

# 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?



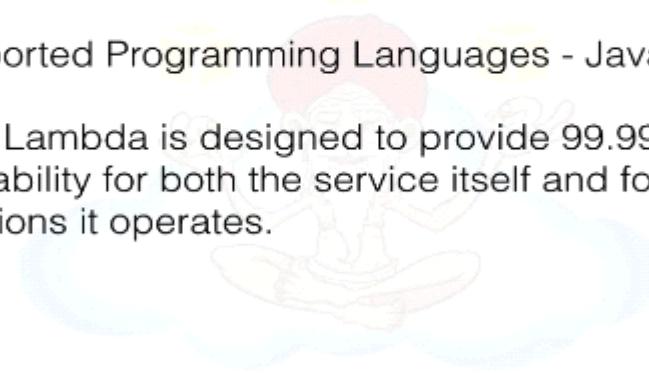
You can use AWS Lambda to respond to table updates in Amazon DynamoDB, modifications to objects in Amazon S3 buckets, messages arriving in an Amazon Kinesis stream, AWS API call logs created by AWS CloudTrail, and custom events from mobile applications, web applications, or other web services.

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.



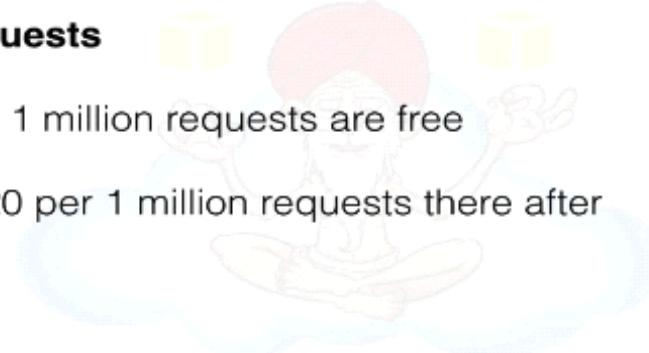
## Lambda - Pricing



### 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 -**

EC2 101

What is EC2?



A CLOUD GURU



**Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.**

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).

EC2 101

What is EC2?



A CLOUD GURU



**Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.**

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

# EC2 Options



- On Demand - allow you to pay a fixed rate by the hour with no commitment.
- Reserved - provide you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. 1 Year or 3 Year Terms
- Spot - enable you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times.

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

## On Demand



- Users that want the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment
- Applications with short term, spiky, or unpredictable workloads that cannot be interrupted
- Applications being developed or tested on Amazon EC2 for the first time

## Reserved



- Applications with steady state or predictable usage
- Applications that require reserved capacity
- Users able to make upfront payments to reduce their total computing costs even further

# Spot



- Applications that have flexible start and end times
- Applications that are only feasible at very low compute prices
- Users with urgent computing needs for large amounts of additional capacity

## Spot Prices - Exam Tip



**If the Spot instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for any hour in which the instance ran.**

EC2 instance Types (Just for information)-

Family	Specialty	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	File servers/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.

EC2 101

## What is EBS?



A CLOUD GURU

**Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use a block device. Amazon EBS volumes are placed in a specific Availability Zone, where they are automatically replicated to protect you from the failure of a single component.**

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-

EC2 101

## EBS Volume Types



A CLOUD GURU

- General Purpose SSD (GP2)
  - Designed for 99.999% availability
  - Ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for short periods for volumes under 1Gib.
- Provisioned IOPS SSD (IO1)
  - Designed for I/O intensive applications such as large relational or NoSQL databases. Use if you need more than 10,000 IOPS
- Magnetic (Standard)
  - Lowest cost per gigabyte of all EBS volume types. Magnetic volumes are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important.

Exam Tips -

## Exam Tips EC2



- 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.

## Exam Tips EBS



- EBS Consists of;
  - General Purpose 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-

The screenshot shows the AWS EC2 Dashboard. The left sidebar contains a navigation menu with sections like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling, and Launch Configurations. The main content area has three main sections: 'Resources' (listing 0 Running Instances, 0 Dedicated Hosts, 0 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 0 Snapshots, 0 Load Balancers, and 1 Security Groups), 'Create Instance' (with a 'Launch Instance' button), and 'Service Health' (showing EU West (Ireland) status: operating normally for EU West (Ireland) and availability zones eu-west-1a, eu-west-1b, and eu-west-1c). On the right, there's an 'Account Attributes' section with links to Supported Platforms (VPC, vpc-fd83a398), Default VPC (vpc-fd83a398), and Resource ID length management. Below that is an 'Additional Information' section with links to Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us. The bottom right features the 'AWS Marketplace' section, which lists products like Tableau Server (10 users), SAP HANA One 244GiB, and others.

**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.

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

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

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

**Quick Start**

		1 to 22 of 22 AMIs >		
<input type="checkbox"/> My AMIs	 <b>Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-bff32ccc</b>	Amazon Linux	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	<input type="button" value="Select"/>
<input type="checkbox"/> AWS Marketplace	 <b>Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-8b8c57f8</b>	Red Hat	Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type	<input type="button" value="Select"/>
<input type="checkbox"/> Community AMIs	 <b>SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-f4278487</b>	SUSE Linux	SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	<input type="button" value="Select"/>
<input type="checkbox"/> Free tier only ⓘ	 <b>Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-47a23a30</b>	Ubuntu	Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	<input type="button" value="Select"/>
	 <b>Microsoft Windows Server 2012 R2 Base - ami-7943ec0a</b>	Windows	Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]	<input type="button" value="Select"/>

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".

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

**Step 2: Choose an Instance Type**

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

Filter by:

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	<b>t2.micro</b> Free tier eligible	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
<input type="checkbox"/>	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
<input type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate

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.

The screenshot shows the 'Step 3: Configure Instance Details' page of the AWS EC2 wizard. The top navigation bar includes links for 'Choose AMI', 'Choose Instance Type', 'Configure Instance' (which is active), 'Add Storage', 'Tag Instance', 'Configure Security Group', and 'Review'. The main section is titled 'Step 3: Configure Instance Details' and contains the following configuration fields:

- Number of instances:** 1 (input field) | **Launch into Auto Scaling Group:** (button)
- Purchasing option:** Request Spot instances (checkbox)
- Network:** vpc-fd83a398 (172.31.0.0/16) (default) | **Create new VPC:** (button)
- Subnet:** No preference (default subnet in any Availability Zone) | **Create new subnet:** (button)
- Auto-assign Public IP:** Use subnet setting (Enable) | **Create new IAM role:** (button)
- IAM role:** None | **Create new IAM role:** (button)
- Shutdown behavior:** Stop | **Protect against accidental termination:** (checkbox)
- Enable termination protection:** (checkbox)
- Monitoring:** Enable CloudWatch detailed monitoring | **Additional charges apply:** (checkbox)
- Tenancy:** Shared - Run a shared hardware instance | **Additional charges will apply for dedicated tenancy:** (checkbox)

At the bottom left is a link to 'Advanced Details'. At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch' (highlighted in blue), and 'Next: Add Storage'.

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.

## Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-b83e5171	8	General Purpose SSD (GP2)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

**Add New Volume**

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.

**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.

## Step 6: Configure Security Group

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

Assign a security group:  Create a new security group

Select an existing security group

Security group name: launch-wizard-1

Description: launch-wizard-1 created 2016-01-19T09:55:13.003+00:00

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere <input checked="" type="checkbox"/> 0.0.0.0/0 <input type="checkbox"/>

**Add Rule**



**Warning**

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

**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.

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

### Step 7: Review Instance Launch

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

**AMI Details** [Edit AMI](#)

**Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-bff32ccc**

**Free tier eligible** The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

**Instance Type** [Edit instance type](#)

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

**Security Groups** [Edit security groups](#)

Security group name: Web-DMZ  
Description: Web-DMZ

Type <a href="#">i</a>	Protocol <a href="#">i</a>	Port Range <a href="#">i</a>	Source <a href="#">i</a>
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0

**Instance Details** [Edit instance details](#)

**Storage** [Edit storage](#)

**Tags** [Edit tags](#)

[Cancel](#) [Previous](#) [Launch](#)

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.

Select an existing key pair or create a new key pair [X](#)

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

[Key pair name](#) MyEC2Key [Download Key Pair](#)

**... You have to download the private key file (\*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.**

[Cancel](#) [Launch Instances](#)

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 -

## Launch Status

### Your instances are now launching

The following instance launches have been initiated: i-8d575206 [View launch log](#)

### Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

#### How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can connect to them from the Instances screen. [Find out](#) how to connect to your instances.

#### Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

[Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)

[Create and attach additional EBS volumes](#) (Additional charges may apply)

[Manage security groups](#)

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 -

Instances										
Actions		Instance Details								
Filter by tags and attributes or search by keyword										
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	
WebServer	i-8d575206	t2.micro	eu-west-1a	<span>running</span>	<span>Initializing</span>	<span>None</span>	ec2-52-48-78-87.eu-we...	52.48.78.87	MyEC2Key	

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 - Lab

### Lab Summary

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.

## 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.

The screenshot shows the AWS EC2 Dashboard with the 'Security Groups' section selected. On the left, there's a sidebar with various navigation links like Instances, AMIs, and Auto Scaling. The main area displays a table of security groups with columns for Name, Group ID, Group Name, VPC ID, and Description. Two entries are listed: 'sg-4de8ae29' (Web-DMZ) and 'sg-a6f58cc2' (default). Below this, a detailed view for the 'sg-4de8ae29' group is shown, specifically the 'Inbound' tab of its rule editor. The table lists two rules: one for HTTP (TCP port 80) and one for SSH (TCP port 22), both with 'Anywhere' as the source.

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,

This screenshot shows the 'Edit inbound rules' dialog box. It contains a table with columns for Type, Protocol, Port Range, and Source. There are two entries: one for HTTP (TCP port 80, Anywhere source) and one for SSH (TCP port 22, Anywhere source). At the bottom, there are 'Add Rule', 'Cancel', and 'Save' buttons.

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 -

The screenshot shows a summary slide with the title 'Security Group Lab'. It contains a list of bullet points:

- All Inbound Traffic is Blocked
- All Outbound Traffic is Allowed
- Changes to Security Groups take effect immediately
- You can have any number of EC2 instances within a security group.
- Security Groups are **STATEFUL**.
- If you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again.

## EC2 - Volume vs Snapshots

Monday, December 26, 2016 12:36 PM

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

- 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.
- If this is your first snapshot, it may take some time to create.

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 -

The screenshot shows the AWS EC2 Dashboard with the 'Volumes' section selected. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, AMIs, and Elastic Block Store. Under 'Elastic Block Store', 'Volumes' is highlighted. The main area shows a table of volumes:

Name	Volume ID	Size	Type	Snapshot	Created	Availability Zone	State	Alarm Status	A
vol-fd1431ed	8 GiB	gp2	snap-3efdf3c4	December 17, 2014	eu-west-1a	in-use	None		

Below the table, a detailed view for 'vol-fd1431ed' is shown. The 'Description' tab is selected, displaying the following information:

Volume ID	vol-fd1431ed
Size	8 GiB
Created	December 17, 2014 10:15:11 AM UTC
State	In-use
Attachment information	i-58658bbf (Webserver1) :/dev/xvda (attached)
Alarm status	None
Snapshot	snap-3efdf3c4
Availability Zone	eu-west-1a
Encrypted	Not Encrypted
KMS Key ID	

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.

The screenshot shows the 'Create Volume' dialog box. It has the following fields:

- Type: Magnetic
- Size (GiB): 10 (Min: 1GiB, Max: 1024GiB)
- IOPS: (empty input field)
- Availability Zone: eu-west-1a
- Snapshot ID: Search (case-insensitive) (empty input field)
- Encryption:  Encrypt this volume

At the bottom are 'Cancel' and 'Create' buttons.

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 -

EC2 Dashboard

- Events
- Tags
- Reports
- Limits

**INSTANCES**

- Instances
- Spot Requests
- Reserved Instances

**IMAGES**

- AMIs
- Bundle Tasks

**ELASTIC BLOCK STORE**

- Volumes**
- Snapshots

**NETWORK & SECURITY**

- Security Groups
- Elastic IPs
- Placement Groups
- Load Balancers
- Key Pairs
- Network Interfaces

**AUTO SCALING**

- Launch Configurations
- Auto Scaling Groups

**Create Volume** Actions ▾

Name	Volume ID	Size	Volume Type	Snapshot	Created	Availability Zone	State	Alarm Status	Actions
vol-a5f7d3b5	vol-a5f7d3b5	10 GiB	standard		December 18, 2014	eu-west-1a	available	None	...
systempartitionec2	vol-fd1431ed	8 GiB	gp2	snap-3efdf3c4	December 17, 2014	eu-west-1a	in-use	None	...

**Volumes: vol-a5f7d3b5**

Description	Status Checks	Monitoring	Tags
Volume ID: vol-a5f7d3b5	Size: 10 GiB	Created: December 18, 2014 7:06:03 PM UTC	State: available
Attachment information		Volume type: standard	Alarm status: None
			Snapshot: -
			Availability Zone: eu-west-1a
			Encrypted: Not Encrypted
			KMS Key ID:
			KMS Key Aliases:

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 -

**Attach Volume**

Volume	vol-a5f7d3b5 in eu-west-1a
Instance	i-58658bbf in eu-west-1a
Device	/dev/sdf
Linux Devices: /dev/sdf through /dev/sdp <small>Note: Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.</small>	

Cancel Attach

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 hit 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.

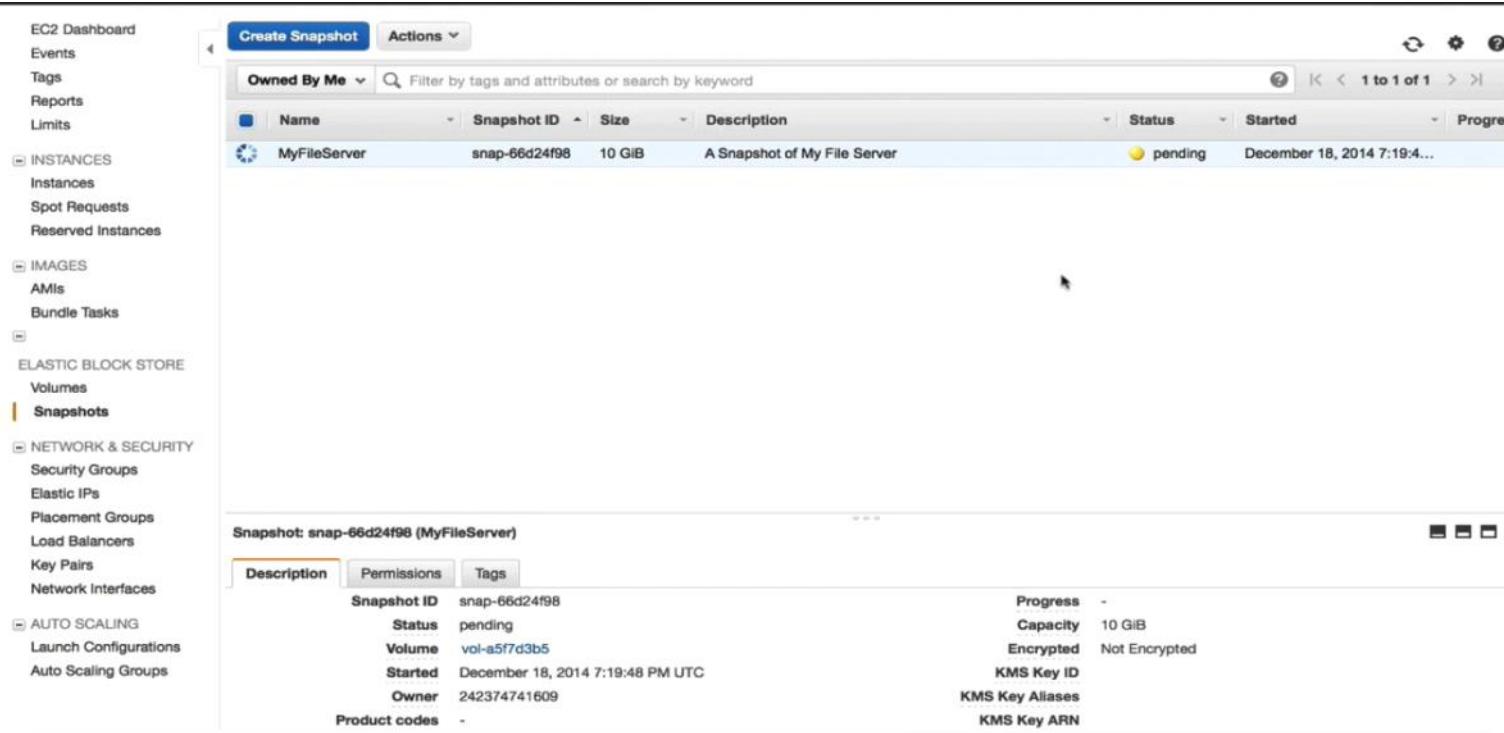
Now we can create snapshot of this volume -

**Create Snapshot**

Volume	vol-a5f7d3b5
Name	MyFileServer
Description	A Snapshot of My File Server
Encrypted	No

Cancel Create

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.



The screenshot shows the AWS EC2 Dashboard with the 'Schemas' section selected. A table lists a single snapshot:

Name	Snapshot ID	Size	Description	Status	Started	Progress
MyFileServer	snap-66d24f98	10 GiB	A Snapshot of My File Server	pending	December 18, 2014 7:19:4...	-

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.



**Create Volume**

Snapshot ID: snap-66d24f98 (MyFileServer)

Type: General Purpose (SSD)

Size (GiB): 10 (Min: 10GiB, Max: 1024GiB)

IOPS: 30 / 3000 (3000 IOPS bursts and baseline of 3 IOPS per GB)

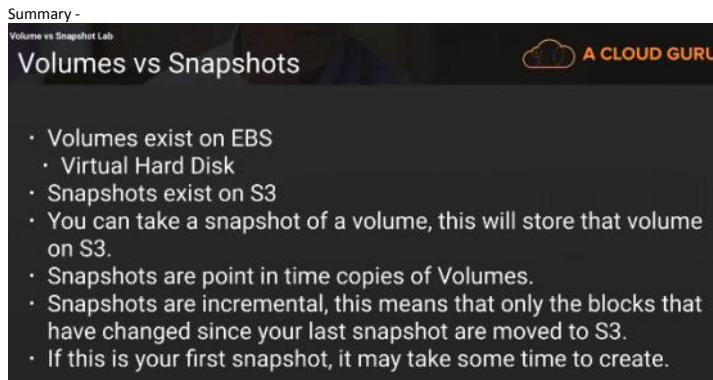
Availability Zone: eu-west-1a

Encryption: Not Encrypted

**Create**

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**

**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.
- If this is your first snapshot, it may take some time to create.

## Volumes vs Snapshots - Security



- 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

## Snapshots of Root Device Volumes



- 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 -



A Cloud Guru

RAID, Volumes & Snapshots

- RAID = Redundant Array of Independent Disks
- RAID 0 - Striped, No Redundancy, Good Performance
- RAID 1 - Mirrored, Redundancy
- RAID 5 - Good for reads, bad for writes, AWS does not recommend ever putting RAID 5's on EBS
- RAID 10 - Striped & Mirrored, Good Redundancy, Good Performance.

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.

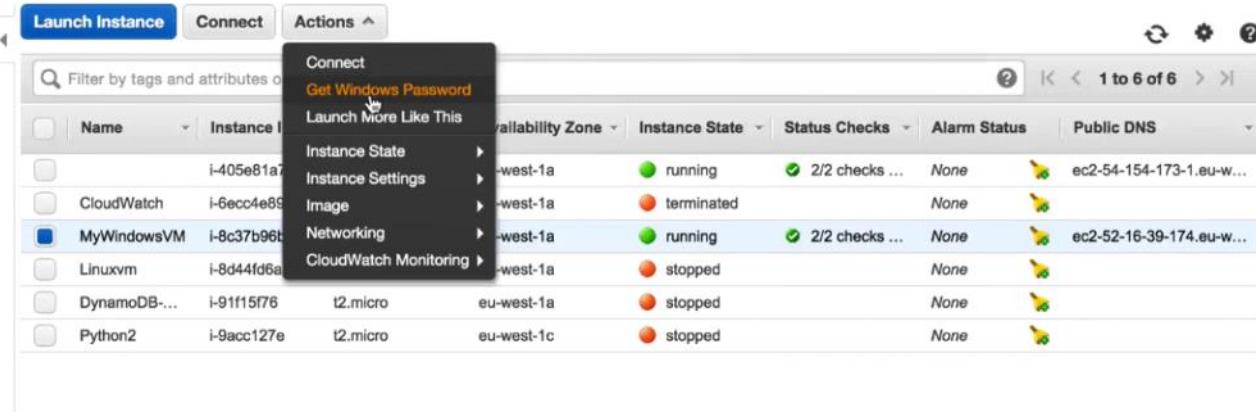
Now we will use windows EC2 machine, for that go to main EC2 console and from here we will once again create a new instance with windows OS.

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".



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Images, and Auto Scaling. The Instances section is currently selected. The main area displays a table of instances. One instance, 'MyWindowsVM', is selected and has a context menu open over it. The menu items are 'Connect', 'Get Windows Password' (which is highlighted in orange), and 'Launch More Like This'. Below the table, there's a detailed view for the selected instance ('i-8c37b96b'). It shows the instance ID, state (running), type (t2.micro), and network details (Public DNS: ec2-52-16-39-174.eu-west-1.compute.amazonaws.com). There are tabs for Description, Status Checks, Monitoring, and Tags.

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

## Retrieve Default Windows Administrator Password

X

To access this instance remotely (e.g. Remote Desktop Connection), you will need your Windows Administrator password. A default password was created when the instance was launched and is available encrypted in the system log.

To decrypt your password, you will need your key pair for this instance. Browse to your key pair, or copy and paste the contents of your private key file into the text area below, then click Decrypt Password.

The following Key Pair was associated with this instance when it was created.

**Key Name** MyWindowsKey

In order to retrieve your password you will need to specify the path of this Key Pair on your local machine:

**Key Pair Path**  No file chosen

Or you can copy and paste the contents of the Key Pair below:

Paste contents of private key file here

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 -

## Retrieve Default Windows Administrator Password

X



### Password Decryption Successful

The password for instance i-8c37b96b (MyWindowsVM) was successfully decrypted.



### Password change recommended

We recommend that you change your default password. Note: If a default password is changed, it cannot be retrieved through this tool. It's important that you change your password to one that you will remember.

You can connect remotely using this information:

**Public IP** 52.16.39.174

**User name** Administrator

**Password** Ax)Dxur:&

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-

File Action View Help

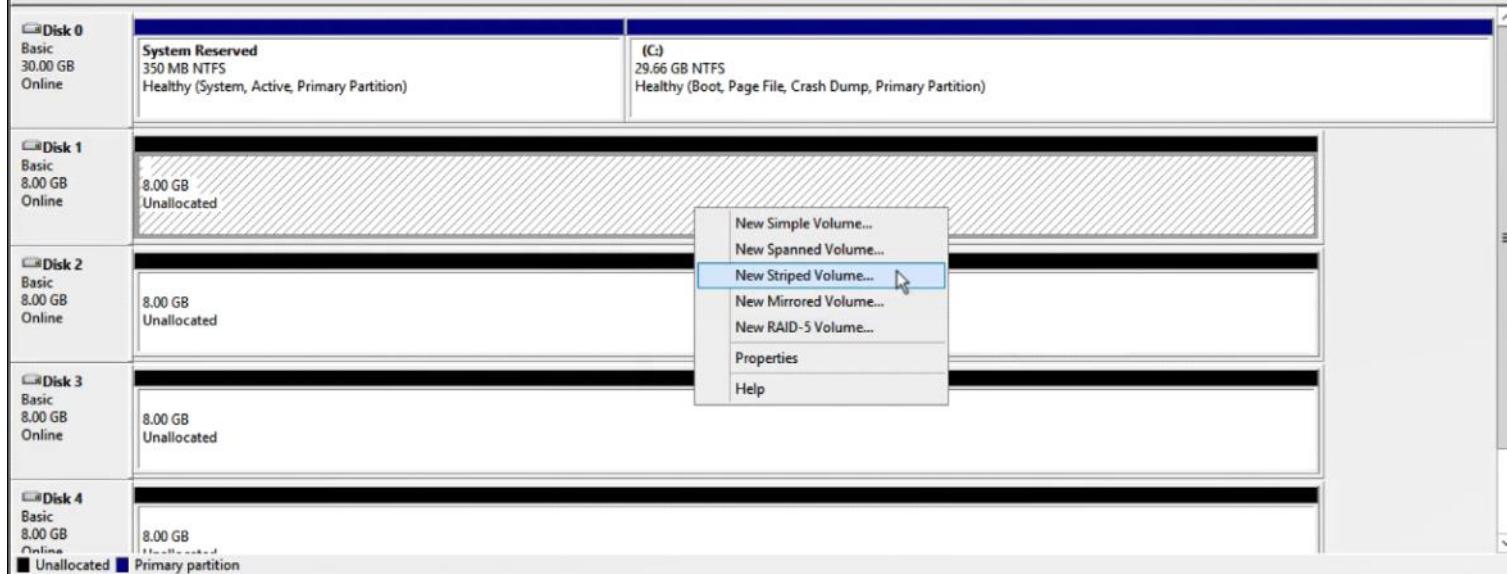
Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free
(C:)	Simple	Basic	NTFS	Healthy (B...)	29.66 GB	6.91 GB	23 %
(D:)	Simple	Basic	NTFS	Healthy (P...)	8.00 GB	7.96 GB	100 %
(E:)	Simple	Basic	NTFS	Healthy (P...)	8.00 GB	7.96 GB	100 %
(F:)	Simple	Basic	NTFS	Healthy (P...)	8.00 GB	7.96 GB	100 %
(G:)	Simple	Basic	NTFS	Healthy (P...)	8.00 GB	7.96 GB	100 %
System Reserved	Simple	Basic	NTFS	Healthy (S...)	350 MB	88 MB	25 %

Disk 0 Basic 30.00 GB Online	<b>System Reserved</b> 350 MB NTFS Healthy (System, Active, Primary Partition)	(C:) 29.66 GB NTFS Healthy (Boot, Page File, Crash Dump, Primary Partition)
Disk 1 Basic 8.00 GB Online	(D:) 8.00 GB NTFS Healthy (Primary Partition)	
Disk 2 Basic 8.00 GB Online	(E:) 8.00 GB NTFS Healthy (Primary Partition)	
Disk 3 Basic 8.00 GB Online	(F:) 8.00 GB NTFS Healthy (Primary Partition)	
Disk 4 Basic 8.00 GB Online	(G:) 8.00 GB NTFS Primary partition	
Unallocated		

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-

Volume	Layout	Type	File System	Status	Capacity	Free Spa...	% Free
(C:)	Simple	Basic	NTFS	Healthy (B...)	29.66 GB	6.91 GB	23 %
New Volume (D:)	Striped	Dynamic	NTFS	Healthy	31.99 GB	31.91 GB	100 %
System Reserved	Simple	Basic	NTFS	Healthy (S...)	350 MB	88 MB	25 %

Disk 0 Basic 30.00 GB Online	System Reserved 350 MB NTFS Healthy (System, Active, Primary Partition)	(C) 29.66 GB NTFS Healthy (Boot, Page File, Crash Dump, Primary Partition)
Disk 1 Dynamic 8.00 GB Online	New Volume (D:) 8.00 GB NTFS Healthy	
Disk 2 Dynamic 8.00 GB Online	New Volume (D:) 8.00 GB NTFS Healthy	
Disk 3 Dynamic 8.00 GB Online	New Volume (D:) 8.00 GB NTFS Healthy	
Disk 4 Dynamic 8.00 GB Online	New Volume (D:) 8.00 GB NTFS Healthy	
Unallocated	Primary partition	Striped volume

Here we can get high I/O from this.

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

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.

## EC2 - Instance Store vs EBS

Monday, December 26, 2016 3:32 PM

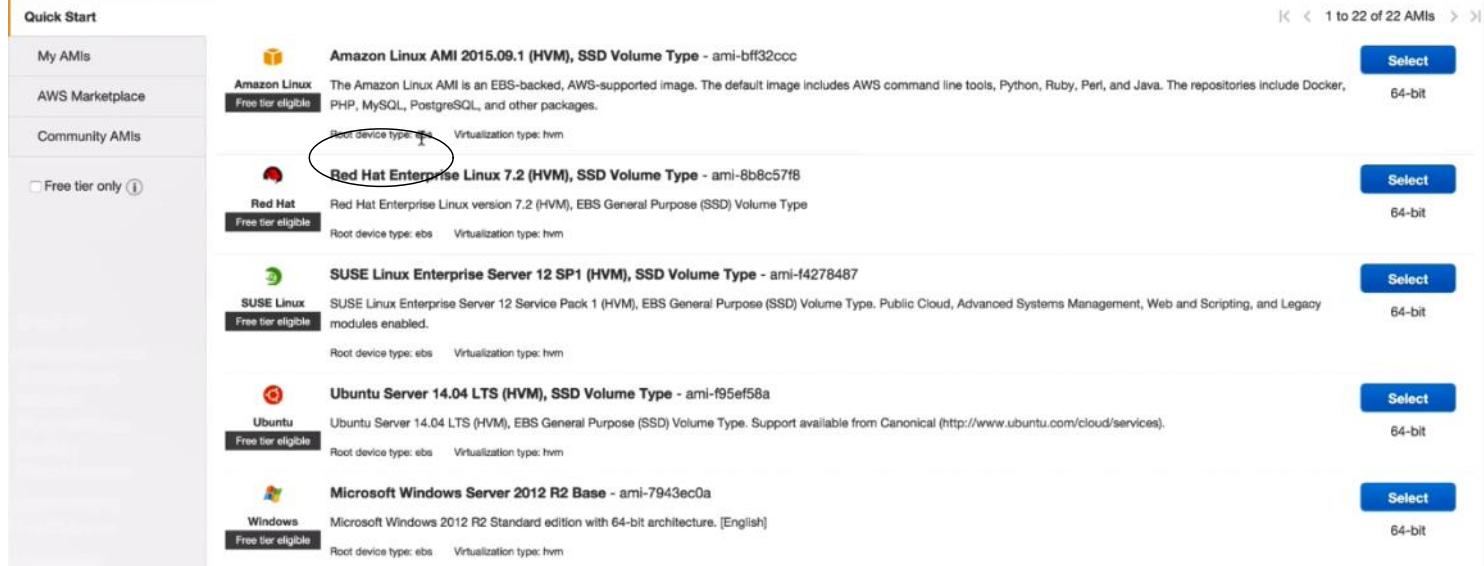
You can select your AMI based on:



A CLOUD GURU

- Region (see Regions and Availability Zones)
- Operating system
- Architecture (32-bit or 64-bit)
- Launch Permissions
- Storage for the Root Device (Root Device Volume)
  - Instance Store (EPHEMERAL STORAGE)
  - EBS Backed Volumes

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 see that it also shows what type of root device you are using-



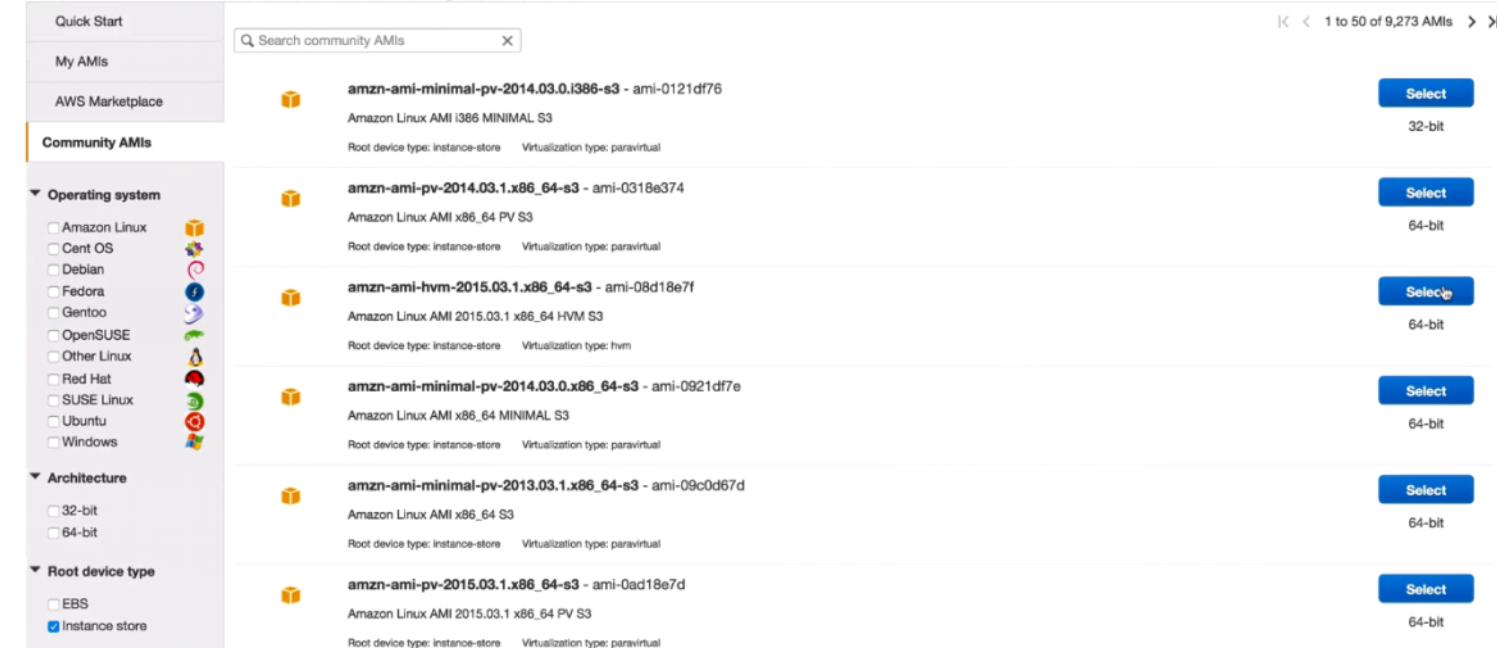
The screenshot shows the AWS Lambda console with the 'Community AMIs' section selected. It lists several AMIs with their names, descriptions, root device types, virtualization types, and 'Select' buttons.

AMI Name	Description	Root device type	Virtualization type	Action
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-bff32ccc	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	ebs	hvm	Select
Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-8b8c57f8	Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type	ebs	hvm	Select
SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-f4278487	SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	ebs	hvm	Select
Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-f95ef58a	Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	ebs	hvm	Select
Microsoft Windows Server 2012 R2 Base - ami-7943ec0a	Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]	ebs	hvm	Select

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

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

[Cancel and Exit](#)



The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' step. On the left, there is a sidebar with filters for 'Operating system', 'Architecture', and 'Root device type'. The 'Root device type' filter has 'Instance store' selected. The main area lists several AMIs with their names, descriptions, root device types, virtualization types, and 'Select' buttons.

AMI Name	Description	Root device type	Virtualization type	Action
amzn-ami-minimal-pv-2014.03.0.i386-s3 - ami-0121df76	Amazon Linux AMI i386 MINIMAL S3	instance-store	paravirtual	Select
amzn-ami-pv-2014.03.1.x86_64-s3 - ami-0318e374	Amazon Linux AMI x86_64 PV S3	instance-store	paravirtual	Select
amzn-ami-hvm-2015.03.1.x86_64-s3 - ami-08d18e7f	Amazon Linux AMI 2015.03.1 x86_64 HVM S3	instance-store	hvm	Select
amzn-ami-minimal-pv-2014.03.0.x86_64-s3 - ami-0921df7e	Amazon Linux AMI x86_64 MINIMAL S3	instance-store	paravirtual	Select
amzn-ami-minimal-pv-2013.03.1.x86_64-s3 - ami-09c0d67d	Amazon Linux AMI x86_64 S3	instance-store	paravirtual	Select
amzn-ami-pv-2015.03.1.x86_64-s3 - ami-0ad18e7d	Amazon Linux AMI 2015.03.1 x86_64 PV S3	instance-store	paravirtual	Select

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

<input checked="" type="checkbox"/>	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
<input checked="" type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
<input type="checkbox"/>	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High
<input checked="" type="checkbox"/>	Compute optimized	c4.large	2	3.75	EBS only	Yes	Moderate
<input checked="" type="checkbox"/>	Compute optimized	c4.xlarge	4	7.5	EBS only	Yes	High
<input checked="" type="checkbox"/>	Compute optimized	c4.2xlarge	8	15	EBS only	Yes	High
<input checked="" type="checkbox"/>	Compute optimized	c4.4xlarge	16	30	EBS only	Yes	High
<input checked="" type="checkbox"/>	Compute optimized	c4.8xlarge	36	60	EBS only	Yes	10 Gigabit
<input type="checkbox"/>	Compute optimized	c3.large	2	3.75	2 x 16 (SSD)	-	Moderate
<input type="checkbox"/>	Compute optimized	c3.xlarge	4	7.5	2 x 40 (SSD)	Yes	Moderate
<input type="checkbox"/>	Compute optimized	c3.2xlarge	8	15	2 x 80 (SSD)	Yes	High
<input type="checkbox"/>	Compute optimized	c3.4xlarge	16	30	2 x 160 (SSD)	Yes	High
<input type="checkbox"/>	Compute optimized	c3.8xlarge	32	60	2 x 320 (SSD)	-	10 Gigabit
<input type="checkbox"/>	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.

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.

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

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

- instance store** (i-29d9d4a2): Status: running, Public DNS: ec2-52-49-20-25.eu-west-1.compute.amazonaws.com, Public IP: 52.49.20.25, Key Name: MyEC2Key
- EBS** (i-00d8d58b): Status: running, Public DNS: ec2-52-31-198-229.eu-west-1.compute.amazonaws.com, Public IP: 52.31.198.229, Key Name: MyEC2Key

A context menu is open over the 'instance store' instance, with the 'Stop' option highlighted.

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

The screenshot shows the AWS EC2 Instances page with the same two instances. The context menu over the 'instance store' instance now has 'Reboot' highlighted.

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 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 -

**EBS vs Instance Store**

All AMIs are categorized as either backed by Amazon EBS or backed by instance store.

**For EBS Volumes:** The root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.

**For Instance Store Volumes:** The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

#### Exam Tips -

**EBS vs Instance Store - Exam Tips**

- 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.

## EC2 - Create an AMI

Monday, December 26, 2016 4:09 PM

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

- A template for the root volume for the instance (for example, an operating system, an application server, and applications)
- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it's launched

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.

Ryan Kroonenburg - Ireland - Support -

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
Webserver1	i-58658bbf	t2.micro	eu-west-1a	running	2/2 checks...	None	ec2-54-154-110-158...	54.154.110.158

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

Here is our snapshot -

Create Snapshot Actions

- Delete
- Create Volume
- Create Image
- Copy
- Modify Snapshot Permissions
- Add/Edit Tags

Name	Description	Status	Started	Progress
MyDefaultWeb	This is the default web server	completed	December 19, 2014 10:25...	available

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 -

**Create Image from EBS Snapshot**

Name	MyDefaultWebserverAMI	Description	My Default Webserver				
Architecture	x86_64	Virtualization type	Paravirtual				
Root device name	/dev/sda1	Kernel ID	Use default				
RAM disk ID	Use default						
<b>Block Device Mappings</b>							
Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-a458c25a	8	General Purpose (SSD)	24 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
<a href="#">Add New Volume</a>							
							<a href="#">Cancel</a> <a href="#">Create</a>

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".

**Modify Image Permissions**

This image is currently:  Public  Private

AWS Account Number

This image currently has no permissions

AWS Account Number  [Add Permission](#)

Add "create volume" permissions to the following associated snapshots when creating permissions:  
 snap-a458c25a

[Cancel](#) [Save](#)

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 -**  
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.

## 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 -

1. Define Load Balancer    2. Assign Security Groups    3. Configure Security Settings    4. Configure Health Check    5. Add EC2 Instances    6. Add Tags    7. Review

### Step 1: Define Load Balancer

#### Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name: MyWebDMZ

Create LB Inside: My Default VPC (172.31.0.0/16)

Create an internal load balancer:

Enable advanced VPC configuration:

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

Here the steps are self-explanatory.

Then we will assign security groups-

1. Define Load Balancer    2. Assign Security Groups    3. Configure Security Settings    4. Configure Health Check    5. Add EC2 Instances    6. Add Tags    7. Review

### Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group:  Create a new security group  
 Select an existing security group

Filter VPC security groups

Security Group ID	Name	Description	Actions
sg-a6f58cc2	default	default VPC security group	<a href="#">Copy to new</a>
sg-4de8ae29	Web-DMZ	Web-DMZ	<a href="#">Copy to new</a>

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.

## Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol	HTTP
Ping Port	80
Ping Path	/index.html

### Advanced Details

Response Timeout	5	seconds
Health Check Interval	30	seconds
Unhealthy Threshold	2	
Healthy Threshold	10	

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 our EC2 instance,

## Step 5: Add EC2 Instances

The table below lists all your running EC2 Instances. Check the boxes in the Select column to add those instances to this load balancer.

VPC vpc-fd83a398 (172.31.0.0/16)

Instance	Name	State	Security Groups	Zone	Subnet ID	Subnet CIDR
i-8d575206	WebServer	running	Web-DMZ	eu-west-1a	subnet-6490a213	172.31.16.0/20

### Availability Zone Distribution

1 Instance in eu-west-1a

- Enable Cross-Zone Load Balancing
- Enable Connection Draining 300 seconds

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.

Create Load Balancer Actions

Filter: Search Load Balancers

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Check	Created At
MyWebDMZ	MyWebDMZ-280429352.eu...	80 (HTTP) forwarding to 80 (...	eu-west-1b, eu-west-1c...	1 Instance	HTTP:80/healthcheck.html	January 19, 2016 at 3:49:41 ...

Load balancer: MyWebDMZ

Instances

Connection Draining: Enabled, 300 seconds (Edit)

Edit Instances

Instance ID	Name	Availability Zone	Status	Actions
i-8d575206	WebServer	eu-west-1a	InService	Remove from Load Balancer

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

Create Load Balancer Actions ▾

Filter:  X

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Check	Created At
MyWebDMZ	MyWebDMZ-260429352.eu...	80 (HTTP) forwarding to 80 (...)	eu-west-1b, eu-west-1c...	1 Instance	HTTP:80/healthcheck.html	January 19, 2016 at 3:49:41 ...

Load balancer: MyWebDMZ

Description Instances Health Check Monitoring Security Listeners Tags

**DNS Name:** MyWebDMZ-260429352.eu-west-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an "A" record with any specific IP address. If you want to use a friendly DNS name for your load balancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see Using Domain Names With Elastic Load Balancing.

**Scheme:** Internet-facing

**Status:** 1 of 1 instances in service

**Port Configuration:** 80 (HTTP) forwarding to 80 (HTTP)  
Stickiness: Disabled ([Edit](#))

**Availability Zones:** subnet-13df954a - eu-west-1b,  
subnet-3a70656f - eu-west-1c,  
subnet-6490a213 - eu-west-1a

**Cross-Zone Load Balancing:** Enabled ([Edit](#))

**Source Security Group:** 566216698943/Web-DMZ  
Owner Alias: 566216698943  
Group Name: Web-DMZ

**Hosted Zone ID:** Z3NF1Z3NOM5OY2

**VPC ID:** vpc-fd83a398

**Access Logs:** Disabled ([Edit](#))

**Connection Settings:** Idle Timeout: 60 seconds ([Edit](#))

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 -

Elastic Load Balancers  A CLOUD GURU

- In Service or Out of Service
- Health Checks
- Have their own DNS name. You are never given an IP address.

## 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.

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

### Step 3: Configure Instance Details

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

Number of instances 1 Launch into Auto Scaling Group

Purchasing option Request Spot Instances

Network vpc-fd83a398 (172.31.0.0/16) (default) Create new VPC

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

Auto-assign Public IP Use subnet setting (Enable)

IAM role None Create new IAM role

Shutdown behavior Stop

Enable termination protection Protect against accidental termination

Monitoring Enable CloudWatch detailed monitoring Additional charges apply.

Tenancy Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.

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.

## Amazon CloudWatch Pricing

### Free Tier

You can get started with Amazon CloudWatch for free. Many applications should be able to operate within these free tier limits.

- New and existing customers also receive 3 dashboards of up to 50 metrics each per month at no additional charge
- Basic Monitoring metrics (at five-minute frequency) for Amazon EC2 instances are free of charge, as are all metrics for Amazon EBS volumes, Elastic Load Balancers, and Amazon RDS DB instances.
- New and existing customers also receive 10 metrics (applicable to Detailed Monitoring for Amazon EC2 instances, Custom Metrics, or CloudWatch Logs), 10 alarms, and 1 million API requests each month at no additional charge.
- New and existing customers also receive 5 GB of data ingestion and 5 GB of archived storage per month at no additional charge.

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.

Region: US East (N. Virginia)

### Amazon CloudWatch Metrics

- \$3.00 per dashboard per month

### Detailed Monitoring for Amazon EC2 Instances

- \$3.50 per instance per month for Detailed Monitoring at 1-minute frequency

### Amazon CloudWatch Custom Metrics

- \$0.50 per metric per month

### Amazon CloudWatch Alarms

- \$0.10 per alarm per month

### Amazon CloudWatch API Requests

- \$0.01 per 1,000 GetMetricStatistics, ListMetrics, or PutMetricData requests

### Amazon CloudWatch Logs\*

- \$0.50 per GB ingested\*\*
- \$0.03 per GB archived per month\*\*\*
- Data Transfer OUT from CloudWatch Logs is priced equivalent to the "Data Transfer OUT from Amazon EC2 To" and "Data Transfer OUT from Amazon EC2 to Internet" tables on the [EC2 Pricing Page](#).

### Amazon CloudWatch Events - Custom Events\*\*\*\*

- \$1.00 per million custom events generated\*\*\*\*\*

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 -

Browse Metrics  X

## CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: 75

EBS Metrics : 20	EC2 Metrics : 12	ELB Metrics : 40
Per-Volume Metrics : 20	Per-Instance Metrics : 12	Per-LB Metrics : 6 Per LB, per AZ Metrics : 8 By Availability Zone : 8 Across All LBs : 6 By Namespace : 6 By Service : 6
S3 Metrics : 3		
Storage Metrics : 3		

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

EC2  X 1 to 12 of 12 Metrics < >

Showing all results (12) for EC2 > Per-Instance Metrics. For more results expand your search to All EC2 Metrics.

Select All | Clear

**EC2 > Per-Instance Metrics**

InstanceId	InstanceName	Metric Name
<input type="checkbox"/> i-8d575206	WebServer	CPUCreditBalance
<input type="checkbox"/> i-8d575206	WebServer	CPUCreditUsage
<input checked="" type="checkbox"/> i-8d575206	WebServer	CPUUtilization
<input type="checkbox"/> i-8d575206	WebServer	DiskReadBytes
<input type="checkbox"/> i-8d575206	WebServer	DiskReadOps
<input type="checkbox"/> i-8d575206	WebServer	DiskWriteBytes
<input type="checkbox"/> i-8d575206	WebServer	DiskWriteOps
<input type="checkbox"/> i-8d575206	WebServer	NetworkIn
<input type="checkbox"/> i-8d575206	WebServer	NetworkOut
<input type="checkbox"/> i-8d575206	WebServer	StatusCheckFailed
<input type="checkbox"/> i-8d575206	WebServer	StatusCheckFailed_Instance
<input type="checkbox"/> i-8d575206	WebServer	StatusCheckFailed_System

**Title: CPUUtilization (Percent)** Average 5 Minutes

Value: 9.466 (Percent)  
Time: 2016/01/19 10:15 UTC  
Metric: CPUUtilization  
Namespace: AWS/E2  
InstanceId: i-8d575206 (WebServer)  
Y-Axis: Left [switch]

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 -

CloudWatch Dashboards NEW

My-Overview Actions Add widget Save dashboard 1 Days C

CPUUtilization

Time range: Minutes > Hours > Days > Period: Auto (5m) > Time zone: UTC >

Selected Metrics: EBS, EC2, ELB, S3

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 -

CloudWatch  
 Dashboards NEW  
 • My-Overview  
 Alarms  
 ALARM  
 INSUFFICIENT  
 OK  
 Billing  
**Events NEW**  
 Rules  
 Logs  
 Metrics  
 Selected Metrics  
 EBS  
 EC2  
 ELB  
 S3

**Welcome!**

We are excited for you to try Amazon CloudWatch Events, you may [email us](#) directly with feedback, or use the Feedback button at the bottom of the page.

## Welcome to CloudWatch Events

CloudWatch Events helps you to respond to state changes in your AWS resources. When your resources change state they automatically send events into an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a pre-determined schedule. For example, you can configure rules to:

- Automatically invoke an AWS Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the Running state
- Direct specific API records from CloudTrail to a Kinesis stream for detailed analysis of potential security or availability risks
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume

[Create rule](#)

### Start Responding to CloudWatch Events

Determine events of interest in the CloudWatch Events stream

Create rules to select events of interest

Specify actions to take when a rule matches an event

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

**When we go to Logs -**

CloudWatch  
 Dashboards NEW  
 • My-Overview  
 Alarms  
 ALARM  
 INSUFFICIENT  
 OK  
 Billing  
**Events NEW**  
 Rules  
**Logs**  
 Metrics  
 Selected Metrics  
 EBS  
 EC2  
 ELB  
 S3

**Welcome to CloudWatch Logs**

CloudWatch Logs helps you to aggregate, monitor, and store logs. For example, you can:

- Monitor HTTP response codes in Apache logs
- Receive alarms for errors in kernel logs
- Count exceptions in application logs

To start sending your logs to CloudWatch, click the Quick Start Guide and follow the instructions. To explore CloudWatch Logs before sending any data, click "Create Log Group" to create your first Log Group.

[Quick Start Guide](#) [Create log group](#)

### Start Sending Log Data to CloudWatch

Install the Agent

Install and configure the CloudWatch Logs agent to send your logs to the CloudWatch Logs service.

Monitor

Create metric filters to automatically monitor the logs sent to CloudWatch Logs.

Access

View the log data you have sent and stored in CloudWatch Logs.

**Agent Installation Options**

- Install on an EC2 instance
- Install using CloudFormation
- Install using Chef

**Additional Info**

- Documentation
- All CloudWatch Resources
- Forums

**Now when we go to Alarms -**

**Create Alarm**   [Modify](#)   [Copy](#)   [Delete](#)

Filter: [All alarms](#) [Search Alarms](#)

State	Name	Threshold	Config Status
No records found.			

0 Alarms selected

Select an alarm above

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

**Create Alarm**

[1. Select Metric](#)   [2. Define Alarm](#)

[Browse Metrics](#)   [Search Metrics](#)

### CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: 75

EBS Metrics: 20	EC2 Metrics: 12	ELB Metrics: 40
Per-Volume Metrics: 20	Per-Instance Metrics: 12	Per-LB Metrics: 6
		Per LB, per AZ Metrics: 8
		By Availability Zone: 8
		Across All LBs: 6
		By Namespace: 6
		By Service: 6

S3 Metrics: 3

Storage Metrics: 3

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

**1. Select Metric**   **2. Define Alarm**

EC2  1 to 12 of 12 Metrics

Per-Instance Metrics

EC2 > Per-Instance Metrics

InstanceId	InstanceName	Metric Name
i-8d575206	WebServer	CPUCreditBalance
i-8d575206	WebServer	CPUCreditUsage
<input checked="" type="checkbox"/> i-8d575206	WebServer	CPUUtilization
i-8d575206	WebServer	DiskReadBytes
i-8d575206	WebServer	DiskReadOps
i-8d575206	WebServer	DiskWriteBytes
i-8d575206	WebServer	DiskWriteOps
i-8d575206	WebServer	NetworkIn
i-8d575206	WebServer	NetworkOut
i-8d575206	WebServer	StatusCheckFailed
i-8d575206	WebServer	StatusCheckFailed_Instance
i-8d575206	WebServer	StatusCheckFailed_System

**Title:** CPUUtilization (Percent) **Average** **5 Minutes**

Left axis units: Percent

**Time Range**  
Relative  Absolute  UTC (GMT)  
From: 12 hours ago   
To: 0 minutes ago   
Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

Now hit next and go to next window -

**Create Alarm**

**1. Select Metric**   **2. Define Alarm**

### Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name:

Description:

Whenever: CPUUtilization  
is:  $\geq 0$     
for: 1 consecutive period(s)

### Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM

Send notification to: Select a notification list

### Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: AWS/EC2  
InstanceId: i-8d575206  
InstanceName: WebServer  
Metric Name: CPUUtilization  
Period: 5 Minutes    
Statistic: Average

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 -

**1. Select Metric** **2. Define Alarm**

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

**Name:** CPU Utilization > 80%  
**Description:** CPU Utilization > 80%

**Whenever:** CPUUtilization  
**is:**  $\leq t$  5  
**for:** 1 consecutive period(s)

**Actions**  
Define what actions are taken when your alarm changes state.

**EC2 Action** Delete

**Whenever this alarm:** State is ALARM

**Take this action:**
 Recover this instance Recover this instance from termination protection  
 Stop this instance Stop this instance from termination protection  
 Terminate this instance Terminate this instance from termination protection  
 Reboot this instance Reboot this instance from termination protection

This will terminate your EC2 instance (i-8d575206). You will not be able to terminate this instance if termination protection is enabled.

AWS will create the following IAM role in your account so that AWS can perform this action. [Learn more](#).

Create IAM role: **EC2ActionsAccess** (show IAM policy document)

[+ Notification](#) [+ AutoScaling Action](#) [+ EC2 Action](#)

[Cancel](#) [Previous](#) [Next](#) **Create Alarm**

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-Scaling group we will cover that later.

**Summary -**

CloudWatch Lab A CLOUD GURU

### What can I do with Cloudwatch?

- 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.

## 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".

Create New Users User Actions ▾						
Showing 4 results						
User Name	Groups	Password	Password Last Used	Access Keys	Creation Time	
FredJones	1		N/A	1 active	2015-12-23 12:31 UTC	
JoeBlogs	0		N/A	1 active	2016-01-19 17:24 UTC	
JohnSmith	1		N/A	1 active	2015-12-23 12:31 UTC	
RyanKroonenburg	1	✓	Never	1 active	2015-12-23 12:31 UTC	

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.

Create New Group Group Actions ▾				
Showing 2 results				
Group Name	Users	Inline Policy	Creation Time	
AdminGroup	3		2015-12-23 12:39 UTC	
MyS3Group	0		2016-01-19 17:25 UTC	

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).

EC2-NoRoles i-4e86ddc3 t2.micro eu-west-1b running Initializing None ec2-52-31-226-95.eu-w... 52.31.226.95 MyEC2Key																																						
<table border="1"><tr><td>instance type</td><td>t2.micro</td></tr><tr><td>Private DNS</td><td>ip-172-31-41-91.eu-west-1.compute.internal</td></tr><tr><td>Private IPs</td><td>172.31.41.91</td></tr><tr><td>Secondary private IPs</td><td></td></tr><tr><td>VPC ID</td><td>vpc-fd83a398</td></tr><tr><td>Subnet ID</td><td>subnet-13df954a</td></tr><tr><td>Network Interfaces</td><td>eth0</td></tr><tr><td>Source/dest. check</td><td>True</td></tr><tr><td>EBS-optimized</td><td>False</td></tr><tr><td>elastic ip</td><td></td></tr><tr><td>Availability zone</td><td>eu-west-1b</td></tr><tr><td>Security groups</td><td>Web-DMZ, view rules</td></tr><tr><td>Scheduled events</td><td>No scheduled events</td></tr><tr><td>AMI ID</td><td>amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-bff32ccc)</td></tr><tr><td>Platform</td><td>-</td></tr><tr><td>IAM role</td><td></td></tr><tr><td>Key pair name</td><td>MyEC2Key</td></tr><tr><td>Owner</td><td>566216698943</td></tr><tr><td>Launch time</td><td>January 19, 2016 at 5:23:48 PM UTC (less than one hour)</td></tr></table>	instance type	t2.micro	Private DNS	ip-172-31-41-91.eu-west-1.compute.internal	Private IPs	172.31.41.91	Secondary private IPs		VPC ID	vpc-fd83a398	Subnet ID	subnet-13df954a	Network Interfaces	eth0	Source/dest. check	True	EBS-optimized	False	elastic ip		Availability zone	eu-west-1b	Security groups	Web-DMZ, view rules	Scheduled events	No scheduled events	AMI ID	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-bff32ccc)	Platform	-	IAM role		Key pair name	MyEC2Key	Owner	566216698943	Launch time	January 19, 2016 at 5:23:48 PM UTC (less than one hour)
instance type	t2.micro																																					
Private DNS	ip-172-31-41-91.eu-west-1.compute.internal																																					
Private IPs	172.31.41.91																																					
Secondary private IPs																																						
VPC ID	vpc-fd83a398																																					
Subnet ID	subnet-13df954a																																					
Network Interfaces	eth0																																					
Source/dest. check	True																																					
EBS-optimized	False																																					
elastic ip																																						
Availability zone	eu-west-1b																																					
Security groups	Web-DMZ, view rules																																					
Scheduled events	No scheduled events																																					
AMI ID	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-bff32ccc)																																					
Platform	-																																					
IAM role																																						
Key pair name	MyEC2Key																																					
Owner	566216698943																																					
Launch time	January 19, 2016 at 5:23:48 PM UTC (less than one hour)																																					

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]: 7dFQVn3t5hYRMt16B5ACdL1Ce+n/Wj56sdwkVpw
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".

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with options like Dashboard, Search IAM, Details, Groups, Users, Roles (which is selected and highlighted in orange), Policies, Identity Providers, Account Settings, and Credential Report. Below that is an Encryption Keys section. The main area has tabs for Create New Role and Role Actions. A search bar labeled 'Filter' is present. The results table shows two entries:

Role Name	Creation Time
acloudgurutestbucket1234-mybackupbucketacloudguru-s3-repl-role	2016-01-08 13:32 UTC
S3-Admin-Access	2015-12-23 12:47 UTC

At the bottom right of the main area, it says 'Showing 2 results'.

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 -

The screenshot shows a summary slide with the title 'Roles Lab' and the 'A CLOUD GURU' logo. The slide contains the following text:

Summary -

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.

# 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-

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

### Step 3: Configure Instance Details

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

**Number of instances** (1)  Launch into Auto Scaling Group (i)

**Purchasing option** (i)  Request Spot instances

**Network** (i) vpc-fd83a398 (172.31.0.0/16) (default)  Create new VPC

**Subnet** (i) No preference (default subnet in any Availability Zone)  Create new subnet

**Auto-assign Public IP** (i) Use subnet setting (Enable)

**IAM role** (i) S3-Admin-Access  Create new IAM role

**Shutdown behavior** (i) Stop

**Enable termination protection** (i)  Protect against accidental termination

**Monitoring** (i)  Enable CloudWatch detailed monitoring  
Additional charges apply.

**Tenancy** (i) Shared - Run a shared hardware instance   
Additional charges will apply for dedicated tenancy.

**Advanced Details**

User data (i)  As text  As file  Input is already base64 encoded

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

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/bin/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 browser, 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 -

The screenshot shows the AWS EC2 Auto Scaling console. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (with sub-options like Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts), Images (AMIs, Bundle Tasks), Elastic Block Store (Volumes, Snapshots), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers), and Auto Scaling (Launch Configurations, Auto Scaling Groups). The 'Launch Configurations' option is currently selected. The main content area is titled 'Welcome to Auto Scaling' and contains a brief introduction to Auto Scaling, a 'Create Auto Scaling group' button, and a note about selecting a region. Below this is a section titled 'Benefits of Auto Scaling' with three categories: 'Reusable Instance Templates' (illustrated with a gear and plus sign icon), 'Automated Provisioning' (illustrated with a checkmark and circular arrow icon), and 'Adjustable Capacity' (illustrated with a stack of three squares and a gear icon). Each category has a brief description and a 'Learn more' link. To the right, there's an 'Additional Information' sidebar with links to Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us.

From here we will create auto scaling group.

## Create Auto Scaling Group

[Cancel and Exit](#)

To create an Auto Scaling group, you will first need to choose a template that your Auto Scaling group will use when it launches instances for you, called a launch configuration. Choose a launch configuration or create a new one, and then apply it to your group.

Later, if you want to use a different template, you can create another launch configuration and apply it to this group, even if you already have instances running in it. Using this method, you can update the software that your group uses when it launches new instances.



### Step 1: Create launch configuration

First, define a template that your Auto Scaling group will use to launch instances. You can change your group's launch configuration at any time.

### Step 2: Create Auto Scaling group

Next, give your group a name and specify how many instances you want to run in it. Your group will maintain this number of instances, and replace any that become unhealthy or impaired.

You can optionally configure your group to adjust its capacity according to demand, in response to Amazon CloudWatch metrics.

[Cancel](#) [Create launch configuration](#)

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 -

## Create Launch Configuration

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

## Quick Start

My AMIs
AWS Marketplace
Community AMIs
<input type="checkbox"/> Free tier only ⓘ

 <b>Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-bff32ccc</b>	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	1 to 21 of 21 AMIs	< < > >
 <b>Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type - ami-8b8c57f8</b>	Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	64-bit	
 <b>SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type - ami-f4278487</b>	SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	64-bit	
 <b>Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-f95ef58a</b>	Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ).	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	64-bit	
 <b>Microsoft Windows Server 2012 R2 Base - ami-7943ec0a</b>	Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	64-bit	
 <b>Microsoft Windows Server 2012 R2 with SQL Server Express - ami-5a43ec29</b>	Microsoft Windows Server 2012 R2 Standard edition, 64-bit architecture, Microsoft SQL Server 2014 Express edition. [English]	<input style="background-color: #0070C0; color: white; border: none; padding: 2px 10px; border-radius: 5px; font-weight: bold; margin-right: 10px;" type="button" value="Select"/>	64-bit	

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. -

## Create Launch Configuration

Name

Purchasing option  Request Spot Instances

IAM role

Monitoring  Enable CloudWatch detailed monitoring  
[Learn more](#)

## ▼ Advanced Details

Kernel ID

RAM Disk ID

User data  As text  As file  Input is already base64 encoded

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

IP Address Type  Only assign a public IP address to instances launched in the default VPC and subnet. (default)  
 Assign a public IP address to every instance.  
 Do not assign a public IP address to any instances.

Note: this option only affects instances launched into an Amazon VPC

 Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

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 -

## Create Auto Scaling Group

Group size (i) Start with  instances

Network (i) vpc-fd83a398 (172.31.0.0/16) (default)  Create new VPC

Subnet (i)

- subnet-6490a213(172.31.16.0/20) | Default in eu-west-1a
- subnet-3a70655f(172.31.0.0/20) | Default in eu-west-1c
- subnet-13df954a(172.31.32.0/20) | Default in eu-west-1b

[Create new subnet](#)

Each instance in this Auto Scaling group will be assigned a public IP address. (i)

**Advanced Details**

Load Balancing (i)  Receive traffic from Elastic Load Balancer(s)

Health Check Type (i)  ELB  EC2

Health Check Grace Period (i)  seconds

Monitoring (i) Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration MyAutoScalingGroup. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.  
[Learn more](#)

Instance Protection (i)

[Cancel](#) [Next: Configure scaling policies](#)

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 -

## Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. [Learn more](#) about scaling policies.

- Keep this group at its initial size
- Use scaling policies to adjust the capacity of this group

We will select 2nd option-

### Create Auto Scaling Group

group automatically, based on your scaling policies.

- Keep this group at its initial size
- Use scaling policies to adjust the capacity of this group

Scale between  and  instances. These will be the minimum and maximum size of your group.

#### Increase Group Size

Name: Increase Group Size  
 Execute policy when: No alarm selected  Add new alarm  
 Take the action: Add  instances   
 Add step [①](#)  
 Instances need:  seconds to warm up after each step

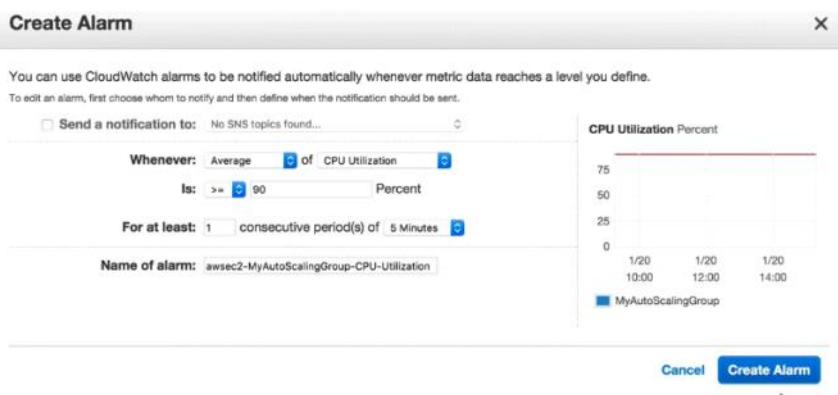
Create a simple scaling policy [①](#)

#### Decrease Group Size

Name: Decrease Group Size  
 Execute policy when: No alarm selected  Add new alarm  
 Take the action: Remove  instances   
 Add step [①](#)

Create a simple scaling policy [①](#)

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-

### Create Auto Scaling Group

- Use scaling policies to adjust the capacity of this group

Scale between  and  instances. These will be the minimum and maximum size of your group.

#### Increase Group Size

Name: Increase Group Size  
 Execute policy when: awsec2-MyAutoScalingGroup-CPU-Utilization [Edit](#) [Remove](#)  
 breaches the alarm threshold: CPUUtilization >= 90 for 300 seconds  
 for the metric dimensions AutoScalingGroupName = MyAutoScalingGroup  
 Take the action: Add  instances  when  <= CPUUtilization < +infinity  
 Add step [①](#)  
 Instances need:  seconds to warm up after each step

Create a simple scaling policy [①](#)

#### Decrease Group Size

Name: Decrease Group Size  
 Execute policy when: No alarm selected  Add new alarm  
 Take the action: Remove  instances   
 Add step [①](#)

Create a simple scaling policy [①](#)

Add step ⓘ

Create a simple scaling policy ⓘ

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-

Cancel Previous Review Next: Configure Notifications

1. Configure Auto Scaling group details    2. Configure scaling policies    3. Configure Notifications    4. Configure Tags    5. Review

### Create Auto Scaling Group

Configure your Auto Scaling group to send notifications to a specified endpoint, such as an email address, whenever a specified event takes place, including: successful launch of an instance, failed instance launch, instance termination, and failed instance termination.

If you created a new topic, check your email for a confirmation message and click the included link to confirm your subscription. Notifications can only be sent to confirmed addresses.

Send a notification to: SysAdminTeam [use existing topic](#)

With these recipients: acloudguru@gmail.com

Whenever instances:

- launch
- terminate
- fail to launch
- fail to terminate

[Add notification](#)

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-

1. Configure Auto Scaling group details    2. Configure scaling policies    3. Configure Notifications    4. Configure Tags    5. Review

### Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

#### Auto Scaling Group Details

Edit details

Group name	MyAutoScalingGroup
Group size	3
Minimum Group Size	1
Maximum Group Size	5
Subnet(s)	subnet-3a70655f,subnet-6490a213,subnet-13df954a
Load Balancers	MyWebDMZ
Health Check Type	ELB
Health Check Grace Period	150
Detailed Monitoring	No
Instance Protection	None

#### Scaling Policies

Edit scaling policies

#### Notifications

Edit notifications

SysAdminTeam  
(acloudguru@gmail.com) launch, terminate, fail to launch, fail to terminate

#### Tags

Edit tags

StaffID	43534534	tag new instances
Department	Test&Dev	tag new instances

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

EC2 Dashboard    Launch Instance    Connect    Actions ⏮

Events    Tags    Reports    Limits

Instances    Instances    Spot Requests    Reserved Instances    Scheduled Instances    Commands    Dedicated Hosts

Filter by tags and attributes or search by keyword

Name    Instance ID    Instance Type    Availability Zone    Instance State    Status Checks    Alarm Status    Public DNS    Public IP    Key Name

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
metadata	i-bcebe31	t2.micro	eu-west-1b	terminated	None	None	None	52.16.2.58	MyEC2Key
	i-459592ce	t2.micro	eu-west-1a	running	2/2 checks ...	None	ec2-52-16-2-58.eu-wes...	52.16.2.58	MyEC2Key
	i-b7c4e13a	t2.micro	eu-west-1b	running	2/2 checks ...	None	ec2-52-16-88-234.eu-w...	52.16.88.234	MyEC2Key
	i-0c18e584	t2.micro	eu-west-1c	running	2/2 checks ...	None	ec2-52-48-25-160.eu-w...	52.48.25.160	MyEC2Key

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-

[Create Load Balancer](#) [Actions ▾](#)

Filter:  Search Load Balancers [X](#)

Load Balancer Name	DNS Name	Port Configuration	Availability Zones	Instance Count	Health Check	Created At
MyWebDMZ	MyWebDMZ-260429352.eu...	80 (HTTP) forwarding to 80 (...	eu-west-1b, eu-west-1c...	1 Instance	HTTP:80/healthcheck.html	January 19, 2016 at 3:49:41 ...

Load balancer: MyWebDMZ [Edit](#) [Delete](#) [Details](#) [Logs](#)

[Description](#) [Instances](#) [Health Check](#) [Monitoring](#) [Security](#) [Listeners](#) [Tags](#)

Connection Draining: Enabled, 300 seconds ([Edit](#))

[Edit Instances](#)

Instance ID	Name	Availability Zone	Status	Actions
i-b7c4e13a		eu-west-1b	InService ⓘ	<a href="#">Remove from Load Balancer</a>

[Edit Availability Zones](#)

Availability Zone	Subnet ID	Subnet CIDR	Instance Count	Healthy?	Actions
eu-west-1b	subnet-13df954a	172.31.32.0/20	1	Yes	<a href="#">Remove from Load Balancer</a>
eu-west-1c	subnet-3a70655f	172.31.0.0/20	0	No (Availability Zone contains no healthy instances)	<a href="#">Remove from Load Balancer</a>
eu-west-1a	subnet-6490a213	172.31.16.0/20	0	No (Availability Zone contains no healthy instances)	<a href="#">Remove from Load Balancer</a>

Here we can see that only one instance is running.

Now we can see from our EC2 main console page -

[Launch Instance](#) [Connect](#) [Actions ▾](#)

Filter by tags and attributes or search by keyword [?](#) [X](#) [1 to 5 of 5 > |](#)

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
metadata	i-boebce31	t2.micro	eu-west-1b	terminated	None	None			MyEC2Key c
bootstrapping	i-98929513	t2.micro	eu-west-1a	pending	Initializing	None			MyEC2Key c
instance-0	i-0c18e584	t2.micro	eu-west-1c	terminated	None	None			MyEC2Key c
instance-1	i-459592ce	t2.micro	eu-west-1a	terminated	None	None			MyEC2Key c
instance-2	i-b7c4e13a	t2.micro	eu-west-1b	running	2/2 checks ...	None	ec2-52-18-88-234.eu-w...	52.16.88.234	MyEC2Key c

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 with 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 -

EC2 Placement Groups

## What is a Placement Group?



A CLOUD GURU

A placement group is a logical grouping of instances within a single Availability Zone. Using placement groups enables applications to participate in a low-latency, 10 Gbps network. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both.

Few things to remember about placement groups-

EC2 Placement Groups

## EC2 Placement Groups



A CLOUD GURU

- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group must be unique within your AWS account.
- Only certain types of instances can be launched in a placement group (Compute Optimized, GPU, Memory Optimized, Storage Optimized)
- AWS recommend homogenous instances within placement groups.
- You can't merge placement groups.
- You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

## EC2 - EFS - lab

Tuesday, December 27, 2016 4:14 PM

EFS stands for - elastic file system.

**What is EFS**

A CLOUD GURU

Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

Features -

**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

Now we will add EFS -

Amazon Elastic File System (EFS)

Amazon EFS provides file storage for use with your EC2 instances.

Create file system

Getting started guide



### Create

Create an Amazon EFS file system to store your files in the Amazon cloud. A file system grows and shrinks automatically with the files you put in, and you pay only for what you use.



### Access

Write files to and read files from your Amazon EFS file system via the NFSv4 protocol. Any number of EC2 instances can work with your file system at the same time, and your instances can be in multiple Availability Zones in a region.

### Manage

You can easily administer your file system using the Amazon EFS console, CLI, and SDK.

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

## Create file system

### Step 1: Configure file system access

Step 2: Add tags

Step 3: Review and create

### Configure file system access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system via a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC vpc-86e13be3 (default)

#### Create mount targets

Instances connect to a file system via mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

	Availability Zone	Subnet	IP address	Security group	
<input checked="" type="checkbox"/>	us-west-2a	subnet-4f56eb2a (default)	Automatic	sg-cb1254ae - default	
<input checked="" type="checkbox"/>	us-west-2b	subnet-291fcc5e (default)	Automatic	sg-cb1254ae - default	
<input checked="" type="checkbox"/>	us-west-2c	subnet-e7c232be (default)	Automatic	sg-cb1254ae - default	

[Cancel](#)

[Next Step](#)

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 -

## Create file system

### Step 1: Configure file system access

### Step 2: Add tags

Step 3: Review and create

### Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key : value pair. For example, you can define a tag with key : value pair Corporate Department : Sales and Marketing. At a minimum, we recommend a "Name" tag.

Key	Value	Remove
Name	MyEFSFileSystem	
Add New Key		

[Cancel](#)

[Previous](#)

[Next Step](#)

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

## Create file system

### Step 1: Configure file system access

### Step 2: Add tags

### Step 3: Review and create

### Review and create

Review the configuration below before proceeding to create your file system.

#### File system access

VPC	Availability Zone	Subnet	IP address	Security group
vpc-86e13be3 (default)	us-west-2a	subnet-4f56eb2a (default)	Automatic	sg-cb1254ae - default
	us-west-2b	subnet-291fcc5e (default)	Automatic	sg-cb1254ae - default
	us-west-2c	subnet-e7c232be (default)	Automatic	sg-cb1254ae - default

#### Tags

Name: MyEFSFileSystem

[Cancel](#)

[Previous](#)

[Create File System](#)

Finally after clicking OK we will see following window -

**File systems**

**Success!**

You have created a file system. You can mount your file system from an EC2 instance with an NFSv4 client installed. Click [here](#) for instructions.

Create file system		Actions			
	Name	File system ID	Metered size	Number of mount targets	Creation date
	MyEFSFileSystem	fs-79e40cd0	6.0 KIB	3	2016-01-22T16:05:30Z

**Other details**

Owner ID: 242374741609  
Life cycle state: Available

**Tags**

Name: MyEFSFileSystem

**File system access**

DNS names  
EC2 mount instructions

**Mount targets**

VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Life cycle state
vpc-86e13be3 (default)	us-west-2a	subnet-4f56eb2a (default)	172.31.37.189	fsmt-da2cc773	eni-0d72a175		Creating
	us-west-2b	subnet-291fcc5e (default)	172.31.25.90	fsmt-dd2cc774	eni-4be57d00		Creating
	us-west-2c	subnet-e7c232be (default)	172.31.3.139	fsmt-dc2cc775	eni-90f6b2ca		Creating

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-

### Step 5: Add EC2 Instances

The table below lists all your running EC2 Instances. Check the boxes in the Select column to add those instances to this load balancer.

VPC vpc-86e13be3 (172.31.0.0/16)

Instance	Name	State	Security Groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-58b2799f	UsWest2A	running MyWebDMZ	us-west-2a	subnet-4f56eb2a	172.31.32.0/20
<input checked="" type="checkbox"/>	i-f5e8f92c	UsWest2B	running MyWebDMZ	us-west-2b	subnet-291fcc5e	172.31.16.0/20

#### Availability Zone Distribution

1 instance in us-west-2a  
1 instance in us-west-2b

Enable Cross-Zone Load Balancing   
 Enable Connection Draining 300 seconds

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

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
<input type="checkbox"/>	west2.1	i-4eab6089	t2.micro	us-west-2a	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input type="checkbox"/>	west2	i-50ab6097	t2.micro	us-west-2a	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input type="checkbox"/>	west1	i-57f1e08e	t2.micro	us-west-2b	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input checked="" type="checkbox"/>	UsWest2A	i-58b2799f	t2.micro	us-west-2a	<span style="color: green;">●</span> running	<span style="color: yellow;">■</span> Initializing	None	<span style="color: green;">●</span>	ec2-54-201-47-66.us-w...	54.201.47.66
<input type="checkbox"/>		i-bdbc777a	t2.micro	us-west-2a	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input type="checkbox"/>		i-bebc7779	t2.micro	us-west-2a	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input type="checkbox"/>		i-bfbc7778	t2.micro	us-west-2a	terminated	<span style="color: red;">●</span>	None	<span style="color: green;">●</span>		MyOrgeonKey...
<input type="checkbox"/>	UsWest2B	i-f5ebf92c	t2.micro	us-west-2b	<span style="color: green;">●</span> running	<span style="color: yellow;">■</span> Initializing	None	<span style="color: green;">●</span>	ec2-54-213-65-98.us-w...	54.213.65.98

Instance: i-58b2799f (UsWest2A) Public DNS: ec2-54-201-47-66.us-west-2.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
<p>Instance ID: i-58b2799f            Instance state: running            Instance type: t2.micro            Private DNS: ip-172-31-46-212.us-west-2.compute.internal            Private IPs: 172.31.46.212            Secondary private IPs:            VPC ID: vpc-86e13be3</p>	<p>Public DNS: ec2-54-201-47-66.us-west-2.compute.amazonaws.com            Public IP: 54.201.47.66            Elastic IP: -            Availability zone: us-west-2a            Security groups: MyWebDMZ, view rules            Scheduled events: No scheduled events            AMI ID: amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-f0091d91)</p>		

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 -

	Name	File system ID	Metered size	Number of mount targets	Creation date
<span style="color: blue;">●</span>	MyEFSFileSystem	fs-79e40cd0	6.0 KB	3	2016-01-22T16:05:30Z

Other details

Owner ID: 242374741609	Tags	Manage tags
Life cycle state: Available		

File system access

DNS names  
[EC2 mount instructions](#) (link)

Mount targets

VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Life cycle state
vpc-86e13be3 (default)	us-west-2a	subnet-4f56eb2a (default)	172.31.37.189	fsmt-da2cc773	eni-0d72a175	sg-cb1254ae - default	Available
	us-west-2b	subnet-291fcc5e (default)	172.31.25.90	fsmt-dd2cc774	eni-4be57d00	sg-cb1254ae - default	Available
	us-west-2c	subnet-e7c232be (default)	172.31.3.139	fsmt-dc2cc775	eni-90f6b2ca	sg-cb1254ae - default	Available

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

EC2 mount instructions

Setting up your EC2 instance

- Using the [Amazon EC2 console](#), associate your EC2 instance with a VPC security group that enables access to your mount target. For example, if you assigned the "default" security group to your mount target, you should assign the "default" security group to your EC2 instance. (learn more about [using VPC security groups with Amazon EFS](#))
- Open an SSH client and connect to your EC2 instance. (find out how to [connect](#))
- Install the nfs client on your EC2 instance.
  - On an Amazon Linux, Red Hat Enterprise Linux, or SuSE Linux instance:  
`sudo yum install -y nfs-utils`
  - On an Ubuntu instance:  
`sudo apt-get install nfs-common`

Mounting your file system

- Open an SSH client and connect to your EC2 instance. (find out how to [connect](#))
- Create a new directory on your EC2 instance, such as "efs".  
`sudo mkdir efs`
- Mount your file system using the DNS name. The following command looks up your EC2 instance's Availability Zone (AZ) using

[Close](#)

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

Now for this command-

- Mount your file system using the DNS name. The following command looks up your EC2 instance's Availability Zone (AZ) using the EC2 instance metadata URI 169.254.169.254, then mounts the file system using the DNS name for that AZ. ([what is EC2 instance metadata?](#))
- `sudo mount -t nfs4 $(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone).fs-79e40cd0.efs.us-west-2.amazonaws.com:/efs`

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-

Instance ID	Name	Availability Zone	Status	Actions
i-58b2799f	UsWest2A	us-west-2a	InService	<a href="#">Remove from Load Balancer</a>
i-f5e0f92c	UsWest2B	us-west-2b	InService	<a href="#">Remove from Load Balancer</a>

Both of our servers are InService.

# EC2 - Summary

Tuesday, December 27, 2016 4:50 PM

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 Purpose 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 Instance Types



Family	Specialty	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 Exam Tips



- 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.

## Volumes vs Snapshots



- 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.

## Volumes vs Snapshots - Security



- 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

## Snapshots of Root Device Volumes



- To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.

## EBS vs Instance Store - Exam Tips



- 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.

## 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.

## 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.

## 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.

## Exam Tips



A CLOUD GURU

- Standard Monitoring = 5 Minutes
- Detailed Monitoring = 1 Minute
- CloudWatch is for performance monitoring
- CloudTrail is for auditing

## What can I do with Cloudwatch?



- 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 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 Meta-data



- 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 Features



- 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

## What Is Lambda?



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