Lambda - Overview

Tuesday, December 27, 2016 5:07 PM

What is Lambda?

AWS Lambda is a compute service that runs your code in response to events and automatically manages the underlying compute resources for you.

AWS Lambda can automatically run code in response to modifications to objects in Amazon S3 buckets, messages arriving in Amazon Kinesis streams, or table updates in Amazon DynamoDB.

What Actually Is It?

- Data Centres
- Hardware
- Assembly Code/Protocols
- High Level Languages
- Operating Systems
- AWS API's
- AWS Lambda

Lambda

Lambda runs your code on high-availability compute infrastructure and performs all the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging.

All you need to do is supply the code.

What Events Trigger Lambda?

You can use AWS Lambda to respond to table updates in Amazon DynamoDB, modifications to objects in Amazon S3 buckets, messages arriving in an Amazon Kinesis stream, AWS API call logs created by AWS CloudTrail, and custom events from mobile applications, web applications, or other web services.

It is an old video, now lambda also supports python and java too.

Lambda

- Supported Programming Languages - Javascript
- AWS Lambda is designed to provide 99.99% availability for both the service itself and for the functions it operates.

Lambda - Pricing

Requests

First 1 million requests are free

$0.20 per 1 million requests there after

Lambda - Pricing

Duration

Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $0.00001667 for every GB-second used.

Lambda - Pricing

Free Tier

1M free requests per month and 400,000 GB-seconds of compute time per month. The memory size you choose for your Lambda functions determines how long they can run in the free tier. The Lambda free tier does not automatically expire at the end of your 12 month AWS Free Tier term, but is available to both existing and new AWS customers indefinitely.

Lambda - Pricing

Memory (MB)	Free tier seconds per month	Price per 100ms ($)
128	3,200,000	0.000000208
192	2,133,333	0.000000313
256	1,600,000	0.000000417
320	1,280,000	0.000000521
384	1,066,667	0.000000625
448	914,286	0.000000729
512	800,000	0.000000834
576	711,111	0.000000938
640	640,000	0.000001042
704	581,818	0.000001146
768	533,333	0.000001250
832	492,308	0.000001354
896	457,143	0.000001459
960	426,667	0.000001563
1024	400,000	0.000001667

An example for pricing -

Lambda

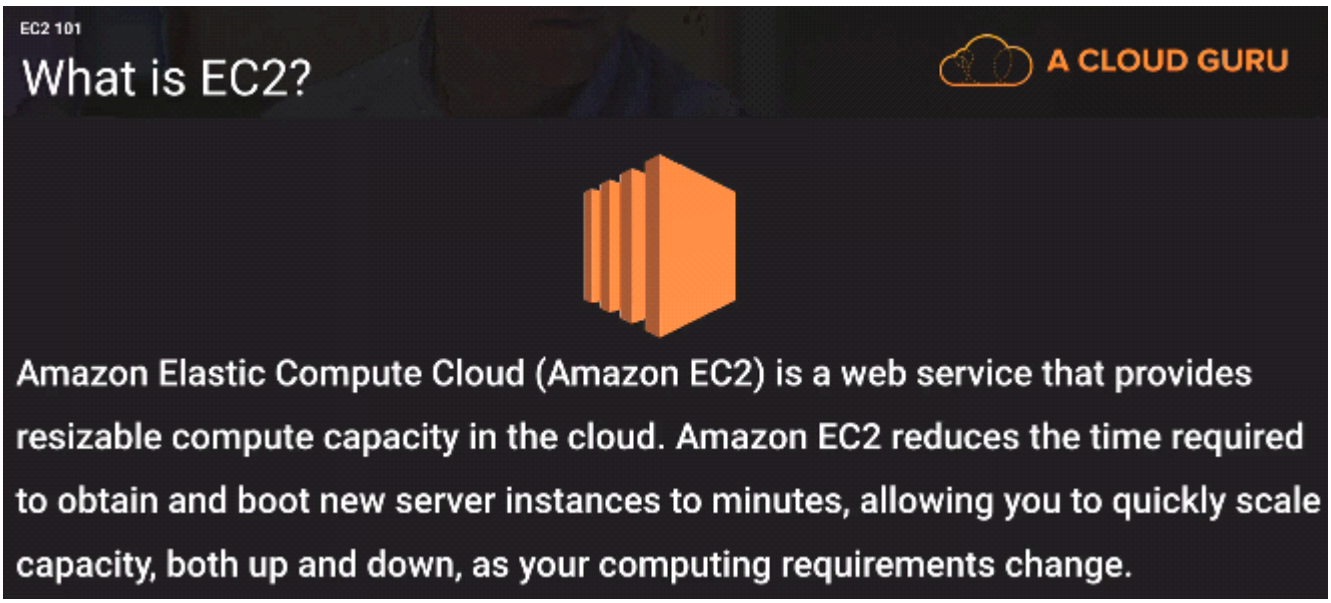
- The monthly compute price is $0.00001667 per GB-s and the free tier provides 400,000 GB-s.
- Total compute (seconds) = 3M * (1s) = 3,000,000 second
- Total compute (GB-s) = 3,000,000 * 512MB/1024 = 1,500,000 GB-s
- Total compute – Free tier compute = Monthly billable compute GB- s
- 1,500,000 GB-s – 400,000 free tier GB-s = 1,100,000 GB-s
- Monthly compute charges = 1,100,000 * $0.00001667 = $18.34

EC2 - 101

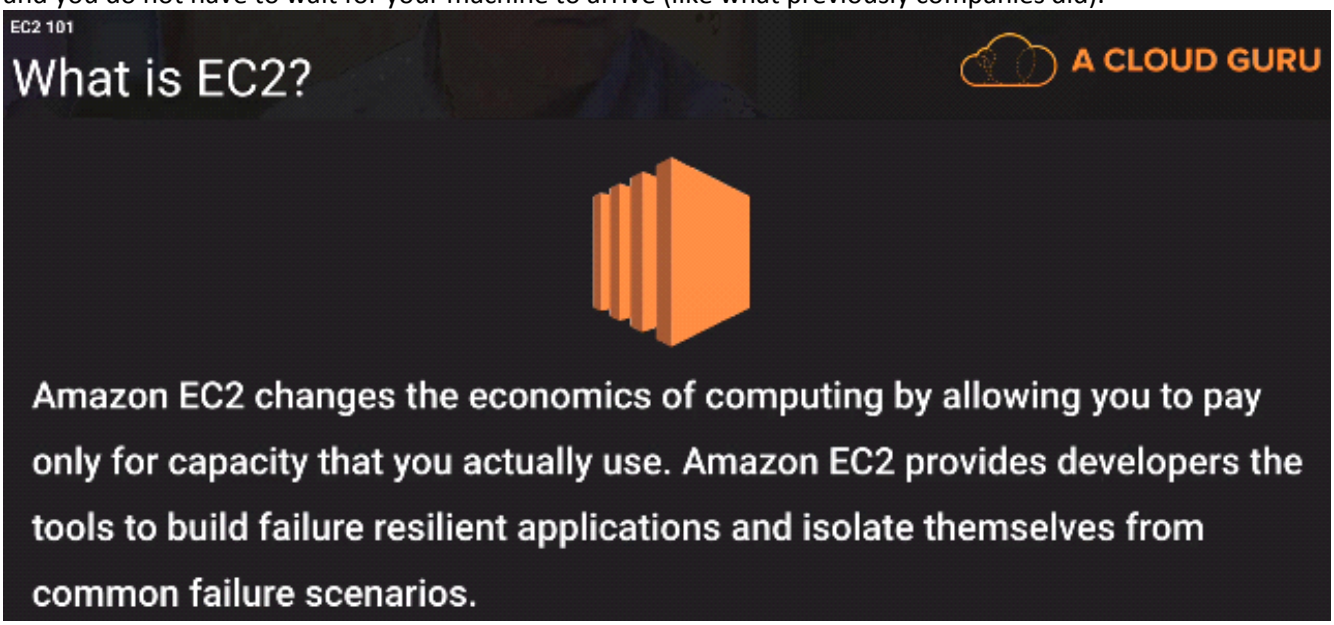
Wednesday, December 21, 2016 5:27 PM

EC2 - It stands for Elastic Compute and this is backbone of AWS system , basically we use this as for a virtual machine.

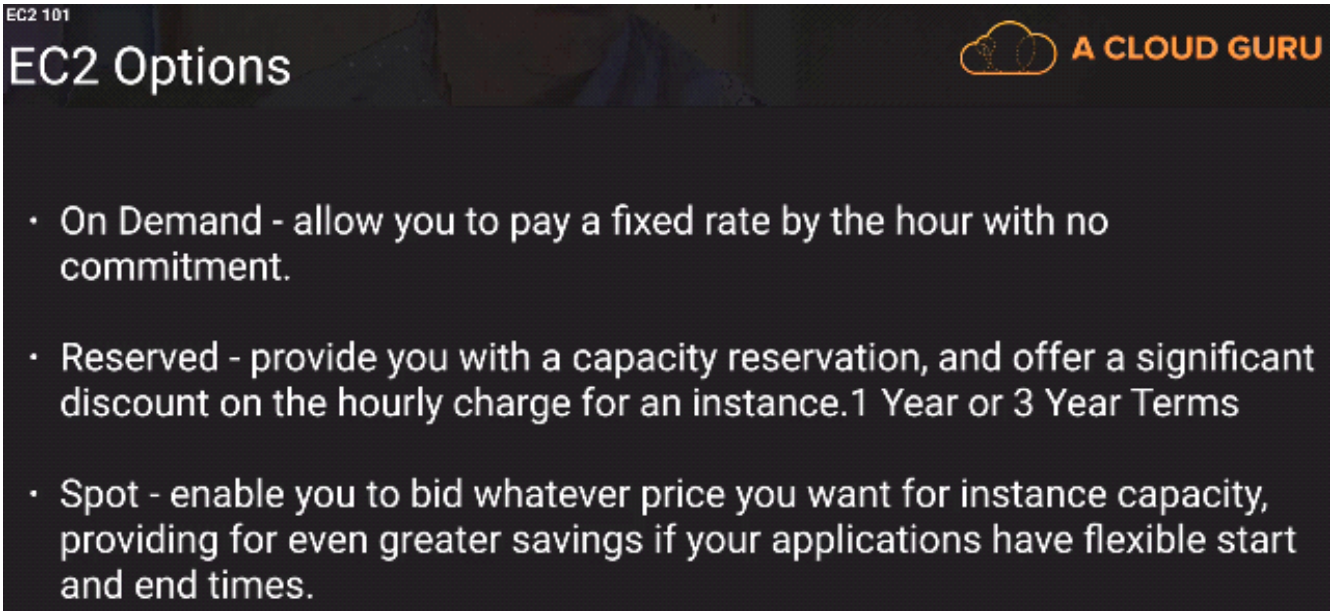
Definition -



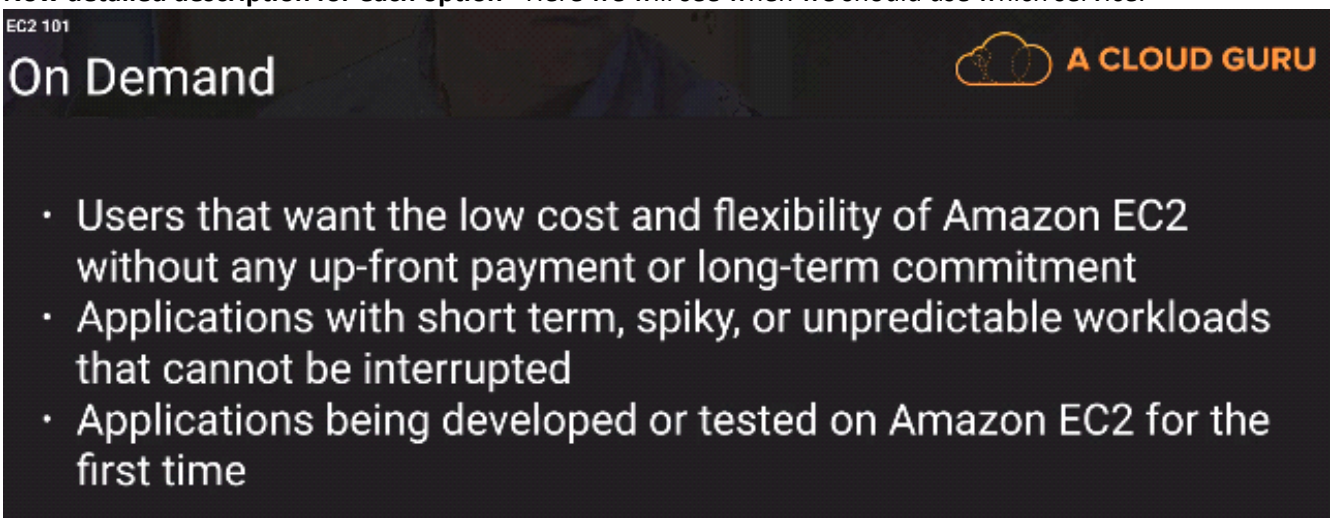
Best thing about this is that, it is instant go service like you can start using with this service in minutes and you do not have to wait for your machine to arrive (like what previously companies did).

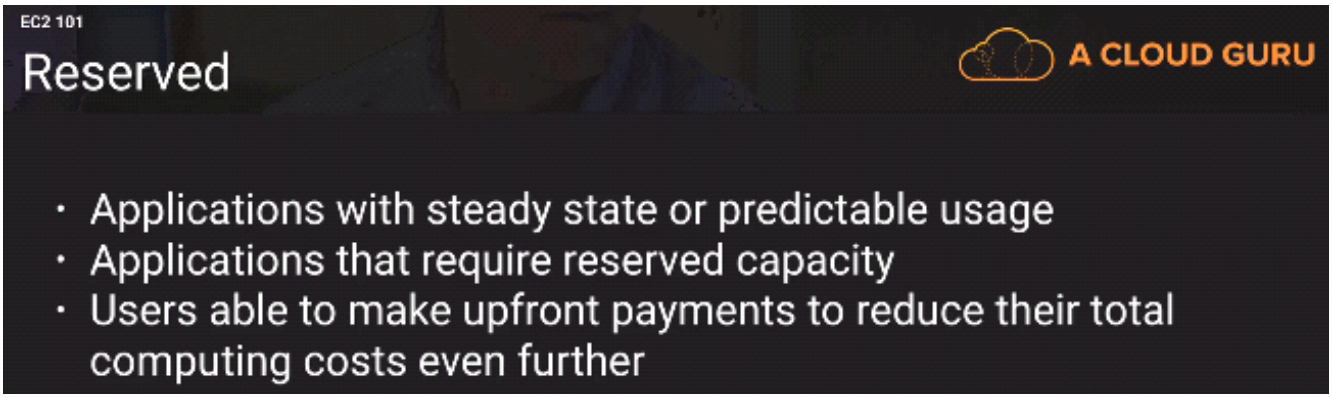


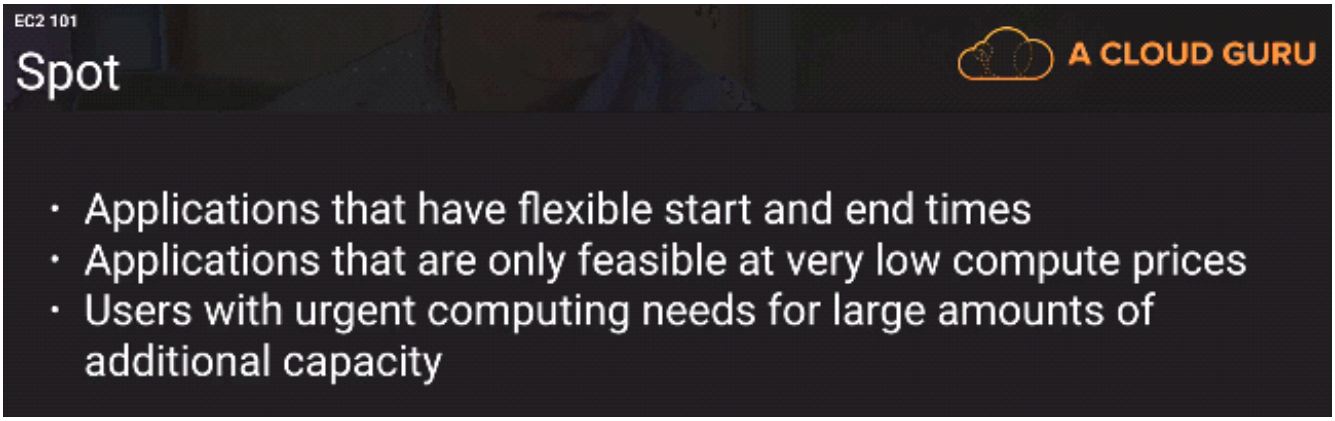
In order to prepare for exam we must know different pricing models, here is the detail for that -

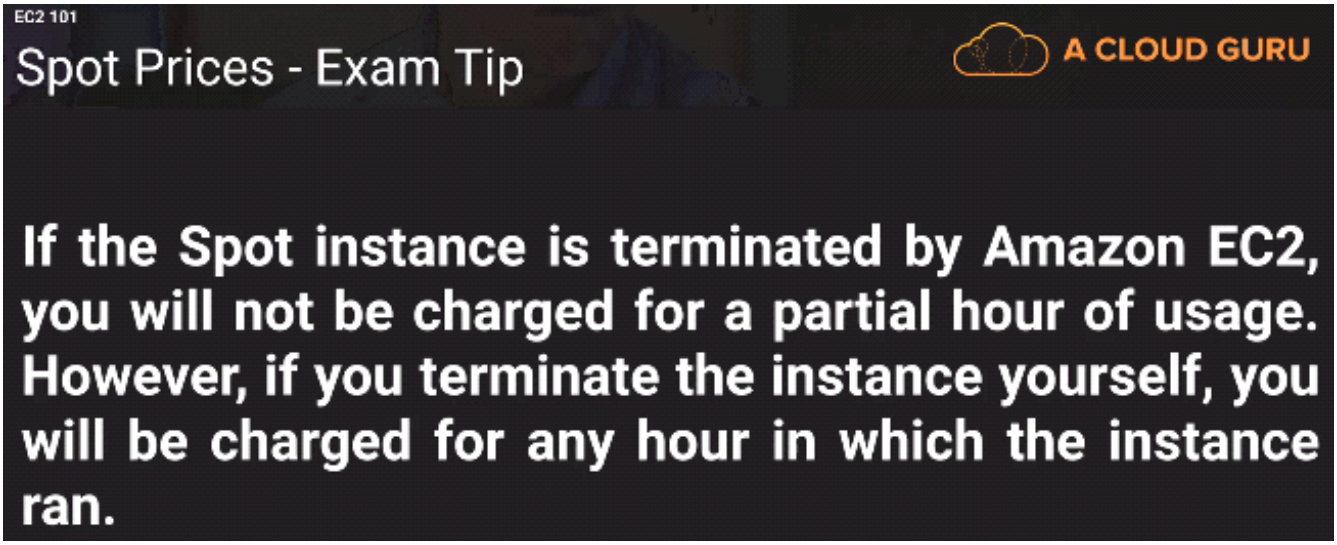


Now detailed description for each option - Here we will see when we should use which service.





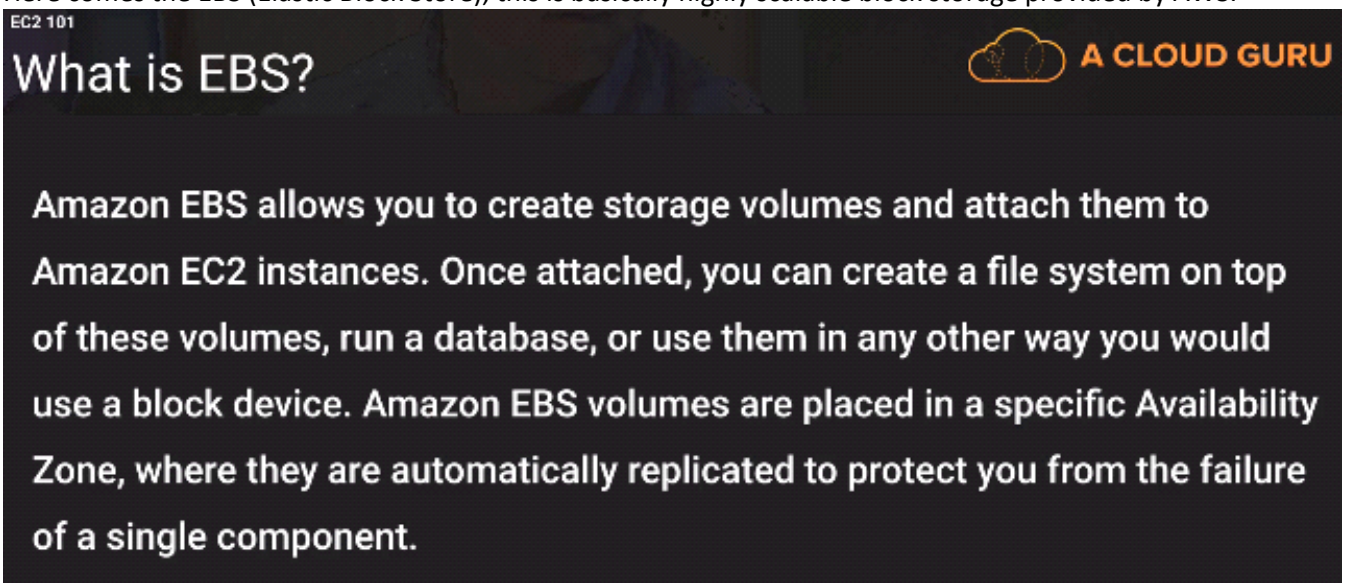




EC2 instance Types (Just for information)-

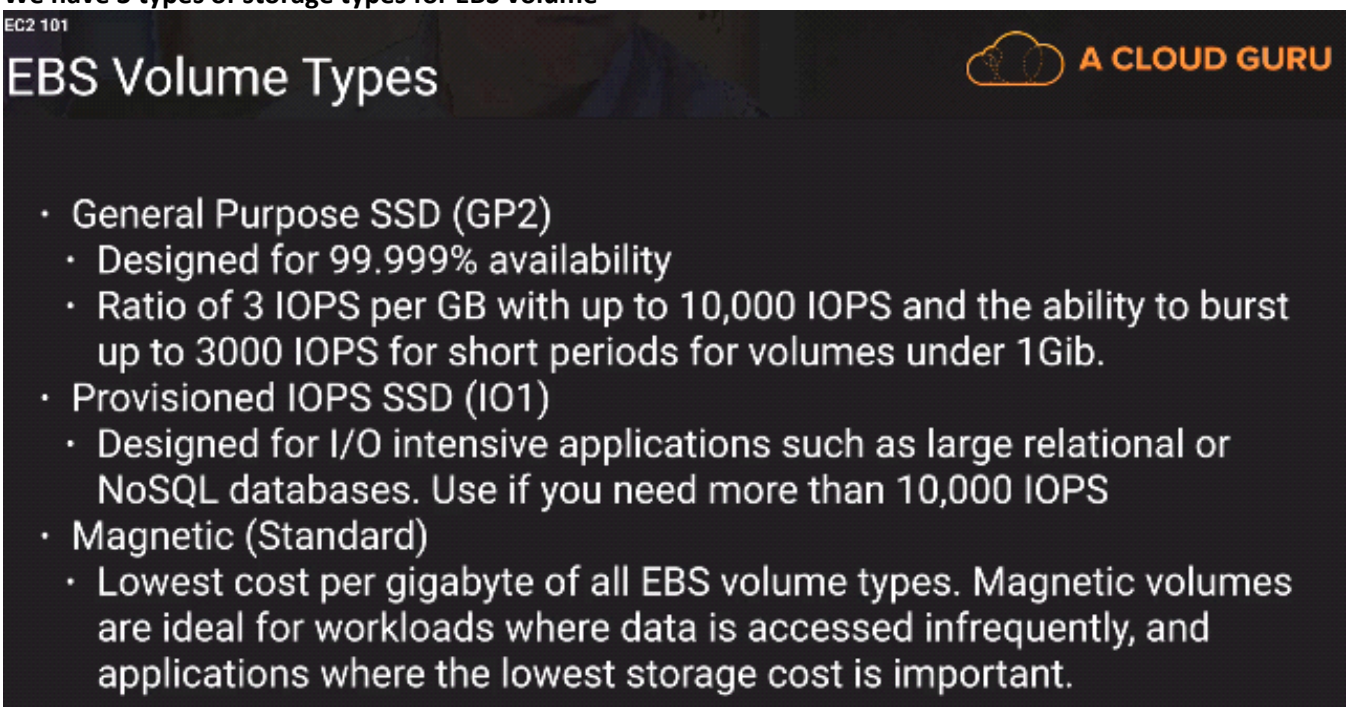
Family	Speciality	Use case
T2	Lowest Cost, General Purpose	Web Servers/Small DBs
M4	General Purpose	Application Servers
M3	General Purpose	Application Servers
C4	Compute Optimized	CPU Intensive Apps/DBs
C3	Compute Optimized	CPU Intensive Apps/DBs
R3	Memory Optimized	Memory Intensive Apps/DBs
G2	Graphics/General Purpose GPU	Video Encoding/Machine Learning/3D Application Streaming
I2	High Speed Storage	NoSQL DBs, Data Warehousing etc
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop

Now we are having our own EC2 machine, which we can choose based on our work and demand, but now we have to attach volume to our VM.
Here comes the EBS (Elastic Block Store), this is basically highly scalable block storage provided by AWS.

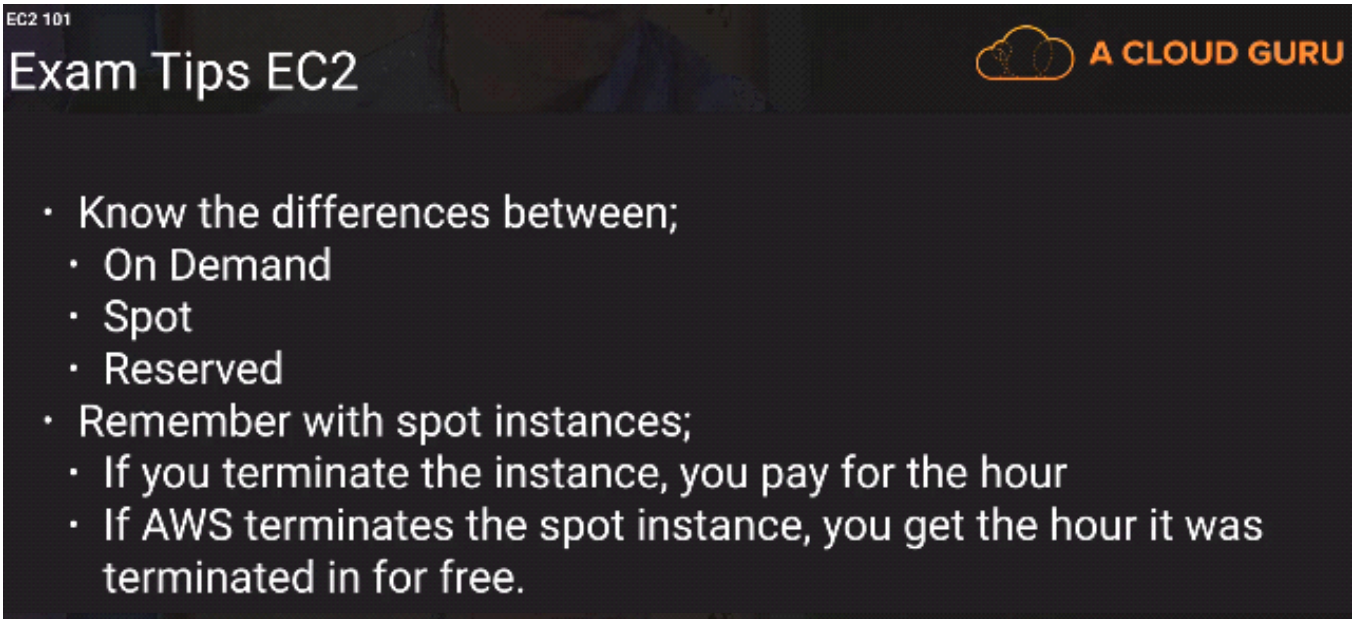


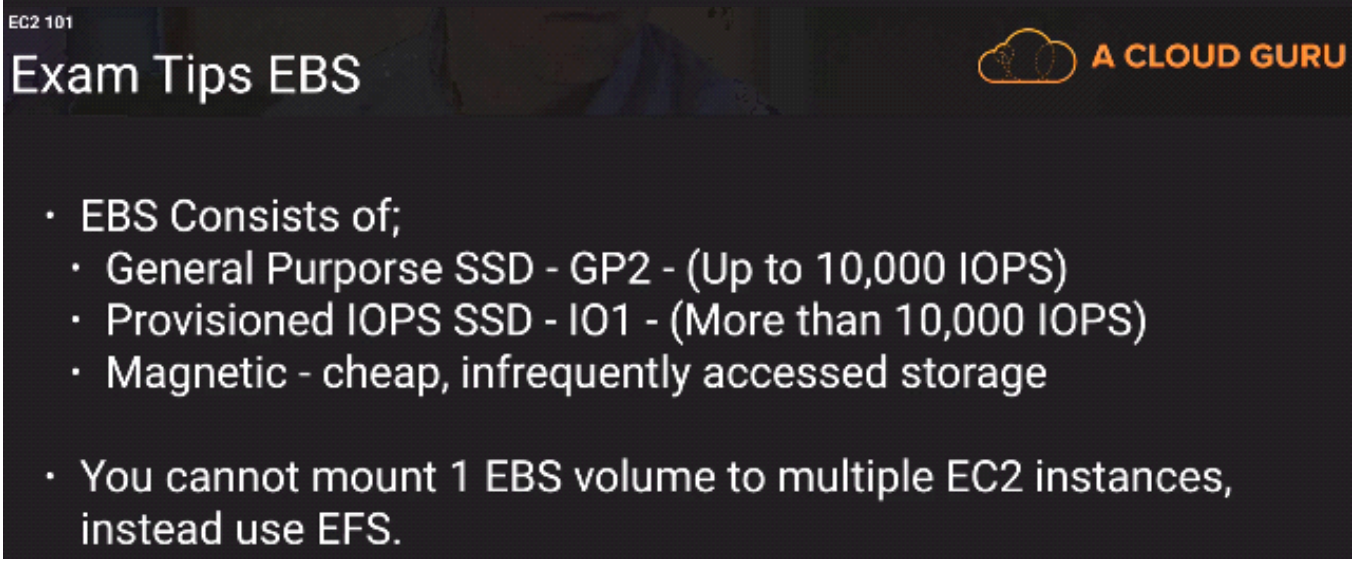
One thing to remember here is that you cannot attach one EBS volume to more than one EC2 instance, but you can have multiple EBS volume to one EC2 machine.

We have 3 types of storage types for EBS volume-



Exam Tips -


EC2 101
Exam Tips EC2
A CLOUD GURU
• Know the differences between;
• On Demand
• Spot
• Reserved
• Remember with spot instances;
• If you terminate the instance, you pay for the hour
• If AWS terminates the spot instance, you get the hour it was terminated in for free.

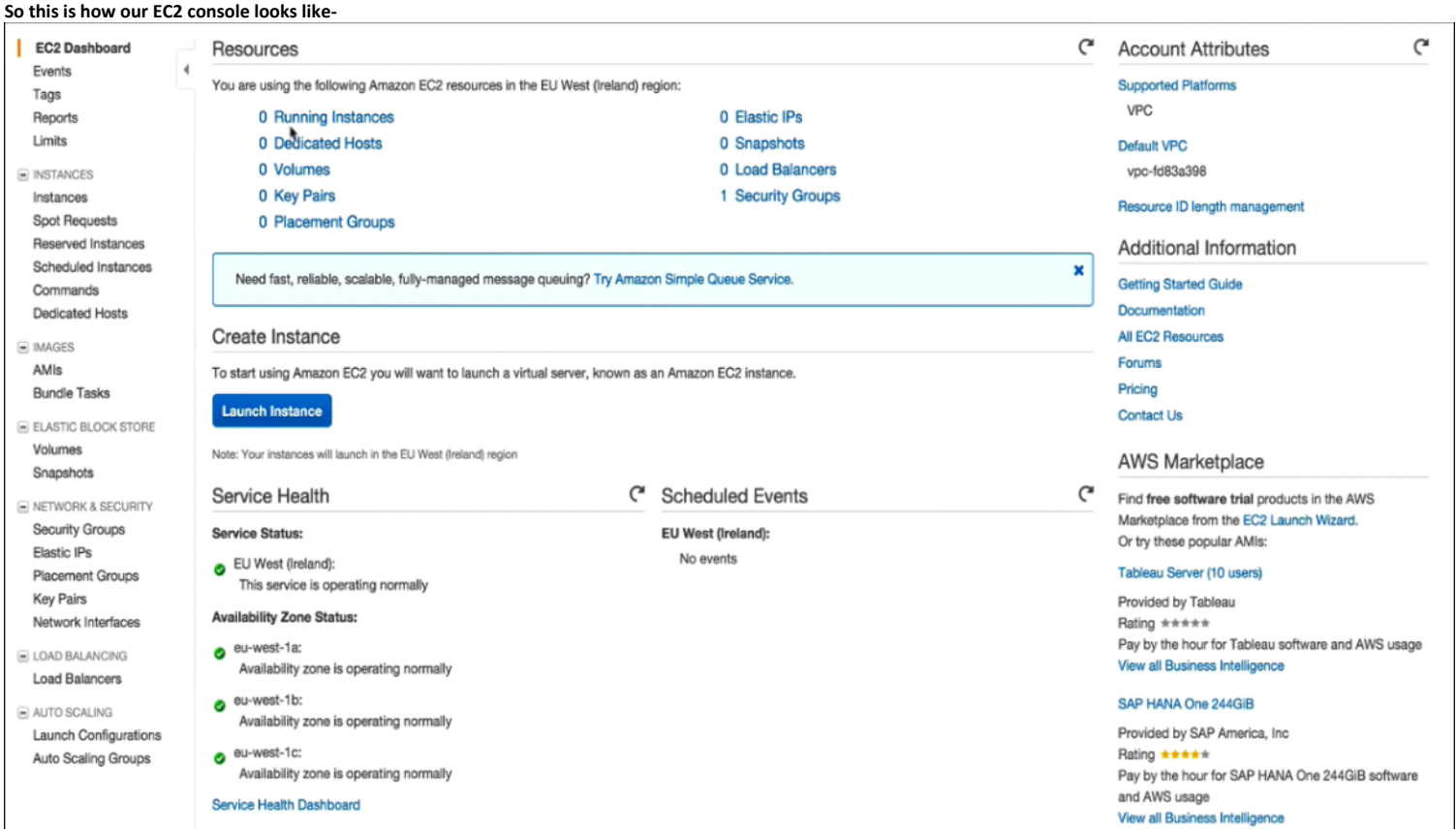

EC2 101
Exam Tips EBS
A CLOUD GURU
• EBS Consists of;
• General Purporse SSD - GP2 - (Up to 10,000 IOPS)
• Provisioned IOPS SSD - IO1 - (More than 10,000 IOPS)
• Magnetic - cheap, infrequently accessed storage
• You cannot mount 1 EBS volume to multiple EC2 instances, instead use EFS.

EC2 - launch Instance - Lab

Friday, December 23, 2016 3:30 PM

Here we will see how to create your own EC2 instance, how to create snapshot from that EC2 instance.

So this is how our EC2 console looks like-



Left panel is basically menu for EC2, we are having various in this sections, like - instances, images, EBS, Network and security, Load Balancing, Auto Scaling (We will learn about these things later on).

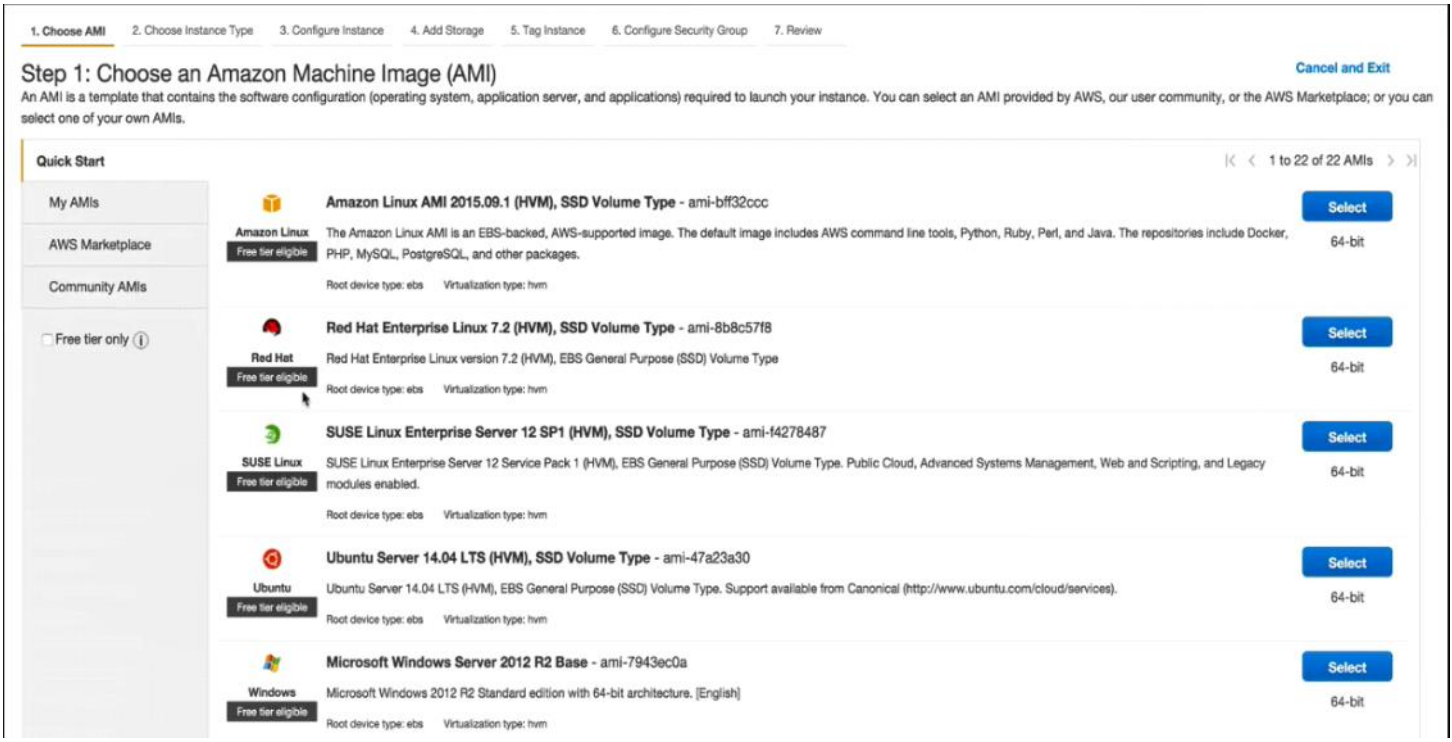
Right Panel is basically support provided by AWS, so that we can get to know how to use this.

Middle Panel - It is basically divided into 3 parts,

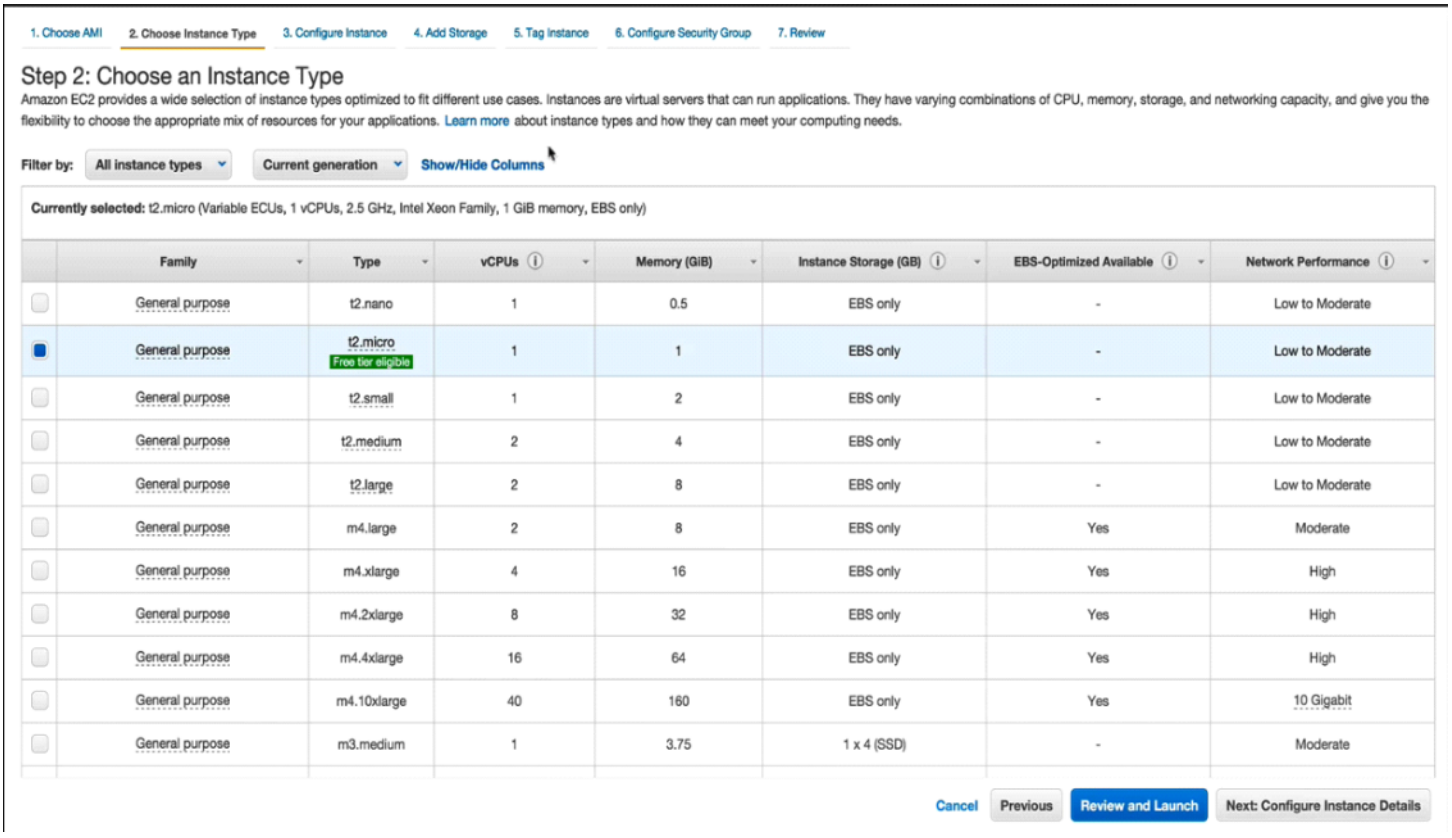
Resources - Current running Instance, Volumes and other list, what we are using currently.
Create Instance - If we want to create new instance, we can do this from here.
Service Health - ?????

So let's go and create a new instance for our EC2 machine.
Hit on the Create instance Button on the Middle Panel.
Now here in order to create our VM, we have to choose our machine's specs, like OS, application, memory, Disk Space.
Steps to create VM-

1. Choose an amazon VM - Here we have list of OS that we can choose to work with, here AMI means amazon creates this machine instances with prebuild applications in it, so that we do not have to create those application.
   Here we have on the left panel option for selection VM from category too -
   Now just go and selects what so ever AMI you want to work with, here we are choosing amazon Linux AMI.



2. After that it will ask for choosing instance type. Here we have list of all types of instances with detailed information about vCPUs, Memory and Disk Space as well.
   Here we are choosing t2.micro and then we hit on the right down button "Next: Configure Instance Details".



3. Here we have option to Configure our Instance Details - Here we have following options,
   Number of Instances - how many EC2 machine you want to run.
   Purchasing option - we have seen in the types of payment, we had "Spot instance", we can avail that here (But it is not present in t2.micro, since it is free tier).
   If we choose Spot Instance option then we see following options
   Price - we have current and Maximum price that we pay for our EC2 machine.
   Launch Group - ????

Time - Here we have request valid from and request valid to , like we can set from when to when our machine should be working.

If we uncheck Spot Instance then we see following option -

Network - what network on we are working, whether it is created default, or we can even set our own VPC to work with it. (VPC will be covered in next to next chapter)

Subnet - Choose this in which availability zone you want to have access, choose default to work with any availability zone to work on that region.
(we will cover this in VPC).

Auto assign Public IP - Here we can assign Public IP, keep this default.

IAM role - use your created IAM role that you created while working with IAM.

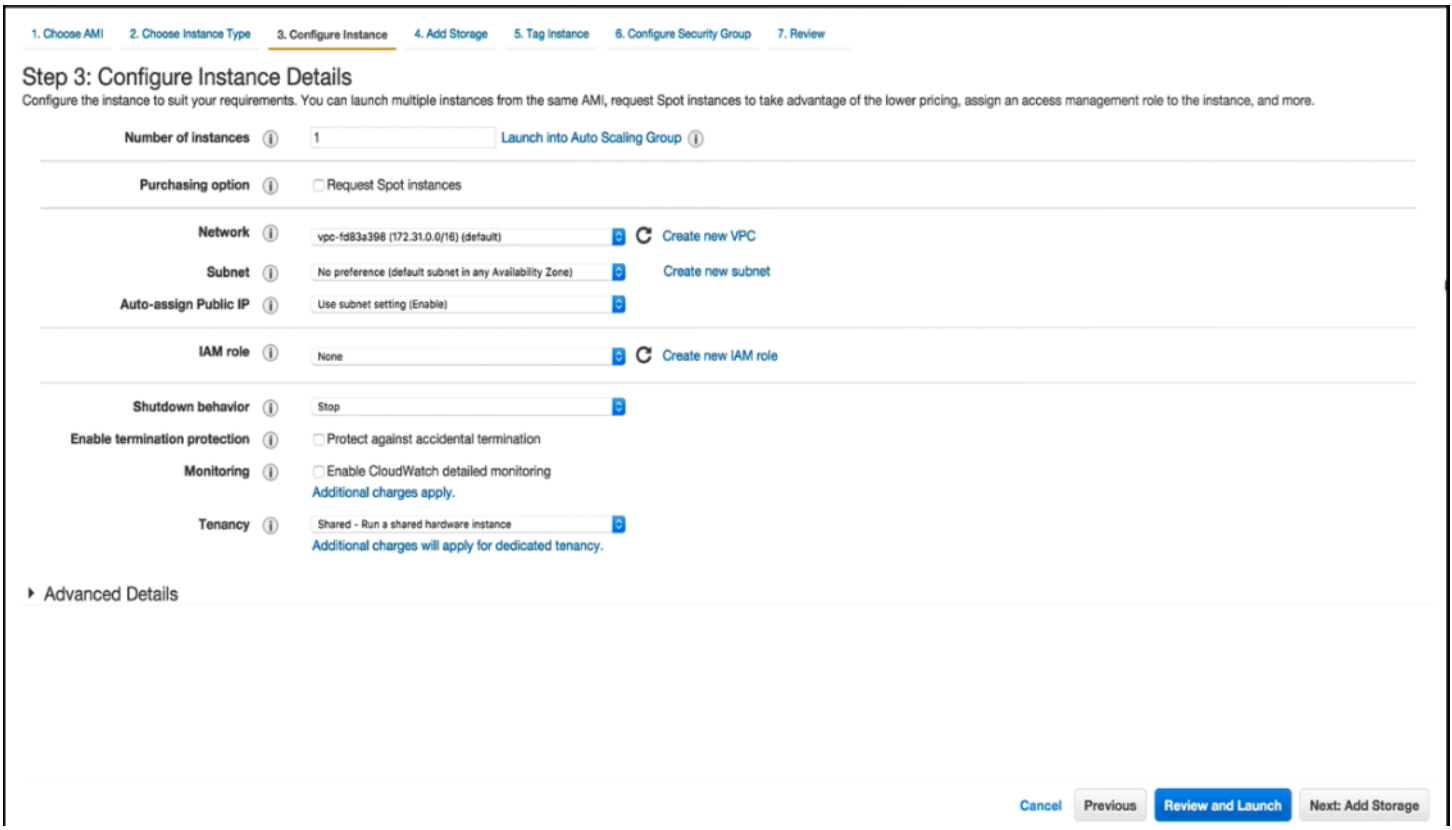
Shutdown behavior - Here we can choose whether if we shutdown our machine then what should it do, ,here we have 2 option - stop or terminate.

Enable Termination Protection - if our machine is not working properly, then it saves us from accident termination, on the other hand if someone wants to terminate this machine, so it will give a prompt to save our machine termination.

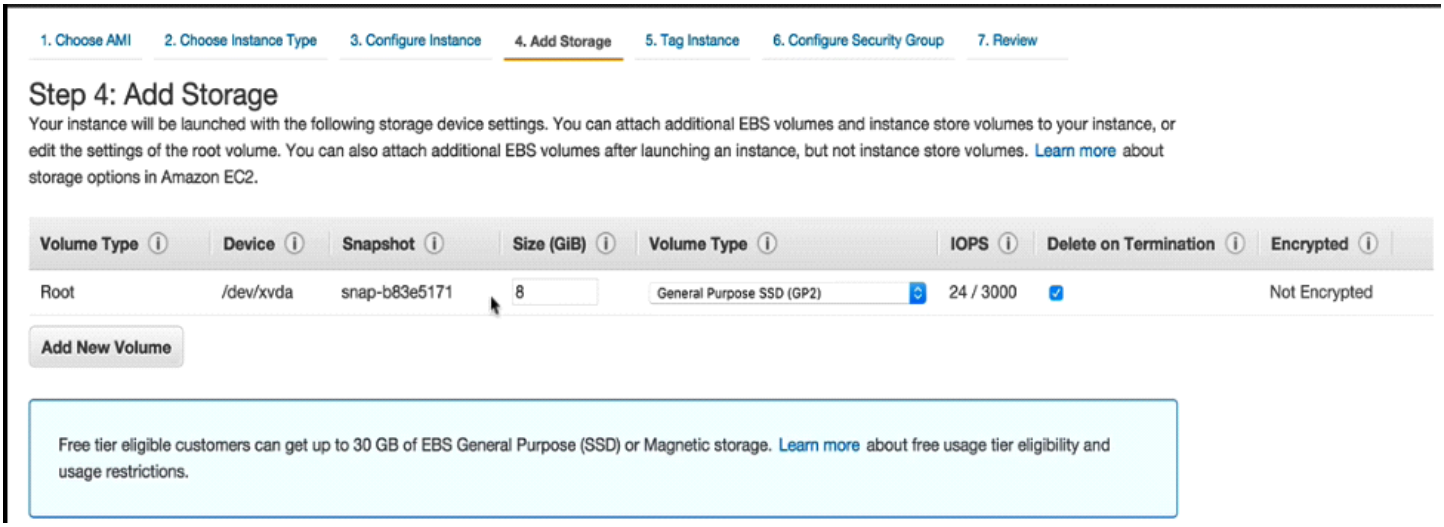
Monitoring - Here we can monitor our machine, with the help of Cloudwatch, although we have some basic Cloudwatch features but this gives us detailed Cloudwatch monitoring but for this extra charges may apply.

Tenancy - Whether we want to run this on shared instance or not , we will cover this later.

We also have **Advance Details** for this as well, here you can pass your script, when you first start your instance, this script will run with your machine.

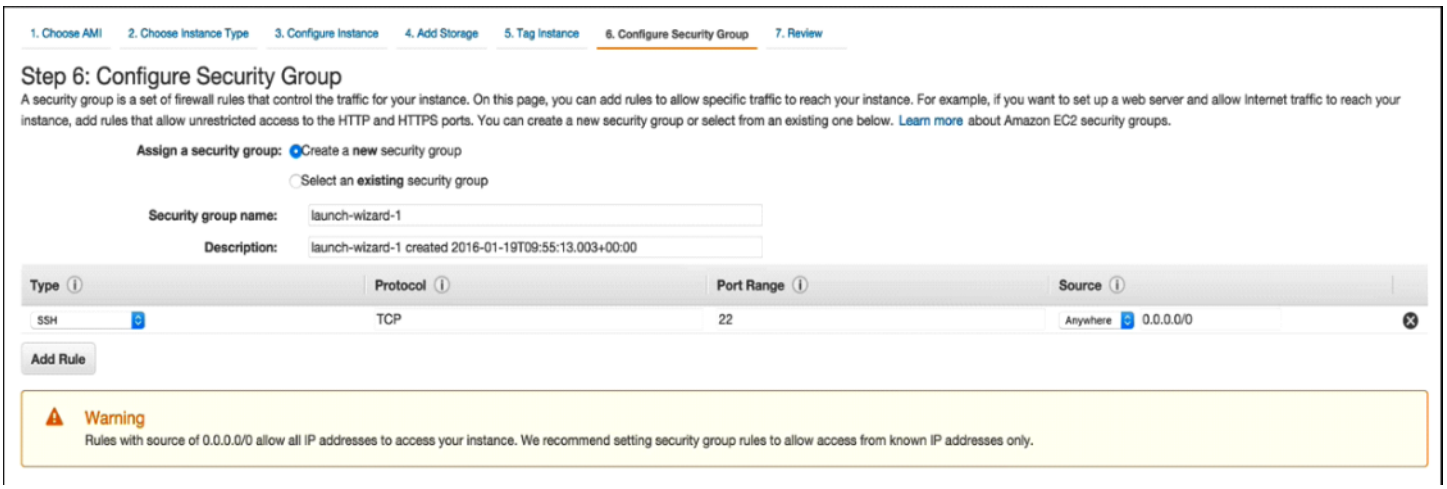


4. **Add Storage -** Here we can add storage to our EC2 machine from this page, here by default we have Root as our Volume Drive and this is basically the drive where our OS is installed, here for the Volume we have list for the Volume drive what we are adding,
Here we have volume type, Device, Snapshot, Size, Volume Type, IOPS, Delete on Termination, Encrypted.
You can understand these things as well.
One thing to Notice here is that you cannot encrypt this root device, from this on you can encrypt any other volume if you want to add, if you want to encrypt this Root Volume then you have to encrypt this using third party tool like Bitlocker.

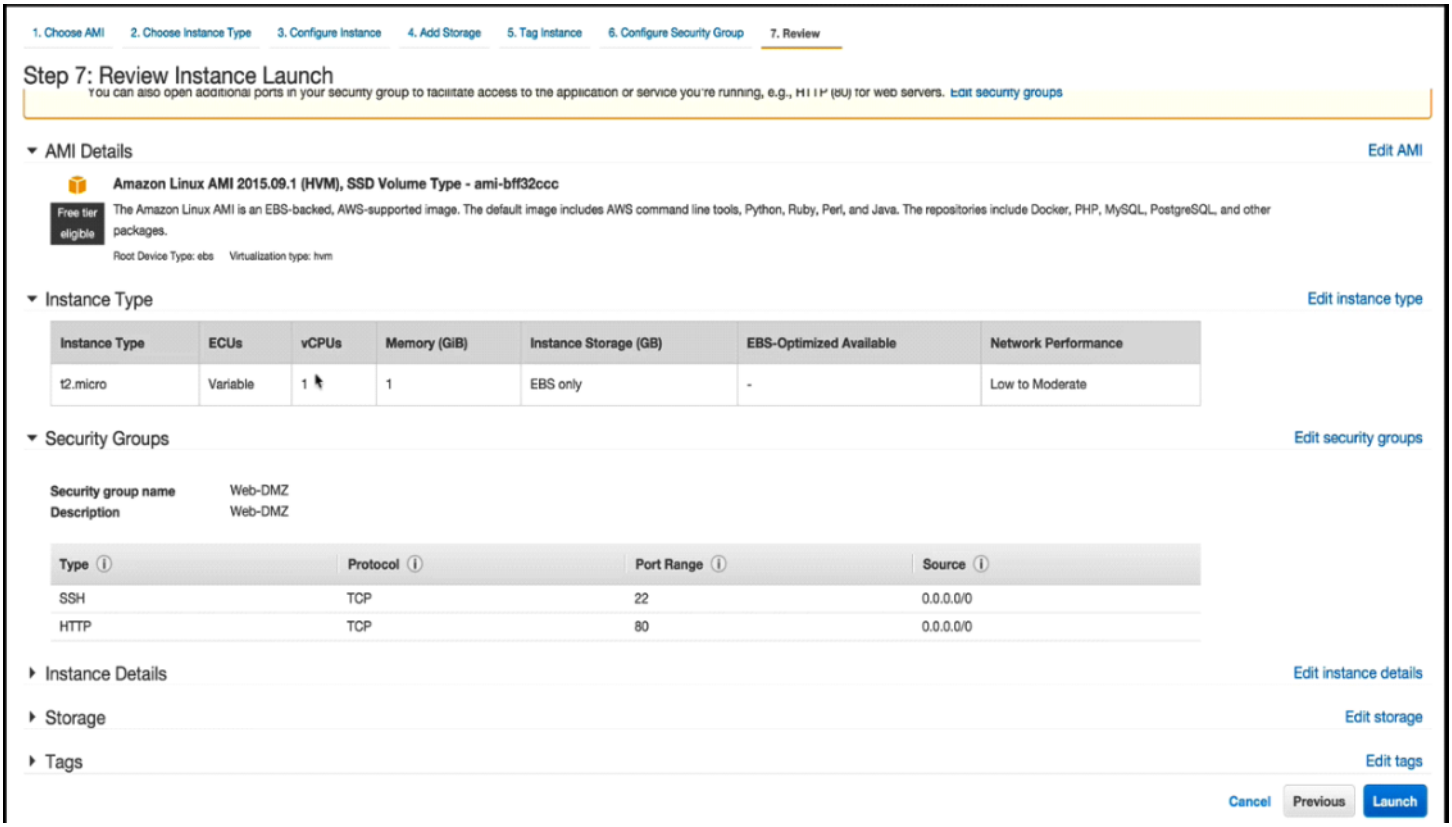


5. **Tag Instance -** In this step we tag our instance, I have no idea why we use this maybe we will see this in upcoming videos.

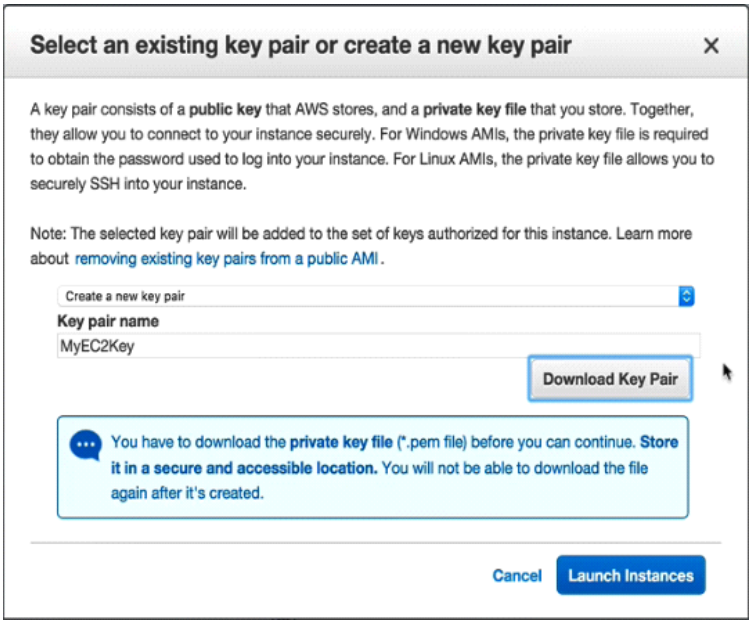
6. **Configure Security Group -** This is basically a virtual Firewall for our EC2 instance, Here we have type of connection for incoming connection, by default it blocks all the incoming connection except SSH connection. Select from what connection you want to access you VM.
After setting your VM now you can go to step 7.



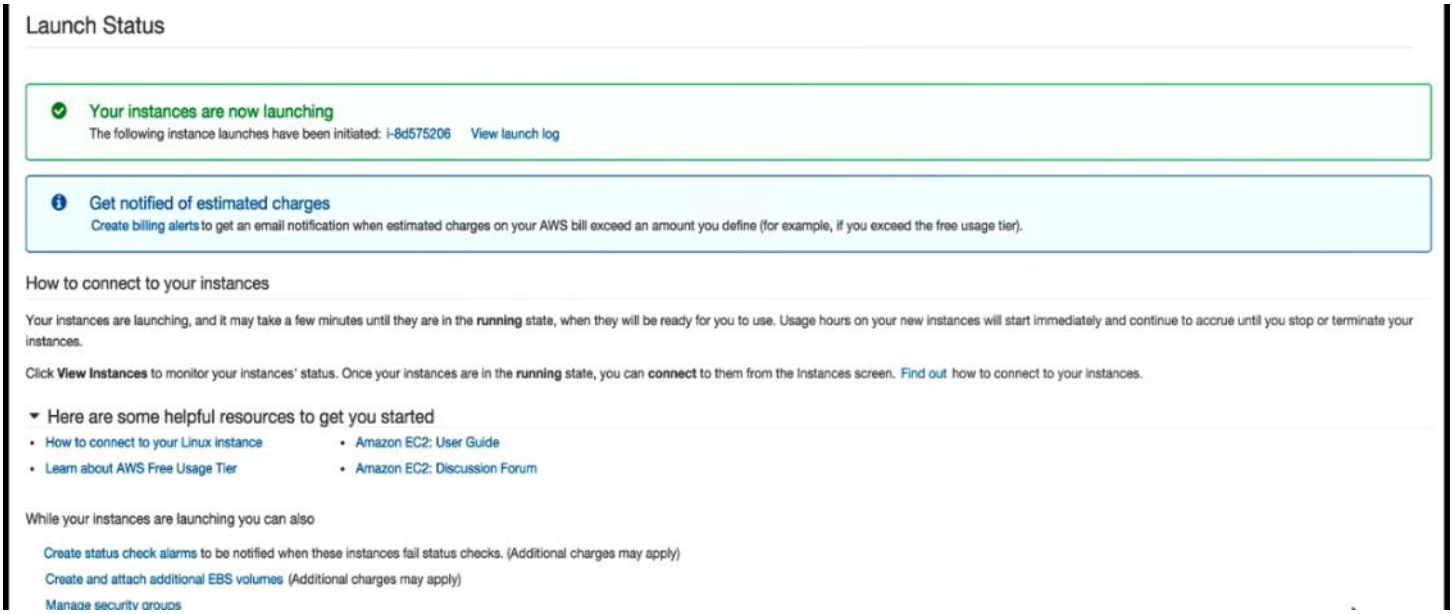
7. **Review Instance Launch -** Here we will review our instance before launching our machine, so that if we want to edit any aspect from this we can edit that from going to previous step.



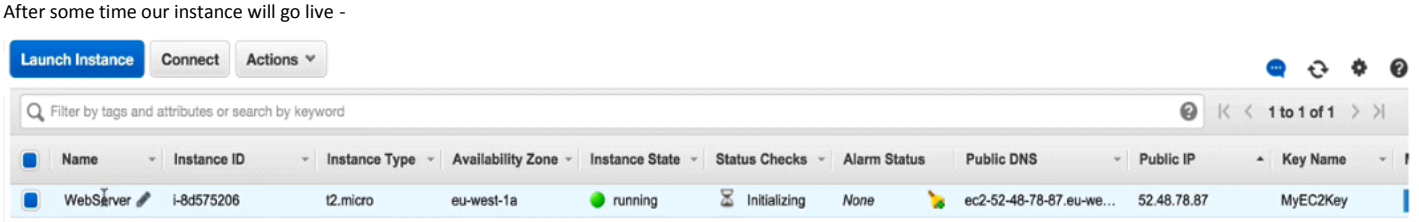
After verifying each and every step, go and hit Launch Instance.
Here a prompt will pop for saving your private key pair, that will be used to connect to your EC2 machine via SSH connection.



Basically we call this private key, because generally private key is used to open lock and public keys are used to close any gateway with those locks.
When you will hot on the launch instance, then you will see following window -

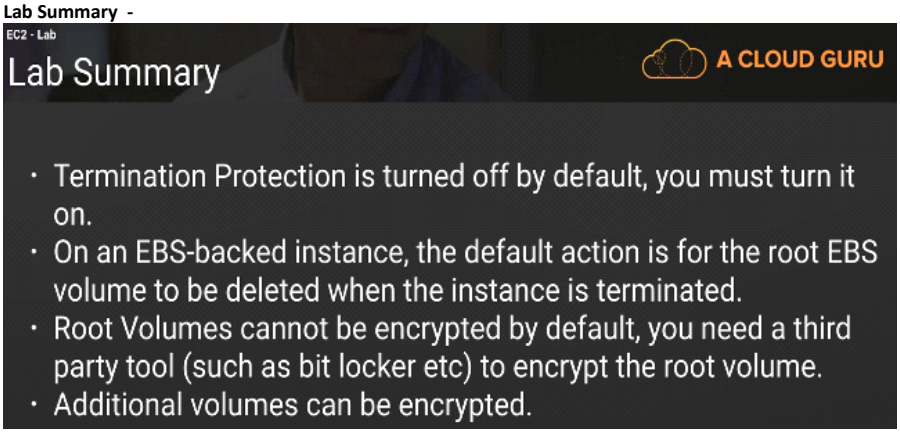


Now it will take some time to launch your instance, it will take around 1-2 minutes to create your instance, Up and Running.
After some time our instance will go live -



Here we can see our instance IP(which is in public IP) from where we can login to our system.
Now if we want to create our snapshot, delete volume , or edit our instance, we can do that from EC2 console.

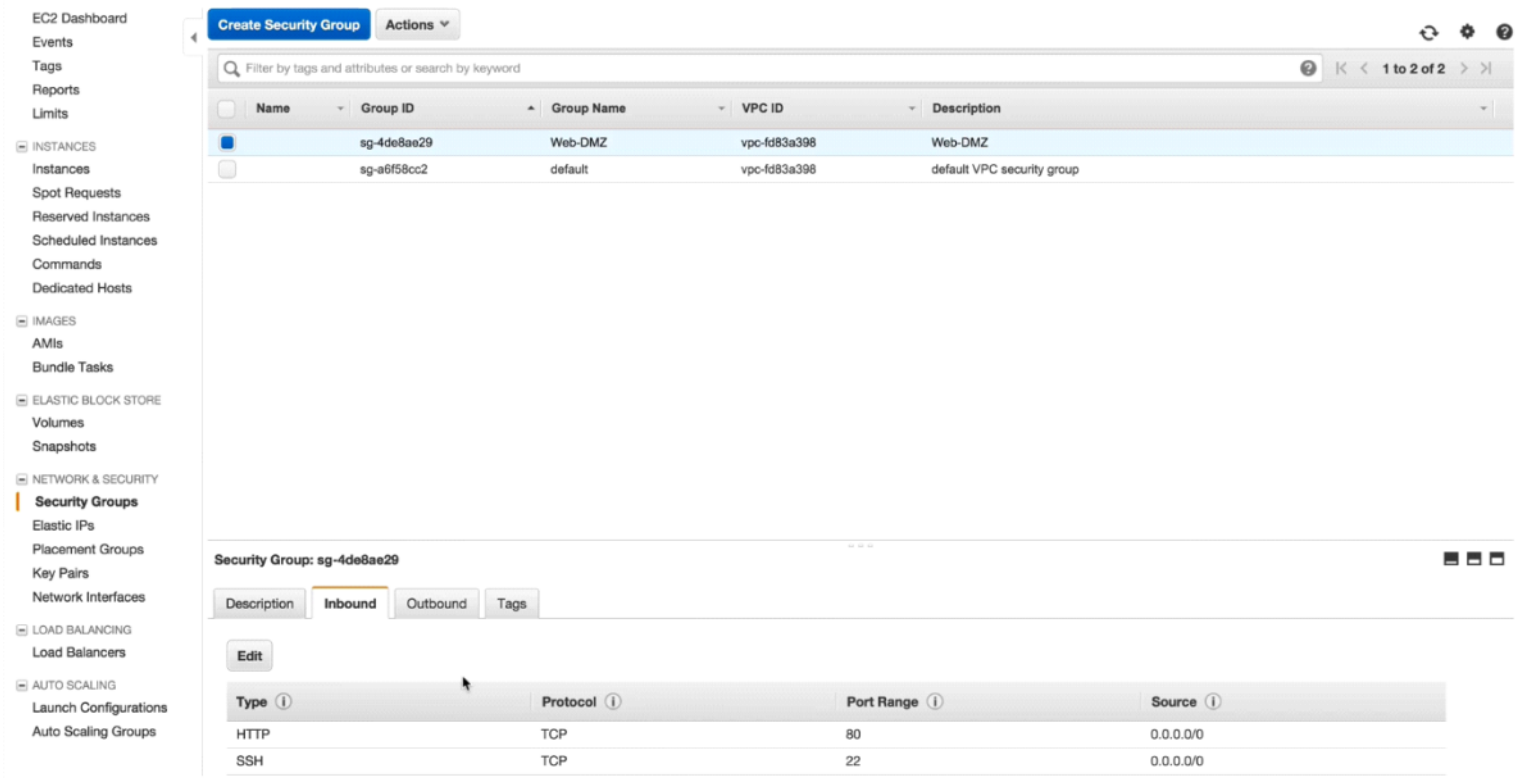
Lab Summary -



EC2 - Security Groups

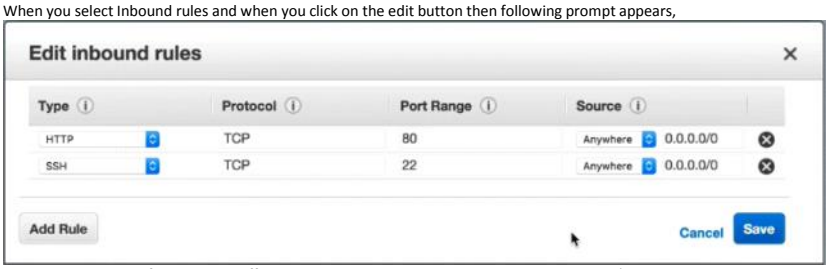
Monday, December 26, 2016 12:11 PM

Basically security Groups are for traffic control, with the help of this we can control that what kind of protocols are accessible for inbound and outbound traffic.
By default all the inbound traffics are blocked and all the outbound traffics are available.
When you click on the security groups from the EC2 console main page, you will see list of your security groups.



Here we have details for our security groups at bottom of the page, we have -
- Description - Detailed description about our security group.
- Inbound - what kind of connections are allowed
- Outbound - what kind of connection are blocked.
- Tags - different key and value pairing.

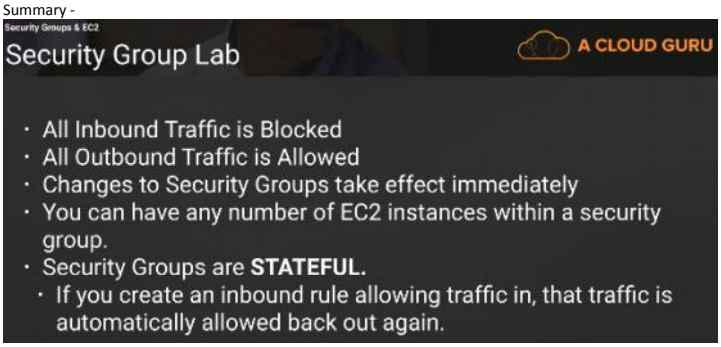
When you select Inbound rules and when you click on the edit button then following prompt appears,



Here we can add rule for inbound traffic, on the source button we have 3 options, anywhere (user can access from anywhere), only from current machine IP (you can access this machine only from your IP), custom (you can set your custom IP).

When you edit any security filter it takes effect immediately (there is no delay for this).
One thing to notice here is that you cannot deny any rule in Inbound tab and for the Outbound tab if none type of connection type is selected by default then it is allowed by default, but if you choose any type of connection then only that connection is allowed by default.

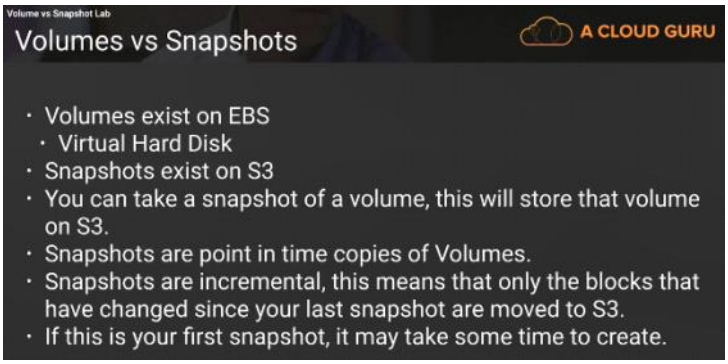
Summary -



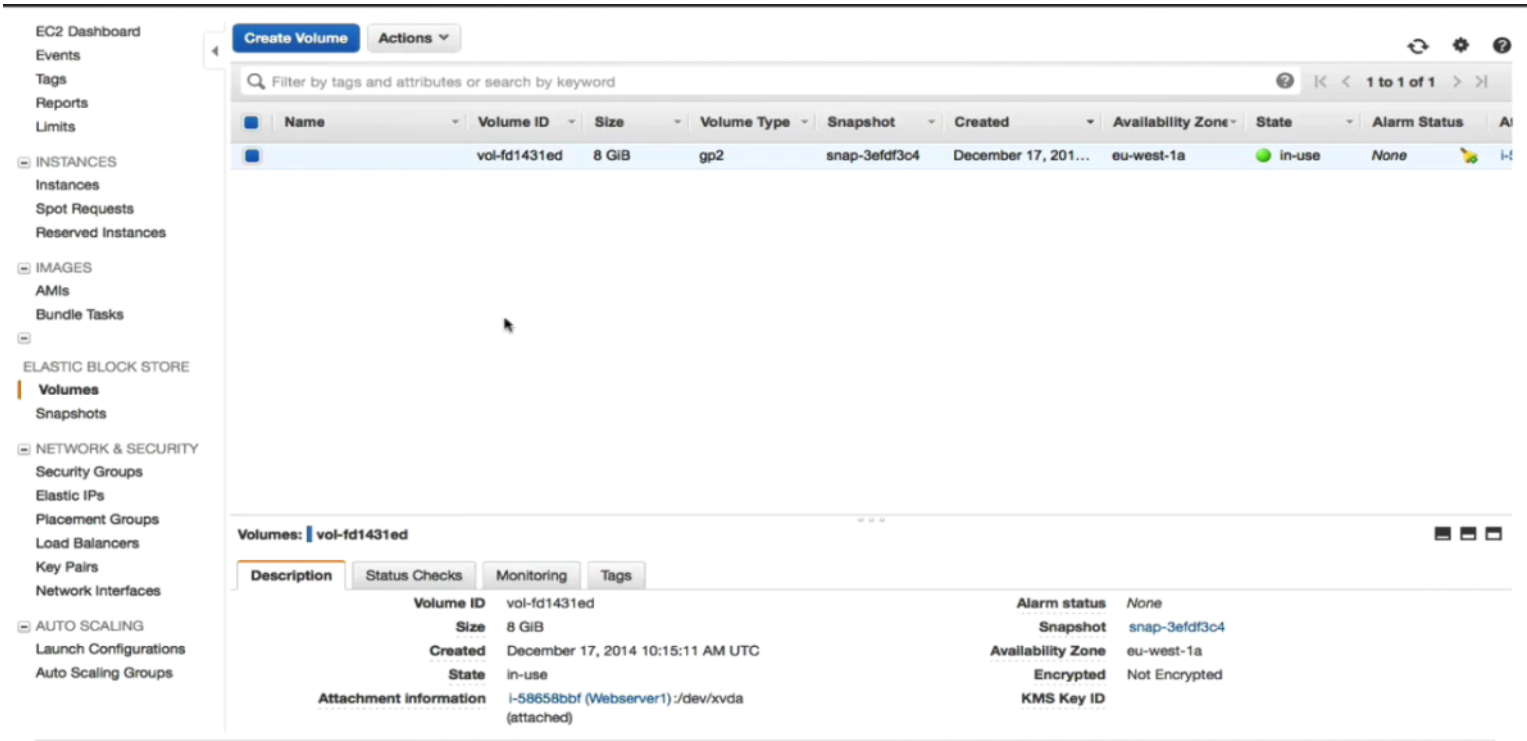
EC2 - Volume vs Snapshots

Monday, December 26, 2016 12:36 PM

Here we will learn about what is the different between volume and snapshot.

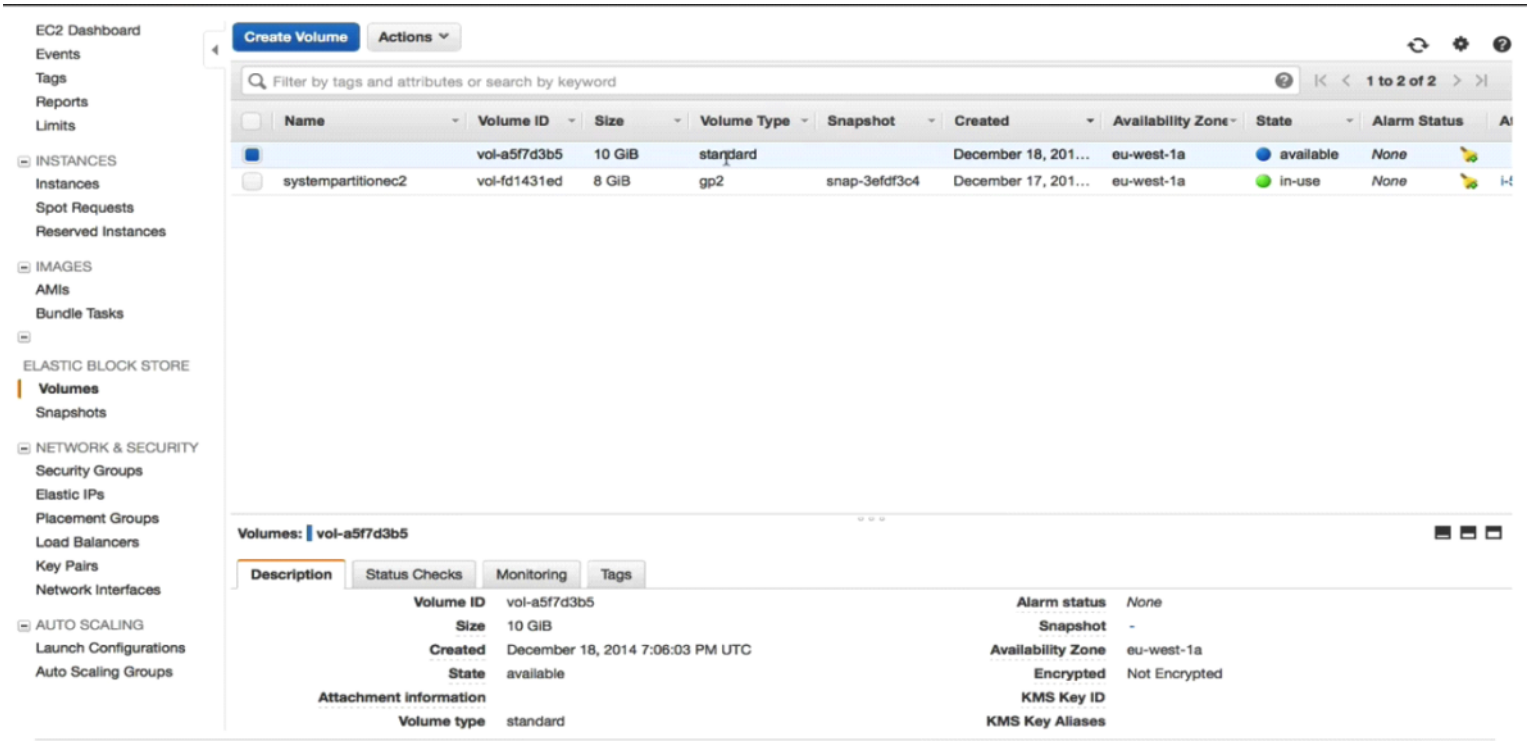


Here first of all we will learn how to attach new volume to our machine ->
On our EC2 console we will click on the volume button, which will take us to new windows -

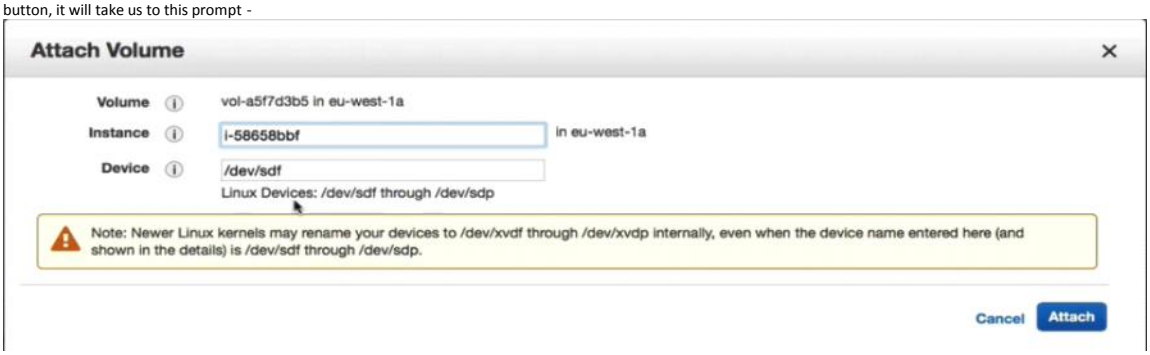


Here we have following information about our volume, ,basically this is root volume of our running EC2 instance.
Now in order to add more volume to our system, first of all we have to create new volume for our system, we will click on the **Create Volume** button, which will take us to new window -
Here we will create a magnetic disk type volume.

Now hit on the create button and it will take some time to create our volume, next step is to attach this volume to our EC2 machine, when our volume will get created, then following screen will appear -

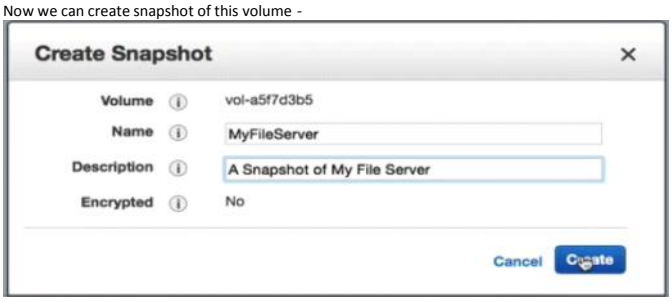


Here now we will click on the Actions drop down button and there we will click on the attach volume button, it will take us to this prompt -

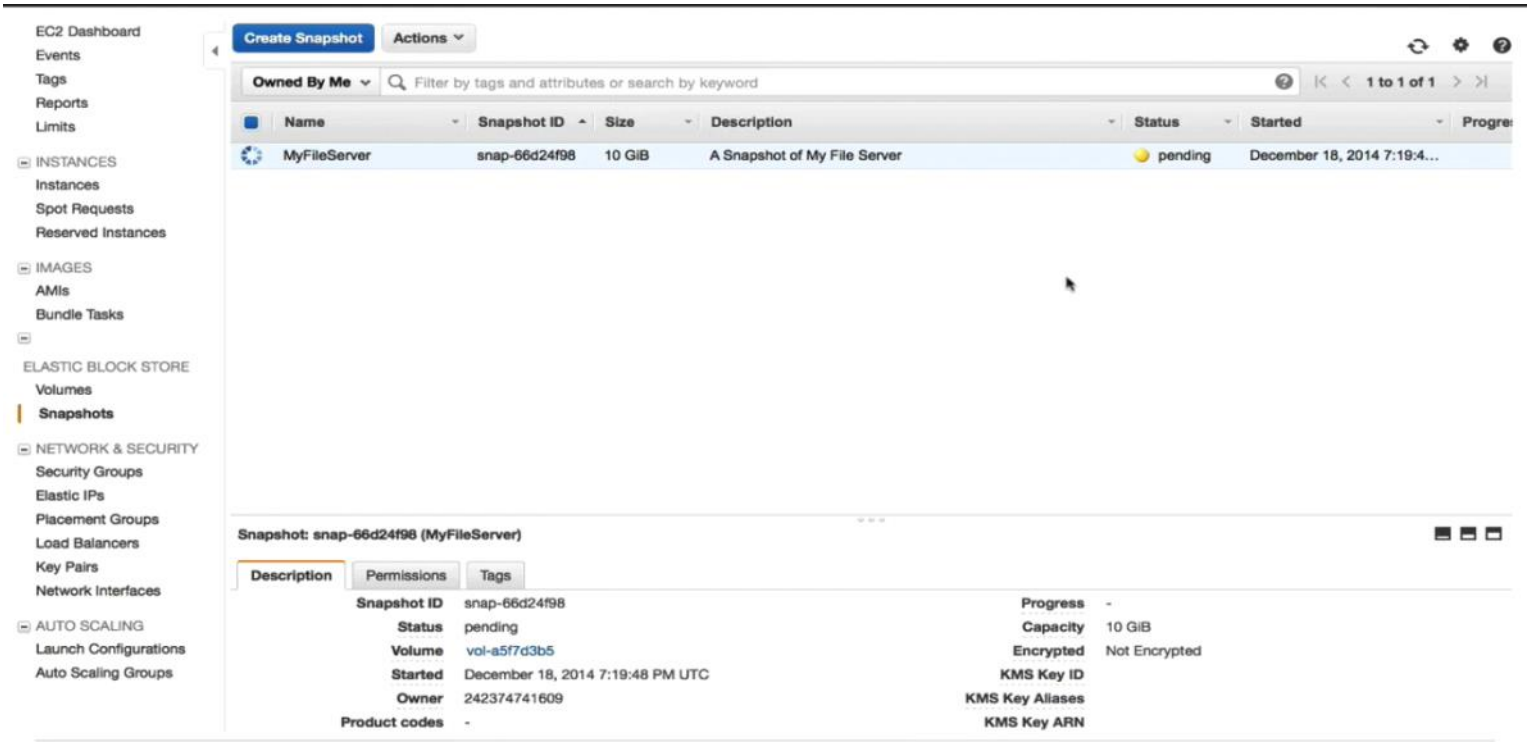


Here we will have to provide our EC2 instance and on the Device it will provide us where this volume will get added on our machine, now we will hot on the attach windows and it will take some more time to attach our volume to our EC2 machine.

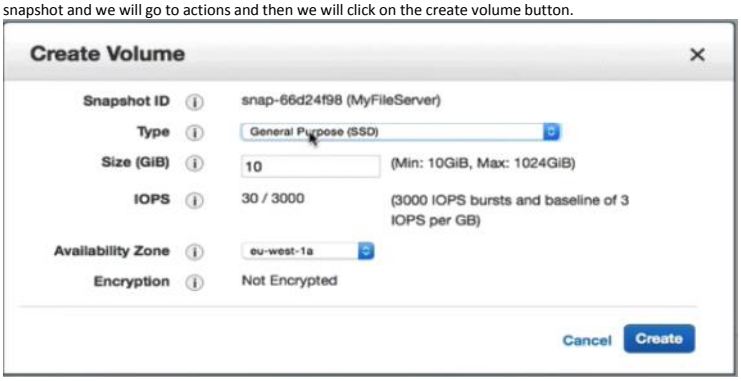
Now we will detach this volume from our EC2 machine, here again we select out volume and then on actions button we will hit on the detach volume - it will take some time to perform this operation. Now this volume is free and is not attached to any EC2 machine.
Now we can create snapshot of this volume -



Now when we click on the snapshots on the EC2 console's left panel's EBS section, we will see that our snapshot is being created. Since this is first time we are creating snapshot of this volume then it will take some time to get created.

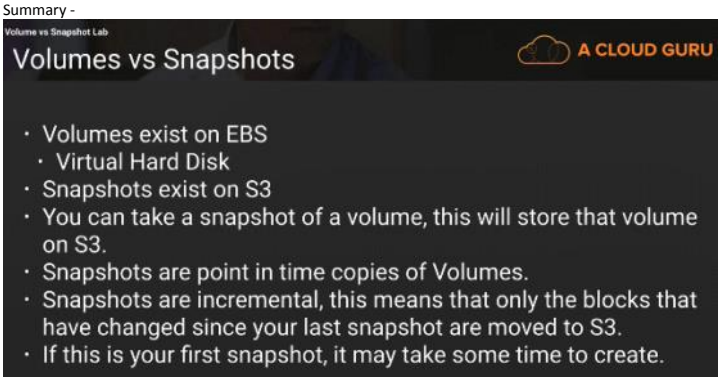


Now since we have snapshot of our volume we can delete our volume, and later on we can restore our data from that snapshot.
Now we will restore our volume from our created snapshot, from the snapshot page we will select our snapshot and we will go to actions and then we will click on the create volume button.

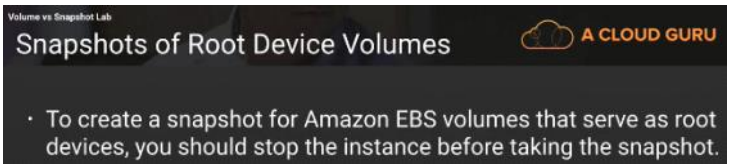


Again we can go from here as what we have done while creating new volume, here one thing to notice is that when we have created our volume at that time we took our volume type Magnetic but now from that snapshot of that volume we have all the three options to choose, which is good to use, since we can do testing of our volume in magnetic and we can restore this disk on our server with maybe general purpose or high IOPS SSD volume.
One more thing to notice is that it is so fast to create volume from snapshot of volume.

Summary -



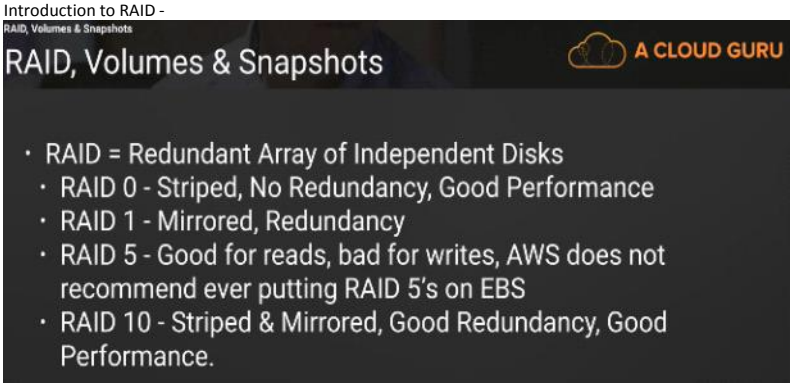

Volume vs Snapshot Lab
Volumes vs Snapshots - Security
A CLOUD GURU
Snapshots of encrypted volumes are encrypted automatically.
Volumes restored from encrypted snapshots are encrypted automatically.
You can share snapshots, but only if they are unencrypted.
These snapshots can be shared with other AWS accounts or made public


Volume vs Snapshot Lab
Snapshots of Root Device Volumes
A CLOUD GURU
To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

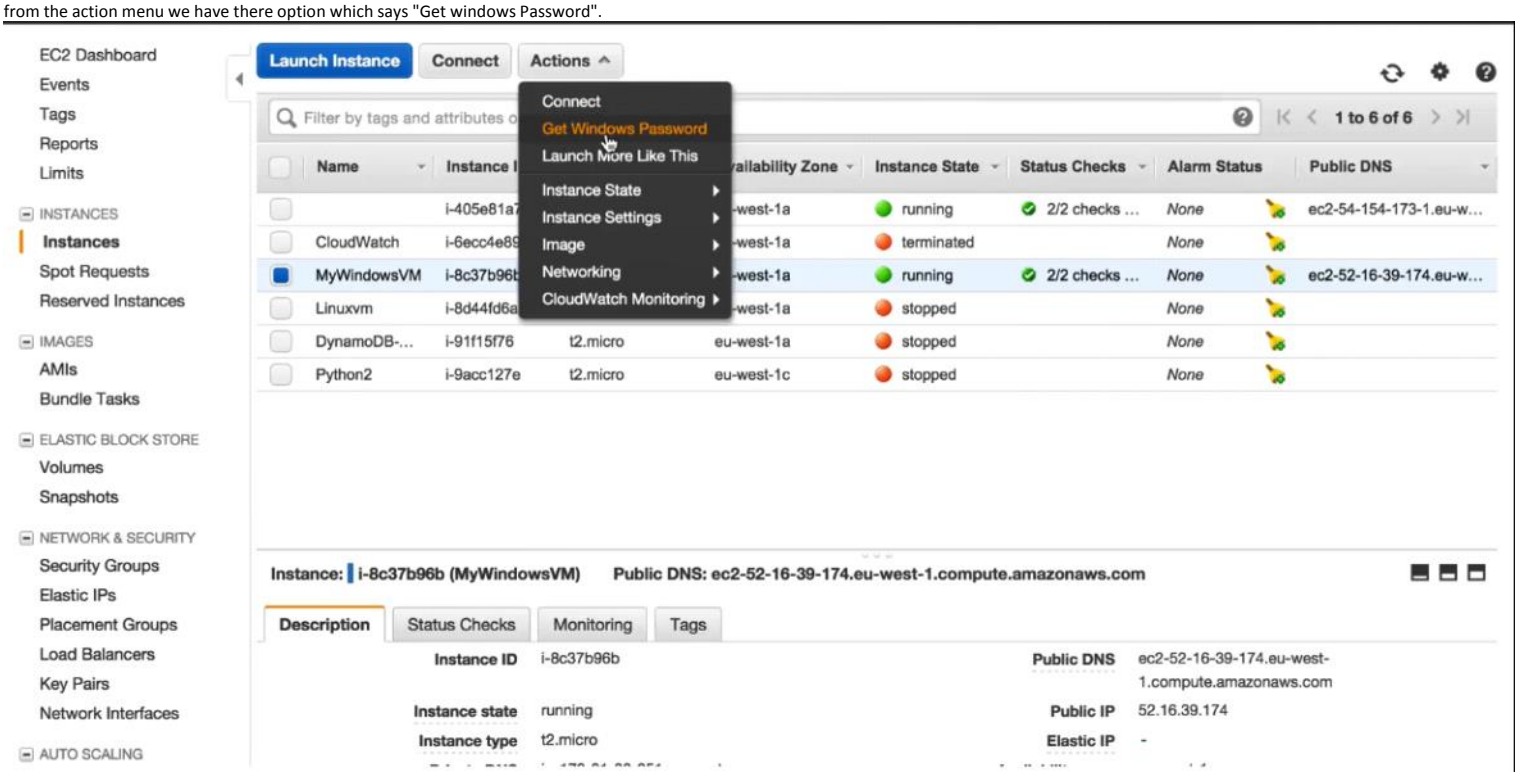
EC2 - Windows RAID, Volumes and Snapshots

Monday, December 26, 2016 1:11 PM

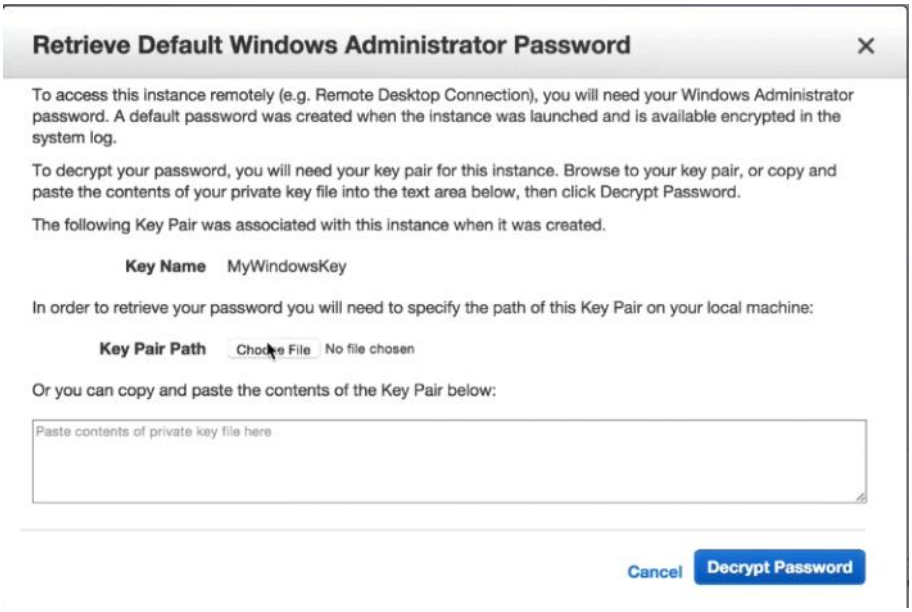
Introduction to RAID -



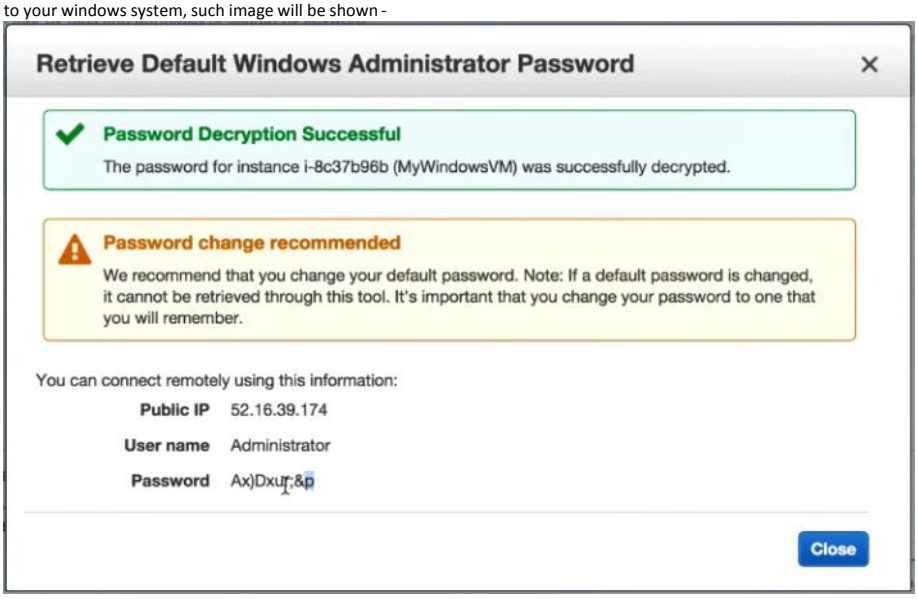
We use this RAID array when we do not get required I/O for our processing.
So basically you are taking multiple EBS volumes and you are clubbing those volumes to create a single volume of redundant array of independent disk.
So you may get exam question that you are not getting required I/O then your solution would be to get multiple EBS volume and use them as RAID volume.
In order to work with RAID, first of all we have to enable RDP inbound rule under our security filter's property section.
Now we will use windows EC2 machine, for that go to main EC2 console and from here we will once again create a new instance with windows OS.
Follow all the steps that we have used to create EC2 machine under EC2-lab.
In volume section we will have root volume, apart from that we will include 4 more EBS 8GB GP2
It will take some more time than Linux machine.
Now once our machine is ready to use, we know that in order to login to our VM using RDP (since this is our windows machine), we have to create username and password for our machine, by default the password is "administrator" now for the password we have to select our machine from EC2 console and from the action menu we have there option which says "Get windows Password".



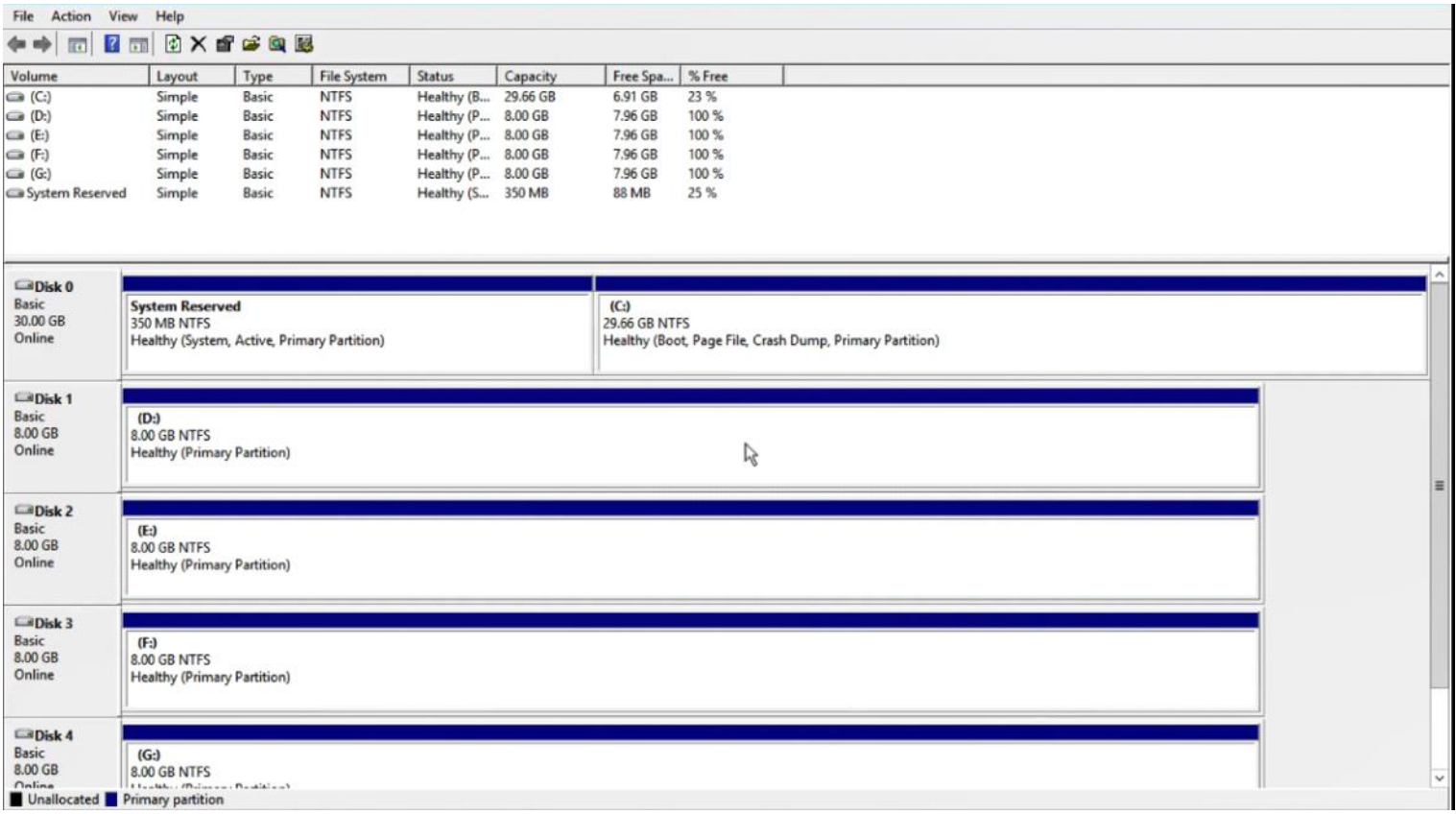
Now when you hit on the "Get Windows Password" it will take you to another prompt-



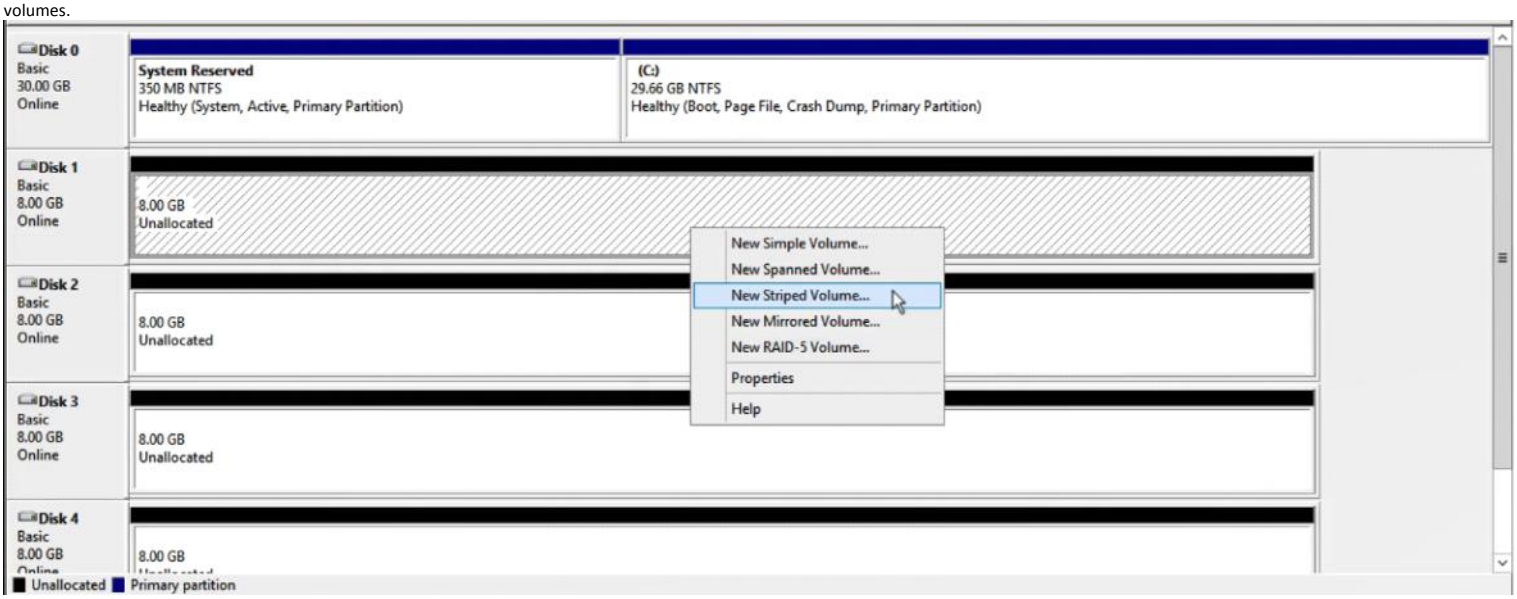
Here you have to specify your private key what you have downloaded when you created your EC2 instance.
Once you will upload your private key then it will show you your username and password for logging on to your windows system, such image will be shown -



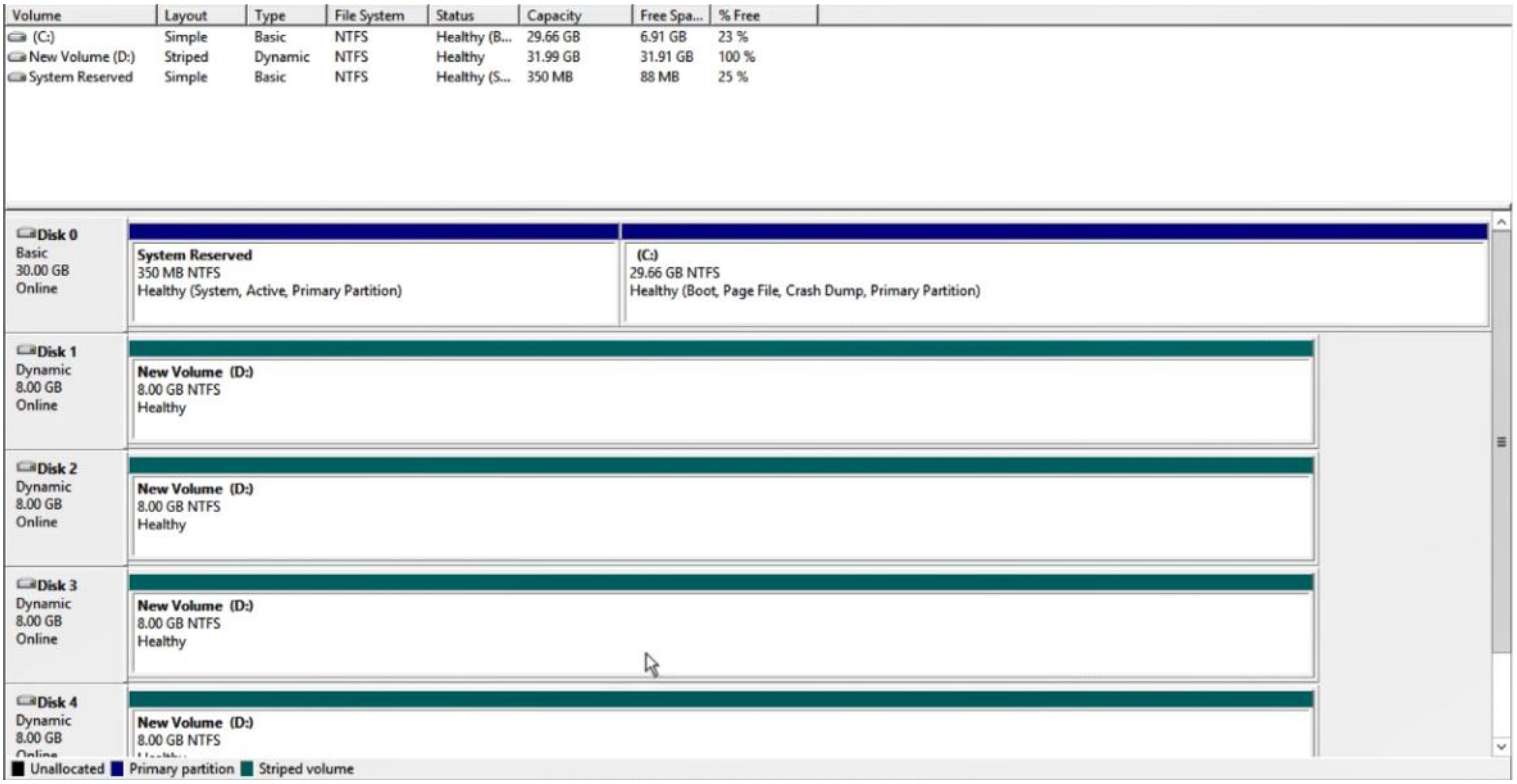
Now we will login to our Windows EC2 machine using above credentials, -
When you will login to your machine and you will go to disk management, then we will see following window-



Now since in order to increase our I/O speed, we will merge this volumes so that we can use independent arrays for better I/O performance.
Here first of all we will delete all the EBS volume, then we will create "New Striped Volume" from these volumes.

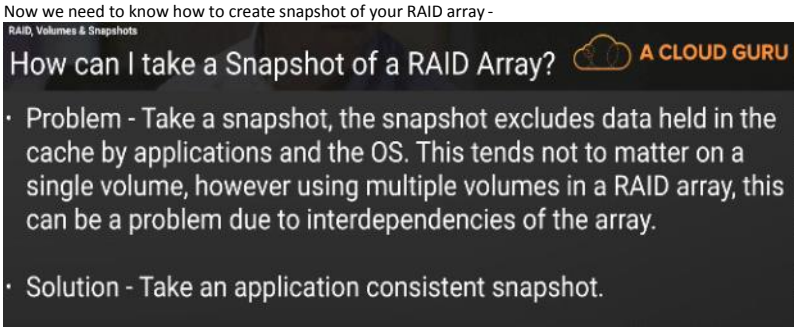


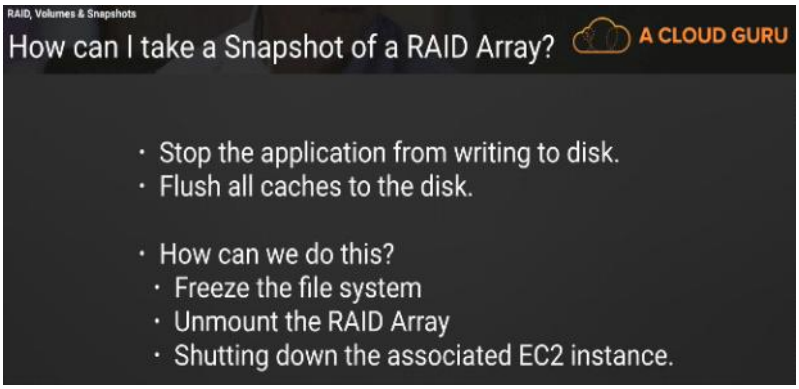
Here Striped Volume is like RAID-0 Volume.
Now add all of your deleted drives to a single drive.
Now instead of becoming one single drive, it will be a drive but distributed to 4 parts as you can see that in image-



Here we can get high I/O from this.

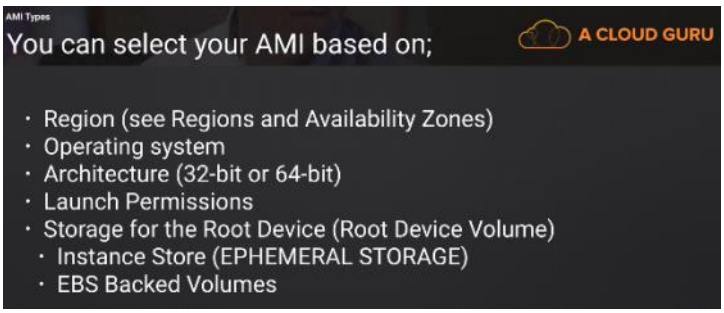
Now we need to know how to create snapshot of your RAID array -



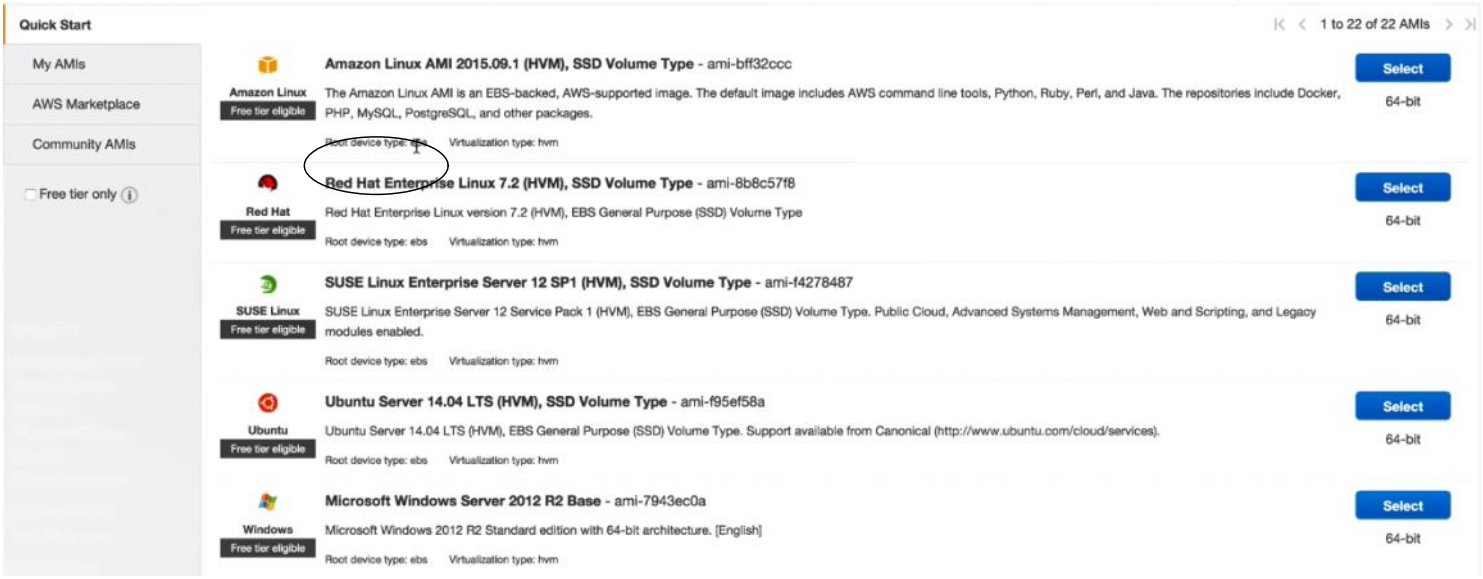


EC2 - Instance Store vs EBS

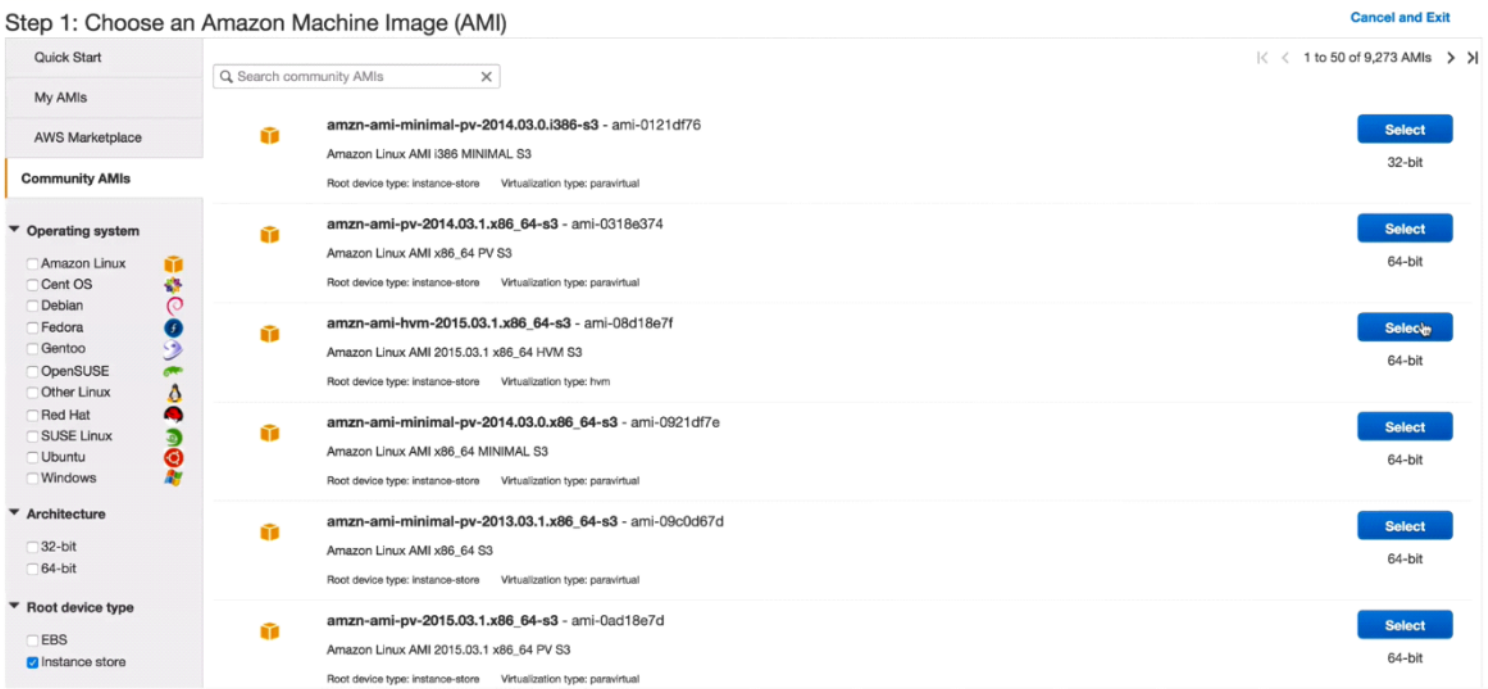
Monday, December 26, 2016 3:32 PM



Whenever you want to create new EC2 machine then at that time it asks you to choose OS, at the bottom of the selection you se e that it also shows what type of root device you are using -



But while creating your machine when you hit on the "Community AMI" on the left Panel of your console(as provided above imag e), then you will see such page -



Here you are having list of options from which you can filter your own OS type, Root device type.
Here we will use Instance Store and we will launch HVM virtualization machine and we will launch that machine.
Here while choosing type of machine then we do not get EBS option for selection our machine -

Step 2: Choose an Instance Type

⊘	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
■	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
☐	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
☐	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
☐	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High
⊘	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate
⊘	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	High
⊘	Compute optimized	c4.2xlarge	8	15	EBS only	Yes	High
⊘	Compute optimized	c4.4xlarge	16	30	EBS only	Yes	High
⊘	Compute optimized	c4.8xlarge	36	60	EBS only	Yes	10 Gigabit
☐	Compute optimized	c3.large	2	3.75	2 x 16 (SSD)	-	Moderate
☐	Compute optimized	c3.xlarge	4	7.5	2 x 40 (SSD)	Yes	Moderate
☐	Compute optimized	c3.2xlarge	8	15	2 x 80 (SSD)	Yes	High
☐	Compute optimized	c3.4xlarge	16	30	2 x 160 (SSD)	Yes	High
☐	Compute optimized	c3.8xlarge	32	60	2 x 320 (SSD)	-	10 Gigabit
☐	GPU instances	g2.2xlarge	8	15	1 x 60 (SSD)	Yes	High

Since those are restricted, now we will use m3.medium , when you will go to "Add Storage" option
then it will show something window-

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

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional instance store volumes to your instance. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

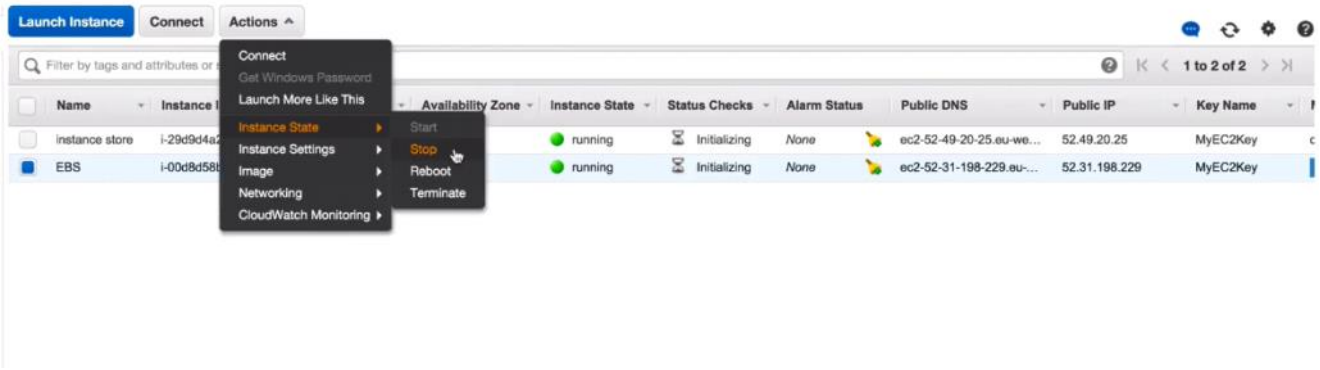
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted	
Instance Store 0	/dev/sdb	N/A	N/A	N/A	N/A	N/A	Not Encrypted	⊗

General Purpose (SSD) volumes provide the ability to burst to 3000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB. Set my root volume to General Purpose (SSD).

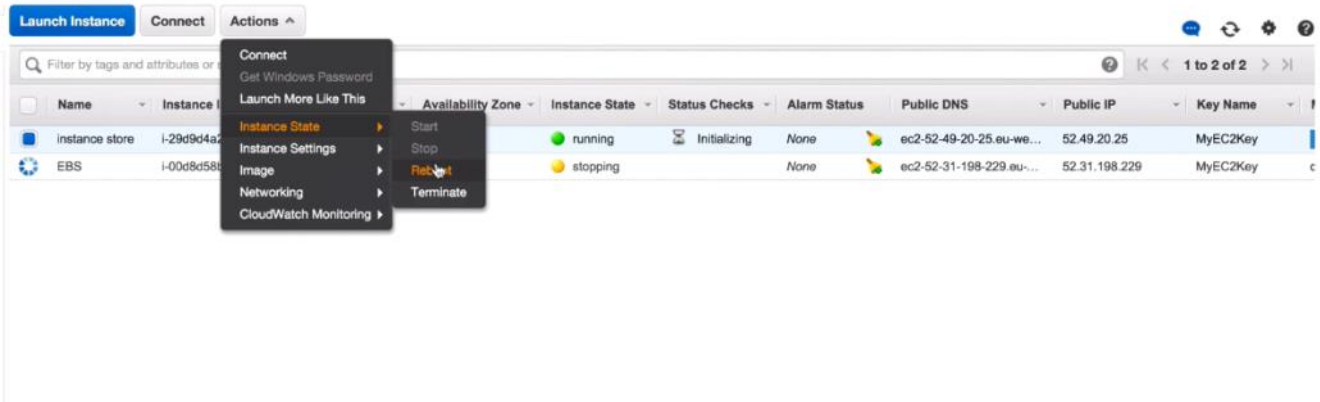
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel | Previous | Review and Launch | Next: Tag Instance

Here it says that we can add instance volume but if our machine is active then at that time we cannot add instance, storage for our machine, on that other hand we can add EBS volume when our instance has been launched.
When we choose Instance Storage, at that time we cannot do many things with this likewise what we were able to do with EBS volume.
We have created 2 instances, one with EBS volume and another one is with Instance Storage.
Here when we select the EBS volume and when we click on the actions ->Instance State, then there we see 3 options.

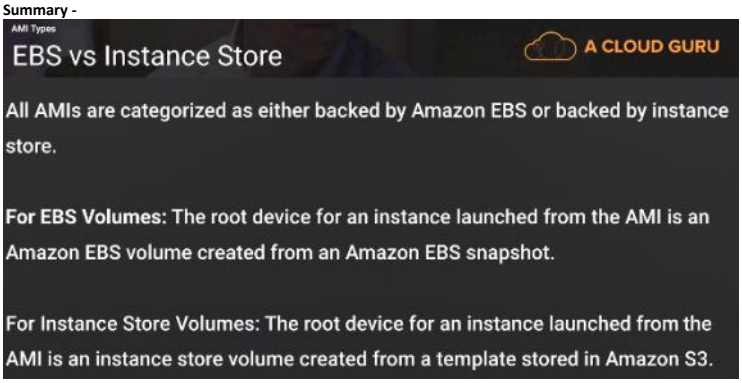


But when we select our "Instance Storage" instance we see that -

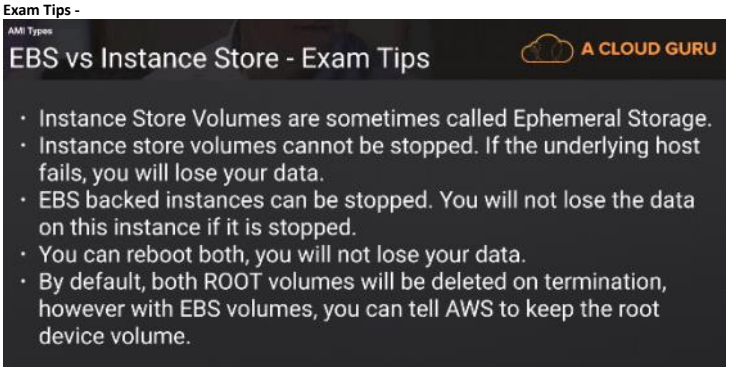


There is no option such that for Stop instance what we have seen in our EBS volume instance.
So basically we cannot stop our "Instance Storage" Instance.
Sometime we see that our memory is in high use and at that time, our system
does not work properly so what we have to do is that we have to stop our instance and again
we start our instance then it works fine, but with the "Instance Storage" instance we cannot
do to that with it, so if such problem arrives at "instance store" then you cannot do anything, maybe
your instance will get corrupted.
Now in case of EBS volume we can attach this volume from current EC2 machine, whether it is
root volume too, we can do that with EBS but cannot do with "Instance Storage" instance.

Summary -



Exam Tips -

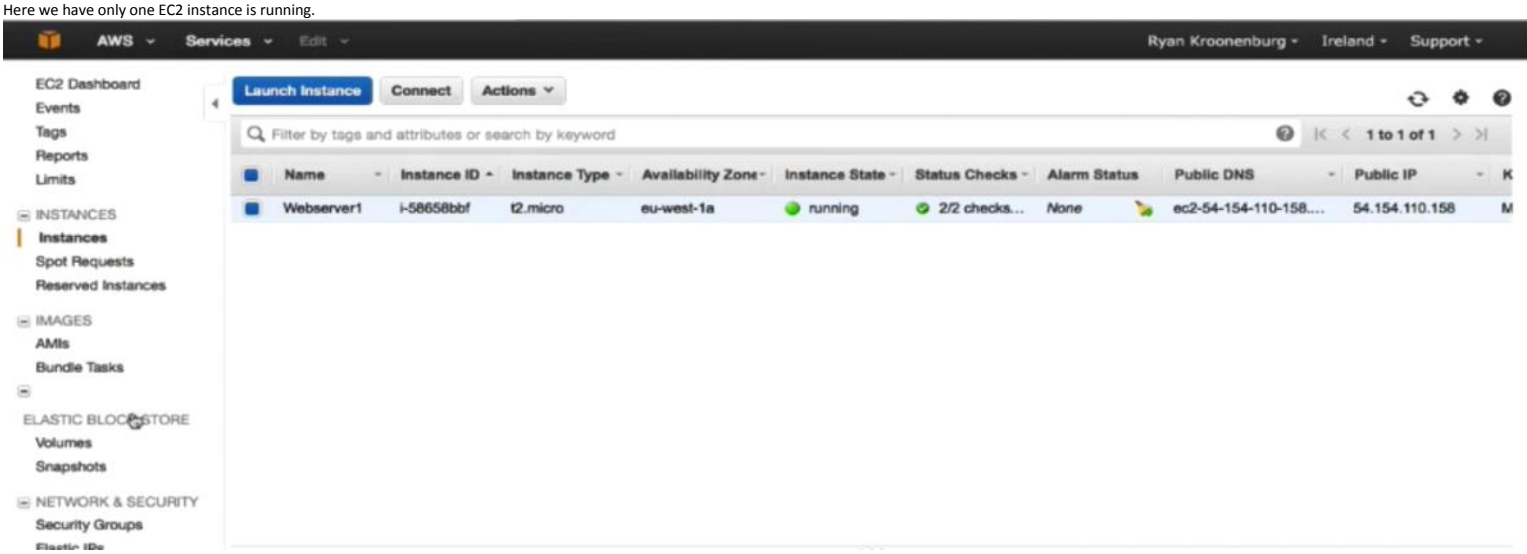


EC2 - Create an AMI

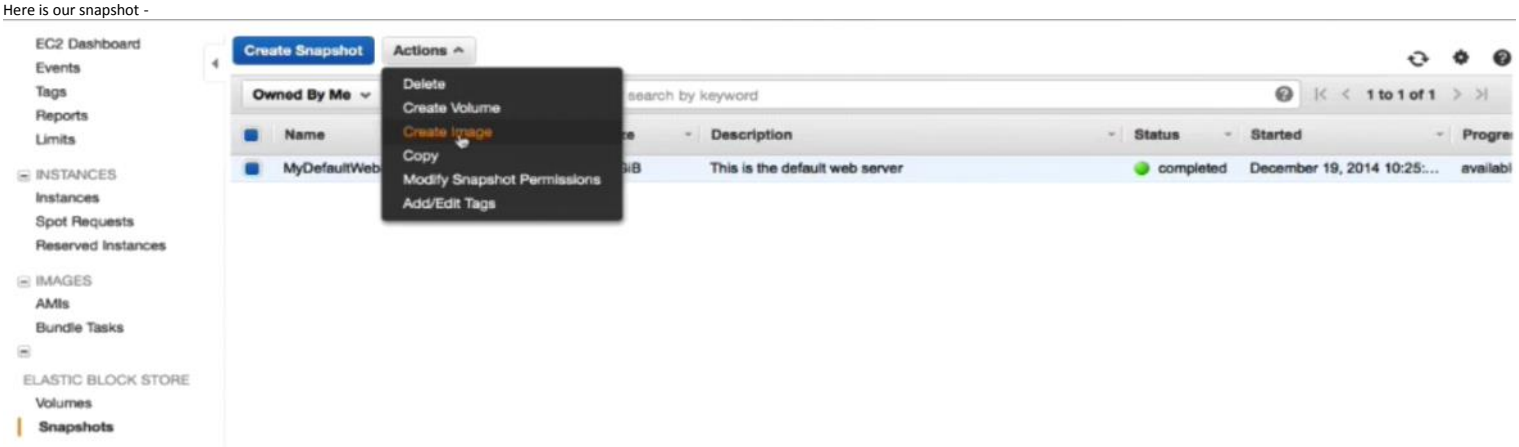
Monday, December 26, 2016 4:09 PM

Now here we will see how to create AMI and how to share this AMI with other Users.

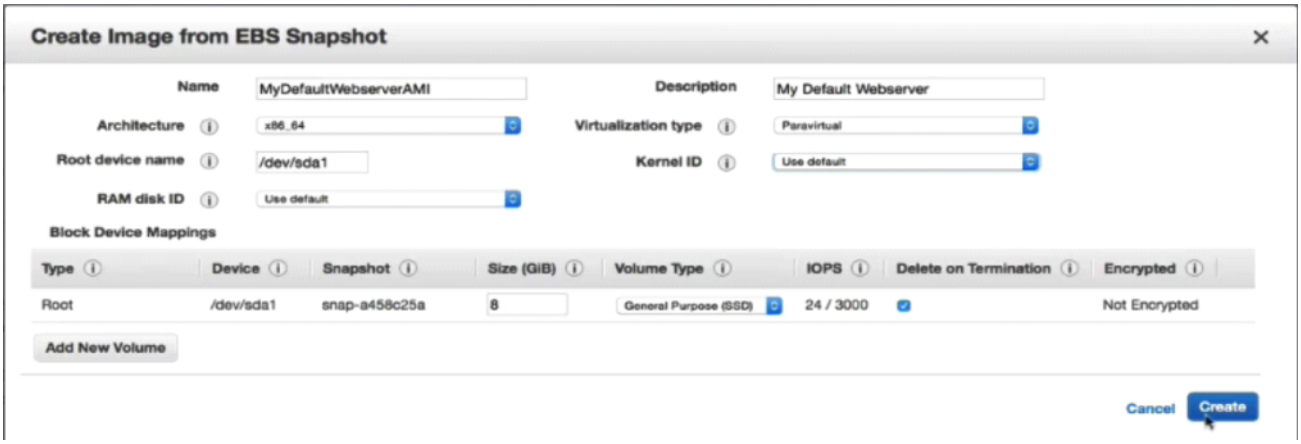
First of all we will go to EC2 console to know all running EC2 instances.
Here we have only one EC2 instance is running.



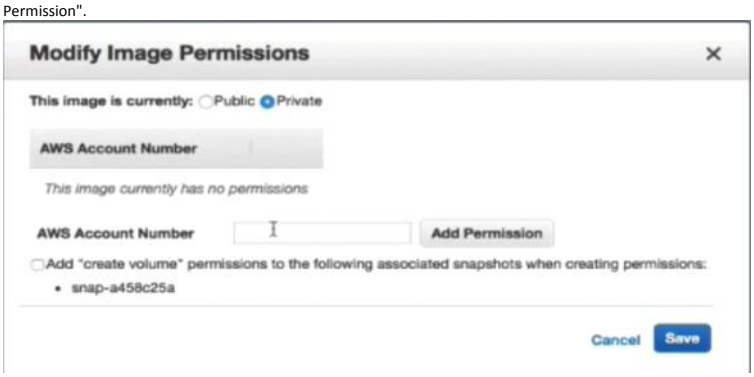
Now we will go to volumes and we will create snapshot of this volume and we will name it "MyDefaultWebServer".
Here is our snapshot -



Now from here we will select our snapshot and we will go to actions we can now that here we can create our own image by clicking on the "Create Image".
It will take us to following prompt -

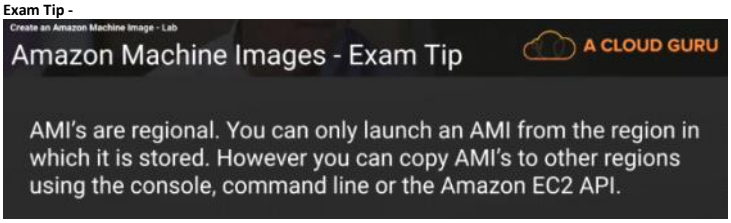


We have inserted following entries like (name and description).
When our image will get created, now we can change permission for our image.
For that we will select our image from AMIs and go to "Actions" and click on the "Modify Image Permission".



We can set our Image to public so that everyone can use this.
But even it is private we can share this image with aws members, but they need to have key to decrypt this image.
Here we will save our image as public and click to save.

Exam Tip -

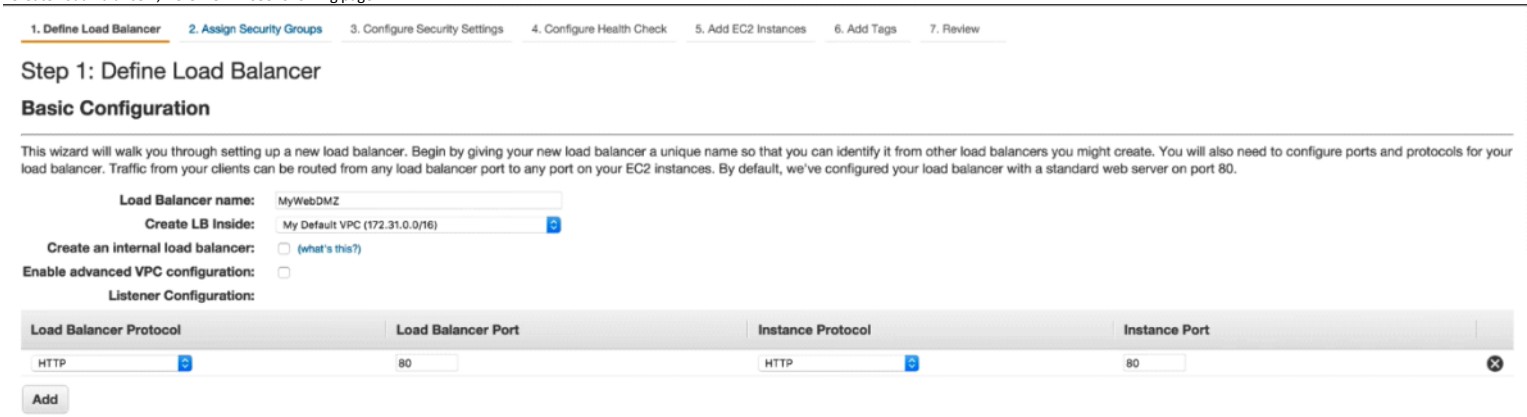


EC2 - Load Balancing

Monday, December 26, 2016 4:29 PM

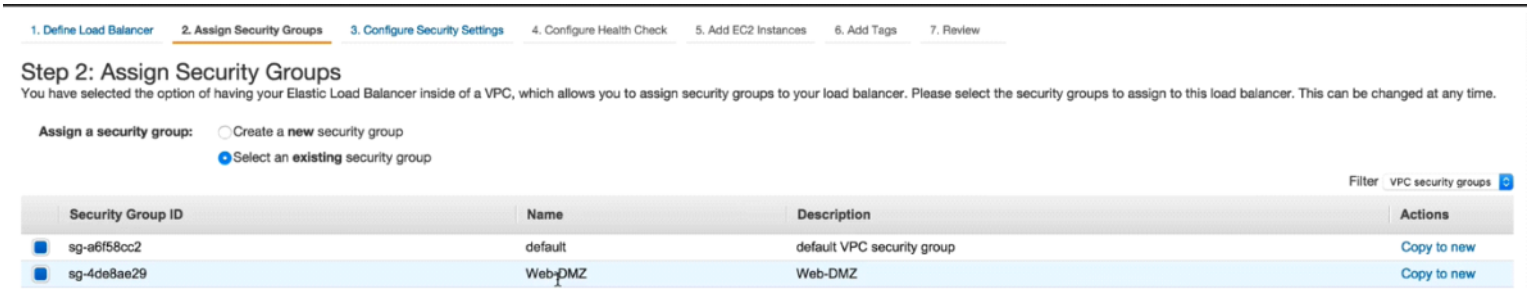
Here we will create our own load balancer -
When we go to our EC2 console, here we see at left panel's downside there is separate section for Load Balancers.
We will see following page -

Now first of all we will create our own new load balancer by clicking on the "Create Load Balancer", Here we will see following page -



Here the steps are self-explanatory.

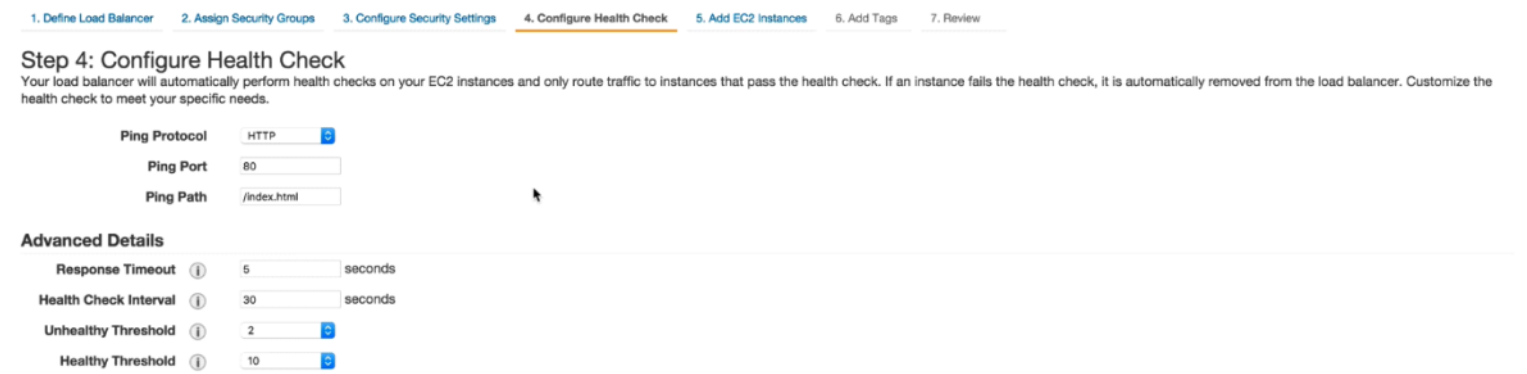
Then we will assign security groups-



Here one thing to note is that you can assign more than one security group to your load balancer.

Configure Security Settings -We will ignore this.

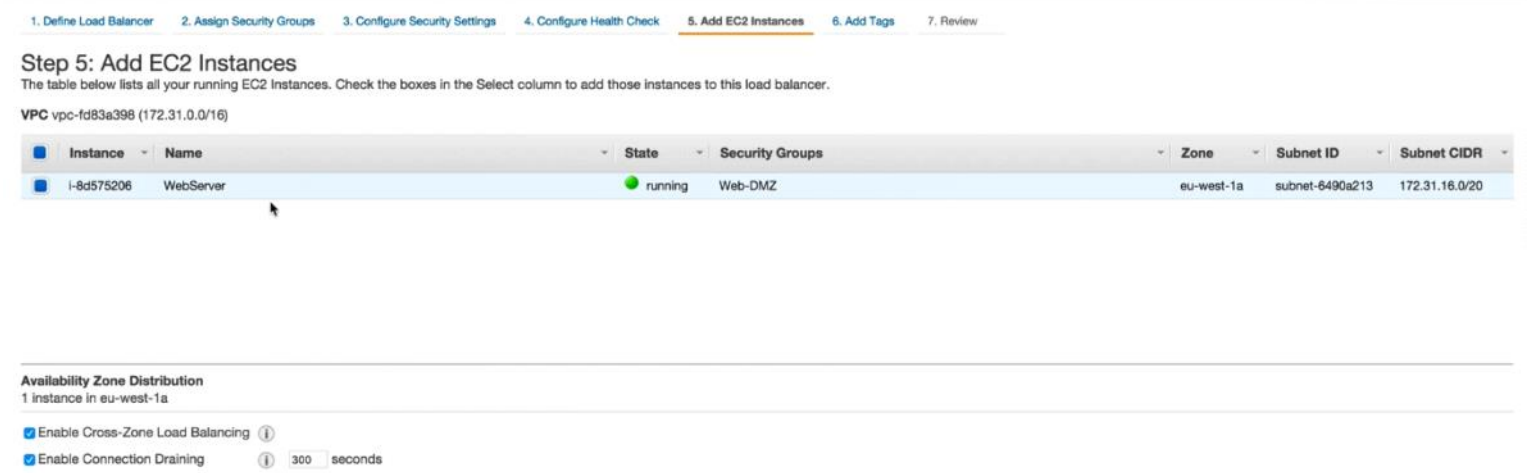
Configure Health check -Basically our load balancer will query an individual file or individual pathway and depending on the results of those queries, it is either going to bring the load balancers in the servers or out of servers.



Here we have following points -

Ping Protocols - Type of protocol that we are using
Ping Port - from which port we are pinging.
Ping Path - our file path what we use to determine for health check.
Response Timeout - time to wait when receiving a response from the health check. It can wait from 2-60 sec.
Health Check Interval - Amount of time that we use for health check.
Healthy and Unhealthy Threshold - In case of Unhealthy threshold we will check it 2 times (since we used 2 for this), so it will use HTTP protocol and 80 port to ping for our file, if it fails then our load balancer will revert out, in case of healthy check we will check it 10 times(since we entered for 10 times), and if for 10 times it will work properly then it shows that our load balancer is working fine.

Now we will add out EC2 instance,

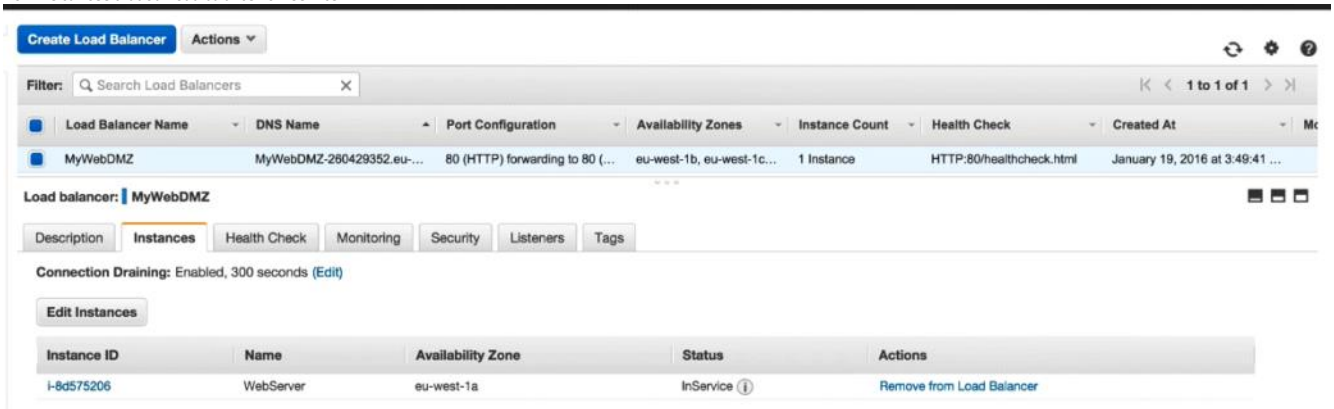


Here it will show us all of our EC2 instances.
Here we have only one EC2 machine we select that and we see at down that it gives us additional option, like "Enable Cross-Zone Load Balancing"(it distributes traffic evenly across all your back-end instance in all Availability Zones) and "Enable Connection Draining"(The number of seconds to allow existing traffic to continue flowing).

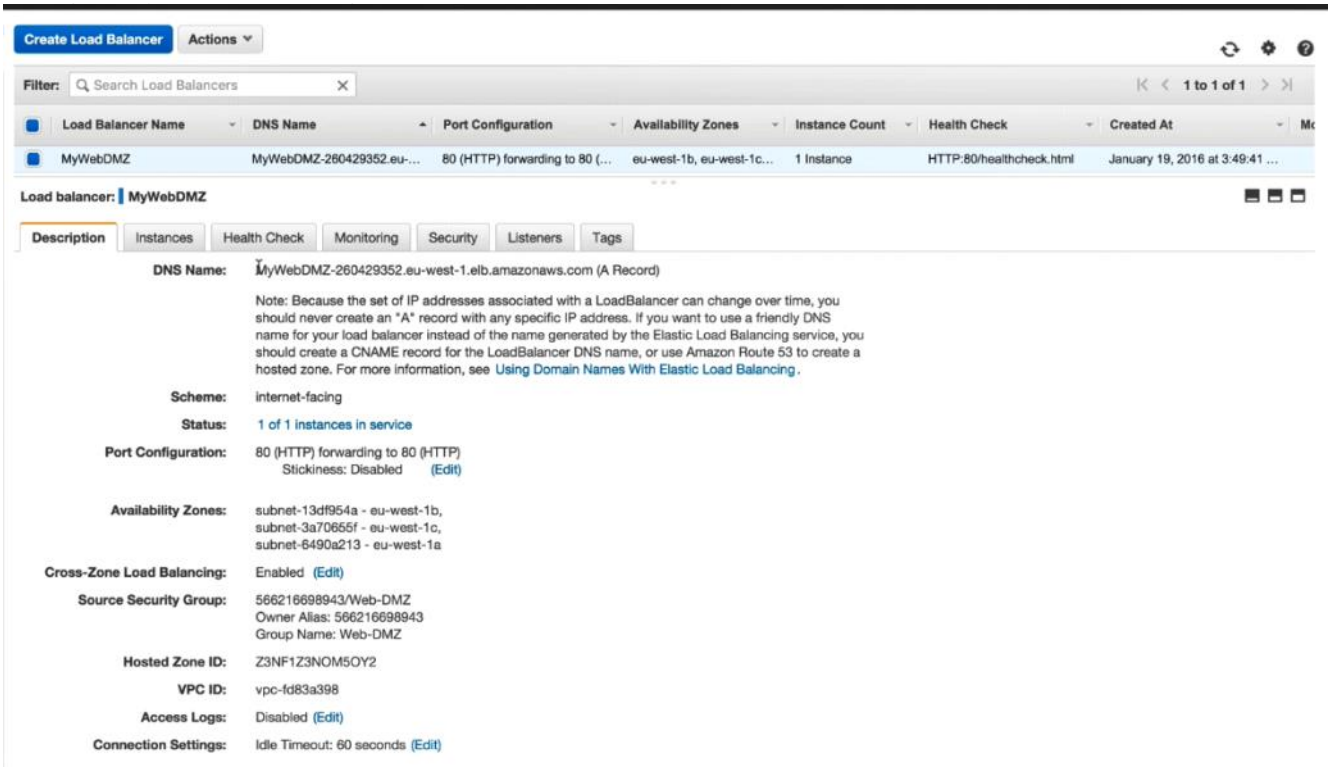
Next step is to add tags- we know what to do in it.
Review- Finally we see review button when we click on that we will see review of our Load Balancer, like we will use. Now we will hit on the create button and our load balancer is created.

Now when we select this load balancer then we will see that in properties panel at downside we will see under instance tab that our load balancer is Out-of-Service.
So we will wait for around 30 seconds to one minute and then it will come to service.

Now we can see that our load balancer is InService.

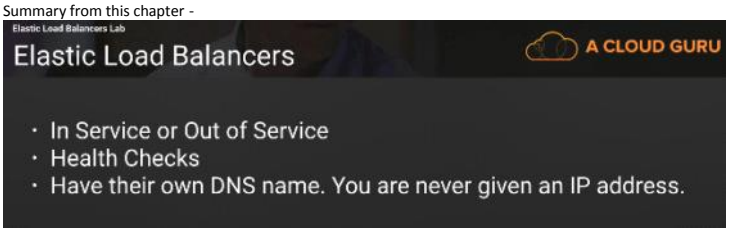


When we go to description tab then we see following information -



Here we have **DNS Name**, we will use this to resolve our load balancer.
One thing to notice is that you do not use elastic IP for your load balancer, you can give public IP for your EC2 machine, but when you deal with multiple EC2 machines then you cannot use public IP, at that time you will use DNS Name as URL.

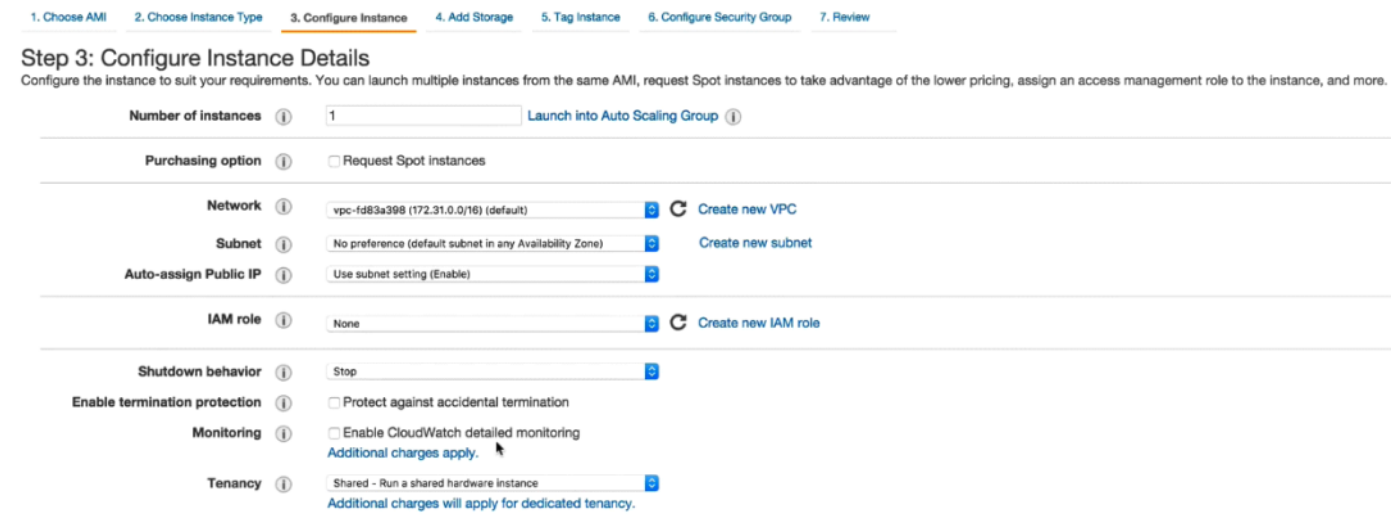
Summary from this chapter -



EC2 - CloudWatch

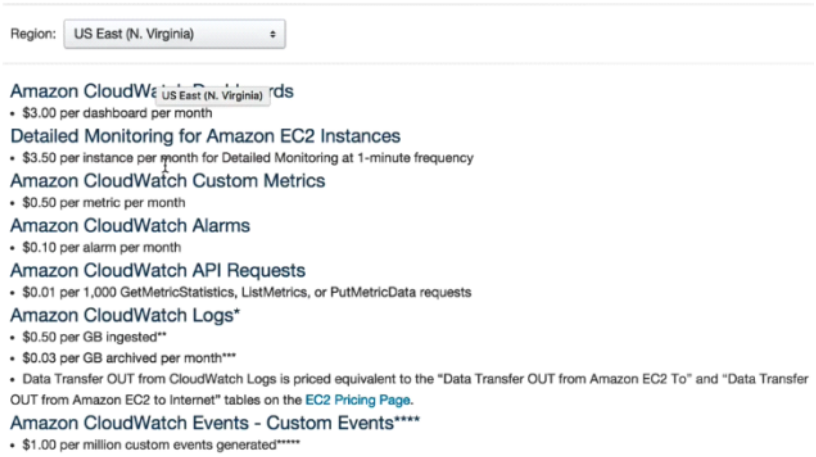
Tuesday, December 27, 2016 10:57 AM

When we create our EC2 machine, so for the "configure Instance details" page we have option for Monitoring. By default all the EC2 machines have basics default monitoring, but for advance level monitoring we use Cloudwatch service, although since it is an additional service you will be charged for using this service.



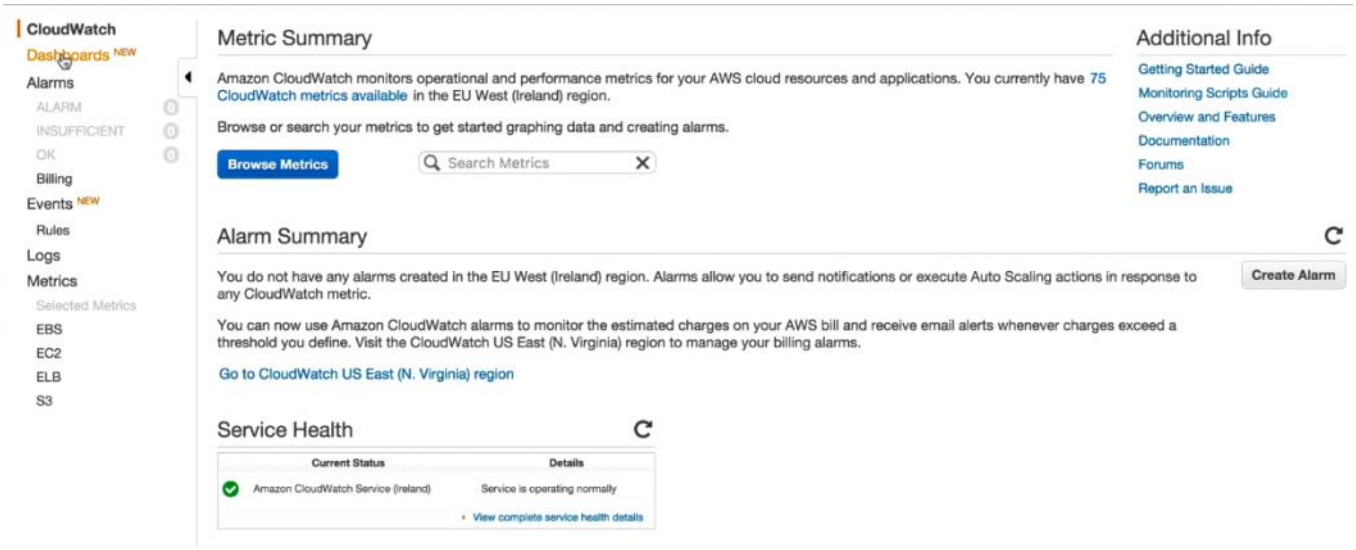
Since it is not fully free service, but still we have some portion of this service that we get for free tier package of EC2 machine.

Now if you want to activate detailed monitoring then price varies from region to region, here we are seeing price details for N. Virginia region.



Let's see how does it work ?

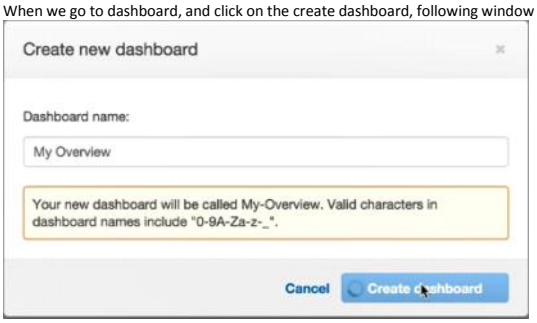
From the main AWS console under management tools you will have CloudWatch (or you can search from the search bar).
When you will click on the CloudWatch link, it will take you to CloudWatch web console.



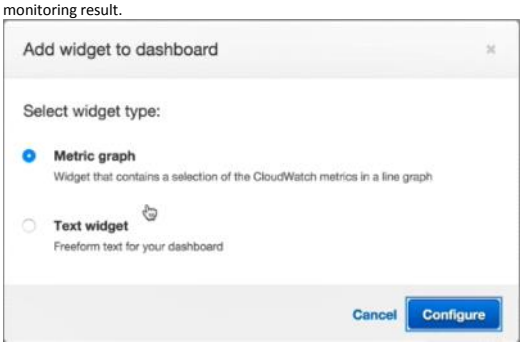
Here on the left panel we have following-

Dashboard - Main Dashboard
Alarms - list of alarms that we have set for monitoring our machine.
Events - It helps you to respond to state changes in your AWS resources. When your resources change state they automatically send events into an event stream.
Logs - Creates logs for your changes what so ever you are keeping eyes on using CloudWatch.
Metrics- Metrics are data about the performance of your systems. By default, several services provide free metrics for resources (such as Amazon EC2 instances, Amazon EBS volumes, and Amazon RDS DB instances). You can also enable detailed monitoring some resources, such as your Amazon EC2 instances, or publish your own application metrics. Amazon CloudWatch can load all the metrics in your account (both AWS resource metrics and application metrics that you provide) for search, graphing, and alarms.

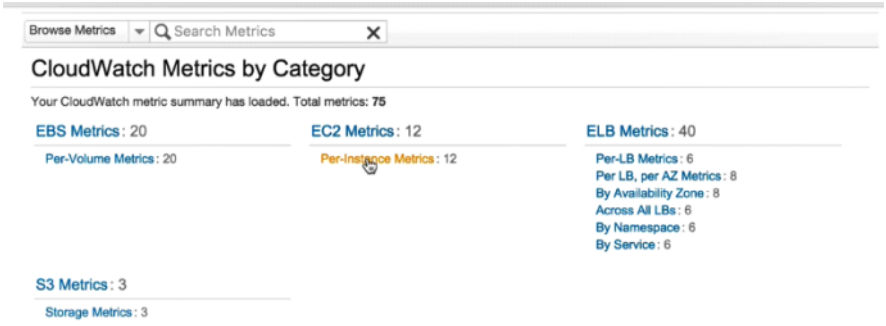
Now we are going to create our own Dashboard -
When we go to dashboard, and click on the create dashboard, following window will appear -



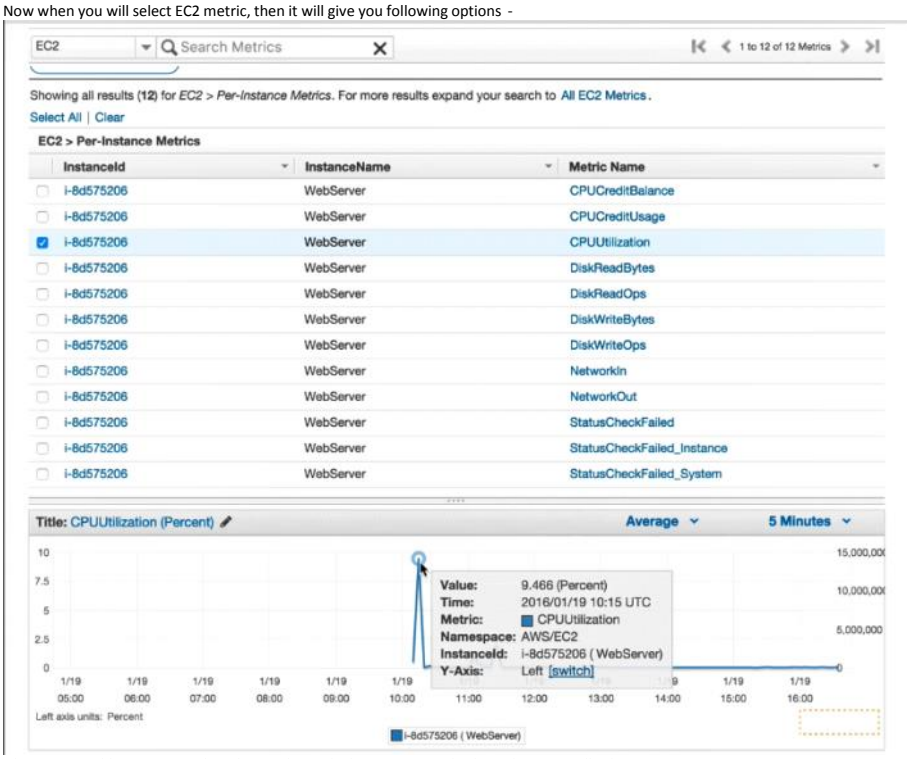
After that it will take you to next page, here you can choose whether you want metric or text view for monitoring result.



Then it will ask you, like where you want to add/bind this metric -



Now when you will select EC2 metric, then it will give you following options -



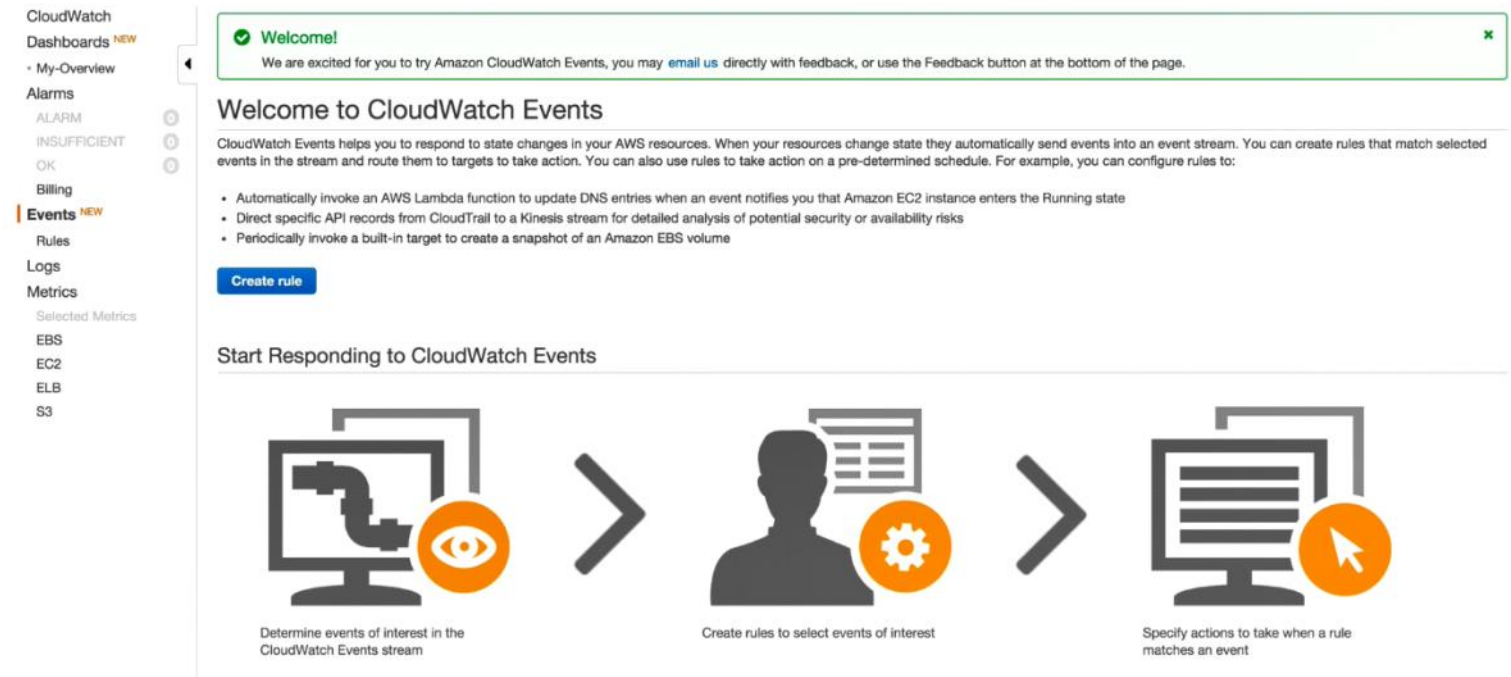
Like you can add metric to each and everything which is present in this list, when you will select any component then it will show metric graph of that component on your EC2 machine.

Sometime in exam we can get questions like what kind of metrics are available in our EC2 metrics list - answer is - CPU, Disk, Network, Status.

Since from the above selection we choose only CPU-Utilization now from the dashboard page from the CloudWatch console, following screen will appear -

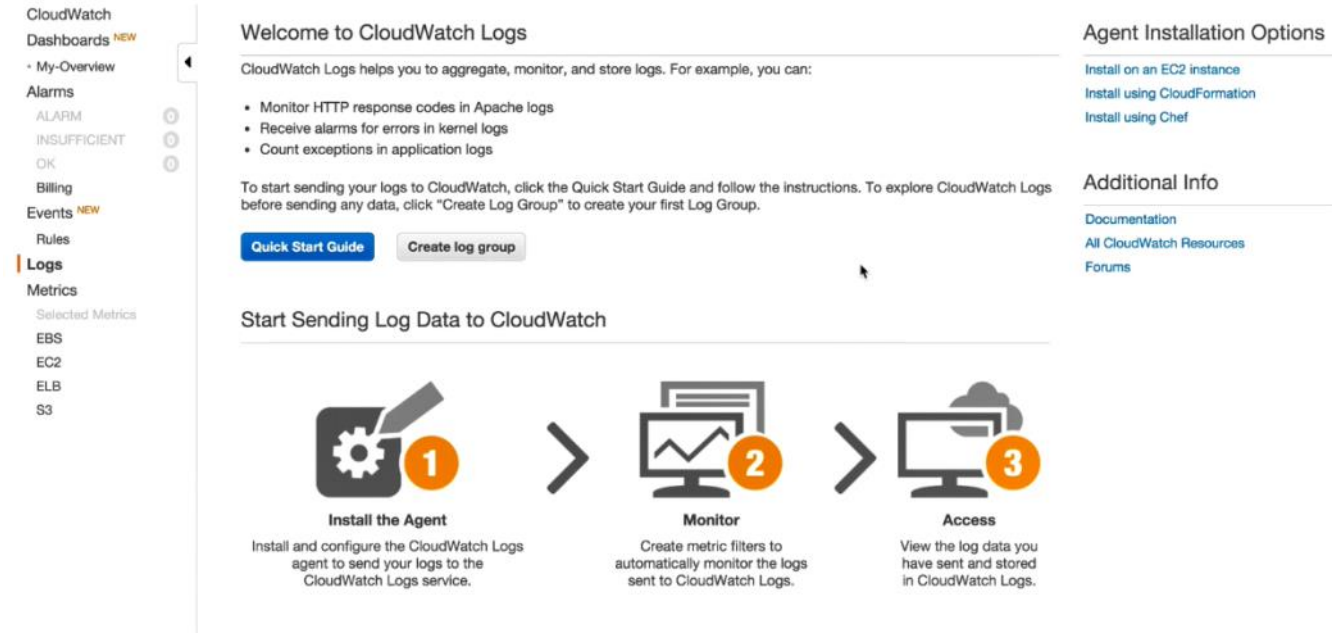
Now if we want to add more widget then we can add that too by clicking above button says "Add widget".

Now we will see what to do with Events -

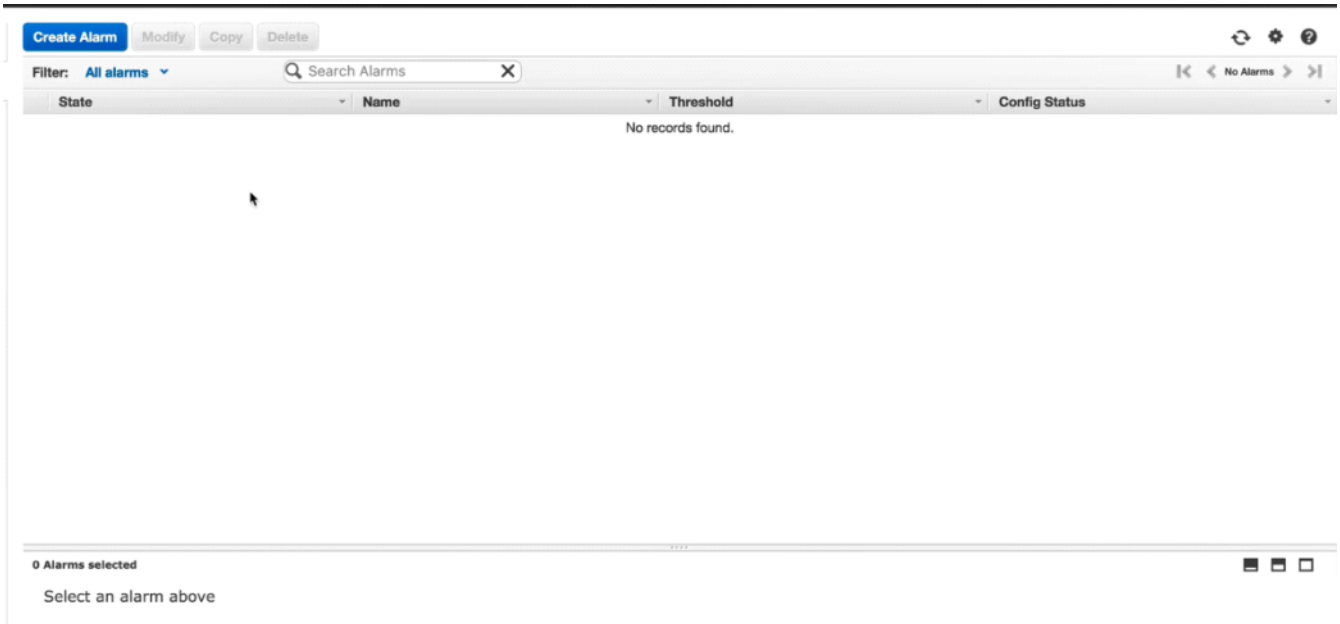


When we go to Events console we see following screen (read through this.)

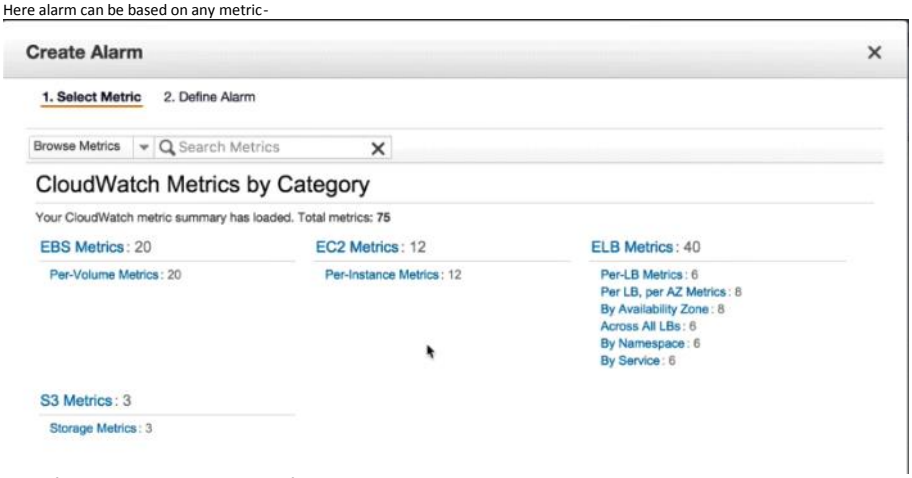
When we go to Logs -



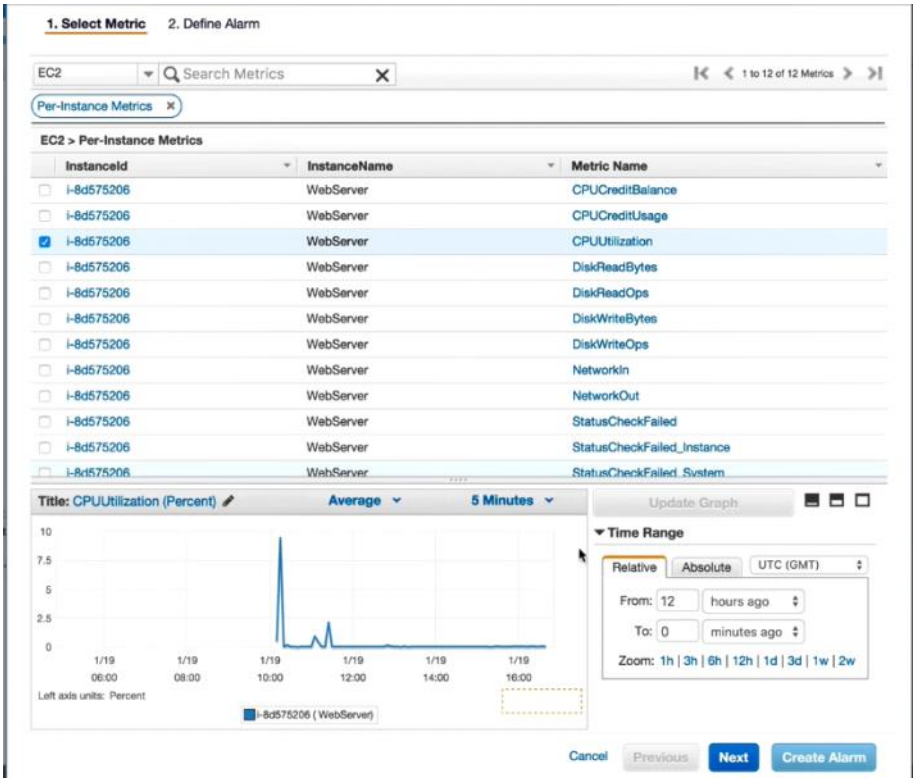
Now when we go to Alarms -



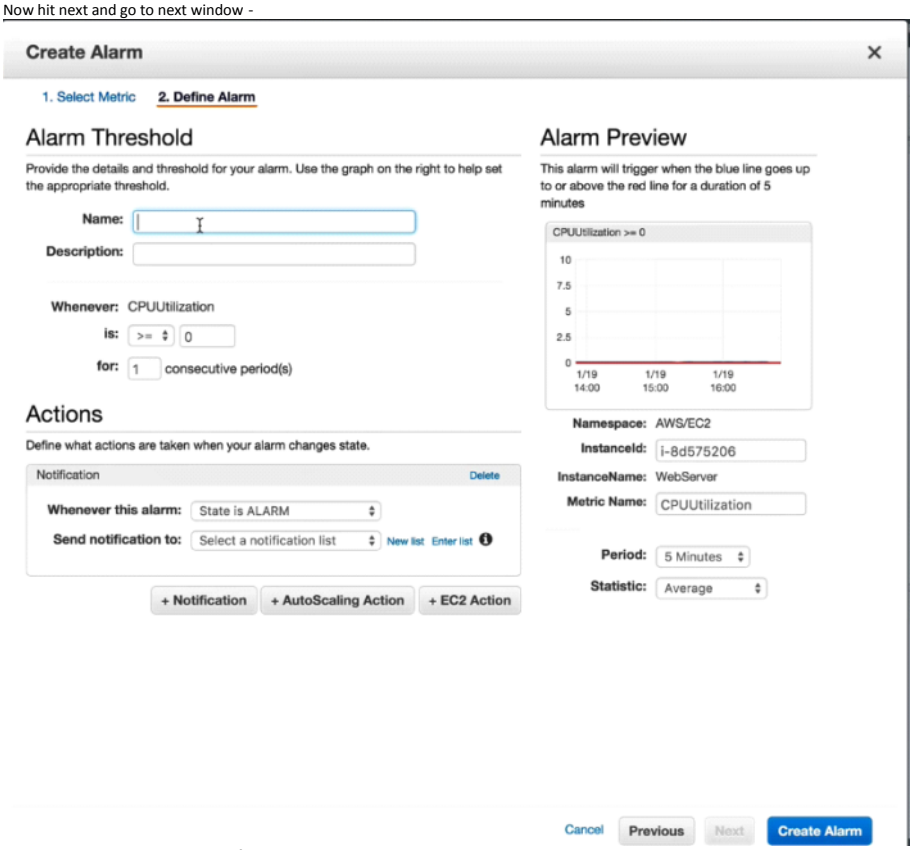
Here we will create new alarm, by clicking on the "Create Alarm" b utton.
Here alarm can be based on any metric-



Let see from here we want to create alarm for EC2 metrics.
Under that we will create alarm for CPU utilization.



Now hit next and go to next window -



From this alarm threshold we have following option -

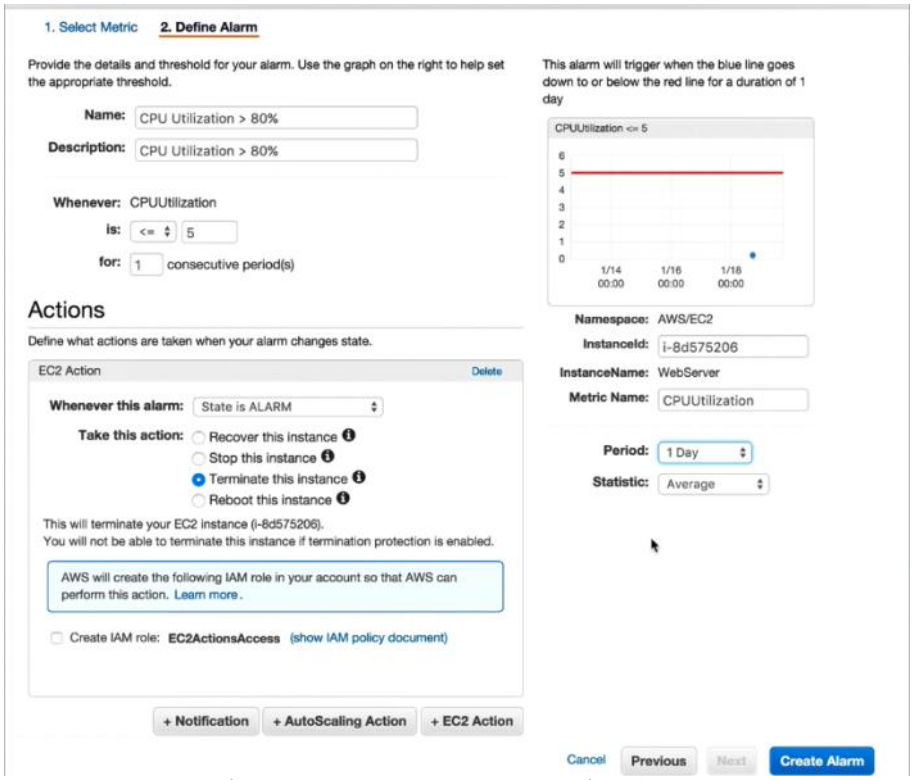
Name - name of our alarm

Description - why we have created this alarm, details about this alarm.

Whenever - here we will determine when to trigger this alarm, like when this alarm will hit and for what time period this should check for this.(for period interval, at right down panel we have period, here we can choose interval for our period).

Action - here we have now what to do when this appears, we can choose "send notification to" and then we will enter email address, so when it will cross the threshold what we have set then it will send an email to email-ID that we have used.

Here we have more options, like "+EC2 Action" by clicking on this we can add more action to this list, then following screen we will see -



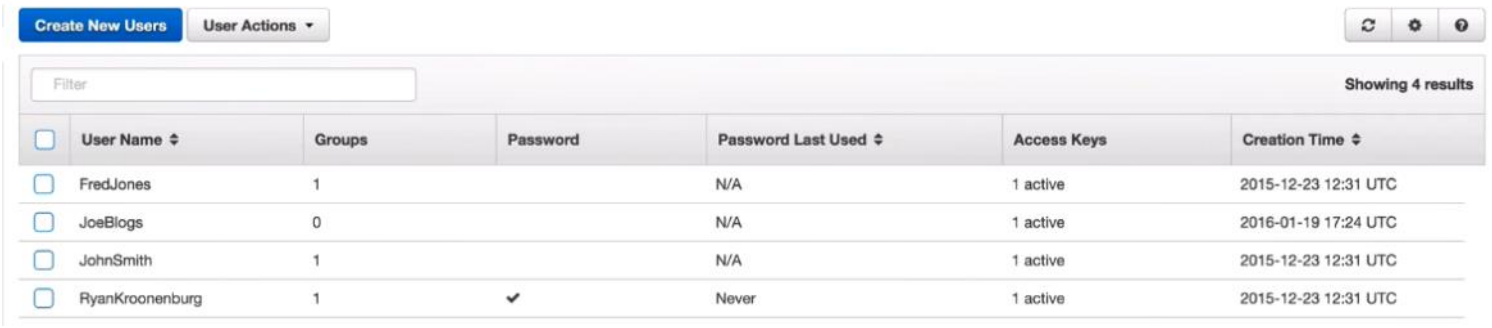
Here we have set CPU-Utilization for less than or equal to 5%, and we are monitoring it for single period for one day, it means that if our CPU-Utilization is less than 5% for whole day then it will terminate our EC2 instance, now how to handle this with Auto-Scalling group we will cover that later.

Summary -

EC2 - CLI

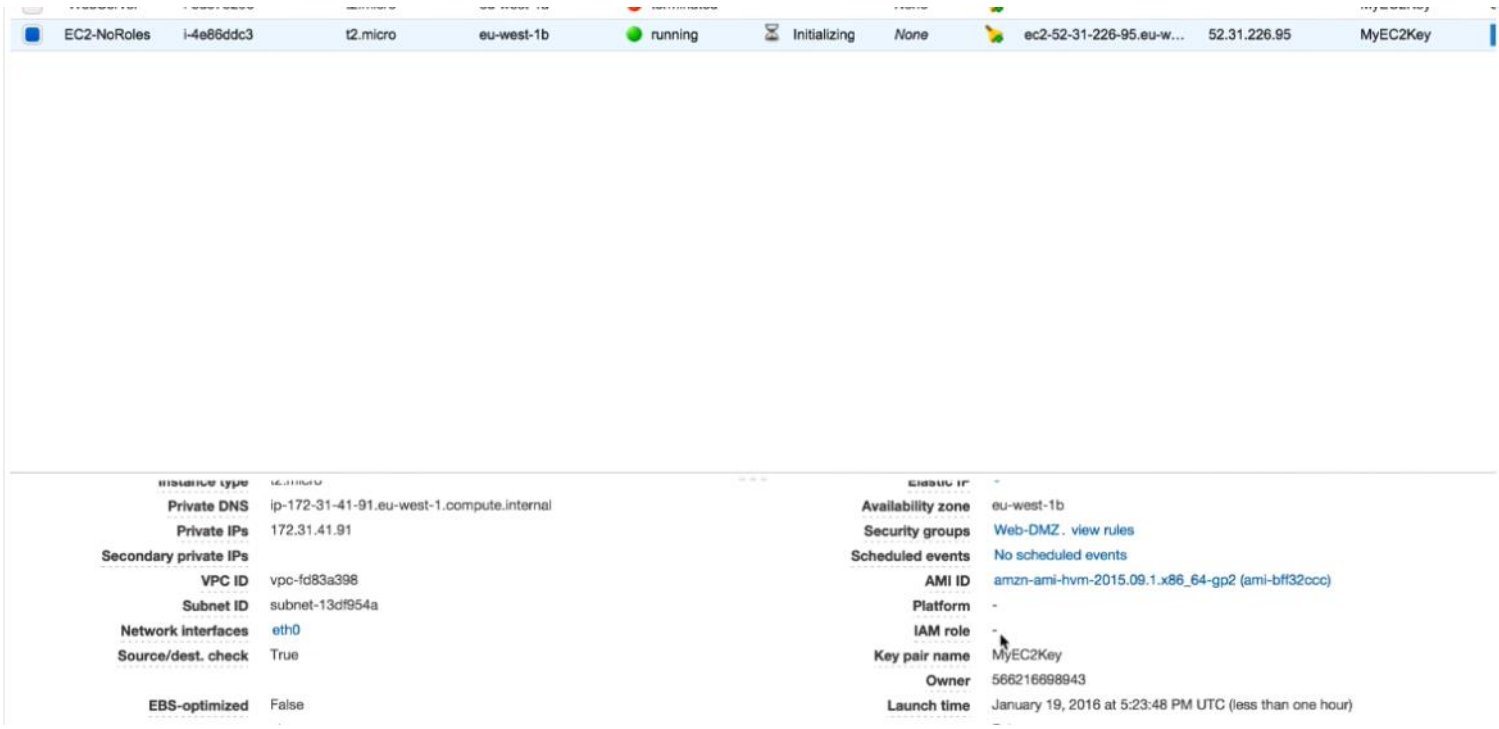
Tuesday, December 27, 2016 12:24 PM

Here we will learn how to use AWS using CLI.
First of all we need to have EC2 machine, what we can easily create with what we have learned so far (while creating EC2 machine, note one thing that we should not put IAM role on that machine, while selecting for IAM role keep it for none, we will manually put IAM role for that).
Since our machine has been created we can now go to IAM role to provide access to our EC2 machine.
For that we will go to IAM console and here we will go to user and then create new user.
We have created our new user "JoeBlogs".



Now we will add this user to group, we will go to groups and create new group having name "MyS3Group" and we will give full S3 access policy to this group.

Now click on your group and add users to your group, just check the name of the user and click on the add users, it will add that user to your group.
Here we will add "JoeBlogs" to our group.
Now we have created group for IAM role.
Now we will go to our EC2 console and here we will see that for our newly machine, there is no IAM rule(since we set it to None).



Now here is a catch, which is very important to know for exam purpose that is you can add IAM user to EC2 machine only when you are creating it.
Once your machine is up and running you cannot add IAM user to that machine.

Now we will login to our new EC2 machine and here we will use CLI to use and access our S3 bucket.
Default command for accessing S3 bucket is

```
aws s3 ls
```

When you will add this to your console following message will appear -

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```

Since we haven't loaded our credentials, first of all we have to add credentials to our machine.
For that we will run following command -

```
[root@ip-172-31-41-91 ec2-user]# aws configure
AWS Access Key ID [None]: AKIAJVS3PIV5ZGEA6NKA
AWS Secret Access Key [None]: 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
Default region name [None]: eu-west-1
Default output format [None]:
```

Here we have used "secret access key" and "access key ID", we have downloaded this key while creating our user and for the region name you can write your region which is close to your geo-location.
We have all the same command here as well what we use for a Linux machine, but we have to include "aws s3" before entering any command so that it will know that we are asking for s3 bucket.
So when you will type -
"aws s3 ls" - it will list all the buckets.
"aws s3 ls --recursive" - it will list all the content on the buckets recursively
For more about S3-CLI we can read it from here.
If you want help then use - **"aws s3 help".**

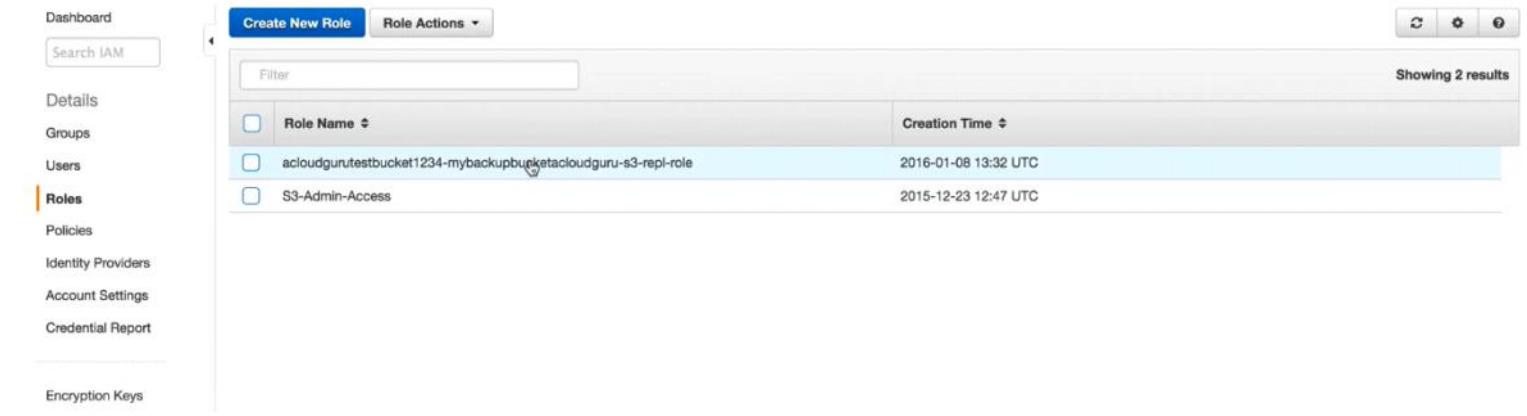
Now if we want to see where our aws credentials are then we can look it at hidden (.aws) directory.
Since while accessing to your S3 bucket you only need to have **access_key_id** and **secret_access_key** then it is required that you should not keep your key to your Ec2 machine, coz if you will put in future your machine public, and if someone will have access to your machine, then it will not be safe to have key at there.
Now here roles comes in the picture, if you have set roles for S3 while creating your EC2 machine, then at that time you would not need to have "access key id" and "secret access key" for accessing s3 bucket from EC2 machine using CLI.

EC2 - Using Roles

Tuesday, December 27, 2016 1:24 PM

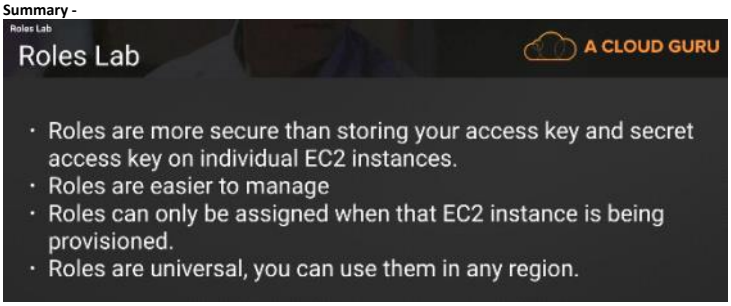
Here we will learn about IAM roles.
Since we know that if we are using EC2 machine and if we have saved our credentials and somehow if someone is able to access our machine then maybe he will be able to access those credentials too, and then he will be able to control users, and security will be compromised.
One more thing to understand is that if we are creating maybe number of Machines and if we are accessing those machine with same credentials that we created while creating our users, and somehow if in future we want to change those credentials then we have to alter those changes in all of our machines.

Now we will login to IAM console -
Here on the left panel we have various options, and one from them is "Roles".



Here currently we have 2 roles.
Once you have your roles, then go back to your EC2 console.
Now when you will create your machine, there you will be able to apply role to your machine (on 3rd step).
One thing to notice here is that you cannot apply role to your ec2 machine once it is up and running, you can only add those roles when you create your machine.
Although you can edit policy document of applied role what you have assigned to your EC2 machine, here one thing to notice is that all the changes on that role will effect immediately and it will be only up to for that EC2 machine, but it will not be changing the role itself.

Summary -

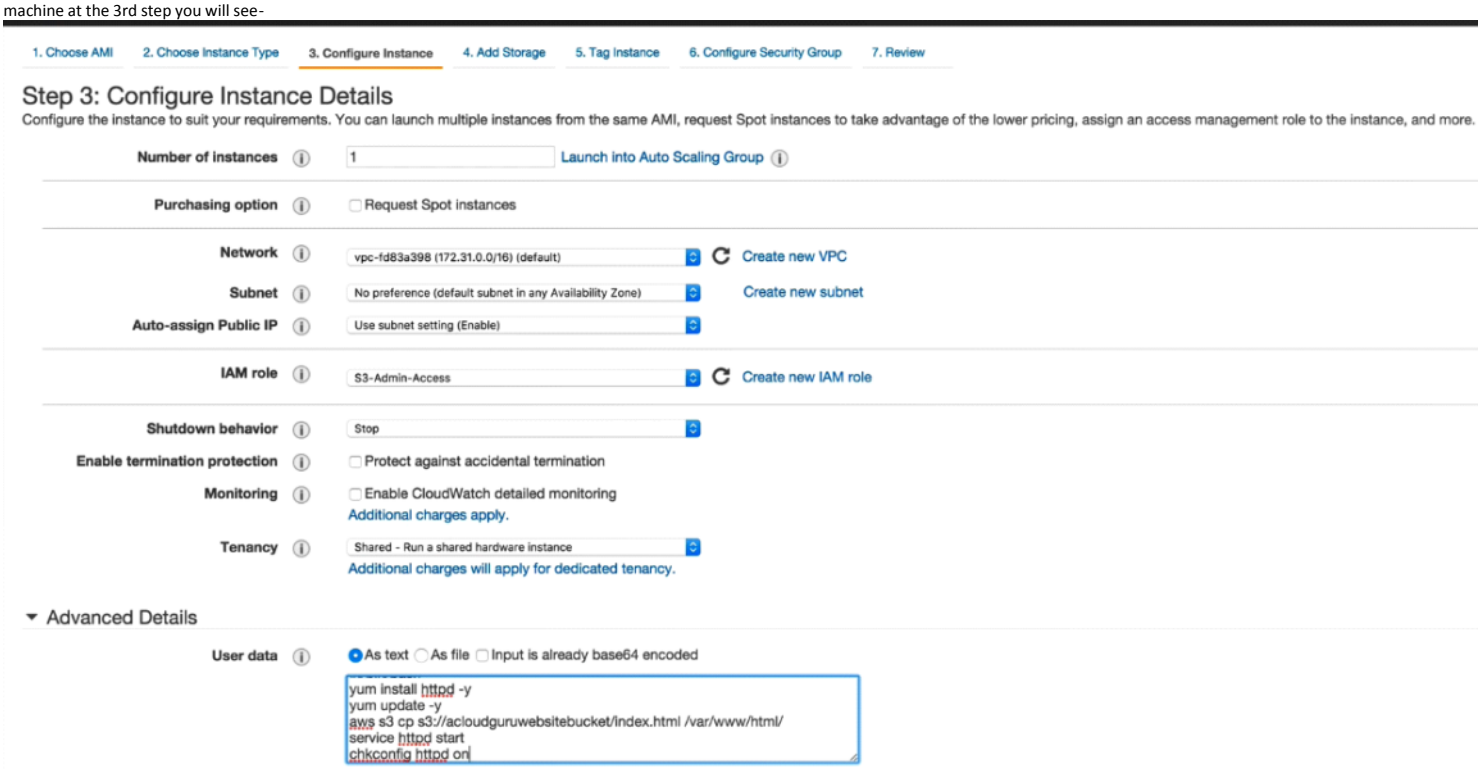


EC2 - Bash Scripts Lab

Tuesday, December 27, 2016 2:42 PM

Here we will user bash script for our EC2 machine.
First of all we will create a bucket and we will put any .html file that we want to show for our webpage.Then we will add bash script, while creating our EC2 machine(on the 3rd step at advance details).

1. Create new bucket and add simple html file that says "hello world".
2. Now create new machine(maybe now you know how to create EC2 machine), while creating machine at the 3rd step you will see-



Here we have defined our IAM role - "S3-admin-access" (what we have created prior).
And for the "Advanced details" we will add some bash script here and then it will get executed once our machine will be live and running, so here we have used following script

```
#!/usr/bash
yum install httpd -y
yum update -y
aws s3 cp s3://bucktename/index.html /var/www/html/
service httpd start
chkconfig httpd on
```

Here what we are doing-

- for first line we are adding this to run via root
- Then it will install apache server
- Then it will update the system
- Then It will copy the html file that we have copied to our s3 bucket
- Then it will start the server
- Then it will check whether server is running or not.

Apart from that there is nothing to add, we will use all the things default.
After doing all the steps now our machine will be ready to run.

Here one thing to notice is that if you are accessing machine from region one and if your bucket for what you have assigned IAM role, is not in the same region then it may give you error, so for best practice it is good to have bucket in same region, ,on what region your machine is running.

Now once our EC2 machine is up and running, we will just copy the public IP of our machine and we will put that in our URL bar of browse, so from here we will be able to see that we are able to view content of that html file that we have created and moved from bucket to our EC2 machine, using advance bash script option while creating.

EC2 - Instance Metadata

Tuesday, December 27, 2016 3:01 PM

Here we will learn data information about our EC2 machine, like our public IP or our Private IP or how we can access this data from command line.
First of all we will create one machine and we will run this machine.
Once you are in your machine, you can view details about your machine's metadata by using this command.

```
[root@ip-172-31-36-83 ec2-user]# curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[root@ip-172-31-36-83 ec2-user]#
```

Here we have used curl command, "Curl" is basically command to access any IP address, it is often used to download files too.
Now you can browse from the list and you can get information about your instance.
If you want to know your public IP -

```
services/[root@ip-172-31-36-83 ec2-user]# curl http://169.254.169.254/latest/meta-data/public-ipv4
52.48.51.207[root@ip-172-31-36-83 ec2-user]#
```

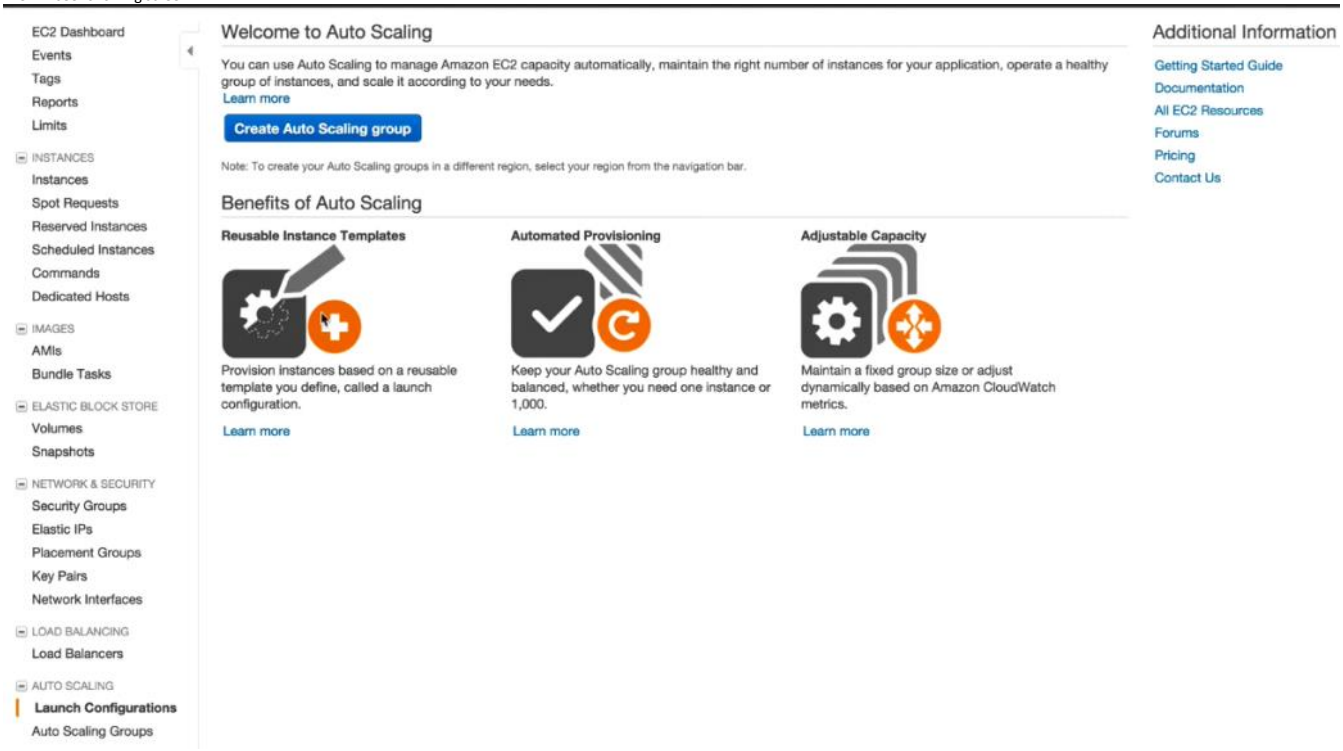
From above other information you can get some more details about this.

EC2 - launch configuration and Auto Scaling Groups

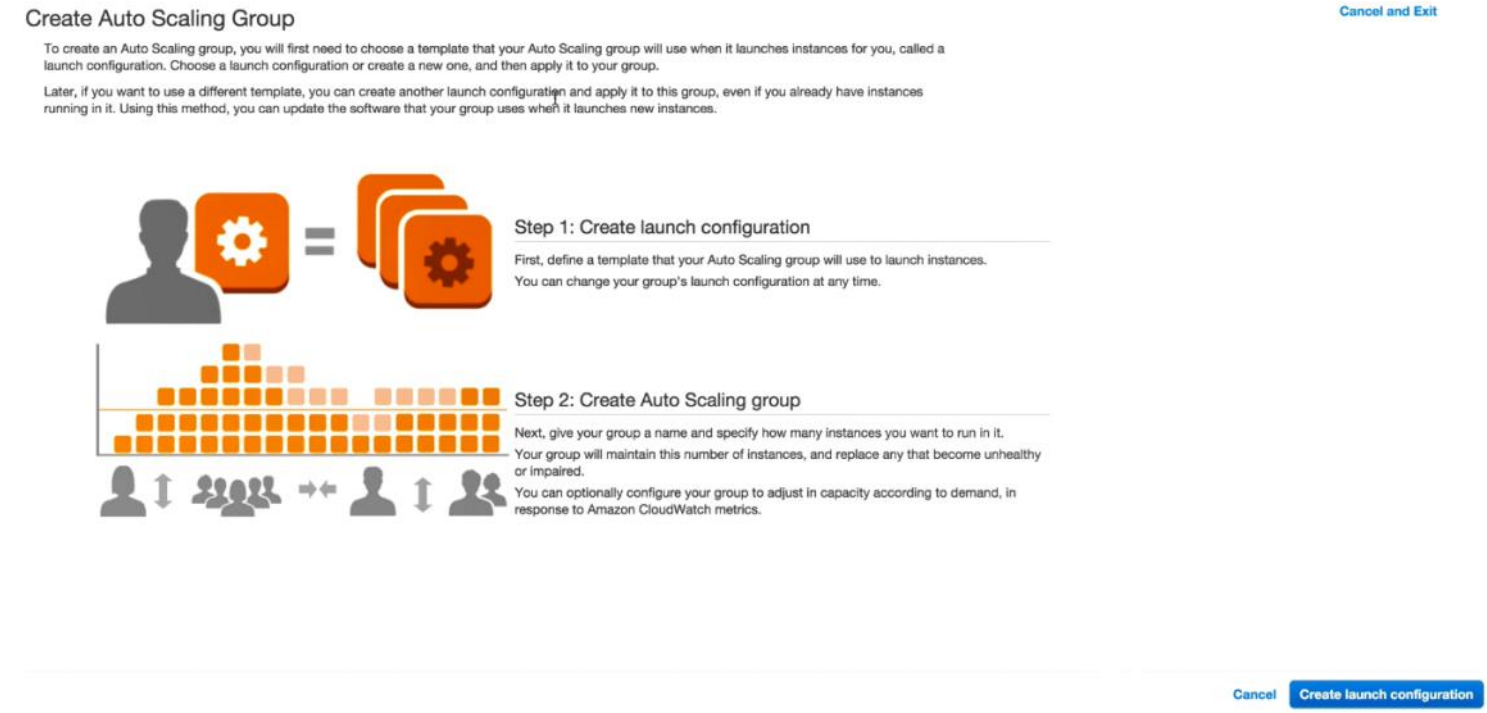
Tuesday, December 27, 2016 3:11 PM

Here we will learn about auto-scaling policy.
First of all we will upload a new html file to our bucket and then we will need to have load balancer active(we know how to create load balancer)

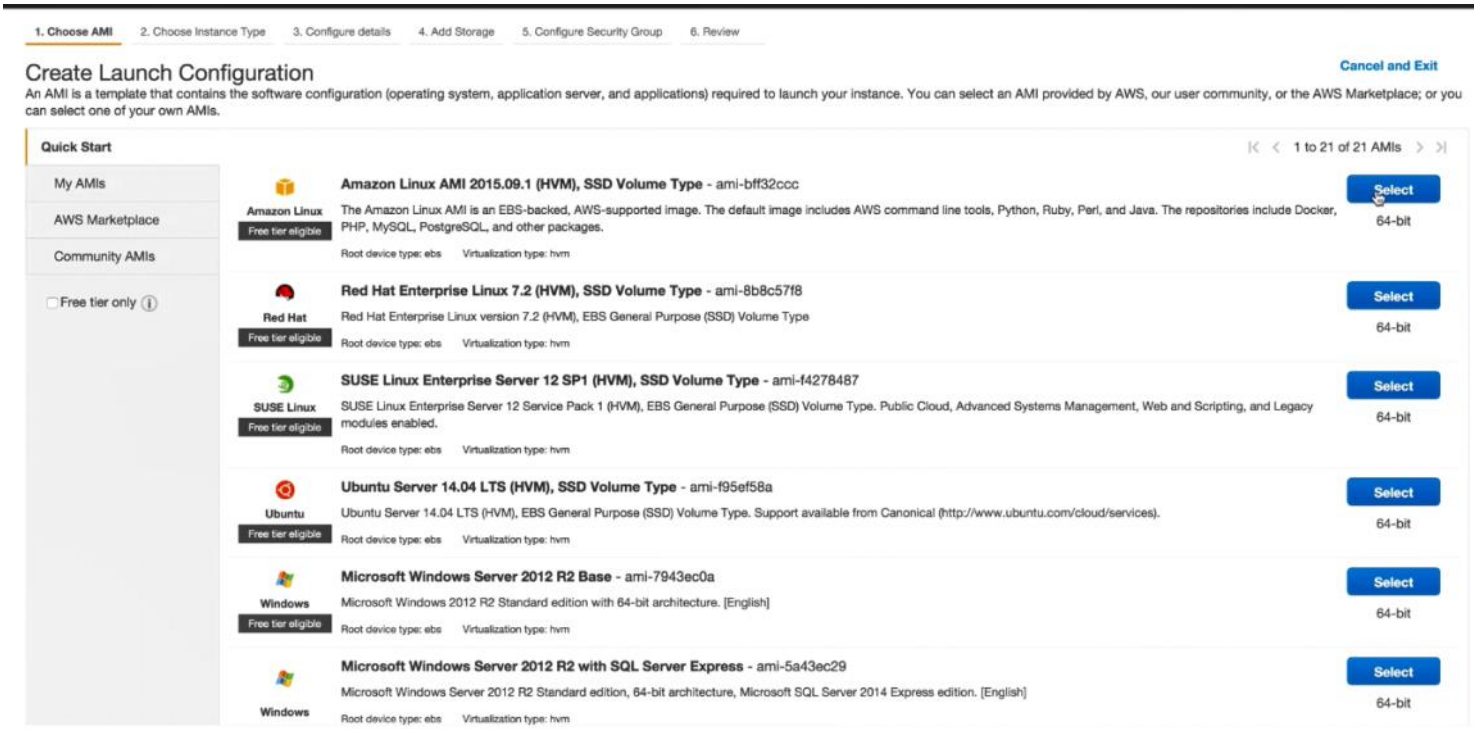
Now we will go to auto scaling section of our EC2 console and from there we will go to launch configurations.
We will see following screen -



From here we will create auto scaling group.

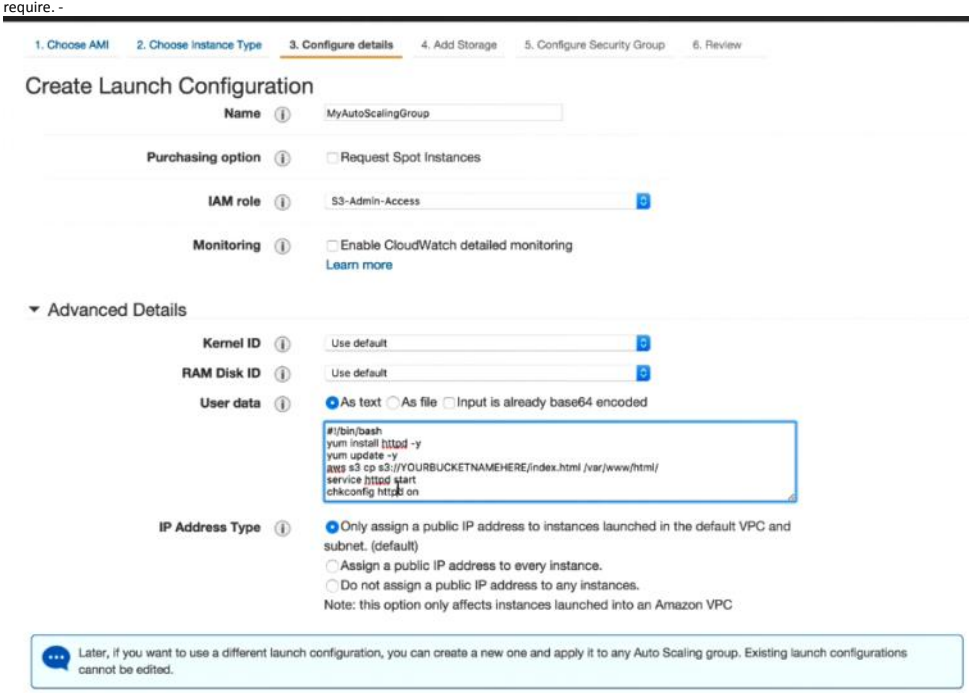


So here before creating any Auto-Scaling group, we have to create launch-configuration group.
We will click on the "create launch configuration".
It will be similar to like creating an EC2 instance.
Here also we will have option to choose AMI -

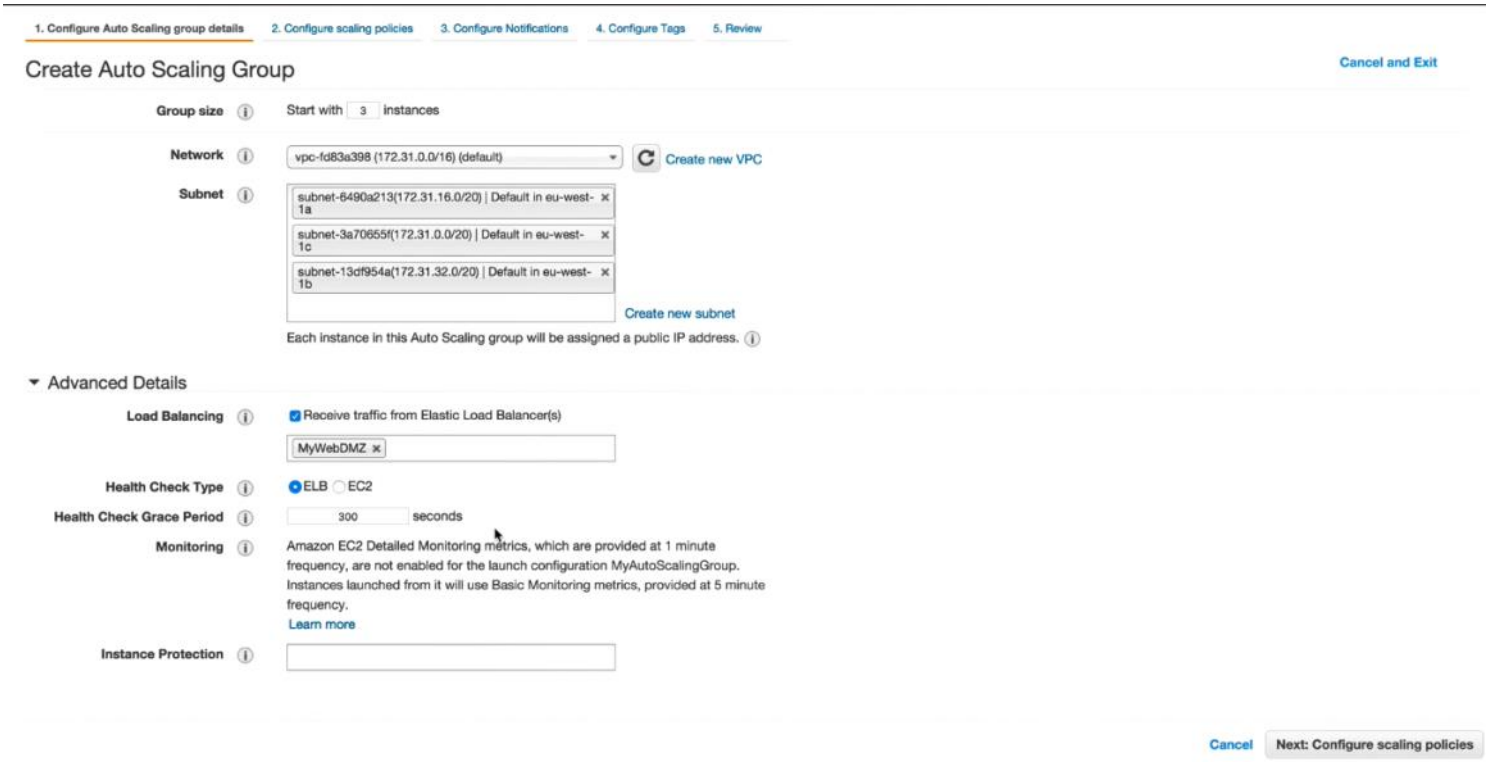


Second window will be same as what we get while creating EC2 machine.

For the third option (configure details) here we have following options, We will fill as per what we require. -



Rest of all the other steps will be same as creating an EC2 machine, at last in case of while creating EC2 machine, after the key popup option, our machine will take some time to get started but here , it will redirect us to this page -

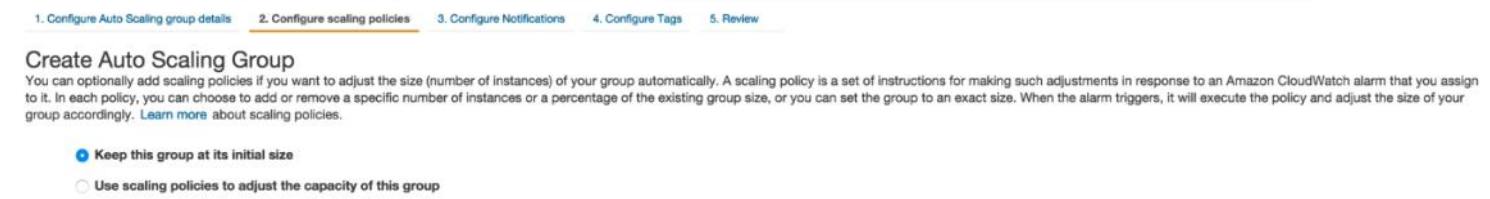


Here we will configure our scaling groups, we have following details to fill in here,

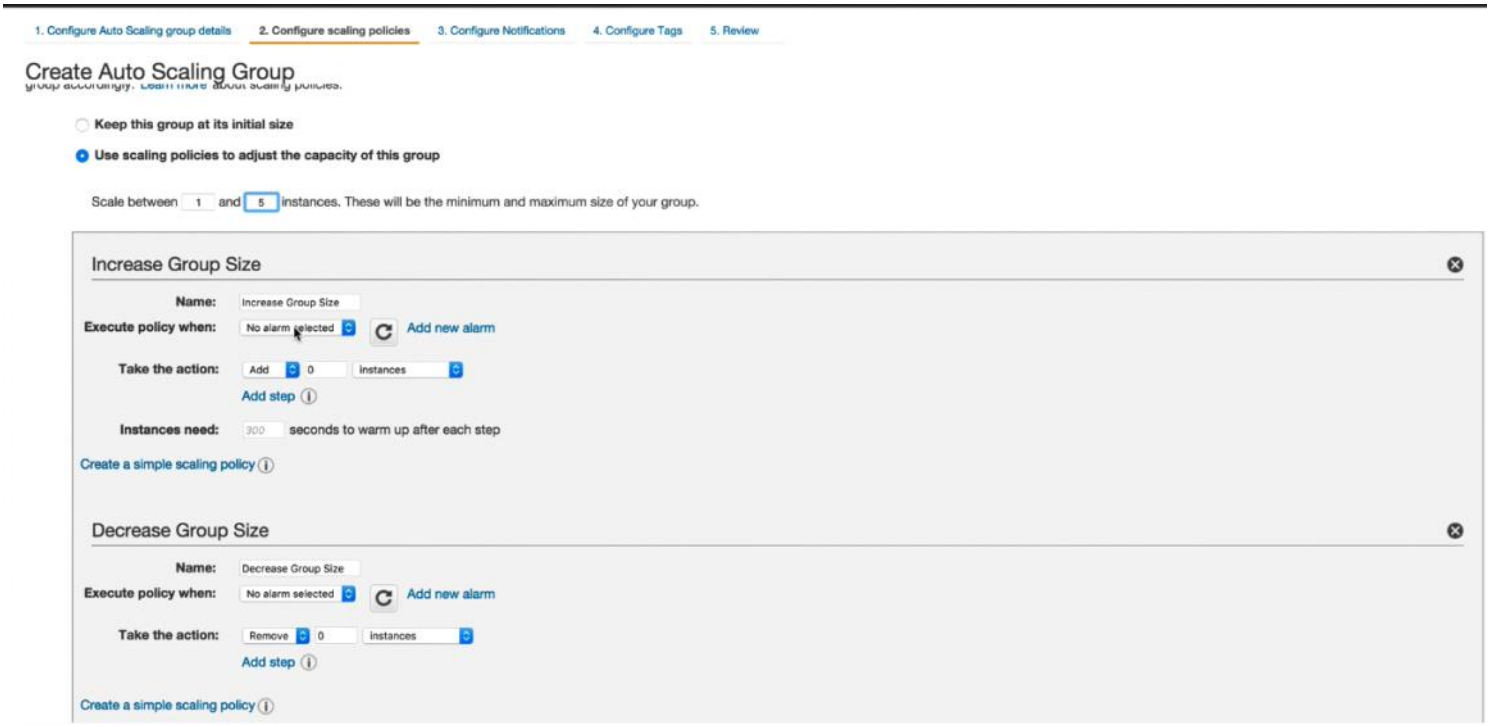
Group size - what size you want for your auto-scaling.
Network - what VPC you need to have (we will learn about VPC later)
Subnet - list of subnet that we want, here one thing to notice is that it is good to use multiple subnet so that you will be easily able to process with your auto-scaling policy. Since we are using all the 3 subnets and we have used Group size =3, hence each and every single will be distributed to those subnets.

Then under the advanced section we have-
Load balancing - whether we want to receive traffic from there or not.
Health Check Type - how we want to check health of our machine.
Grace period - the length of time that Auto-Scaling waits before checking an instance's health status.(default value is 300 and if you use 0 then no grace period).
We will leave rest values as default.

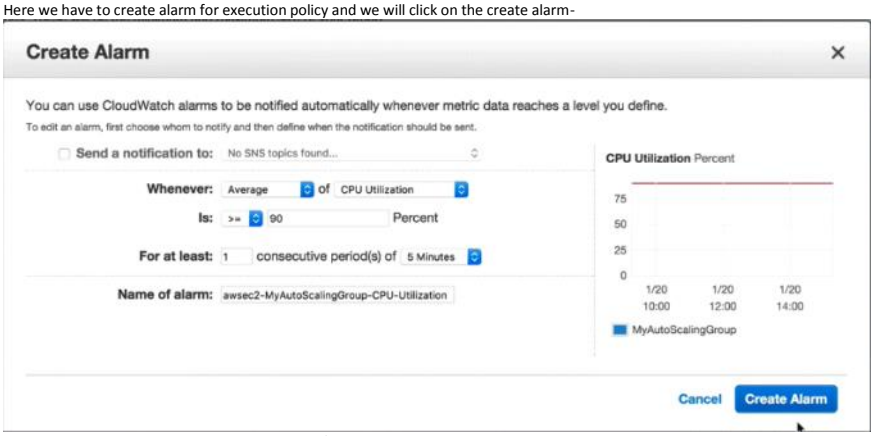
At next we will see following page -



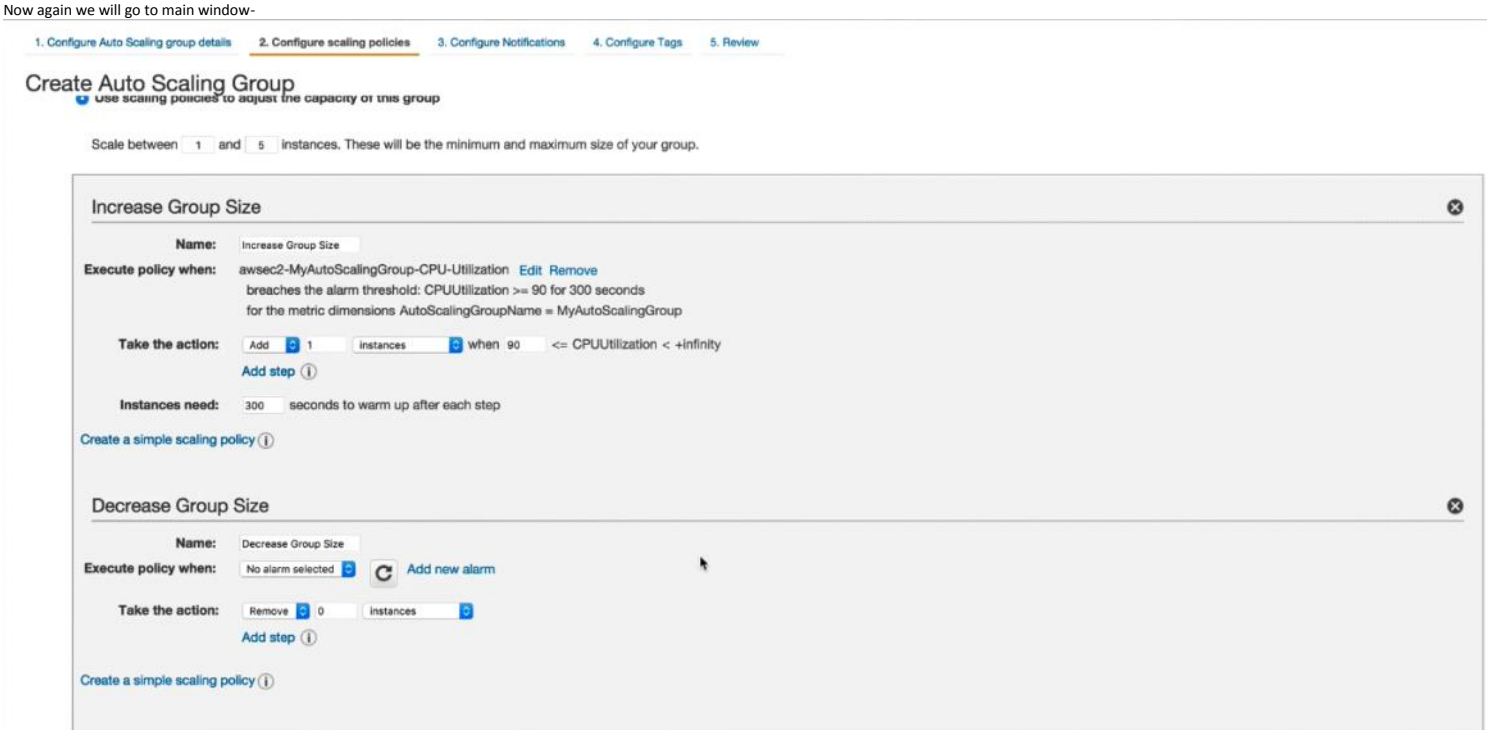
We will select 2nd option-



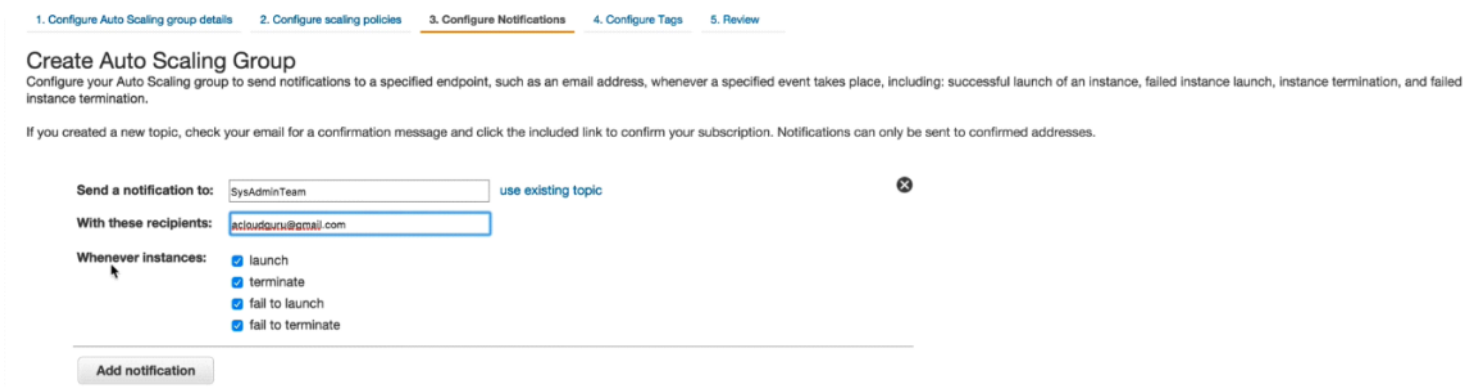
Here we have to create alarm for execution policy and we will click on the create alarm-



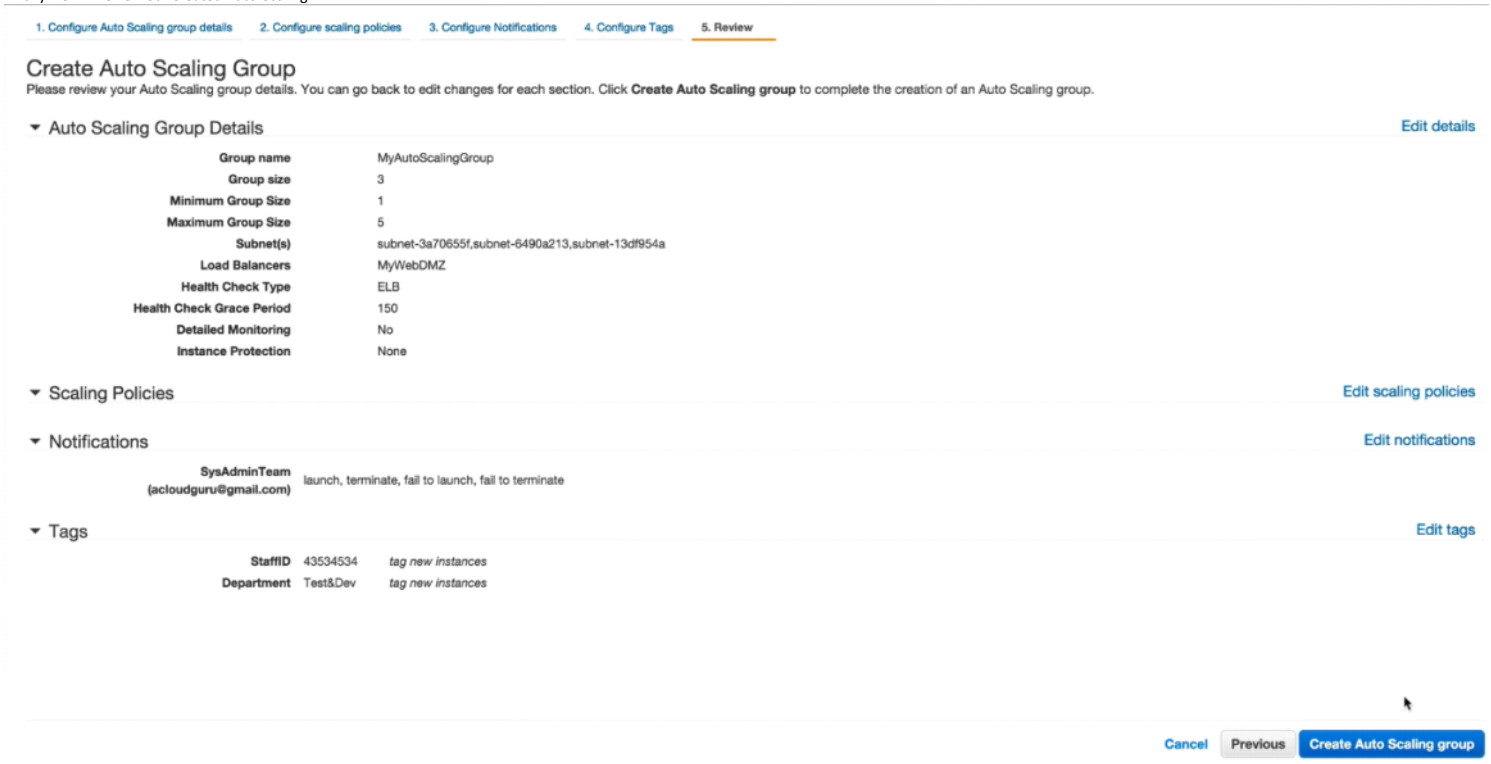
We will create alarm using this prompt, this is self-explanatory -
Now again we will go to main window-



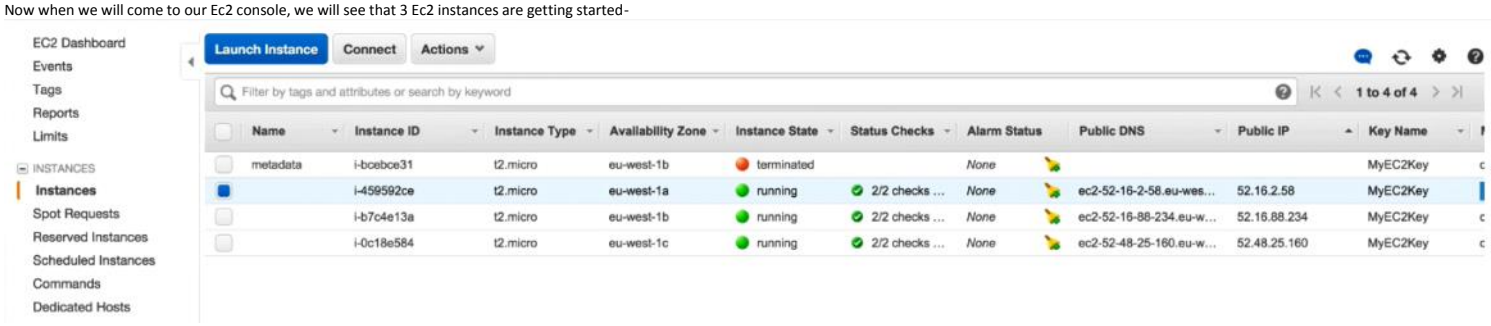
Once our alarm is triggered, it will take action as for increase group size or for decrease group size.
Now we will click next and here we will see-



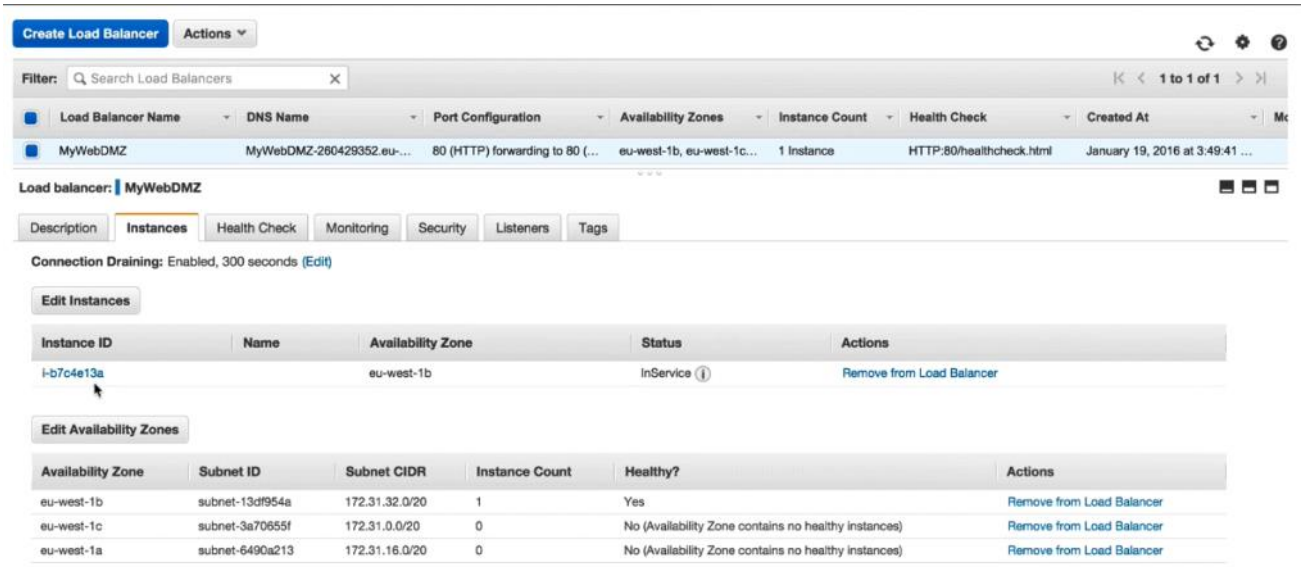
Here we will be able to add whom to send when this happens.
At next again key-pair will come(which we are skipping).
Finally we will review our created Auto-Scaling-



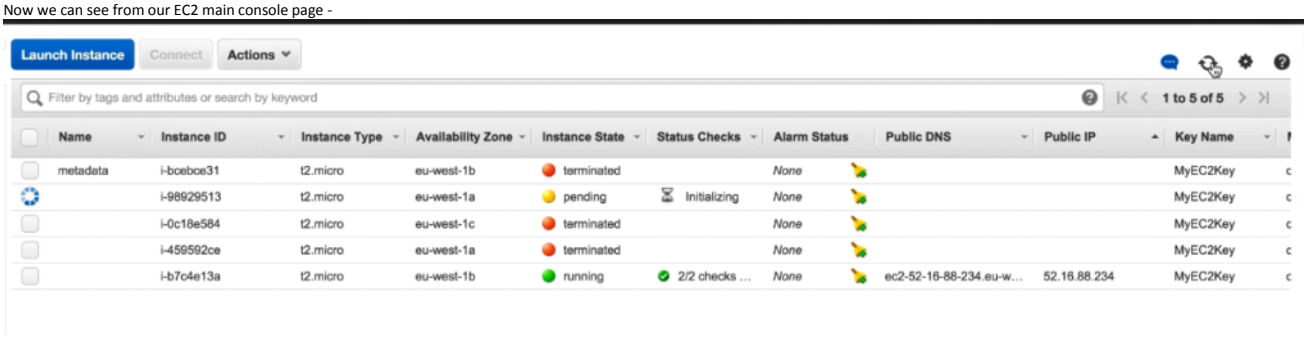
Now when we will come to our Ec2 console, we will see that 3 Ec2 instances are getting started-



Here it is because we choose 3 different subnets for this.
Now let's terminate our running EC2 instances(1st and 3rd).
Now if we see from our load balancer and put DNS name in browser then still it will give us result, since we have still one instance is up and running.
And when we will go to load balancers and hit on the instances, then we will see following window-



Here we can see that only one instance is running.
Now we can see from our EC2 main console page -



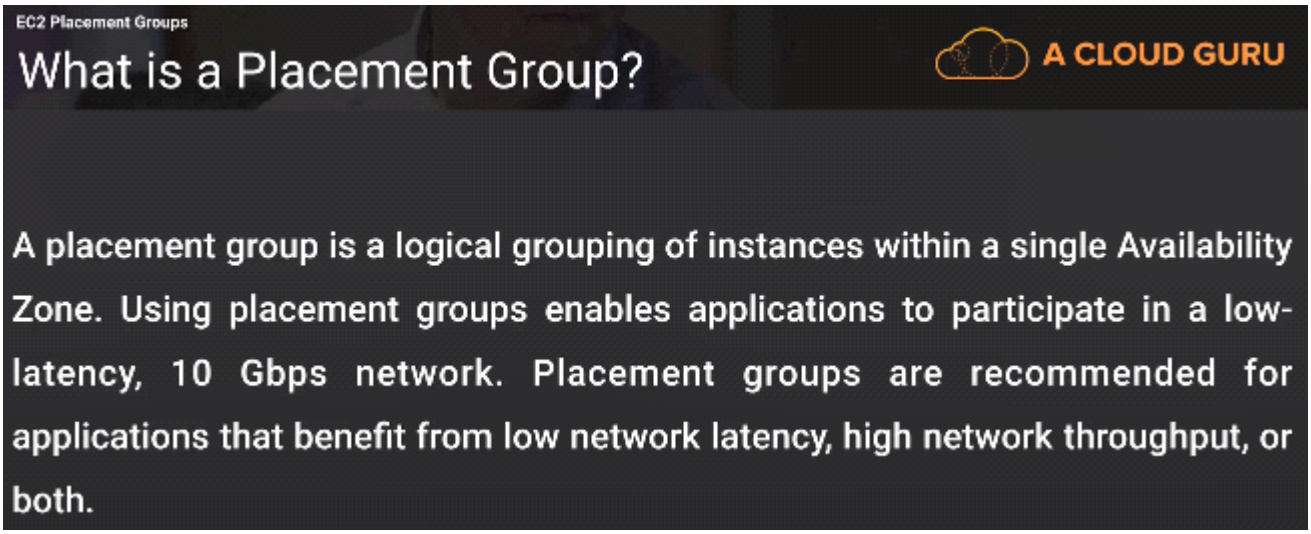
Here we can see that after some time our load balancer will detect that two of our availability servers are down so it will add one more.
And important thing is that my elastic DNS server is still showing us our result.
This will be useful in Route53 (we will learn this later), so in it if our region is not working then whit the help of this we can easily replicate our content to other part of the world.

Once you will delete our Auto-Scaling , it will automatically delete your EC2 instances.

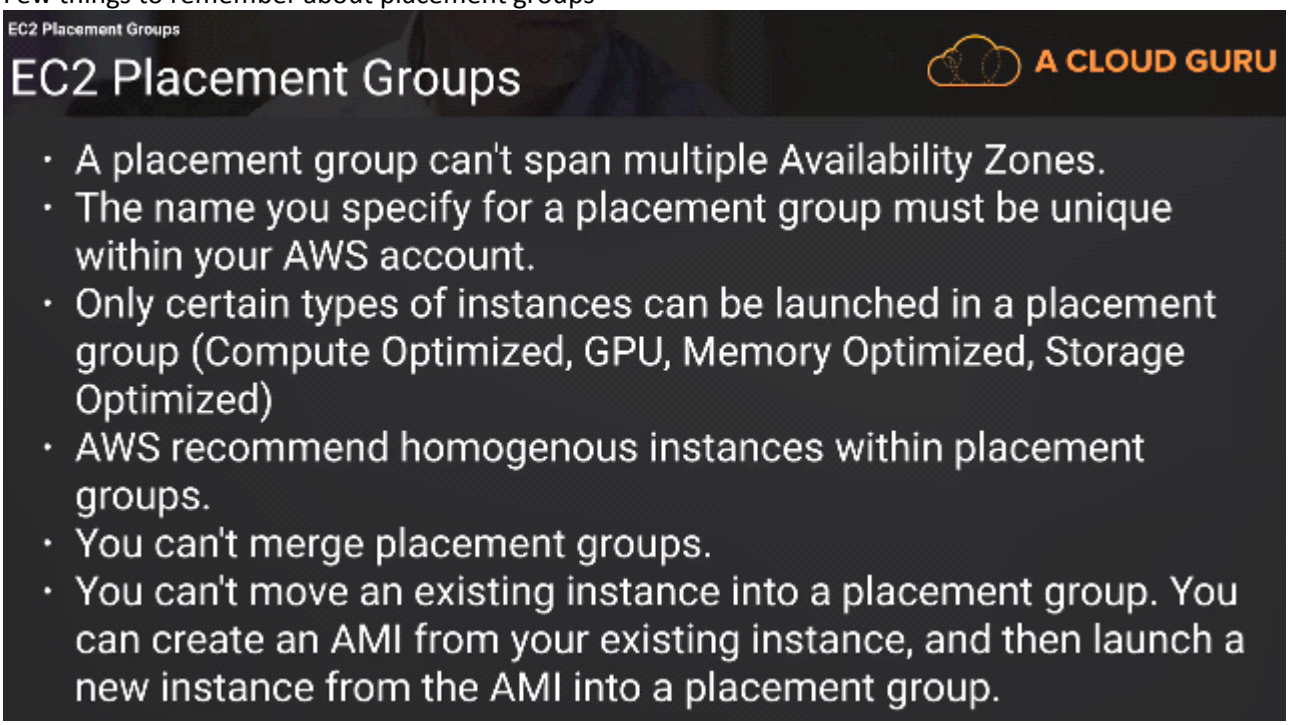
EC2 - Placement Groups

Tuesday, December 27, 2016 4:10 PM

Definition -



Few things to remember about placement groups-

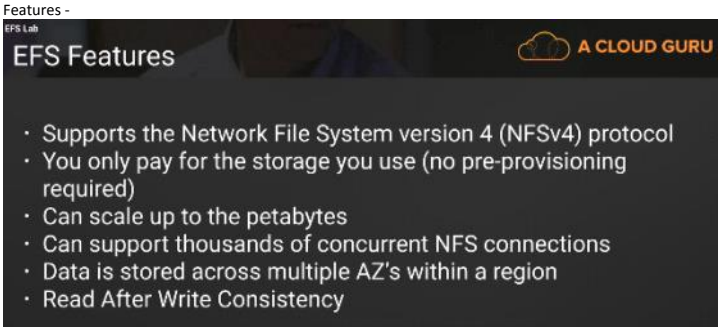


EC2 - EFS - lab

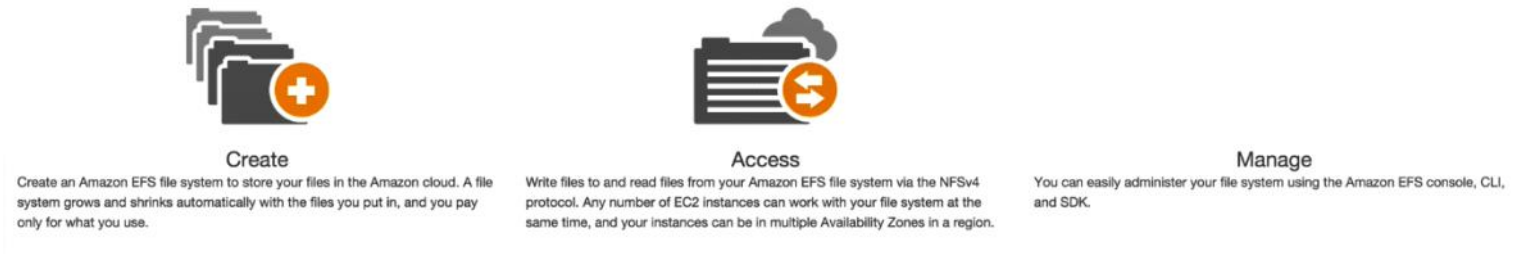
Tuesday, December 27, 2016 4:14 PM

EFS stands for - elastic file system.

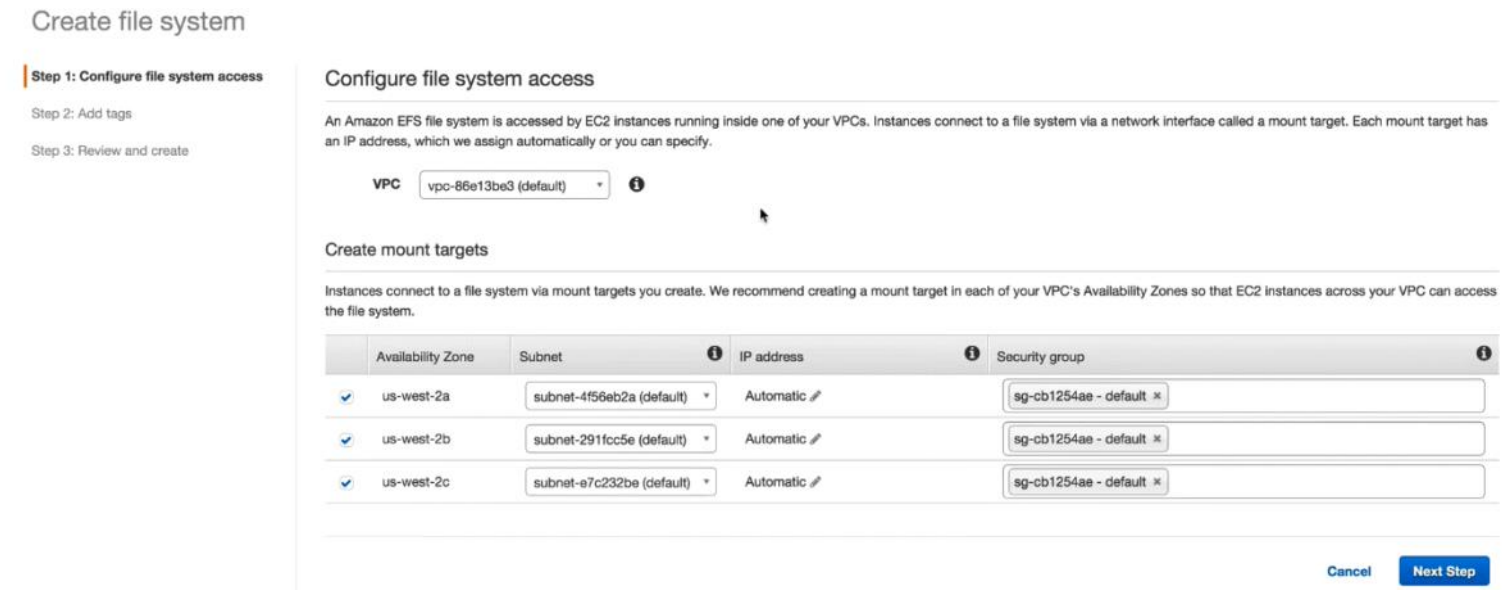
Features -



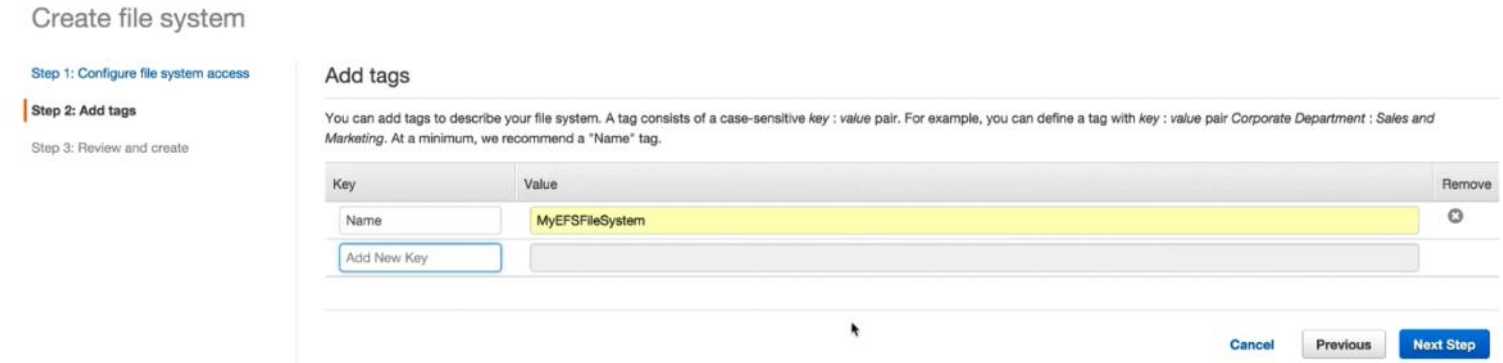
Now we will add EFS -



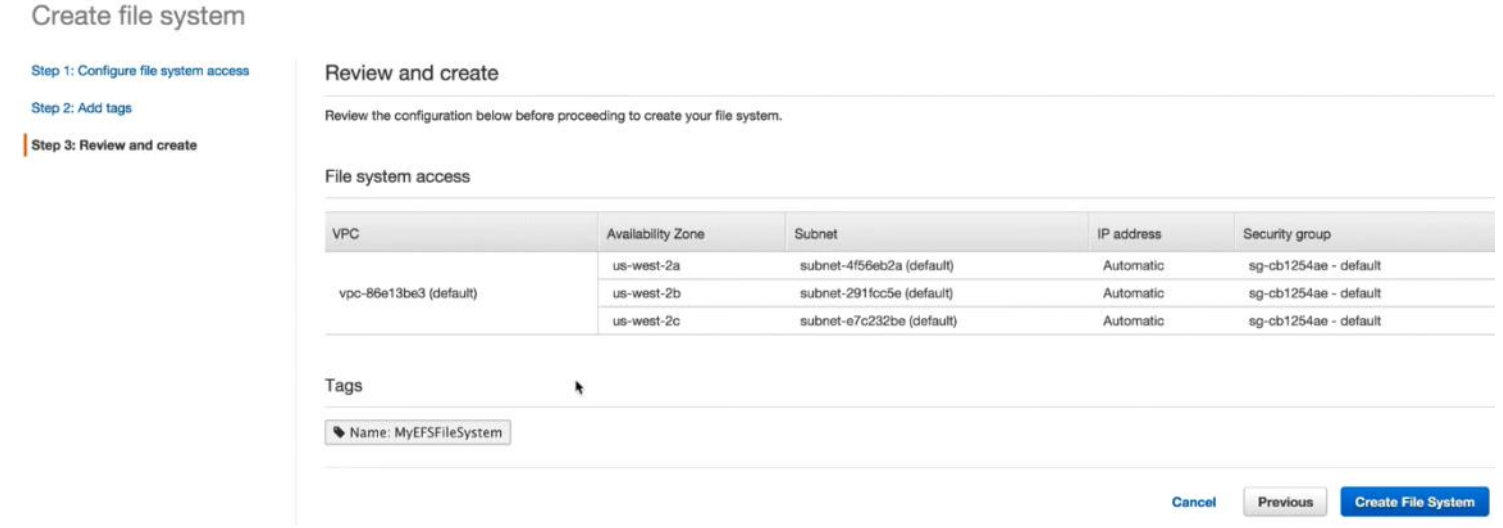
First of all we will create file system for EFS, click on the "Create File System" button.



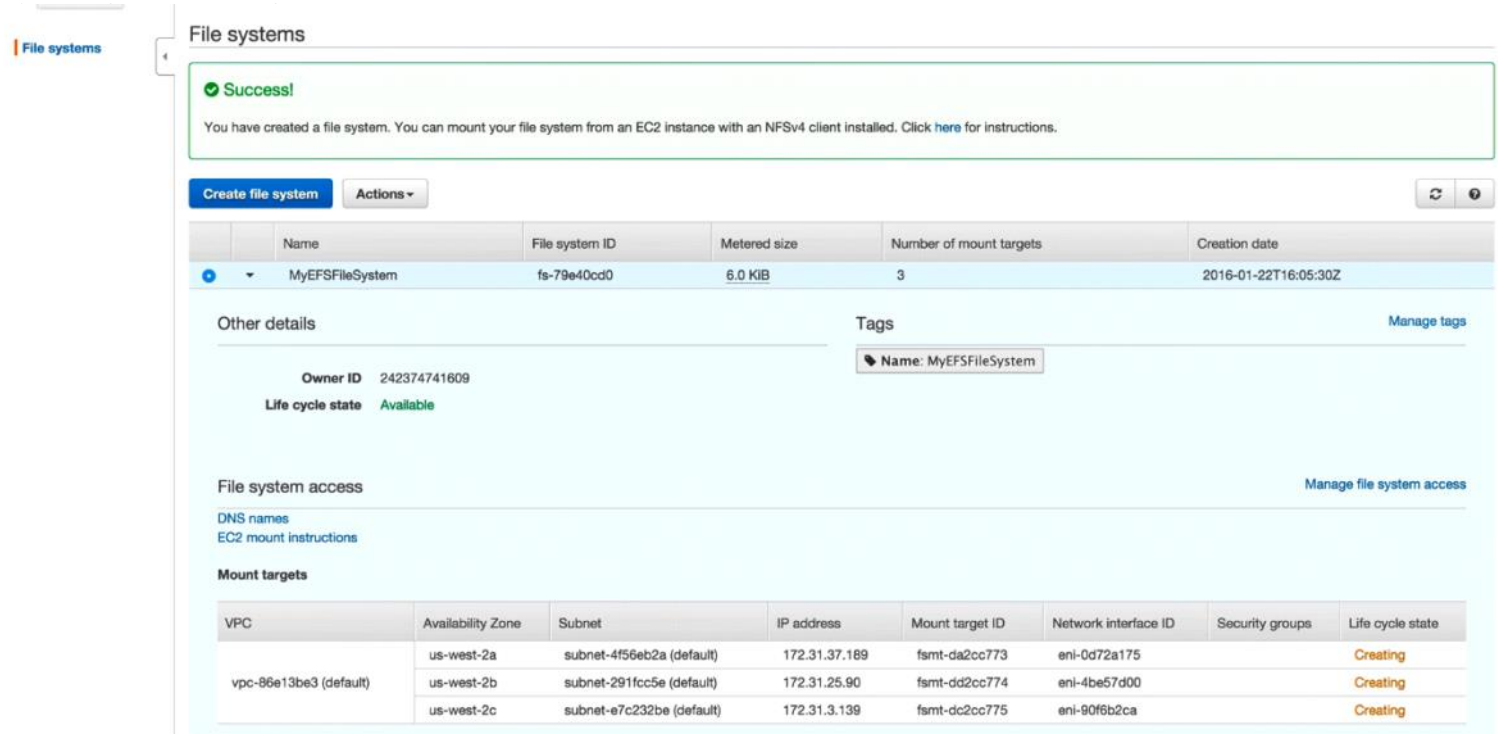
Here we will be able to add VPC to our filesystem and we can add multiple availability zones to our file system. Here options are self-explanatory.
Next window is for adding tags -



Now for third step, we review of our file system -

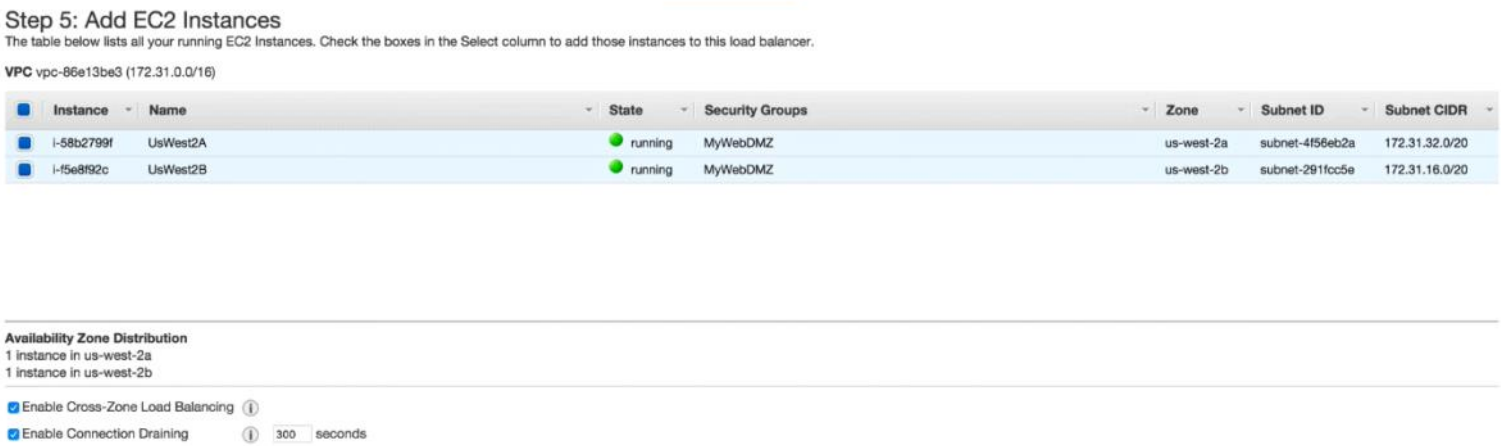


Finally after clicking OK we will see following window -

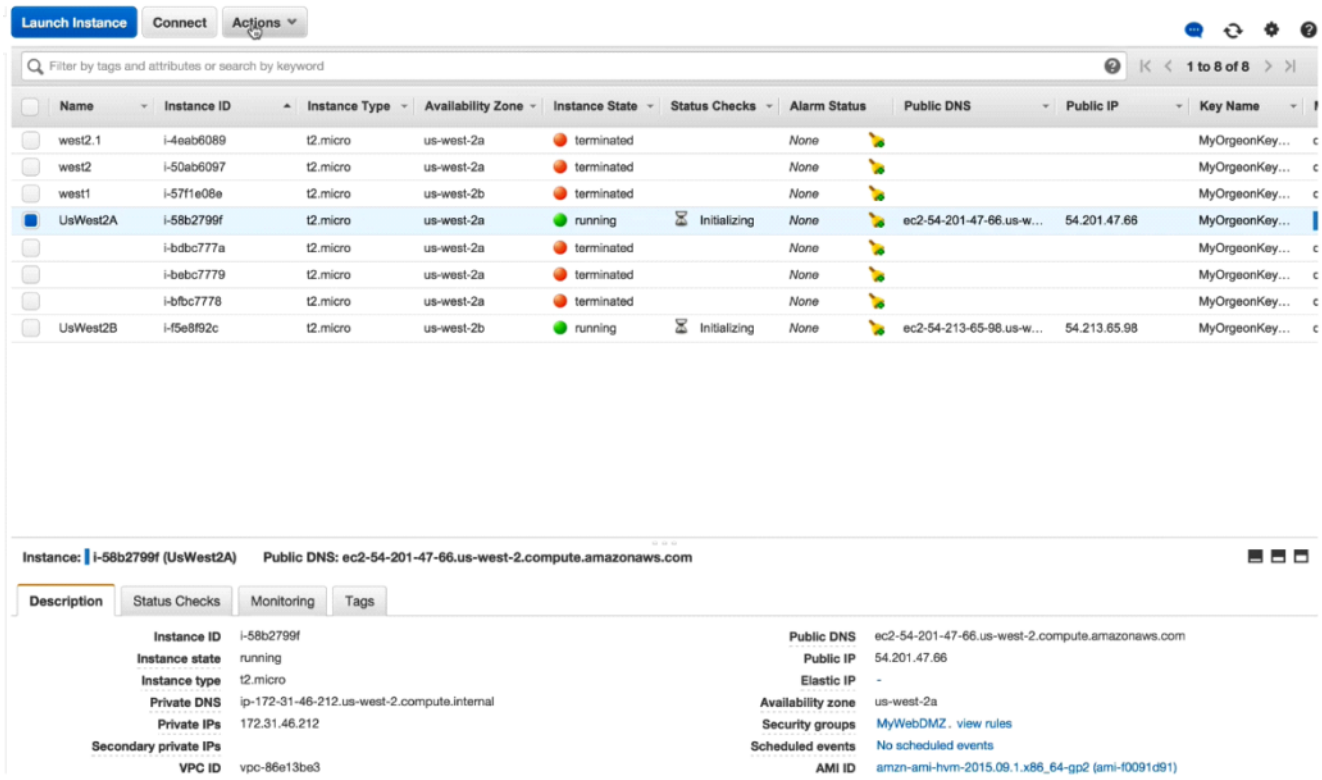


Our file system has been created, now we will move to our EC2 console and we will create one new EC2 machine. Again we will launch one more new EC2 instance, here we will add 2 different availability groups since we have 3 different.
Now we will move to Load balancer and we will create new load balancer.

So here when we will create our load balancer, at step no 5 EC2 instances we will have 2 options since we have 2 running EC2 machines-

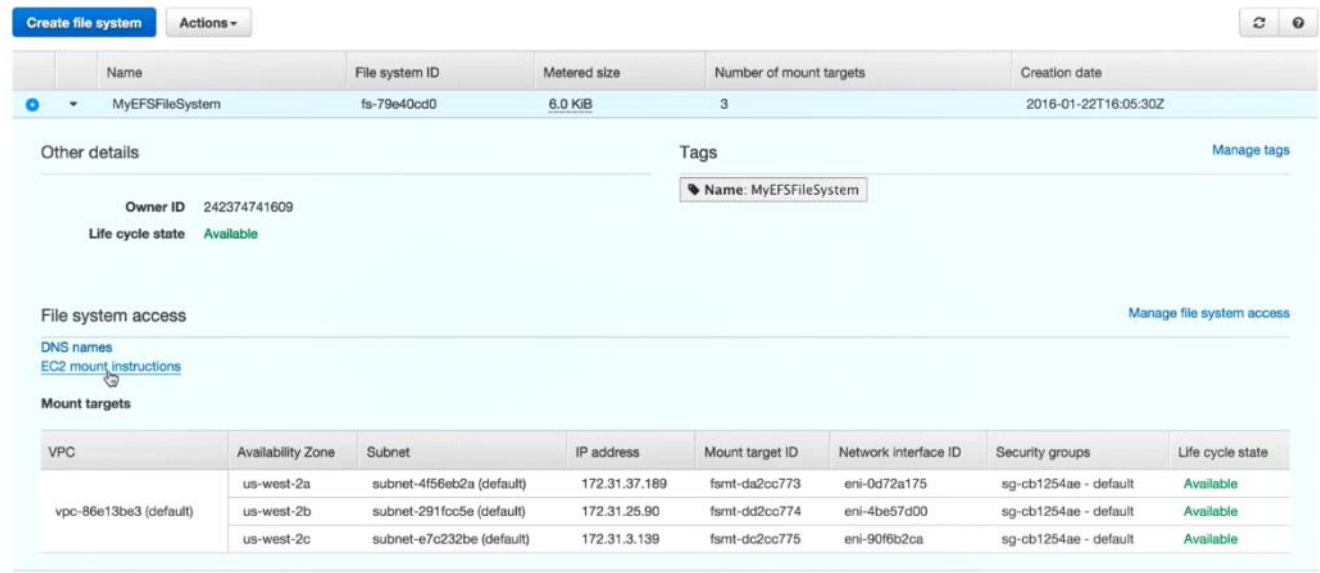


Now we will see that our 2 EC2 machines are running -

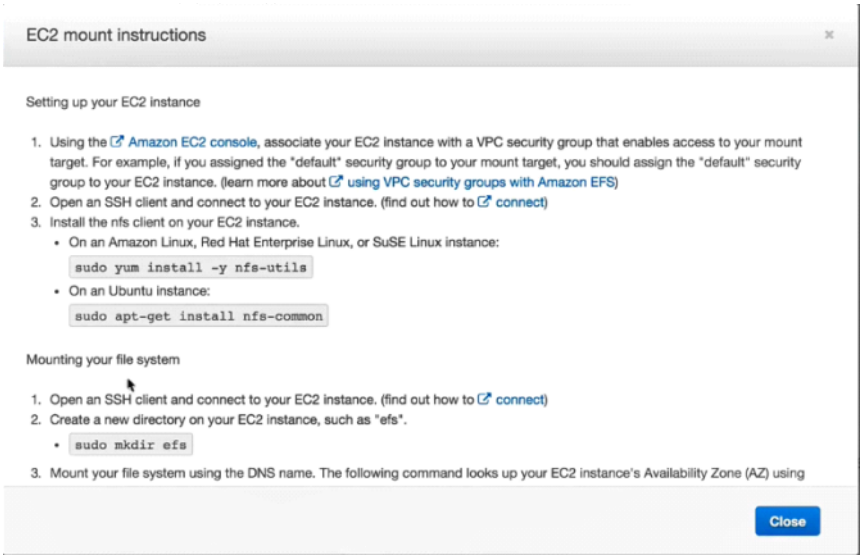


Look here only for running instances.
One thing to notice is that you must have both the EC2 instances under same security group, for what you have created your EFS file system.
In order to change security group, go to action > networking > change security group and add default as well.

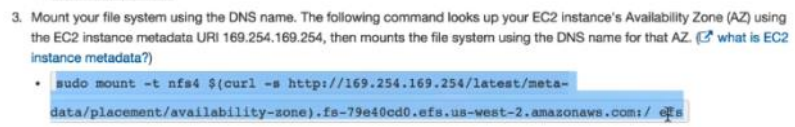
Now we will go to our EFS console -



Here we can see link for EC2 mount instruction,
We click on this link and a pop-up will appear -



But this will mount to EFS directory but we want to install this to apache directory (/var/www/html) .
Now for this command-



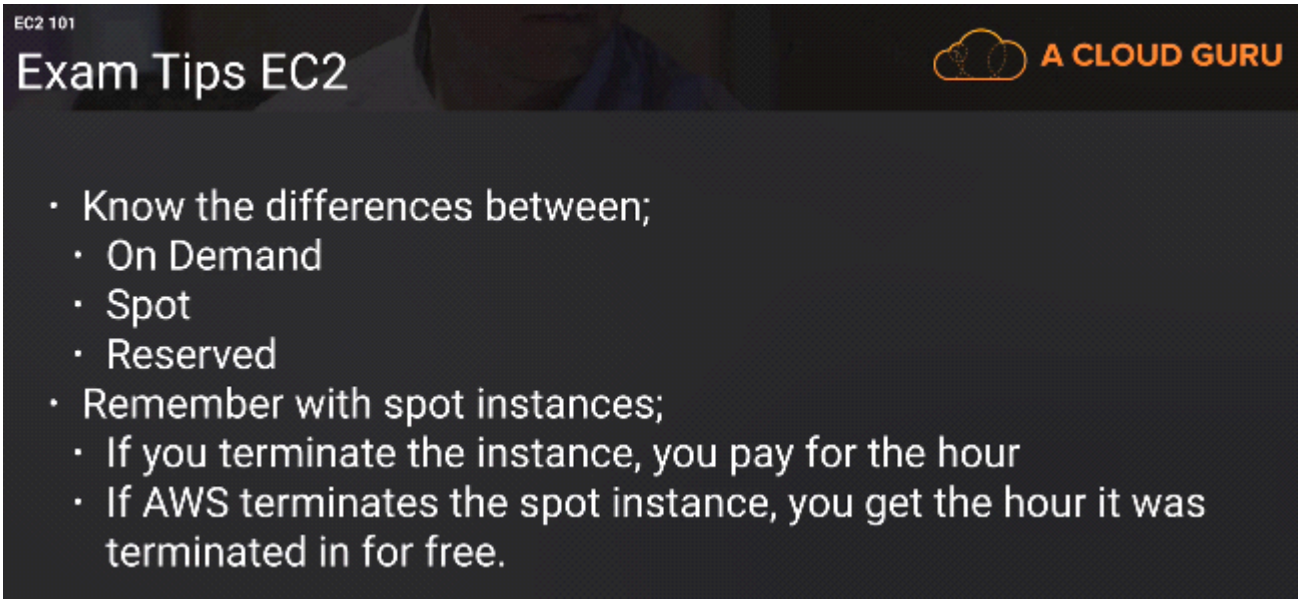
We will replace "efs" with "/var/www/html".
We will apply this command to both of our machines and then we will be able to see that we are able to access this via both the machines.

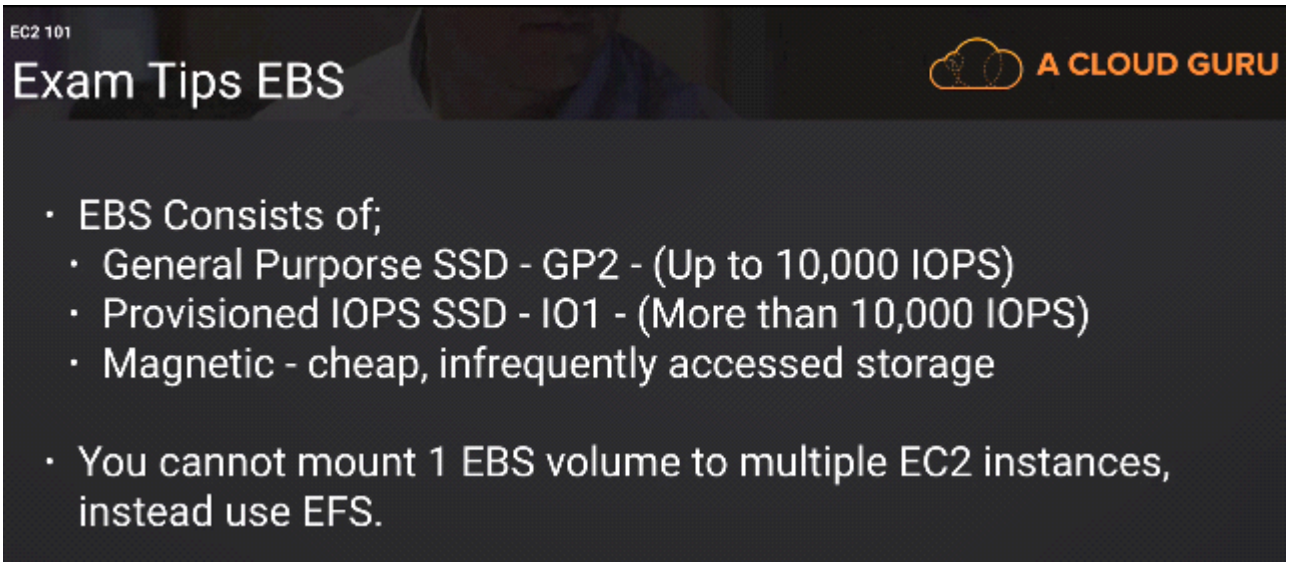
Now when we go to our load balancer dashboard and go to instances tab then we will see that -

Both of our servers are InService.

EC2 - Summary

Tuesday, December 27, 2016 4:50 PM





EC2 101

EC2 Instance Types

A CLOUD GURU

Family	Speciality	Use case
T2	Lowest Cost, General Purpose	Web Servers/Small DBs
M4	General Purpose	Application Servers
M3	General Purpose	Application Servers
C4	Compute Optimized	CPU Intensive Apps/DBs
C3	Compute Optimized	CPU Intensive Apps/DBs
R3	Memory Optimized	Memory Intensive Apps/DBs
G2	Graphics/General Purpose GPU	Video Encoding/Machine Learning/3D Application Streaming
I2	High Speed Storage	NoSQL DBs, Data Warehousing etc
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop

EC2 - Lab

EC2 Lab Exam Tips

A CLOUD GURU

- Termination Protection is turned off by default, you must turn it on.
- On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated.
- Root Volumes cannot be encrypted by default, you need a third party tool (such as bit locker etc) to encrypt the root volume.
- Additional volumes can be encrypted.

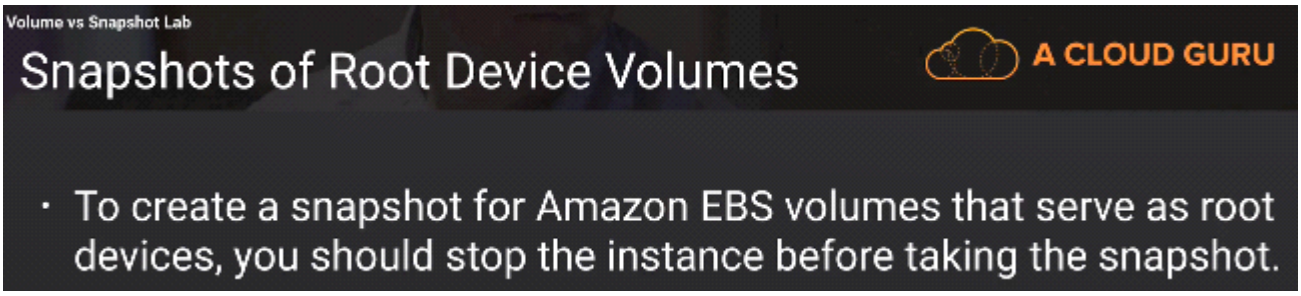
Volume vs Snapshot Lab

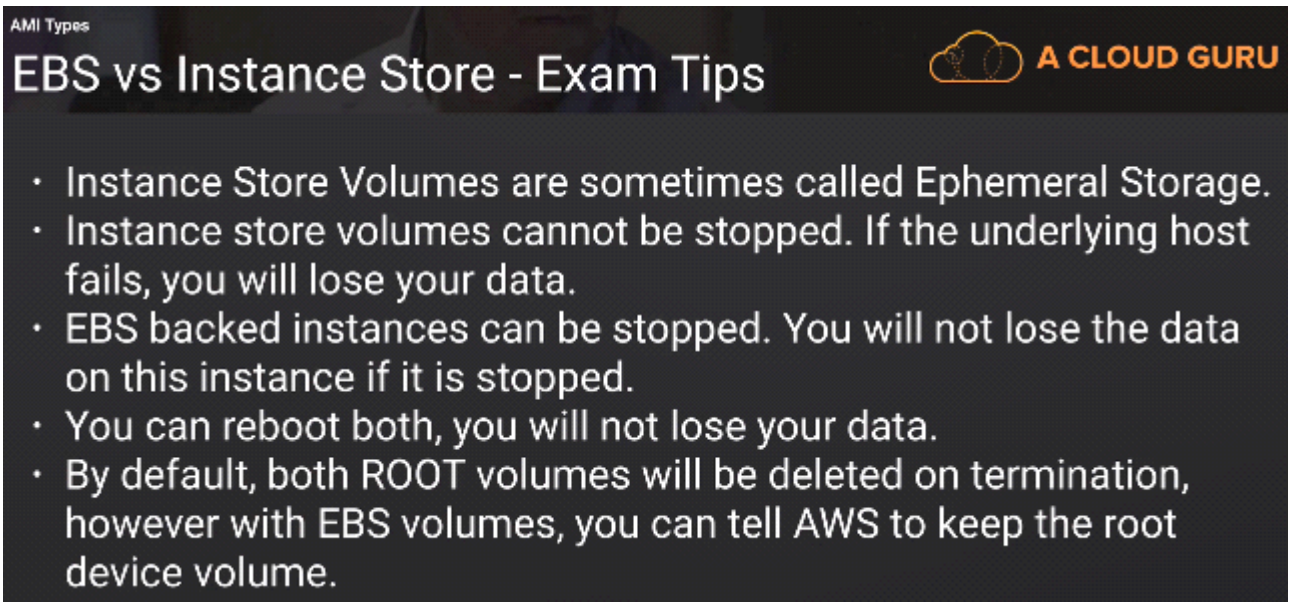
Volumes vs Snapshots

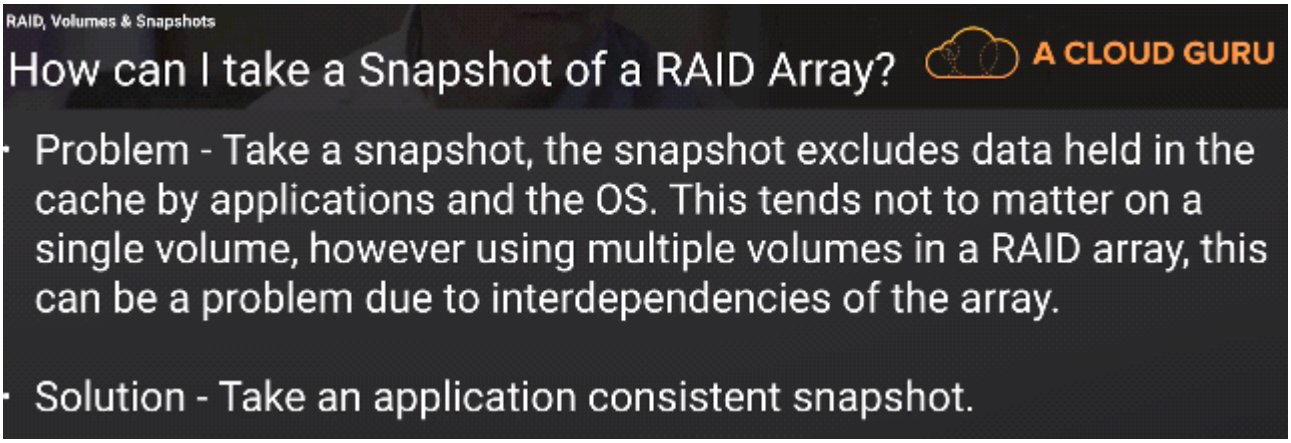
A CLOUD GURU

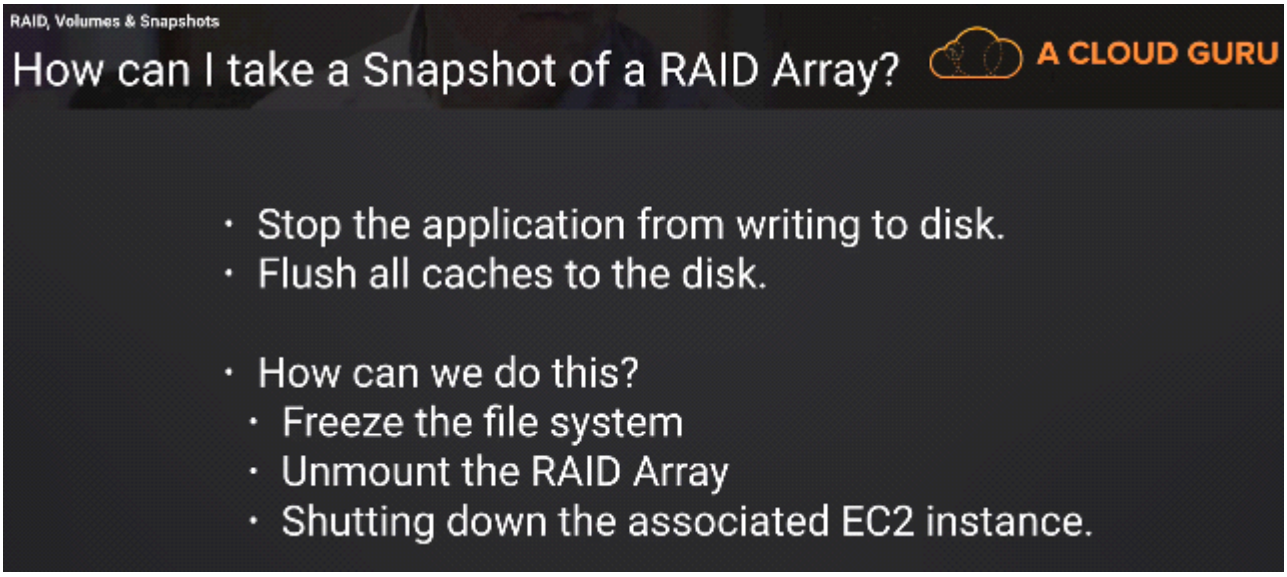
- Volumes exist on EBS
  - Virtual Hard Disk
- Snapshots exist on S3
- You can take a snapshot of a volume, this will store that volume on S3.
- Snapshots are point in time copies of Volumes.
- Snapshots are incremental, this means that only the blocks that have changed since your last snapshot are moved to S3.

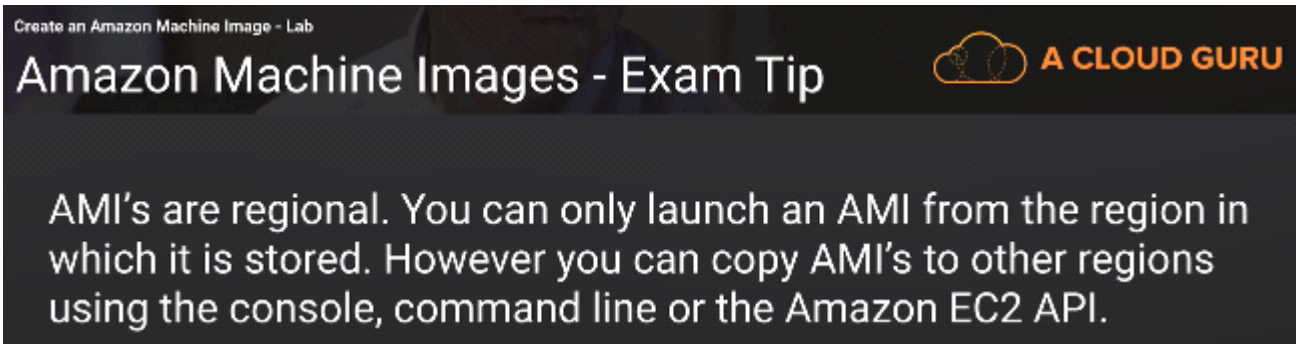

Volume vs Snapshot Lab
Volumes vs Snapshots - Security
A CLOUD GURU
• Snapshots of encrypted volumes are encrypted automatically.
• Volumes restored from encrypted snapshots are encrypted automatically.
• You can share snapshots, but only if they are unencrypted.
• These snapshots can be shared with other AWS accounts or made public

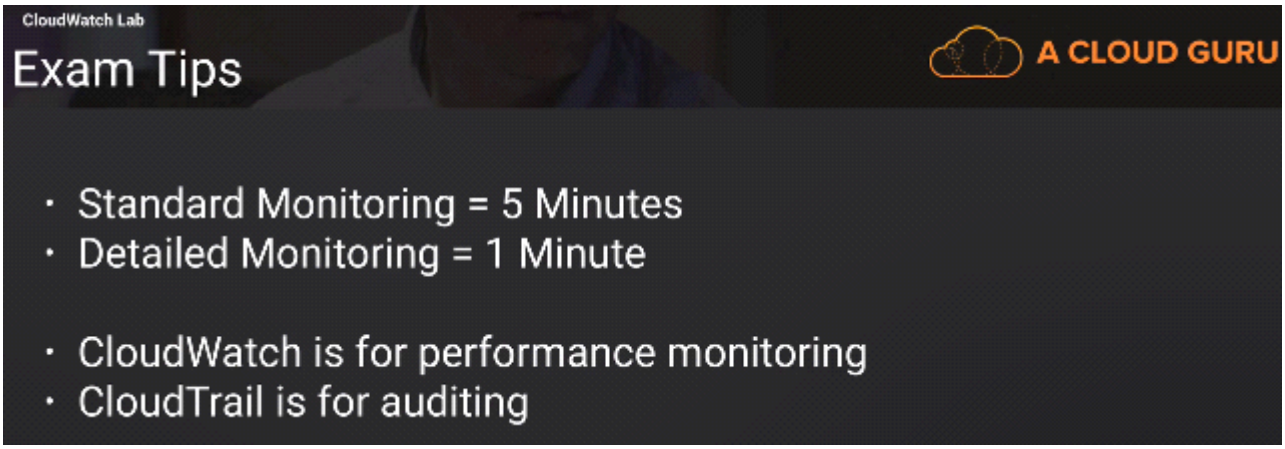

Volume vs Snapshot Lab
Snapshots of Root Device Volumes
A CLOUD GURU
• To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.


AMI Types
EBS vs Instance Store - Exam Tips
A CLOUD GURU
• Instance Store Volumes are sometimes called Ephemeral Storage.
• Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data.
• EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped.
• You can reboot both, you will not lose your data.
• By default, both ROOT volumes will be deleted on termination, however with EBS volumes, you can tell AWS to keep the root device volume.

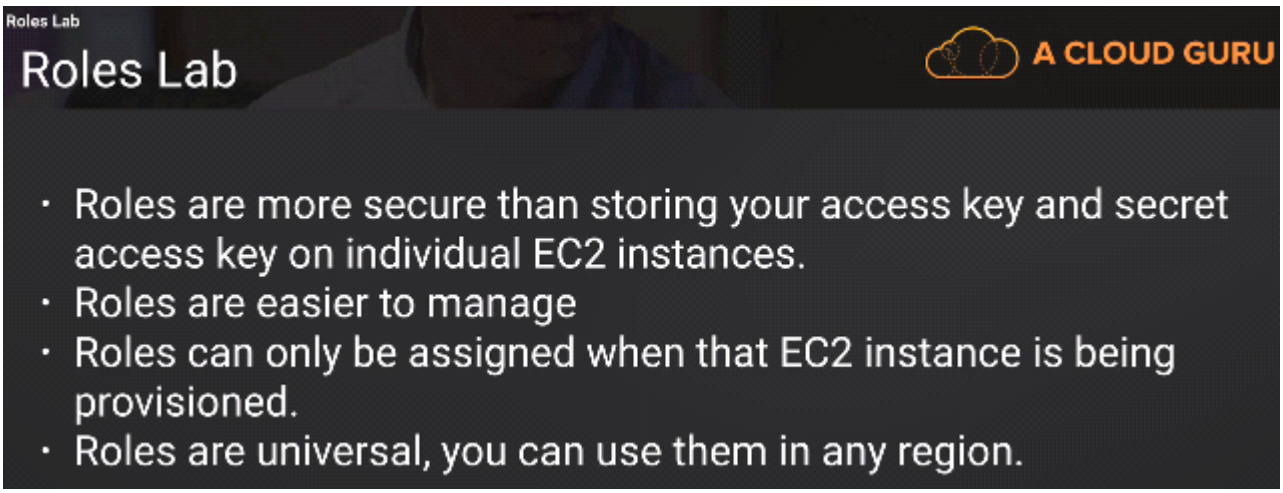

RAID, Volumes & Snapshots
How can I take a Snapshot of a RAID Array?
A CLOUD GURU
Problem - Take a snapshot, the snapshot excludes data held in the cache by applications and the OS. This tends not to matter on a single volume, however using multiple volumes in a RAID array, this can be a problem due to interdependencies of the array.
Solution - Take an application consistent snapshot.

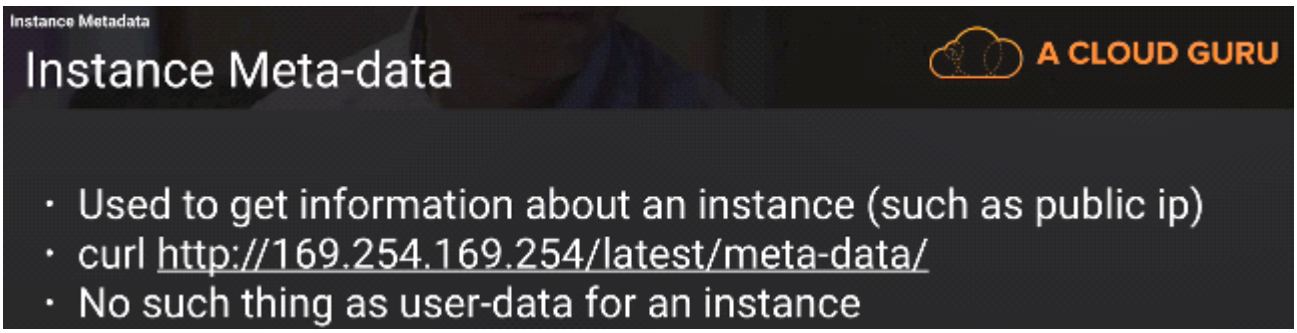

RAID, Volumes & Snapshots
How can I take a Snapshot of a RAID Array?
A CLOUD GURU
Stop the application from writing to disk.
Flush all caches to the disk.
How can we do this?
Freeze the file system
Unmount the RAID Array
Shutting down the associated EC2 instance.

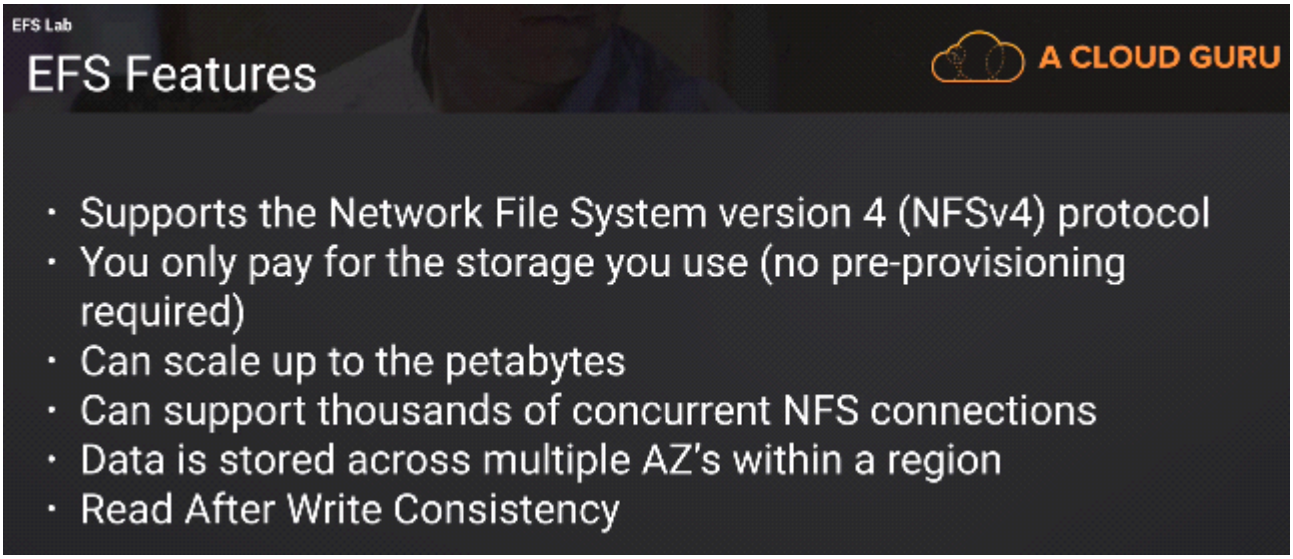

Create an Amazon Machine Image - Lab
Amazon Machine Images - Exam Tip
A CLOUD GURU
AMI's are regional. You can only launch an AMI from the region in which it is stored. However you can copy AMI's to other regions using the console, command line or the Amazon EC2 API.

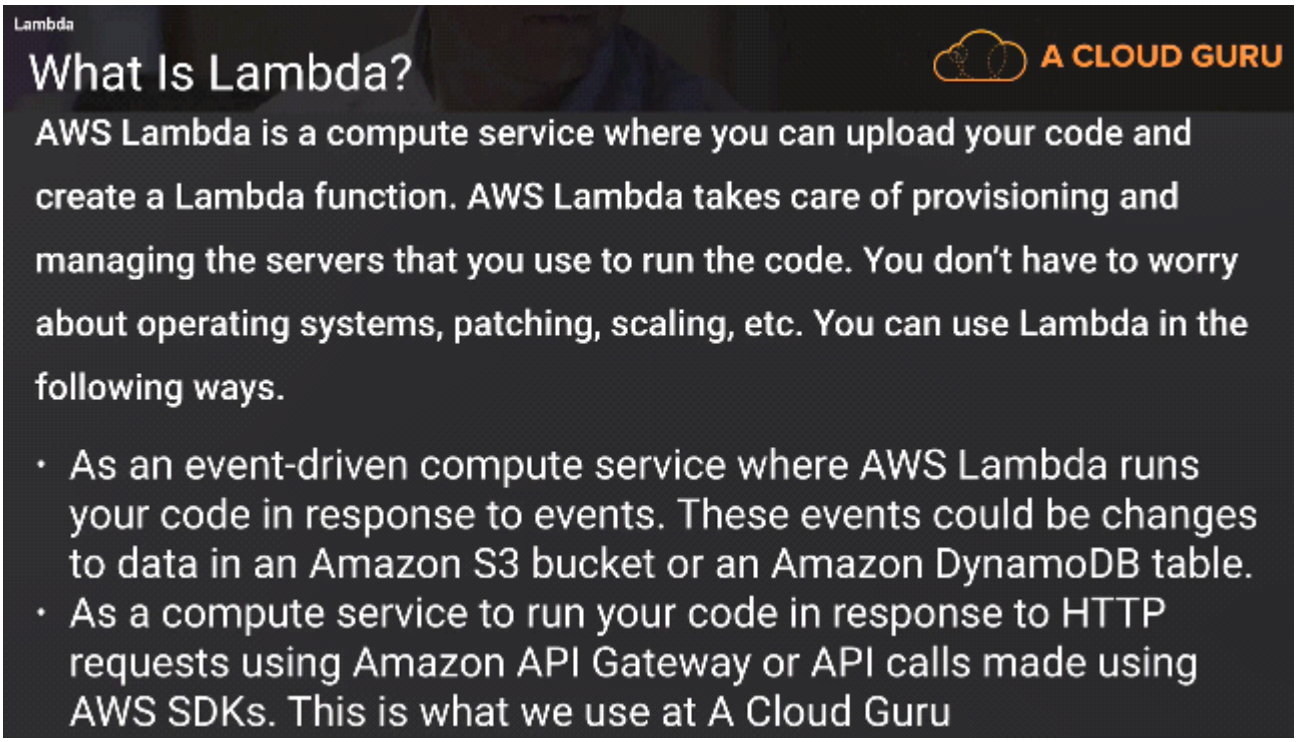

CloudWatch Lab
Exam Tips
A CLOUD GURU
Standard Monitoring = 5 Minutes
Detailed Monitoring = 1 Minute
CloudWatch is for performance monitoring
CloudTrail is for auditing


CloudWatch Lab
What can I do with Cloudwatch?
A CLOUD GURU
• Dashboards - Creates awesome dashboards to see what is happening with your AWS environment.
• Alarms - Allows you to set Alarms that notify you when particular thresholds are hit.
• Events - CloudWatch Events helps you to respond to state changes in your AWS resources.
• Logs - CloudWatch Logs helps you to aggregate, monitor, and store logs.


Roles Lab
Roles Lab
A CLOUD GURU
• Roles are more secure than storing your access key and secret access key on individual EC2 instances.
• Roles are easier to manage
• Roles can only be assigned when that EC2 instance is being provisioned.
• Roles are universal, you can use them in any region.


Instance Metadata
Instance Meta-data
A CLOUD GURU
• Used to get information about an instance (such as public ip)
• curl http://169.254.169.254/latest/meta-data/
• No such thing as user-data for an instance


EFS Lab
EFS Features
A CLOUD GURU
• Supports the Network File System version 4 (NFSv4) protocol
• You only pay for the storage you use (no pre-provisioning required)
• Can scale up to the petabytes
• Can support thousands of concurrent NFS connections
• Data is stored across multiple AZ's within a region
• Read After Write Consistency


Lambda
What Is Lambda?
A CLOUD GURU
AWS Lambda is a compute service where you can upload your code and create a Lambda function. AWS Lambda takes care of provisioning and managing the servers that you use to run the code. You don't have to worry about operating systems, patching, scaling, etc. You can use Lambda in the following ways.
• As an event-driven compute service where AWS Lambda runs your code in response to events. These events could be changes to data in an Amazon S3 bucket or an Amazon DynamoDB table.
• As a compute service to run your code in response to HTTP requests using Amazon API Gateway or API calls made using AWS SDKs. This is what we use at A Cloud Guru