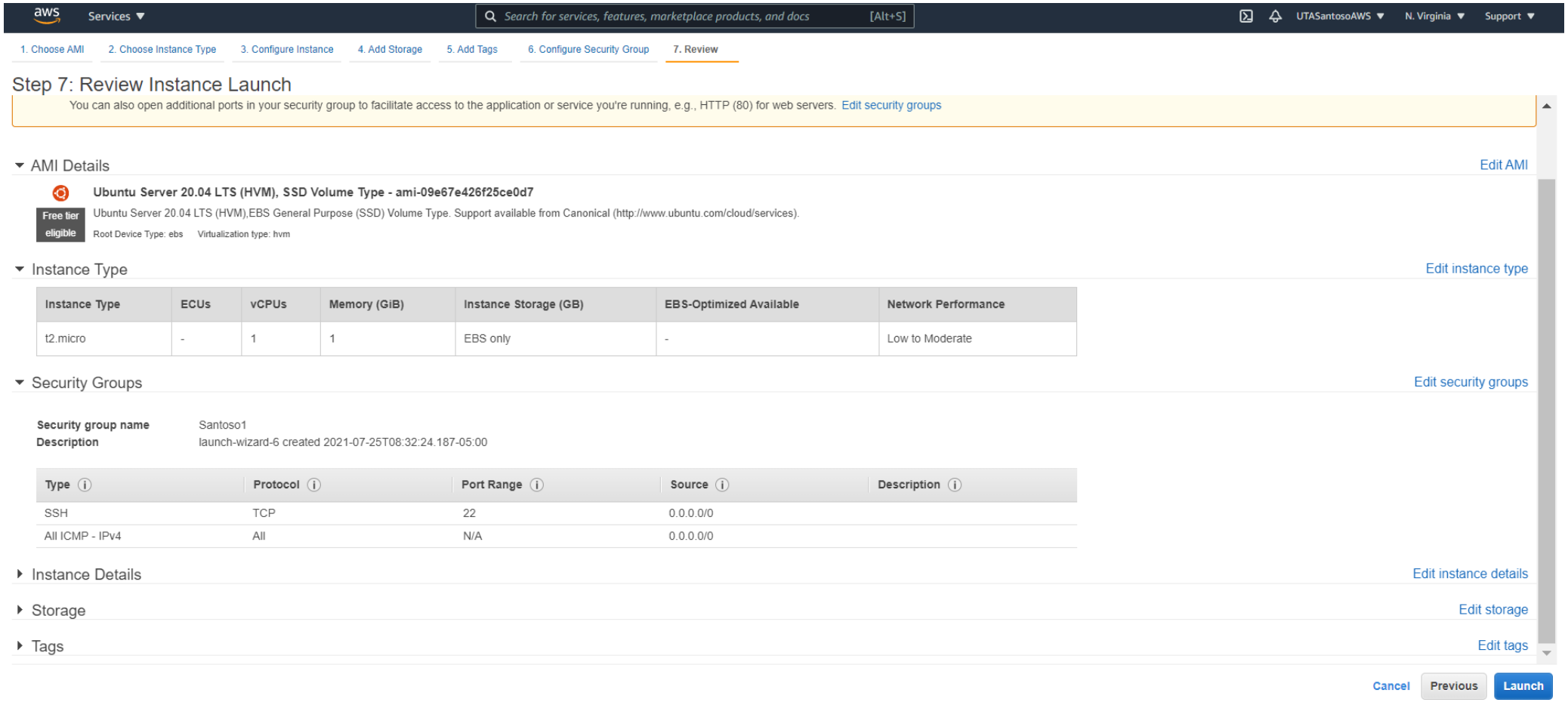




Create a New Security Group

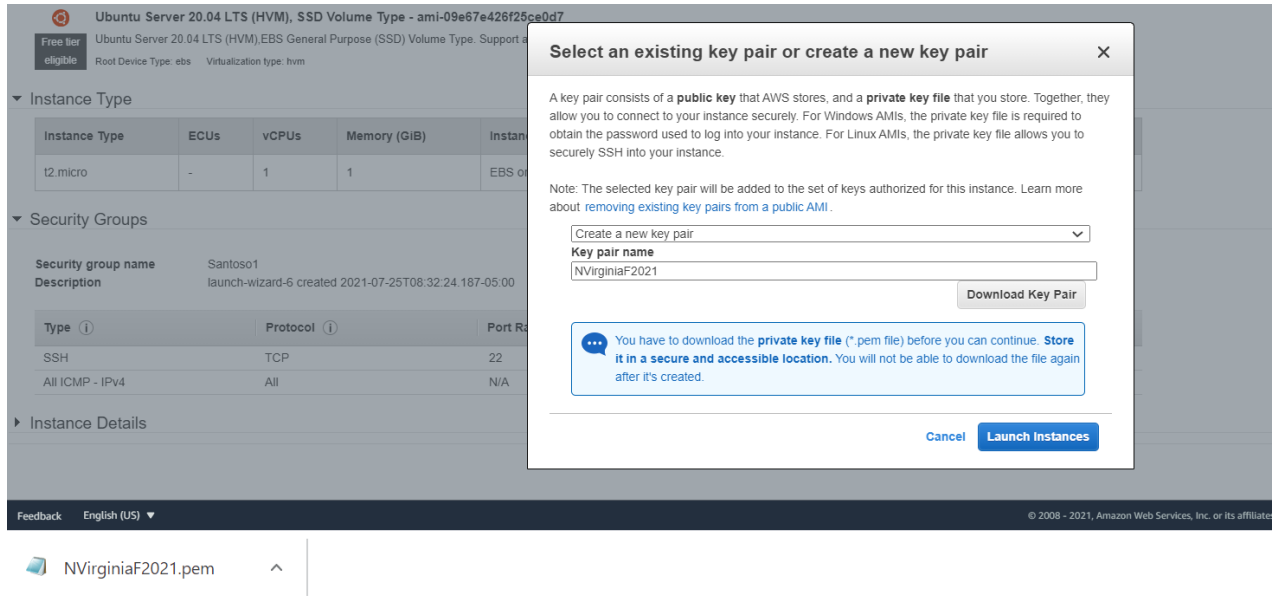
- Here you can set inbound firewall per instance basis.
- I create a new one and name it santoso1
- I add ICMP so we can ping to this EC2 from everywhere (0.0.0.0/0)
- Note: if we want to use http (web server), it also has to be allowed)

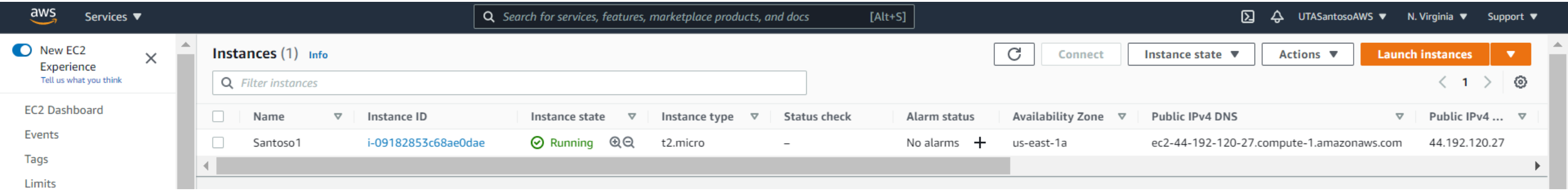
Review

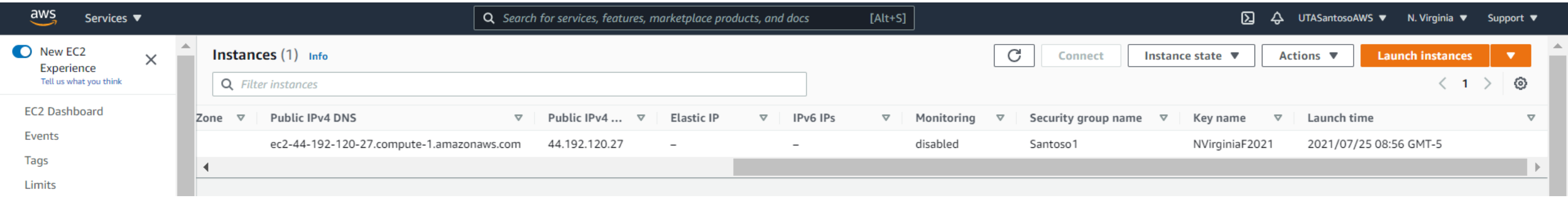

aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
UTASantosoAWS
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
You can also open additional 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 Santoso1
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

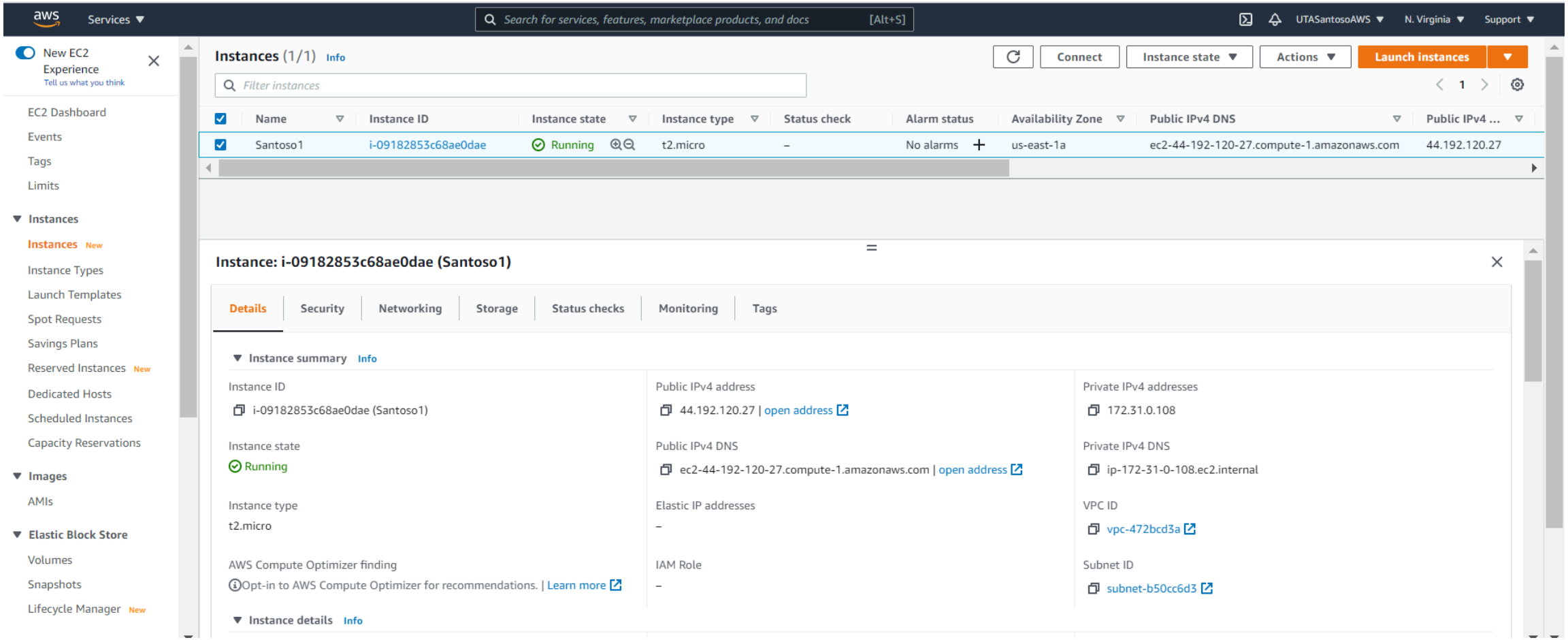
- Create, name, and download a key pair
- Can use existing one (same key pair can be used for multiple instances)
- **Key pair is tied to a region.**
- .pem must be saved in your local laptop drive.
- Convert .pem to .ppk to be used in Putty (this is your private key)
- Launch the instance



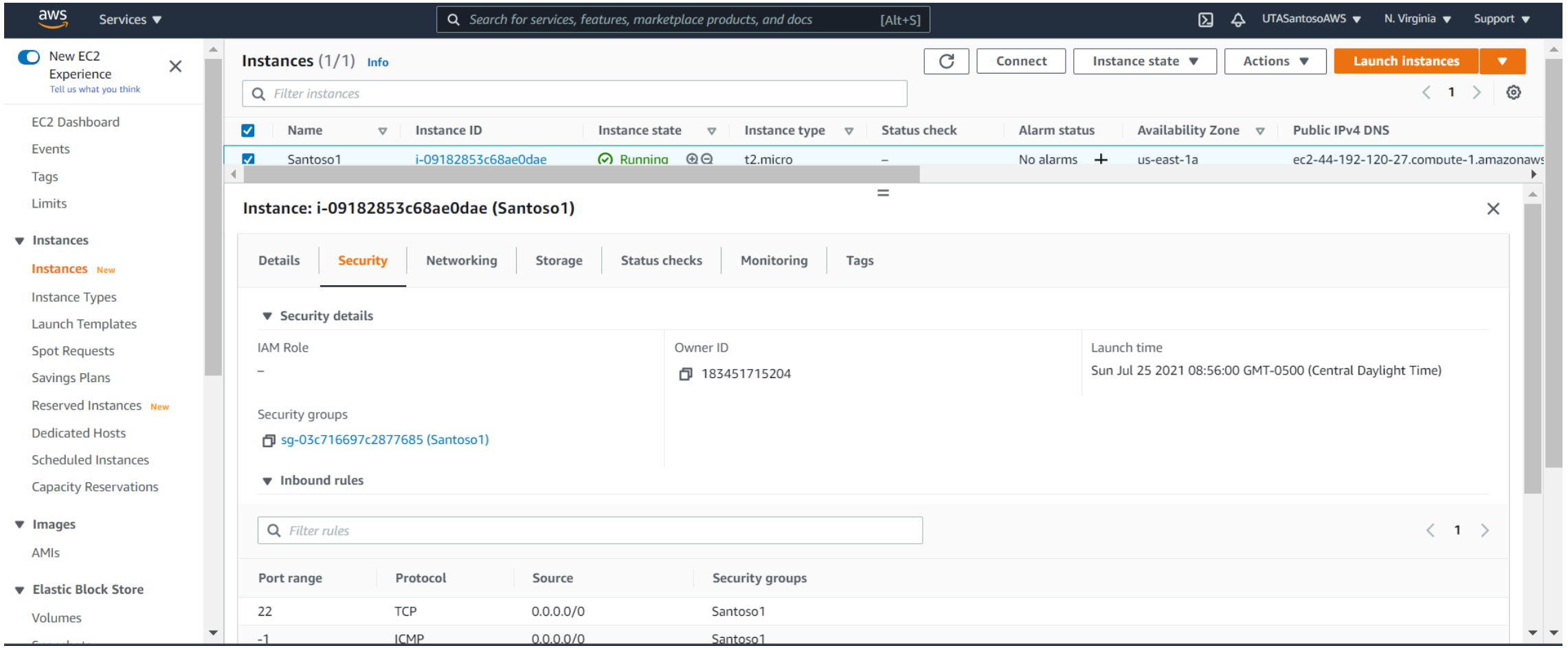

aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
UTASantosoAWS
N. Virginia
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard
Events
Tags
Limits
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Filter instances
1
Name
Instance ID
Instance 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


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
Connect
Instance state
Actions
Launch instances
Filter instances
1
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

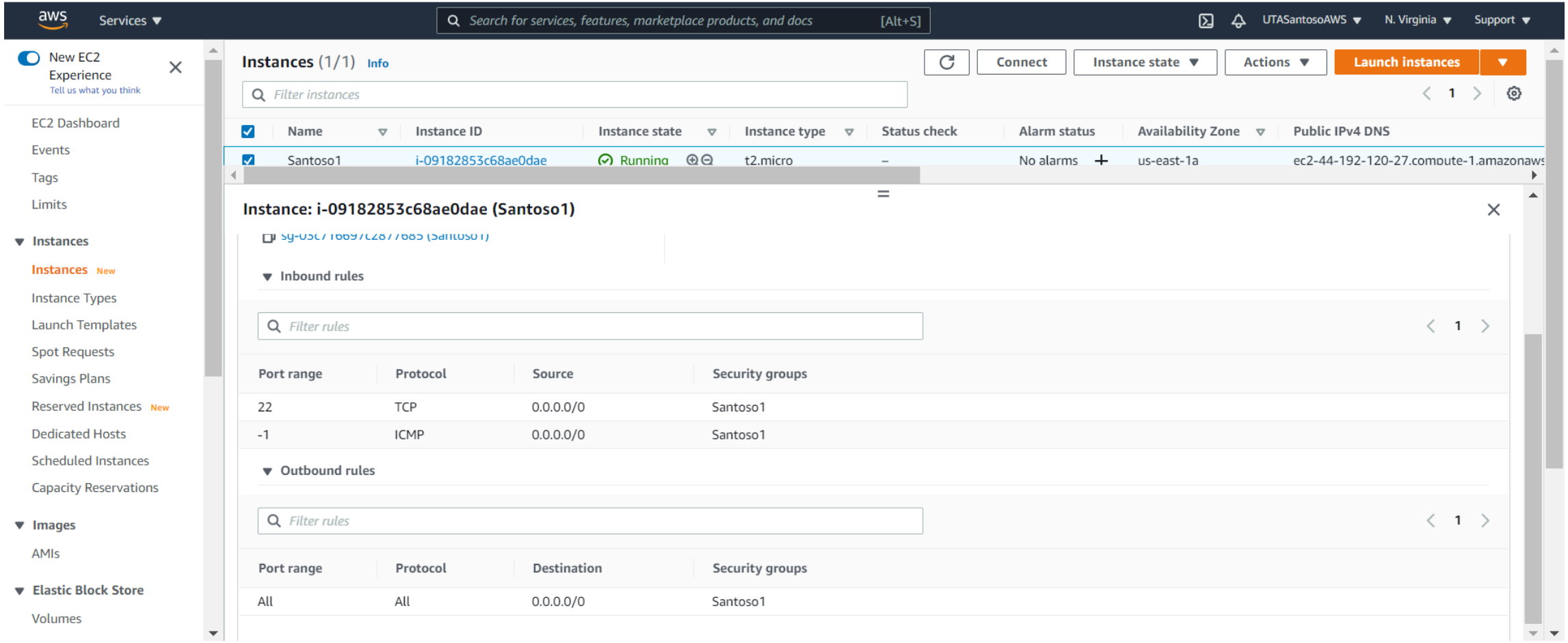
EC2 Detail


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
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager New
Instances (1/1) Info
Filter instances
Connect
Instance state
Actions
Launch instances
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
Instance summary Info
Instance ID
i-09182853c68ae0dae (Santoso1)
Instance state
Running
Instance type
t2.micro
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Public IPv4 address
44.192.120.27 | open address
Public IPv4 DNS
ec2-44-192-120-27.compute-1.amazonaws.com | open address
Elastic IP addresses
–
IAM Role
–
Private IPv4 addresses
172.31.0.108
Private IPv4 DNS
ip-172-31-0-108.ec2.internal
VPC ID
vpc-472bcd3a
Subnet ID
subnet-b50cc6d3
Instance details Info

EC2 Security Group



EC2 Security Group



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.
```

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.

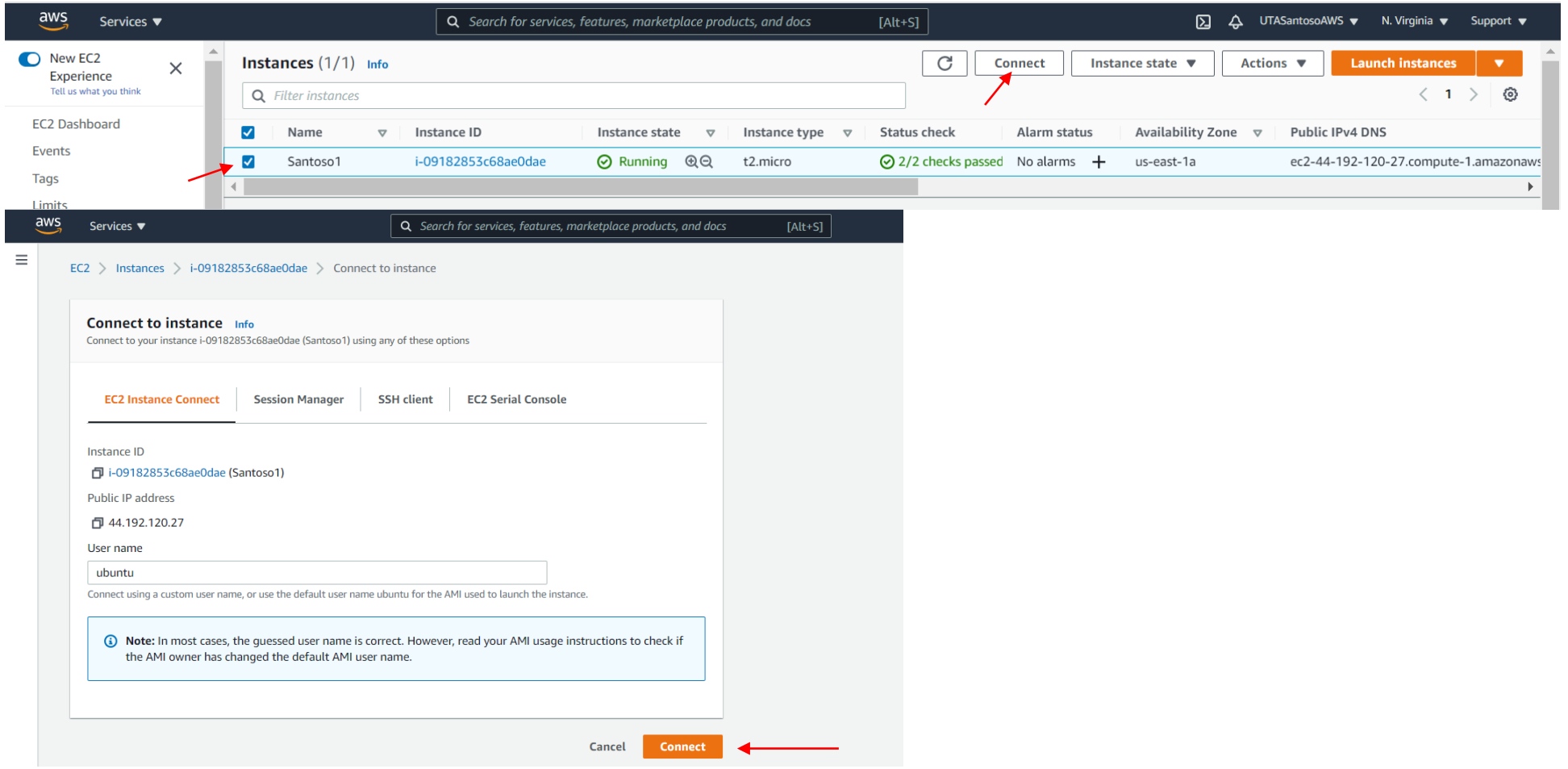
The 3 below are discussed in this chapter (ssh connection):

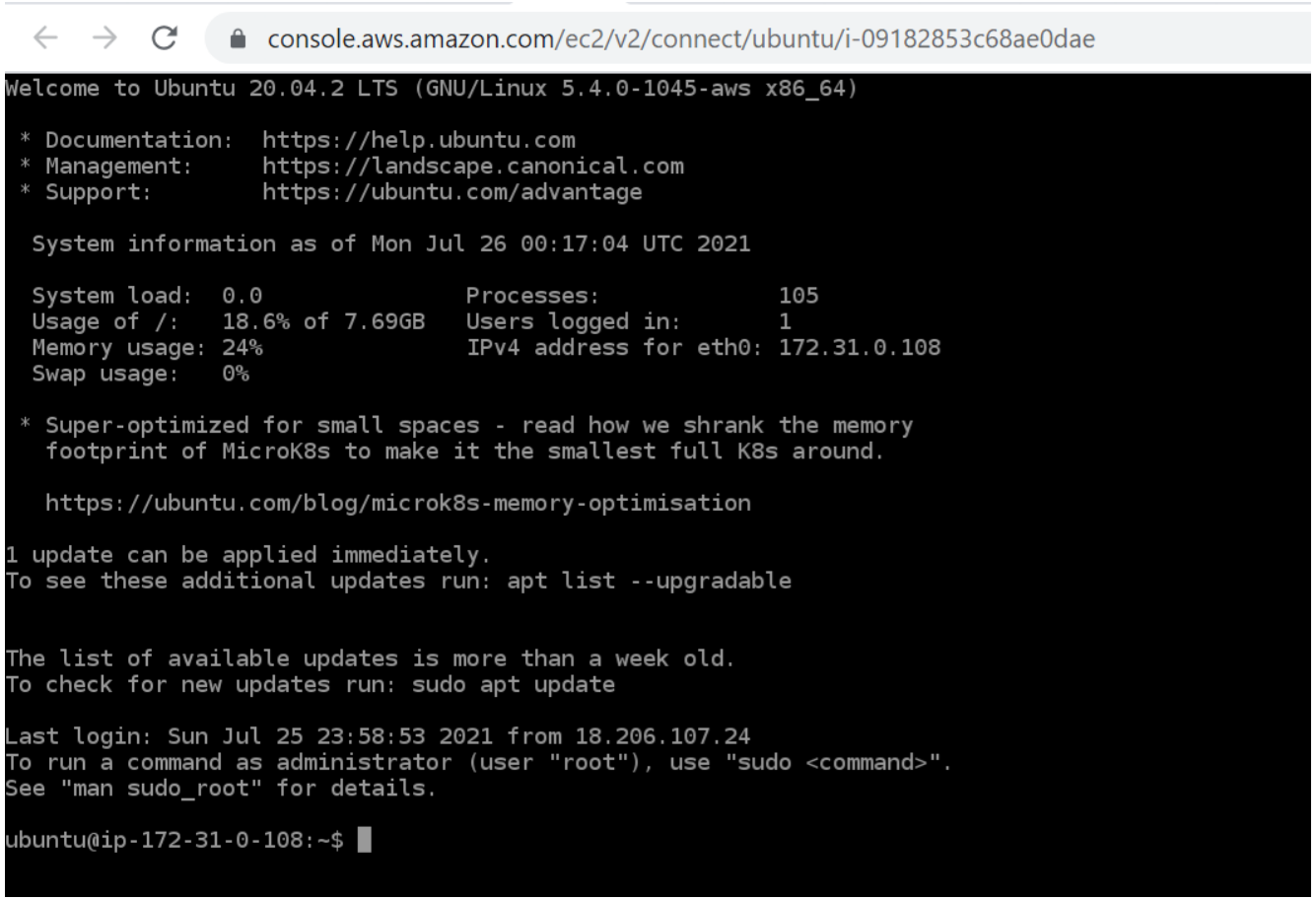
- EC2 Instance Connect
  - can easily connect to instances from browser-based SSH experience in the EC2 console.
  - US East (Ohio and N. Virginia), US West (N. California and Oregon), Asia Pacific (Mumbai, Seoul, Singapore, Sydney, and Tokyo), Canada (Central), EU (Frankfurt, Ireland, London, and Paris), 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)

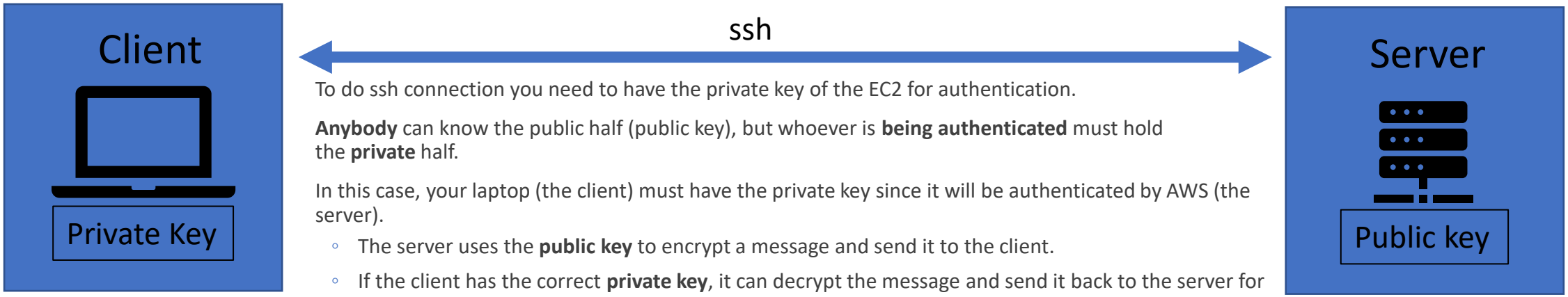




May need to refresh when timed out/hang

SSH Connection – ssh client (through laptop)

ssh Key-Pair for Authentication



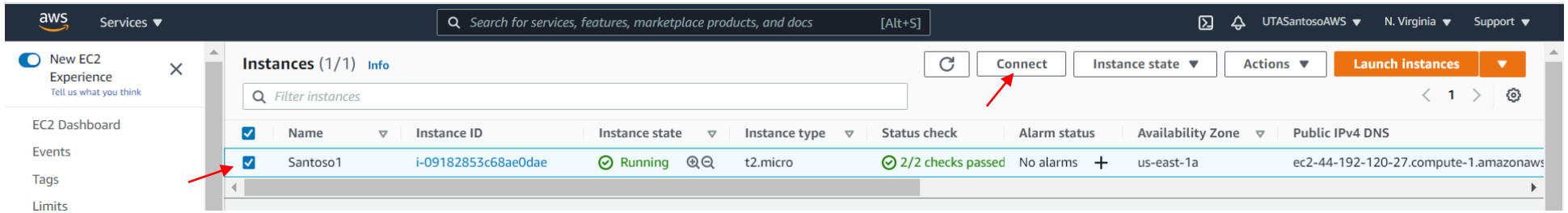
To do ssh connection you need to have the private key of the EC2 for authentication.

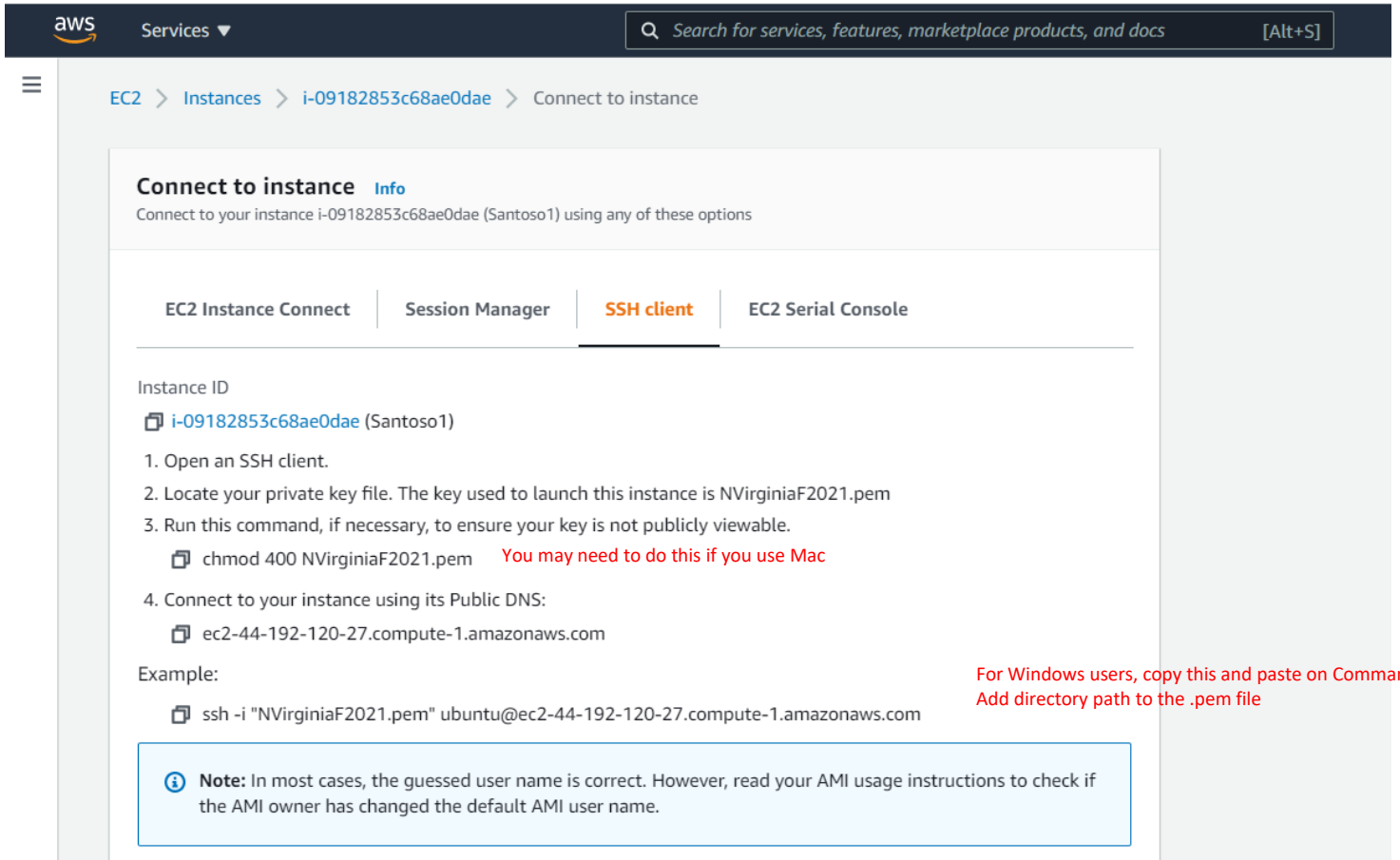
Anybody can know the public half (public key), but whoever is **being authenticated** must hold the **private** half.

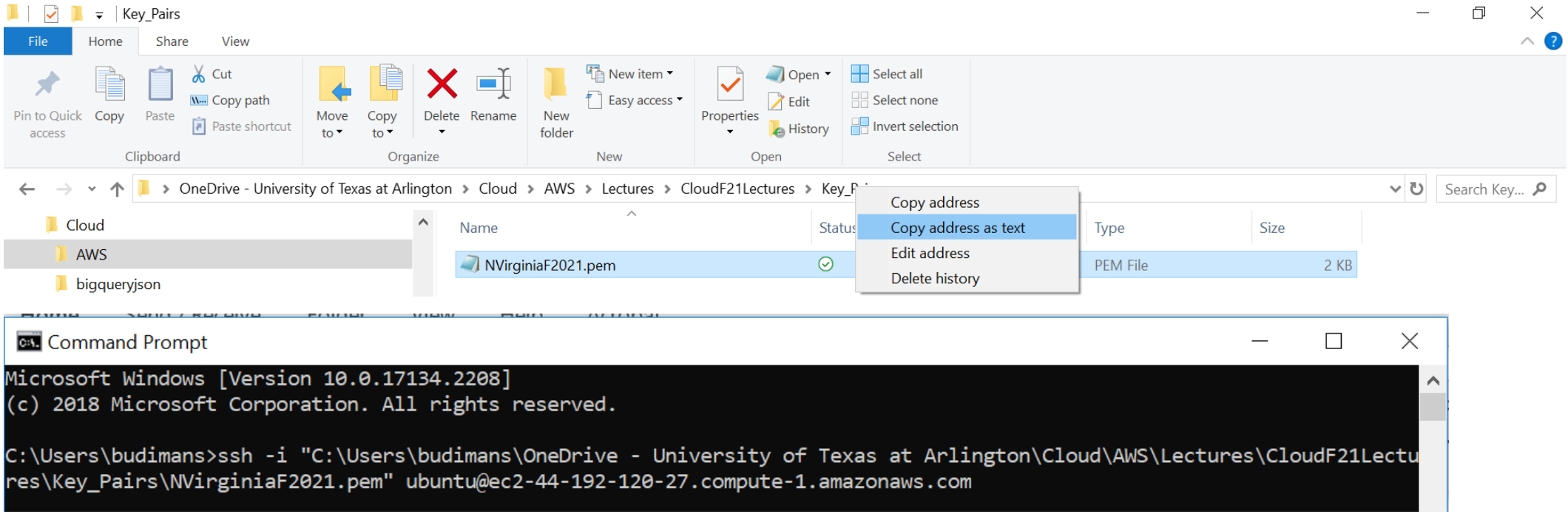
In this case, your laptop (the client) must have the private key since it will be authenticated by AWS (the server).

- The server uses the **public key** to encrypt a message and send it to the client.
- If the client has the correct **private key**, it can decrypt the message and send it back to the server for verification.

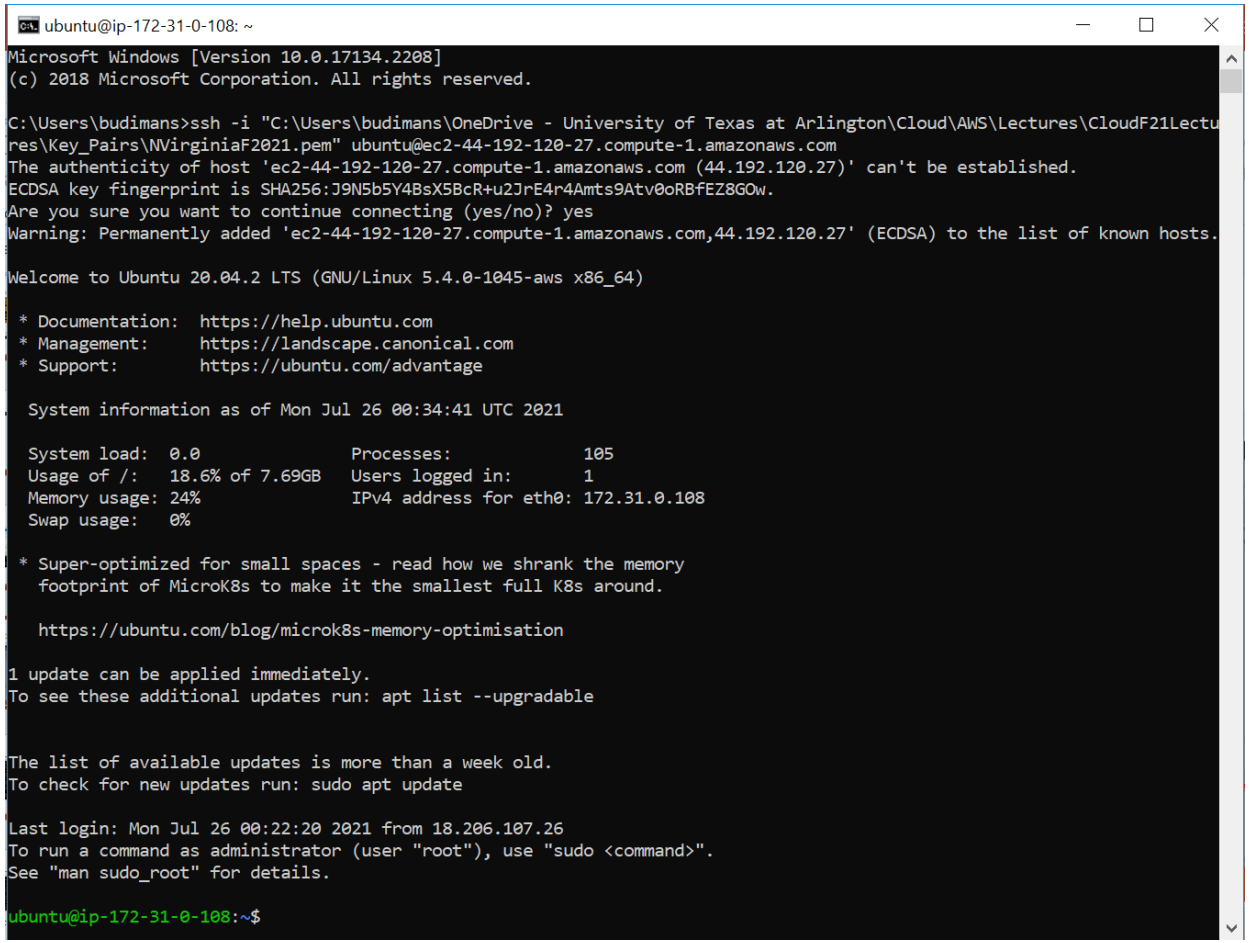
Run ssh command from your laptop's command prompt

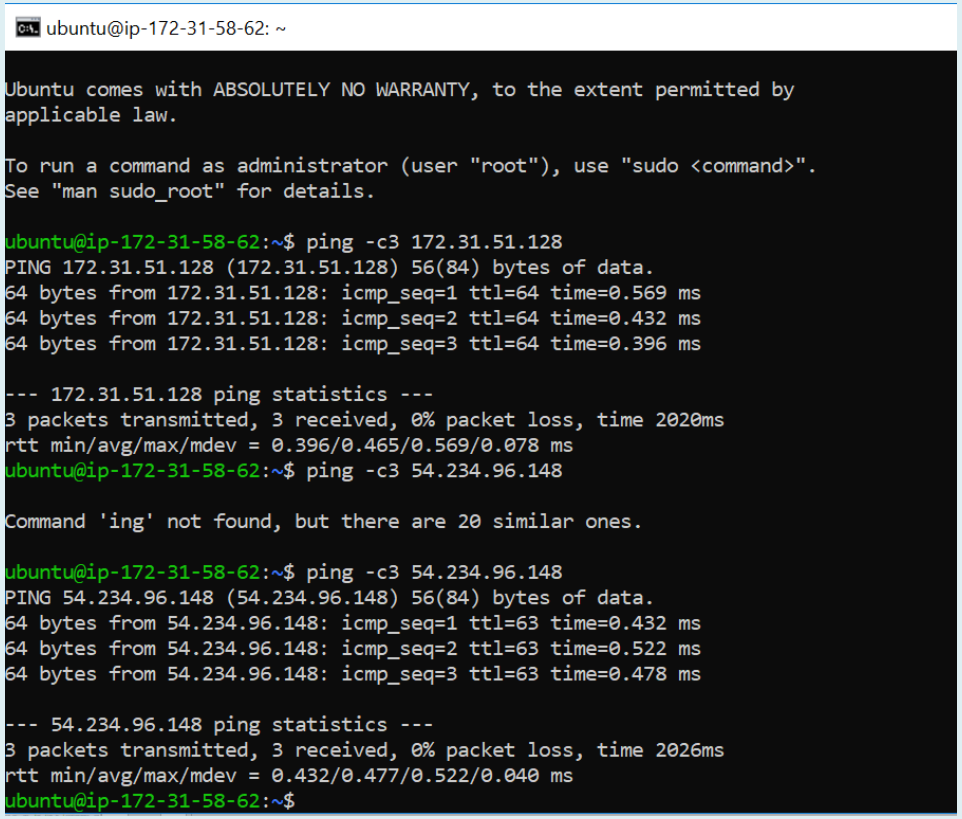



aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
EC2 > Instances > i-09182853c68ae0dae > Connect to instance
Connect to instance Info
Connect to your instance i-09182853c68ae0dae (Santoso1) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 Serial Console
Instance ID
i-09182853c68ae0dae (Santoso1)
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
Example:
ssh -i "NVirginiaF2021.pem" ubuntu@ec2-44-192-120-27.compute-1.amazonaws.com
For Windows users, copy this and paste on Command Prompt
Add directory path to the .pem file
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.



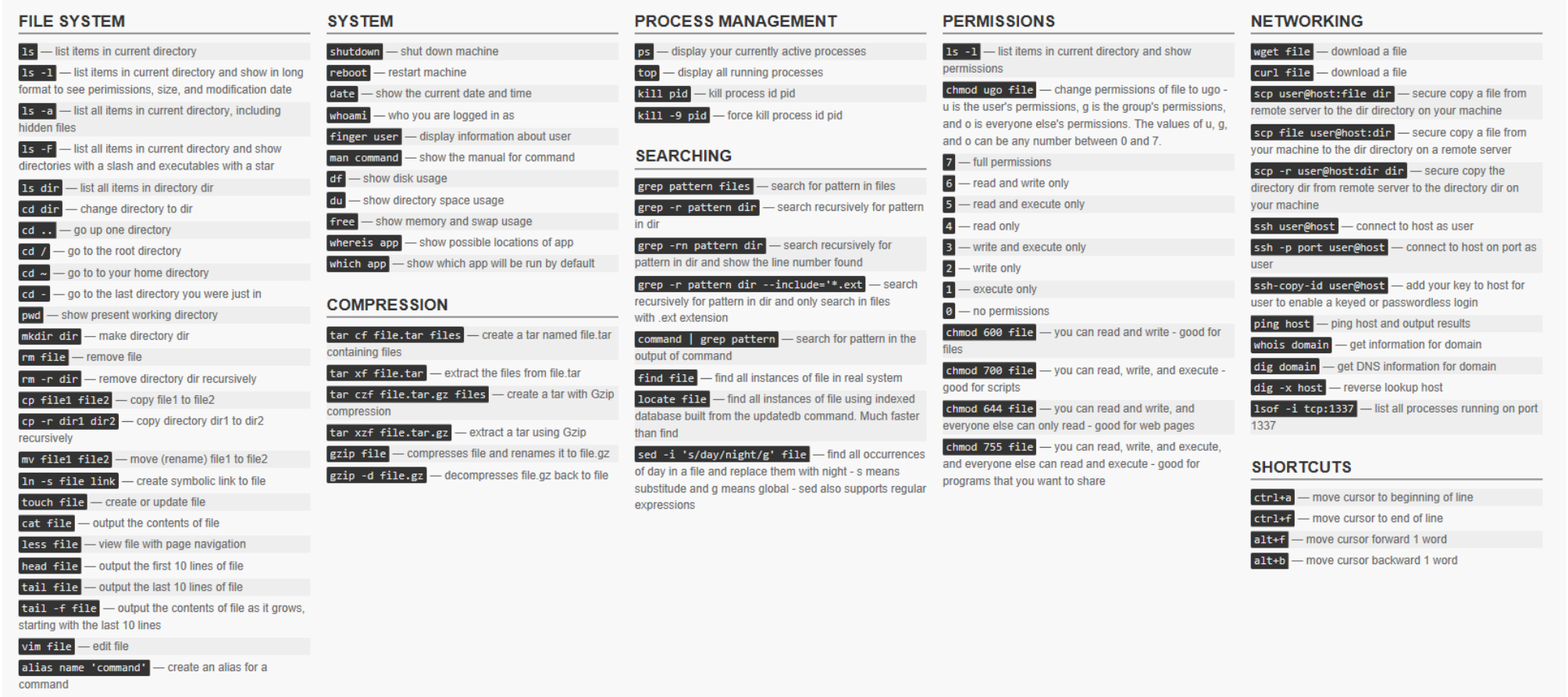
- Goto laptop's command prompt
- run the ssh command (cut and paste from example in the previous page).
- Make sure to include path to the .pem file.
- If using mac, may need to do chmod 400 …
- Sometimes use right click to paste


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+u2JrE4r4Amts9Atv0oRBfEZ8GOw.
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:~$



- 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

- `ls` — list items in current directory
- `ls -l` — list items in current directory and show in long format to see perimissions, size, and modification date
- `ls -a` — list all items in current directory, including hidden files
- `ls -F` — list all items in current directory and show directories with a slash and executables with a star
- `ls dir` — list all items in directory dir
- `cd dir` — change directory to dir
- `cd ..` — go up one directory
- `cd /` — go to the root directory
- `cd ~` — go to to your home directory
- `cd -` — go to the last directory you were just in
- `pwd` — show present working directory
- `mkdir dir` — make directory dir
- `rm file` — remove file
- `rm -r dir` — remove directory dir recursively
- `cp file1 file2` — copy file1 to file2
- `cp -r dir1 dir2` — copy directory dir1 to dir2 recursively
- `mv file1 file2` — move (rename) file1 to file2
- `ln -s file link` — create symbolic link to file
- `touch file` — create or update file
- `cat file` — output the contents of file
- `less file` — view file with page navigation
- `head file` — output the first 10 lines of file
- `tail file` — output the last 10 lines of file
- `tail -f file` — output the contents of file as it grows, starting with the last 10 lines
- `vim file` — edit file
- `alias name 'command'` — create an alias for a command

SYSTEM

- `shutdown` — shut down machine
- `reboot` — restart machine
- `date` — show the current date and time
- `whoami` — who you are logged in as
- `finger user` — display information about user
- `man command` — show the manual for command
- `df` — show disk usage
- `du` — show directory space usage
- `free` — show memory and swap usage
- `whereis app` — show possible locations of app
- `which app` — show which app will be run by default

COMPRESSION

- `tar cf file.tar files` — create a tar named file.tar containing files
- `tar xf file.tar` — extract the files from file.tar
- `tar czf file.tar.gz files` — create a tar with Gzip compression
- `tar xzf file.tar.gz` — extract a tar using Gzip
- `gzip file` — compresses file and renames it to file.gz
- `gzip -d file.gz` — decompresses file.gz back to file

PROCESS MANAGEMENT

- `ps` — display your currently active processes
- `top` — display all running processes
- `kill pid` — kill process id pid
- `kill -9 pid` — force kill process id pid

SEARCHING

- `grep pattern files` — search for pattern in files
- `grep -r pattern dir` — search recursively for pattern in dir
- `grep -rn pattern dir` — search recursively for pattern in dir and show the line number found
- `grep -r pattern dir --include='*.ext` — search recursively for pattern in dir and only search in files with .ext extension
- `command | grep pattern` — search for pattern in the output of command
- `find file` — find all instances of file in real system
- `locate file` — find all instances of file using indexed database built from the updatedb command. Much faster than find
- `sed -i 's/day/night/g' file` — find all occurrences of day in a file and replace them with night - s means substitude and g means global - sed also supports regular expressions

PERMISSIONS

- `ls -l` — list items in current directory and show permissions
- `chmod ugo file` — 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.
- `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
- `chmod 600 file` — you can read and write - good for files
- `chmod 700 file` — you can read, write, and execute - good for scripts
- `chmod 644 file` — you can read and write, and everyone else can only read - good for web pages
- `chmod 755 file` — you can read, write, and execute, and everyone else can read and execute - good for programs that you want to share

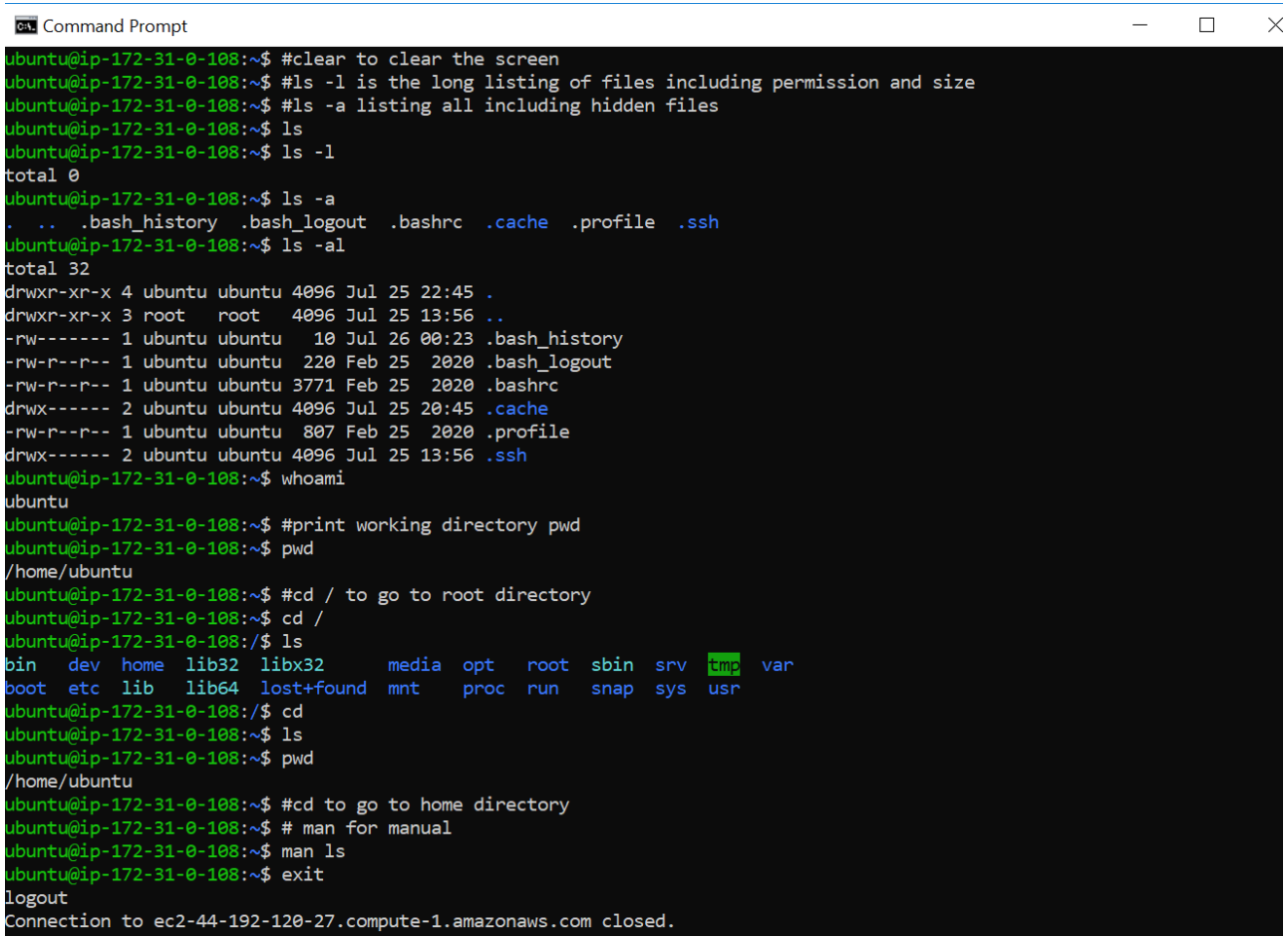
NETWORKING

- `wget file` — download a file
- `curl file` — download a file
- `scp user@host:file dir` — secure copy a file from remote server to the dir directory on your machine
- `scp file user@host:dir` — secure copy a file from your machine to the dir directory on a remote server
- `scp -r user@host:dir dir` — secure copy the directory dir from remote server to the directory dir on your machine
- `ssh user@host` — connect to host as user
- `ssh -p port user@host` — connect to host on port as user
- `ssh-copy-id user@host` — add your key to host for user to enable a keyed or passwordless login
- `ping host` — ping host and output results
- `whois domain` — get information for domain
- `dig domain` — get DNS information for domain
- `dig -x host` — reverse lookup host
- `lsof -i tcp:1337` — list all processes running on port 1337

SHORTCUTS

- `ctrl+a` — move cursor to beginning of line
- `ctrl+f` — move cursor to end of line
- `alt+f` — move cursor forward 1 word
- `alt+b` — move cursor backward 1 word

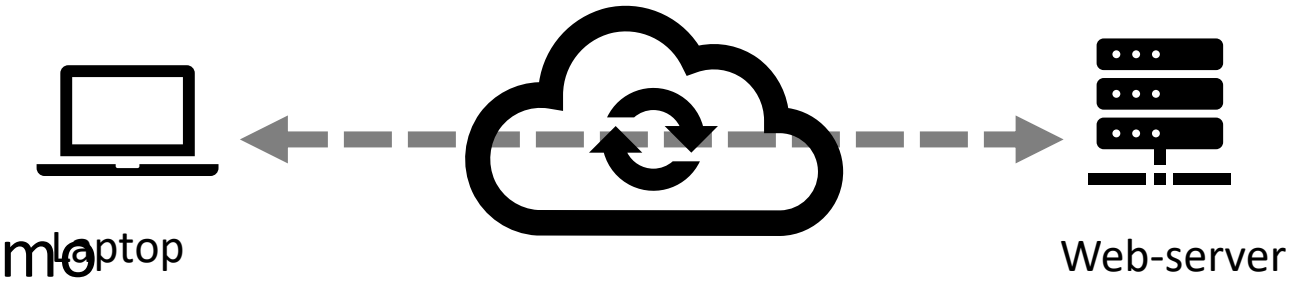
http://cheatsheetworld.com/programming/unix-linux-cheat-sheet/


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.

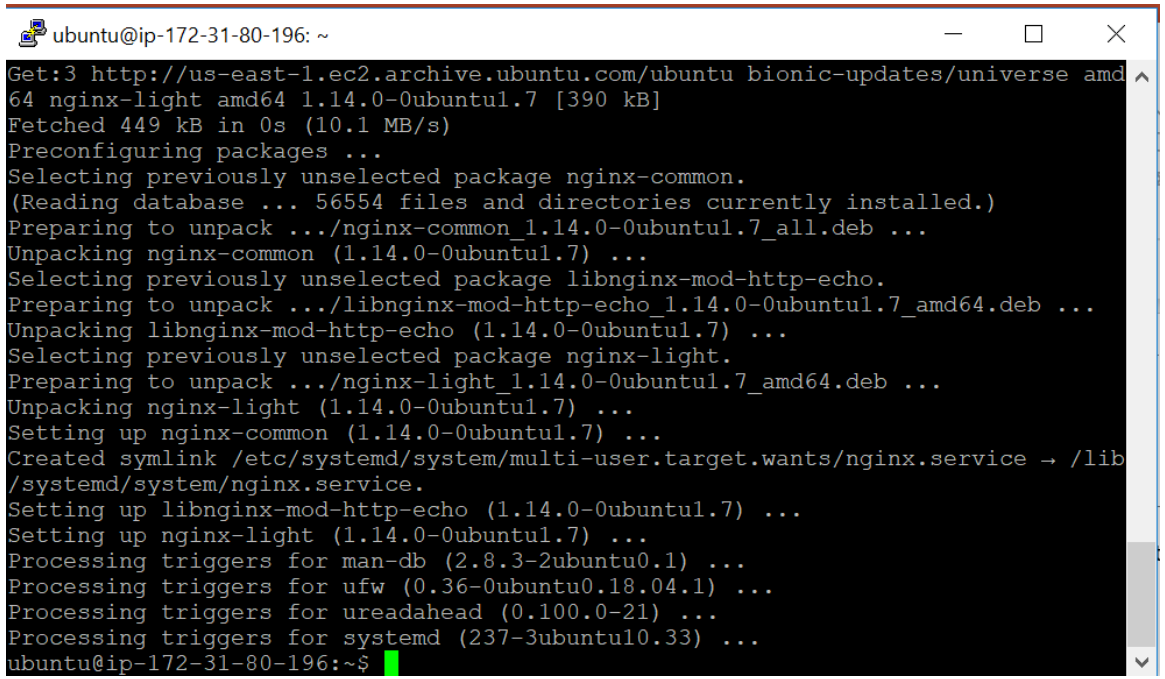
Installing a Web-Server in EC2

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



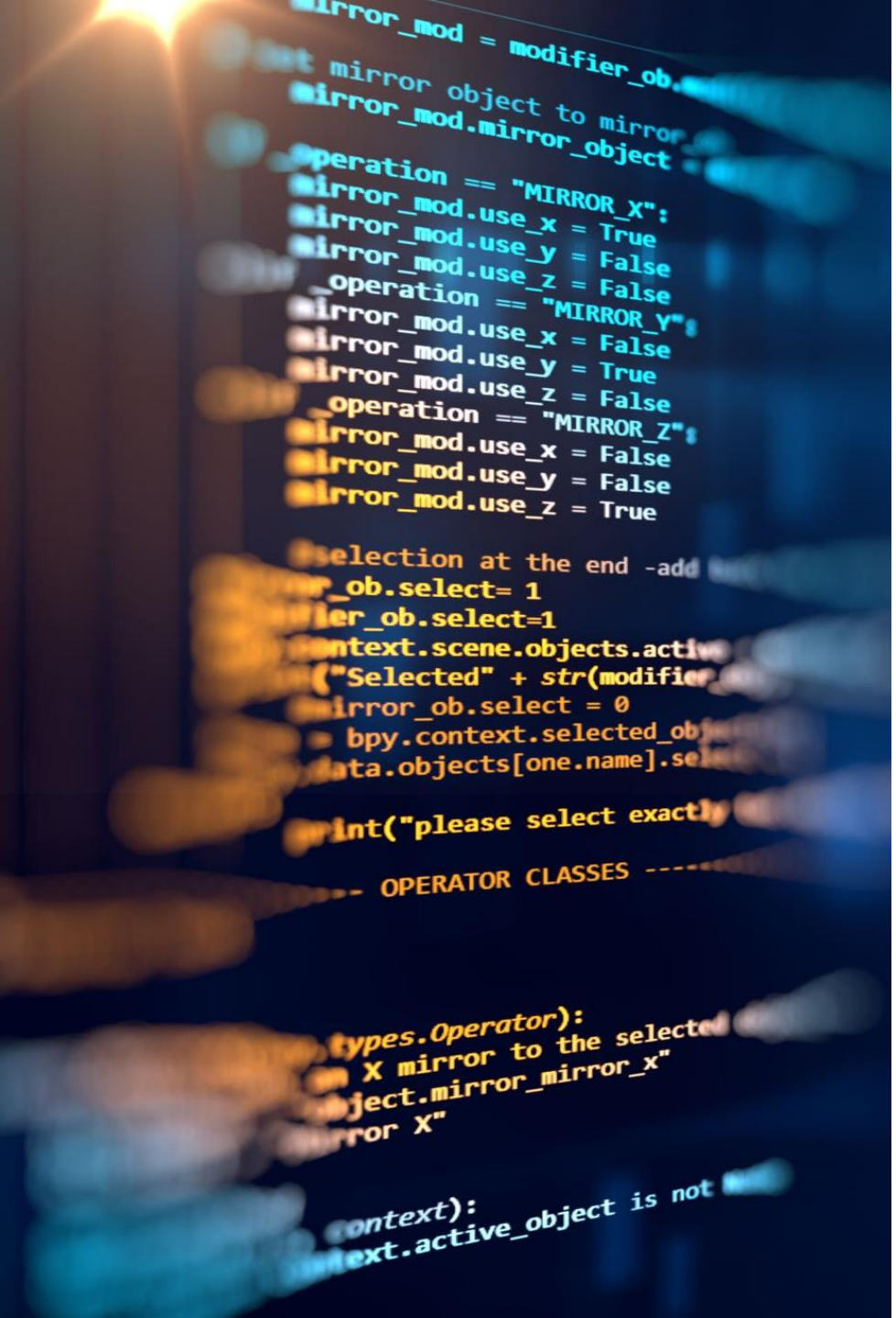
Nginx is an open-source web server. We will use this to do our homework.



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



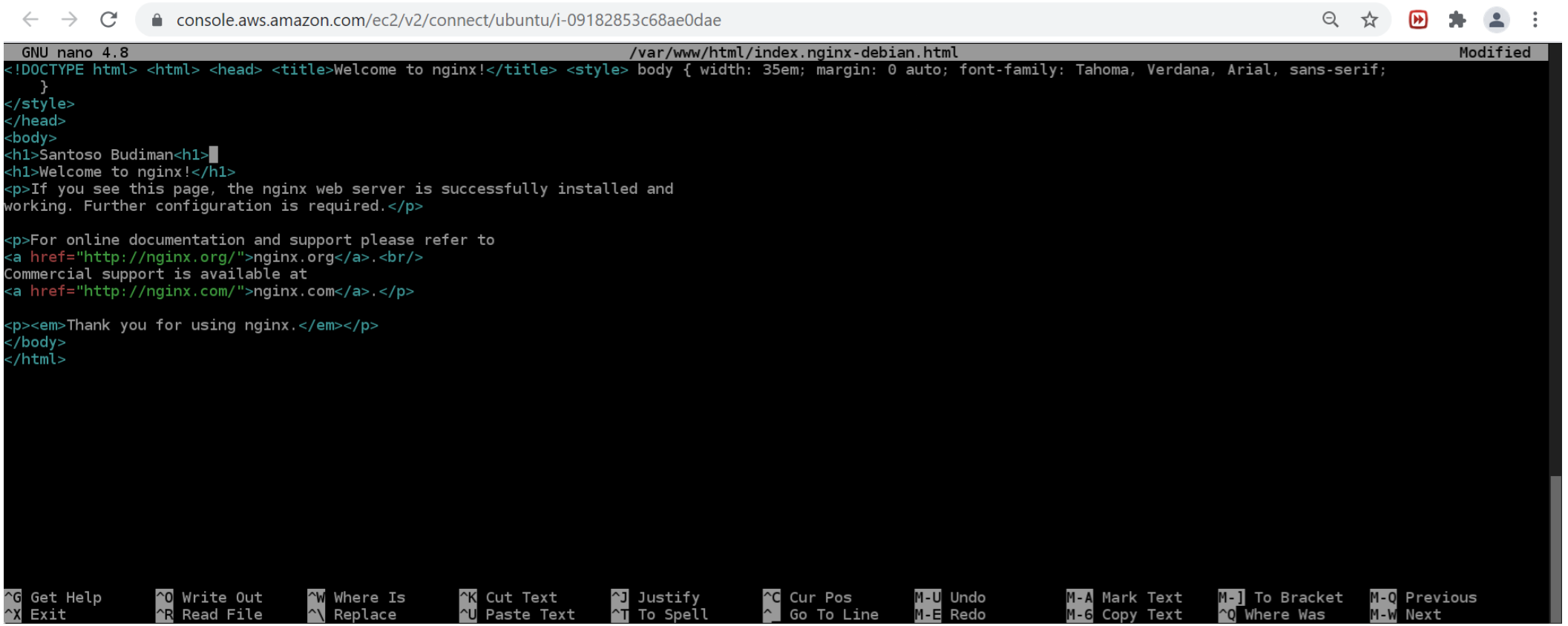
Edit nginx webserver

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.

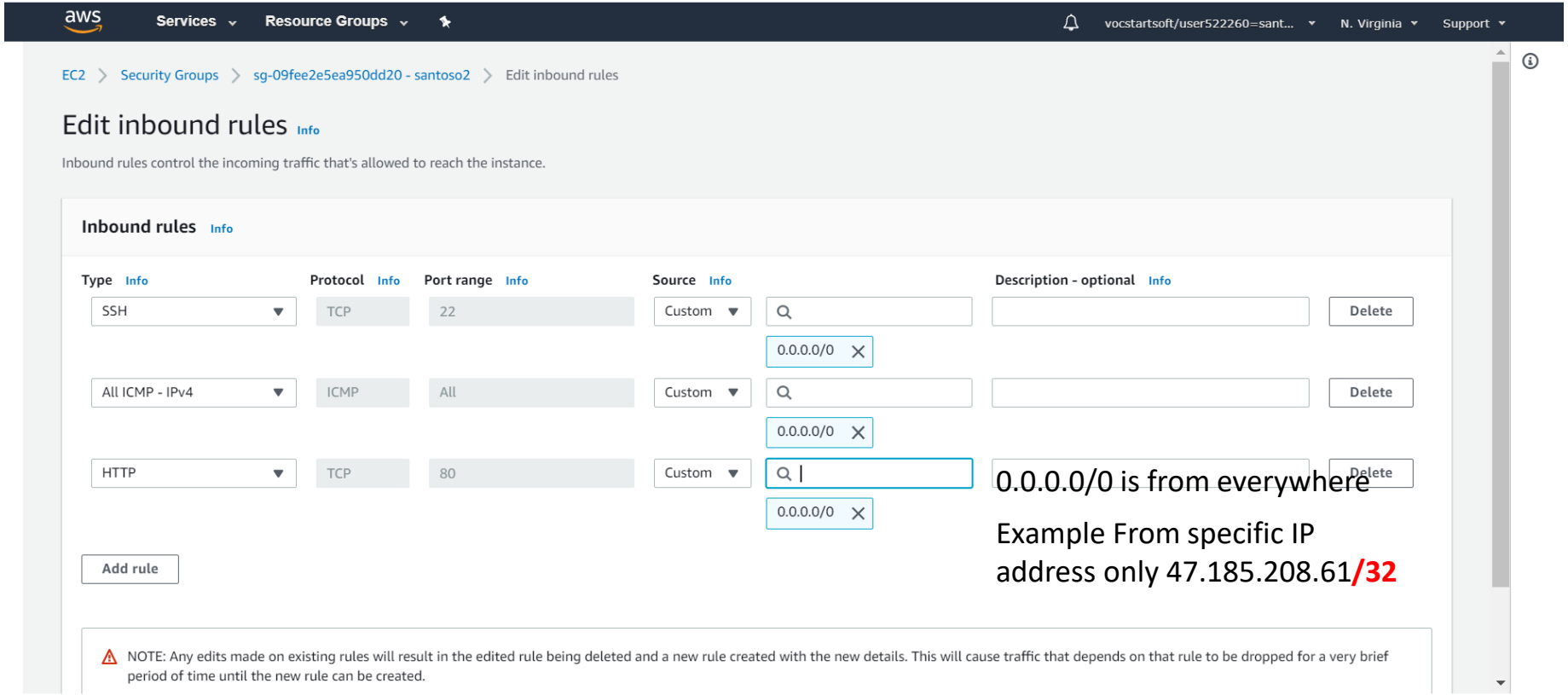
Edit using nano editor:

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

- add above <h1>Welcome nginx…:
- <h1>This is the 2nd 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>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
For online documentation and support please refer to
nginx.org.
Commercial support is available at
nginx.com.
Thank you for using nginx.
</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

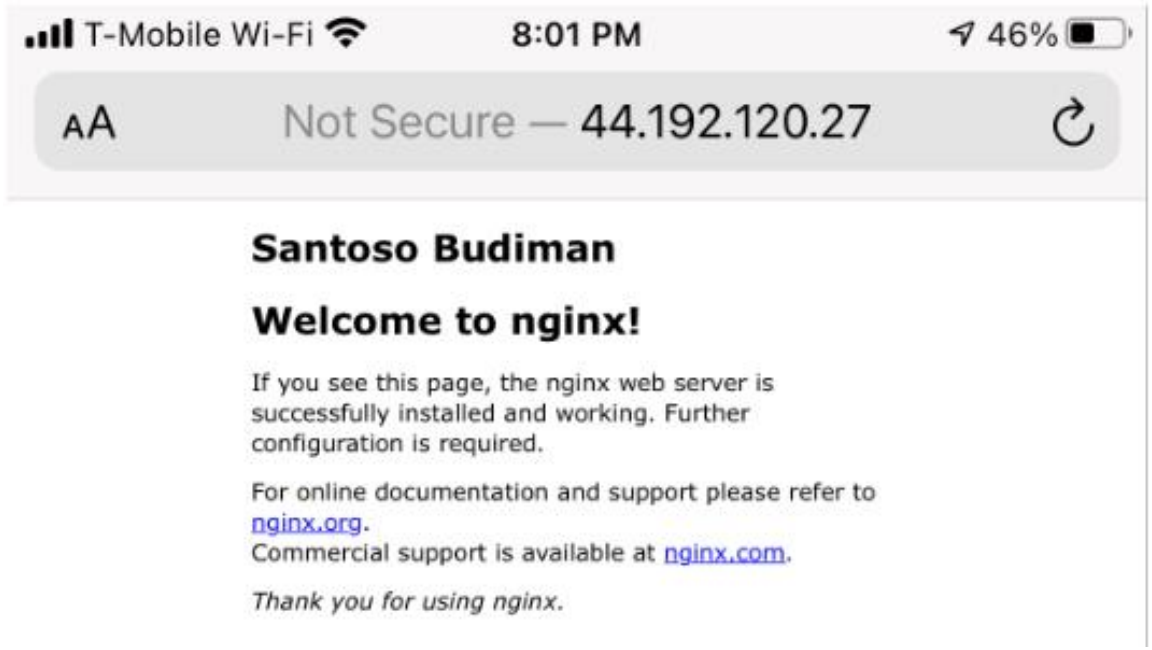
Update Security group on the EC2 to allow http

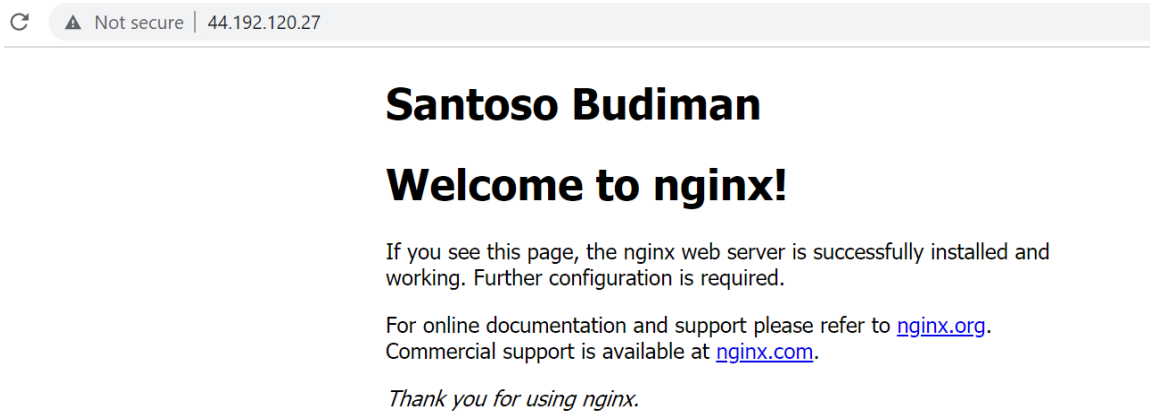


0.0.0.0/0 is from everywhere

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

Test the website





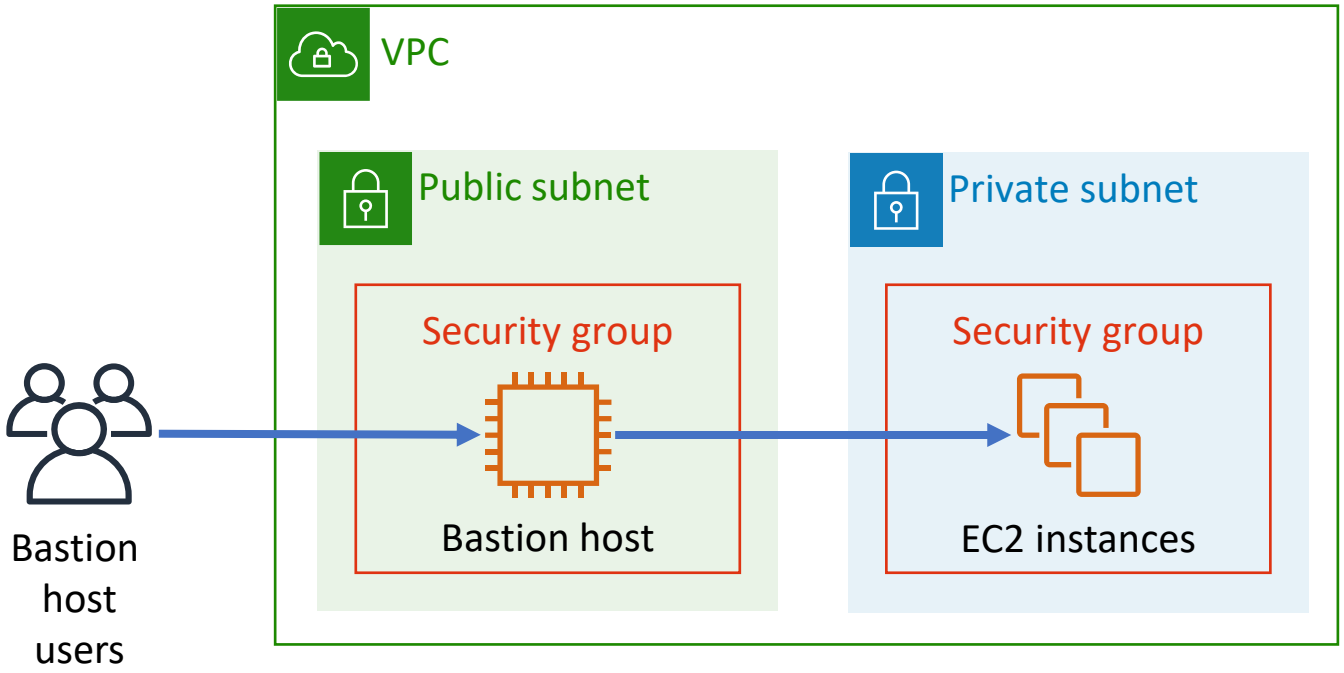
Bastion Host

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

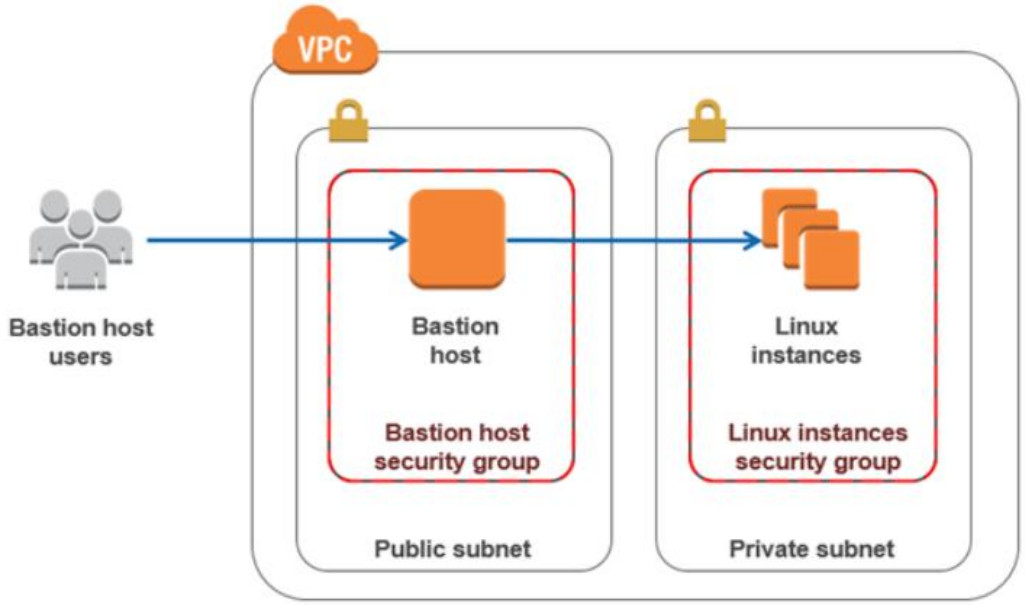
- A server whose purpose is to provide access to a private network from an external network
- Must minimize the chances of penetration

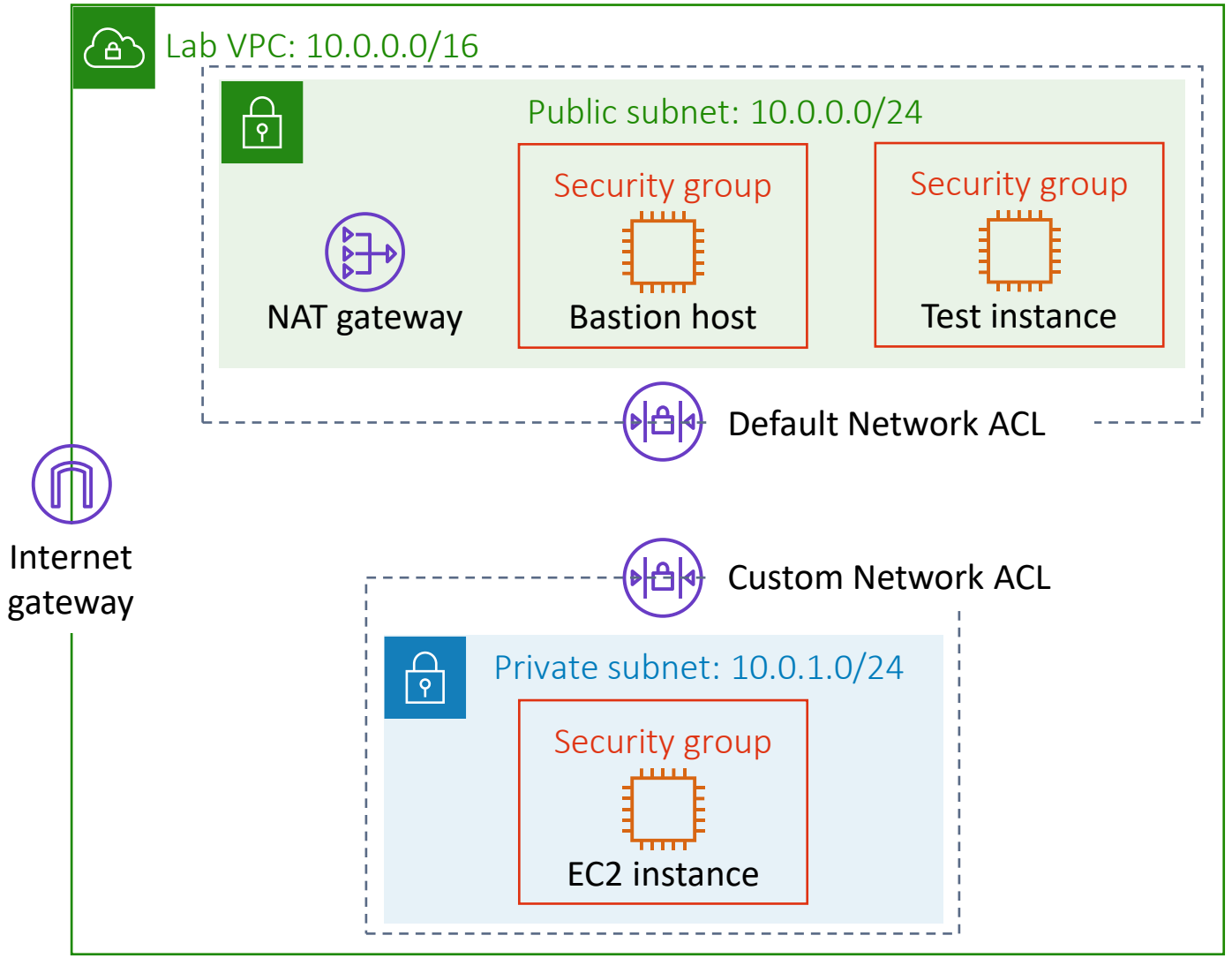


AWS Bastion Host

- 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/




Lab VPC: 10.0.0.0/16
Public subnet: 10.0.0.0/24
Security group
Bastion host
Security group
Test instance
NAT gateway
Default Network ACL
Internet gateway
Custom Network ACL
Private subnet: 10.0.1.0/24
Security group
EC2 instance

SSH from EC2 to EC2 (if the 2nd 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
```

ubuntu@ip-172-31-93-3: ~

```
 GNU nano 2.9.3                                  AWSSantoso.pem

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAgtB3X9KMjVeLea4L4DH+wU/a0FjOSP75ZYEGJraB5IRIeH0UAEoh+evaGA5n
Yl0TTOzXzRZtKH8zGvltMCvmj3gtKoq88Kea08vUXJffoj8BqOEmGZsa/XWGsDpNLSE6443tiSZi
+xKkLW/RjIvi0u2GYswBiK5G6+1yoZHWCsCeyqTSs1PBdXUODcvrHBgsJmVG+l1g5kwkziu618ws
tluthoL57IOmQp3wPsg0dqlmD2v3w+JI2aNde28IqUaQfQSFYXEB8uHxQZhJt1lOrHEYZKul6/sV
vOMkrmZYYxPap9aNI4G/kbMe7oN3sXwdYcI5EYoVACEOByFH+nuv4wIDAQABAoIBABvUrEEHs31y
rarSm5Y0LMYF7kC7VA/1pyuwOpXL6HCssBpqSfIR046vXfQWRGqJ8MxB61untMuT3FSV0mQYQvIt
GDbf03jDv83lpKWabiq/iuPaGtgBxsk70s71aGv8kOHO1UWDlDVkbud3zbZAIJ7Nh/THAGCK623Q
U74KH+ajbBXyqnjX2Nja0cWA+KgfJt/YibLwcF7+arMWHrAiiDKPreq69EFOgtfJM/9LZC+RBcFH
oO0Fjjg7NoMXBxgNi/BgYLKUSudqrc5PcIl7bu3lzbaoZREdqHlQTZksEaGXg4b1fbV8xupVlACa
rQyKSqgKxdDC2Ok4F221h9k7jlkCgYEAy5G4pdYv2EQUkjR/xk5IlyN9l4GivfvGX+NM3w3uJP3a
mGr1HZtlnawydpc2hLgAO10+gZjzvTXwgoQ3Z0N51B8DtX0F3k6G4kTV98AzDaigpcOIGrE4IBA/
RJOndAoh3rlHD4uIl7yumLuxgRn4jID0HrGRvx0IfLtBHILI9K0CgYEApIGr2c9h5ssKsTLKhBMg
CafK9axbLLQWb9zPSwL23aMGVX2CdYMGfiA226wGByjkAZTUnIwo9suIbdNUvy1GQ46GKaz9Cn5W
vWAQ+q5jL08wWgSYczRlQWAGqAXbibVCTxibC6FwN0C6vuRtR+NWHsL/iDKIum2D9uN13u1YOM8C
gYAYqoaKKaLQfKdVc+ELoU6waY99PSsABDLsSXl2CZUSjGPhVgxvUvYEHepbbkUf3b/BcbHyOBtI
kPGcjvK7o3RunHdxlT2Ry0gSwK6Af5xwCj+09jVWrqSTVgSkhZOjvtqNxztyCcE6SZWKUivW7dOt
Cq8JzRBQ7O6rVnE0SDCbXQKBgQCb4lyMOx3Ua1aoby7EVLMpMxz6C+chx0zzl1O/CwIPJ+panGJB
JHXEkpYpNytHIZJdhUCrT42Jv3h6SaWmuyo33hh36XqmtO1Q6nVsdk96hxmUU4J6uI2yk2H7WgGc
zmwVM4XIUaRaVzYeqap8UopLABDQaMJzlk9UZlpNcq8EtQKBgBPwjtZfQxD/mYdaUCQbLrsnAlOF
Ub31JCrSzJoOnVFNGkBZba8XytIqD78mVB2A4dVlpd+5xx8eZSViw/66FjVn7s9zdLM/r7RMruPs
+5LtSCeIVrRqOe766VkfqaZqgz74fAmQHe8X+uN59Rk5UQYgAKChPPN02Nkxp80ULHRO
-----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:~$
```

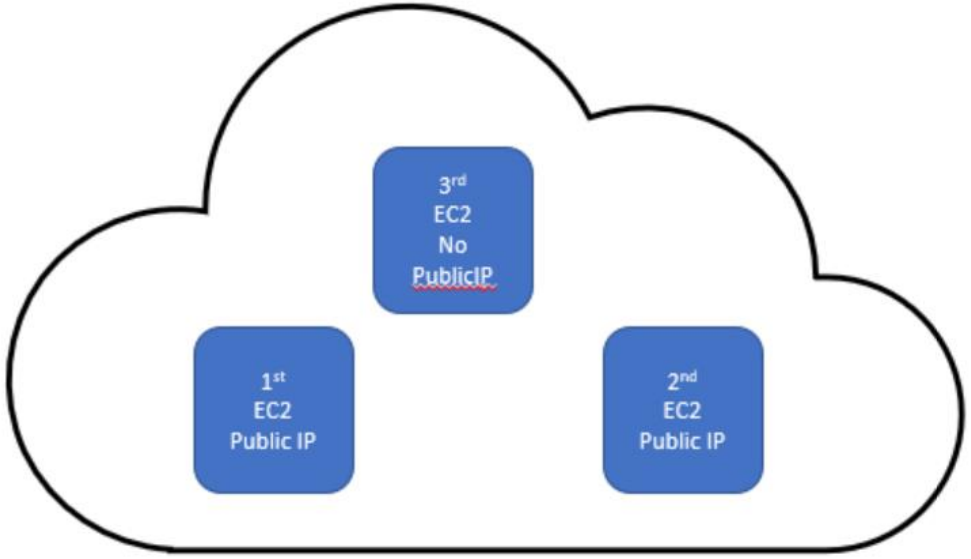
SSh from EC2a to EC2b

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:

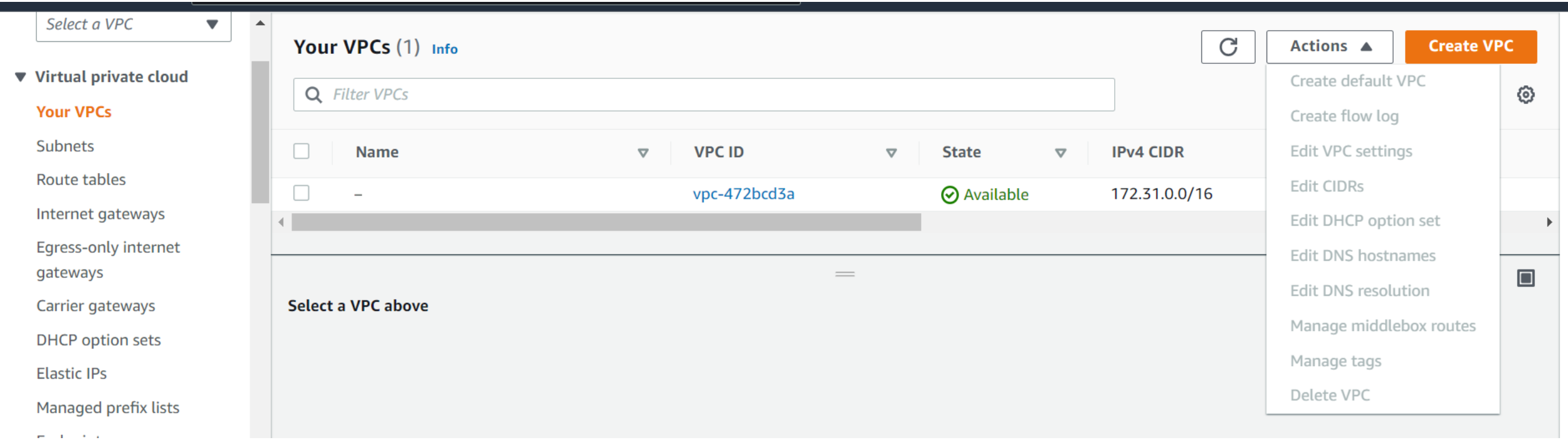
1. Each EC2 has an internal IP address but not always has public IP.
2. You can ping with internal/external IP address depending on the Security Group.
3. Ping and ssh must use public IP address if sent from different network.
4. How do you ssh to the 3rd EC2 (no public IP)?
5. Show the 3 EC2 instances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

Spin up 3EC2s:

- 1st EC2 – web server
  - Has public IP
  - Allow SSH from everywhere
  - Allow ping from everywhere
  - Allow http call from everywhere
  - Bastion host for 3rd EC2
  - Install nginx and edit the html file
- 2nd EC2- general purpose EC2
  - Has public ip
  - Allow SSH from everywhere
  - Allow ping from everywhere
- 3rd EC2 – internal use only
  - No public IP
  - Only allow ssh from 1st EC2
  - Only allow ping from 1st 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)



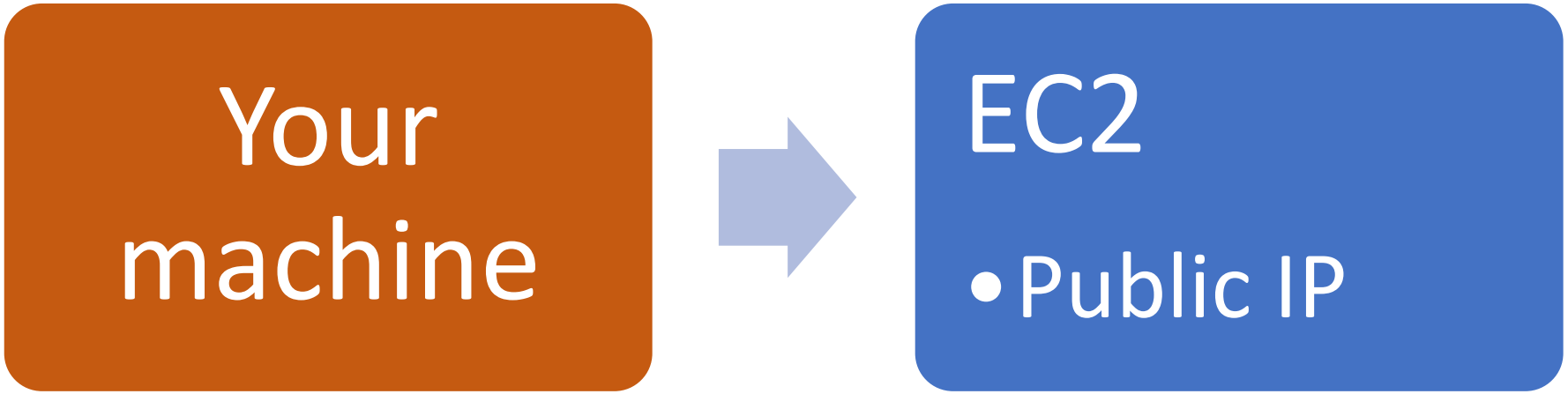
DO NOT DELETE DEFAULT VPC BUT THERE IS A “CREATE DEFAULT VPC” UNDER ACTION. I NEVER TRIED THIS AND HOPEFULLY YOU DON’T NEED TO DO THIS EITHER.

End Of Lecture

WINSCP

Transferring Files

Connection and File Transfer

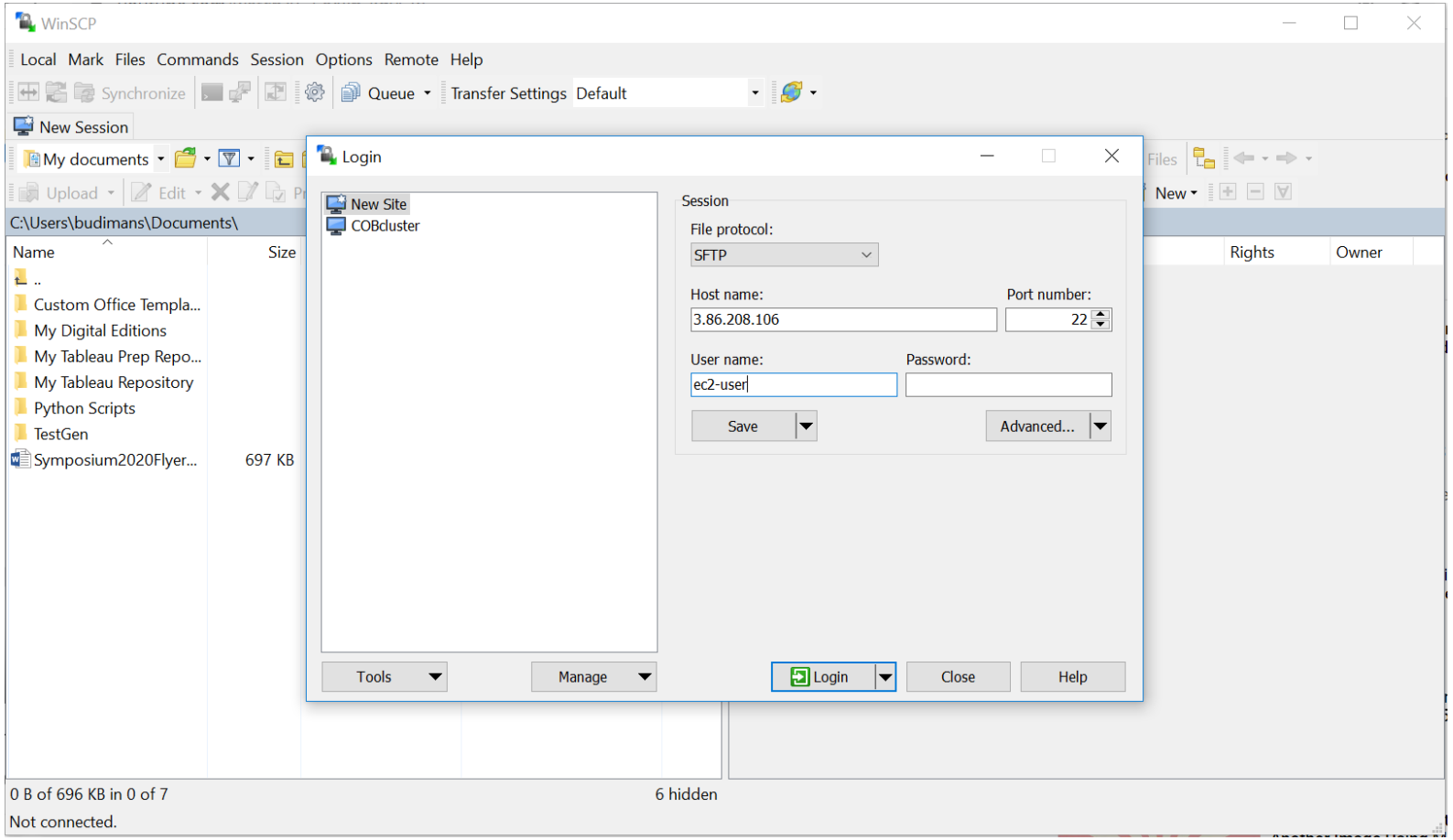


Putty - ssh
WinSCP – transfer files

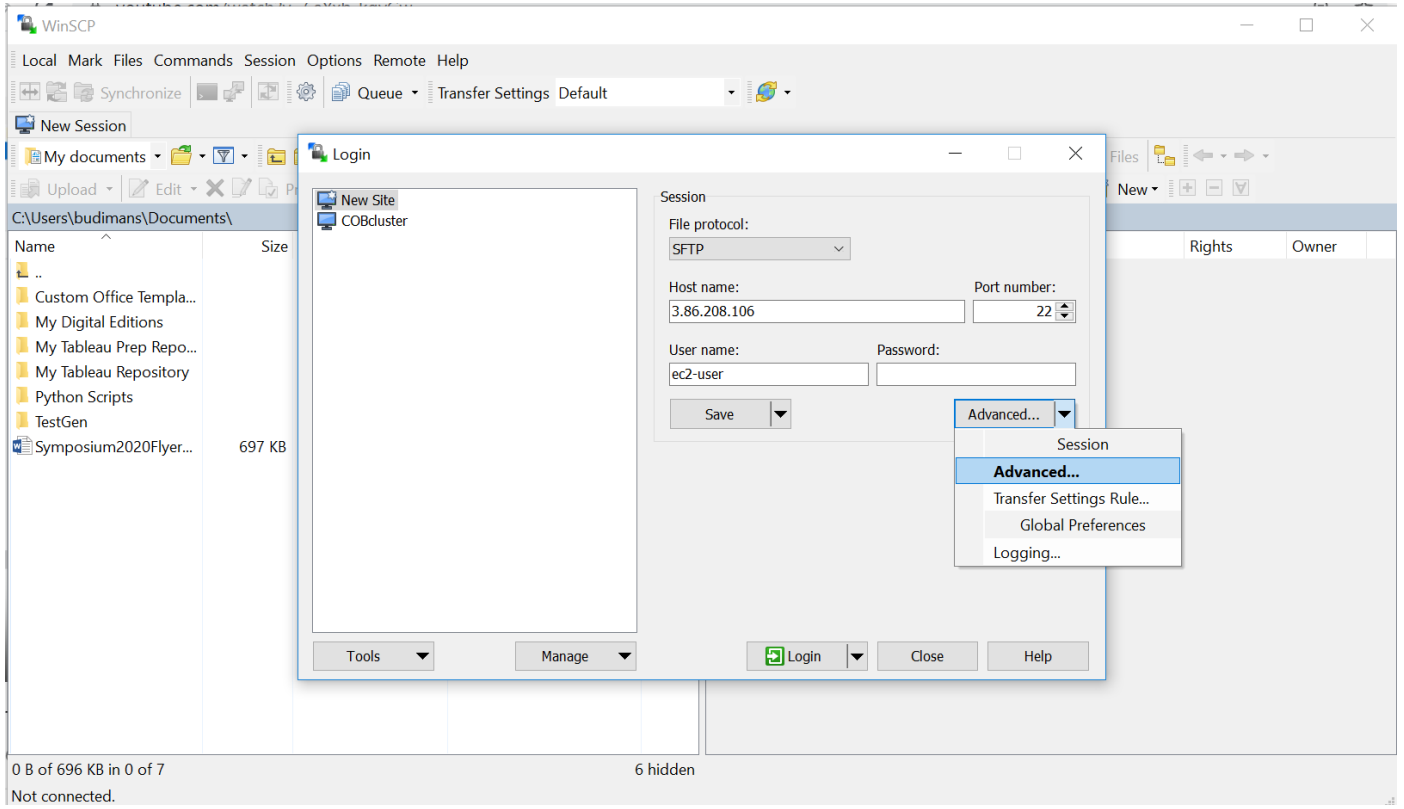
WinSCP vs PuTTY

- **WinSCP** (***Win****dows* ***S****ecure* ***Co****py*) 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

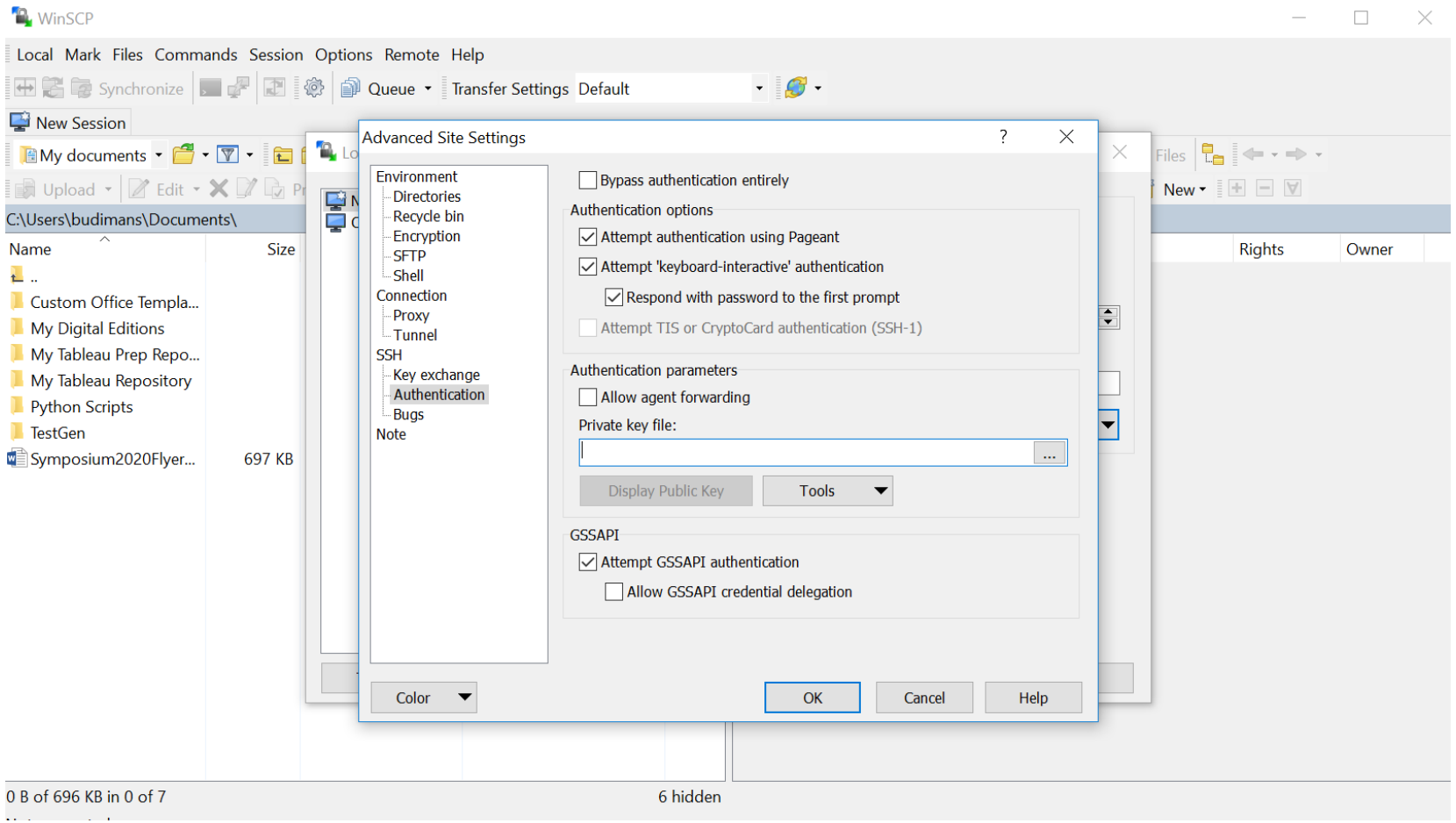


File Transfer using WinSCP

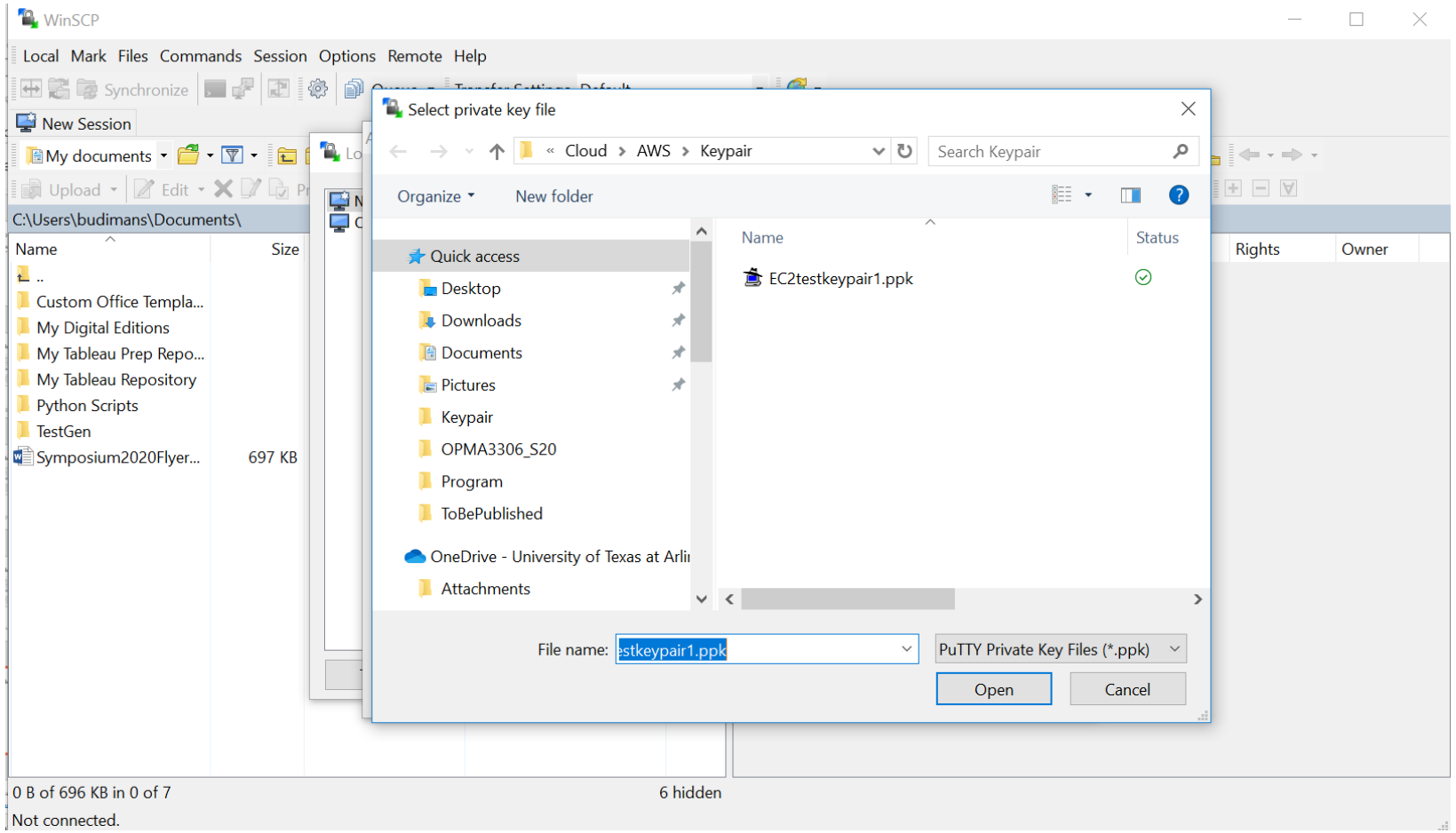


For Amazon Linux, the user name is ec2-user.
For RHEL 5, the user name is either root or ec2-user.
For Ubuntu, the user name is ubuntu

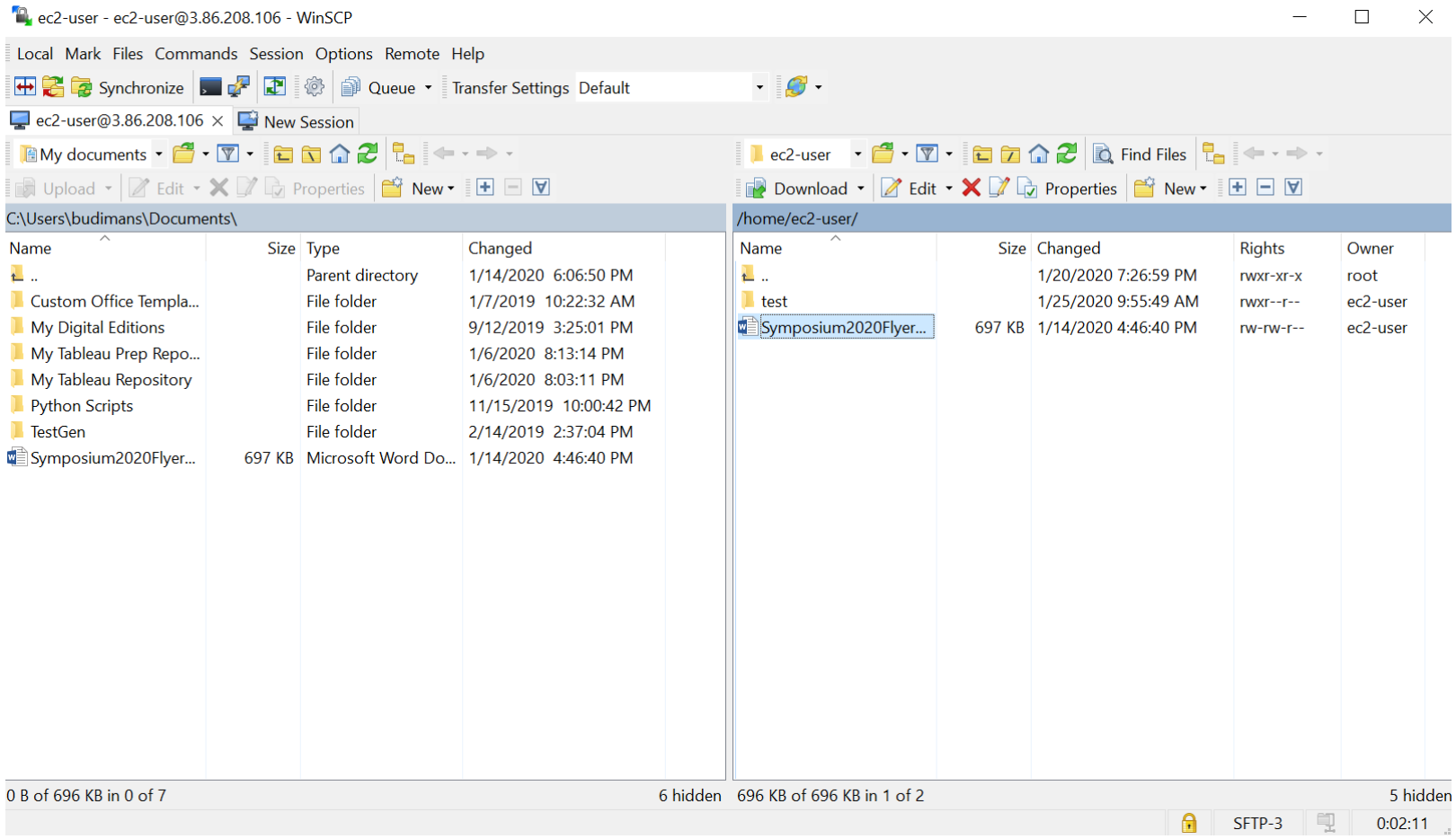
File Transfer using WinSCP



File Transfer using WinSCP



File Transfer using WinSCP



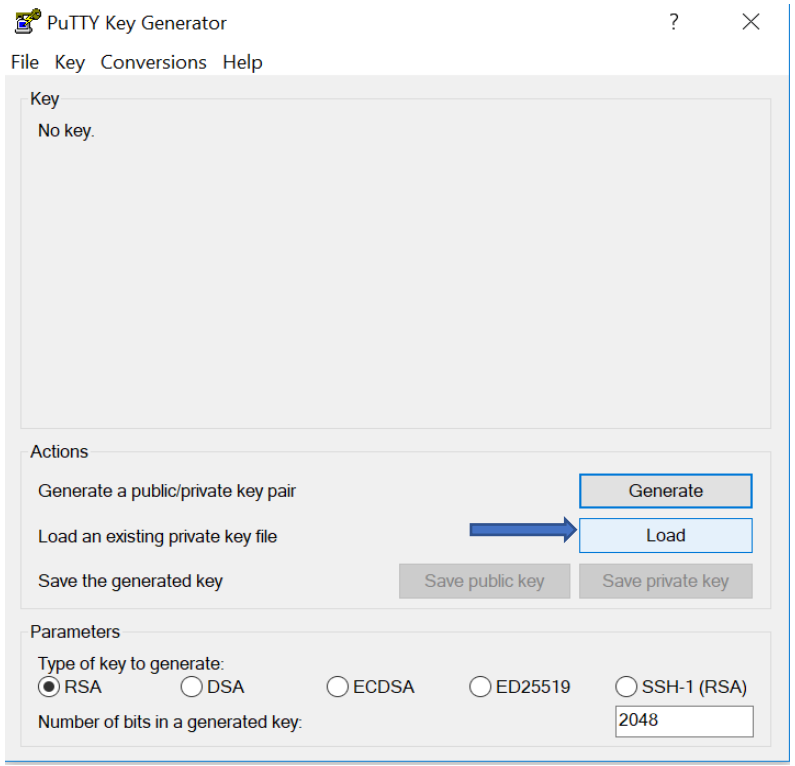
SSH using Putty

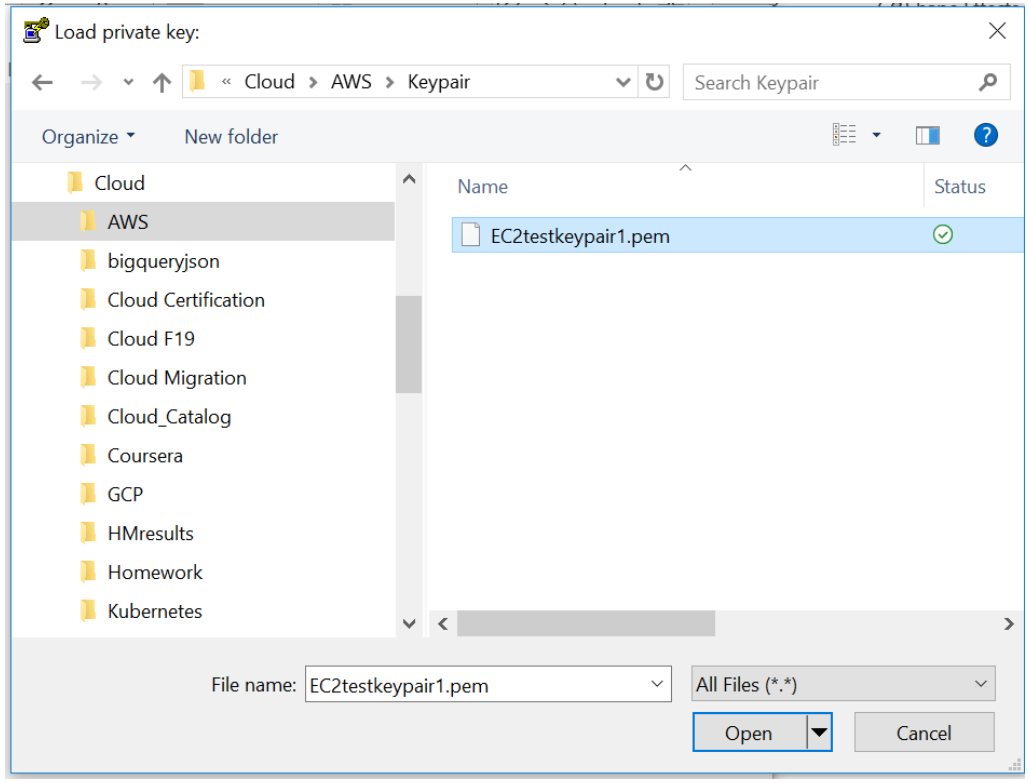
PuTTy Information

PuTTY

- 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=1cc0119ddbebfb800ed7dbf0ce961911&pageid=utassp01_search

SSH - Convert PEM to PPK file (Key Pair)



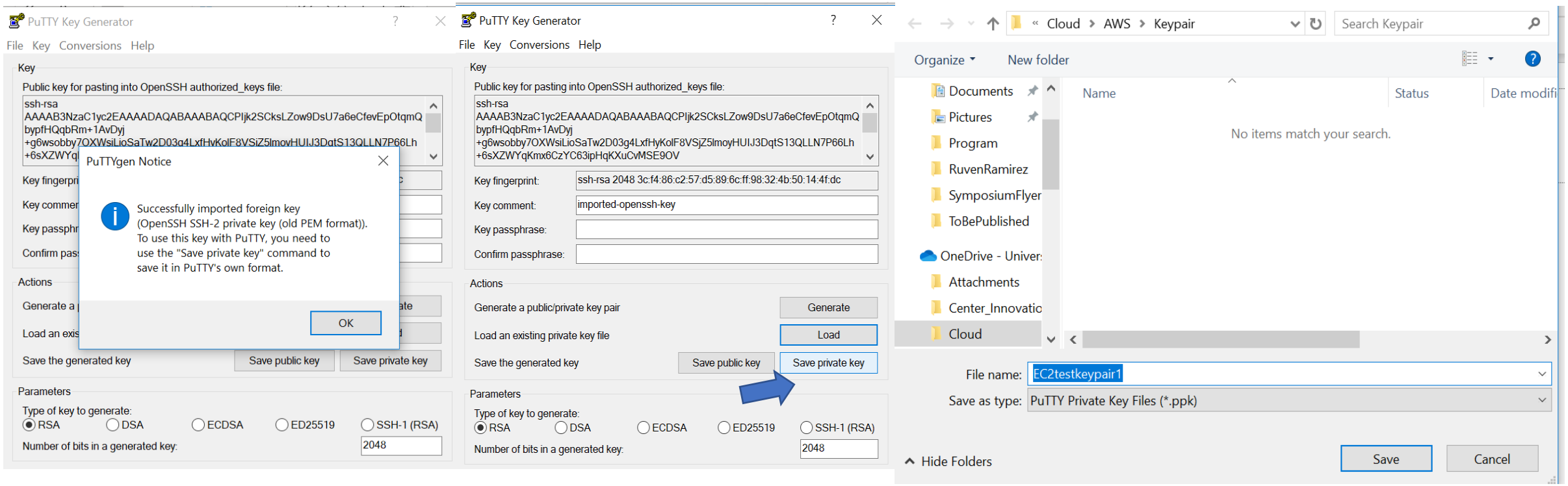


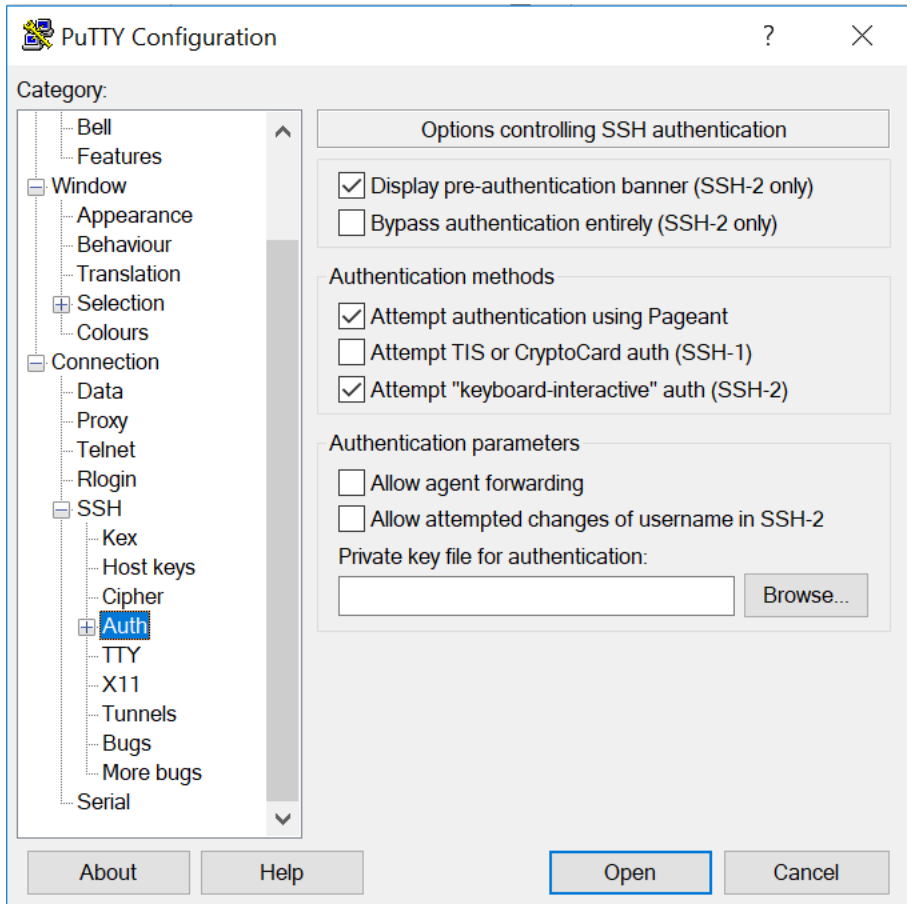
.pem file created in AWS must be saved in a directory
The .pem created in AWS needs to be converted to .ppk

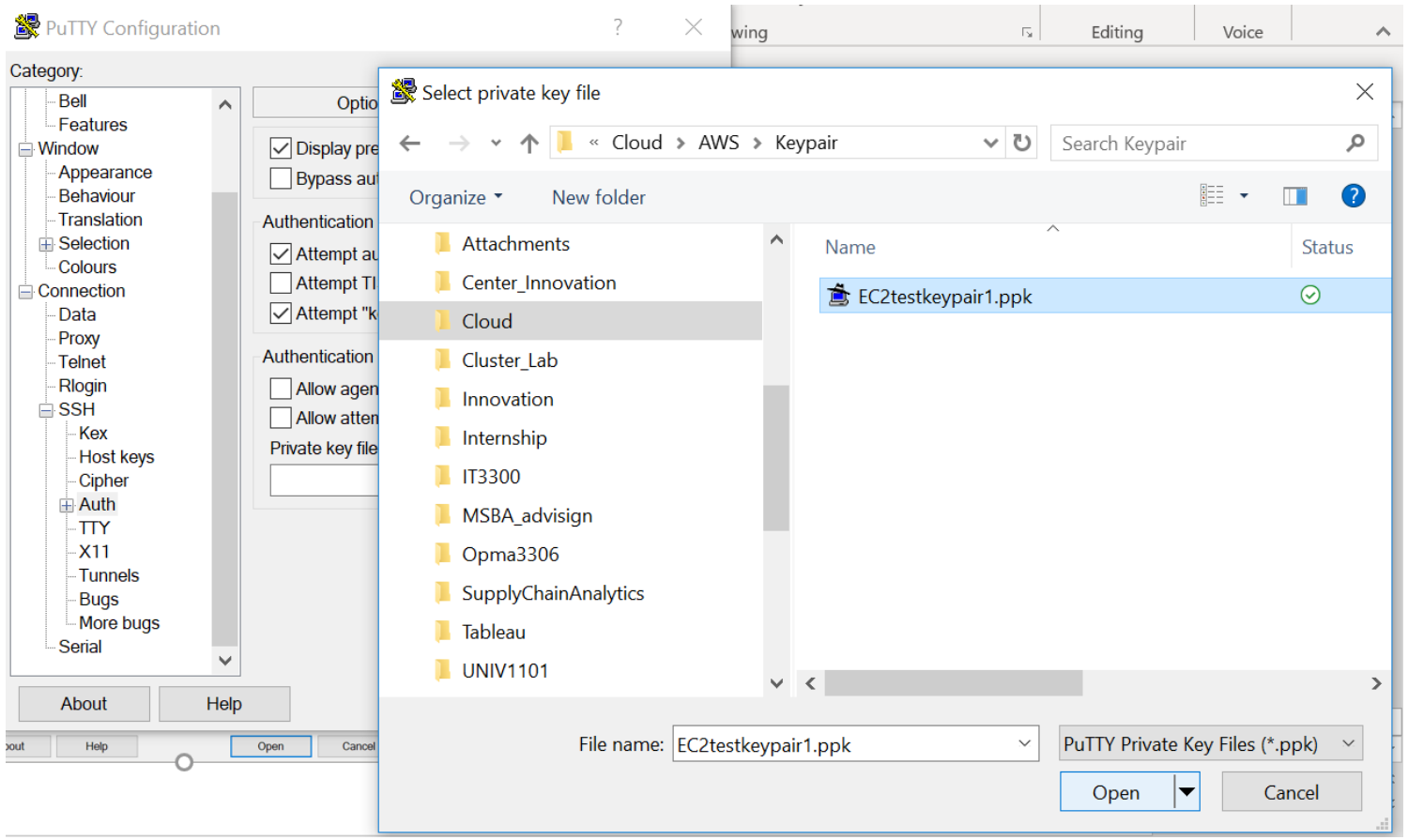
The SAME Key Pair can be used for all instances

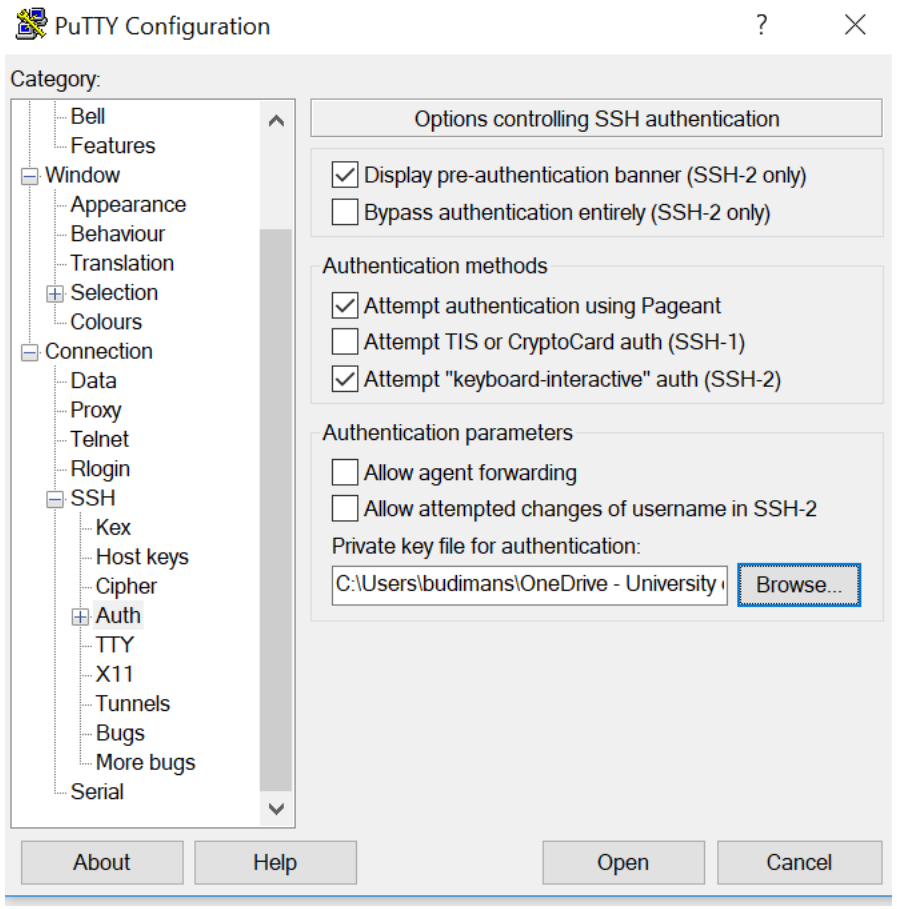
.pem vs .ppk

- 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.


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCPIjk2SCksLZow9DsU7a6eCfevEpOtqmQ
bypfHQqbRm+1AvDyj
+g6wsobby7OXWsiLioSaTw2D03g4LxfHvKolF8VSjZ5lmoyHUIJ3DqtS13QLLN7P66Lh
+6sXZWYqKmx6CzYC63ipHqKXuCvMSE9OV
Key fingerprint:
ssh-rsa 2048 3c:f4:86:c2:57:d5:89:6c:ff:98:32:4b:50:14:4f:dc
Key comment:
imported-openssh-key
Key passphrase:
Confirm passphrase:
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA DSA ECDSA ED25519 SSH-1 (RSA)
Number of bits in a generated key:
2048
PuTTYgen Notice
Successfully imported foreign key
(OpenSSH SSH-2 private key (old PEM format)).
To use this key with PuTTY, you need to
use the "Save private key" command to
save it in PuTTY's own format.
OK
Cloud › AWS › Keypair
Search Keypair
Organize
New folder
Documents
Pictures
Program
RuvenRamirez
SymposiumFlyer
ToBePublished
OneDrive - Univer
Attachments
Center_Innovatio
Cloud
Name
Status
Date modifi
No items match your search.
File name:
EC2testkeypair1
Save as type:
PuTTY Private Key Files (*.ppk)
Hide Folders
Save
Cancel

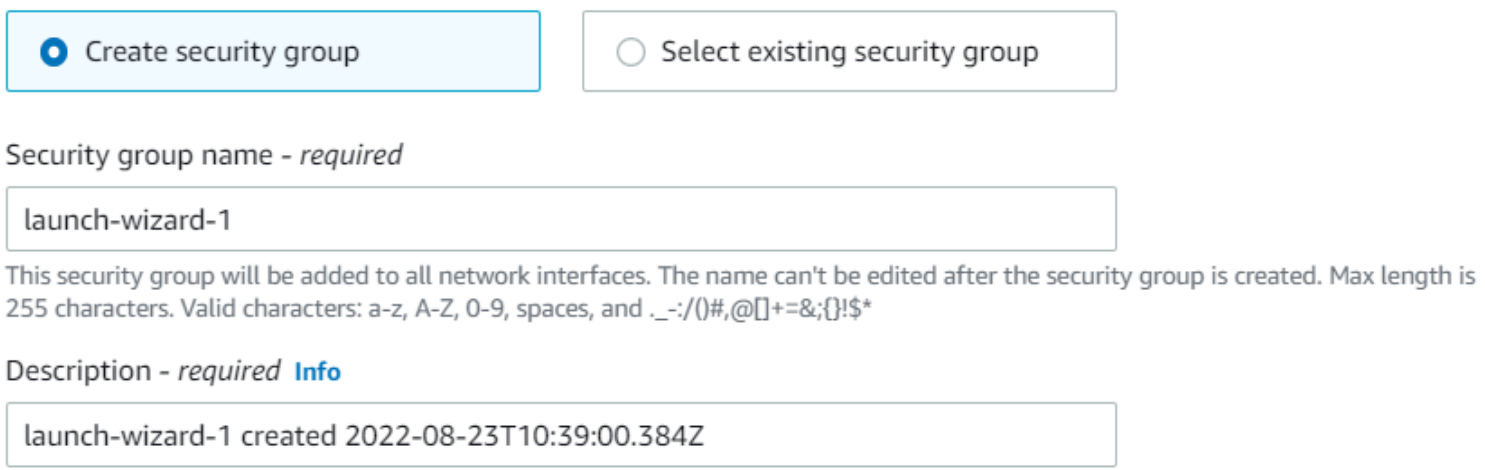

PuTTY Configuration
Category:
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Kex
Host keys
Cipher
Auth
TTY
X11
Tunnels
Bugs
More bugs
Serial
Options controlling SSH authentication
Display pre-authentication banner (SSH-2 only)
Bypass authentication entirely (SSH-2 only)
Authentication methods
Attempt authentication using Pageant
Attempt TIS or CryptoCard auth (SSH-1)
Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
Allow agent forwarding
Allow attempted changes of username in SSH-2
Private key file for authentication:
Browse...
About
Help
Open
Cancel


PuTTY Configuration
Category:
Select private key file
Cloud › AWS › Keypair
Search Keypair
Organize
New folder
Name
Status
EC2testkeypair1.ppk
Attachments
Center_Innovation
Cloud
Cluster_Lab
Innovation
Internship
IT3300
MSBA_advisign
Opma3306
SupplyChainAnalytics
Tableau
UNIV1101
File name: EC2testkeypair1.ppk
PuTTY Private Key Files (*.ppk)
Open
Cancel



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

Note if using aws academy sandbox



- 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)**.