Passerelles numériques

A Gateway for Life

AWS EC2 - Advanced

AWS AMAZON

OBJECTIVe

1. **Assign a static public IP with *Elastic IP***
2. **Monitor an instance using *CloudWatch***
3. **Automatically increase resources using *Auto-Scaling***
4. **Share load between instances using *Load Balancer***
5. **Create *Snapshots***
6. ***Amazon Elastic File System***
7. **Useful tricks & tips for EC2**

Assign a static public IP with *Elastic IP*

Why?

- By default, AWS choose the public IP address assigned to your instance every time you restart it
- If you want your instance to always use the same public IP address every time you restart it, you need to reserve this public IP address using **Elastic IP (EIP)**

How?

- To use an EIP address, you first allocate it to your AWS account, and then associate it with your instance
- You can remove an EIP address from an instance, and re-associate it with a different instance
- AWS charges a small cost of $0.005 per hour

  if an EIP is associated with a non-running instance (stopped or terminated) or if an EIP is associated with your AWS account but not associated with one of your instance

Assign a static public IP with *Elastic IP*

- Allocate an EIP address to your AWS account
- Associate an EIP address to an instance
- Remove an EIP address from an instance
- Remove an IP address from your AWS account

Configure multiple *Network Interfaces*

Why?

- On AWS, a **network interface** acts like a network interface card (**NIC**) on a physical machine
- By default, AWS creates and assign one network interface for each new instance you create
- Like for a physical machine, sometimes you may need your instance to have more than just one network interface
- For example, if you want your instance to become a firewall or a proxy, it will need more than just one network interface

How?

- You can create a network interface, attach it to an instance, detach it from an instance, and attach it to another instance
- Every instance in has a default network interface, called the **primary network interface (eth0)**

Configure multiple *Network Interfaces*

- Check the default network interface on an instance
- Attach a second network interface to an instance and connect it to a different subnet
- Detach a network interface from an instance and attach it to a different instance
- See the network interfaces attached to a system directly from Linux CLI

Monitor an instance using *CloudWatch*

Why?

- Companies and organizations want to make sure their essential instances and applications are **always up and running (available)** to their staff, clients, etc.
- **Real-time monitoring** solutions, such as Zabbix can help you achieving that goal by **monitoring performances and resources utilization** (such as CPU, memory, disk space, bandwidth, connectivity) and creating **alarms** (email, etc.)
- AWS provides its own monitoring solution named **CloudWatch**

How?

- CloudWatch collects all the monitoring data needed from your instance
- You can **visualize** monitoring information from a **dashboard** showing graphs and alerts

Monitor an instance using *CloudWatch*

- Visualize graphs in monitoring dashboard
- Create alarms that send emails automatically
- Simulate high CPU utilization to raise an alarm

What is Amazon EC2 Auto Scaling?

It helps you ensure that you have the correct number of EC2 instances available to handle the load for your application.

It helps ensure that your application always has the right amount of capacity to handle the current traffic demand.

Auto Scaling can detect when an instance is unhealthy, & can replace it. It can also launch instances in multiple Availability Zone.

Auto scaling can dynamically increase and decrease capacity as needed. Because you pay for your EC2 instances you use, you save money by launching instances when they are needed and terminating them when they aren't.

Automatically increase resources using *Auto-Scaling*

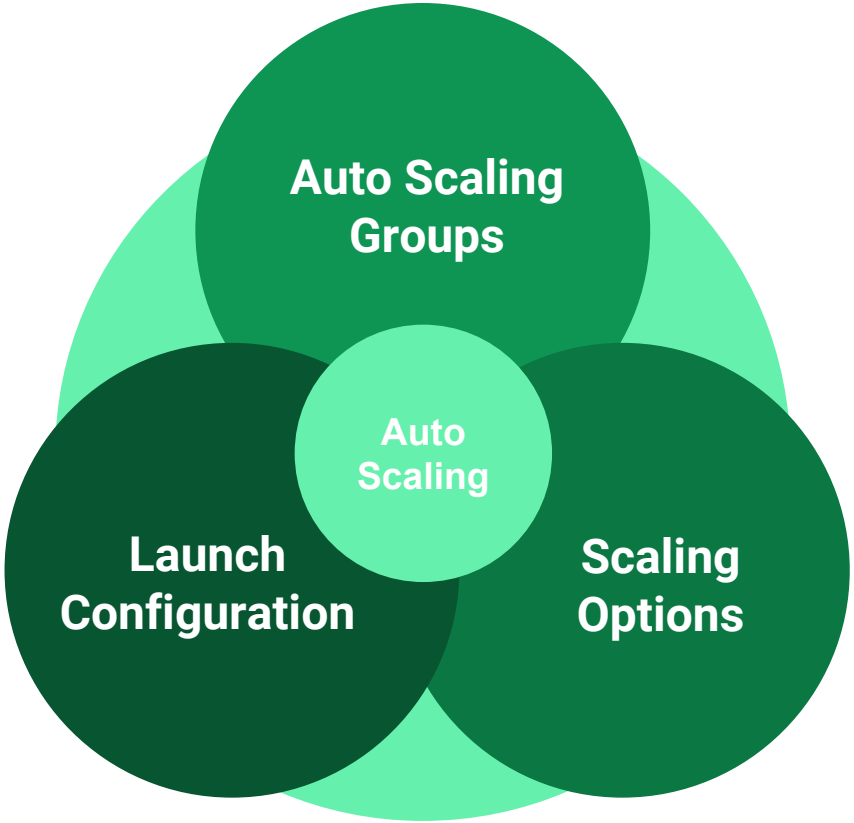
Why?

- Remember, AWS **costs** depends on the **size** of your instance and the **number** of running instances
- Sometimes the number of running instances you need **changes over the time**
    - For example, if you work for a video games company, there will be a lot more people who need to access your instances in the evening and during the weekends. You will need more instances because there are more players using them
    - But you don't want to pay for all these expensive instances in the morning when there are not so many players

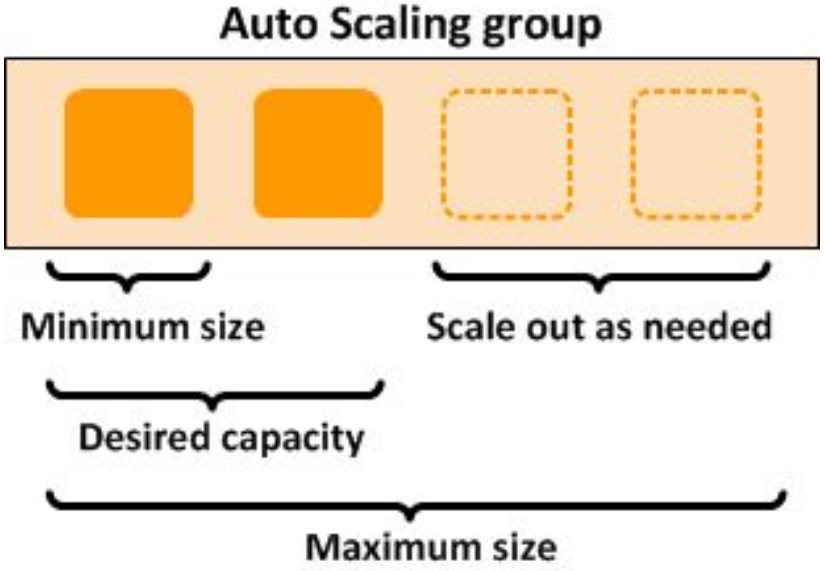
How?

- **Auto-scaling** will do two things for you:
    - It will **automatically create more instances** when your needs **increase** (more players want to play)
    - It will **automatically remove some instances** when your needs **decrease** (less players so less expensive)

Auto-Scaling components



Auto-Scaling components



Minimum Size	01
Desired Capacity	02
Max Size	04

Auto Scaling Group

it is the group of instances that should be in auto scaling based on policy for scaling.

You can specify its minimum (not below), maximum, and desired (running) number of EC2.

Auto-Scaling components

Launch Configuration

uses a launch template or a launch configuration as a configuration template for it's EC2 instances. So you have to create AMI for your application before add to auto-scaling group.

Scaling options

There are several ways for you to scale your auto scaling groups such as:

- Manual scaling
- Scheduled scaling
- Dynamic scaling.

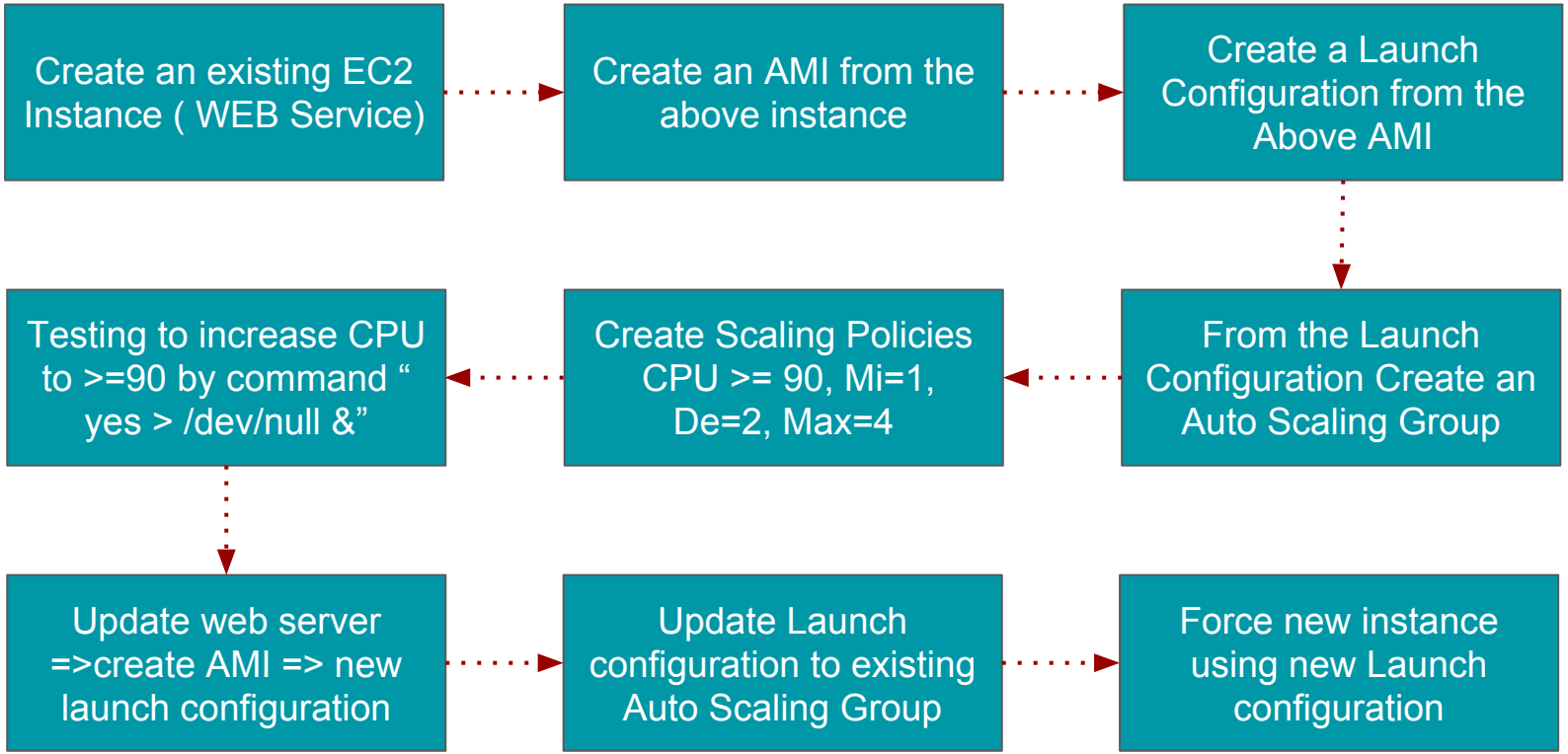
Update application in Auto-Scaling

If you want to update new application on your existing auto-scaling:

- Create New Launch Configuration
- Update auto scaling group and point to new launch configuration
- Perform scale-in and scale-out until all instances are hosting new application version

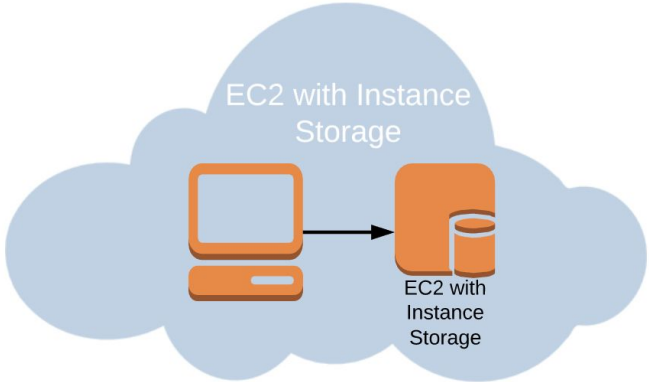
Note: you can’t update your application directly to your auto-scaling

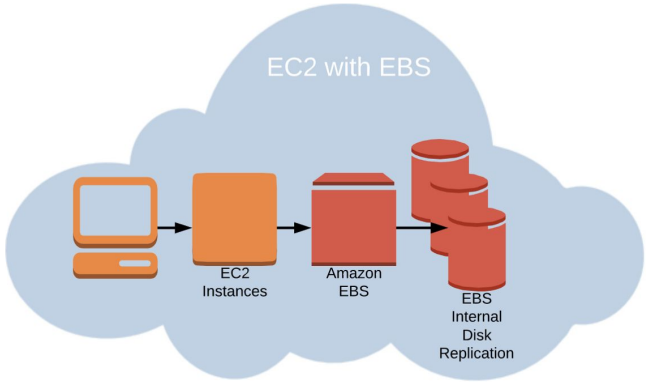
Auto-Scaling Demo + Practice



EC2 Storage Overview

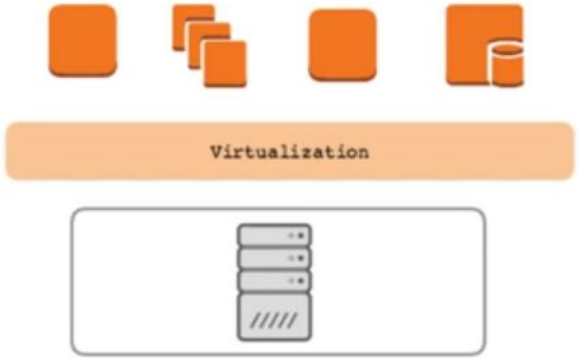
EC2 instances support two types of level storage: EC2 Instances can be launched using either **Elastic Block Store (EBS)** or **Instance Store volume** as root volumes and additional volumes. Your instance price is not include storage.





EC2 Storage (Instance store)

- **AWS Instance Store** provides a temporary block storage volumes for use with EC2.
- This storage is located on the disks that are physically attached to the host computer.
- Size of instance store varies depending on your instant type.



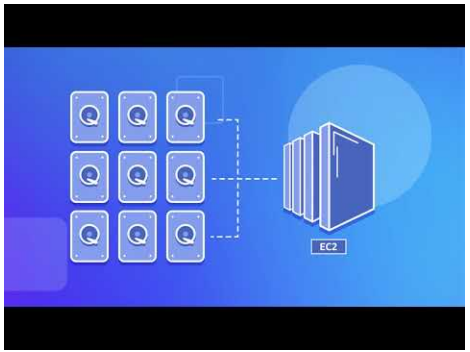
EC2 Storage (Instance store)

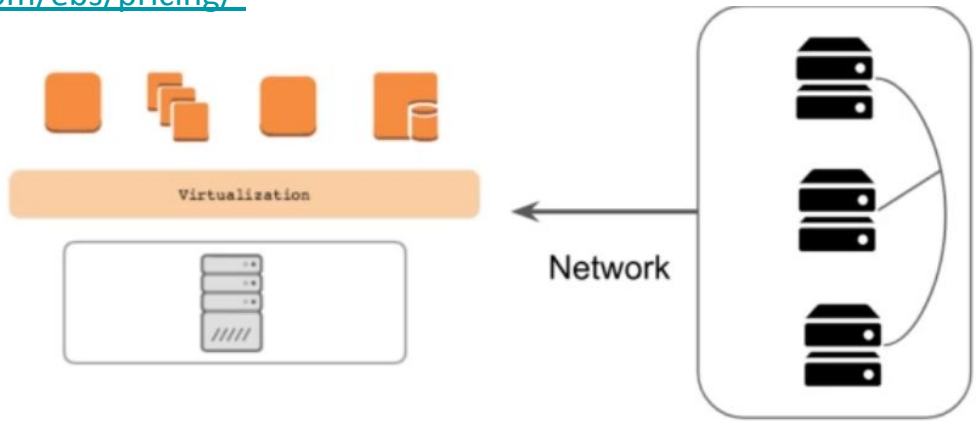
- Data in instance store is lost in following situation:
  - The underlying disk drive fails.
  - The instance stops.
  - THe instance terminates.
- Instance store are included in the cost of EC2 instance, so they are quite cost effective.
- If planning to use instance store, make sure you backup your data to central storage places like S3.

EC2 Storage (EBS)

- AWS ***Elastic Block Store*** is a persistent block storage volumes for use with EC2.
- Each EBS volume is designed for 99.9999% availability & are automatically replicated within its availability zone.
- EBS can be elastic in nature, thus supports dynamic increase in capacity, performance and can change instance type of live volumes.
- Price for the volume is different, it is depend on what type of storage you use.

  More detail go to : https://aws.amazon.com/ebs/pricing/





EC2 Storage Demo + Practice

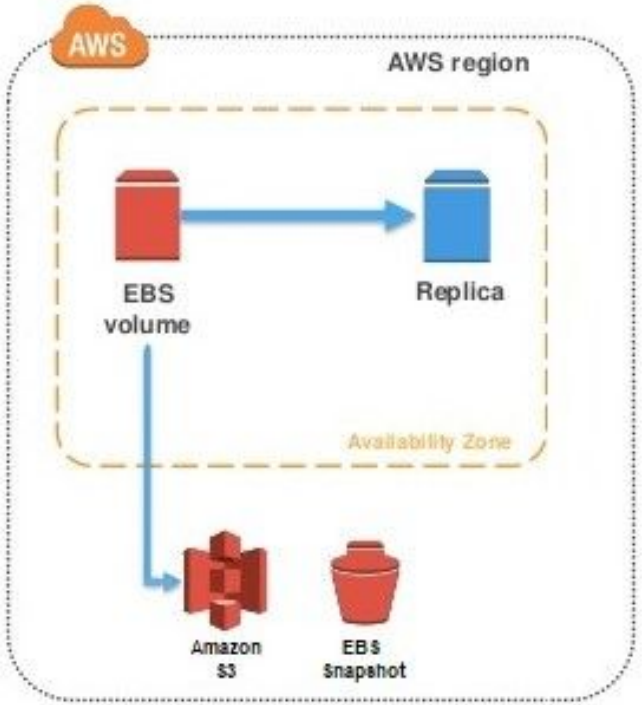
1. Create EC2 ***01*** with volume **EBS**
2. Try to increase size of volume to 20GB
3. Create the new volume with 10GB (any type) and add to existing EC2 ***01***
4. *Create partition => format=>mount=> put some data*
5. Verify your volume size is increasing and have new volume
6. *Disconnect from instance EC201 and add to new EC202 (create new)*
7. Verify your data on new instance EC202

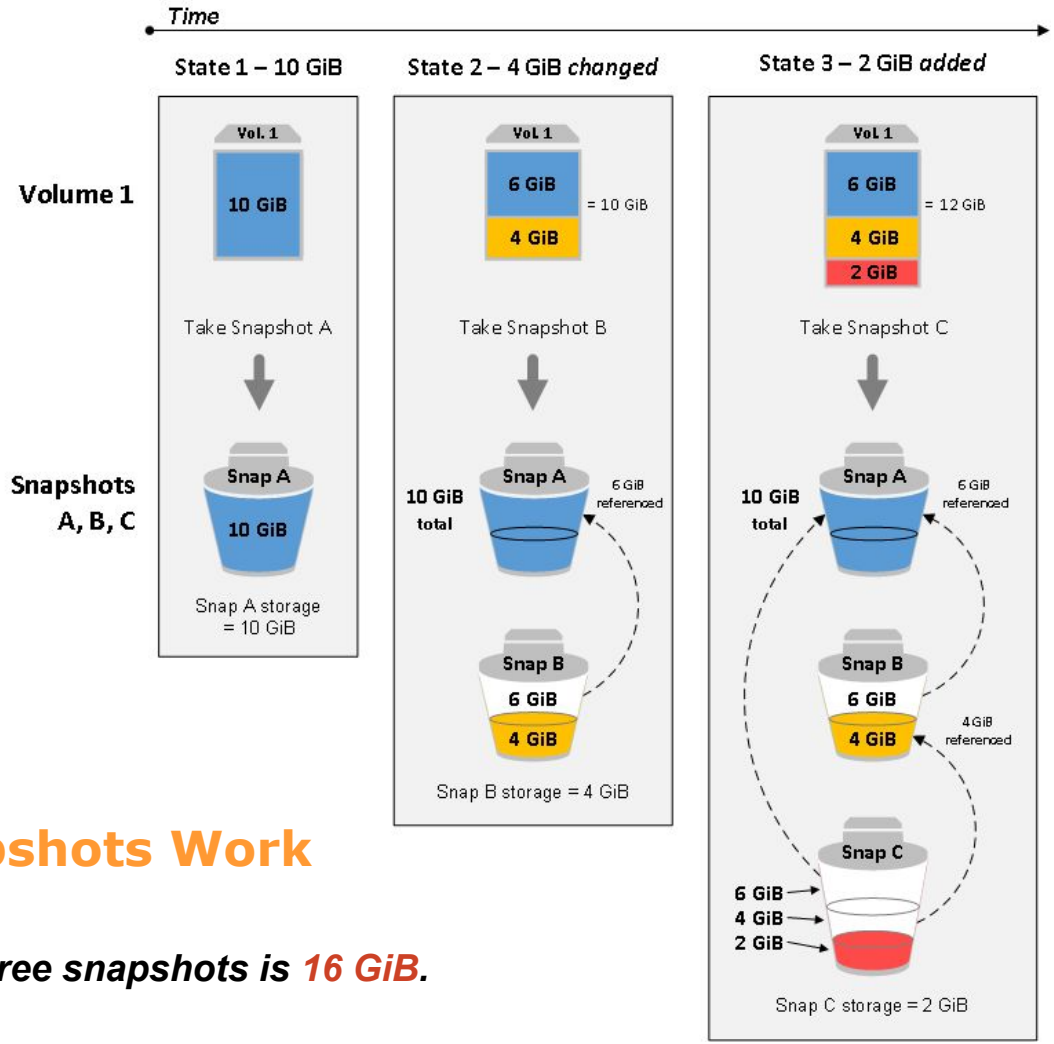
Note:

- volume can increase but not decrease
- You can't create new partition from root volume

EBS Snapshots

- **EBS Snapshot** is used to backup data from EBS Volume to **S3** (simple storage service) by taking an point-in-time snapshot.
- Snapshot is nothing but an **incremental** backups, means a copy containing only the files which are updated. This helps in minimizing the time required to create the backup and space.
- When delete each snapshot is not effective other snapshot.
- You can copy, share and encrypt snapshot
- You can do :
    - Single snapshot: on volume
    - Multiple snapshot: on instance





How Incremental Snapshots Work

The total storage required for the three snapshots is 16 GiB.

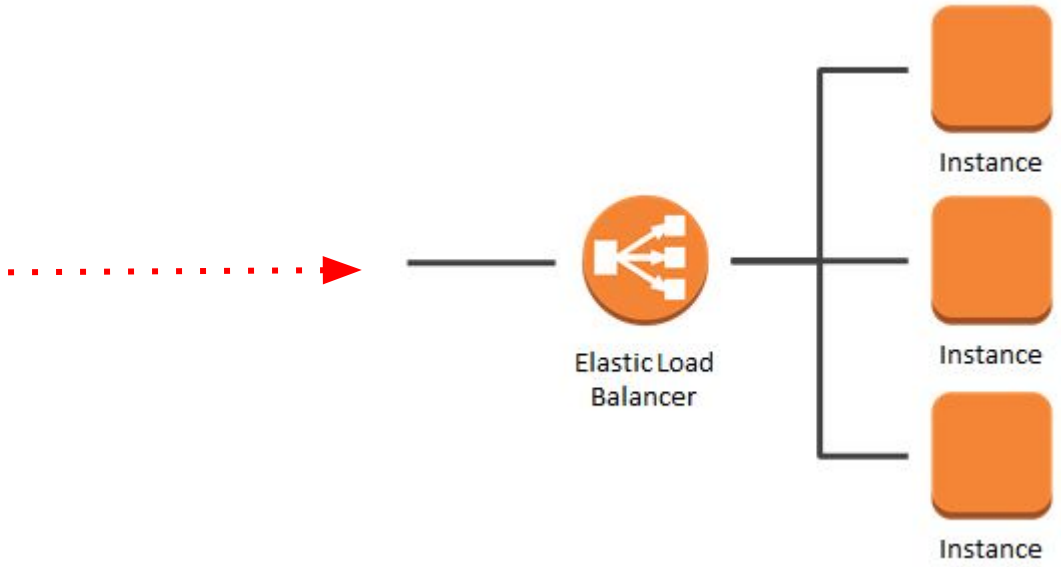
EC2 snapshot Demo + Practice

1. Create new instance “demo01” with volume EBS 8GB
2. Create file demo.txt (add some text)
3. Create the snapshot “demo-A” from the demo01
4. Add new words to demo.txt and create new file “demo1.txt”
5. Create other snapshot “demo-B” from instance demo01
6. Terminate the instance demo01
7. Create image from each snapshot (Image-A, image-B)
8. Create two instances from Image-A, image-B (demo-02, demo-03)
9. Verify your data on each instance

Note: Don’t practice with “fast restore point” on snapshot because it cost a lot money

Load Balancers

- Elastic Load Balancer allows to distribute the incoming traffic to multiple instances.
- ELB is capable of handling rapid change in the network traffic .
- Since it’s managed service, client does not have to worry about high availability.



Load Balancers

There are three main types of load balancer:

- Application Load Balancer: is best suited for load balancing of HTTP and HTTPS traffic on layer 7 connection. .
- Network Load Balancer: is best suited for load balancing of TCP, UDP on layer 4 connection.
- Classic Load Balancer: provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level.

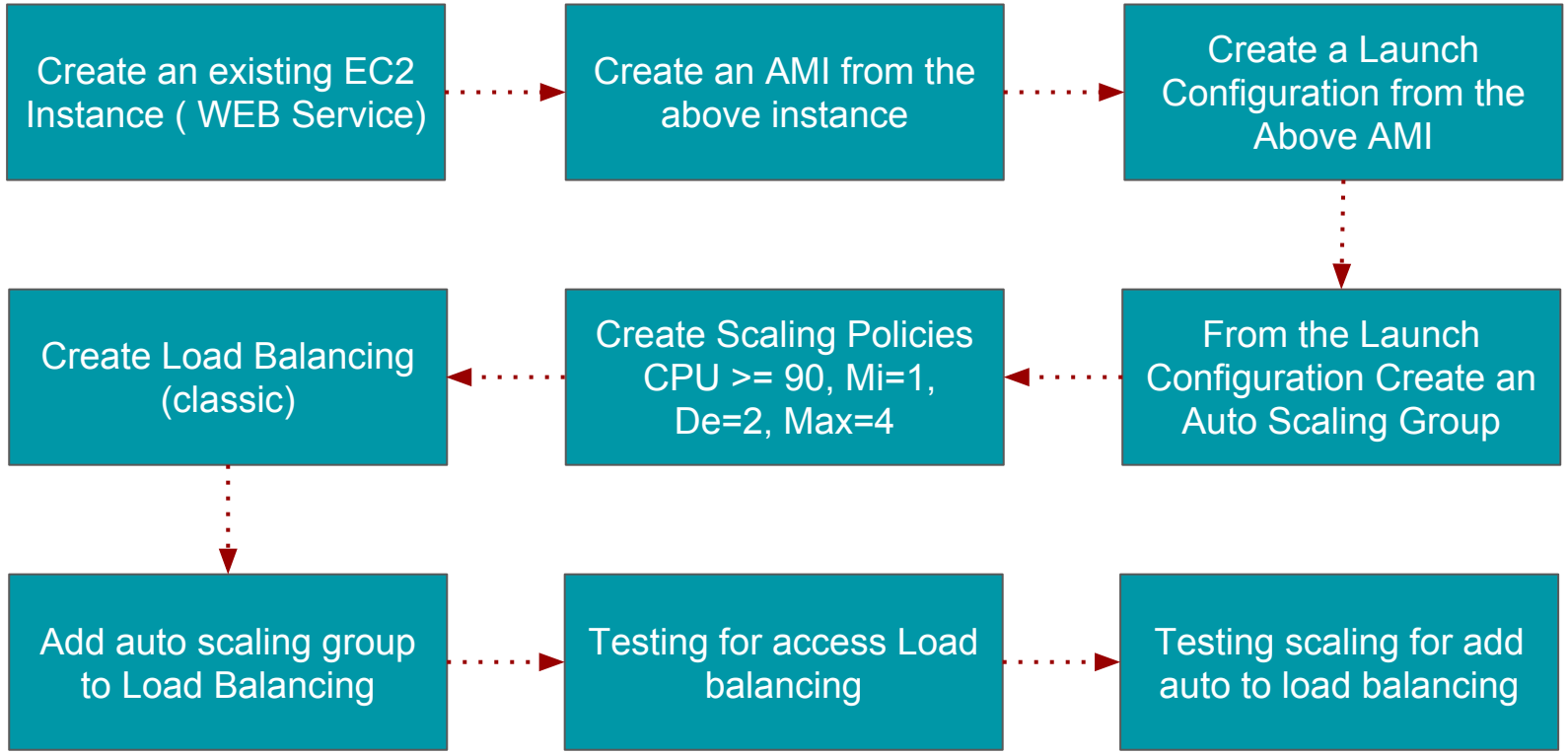
Example pricing:

- $0.025 per Classic Load Balancer-hour (or partial hour)
- $0.008 per GB of data processed by a Classic Load Balancer

Load balancer Demo + Practice 01

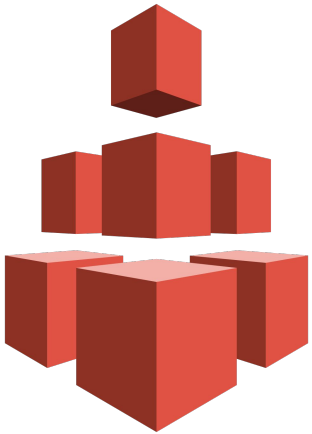
1. Create new two Linux instances
2. Install nginx web service to both instances
    - Change password root " sudo passwd root"
    - Install nginx " yum install nginx" or " sudo amazon-linux-extras install nginx1.12"
    - Start service nginx " service nginx start"
    - Edit html file.
        - Echo **"<h1>demo 01</h1>" > index.html** to instance 01 (/usr/share/nginx/html)
        - Echo **"<h1>demo 02 </h1>" > index.html** to instance 02 (/usr/share/nginx/html)
    - Test access to each instance by ip address or domain name. Why are not working?
3. Create "classic load balancer" allow port 80 to connect to instance
4. Add both instances to load balancer
5. Test access by domain name on load balancer and you will see what happend

Load balancer Demo + Practice 02

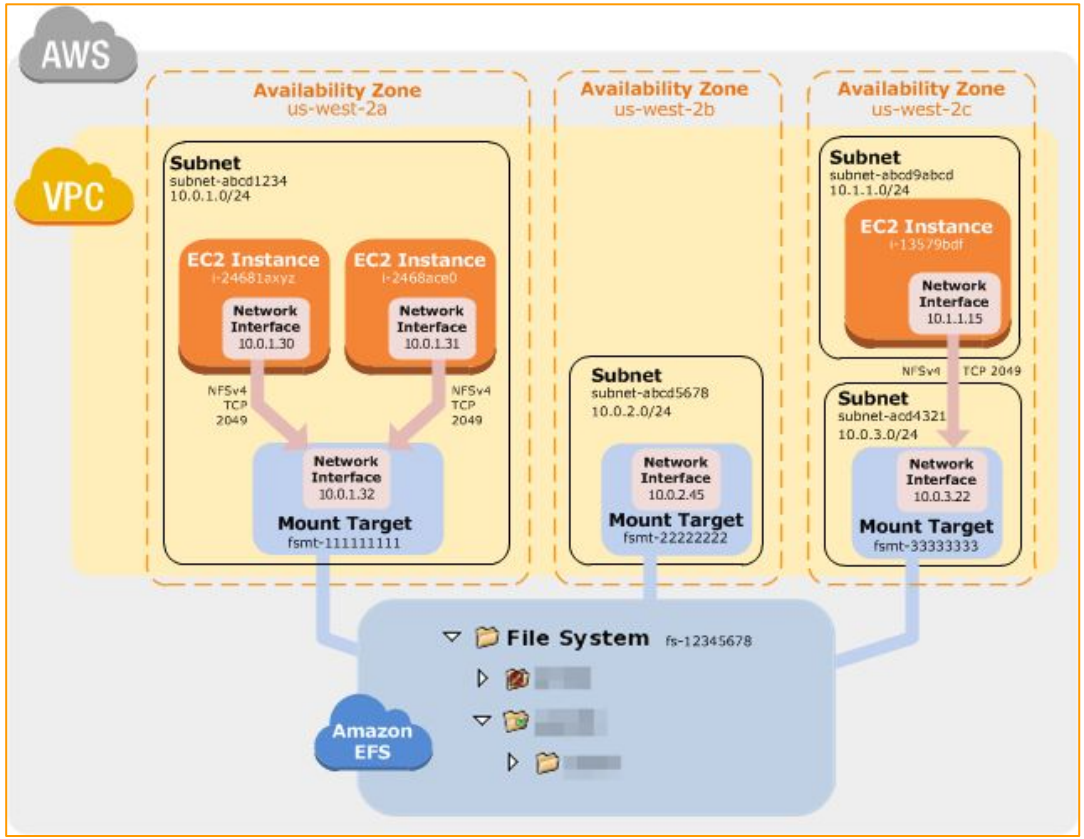


Amazon Elastic File System

- ***Amazon Elastic File System*** (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services based on EC2.
- ***Network File System (NFS)***, is the service allows all network users to access shared files stored on computers of different types. NFS using TCP port= 2049.
- NFS file system for $0.08/GB-Month (always change)



Amazon Elastic File System



Amazon Elastic File System 01

Check Demo and practice in file “ EFS-Practise”

https://drive.google.com/a/passerellesnumeriques.org/file/d/1cNovYjpqBBfWhXrWA-EKqykhBEzptZ5i/view?usp=sharing

Useful tips & tricks for EC2

- Use the name field to help identifying your instance
- Use tags to organize your instances
- You can assign multiple security groups to the same instance
- Make sure you don’t lose your key pair or you won’t be able to connect to your instance anymore
- Consider using snapshots to save your work (like a backup) on your instance

What’s next

What we have learned in this module

- How to use and configure Elastic IP
- How to use and configure CloudWatch
- How to use and configure Auto Scaling
- How to use and configure Load Balancers
- How to use snapshots
- Useful advices on AWS

What we will learn in the next modules

- How to create your first Virtual Private Cloud network
- How to create your first website on AWS
- An a lot more…..

Questions

www.passerellesnumeriques.org