EC2 = Elastic Compute Cloud

What is EC2?

Simplified Definition:
Think of EC2 as your basic desktop computer.

AWS Definition:
"Amazon Elastic Compute Cloud (Amazon EC2) provides ***scalable computing capacity*** in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to ***launch as many or as few virtual servers as you need***, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic."

AWS EC2

Quick Reference

Current Section = *EC2*

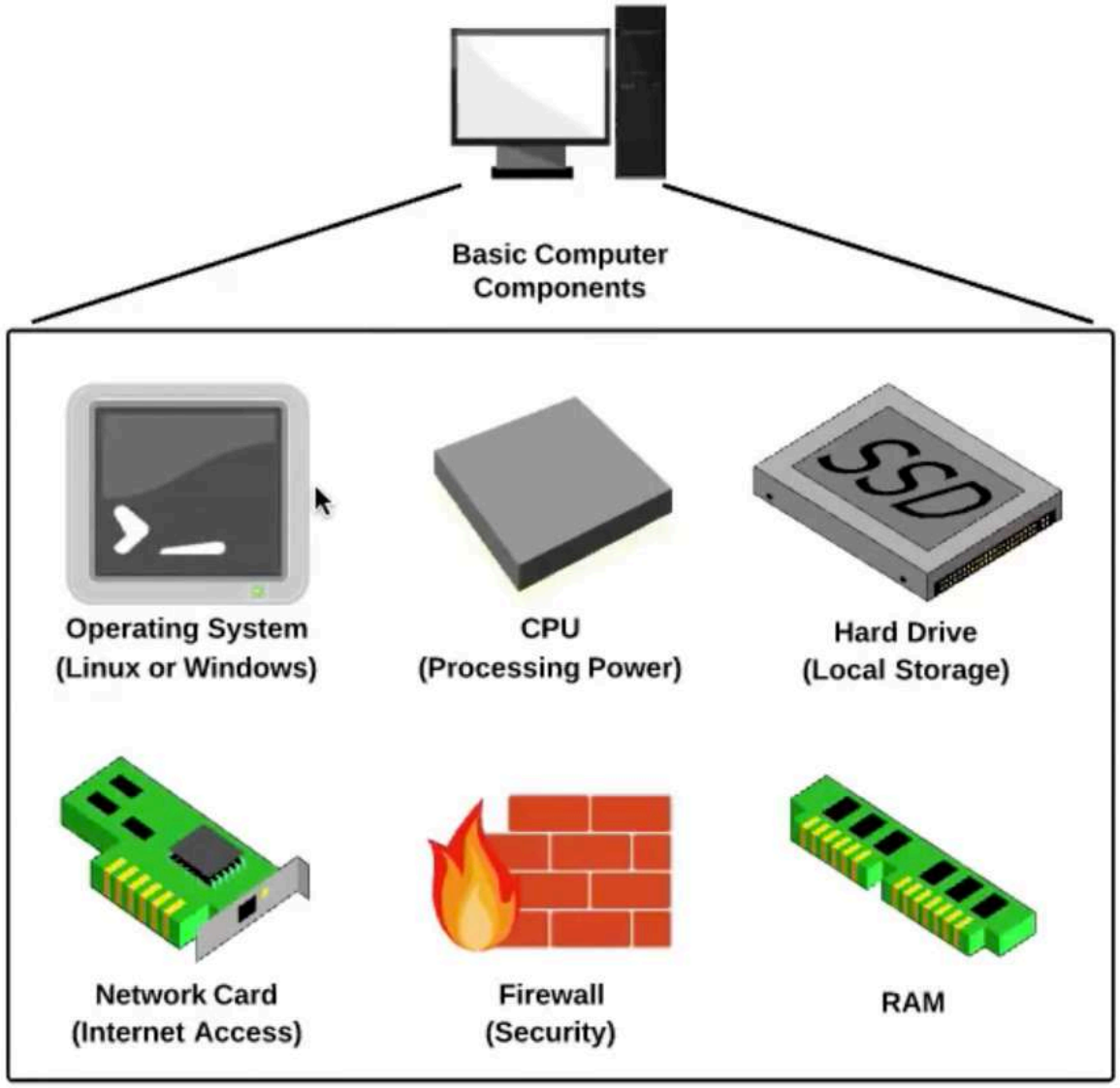
MORE -> 1 2 3 4 5 6

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

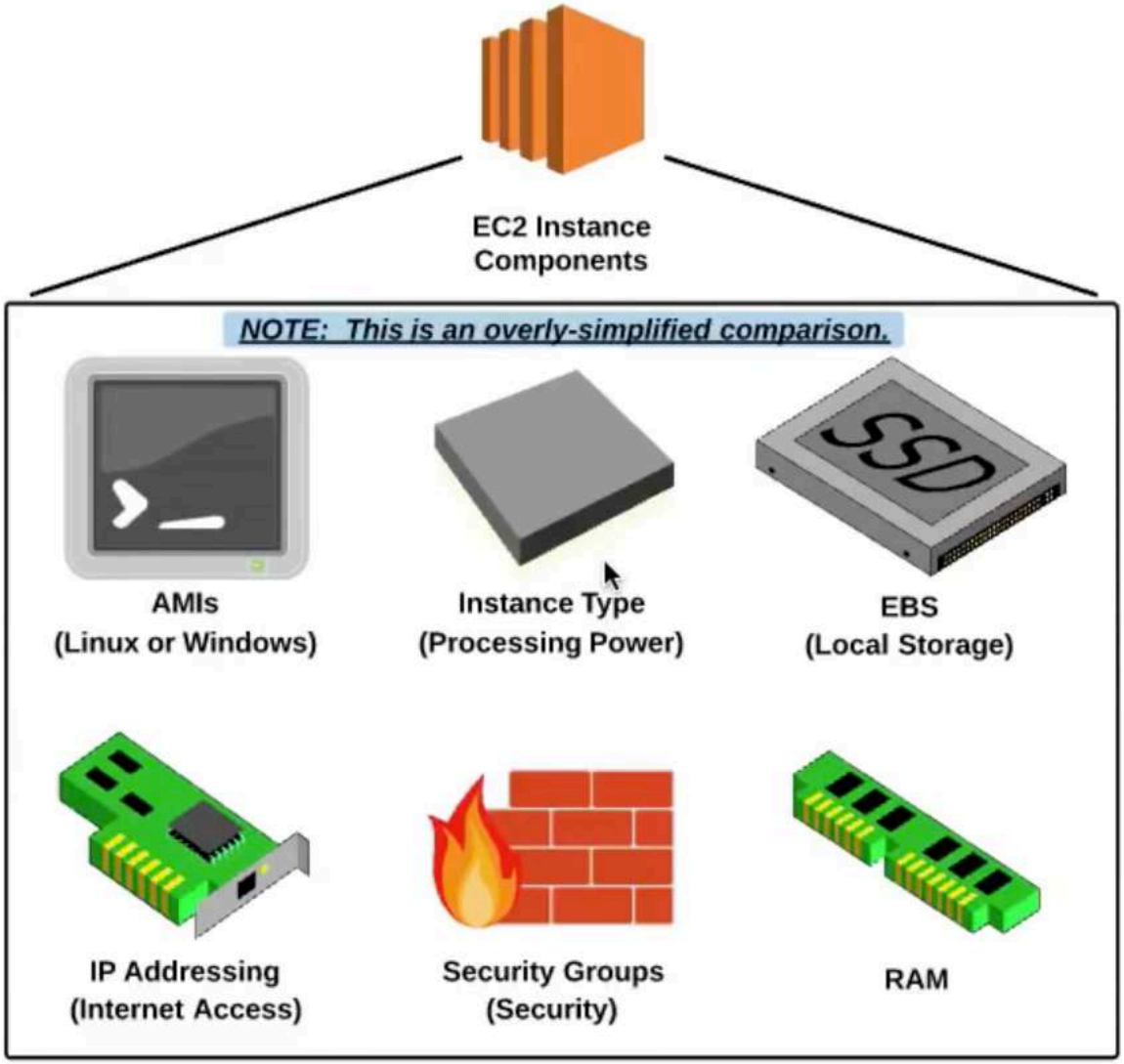
EC2 Basics:

Conceptually Understanding EC2:



EC2 Basics:

Conceptually Understanding EC2:



EC2 Basics:

EC2 Instance Purchasing Options (Most Common):

On-Demand:
On-demand purchasing allows you to choose any ***instance type*** you like and provision/terminate it at any time (on-demand).

(1) Is the ***most expensive*** purchasing option.
(2) Is the ***most flexible*** purchasing option.
(3) You are only charged when the instance is ***running*** (and billed by the hour).
(4) You can provision/terminate an on-demand instance at anytime.

Reserved:
Reserved purchasing allows you to purchase an instance for a ***set time period*** of one (1) or three (3) years.

(1) This allows for a ***significant price discount*** over using on-demand.
(2) You can select to pay upfront, partial upfront, no upfront.
(3) Once you buy a reserved instance, you own it for the selected time period and are ***responsible for the entire price*** - regardless of how often you use it.

Spot:
Spot pricing is a way for you to ***"bid"*** on an instance type and only pay for and use that instance when the spot price is ***equal to or below*** your "bid" price.

(1) This option allows Amazon to sell the use of ***unused instances***, for short amounts of time, at a ***substantial discount***.
(2) ***Spot prices fluctuate*** based on supply and demand in the spot marketplace.
(3) You are ***charged by the minute***.
(4) When you have an active bid, an instance is ***provisioned for you when the spot price is equal to or less than you bid price***.
(5) A provisioned instances ***automattically terminate when the spot price is greater than your bid price***.

Full list of Instance Purchasing Options:
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html

EC2 Basics:

Pricing/Cost Overview:

How are you charged for using EC2?

NOTE: ***Free Tier*** use is available for EC2.

(1) ***Purchasing Options***: (most common - not an inclusive list)
-On-demand
-Reserved
-Spot

(2) ***Instance Type*** - *the instance's processing capacity (think CPU)*:
-General purpose
-Compute optimized
-GPU optimized
-Memory optimized
-Storage optimized

(3) ***EBS Optimized***: (an option for higher IOPS performance)

(4) ***AMI Type*** - (think operating system):
-Linux (price varies based on distro/software packages)
-Windows (price varies based on version/software packages)

(5) ***Data Transfer*** - in/out of the instance.
(6) **Region** the instance is provisioned in.

NOTE: Before doing any major usage of EC2 instances, you should make sure to review AWS's current pricing model to make sure you understand how much you will be required to pay.

Detailed EC2 pricing based on storage class:
https://aws.amazon.com/ec2/pricing/

AMI = Amazon Machine Image

What is AMI?

Simplified Definition:
A preconfigured package required to launch and EC2 instance; includes an ***operating system***, software packages and other required settings.

AWS Definition:
"An Amazon Machine Image (AMI) ***provides the information required to launch an instance***, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as ***many instances from the AMI*** as you need. You can also launch instances from as many different AMIs as you need"

AMIs

Quick Reference

Current Section = *EC2*

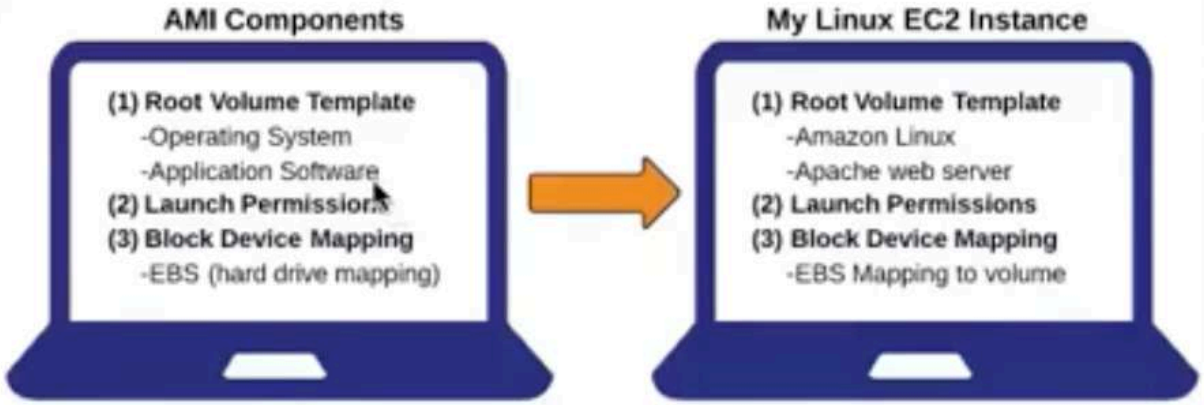
MORE -> 1 2 3 4 5 6

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

AMIs:

Conceptually Understanding AMIs:



AMI of the
"My Linux EC2 Instance"

When you create and AMI, you are essentially creating a template that you can use to launch another EC2 instance that has the exact same components as the original instance.

Quick Reference

Current Section = *EC2*

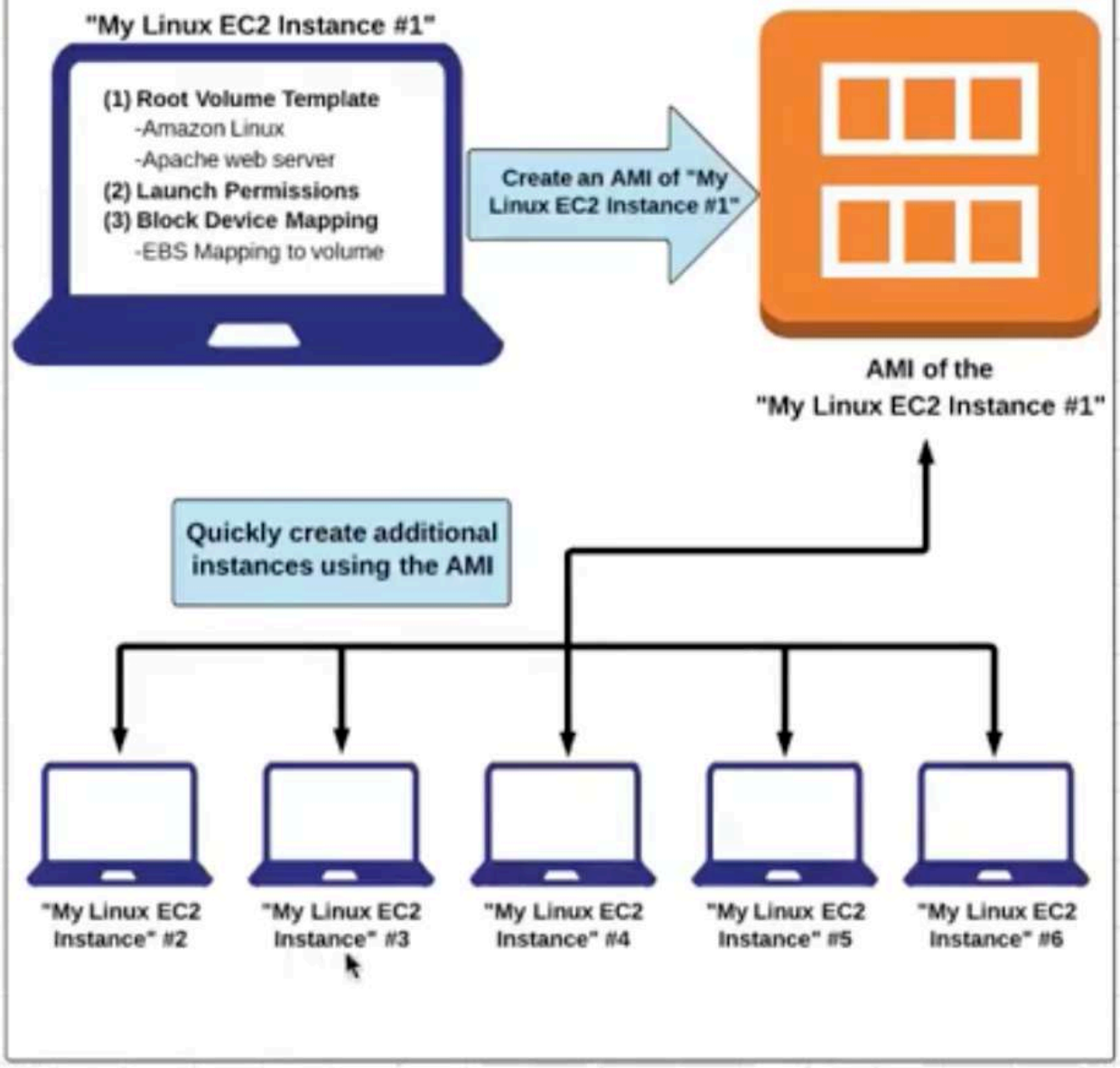
MORE -> 1 2 3 4 5 6

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

AMIs:

Conceptually Understanding AMIs:



AMIs:

When you launch an EC2 Instance, the first thing you do is select an AMI

Selecting an AMI:

AMIs come in three main categories:

(1) **Community AMIs:**

-Free to use

-Generally with these AMIs you are just selecting the OS you want

(2) **AWS Marketplace AMIs:**

-Pay to use

-Generally comes packaged with additional, licenced software

(3) **My AMIs:**

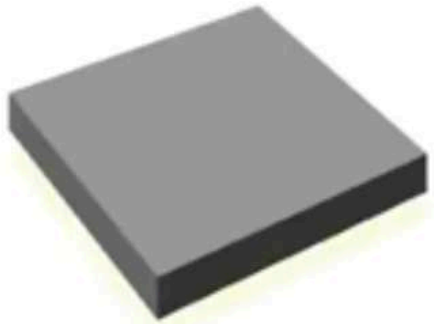
-AMIs that you create youself

Instance Types

What is an Instance Type?

Simplified Definition:
The ***Instance Type*** is the CPU (***computer power***) of your instance.

AWS Definition:
"When you launch an instance, the ***instance type*** that you specify determines the ***hardware of the host computer*** used for your instance. Each instance type offers different ***compute, memory, and storage capabilities*** and are grouped in instance families based on these capabilities. Select an instance type based on the requirements of the application or software that you plan to run on your instance."



Instance Types:

Instance Types Components:

(1) **Family:**
-A way of categorizing instance types based on what they are optomized to do

(2) **Type:**
-Subcategory for each family type

(3) **vCPUs:**
-The number of virtual CPUs the instance type uses

(4) **Memory (GiB):**
-The amount of RAM the instance type uses

(5) **Instance Storage (GB):**
-The local instance storage volume (your hard drive)

(6) **EBS-Optomized Available:**
-Indicates is EBS-optimized is an option for the instance type

(7) **Network Performance:**
-Network performance rating based on its data transfer rate (bandwidth capability)

EBS = **E**lastic **B**lock **S**tore

What is an EBS?

Simplified Definition:
EBS is a ***storage volume*** for an EC2 instance. (***Think of it as a hard drive.***)

AWS Definition:
"Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are ***highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone***. EBS volumes that are attached to an EC2 instance are exposed as ***storage volumes that persist independently from the life of the instance***."

EBS:

Before we move any further, we need to learn about IOPS

What are IOPS?

IOPS = Input/Output Operations per Second

Simplified definition:

The amount of data that can be written to or retrieved from EBS per second.

AWS Definition:

"IOPS are a ***unit of measure*** representing input/output operations per second. The operations are measured in KiB, and the underlying drive technology determines the maximum amount of data that a volume type counts as a single I/O. I/O size is capped at 256 KiB for SSD volumes and 1,024 KiB for HDD volumes because SSD volumes handle small or random I/O much more efficiently than HDD volumes."

What does this all mean?

More IOPS means better volume performance (faster read/write speeds)

What determines the amount of IOPS?

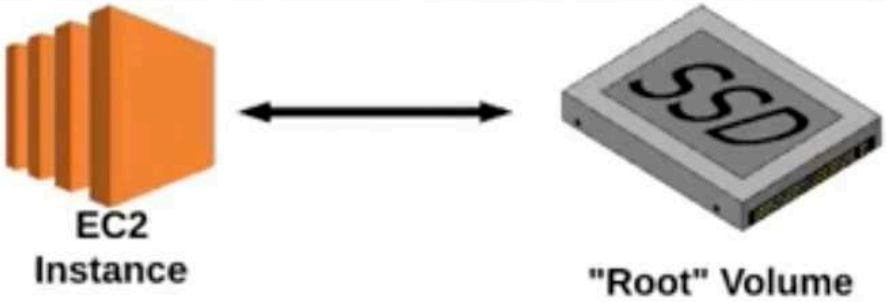
EBS volume size. The larger the storage size (in GiB), the more IOPS the volume has.

EBS:

"Root" vs. Additional EBS Volumes:

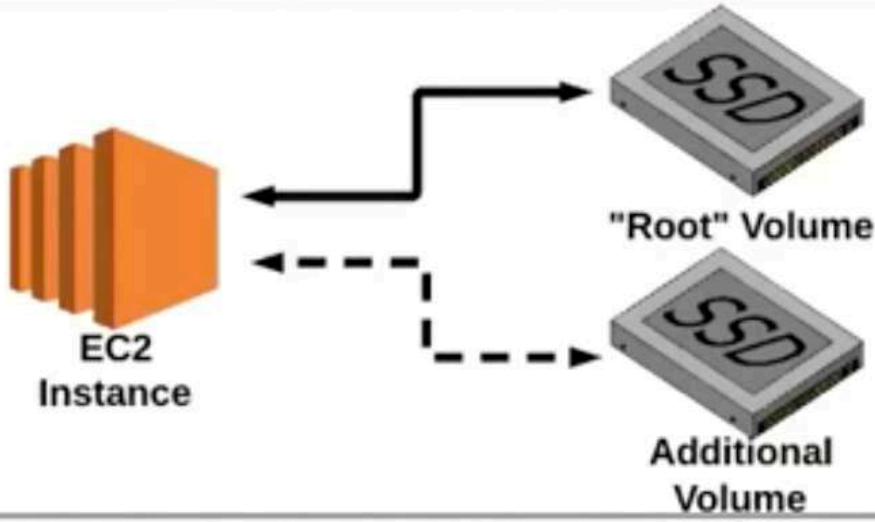
(1) Every EC2 instance MUST have a "root" volume, which may or may not be EBS.

(2) By default, EBS "root" volumes are set to be deleted when the instance is terminated. However, you can choose to have EBS volumes persist after termination.



(3) During the creation of an EC2 instance (or anytime afterwards) you can add additional EBS Volumes to the instance.

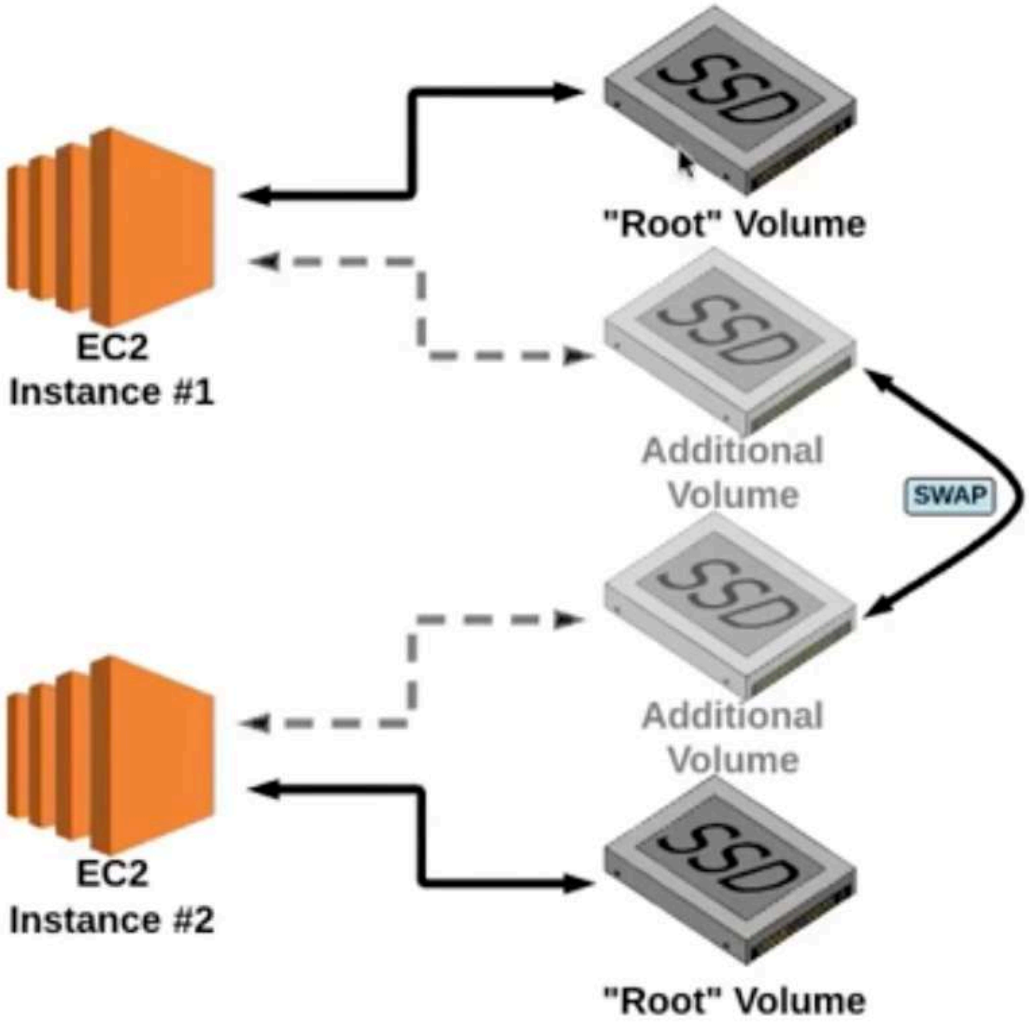
(4) Any additional volume can be attached or detach from the instance at any time, and is NOT deleted (be default) when the instance is terminated.



EBS:

"Root" vs. Additional EBS Volumes:

This means that you can "swap" EBS volumes between different EC2 instances, by "detatching" it from one and "attaching" it to another.



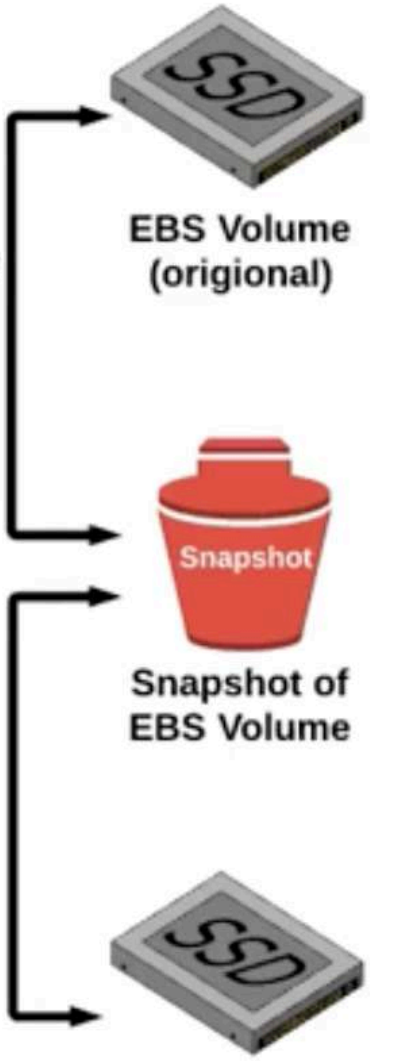
EBS:

Snapshots:

(1) A snapshot is an "image" of an EBS volume that can be **stored as a backup** of the volume OR used to **create a duplicate**.

(2) A snapshot is **NOT an active EBS volume**. You ***cannot attach or dettach*** a snapshot to an EC2 instance.

(3) To ***restore*** a snapshot, you need to c***reate a new EBS volume using the snapshot as its template***.



Security Groups

What are Security Groups?

Simplified Definition:
Security groups are very similar to NACLs in that they ***allow/deny traffic***. However, security groups are found on the ***instance level*** (as opposed to the subnet Level). In addition, the way ***allow/deny "rules" work are different from NACLs***.

AWS Definition:
"A *security group* acts as a ***virtual firewall that controls the traffic for one or more instances***. When you ***launch an instance, you associate one or more security groups with the instance***. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance."

Quick Reference

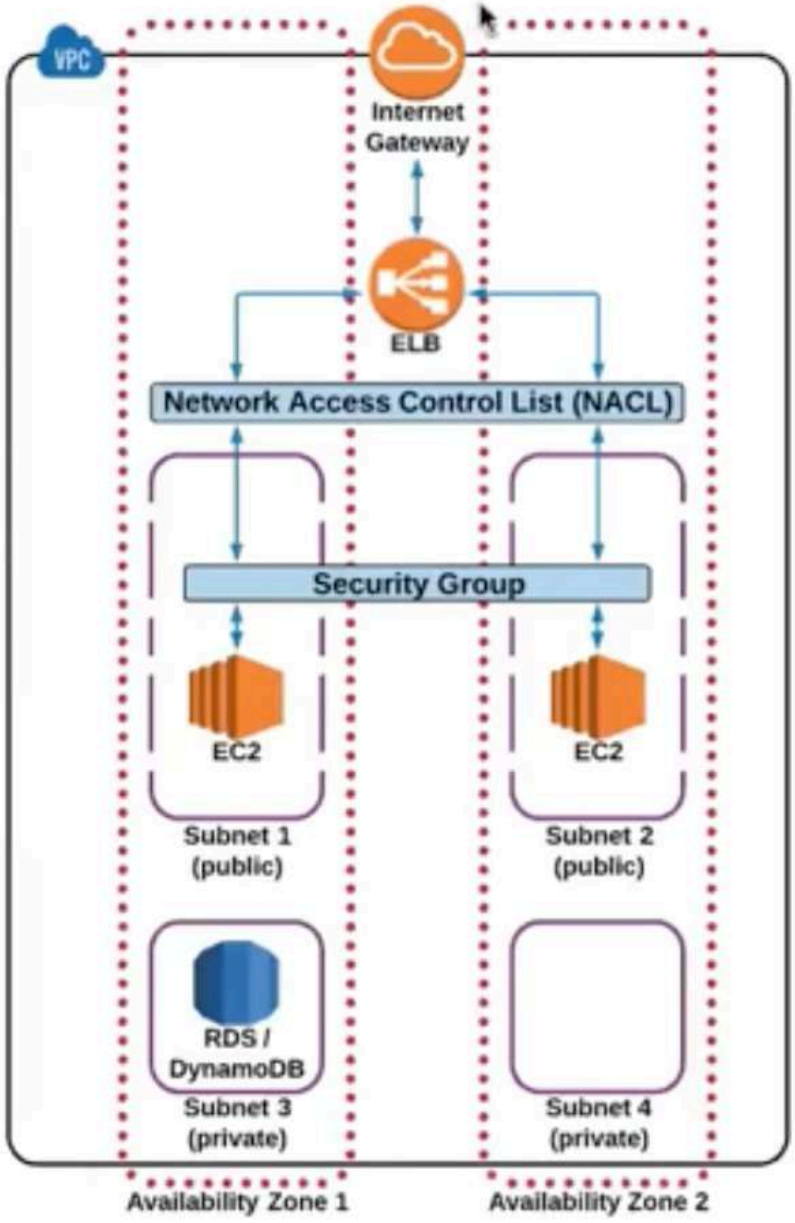
Current Section = *EC2*

MORE -> 1 2 3 4 5 6 7

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

Security Groups:



Quick Reference

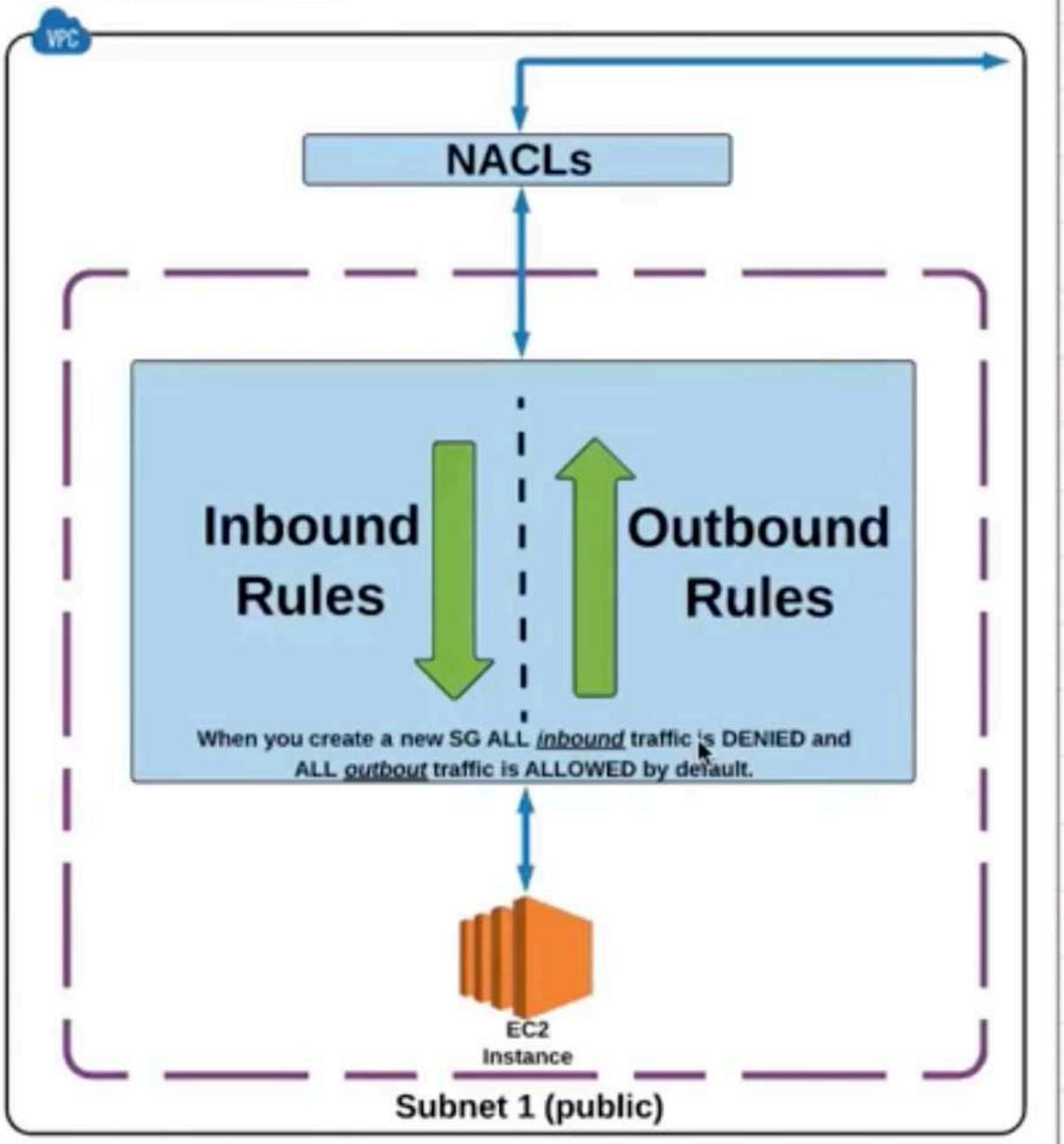
Current Section = *EC2*

MORE -> 1 2 3 4 5 6 7

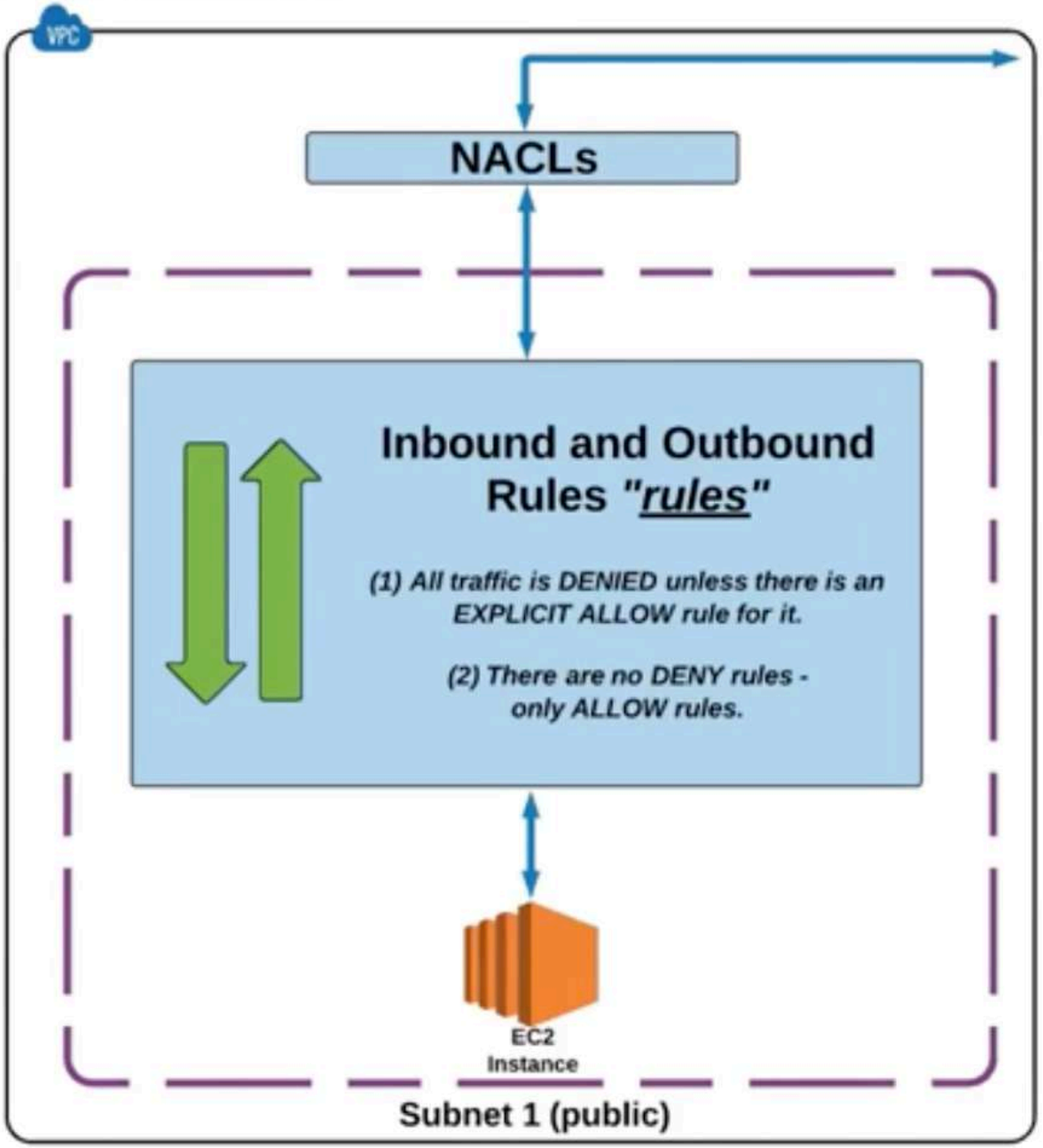
Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

Security Groups:



Security Groups:



Quick Reference

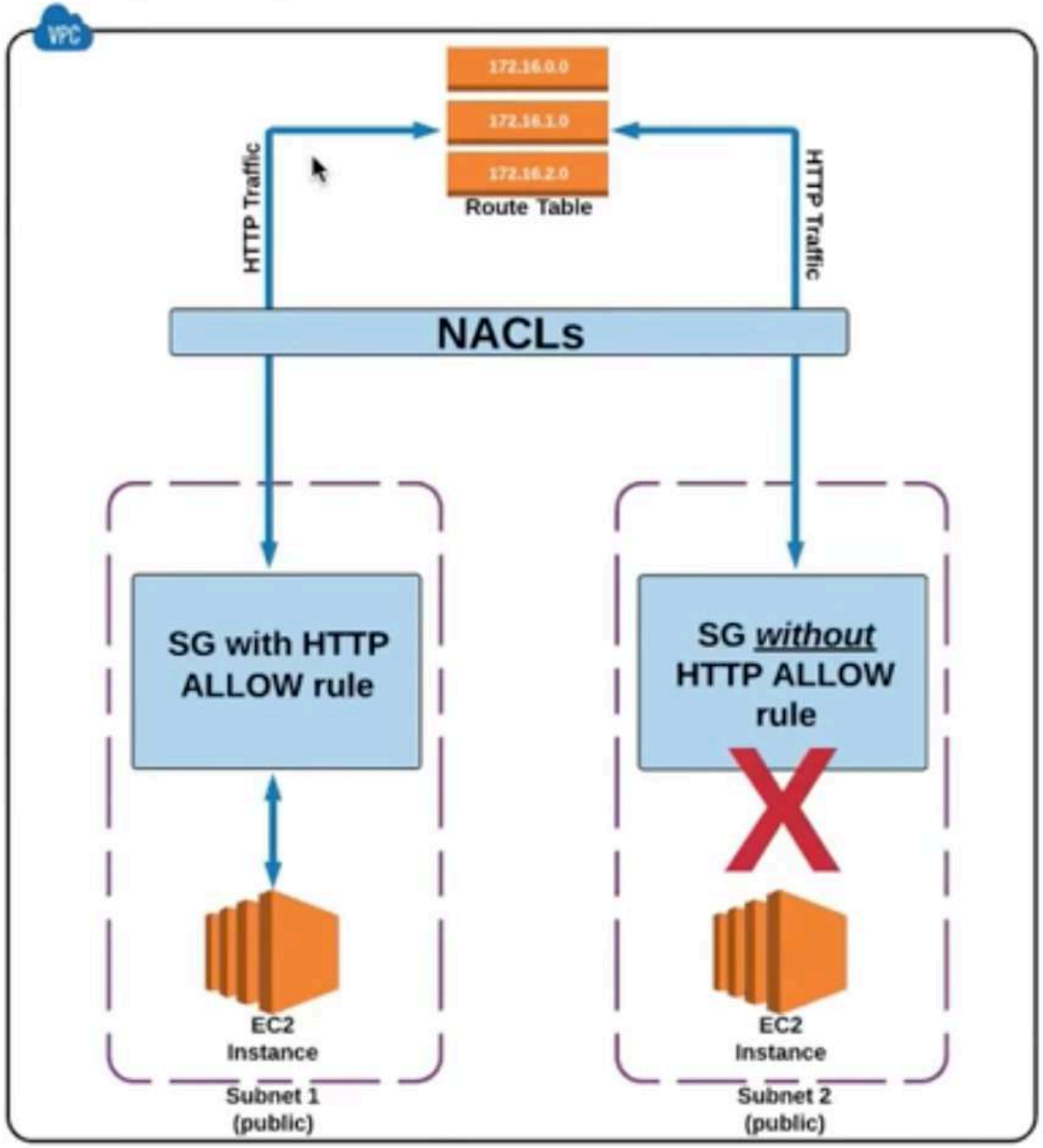
Current Section = *EC2*

MORE -> 1 2 3 4 5 6 7

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

Security Groups:



IP Addressing

What is IP Addressing?

Simplified Definition:
Providing an EC2 instance with ***public*** IP address.

What is an IP address?

Simplified Definition:
An IP address is the instance's "address" on the network.

Quick Example:
(1) Think of network traffic as a piece of postal mail.
(2) Think of an IP address as your home street address.
(3) Someone trying to deliver mail to your house will need your street address to find your location and deliver the mail.
(4) Without a street address the postal worker would never be able to find your home and deliver the mail.

IP Addressing

IP Addressing:

Public and Private IP Addresses:

Private IP Address:

(1) By default, ALL EC2 instances have a private IP address.
(2) Private IP addresses allow for instances to communicate with each other as long as they are located in the same VPC (or broader private network).

Public IP address:

(1) EC2 Instances can be launched with our without a public IP address (by default), depending on VPC/subnet settings.
(2) Public IP addresses are REQUIRED for the instance to communicate with the Internet.

NOTE: The "default" VPC and subnets are configured so that any new instance that is provisioend has a public IP address.

Quick Reference

Current Section = *EC2*

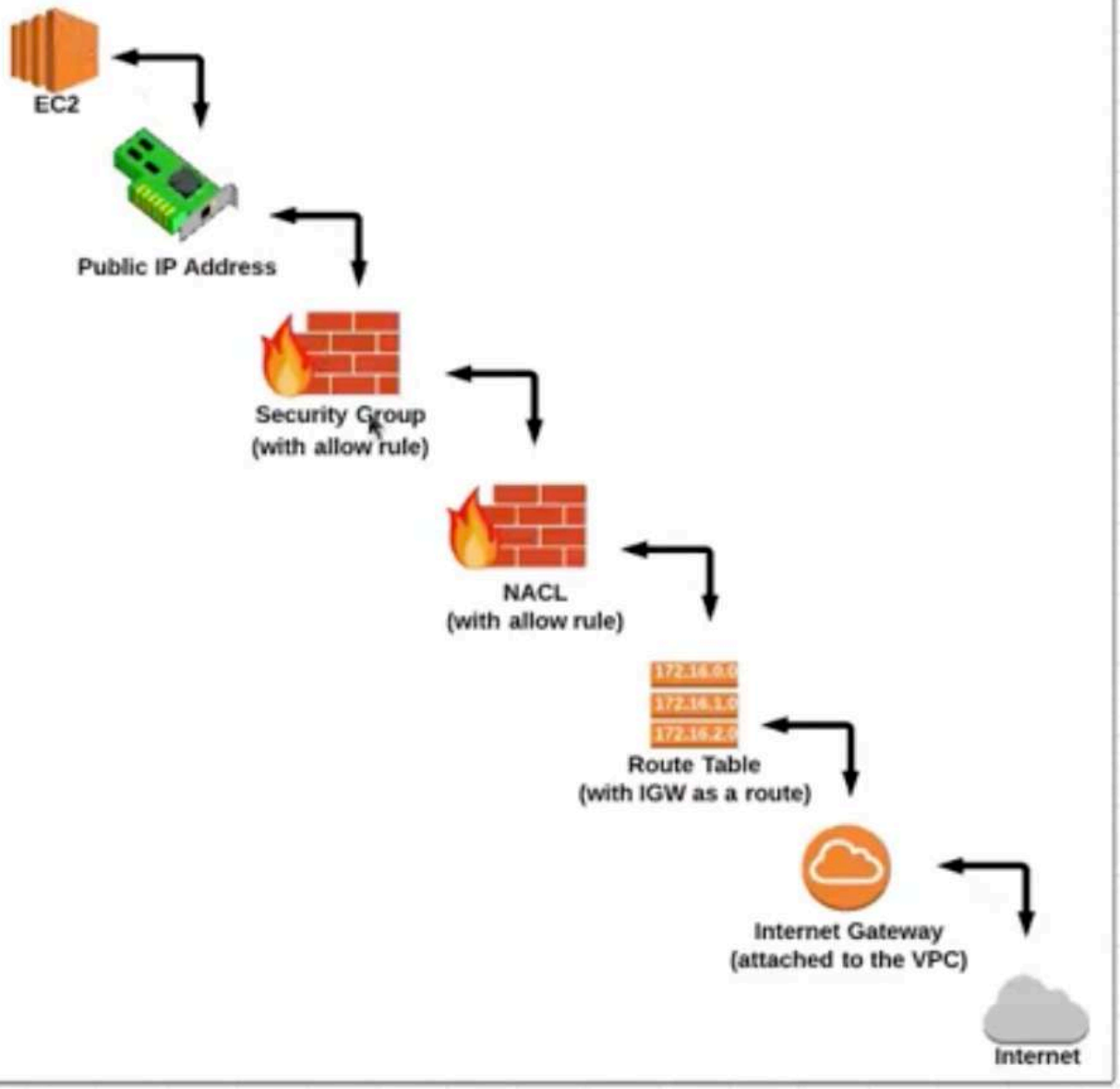
MORE -> 1 2 3 4 5

Lesson Navigation

- Start
- EC2 Basics
- AMI's
- Instance Types
- EBS Volumes
- Security Groups
- IP Addressing
- Provisioning EC2
- Finish
- Back to Main

IP Addressing:

RECAP: Everything an EC2 instance needs to communicate with the Internet.



IP Addressing

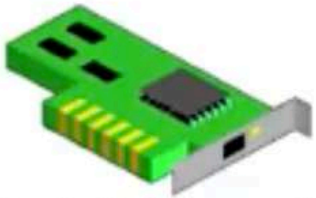
What is IP Addressing?

Simplified Definition:
Providing an EC2 instanace with ***public*** IP address.

What is an IP address?

Simplified Definition:
An IP address is the instance's "address" on the network.

Quick Example:
(1) Think of network traffic as a piece of postal mail.
(2) Think of an IP address as your home street address.
(3) Someone trying to deliver mail to your house will need your street address to find your location and deliver the mail.
(4) Without a street address, the postal worker would never be able to find your home and deliver the mail.



IP Addressing

Launching an EC2 Instance:

Basic Steps:

(1) Select an AMI
(2) Select an Instance Type
(3) Configure Instance Details:
-We are going to use this opportunity to run a *Bash Script* that installs Apache.

```
#!/bin/bash
yum update -y
yum install -y httpd
service httpd start
```

(4) Add Storage
(5) Add a Tag (give the instance a name)
(6) Configure/assign a Security Group
(7) Review & Launch
(8) Create & download a Key Pair

Connecting to an EC2 Instance (Linux/SSH):

Basic Steps:

(1) Select the instance
(2) Under "Actions", choose "Connect"
(3) Follow the instructions
a) Open a terminal to access the command line
b) Navigate into the directory that contains the key pair you downloaded
c) Run the chmod *command on the key pair to change its permissions*
d) Run the "example" command

You should now be connected to the instance!

PROJECT OMEGA

Have you completed the Project Omega infrastrucutre requirements for this section?

~~*(1) An AWS account.*~~
~~*(2) User accounts for the development team with access to core AWS services.*~~
~~*(3) Proper traffic routing into and out of our AWS Virtual Private Cloud (VPC).*~~
~~*(4) A location for bulk storage of files.*~~
(5) Servers to host and run Project Omega.
-One running EC2 instance using the Amazon Linux AMI.
-With Apache server installed.
-Verify that you can access the Apache test page.

(6) A database to store and catalog data.
(7) A way to sent notifications (email or text messages) to Project Omega's team members based on events that may occur with Projects Omega's infrastructure.
(8) A way to internally monitor parts of Project Omega's infrastructure.
(9) Automate the process of distributing incoming (external user) traffic evenly accross Project Omega's AWS resources.
(10) Automate the process of scaling up or scaling down AWS resources based on traffic demand.
(11) Set up and configure a web domain that points to Project Omega's infrastructure.
(12) Test the possibiliy of using "serverless" technology for Project Omega.