AWS: Amazon Web Services Lab Practice Guide

Document has been prepared for lab practice only not for production deployments

Prepared for:
Public

Prepared by:
Ankam Ravi Kumar

Follow Me on Social Networking Sites

Facebook | Google Plus | Twitter | Reddit | LinkedIn | Website | Blog | YouTube

Reach me over Email: aravikumar48@gmail.com or aravi@server-computer.com

If you think this document helped a lot Donate a dollar as complementary

Table of Contents

1. About Author

Ankam Ravi Kumar has more than 10+ years of experience in Information Technology Operations and production support streams. He served more than 5 companies in his career and still continuing.

We provide server and data center related services from purchasing of underlying hardware to provisioning the applications.

Solid industry experience in Infrastructure Management/Customer Support/Operations and Training Domains. I love to help people by sharing my knowledge and skills. I always believe "Power is gained by Sharing Knowledge not hoarding it".

- Operating System Management Such has Linux Different Flavors, Red hat, Fedora, Ubuntu, AIX, Solaris and Windows
- Enterprise Server Management
- Installing and configuring Blade Servers
- Core Storage Management Dell-EMC, IBM and NetApp
- Database Management MSSQL, POSTGRESQL, MariaDB and MySQL
- Process Management ITIL
- Virtualization management RHEV, vSphere, VMware, KVM, Hyper-V and XEN
- Backup and Recovery Management NetVault, Commvault and Symantec Backup Exec
- Application Server Management and Storage Cluster Management
- Data Center Management and Hosting Solutions
- Programming Languages such as PHP and HTML
- Scripting Languages Shell, Perl and Python

Specialized in managing and building the Teams for IT services delivery and Service Support, Training and Operations in both smaller and larger companies. Rich experience and strong exposure in IT Infrastructure & Data Center Management.

Implementation of monitoring solutions for Enterprise, Using Tools Nagios, NagiosXI, Cacti, Solarwinds and LogicMonitor.

2. Services we provide to our customers

Data Storage

Any type of storage categories like DAS, NAS, SAN and Unified. Like Netapp, Dell-EMC, IBM, HP, Hitachi, Pure storage and Synology.

Backup and Recovery

We provide solutions for Online and Offline data backup. RPO and RTO less than ~5Minutes for any disaster recovery.

Networking

Switching and routing. Specialized in Paloalto firewall configurations and VPN. Spam filtering and proxy configurations.

Servers

Starting from server hardware configuration, requirement gathering to installing and configuring. Racking, Operating system and application to production. All brands.

Tape Libraries

We do provide tape library with backup software's. starting from LTO3, LTO4, LTO5, LTO6 and LTO7. Qualstar, Dell, Quantum, HP and IBM.

Telecommunication

Like PRI Lines, SIP, VoIP Services. Software and Hardware solutions for Inband and outband.

Virtualization

Virtualization environment implementation, configurations and migrations. Vmware, Hyper-V and RHEV.

Web Applications

Web application development. web designing and web development.

Application Migrations

We handle a large number of application migrations, data migrations from on-frame to cloud and cloud to on-frame. Any kind of old systems data CIFS shares, User data migrations we will handle with care.

3. Cloud Computing Models

There are three main models for cloud computing. Each model represents a different part of the cloud-computing stack.

3.1. Infrastructure as a Service (IaaS):

Infrastructure as a Service, sometimes abbreviated as IaaS, contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. Infrastructure as a Service provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.

3.2. Platform as a Service (PaaS):

Platforms as a service remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

3.3. Software as a Service (SaaS):

Software as a Service provides you with a completed product that is run and managed by the service provider. In most cases, people referring to Software as a Service are referring to end-user applications. With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece software. A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email program is running on.

4. Amazon Free Tier Account Creation

Read these conditions before creating a free tier account.

- Amazon Elastic Cloud computer EC2 Linux t2.micro 750Hours per month
- 750 Hours t2.micro windows instance per month
- 2000 Put requests of Amazon S3 (single PUT Request max 5GB)
- 20000 Get requests of Amazon S3 (Each request Get request)
- Amazon RDS MySQL DB instance with t2.micro 5GB storage
- MSSQL Express version t2.micro with 20GB GP-SSD Free tier

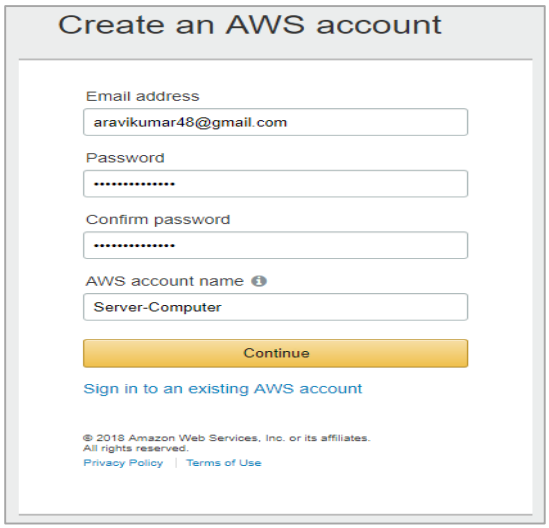
https://aws.amazon.com/free/

Prerequisites:

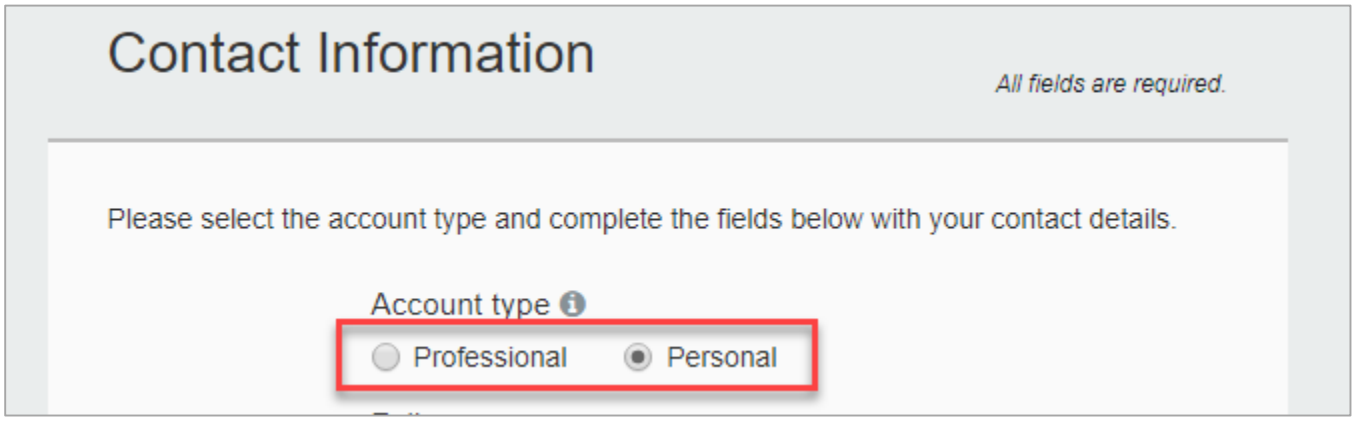
- Credit card with minimum 1$ available balance
- Reachable mobile number for verification

https://aws.amazon.com/console/

Click on 

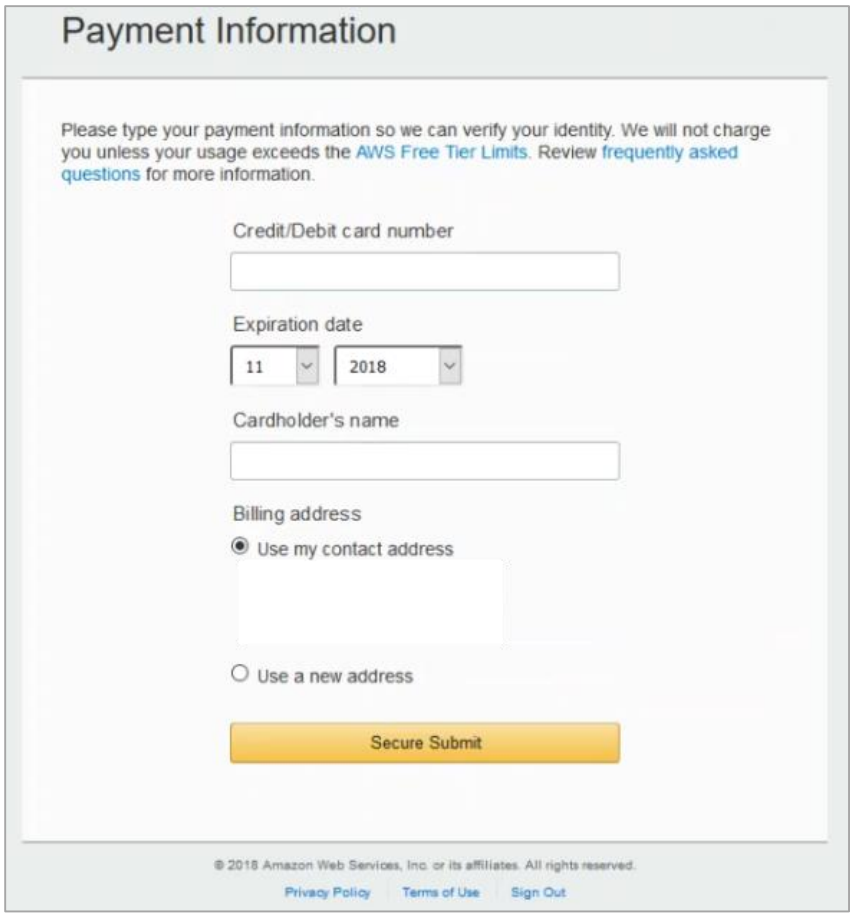


Fill the details example is shown above and **click continue**



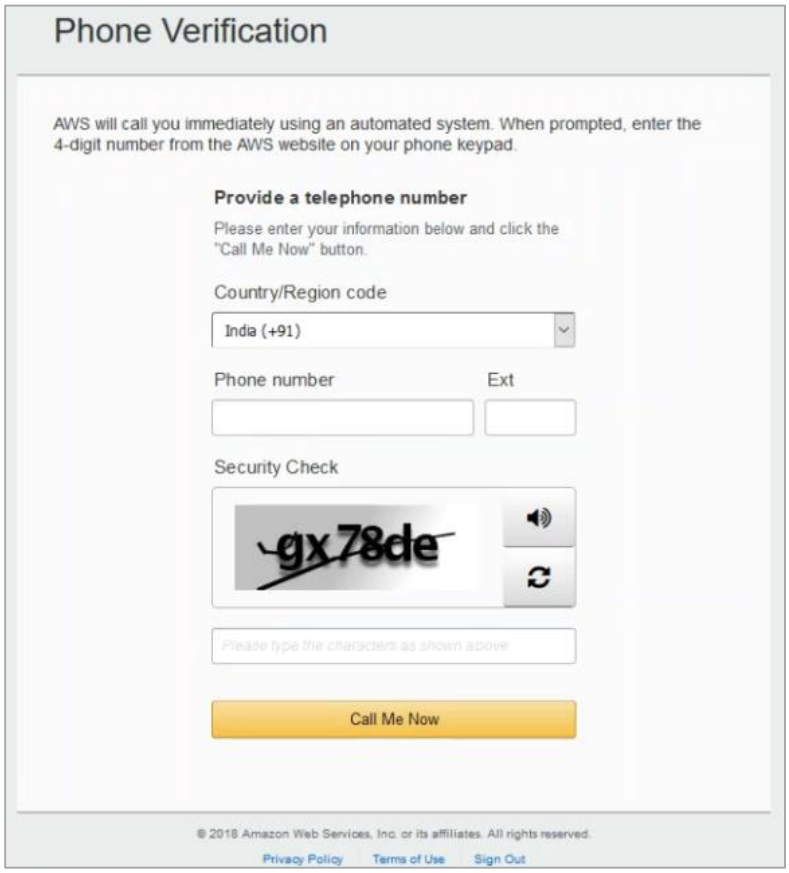
Click on radio button

- Professional is for company
- Personal is for single person



Provide your credit card details correctly, Card Number, Expiry Date and Card Holder Name

Click on **Secure Submit**

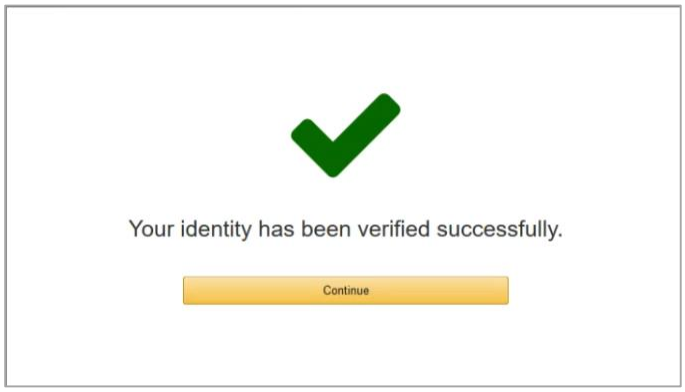


It will ask you to enter phone number, Security check then click on **Call Me Now**

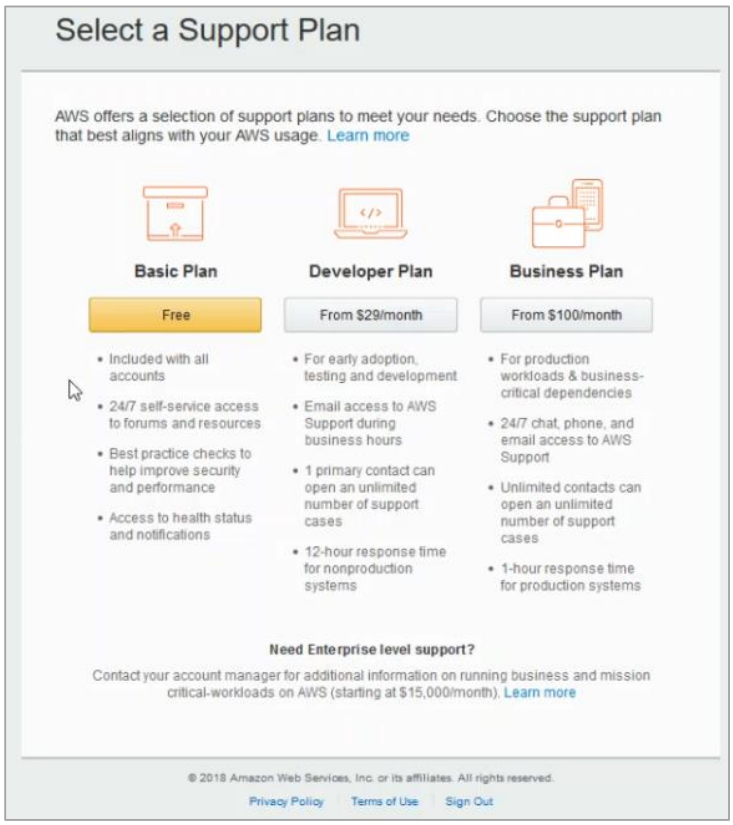
You will receive a call from AWS tele communication and ask you to enter the code displayed on screen.

Note: Listen All the Details carefully and proceed by entering code displayed on screen.

After successful verification



Continue



Select Support plan in this case select **Free**

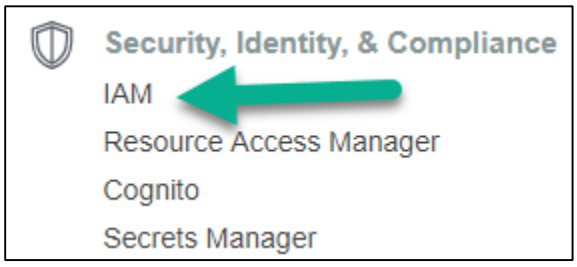
You successfully completed Free Tier Account Creation. Login and Enjoy AWS Free Tier.

AWS Console

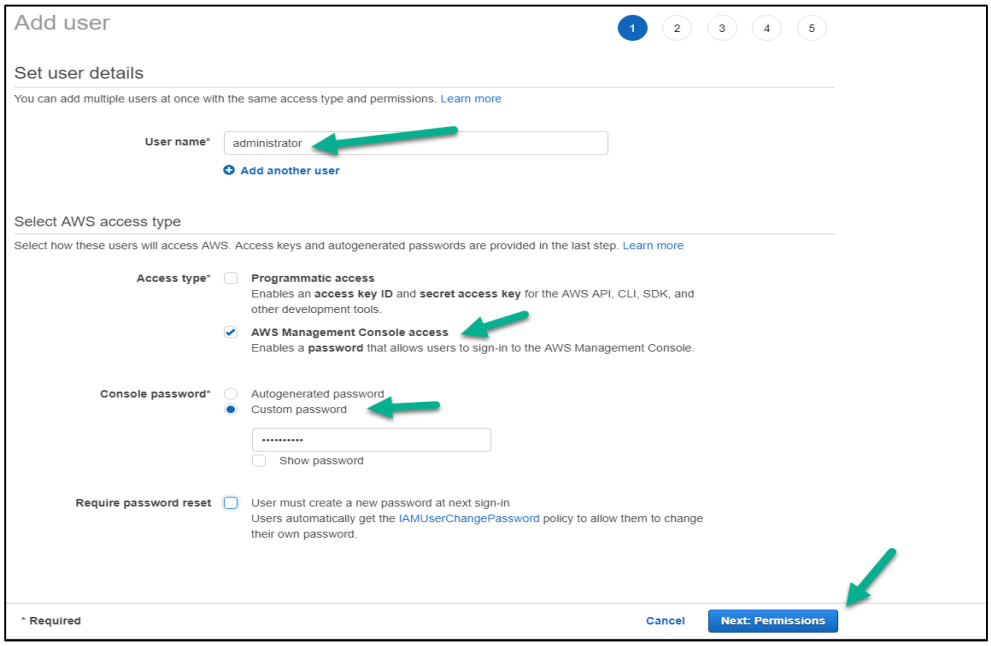
Provide your email address and password to **Sign In**

5. Enabling Multi-Factor Authentication to Secure Your Access

Go To IAM **Services → Security, Identify & Compliance → IAM**



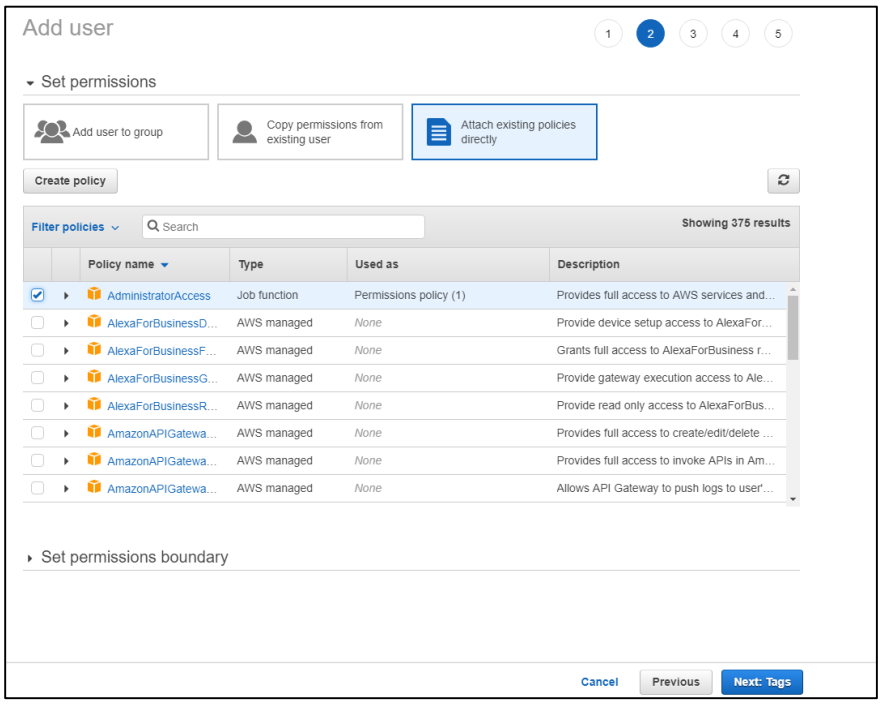
Click on Users → Add User



Provide user name, select access type

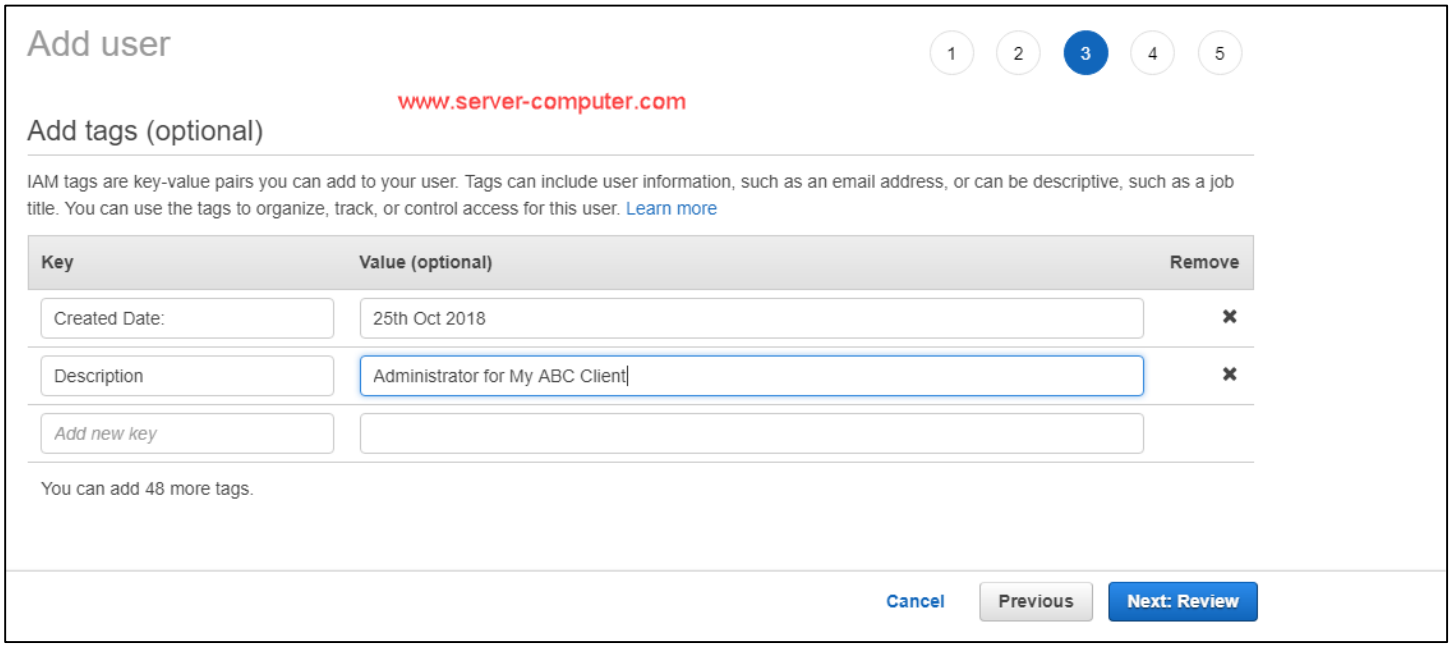
- Programmatic Access – Required for automation, run any operation using programs
- AWS Management Console Access – User will have web console access

Click **Next Permissions**

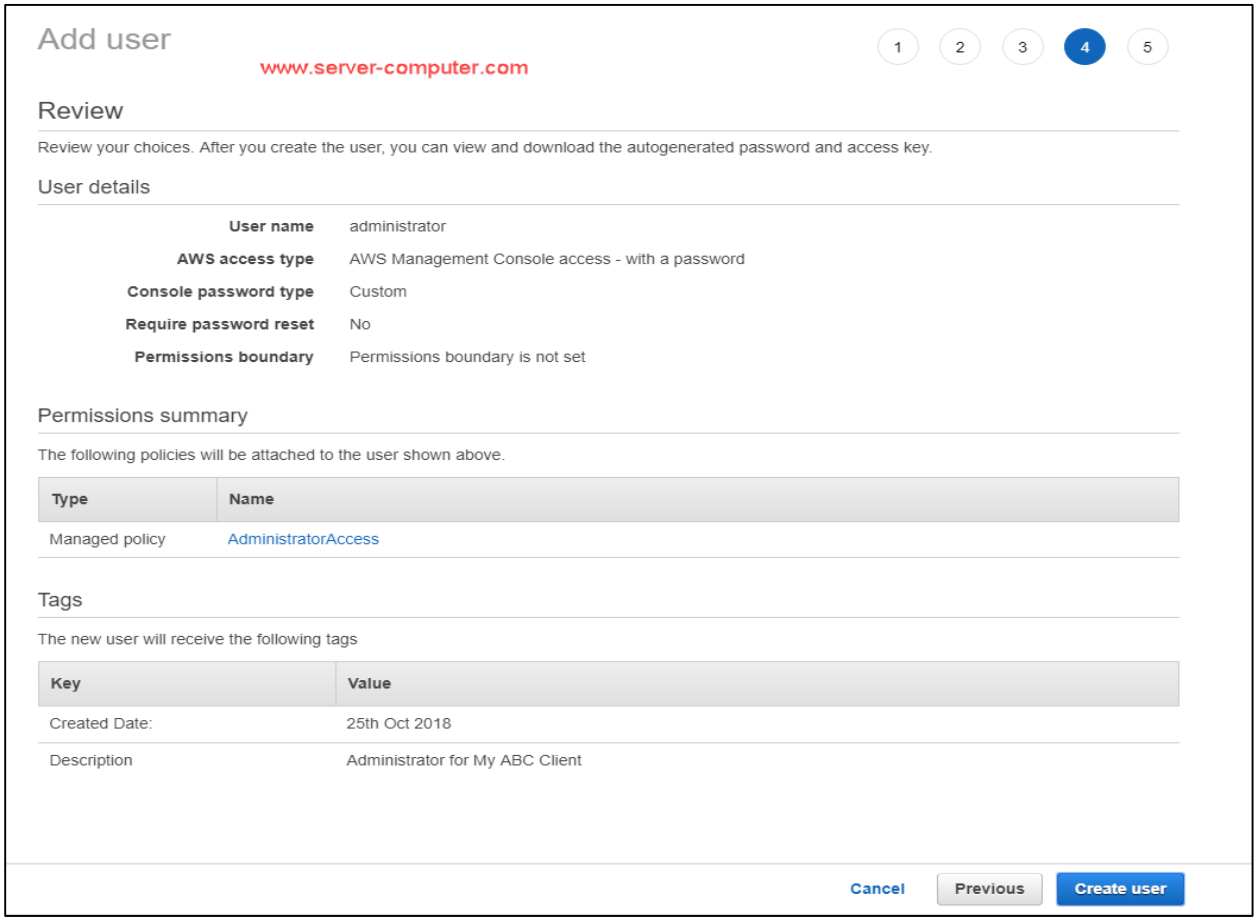


Click **Next: Tags**

Add tags whatever required to identify user



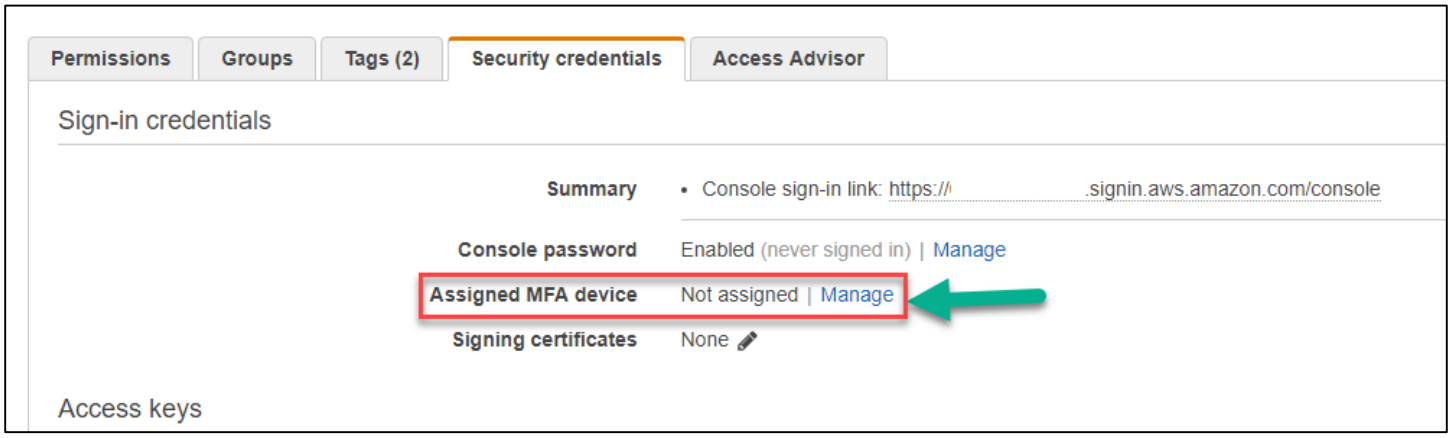
Click **Next: Review**



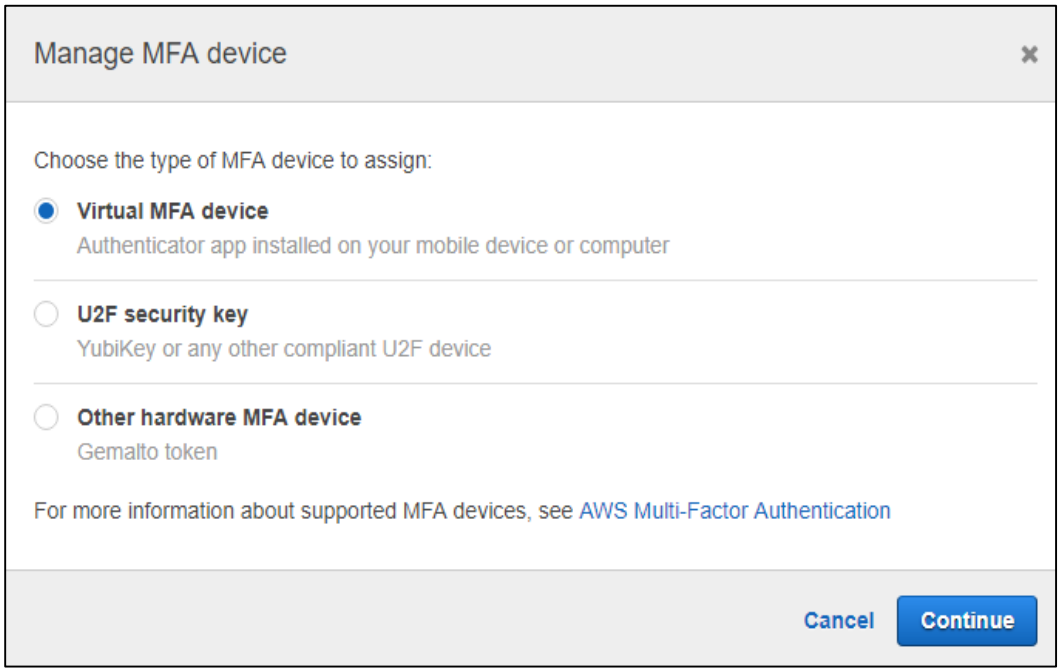
Click **Create User**

User creation has been completed successfully now you will get on access URL with your account number. Note the URL.

Now Click on User name → Security credentials (TAB)



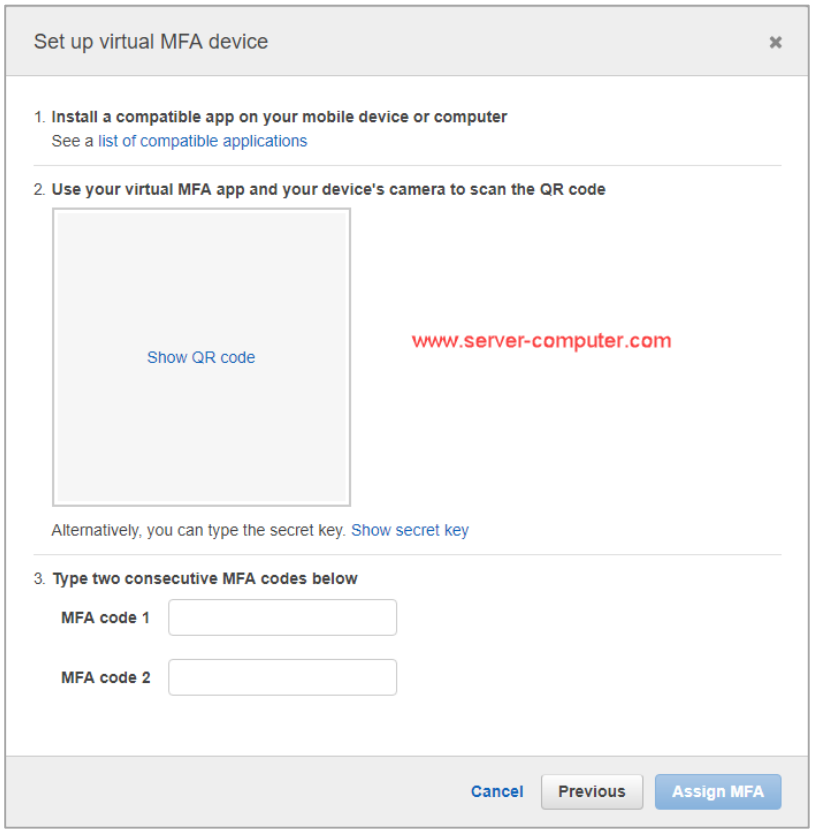
Click on Assigned MFA Device – Manage



Use any method based on your requirement. Here I am showing Virtual MFA Device method

Install Google Authenticator in your smart phone and ready to pair

Click **Continue**



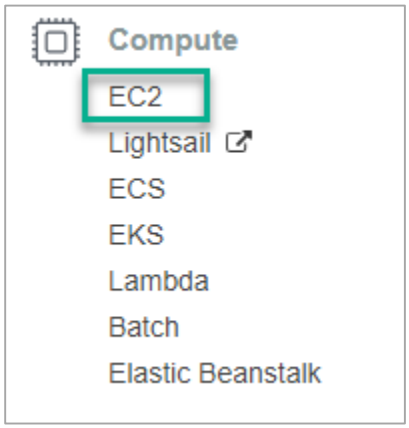
Click in **Show QR Code** and scan the same code from your Google authenticator App. It will generate six digit codes enter one code in first MFA code 1 wait 1 minute and second code in MFA Code 2 Click on **Assign MFA**

That's it, now you successfully enabled MFA (Multi-Factor Authentication).

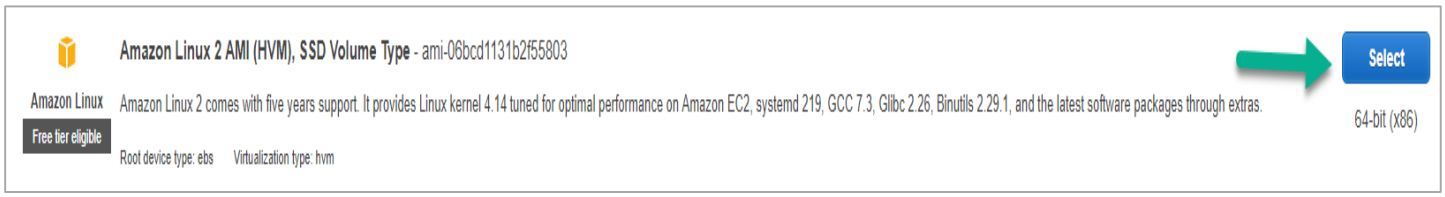
Here after if you want to login, you have to enter credentials and MFA code to Login.

6. Creating First Linux Instance

Login to AWS console, services drop down click on EC2



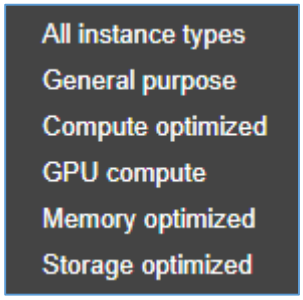
Click on **Launch instance**

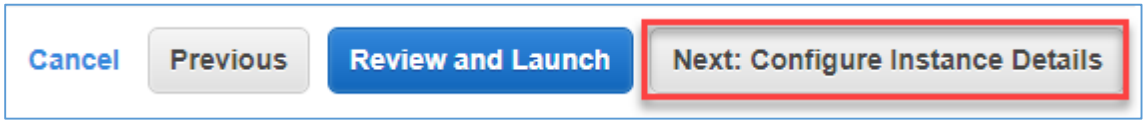


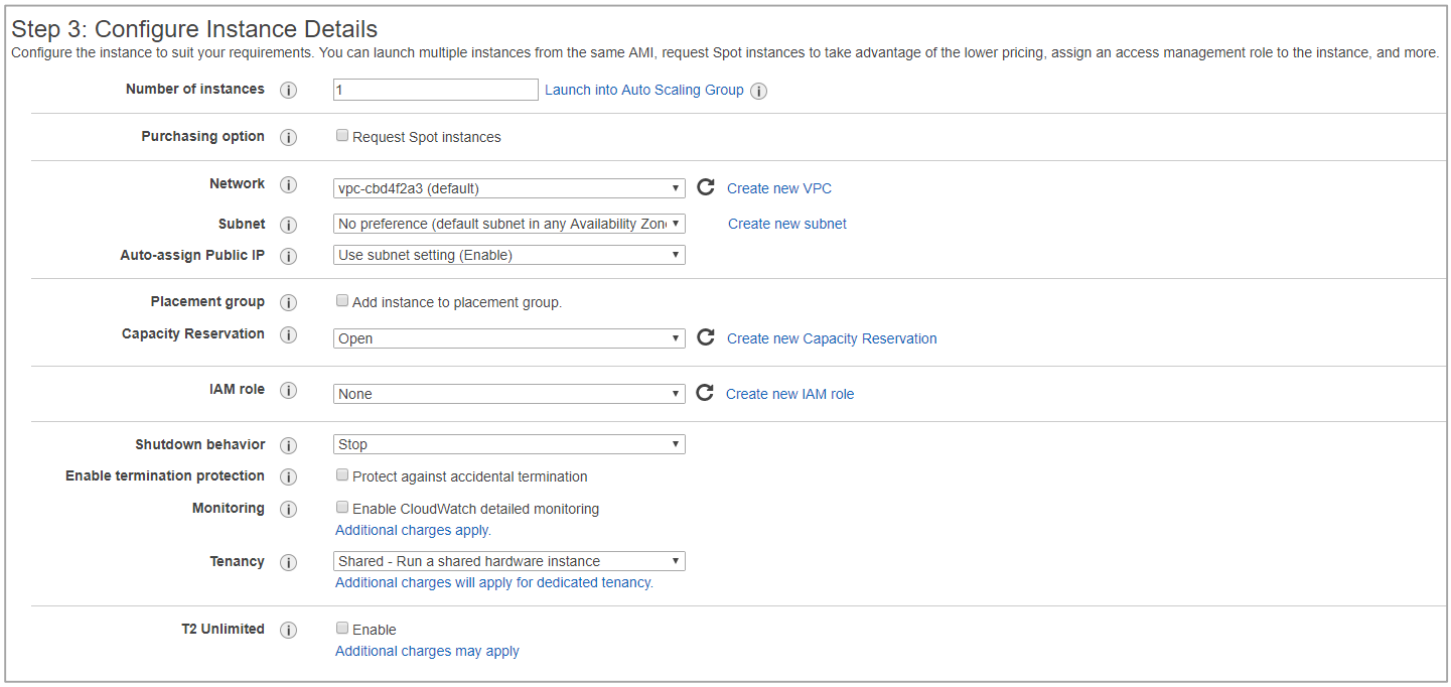
I am selecting Free Tier instance Amazon Linux

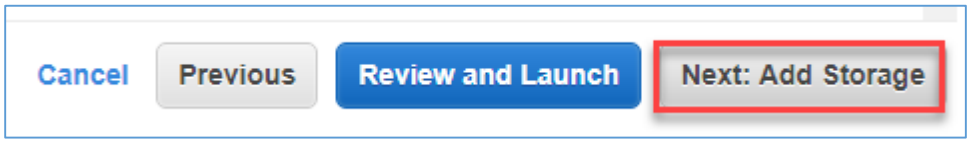
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

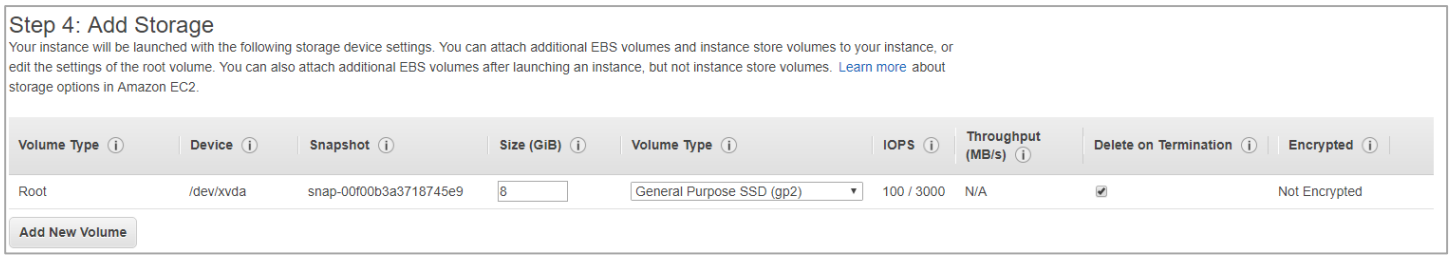
We have below types of instances



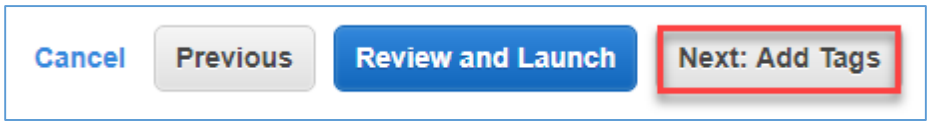




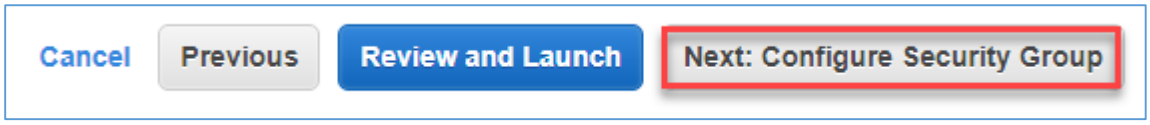


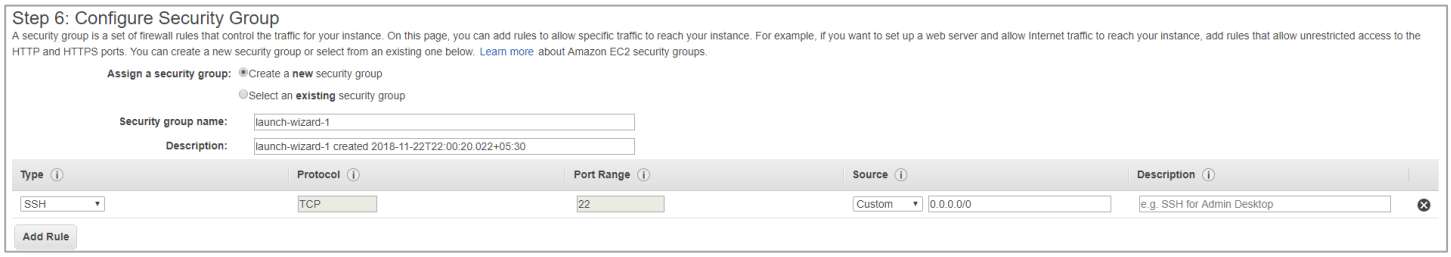


Add storage – EBS Elastic Block Storage volume will attached to your instance



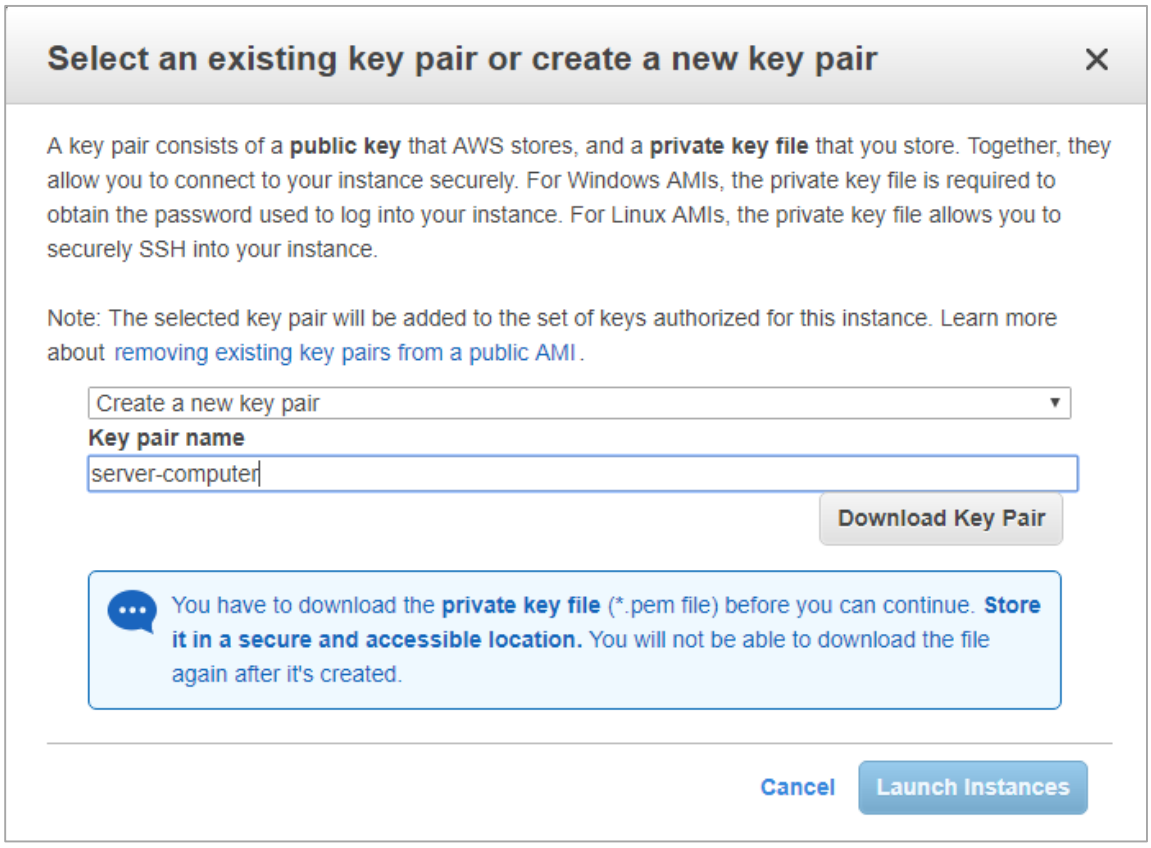
Tags to identify the details about instance (Production/Test/Dev/Client Name)





Using security group we can allow/deny any ports

Verify the details and click on **Launch**



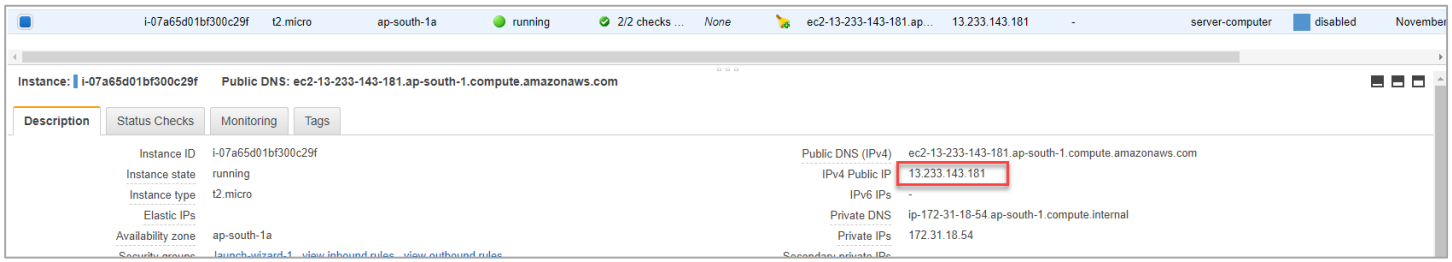
For the first time you **create a new key pair** and **Download Key Pair**

Server-computer.pem file will downloaded, **keep it safe**

Launch Instances

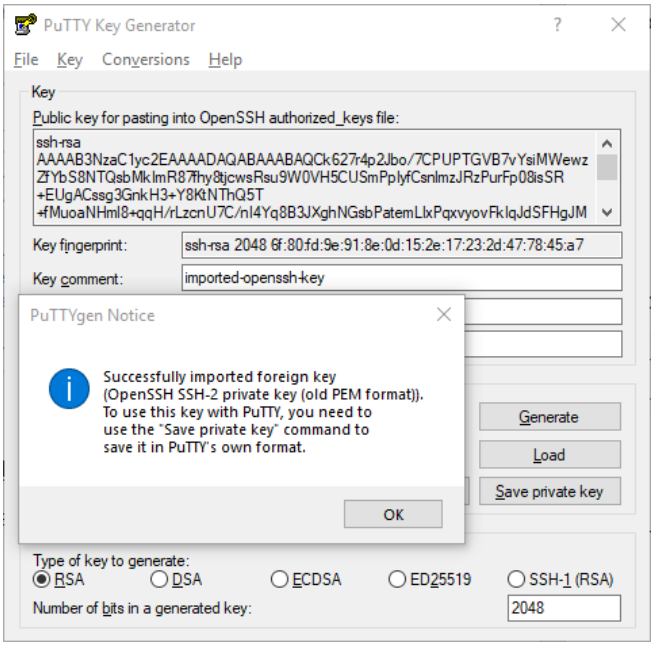
Go to EC2 → See the instances

Click on instance and copy the Public IP Address



Install putty msi installer you will get PuttyGen and Putty for accessing Linux machine

Open puttyGen and load server-computer.pem file

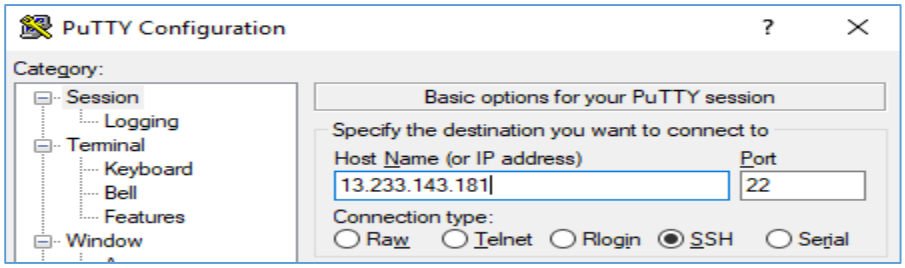


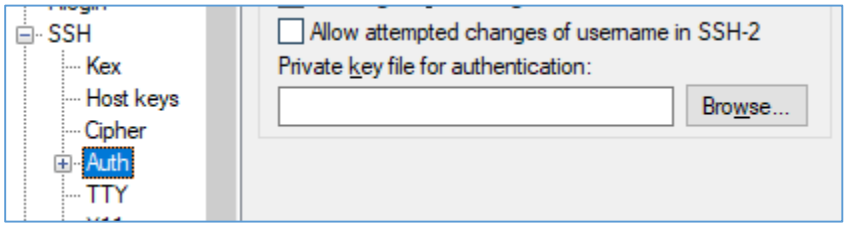
Click Ok.

Save Private Key

In this case, I have used server-computer1.ppk

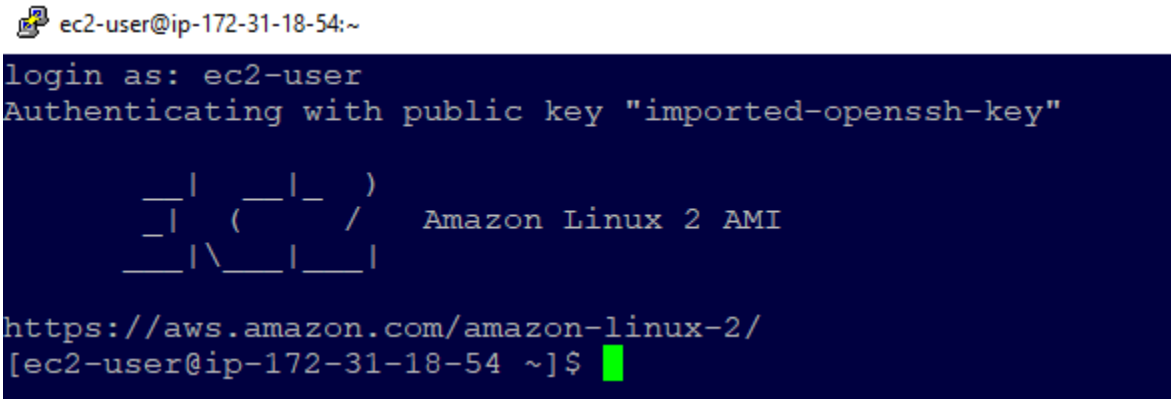
Open putty application and type IP address as shown below





Expand SSH → Click on Auth → Browse and attach .ppk file

Click on **Open**



You successfully logged into your Amazon Linux instance

As example, we are going to install web server in Linux server and access using web browser

https://github.com/techtutorials/aws-lab-guide/blob/aws/webserver.sh

You can also use above shell script to automatically build webserver for you

```
sudo yum update -y;
sudo yum install httpd -y;
sudo service httpd start;
sudo service httpd status;
sudo chkconfig httpd on;
```

Now go back to your EC2 → Security Groups and Add 80 port

Open browser and type your instance public IP address you can access web-server test page.

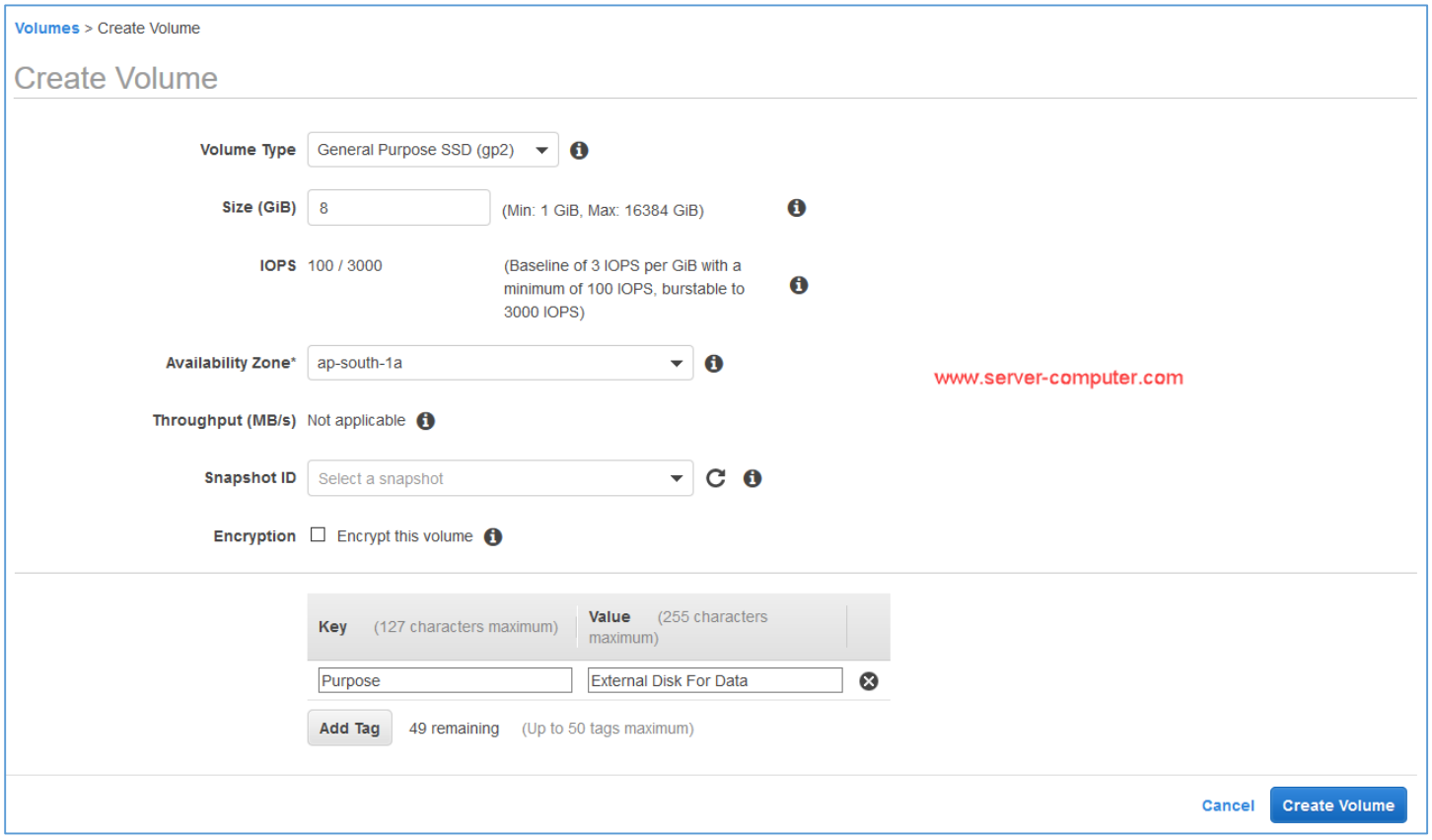
7. Adding New EBS Volume to Linux Instance

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability. Amazon EBS volumes offer the consistent and low-latency performance needed to run your workloads. With Amazon EBS, you can scale your usage up or down within minutes – all while paying a low price for only what you provision. EBS is designed for application workloads that benefit from fine tuning for performance, cost and capacity.

EC2 Console Left side → **Elastic Block Store → Volumes**

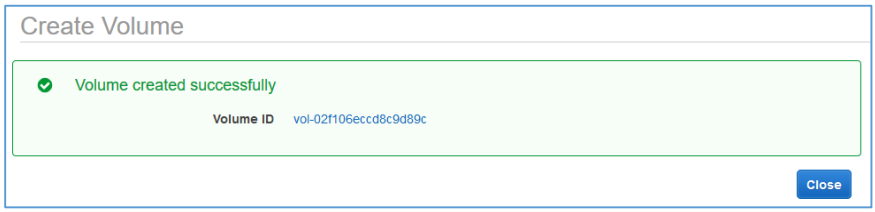
Select required type of EBS Volume from below types

- General Purpose SSD(gp2)
- Provisioned IOPD SSD (io1)
- Cold HDD(sc1)
- Throughput Optimized HDD (st1)
- Magnetic (standard)

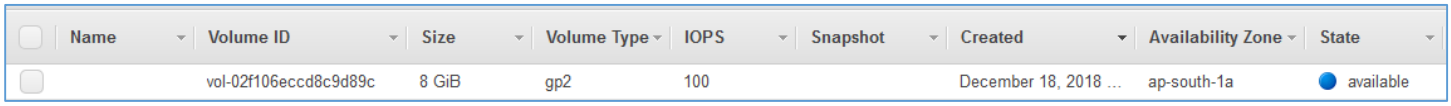


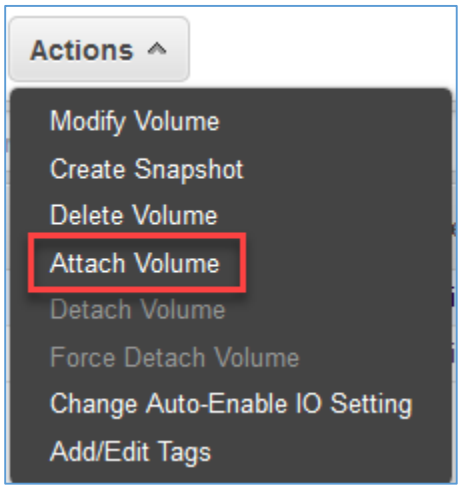
Remember maximum size of EBS volume is 16TB, Select appropriate AZ, if you want to create a volume using existing snapshot select from snapshot ID drop down list. Tick mark Encryption to encrypt data inside volume automatically.

Add tags for easy identification later point of time and click **Create Volume**

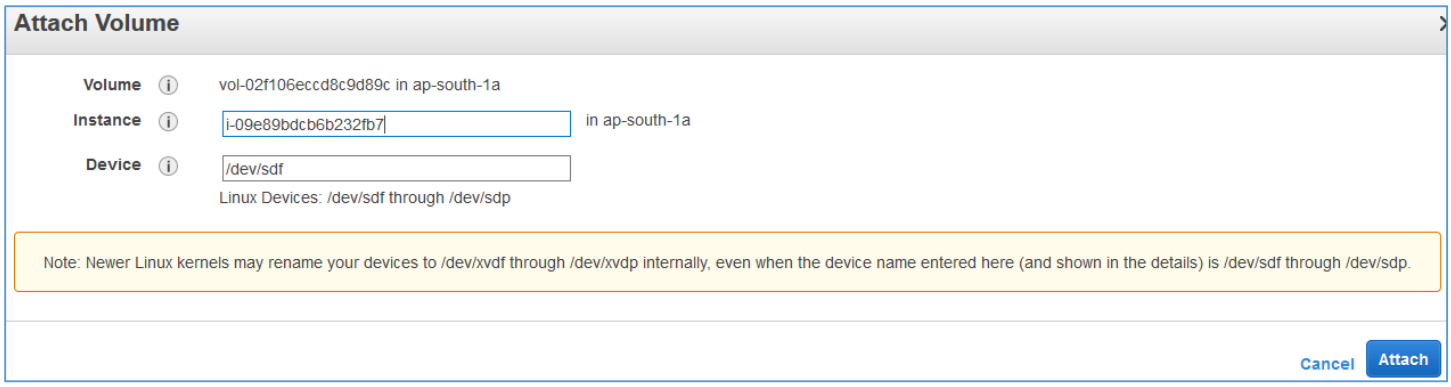


Select created EBS volume to attach to the EC2 instance → Click Actions → Attach Volume





Select instance from drop down list and click attach



Login to instance and see the disk using `fdisk -l` command

```
Disk /dev/xvdf: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

In order to format and create new partition use below commands (shown in screenshot)

```
[root@ip-172-31-28-41 ~]# fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x3dd167db.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-16777215, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-16777215, default 16777215):

Created a new partition 1 of type 'Linux' and of size 8 GiB.

Command (m for help): wq
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@ip-172-31-28-41 ~]# partprobe /dev/xvdf
[root@ip-172-31-28-41 ~]# mkfs.ext4 /dev/xvdf1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
524288 inodes, 2096896 blocks
104844 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2147483648
64 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
[root@ip-172-31-28-41 ~]# mkdir /newpart
[root@ip-172-31-28-41 ~]# mount /dev/xvdf1 /newpart
[root@ip-172-31-28-41 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        476M     0  476M   0% /dev
tmpfs           493M     0  493M   0% /dev/shm
tmpfs           493M  392K  493M   1% /run
tmpfs           493M     0  493M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.2G  6.9G  15% /
tmpfs            99M     0   99M   0% /run/user/1000
/dev/xvdf1      7.8G   36M  7.3G   1% /newpart
```

Successfully created EBS Volume and attached to Linux Ec2 instance.

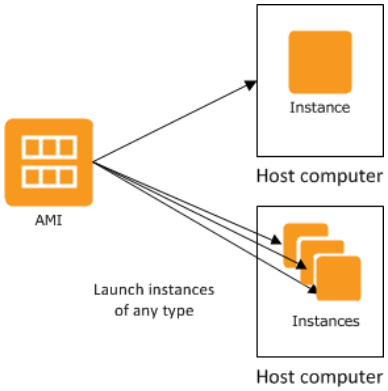
```
[root@ip-172-31-28-41 ~]# umount /newpart
[root@ip-172-31-28-41 ~]# fdisk /dev/xvdf

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): p
Disk /dev/xvdf: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x3dd167db

Device     Boot Start      End  Sectors Size Id Type
/dev/xvdf1       2048 16777215 16775168   8G 83 Linux

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): wq
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

8. Creating Amazon Machine Image (AMI)

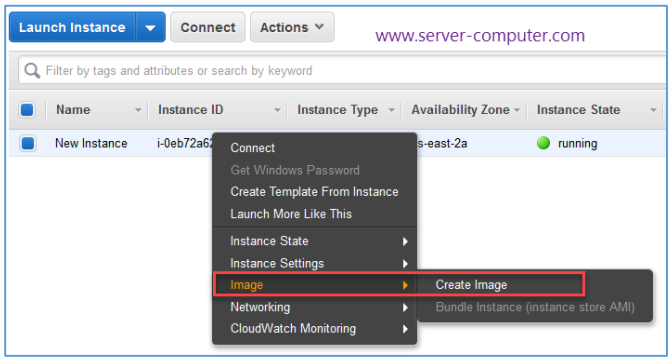
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You must specify a source AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration. You can use different AMIs to launch instances when you need instances with different configurations.

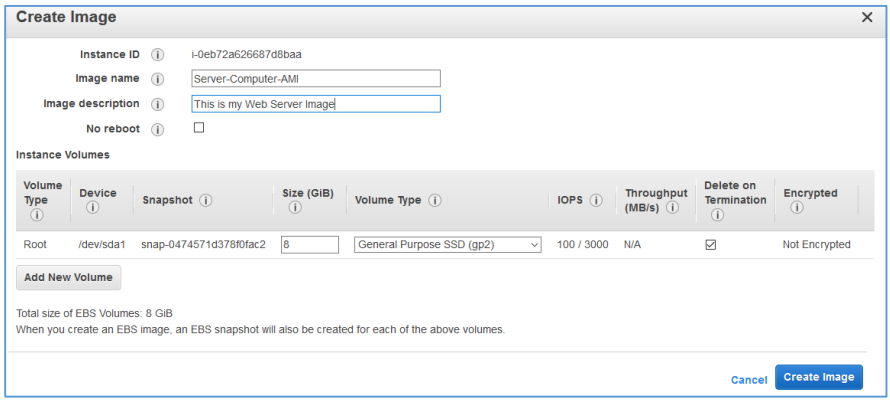
An AMI includes the following:

- 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



First, follow above steps to create EC2 instance, modify all the required settings, and install required applications. Right click on instance **Image** →**Create Image**





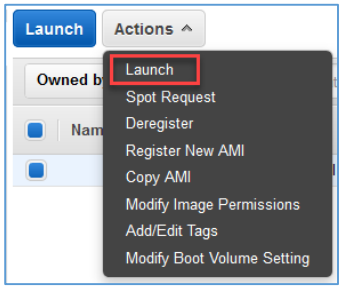
Provide Image name (Easy to Identify), Image Description and Click **Create Image**

It will take few minutes depends on your EC2 instance size.

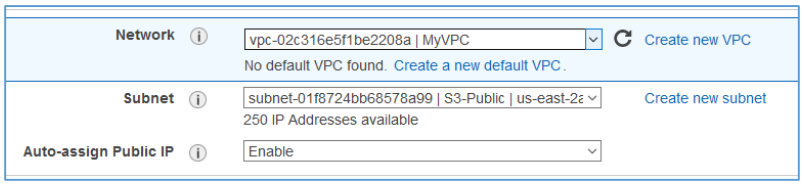
Go to → EC2 → AMIs

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status
	Server-Computer-AMI	ami-03be1b9f43f8b067e	685992403869/S...	685992403869	Private	available

Select AMI → Actions → Launch



Choose Instance Type → Click Next: Configure Instance Details

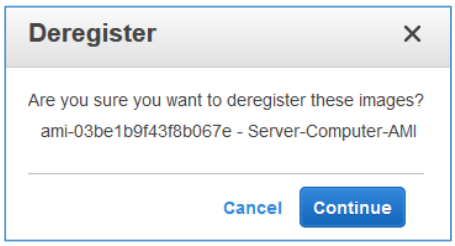


Select appropriate details Click **Next: Add Storage** → **Next: Add Tags** → **Next: Configure Security Group** → **Review and Launch** → **Launch**

That is it your application is ready to use.

Note: Storing AMI will be charged based on your EC2 instance size.

To delete the AMI select AMI → Actions → Deregister

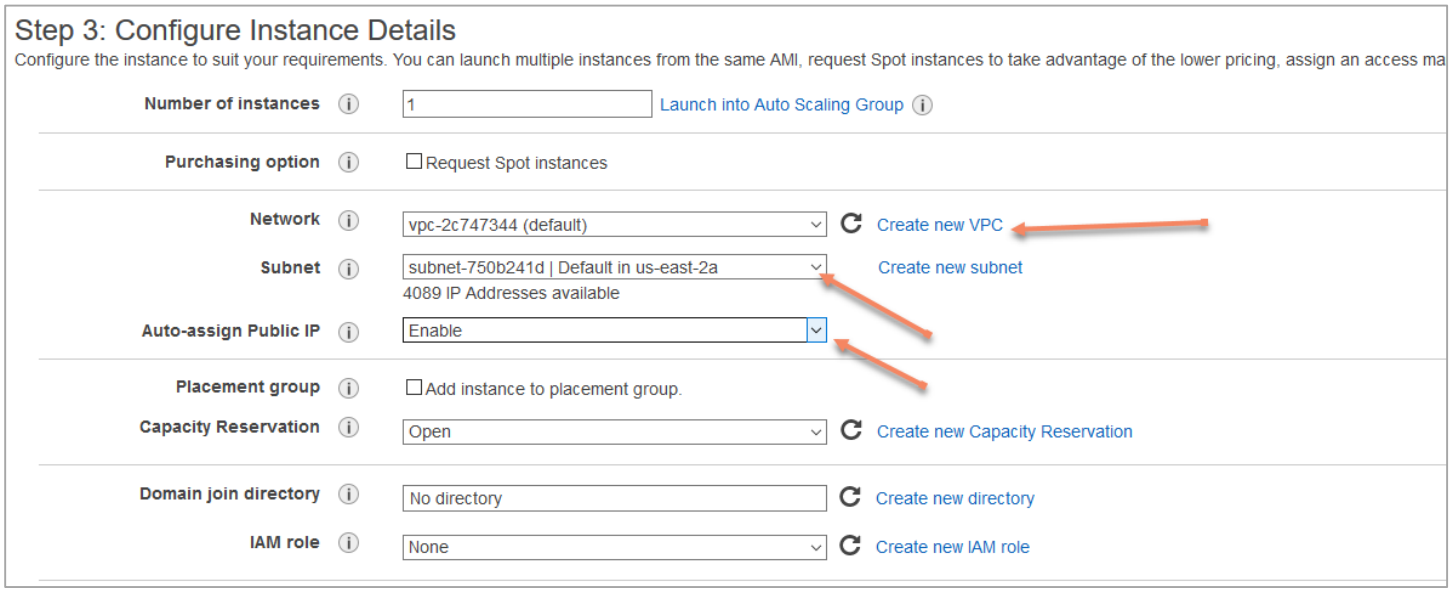


9. Create your First EC2 windows instance

Expand services EC2 → Launch Instance

Select Windows Image

Choose an Instance Type → General Purpose (t2.micro) → Click **Next: Configure Instance Details** →



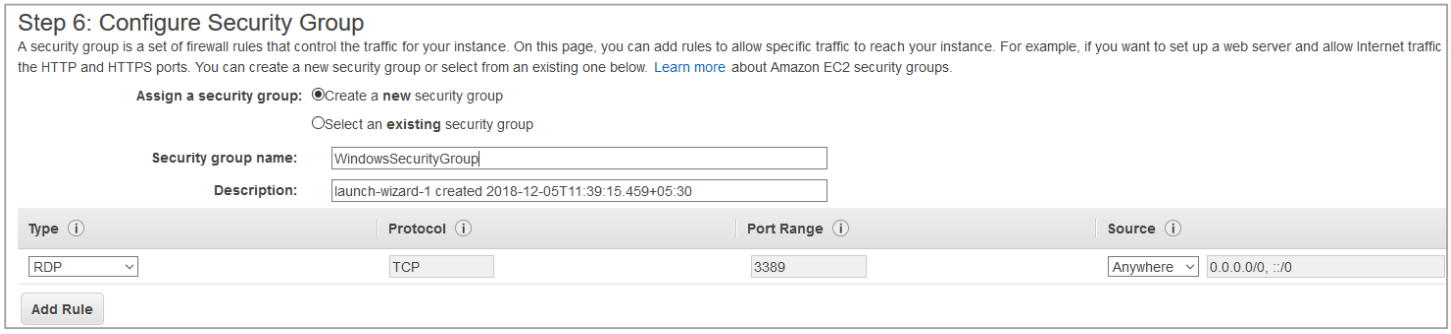
Select VPC, subnet and enable Public IP address.

Click **Next: Add Storage**

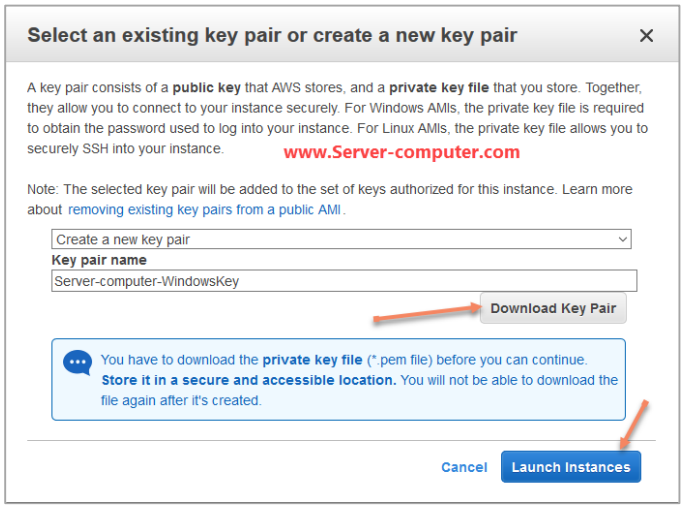
Click **Next: Add Tags**

Add Tags to identify instance details Like Name, Purpose, Account and so and so

Click **Next: Configure Security Group**



Click **Review and Launch**

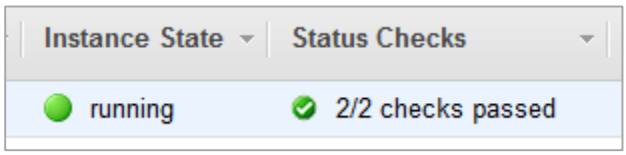


Download Key Pair and **Launch Instance**

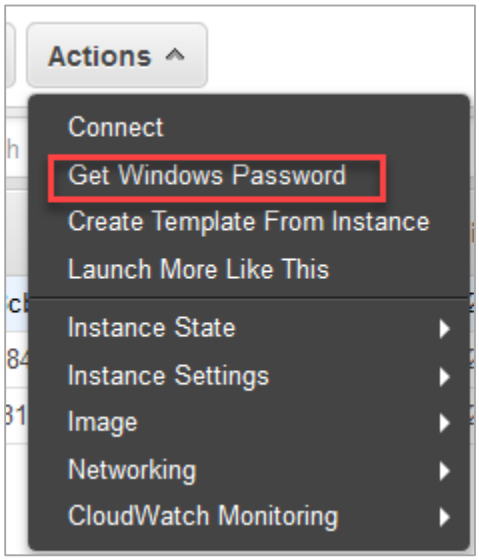
Note: Wait 4 Minutes instance to launch

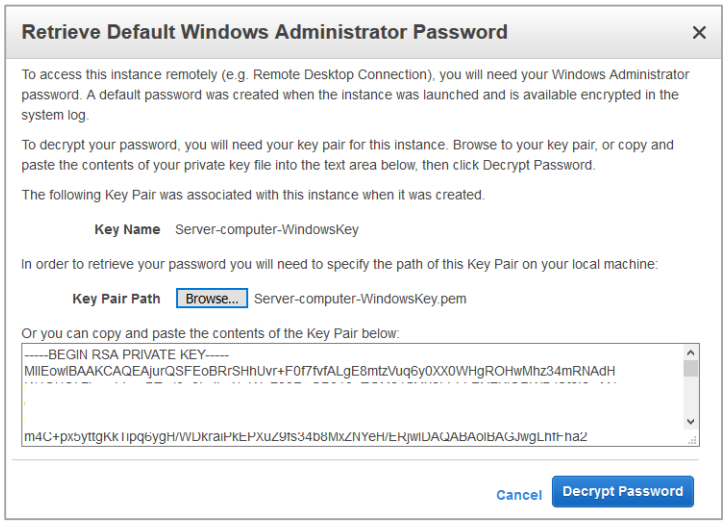
It should display the following:

- Instance State: running
- Status Checks: 2/2 checks passed

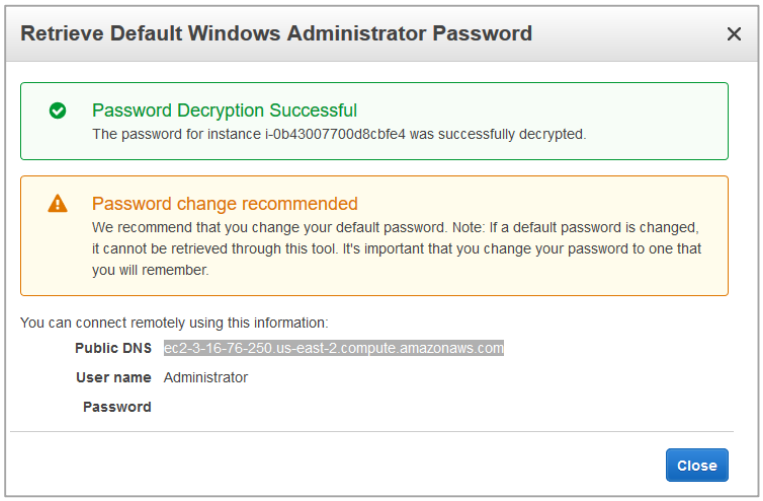


Select instance you have launched → Actions



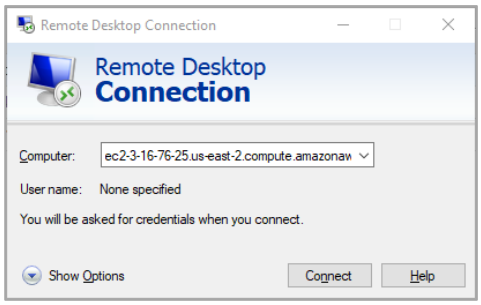


Browse server-computer-WindowsKey.pem file to decrypt and get password



Now you got password successfully. Click **Close**.

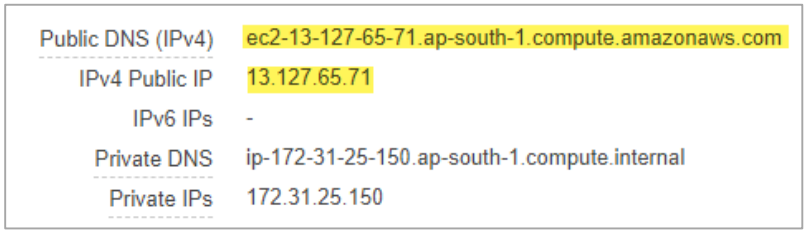
Go to your windows machine Start → Run → mstsc → Ok



Click **connect** and type user name and password you are connected to your EC2 windows instance.

10.Assigning Elastic IP Addresses to Instance (Static IP Address)

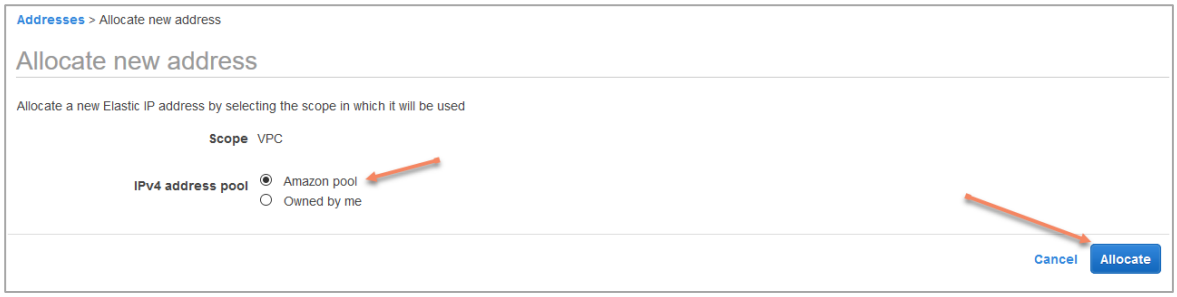
Click on instance name and see instance details like Internal and external IP Address, Host name



However, after stop and start of instance assigned public IP address will release to the amazon free pool

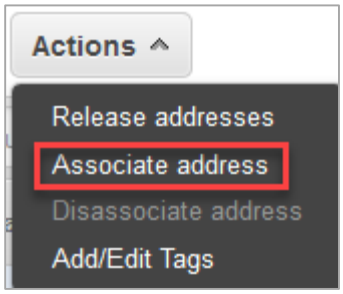
If would like to assign an static public address then navigate to Elastic IP's

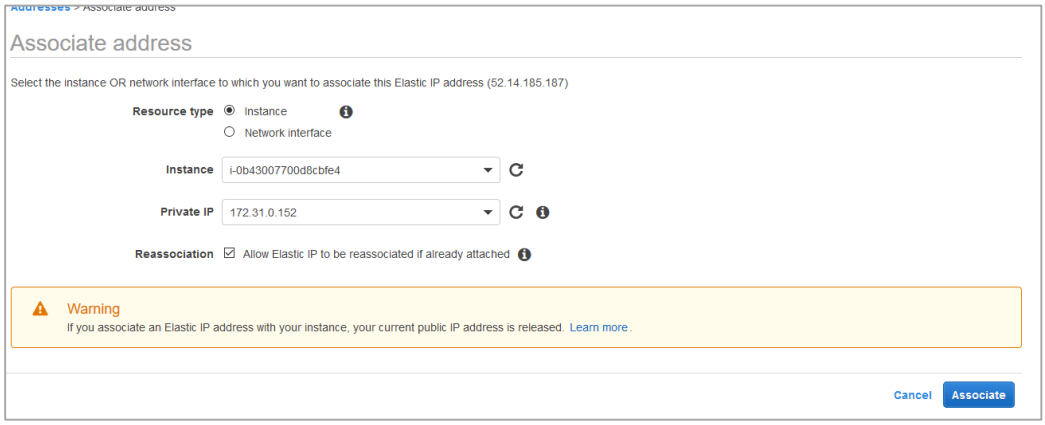
EC2 console right side bar go down → Elastic IPs → Allocate New Address



Click **Allocate**. Amazon allocate you static IP address

Select the IP from Elastic IPs console → Actions → Associate Address





Select Instance ID check Instance ID before allocating. Click **Associate**

Note: If you have, multiple interfaces to the instance click on Radio button **Network Interface** and select correct NIC card name and Local IP Address.

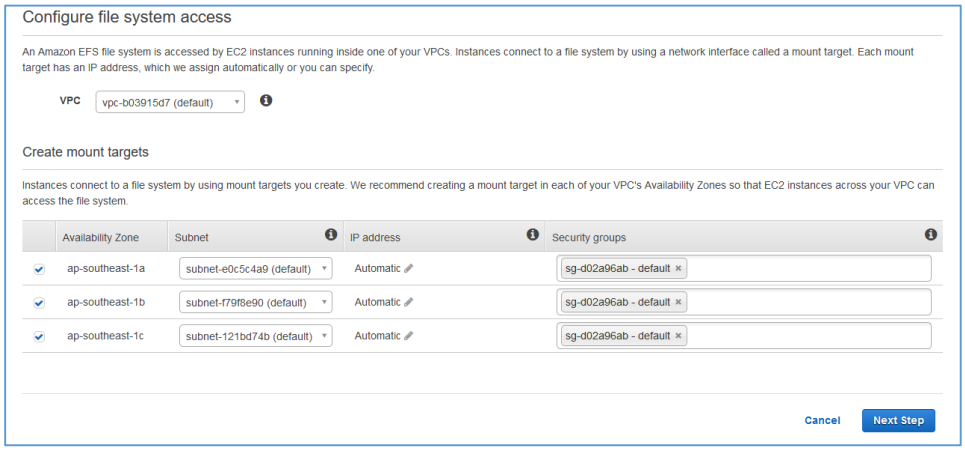
Now your existing instance has static Public IP address, if you restart your instance also you will get same IP address until you detach from instance.

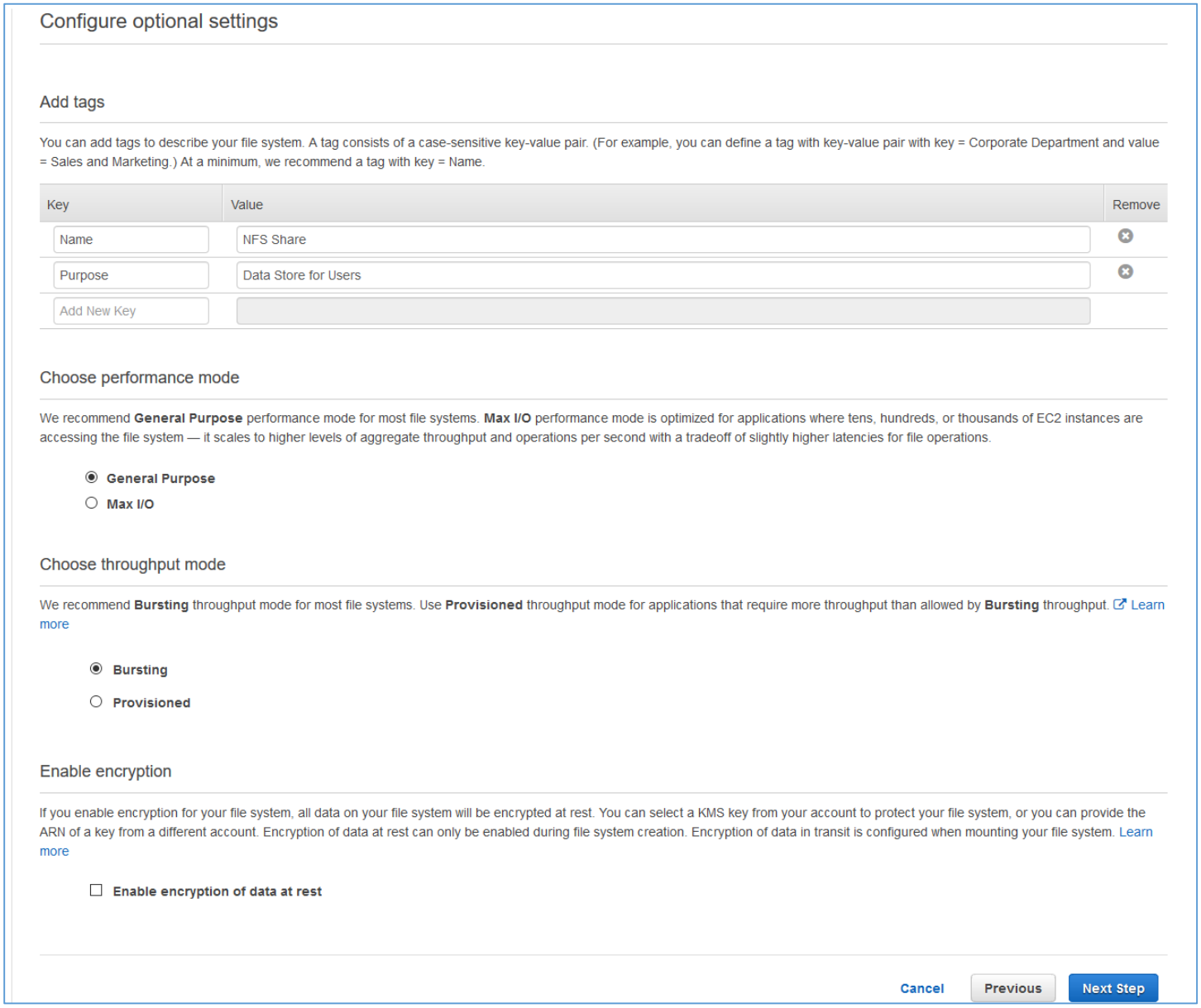
11. Amazon Elastic File System

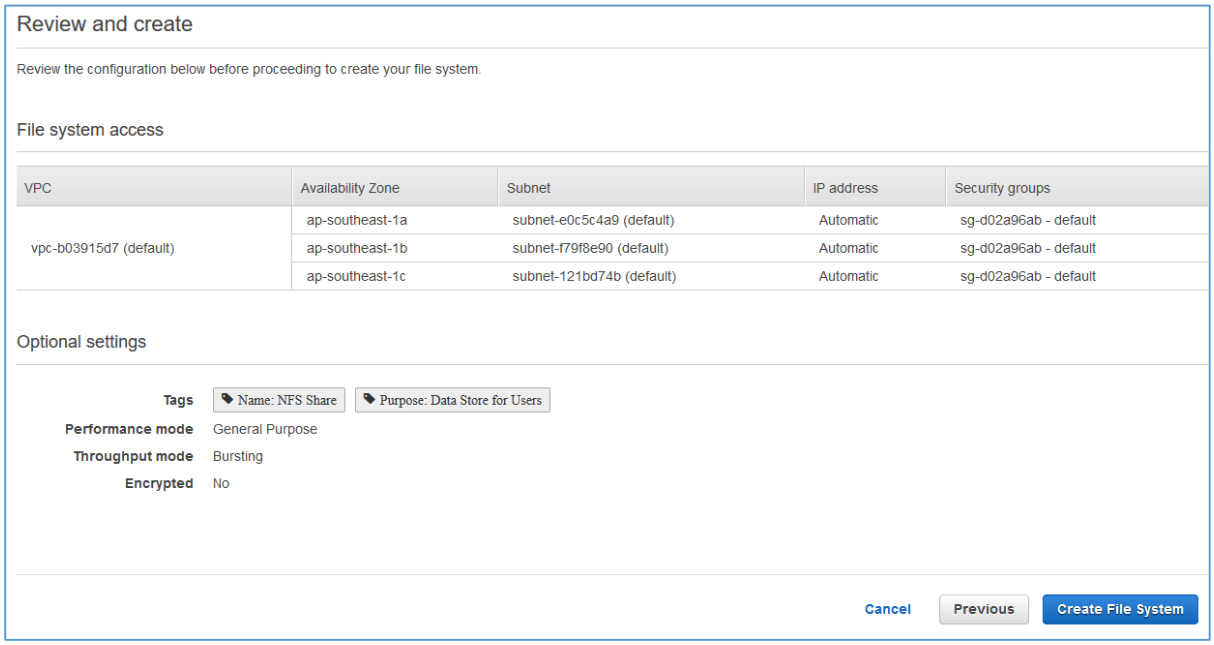
Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it. Amazon EFS has a simple web services interface that allows you to create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

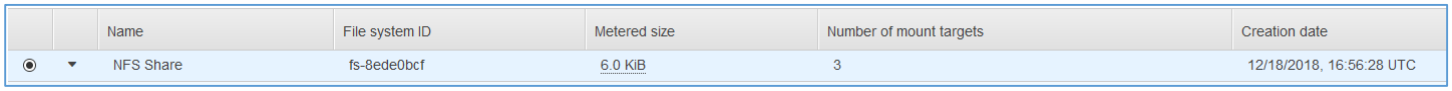
Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with Amazon EFS. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance or server.

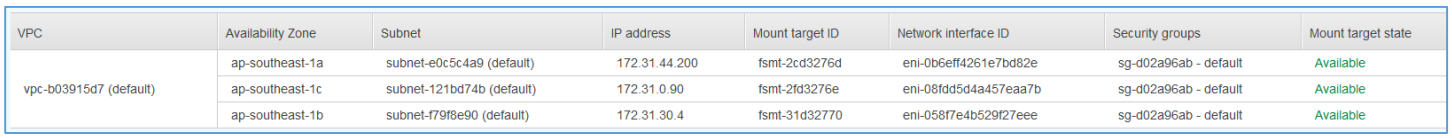
Services → EFS











Login to EC2 Linux instance and mount EFS using below commands

```
# sudo yum -y install nfs-utils*
```

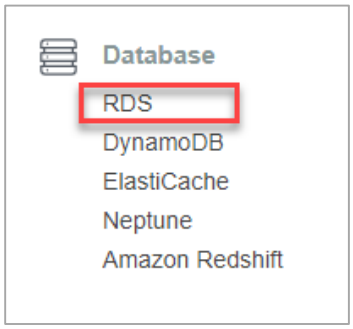
```
# sudo mount -t nfs4 IP_ADDRESS_OF_EFS:/ MOUNTPOINT
```

That's it about EFS.

12.Launching RDS Instance

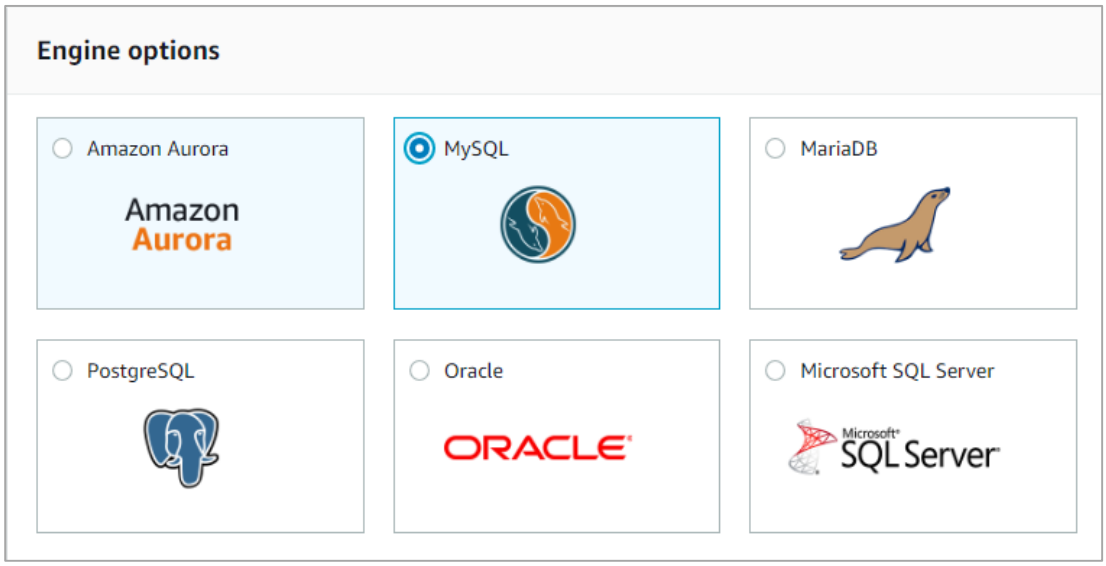
Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Login to **AWS Console** and Click on **services** to list all services. Navigate to **Database → RDS**



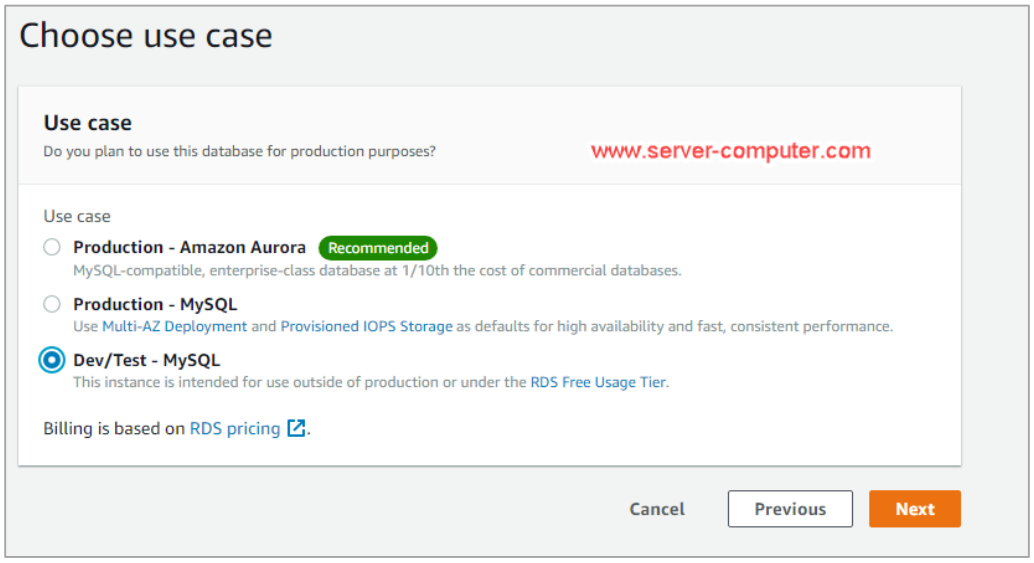
Now we are going to create a new Database instance with empty database

Amazon will support below 5 types of Relational database engines as managed services

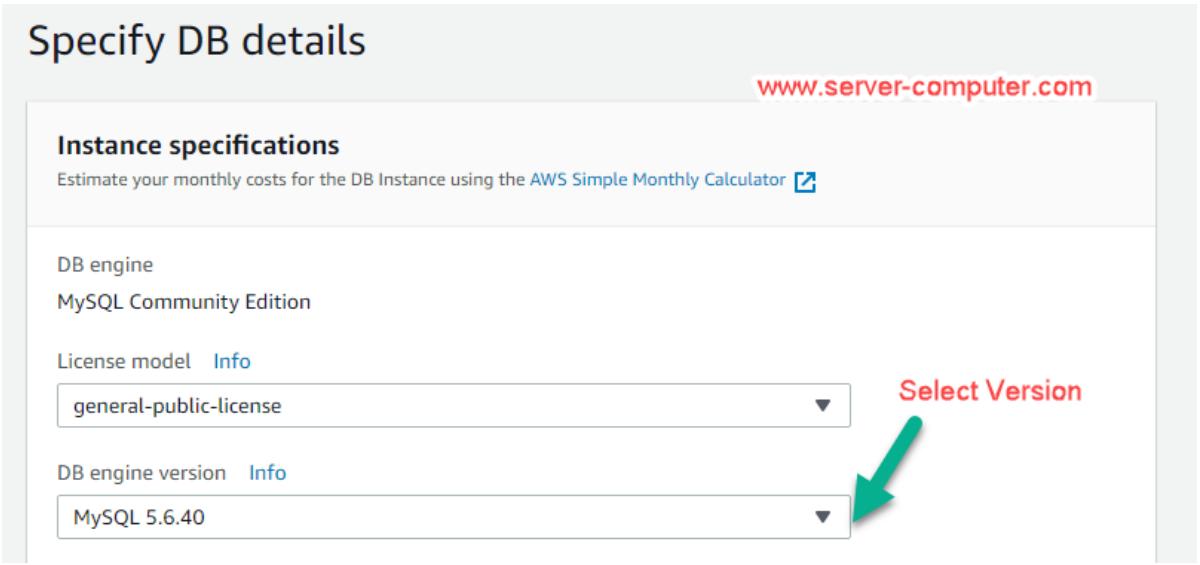


Select any one of the database engine, which you want to launch and Click **Next**

Note: Careful if you are using free tier account. MSSQL and Oracle are charged.

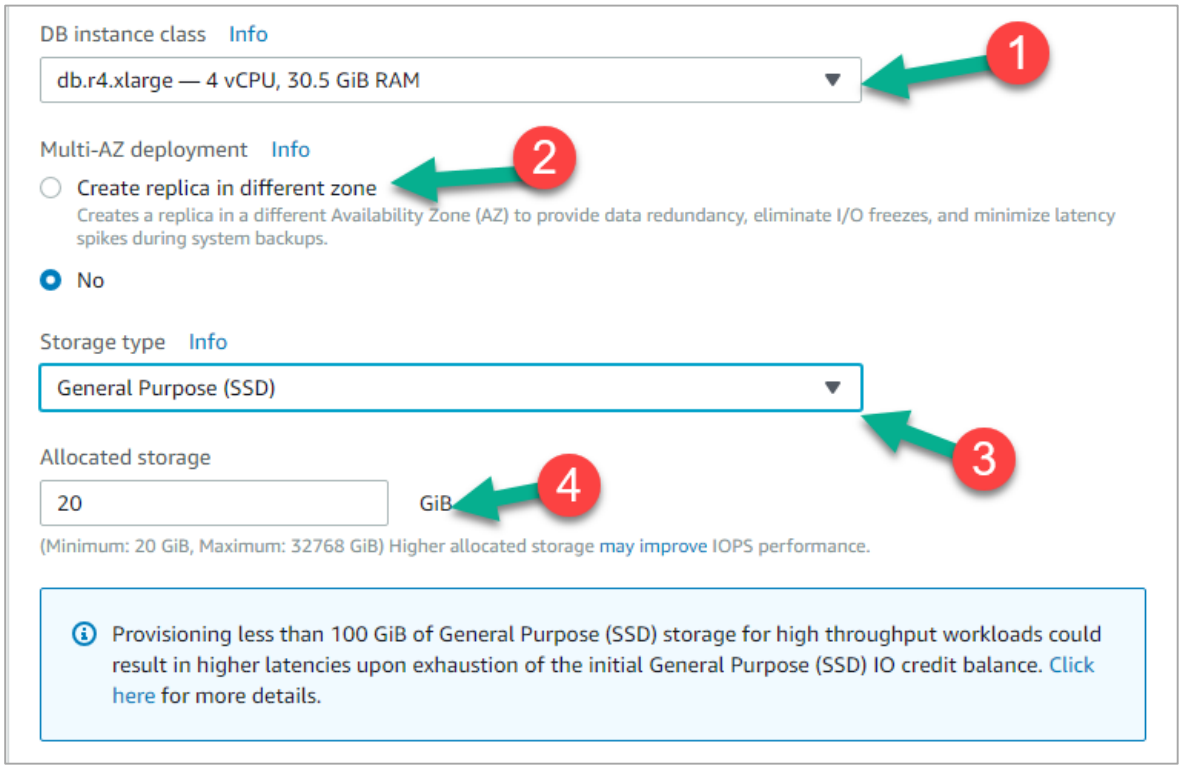


Choose appropriate usage of your instance. In this scenario, I am using Dev/Test instance Click **Next**

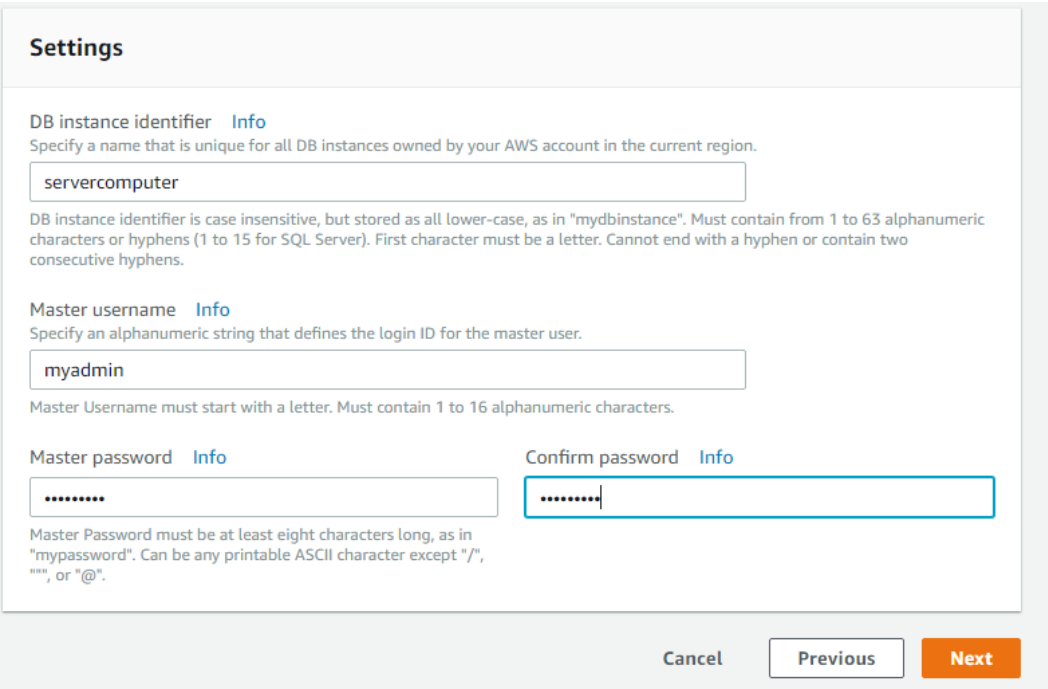


In drop down, select appropriate and required MySQL Version.

Note: If you select Free Tier. Selected version and options will overwritten free options.

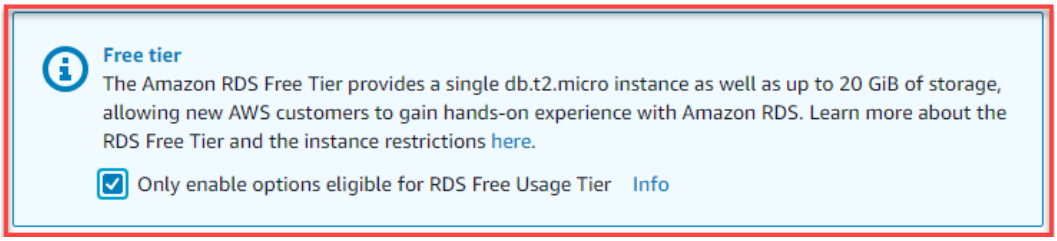


1. Select DB Instance class like required CPU Cores and RAM.
2. Create Replica in Different Zone. (Which means database will be replicated to another available zone for redundant(data protection))
3. General purpose (SSD) or provisioned IOPS (SSD)
    a. General purpose is for low through put applications
    b. Provisioned IOPS is for most read/write operations
4. Size of the storage

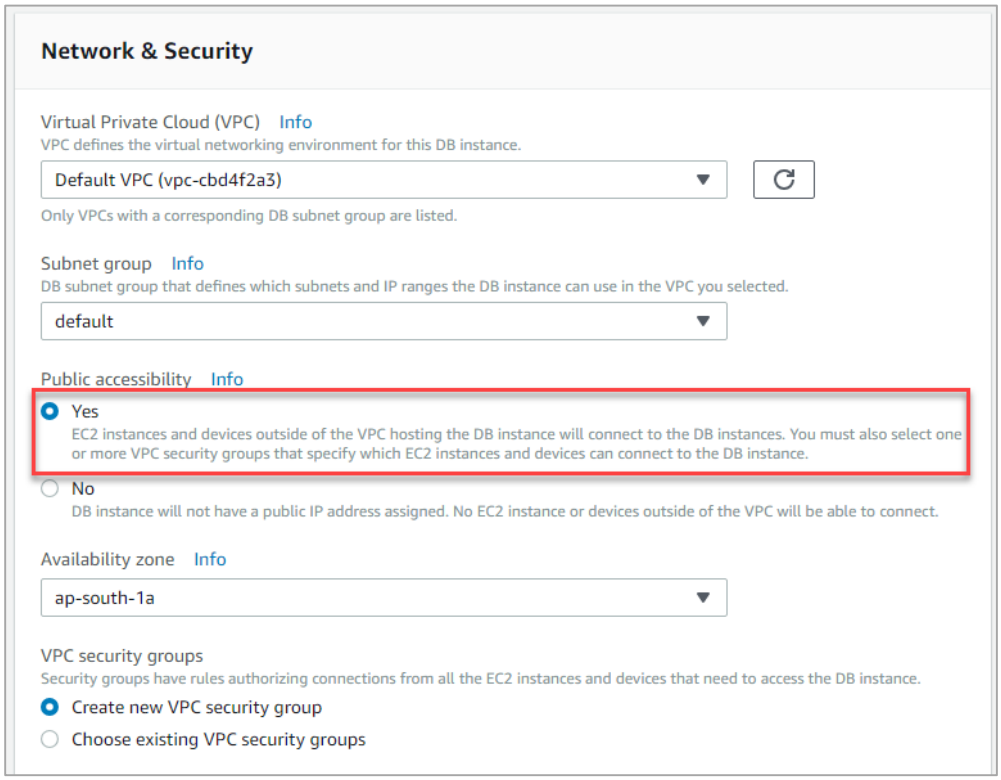


Provide

- Instance name should be unique
- Master username anything you can give without special characters
- Provide master password and remember



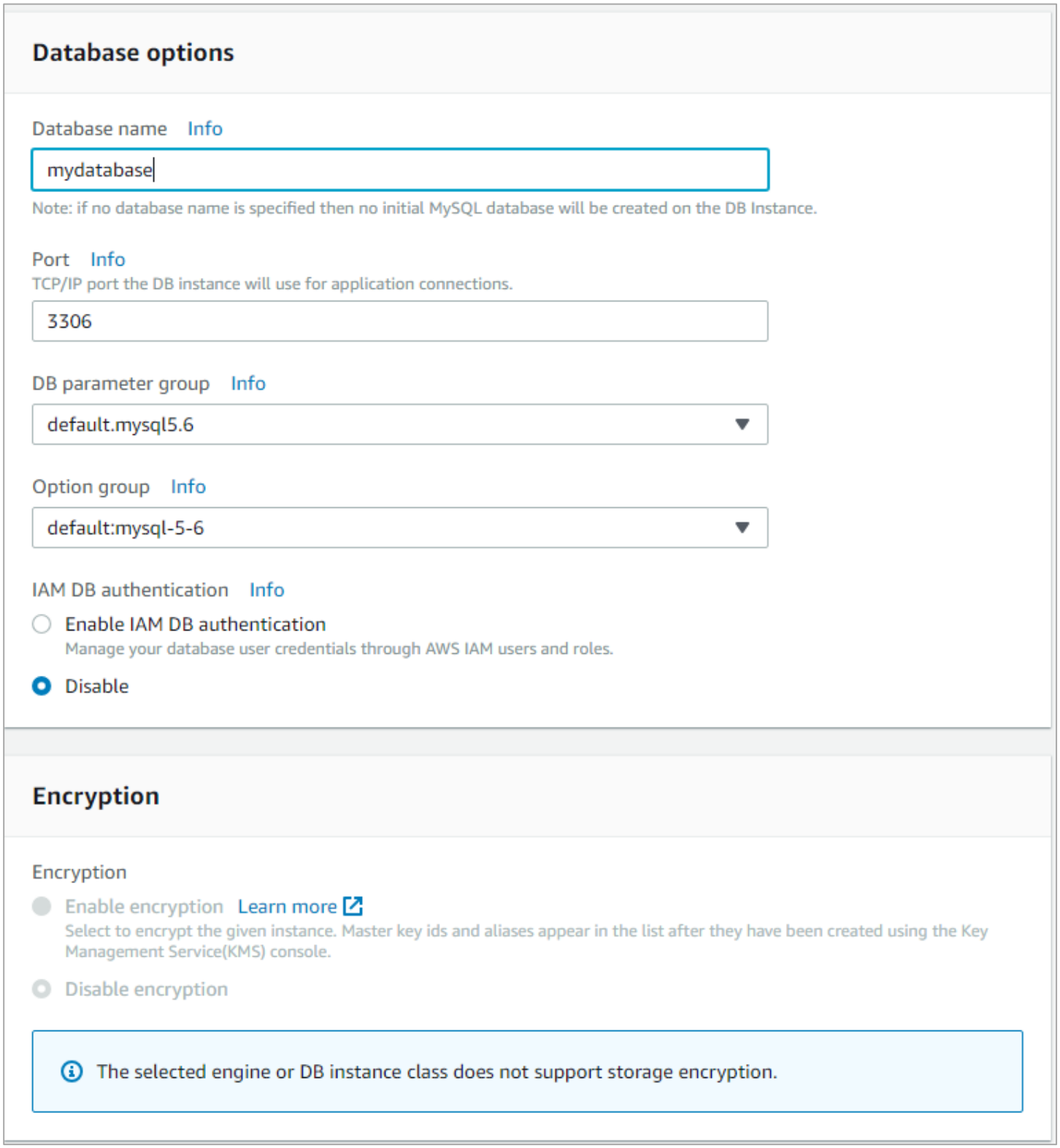
DO NOT FORGOT TO SELECT IF YOU'RE USING FREE TIER OTHERWISE YOU WILL BE CHARGED



Select appropriate VPC and Subnet group (If any)

If you want access database from remote machine put “Public Accessibility” **Yes**

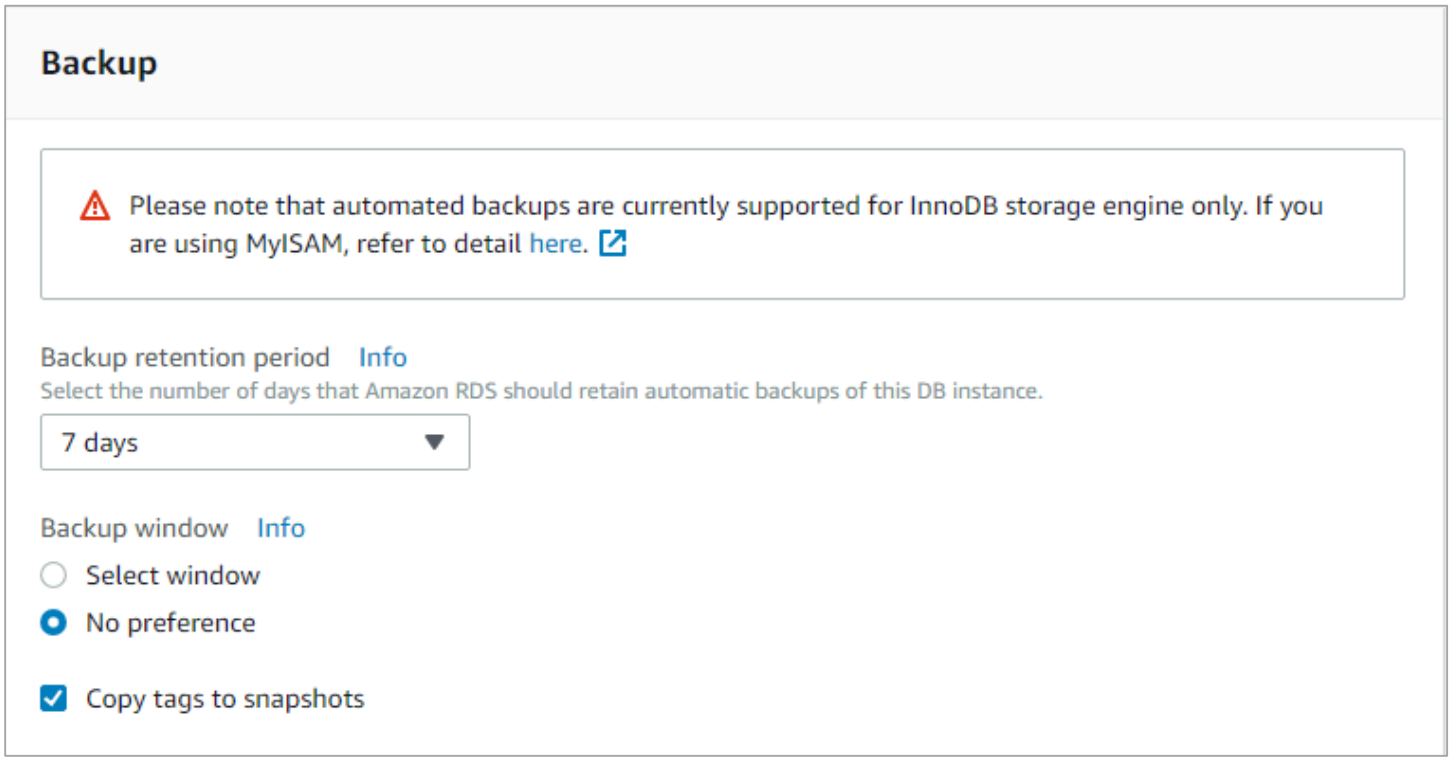
Choose existing VPC security groups if you have already or it will create new security group for this instance access.



Provide database name, default port number is 3306 you can even customize the port number if you want.

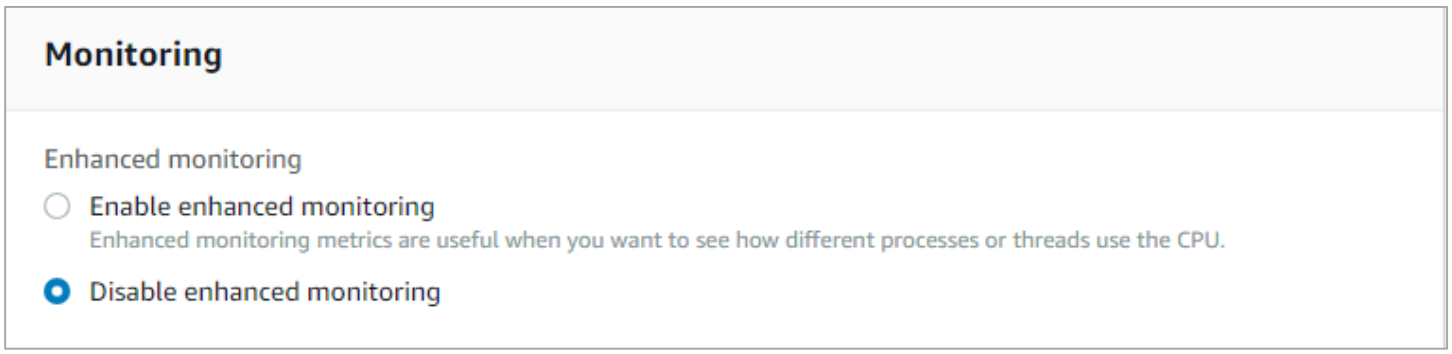
Enabling IAM DB Authentication. IAM Users also can access your instance based on IAM policies.

For free tier encryption option is disabled

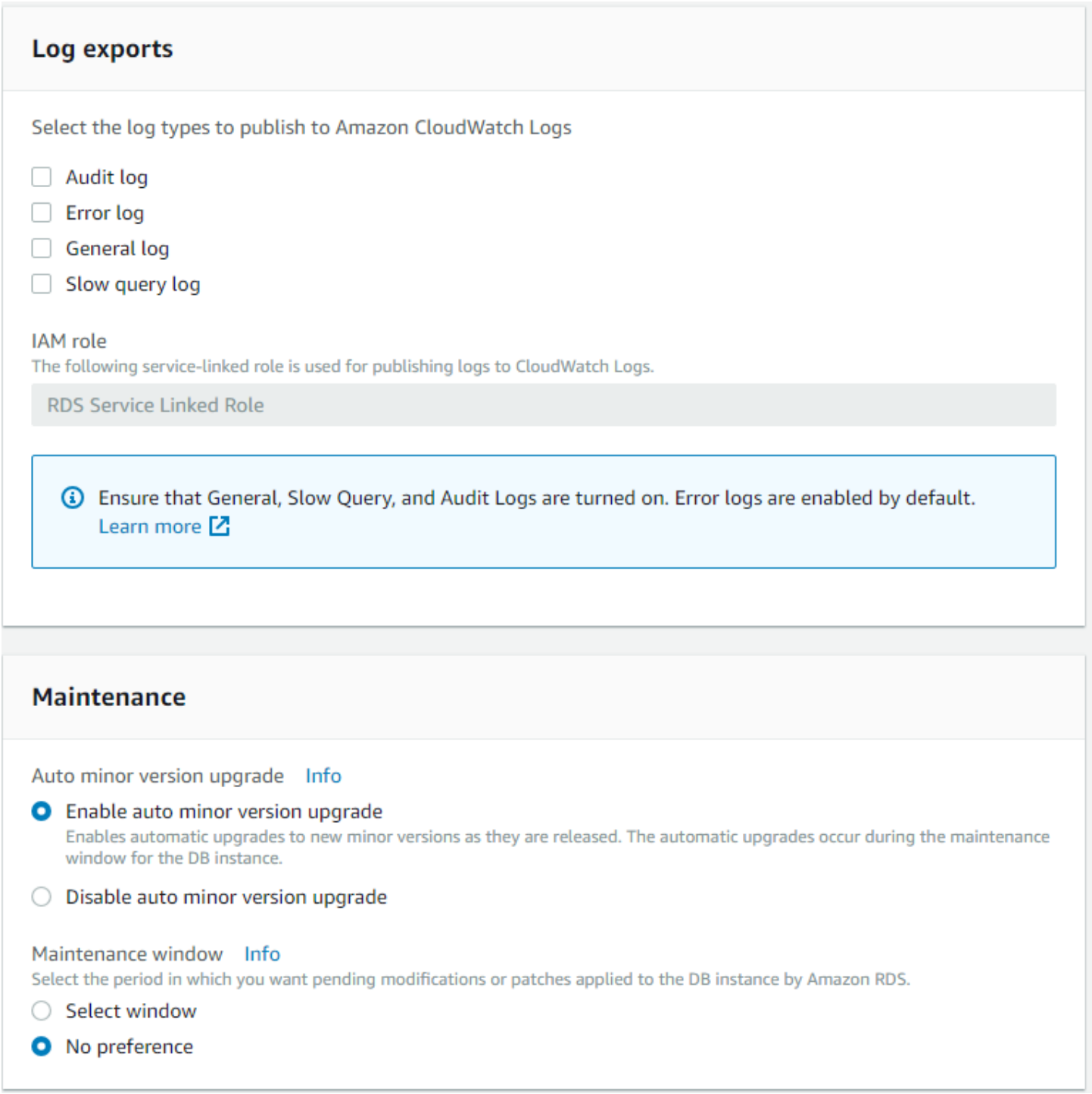


If you want database backups select, the retention max is **35 Days**

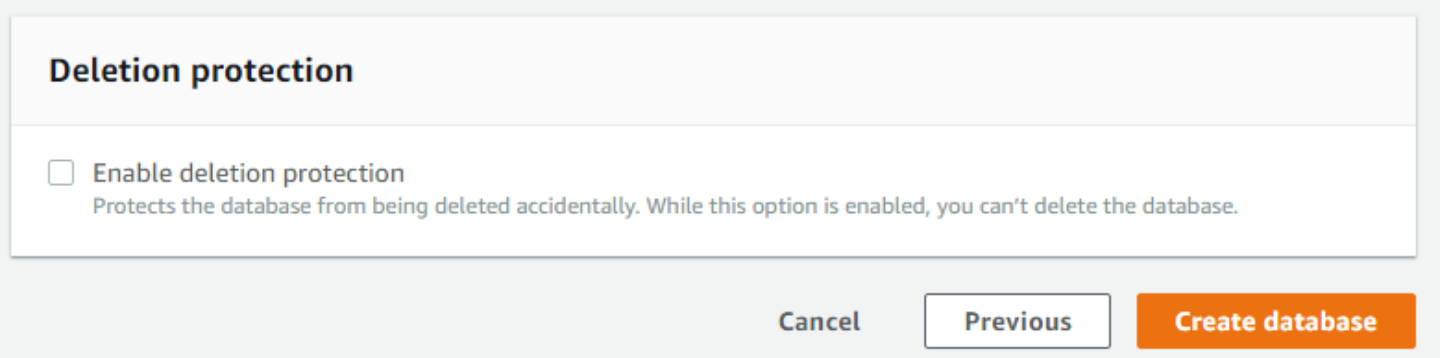
If you have particular backup window for database select it otherwise leave it default.



Enhanced monitoring will charged



Select the options you required



Enabling database protection, you cannot delete database

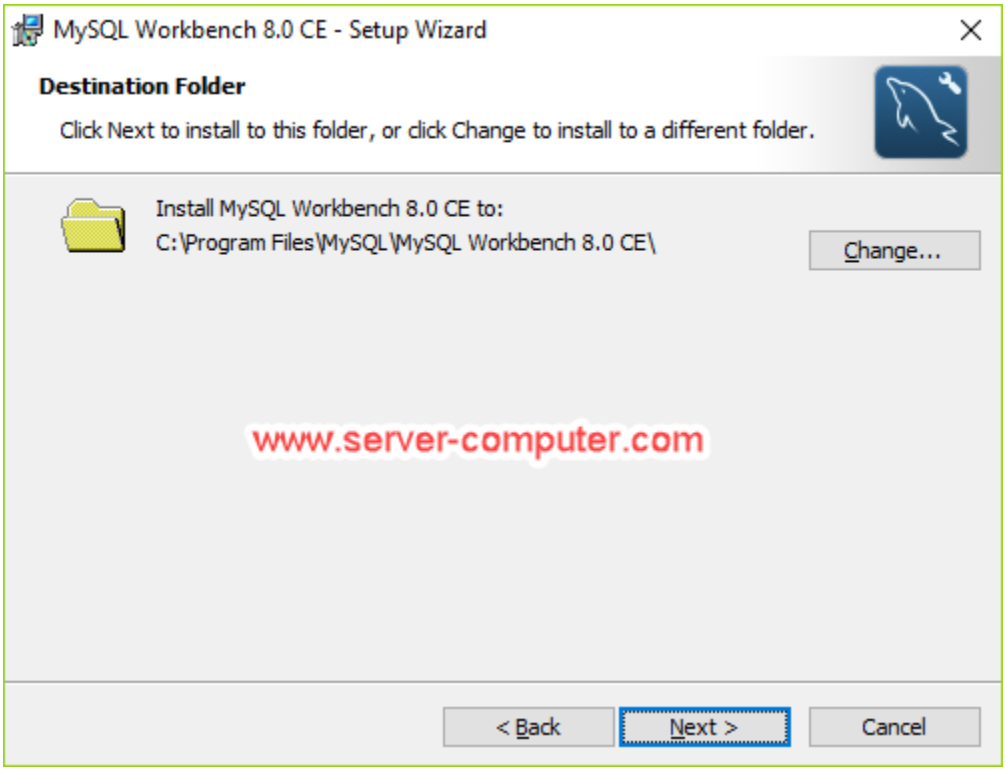
Click **Create Database**

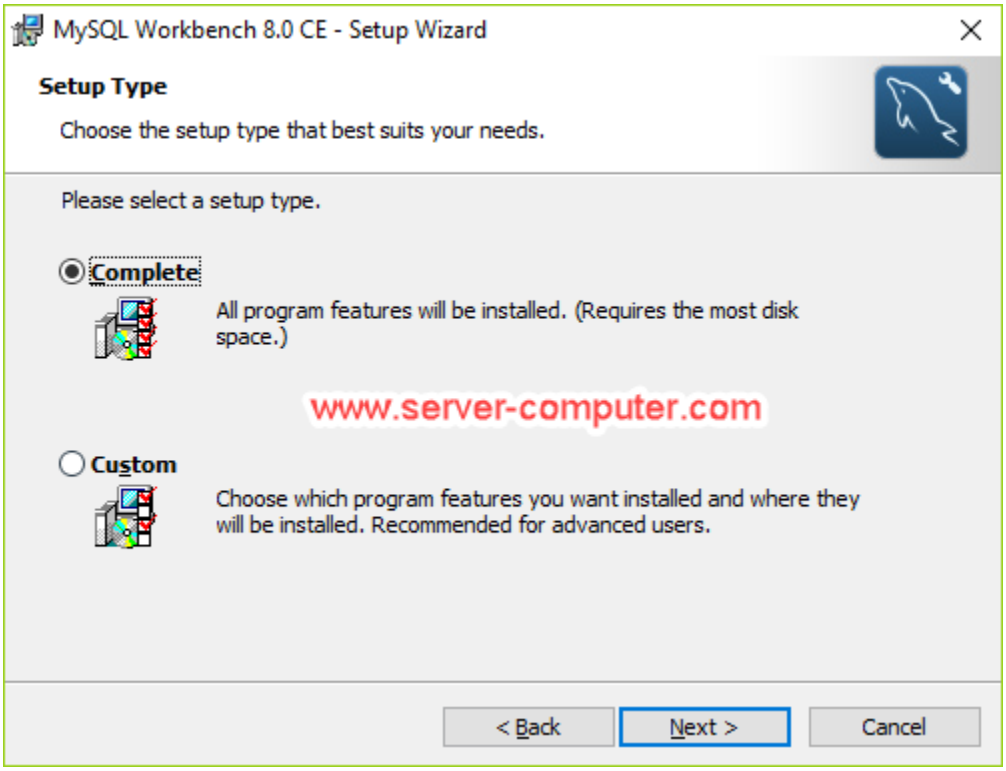
Note: Database instance creation will take at least 10minutes.

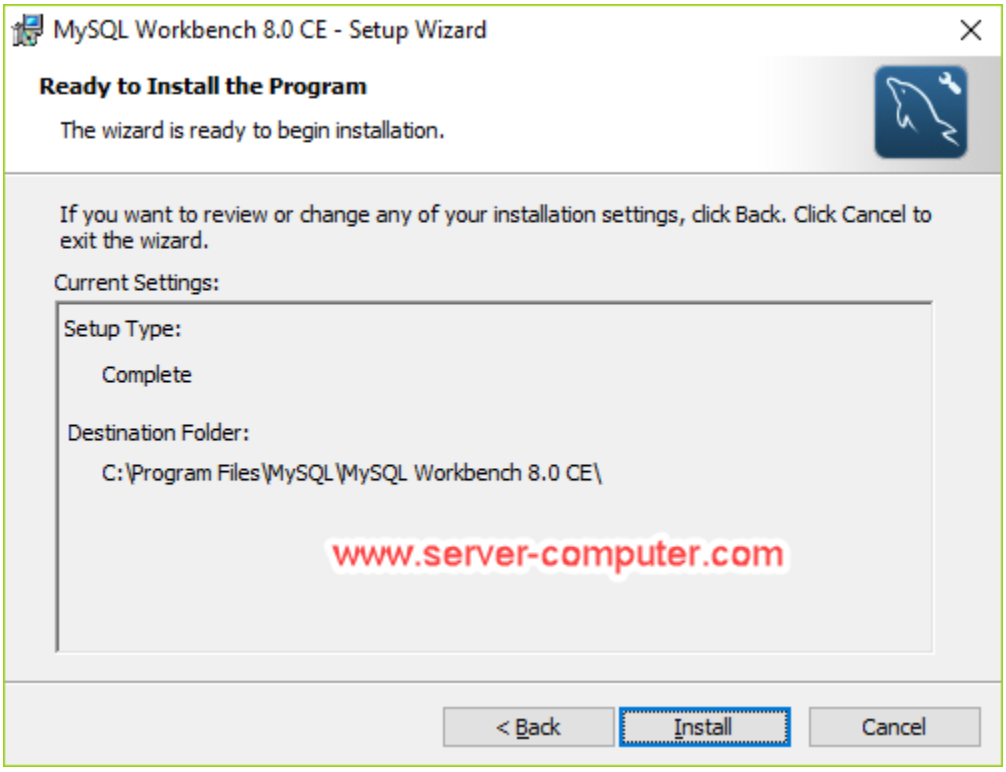
13.Accessing MySQL Instance Using Workbench

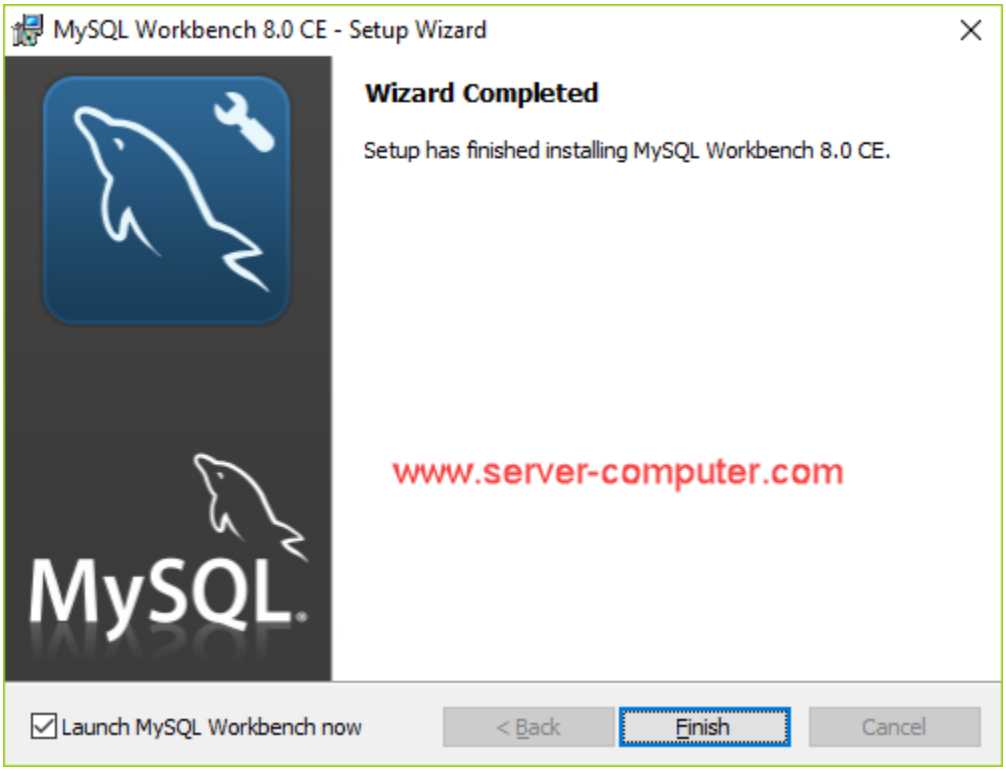
Download MySQL Workbench to access MySQL instance remotely

https://dev.mysql.com/downloads/workbench/

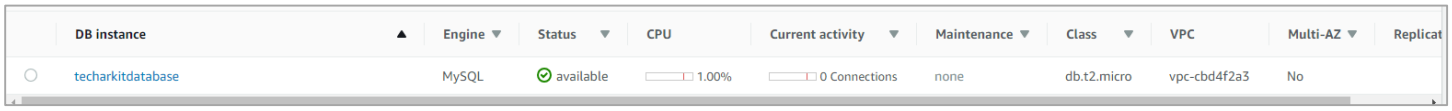








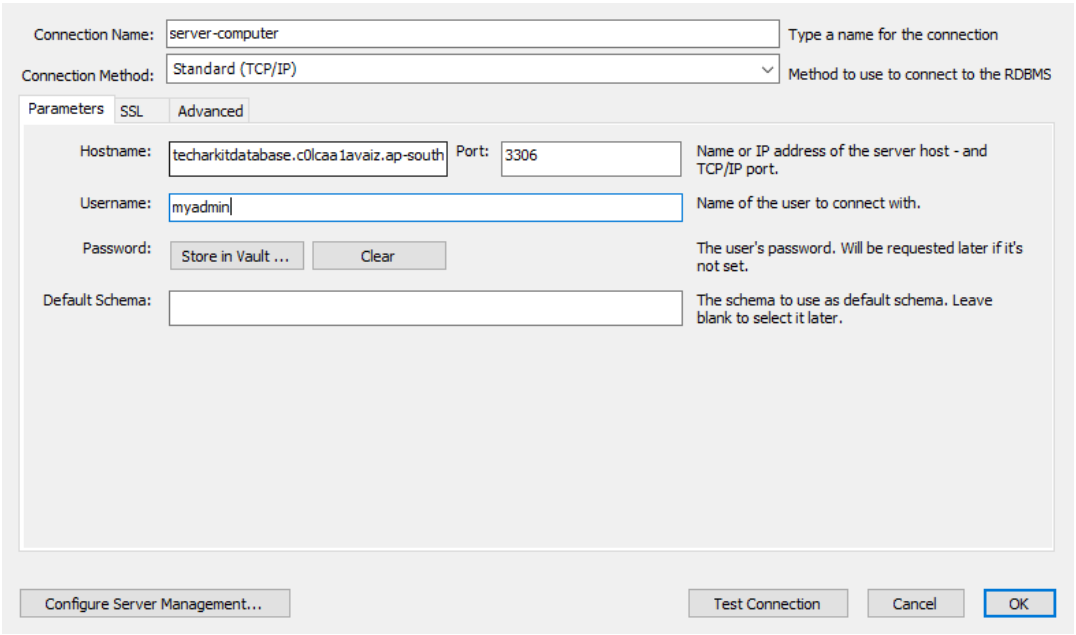
After successful creation you see like below



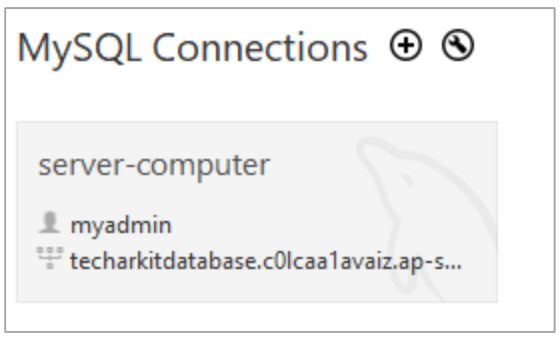
Click on Database name and come down copy the **Endpoint URL**

Open your MySQL workbench and create connection

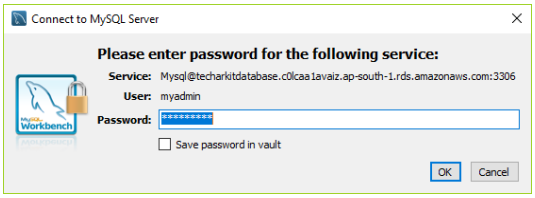
Click on Plus (+) sign to create a New MySQL Connection

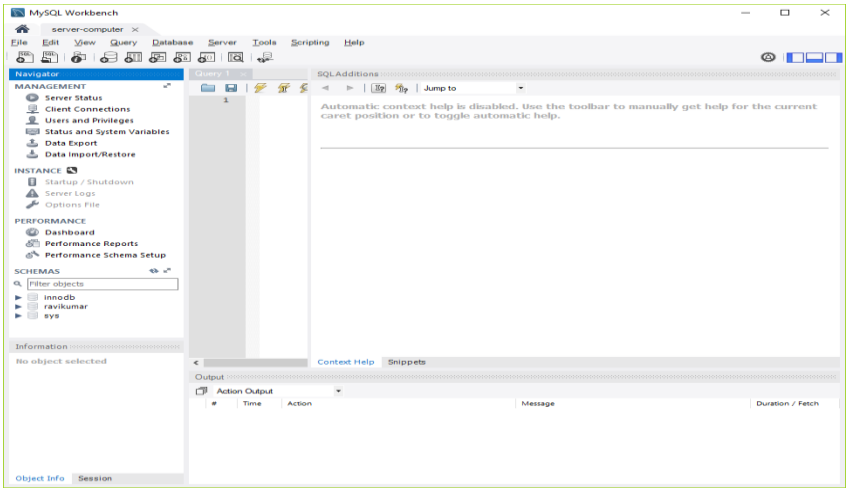


Click **OK**



After successful creation, Click on Connection it will ask you for the password





Successfully launched MySQL RDS Instance and accessed via MySQL Work bench.

Run below queries to create database and some tables on it.

```
create database 'DBNAME';

use DBNAME;
```

Create Table using below query

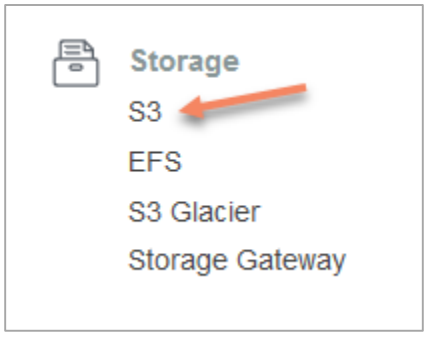
```
create table students(
   student_id INT NOT NULL AUTO_INCREMENT,
   student_title VARCHAR(100) NOT NULL,
   student_author VARCHAR(40) NOT NULL,
   submission_date DATE,
   PRIMARY KEY ( student_id )
);

show databases;

use DBNAME;

show tables;
```

If you know much more database queries like select, insert and delete statement try doing more. Good Luck.

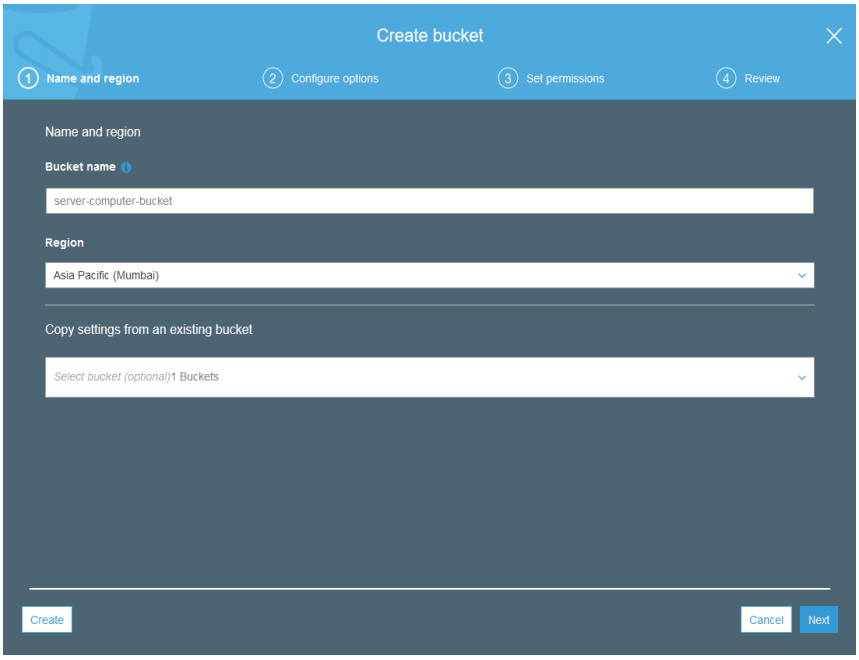
14. AWS S3 Bucket – (Object Storage)

Amazon Simple Storage Service (Amazon S3) is storage for the Internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the AWS Management Console, which is a simple and intuitive web interface.

Login to AWS Console and navigate to **Storage → S3**

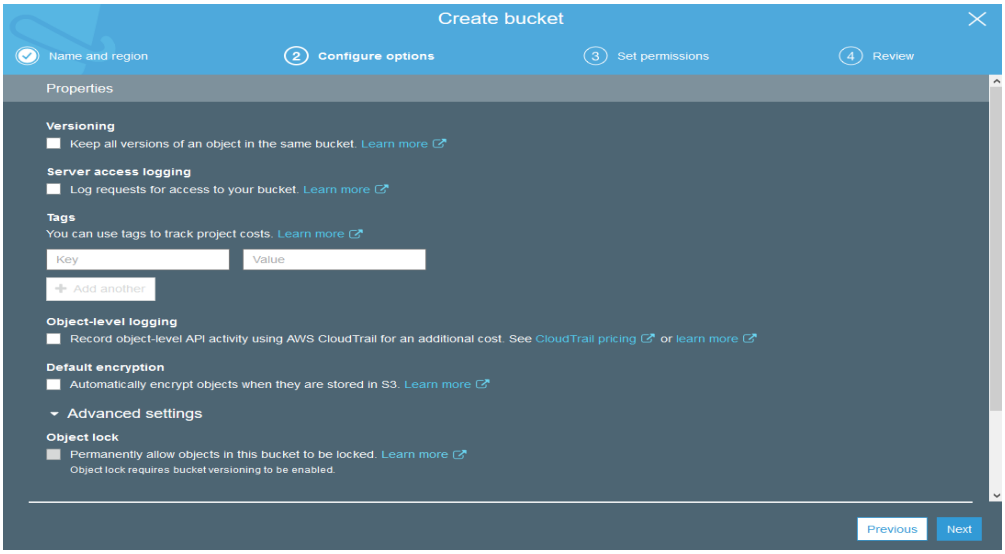


Click on 



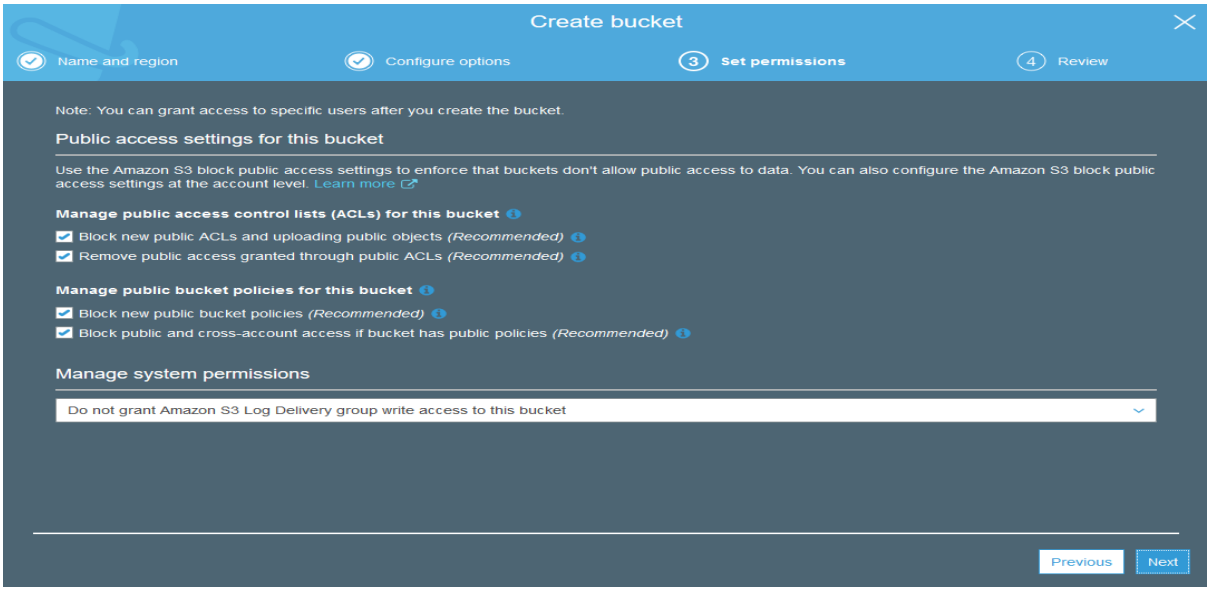
Provide bucket name, it should be a unique name. To Access your S3 bucket over internet it will create DNS entry.

Click **Next**



- **Keep All Version of object** means it will not delete any files if you upload same file multiple times. It will keep all the files as multiple versions
- **Log Requests for access to your bucket** option will log all the actions users did on this particular S3 bucket
- **Object-level Logging** used to monitor all the object level modifications. Additional cost.
- **Encryption** You can encrypt S3 bucket data or Encrypt and upload the data either way your data is encrypted.
- **Object Lock**
- **Cloudwatch request metrics** for monitoring purpose

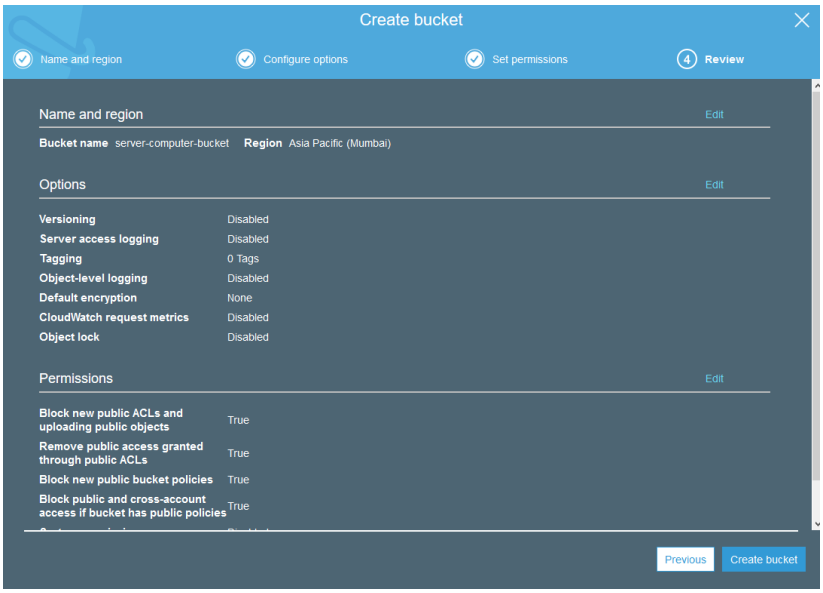
Click **Next**



AWS recent update is to block public access by default, if you want to enable public access to your S3 bucket un-check all above tick marks.

Still you can provide access to other users on bucket level and object level.

Click **Next**



Final Step is to review selected options and Click **Create bucket**

Your S3 bucket created successfully. Click bucket name you will see all the options

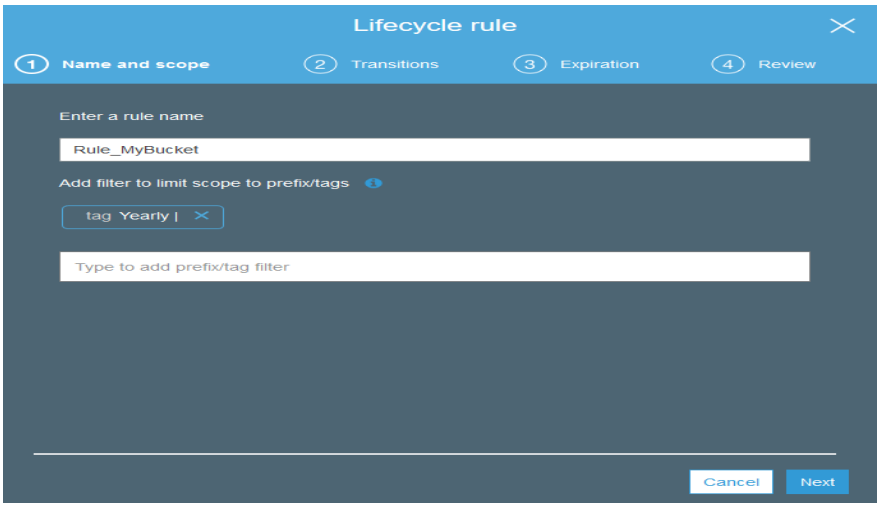
https://s3.ap-south-1.amazonaws.com/server-computer-bucket

Above is the example URL to access your S3 bucket over internet

14.1. AWS S3 Lifecycle Management

Click on **S3 Bucket → Management → Lifecycle**

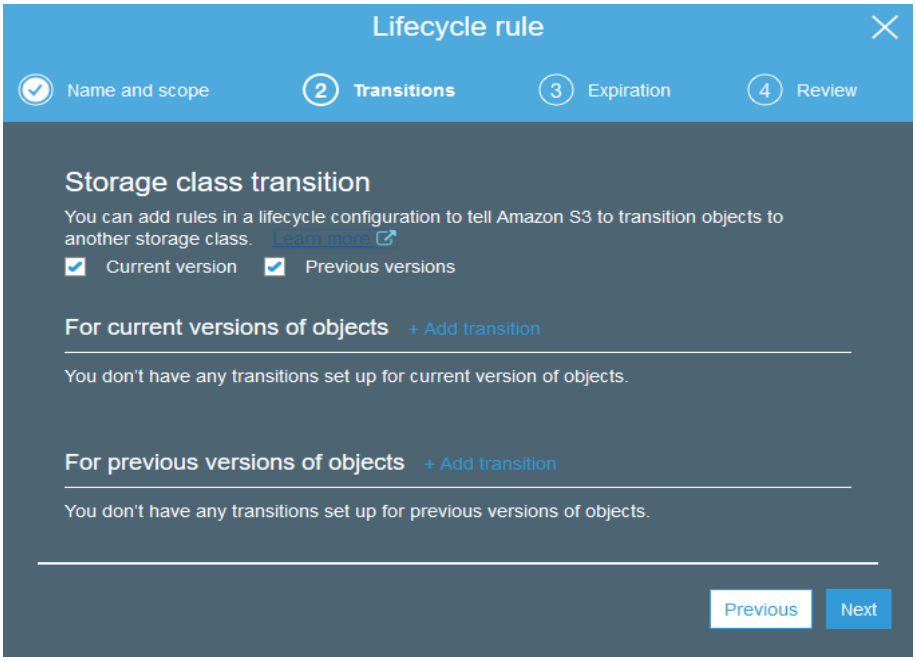
You can manage an objects lifecycle using this feature/rule, which defines



Enter Rule Name

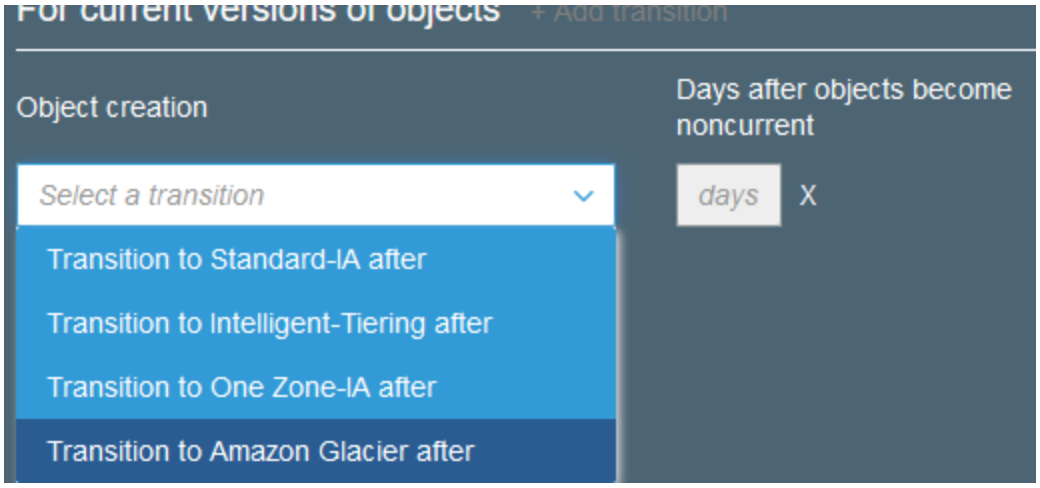
Tag Name if you do not want leave it blank

Click **Next**

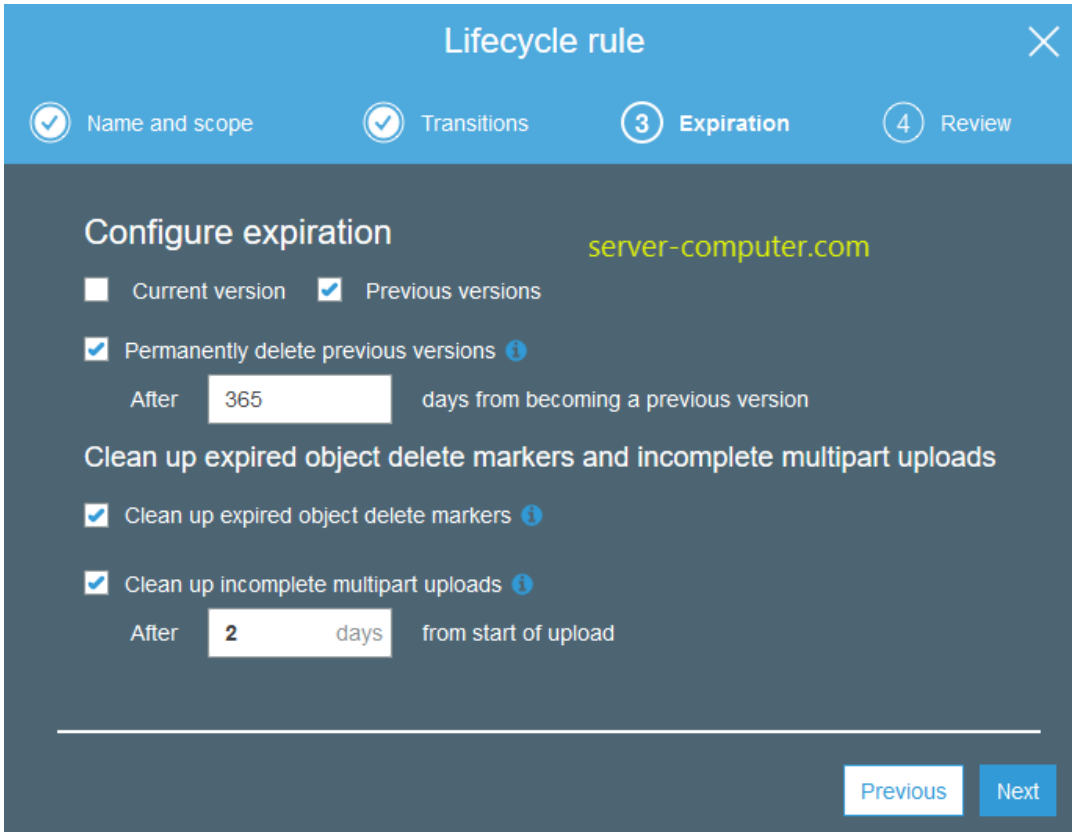


- Current Versions
- Previous Versions

Based on selected versions action will be performed example if you want to keep current versions in A1 or maybe previous versions on Glacier as per your requirement



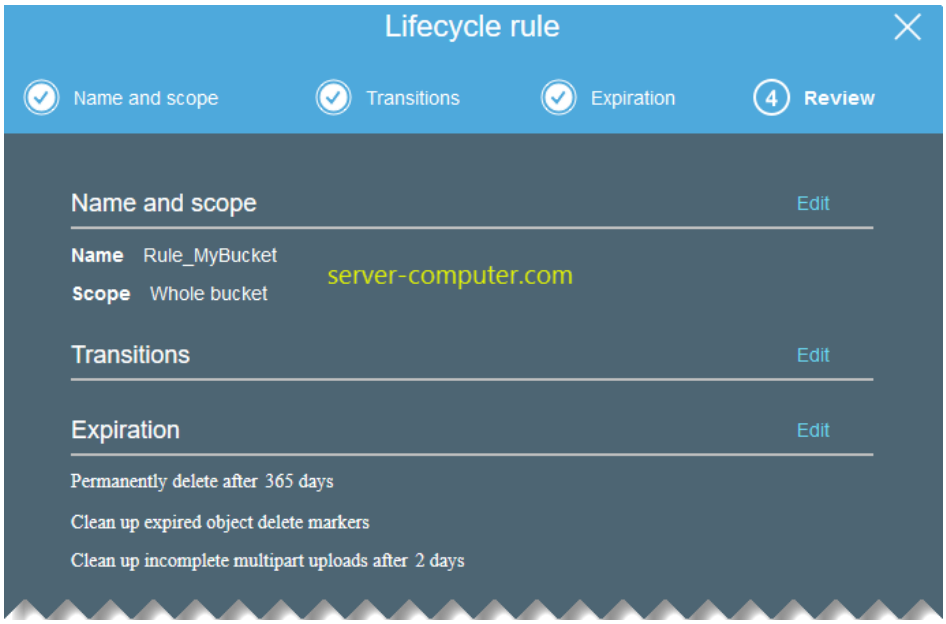
Click **Next**



Explanation: Previous versions of files after 365 days means one year permanently delete from S3 bucket.

Clean up expired and incomplete uploads after 2 days.

Click **Next**

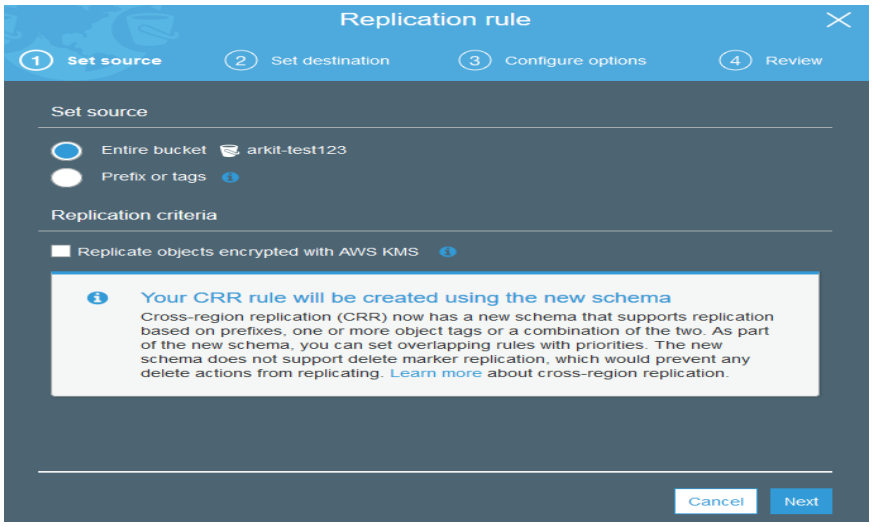


Click **Save.**

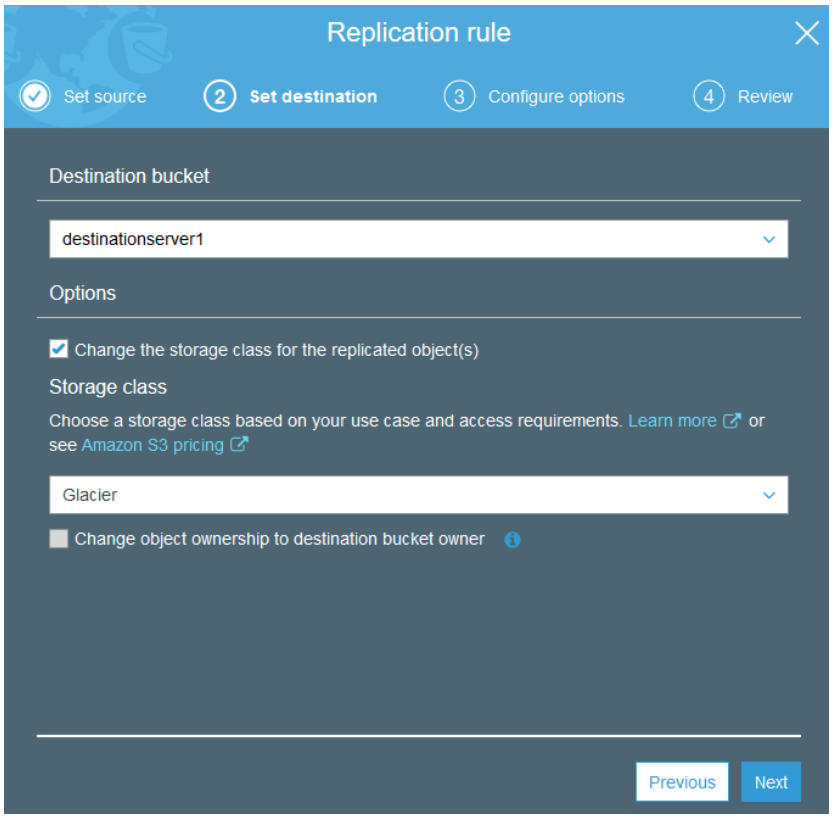
14.2. S3 Bucket Replication to Cross-Region

S3 bucket **Name → Management → Replication**

Note: In order to enable Replication for S3 bucket **Versioning** should enabled.



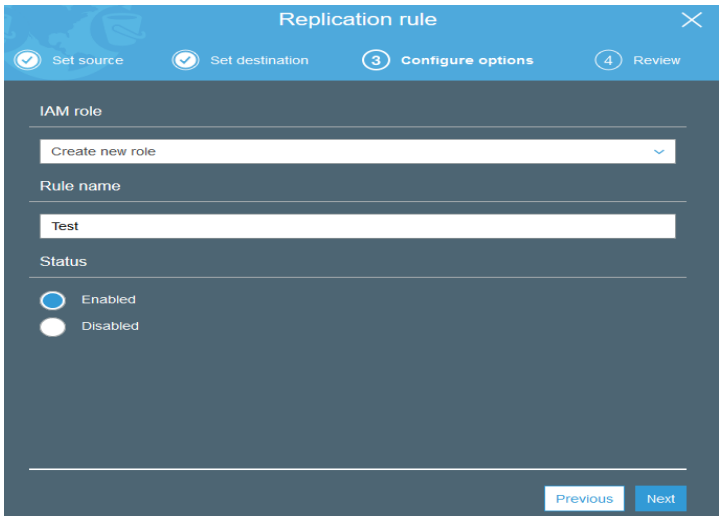
Click **Next**



Select Destination bucket within same account or another account

Options to Change Storage class and permissions in destination

Click **Next**



Select existing IAM Role or Create new for replication. In this case, I am creating new role for replication called Test

Click **Next**

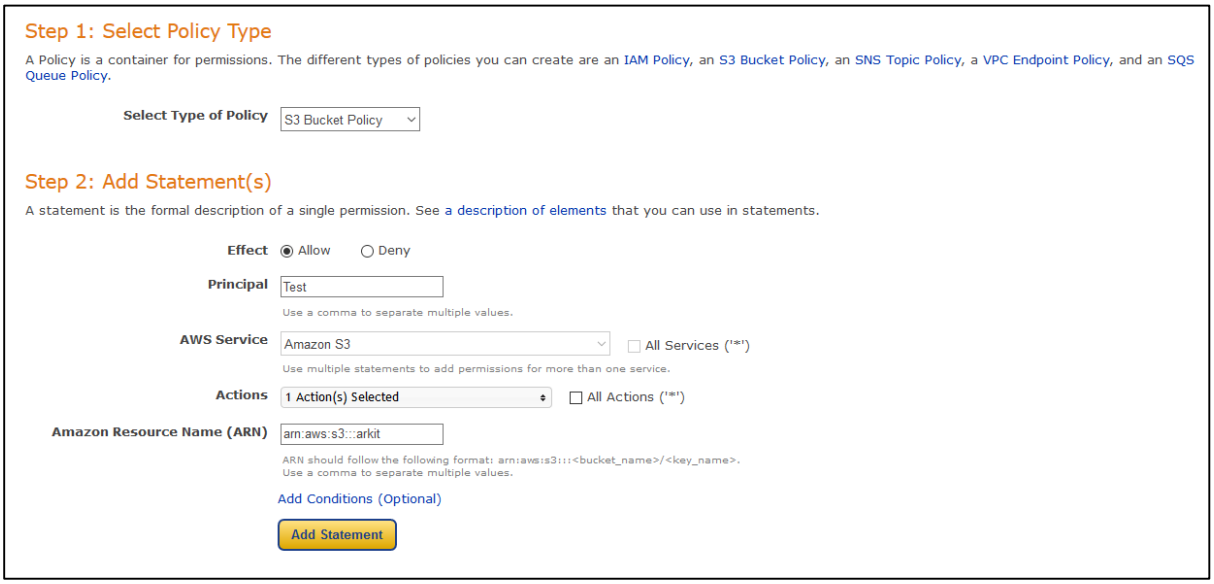
Review final and Click **Save**

14.3. S3 Bucket Policies to control Access

Click on bucket Name → Permissions → bucket policy

https://awspolicygen.s3.amazonaws.com/policygen.html

Go to this above URL and generate policy if you do not know how to write a S3 bucket policy



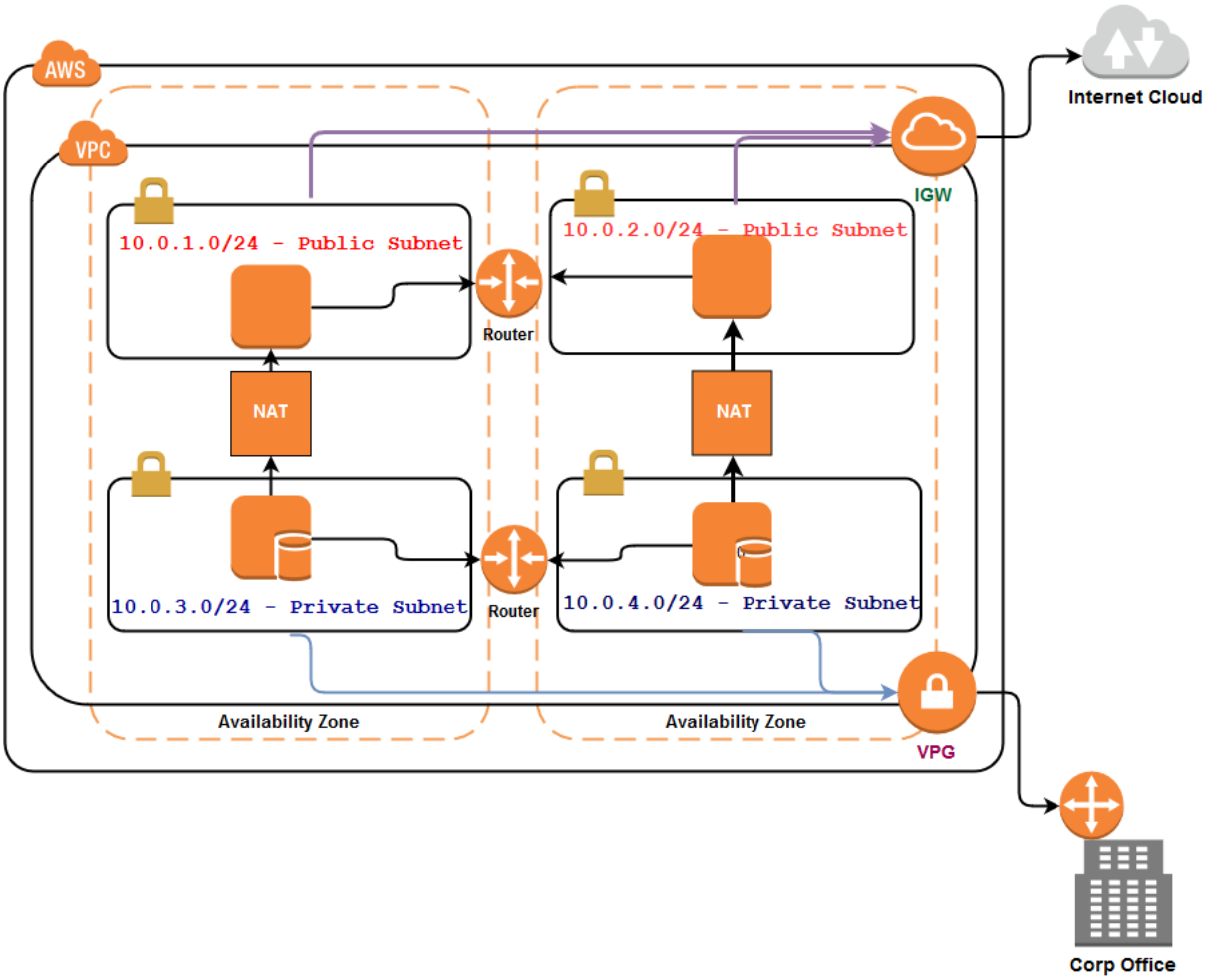
Add Statement and click on **Generate Policy**

```
{
  "Id": "Policy1543401188367",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1543401184049",
      "Action": [
        "s3:ListBucket",
        "s3:ListBucketByTags",
        "s3:ListBucketVersions"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::arkit-prog",
      "Principal": {
        "AWS": [
          "test"
        ]
      }
    }
  ]
}
```

Same policy copy and paste it in policy editor and **save**

15. VPC – Virtual Private Cloud (isolated Network)

A **virtual private cloud** (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.



Picture: 1.1 Typical VPC Example

- EC2 Instance
- Virtual Private Gateway
- Router
- Customer Gateway
- Internet Gateway
- Availability Zone
- VPC subnet

Architecture Explanation:

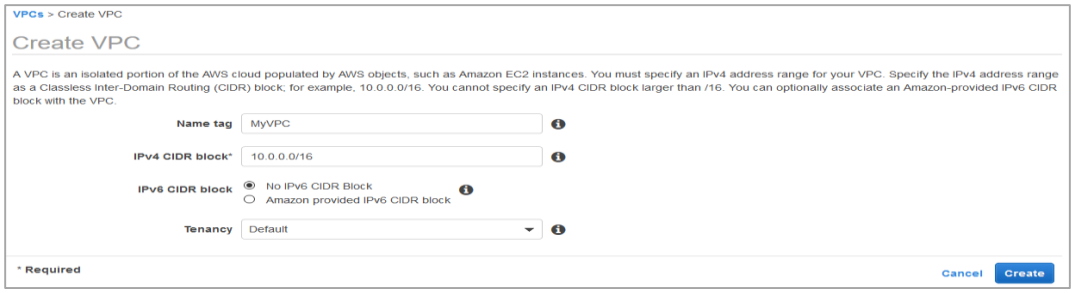
- AWS in single region
- Two Availability zones
- One Virtual Private Cloud

- Four Subnets Two Are Public and Two Are Private subnets
- Four instances Two App Servers, Two Database Servers
- One Internet Gateway to access internet
- One Virtual Private Gateway to Connect Corporate Office
- Two routers one is connected to private subnets, another is connected to public subnets

We would like to host web application with two web app servers and two Database servers. Two Tier architecture. Web app servers will serve to public, from public facing subnets. Database servers are in private network and only have access to app servers and corporate network (VPG).

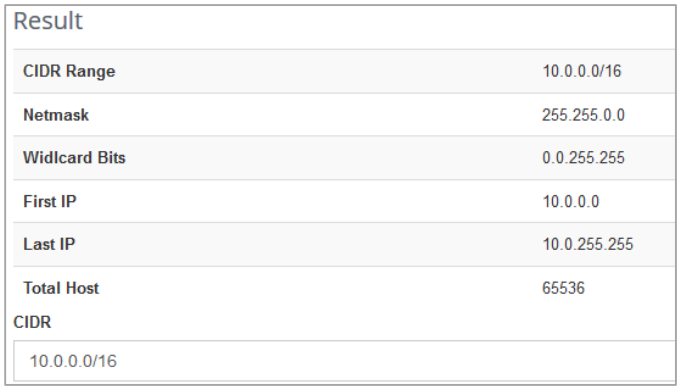
When Database servers want to download any kind of files/patches from internet it routes through NAT Gateway and get the internet data from web app servers.

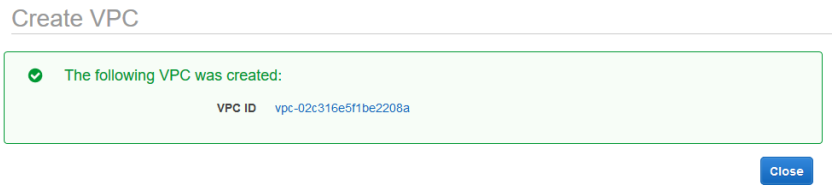
AWS Console → Services → Networking & Content Delivery → VPC → Your VPCs



- **VPC Name:** MyVPC
- **IPv4 CIDR Block:** 10.0.0.0/16 (Use this CIDR Calculator)

Click **Create**

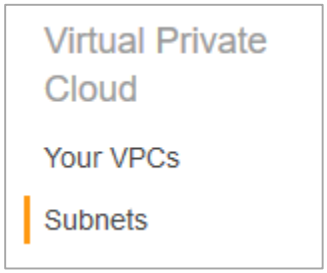


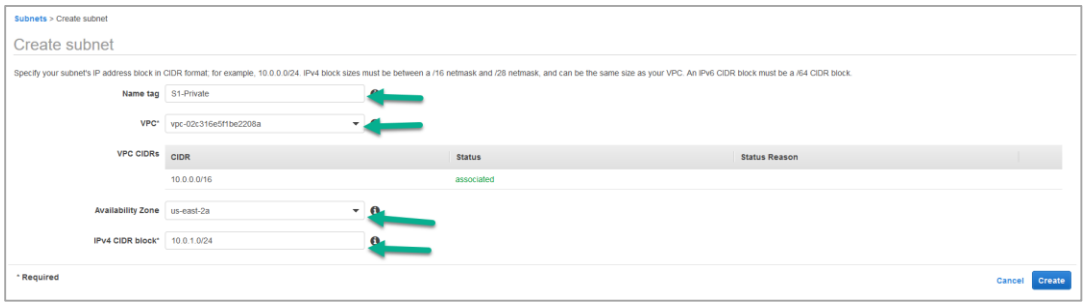


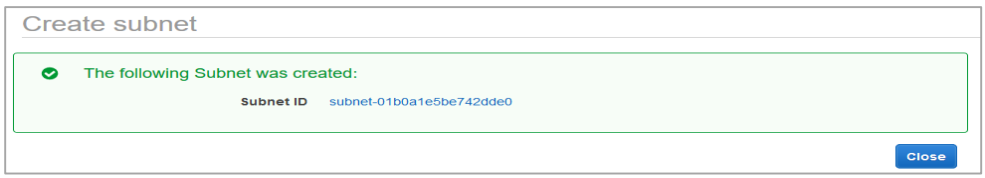
Your VPC created successfully.

15.1. Create subnets

Inside VPC to divide smaller blocks and separation

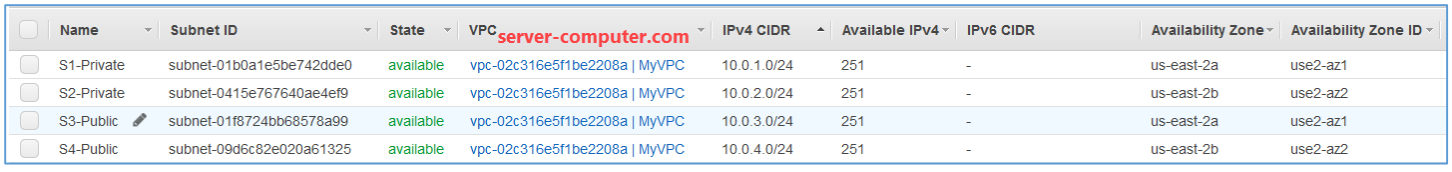






In Similar way, create all four subnets

Subnet Name	Availability Zone	CIDR Block	Private/Public
S1-Private	Us-east-2a	10.0.1.0/24	Private
S2-Private	Us-east-2b	10.0.2.0/24	Private
S3-Public	Us-east-2a	10.0.3.0/24	Public
S4-Public	Us-east-2b	10.0.4.0/24	Public

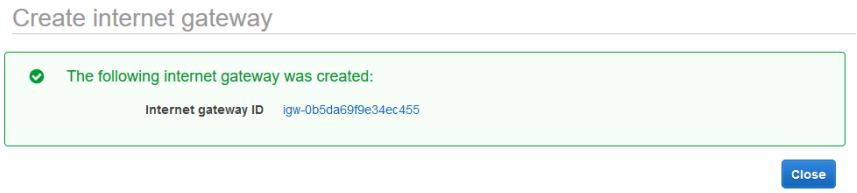


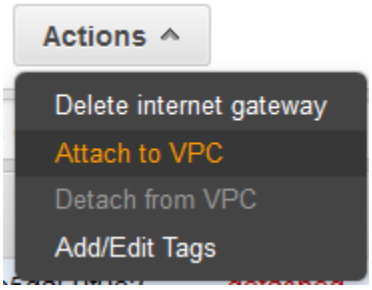
Name	Subnet ID	State	VPC server-computer.com	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Availability Zone ID
S1-Private	subnet-01b0a1e5be742dde0	available	vpc-02c316e5f1be2208a | MyVPC	10.0.1.0/24	251	-	us-east-2a	use2-az1
S2-Private	subnet-0415e767640ae4ef9	available	vpc-02c316e5f1be2208a | MyVPC	10.0.2.0/24	251	-	us-east-2b	use2-az2
S3-Public	subnet-01f8724bb68578a99	available	vpc-02c316e5f1be2208a | MyVPC	10.0.3.0/24	251	-	us-east-2a	use2-az1
S4-Public	subnet-09d6c82e020a61325	available	vpc-02c316e5f1be2208a | MyVPC	10.0.4.0/24	251	-	us-east-2b	use2-az2

15.2. Create Internet gateway and attach to VPC

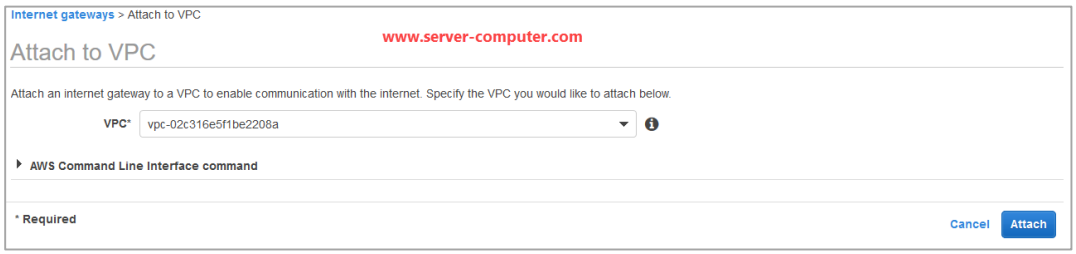
Internet Gateways. An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

Attach to S3 and S4, after attach S3 and S4 become public subnets.





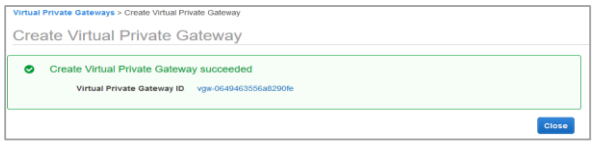
Now attach Internet Gateway to VPC

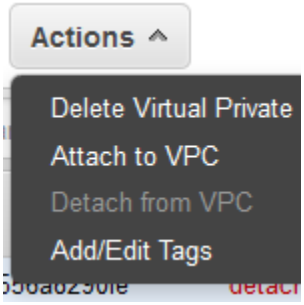


Select MyVPC in drop down menu Click **Attach**

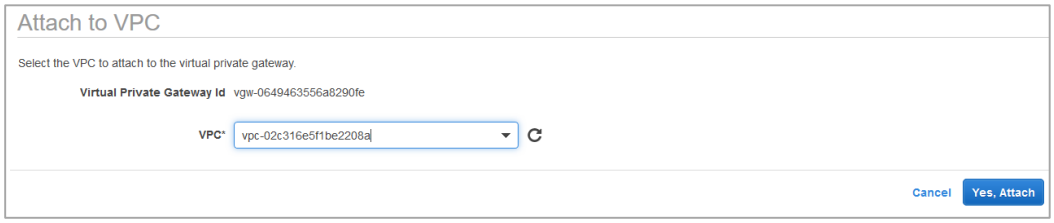
15.3. Create Virtual Private Gateway and Attach to VPC

It can be a physical or software appliance. The anchor on the AWS side of the VPN connection is called a virtual private gateway. The following diagram shows your network, the customer gateway, the VPN connection that goes to the virtual private gateway, and the VPC.





Attach VGW to MyVPC

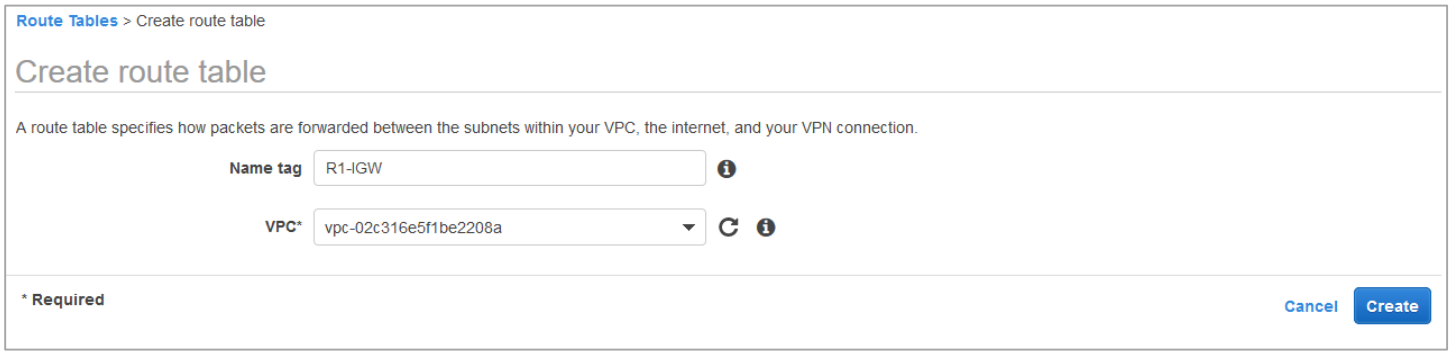


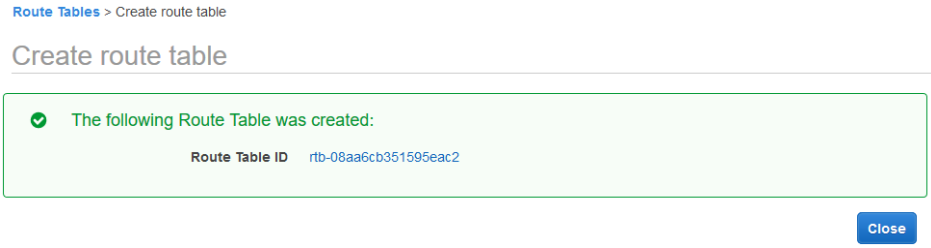
15.4. Create route tables and attach to subnets

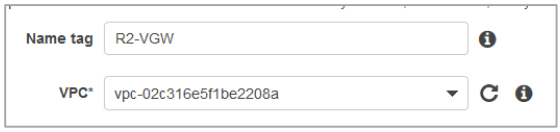
Route Tables. A route table contains a set of rules, called routes that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet.

One route for Internet gateway, another for Virtual private gateway (R1-IGW and R2-VGW)

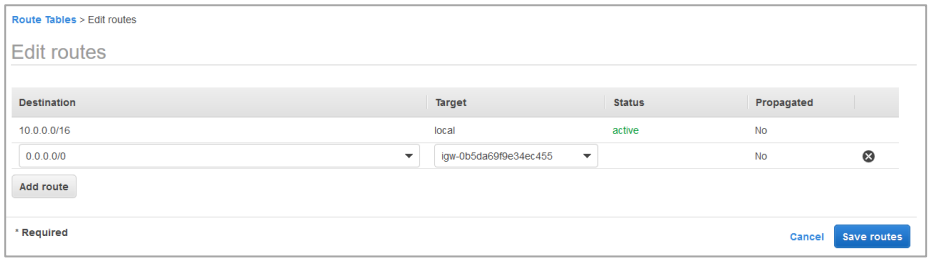
- Route - 0.0.0.0/0 to IGW
- Route - 192.168.0.0/16 to VGW

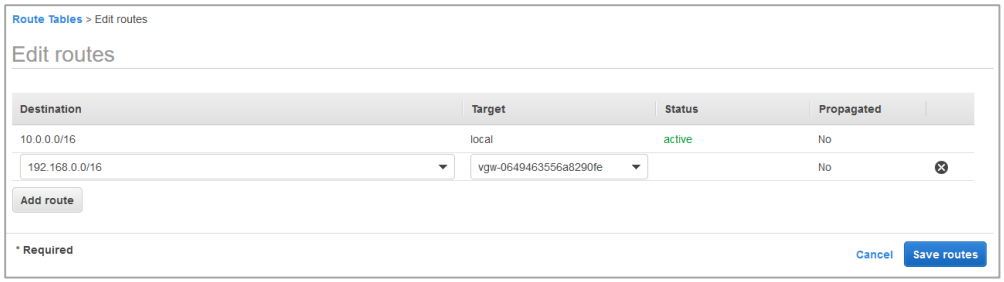




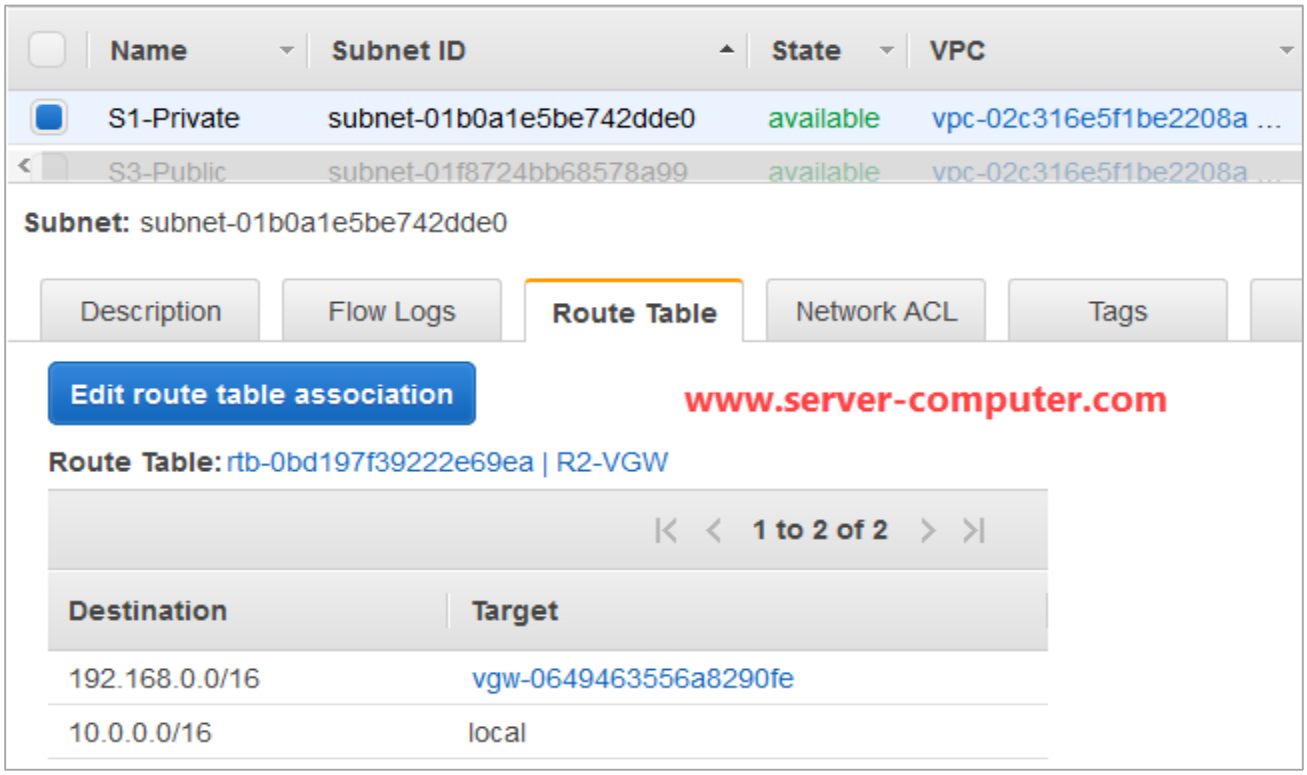


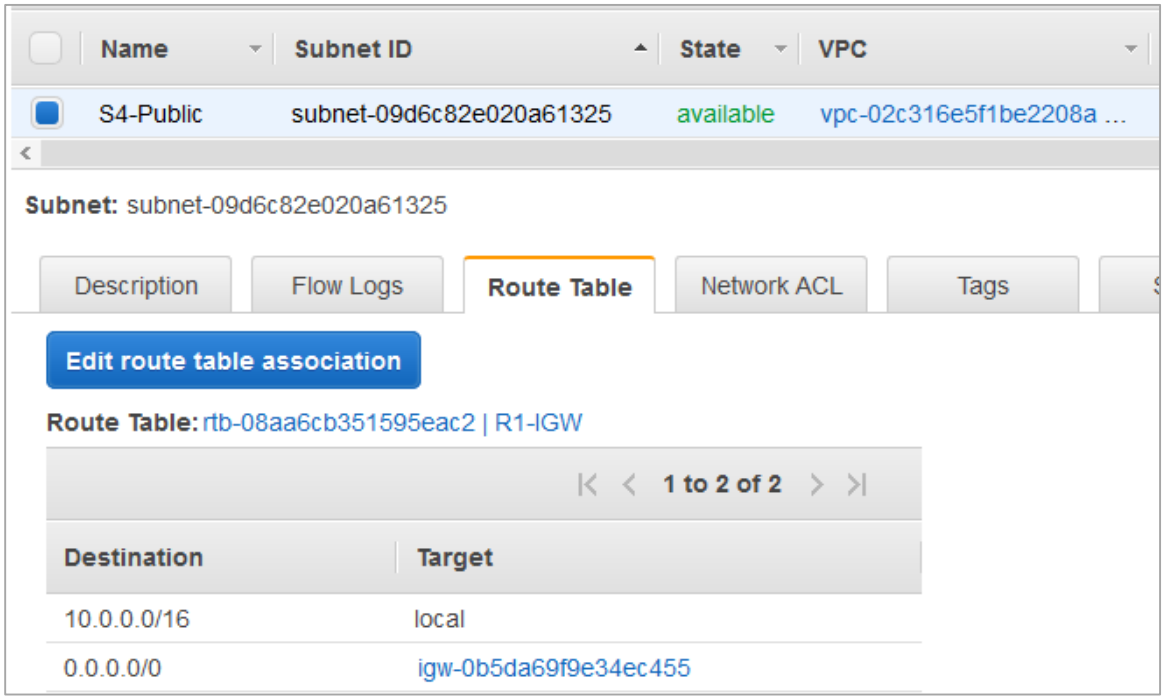
Now edit R1-IGW and add routing rule as mentioned below



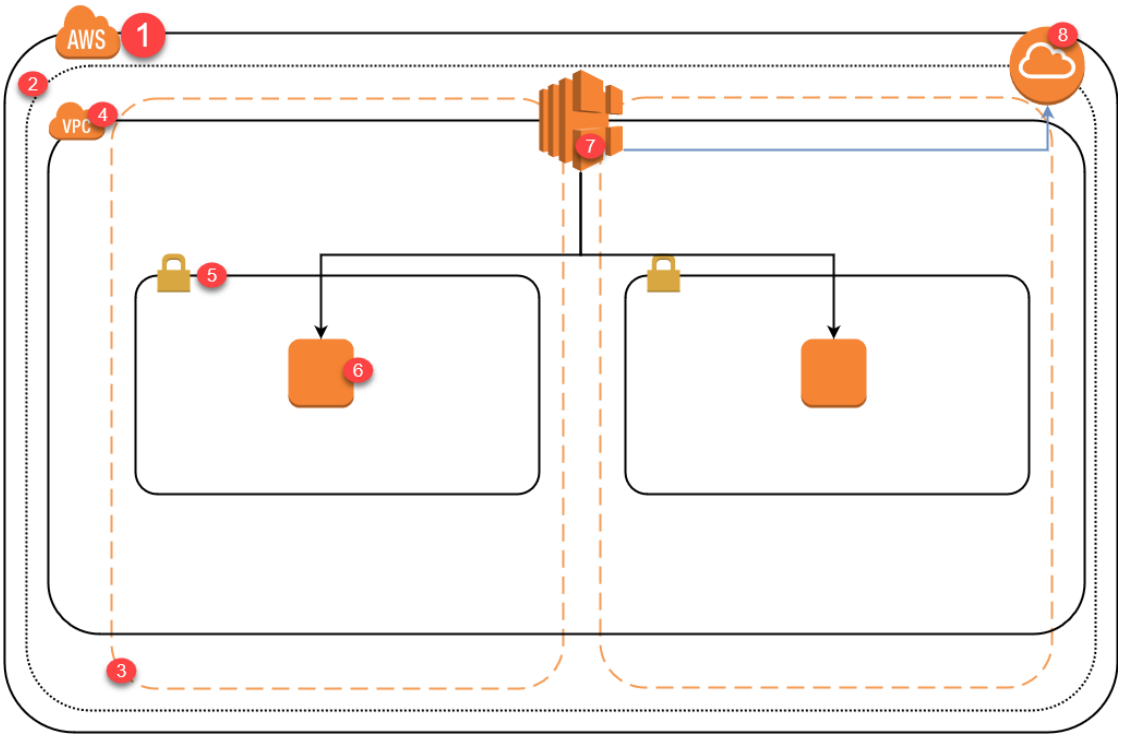


Attach routing tables to subnets. R1-IGW to S3-Public and S4-Public, public network required to have internet access. Attach R2-VGW to S1-Private and S2-Private (No internet become a private subnets)





16. AWS Elastic Load Balancer (ELB)



2.1 Elastic Load Balancer Typical Architecture

1. AWS Cloud
2. Region
3. Availability Zone
4. VPC – Virtual Private Cloud
5. VPC Subnet
6. EC2 Instance Running Webserver
7. Elastic Load Balancer
8. Internet Gateway

Elastic Load Balancing (ELB) is a load-balancing service for Amazon Web Services (AWS) deployments. ELB automatically distributes incoming application traffic and scales resources to meet traffic demands.

A Managed Load Balancing service

- Distributes load incoming application traffic across multiple targets, such as amazon EC2 instances, containers, and IP Addresses
- Recognizes and responds to unhealthy instances
- Can be public or internal-facing
- Uses HTTP, HTTPS, TCP, and SSL Protocols
- Each Load Balancer is given a public DNS name
    - Internet-facing load balancers have DNS names which publicly resolve to the public IP Addresses of the load balancer of the load balancers nodes

- Internal load balancers have DNS names, which publicly resolve to the private IP Addresses of the load balancers nodes.

Types of ELB

1. Application Load Balancer
2. Network Load Balancer
3. Classic Load Balancer

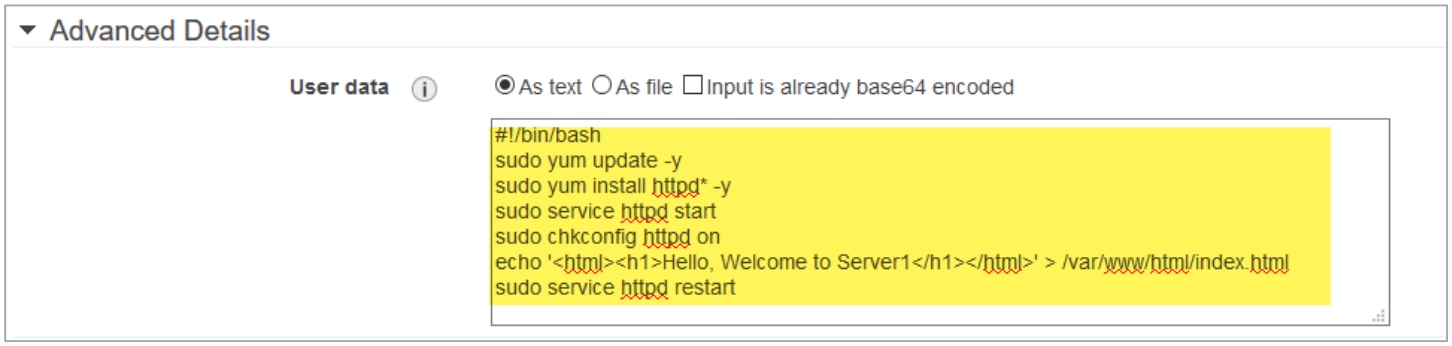
ELB Practical

- Launch two EC2 instances in different AZs
- Enable Web services
- Launch Load Balancer
- Add both instances under load balancer now check traffic

Follow **EC2 Linux instance launch steps** however in step two (configure Instance) go to down to the bottom in advanced section add below script will create auto webserver

https://github.com/techtutorials/aws-lab-guide/blob/aws/webserver.sh

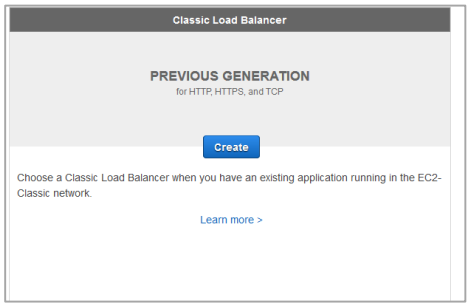
```
#!/bin/bash
sudo yum update -y
sudo yum install httpd* -y
sudo service httpd start
sudo chkconfig httpd on
echo '<html><h1>Hello, Welcome to Server1</h1></html>' > /var/www/html/index.html
sudo service httpd restart
```

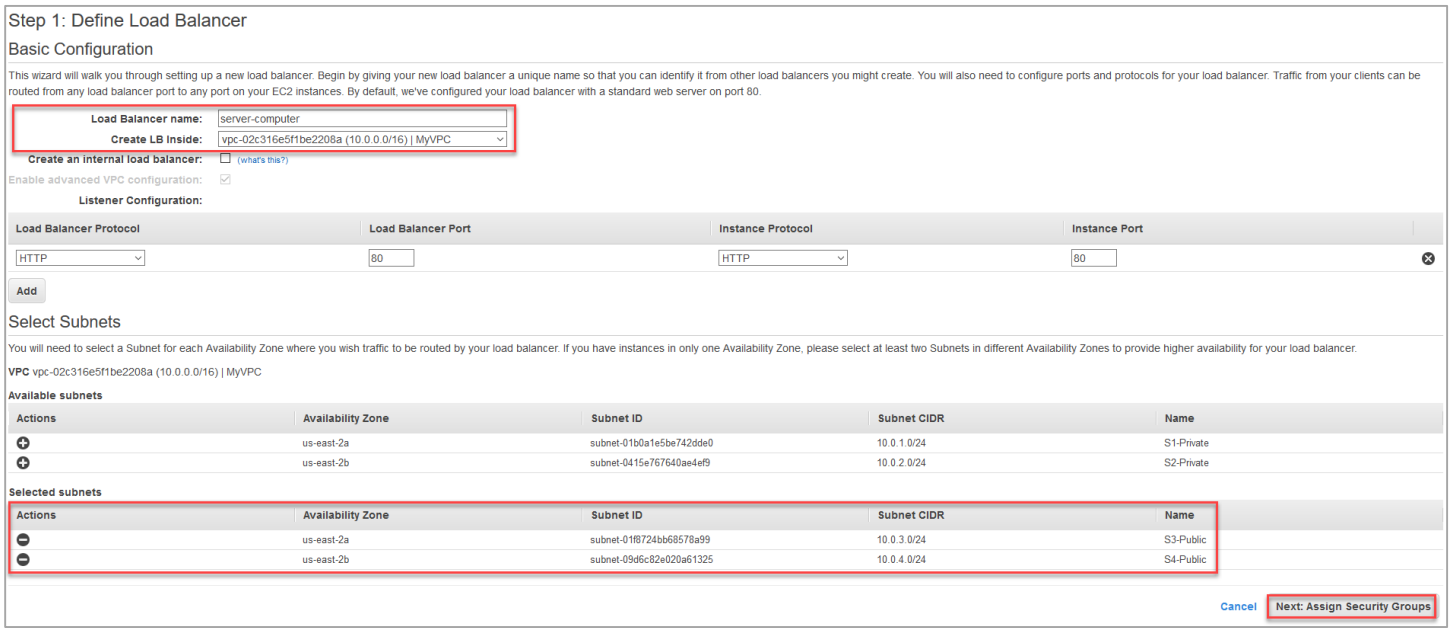


Note: while launching second instance change echo statement to server2

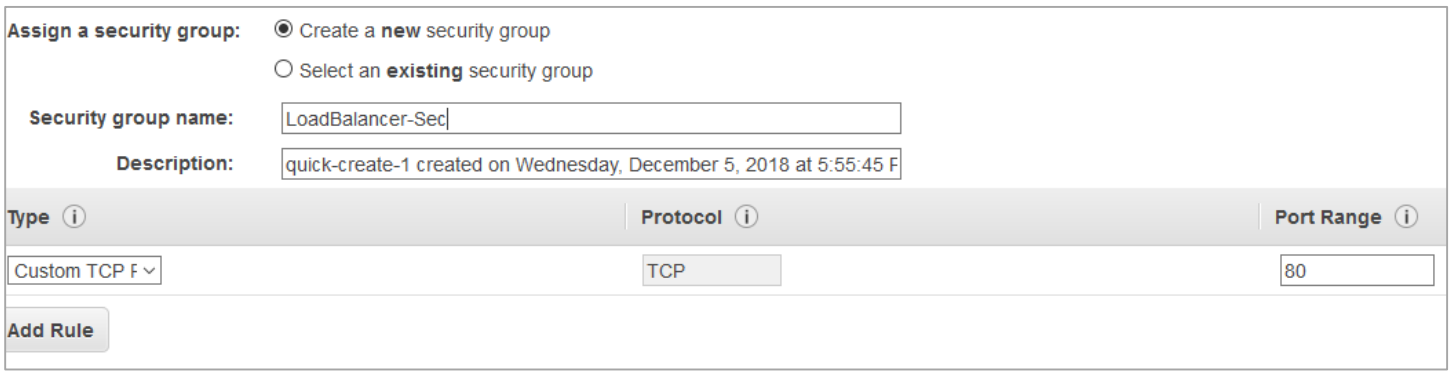
```
echo '<html><h1>Hello, Welcome to Server2</h1></html>' > /var/www/html/index.html
```

Creating Classic Elastic Load Balancer



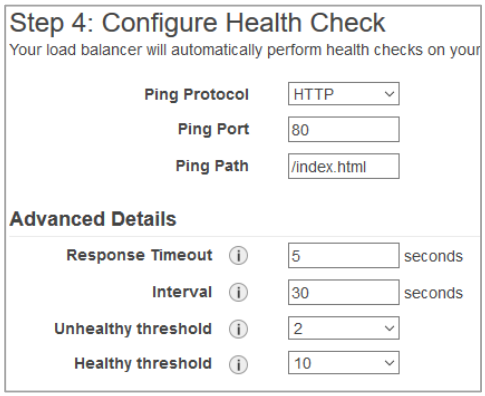


Click **Next: Assign Security Groups**



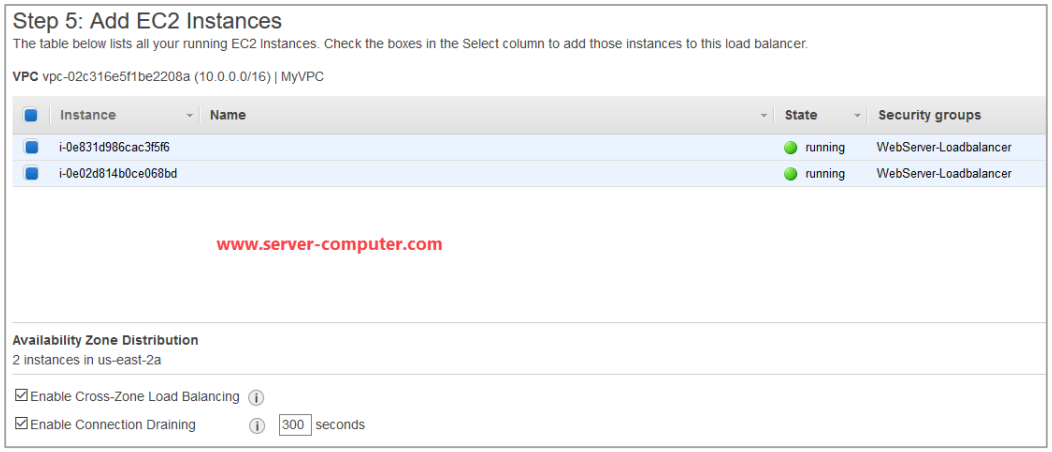
Click **Next: Security Settings**

Click **Next: Configure Health Checks**



Specify your default web file in this example I am using /index.html

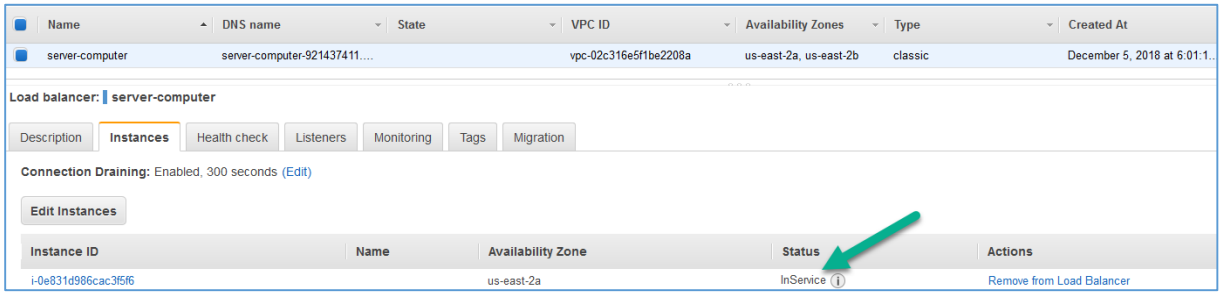
Click **Next: Add EC2 Instances**



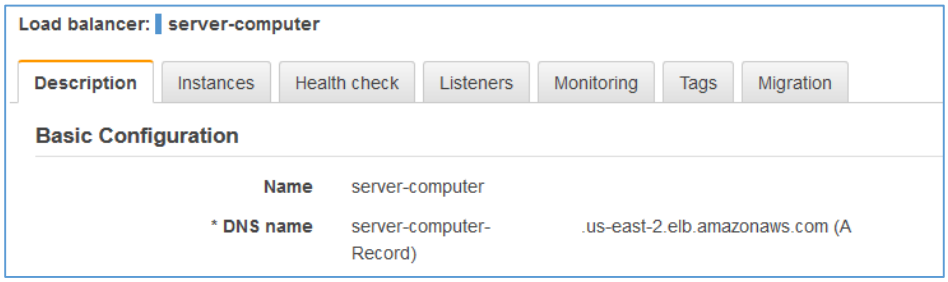
Click **Next: Add Tags**

Click **Review and Create**

Click **Create**



Check instances status should be InService



Load Balancer DNS Name copy it and paste in web browser now fresh twice you will see response is coming from Server1 and Server2

Which concludes load balancer is working fine.

17. AWS CloudTrail – Enable Governance and Auditing

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS services are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

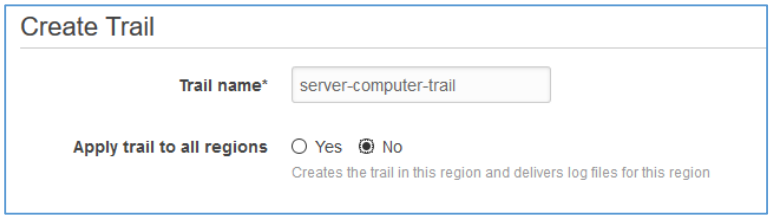
CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to Event history.

Visibility into your AWS account activity is a key aspect of security and operational best practices. You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. You can identify whom or what took which action, what resources were acted upon, when the event occurred, and other details to help you analyze and respond to activity in your AWS account.

17.1. How to Create CloudTrail

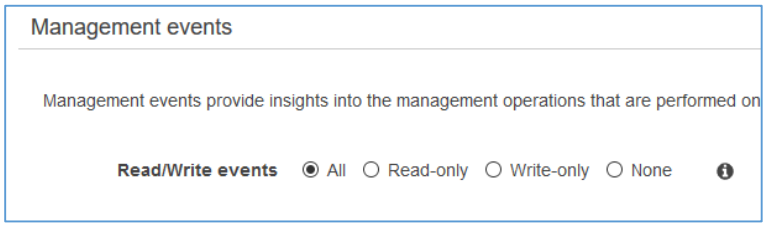
Login to AWS Console → Services → Management & Governance → CloudTrail

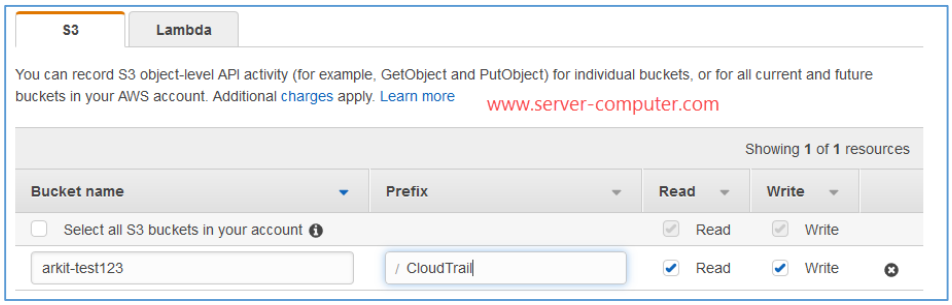
Click on **Create Trail**



Provide trail name as your wish in this case **server-computer-trail**

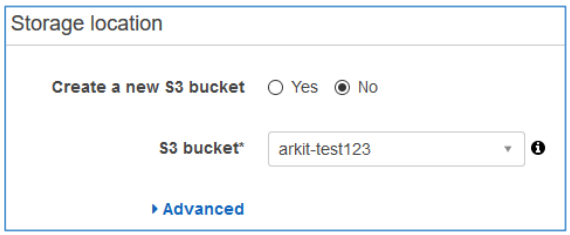
Note: If you want to audit all regions by default select "Yes" radio, button otherwise select "No"



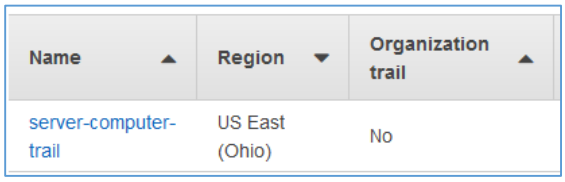


Select S3 bucket where you want to store CloudTrail Logs. CloudTrail logs uses S3 bucket for storing audit logs.

If you did not have S3 bucket created, provide bucket name in storage location section by selecting "Yes" radio button, it will create it for you. Select no if you have existing S3 bucket.



Click **Create**



CloudTrail has been created successfully.

18. Athena Analytics

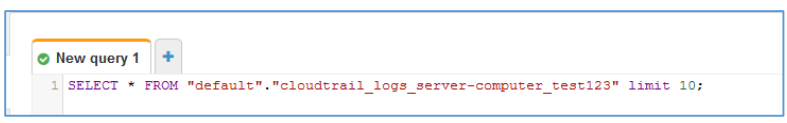
If you would like to create a table in hive using existing logs, you can create by clicking on **Athena table creation**.

```
CREATE EXTERNAL TABLE cloudtrail_logs_server-computer_test123 (
    eventVersion STRING,
    userIdentity STRUCT<
        type: STRING,
        principalId: STRING,
        arn: STRING,
```

```
        accountId: STRING,
        invokedBy: STRING,
        accessKeyId: STRING,
        userName: STRING,
        sessionContext: STRUCT<
            attributes: STRUCT<
                mfaAuthenticated: STRING,
                creationDate: STRING>,
            sessionIssuer: STRUCT<
                type: STRING,
                principalId: STRING,
                arn: STRING,
                accountId: STRING,
                userName: STRING>>>,
    eventTime STRING,
    eventSource STRING,
    eventName STRING,
    awsRegion STRING,
    sourceIpAddress STRING,
    userAgent STRING,
    errorCode STRING,
    errorMessage STRING,
    requestParameters STRING,
    responseElements STRING,
    additionalEventData STRING,
    requestId STRING,
    eventId STRING,
    resources ARRAY<STRUCT<
        arn: STRING,
        accountId: STRING,
        type: STRING>>,
    eventType STRING,
    apiVersion STRING,
    readOnly STRING,
    recipientAccountId STRING,
    serviceEventDetails STRING,
    sharedEventID STRING,
    vpcEndpointId STRING
)
COMMENT 'CloudTrail table for server-computer-test123 bucket'
ROW FORMAT SERDE 'com.amazon.emr.hive.serde.CloudTrailSerde'
STORED AS INPUTFORMAT 'com.amazon.emr.cloudtrail.CloudTrailInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://server-computer-test123/AWSLogs/687993403879/CloudTrail/'
TBLPROPERTIES ('classification'='cloudtrail');
```

Create table and query using athena interface

Analytics → Athena



You can see the data in tabular format

```
DROP TABLE cloudtrail_logs_server-computer_test123;
```

Delete Athena table using above like query (replace table name).

Otherwise, for RAW log go to your S3 bucket and click on bucket name → **AWSLogs** → **Account Number** → You can see all the CloudTrail logs over there.

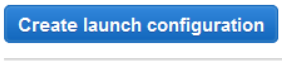
Download the json.gz file and analyze the activities

19. Auto Scaling

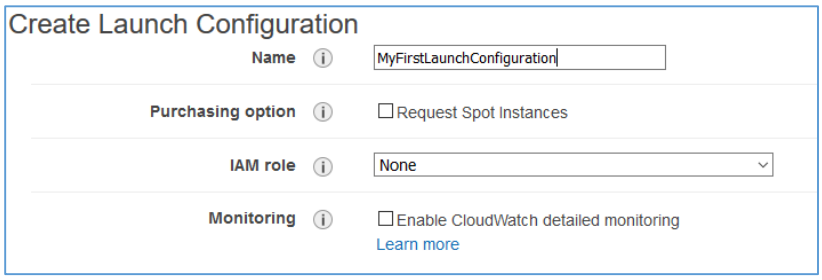
Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

19.1. Launch configuration

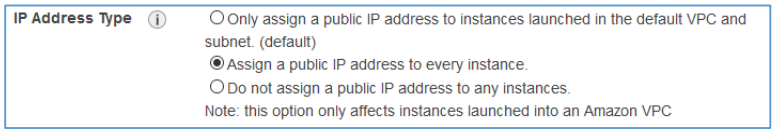
Login to **AWS Console** → **EC2** → (Under Auto Scaling) Click on **Launch Configurations**



- ➔ **Choose AMI** (I select Ubuntu 18.04 LTS)
- ➔ **Choose Instance Type** (t2.micro) Click Next: Configure Details



>> Click **Advanced Details**



Note: In case there is no default VPC available in selected zone (In my case I deleted default VPC).

Click **Next: Add Storage**

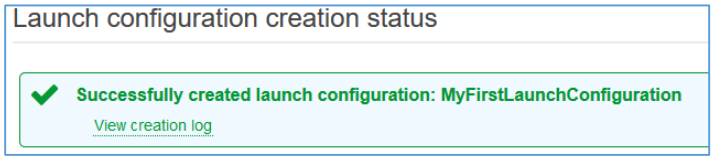
Click **Next: Configure Security Group**

Select existing Security group or create new security group, as you are wish, (Selecting existing would be good)

Click **Review**

Click Create Launch Configuration

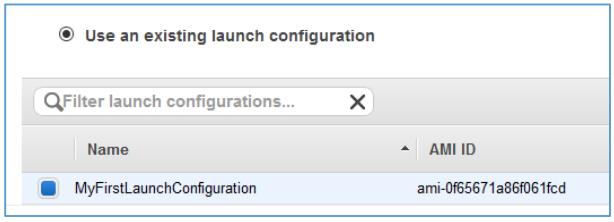
Select the Key Pair or create key pair



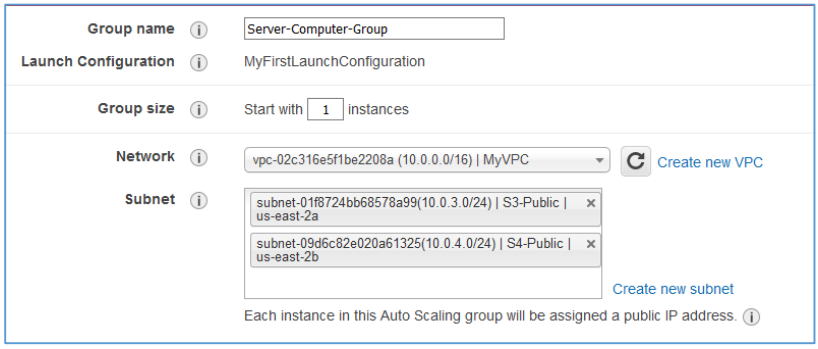
Launch configuration created successfully. Click **Close**

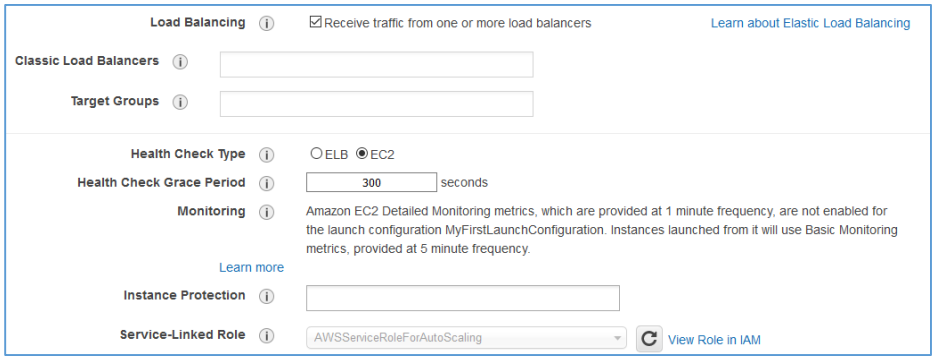
19.2. Auto Scaling Groups

Select Auto Scaling Groups → Create Auto Scaling Group → Select Launch Configuration



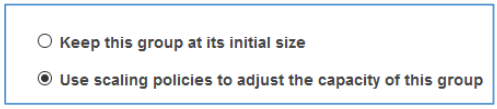
Click **Next Step**





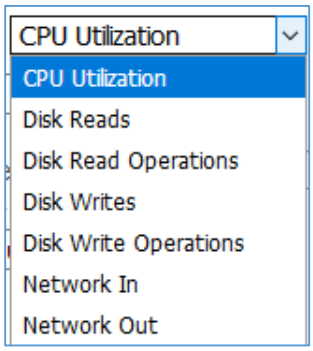
If you are auto-scaling group, want load balancer you can add ELB to auto scaling group

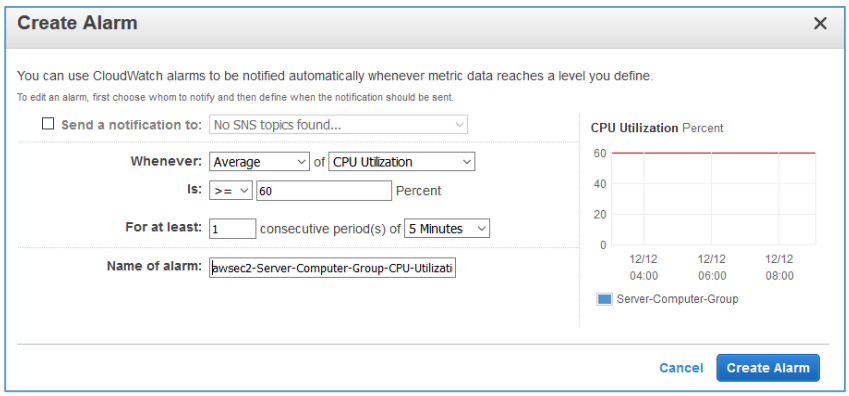
Click **Next: Configure Scaling Policies**



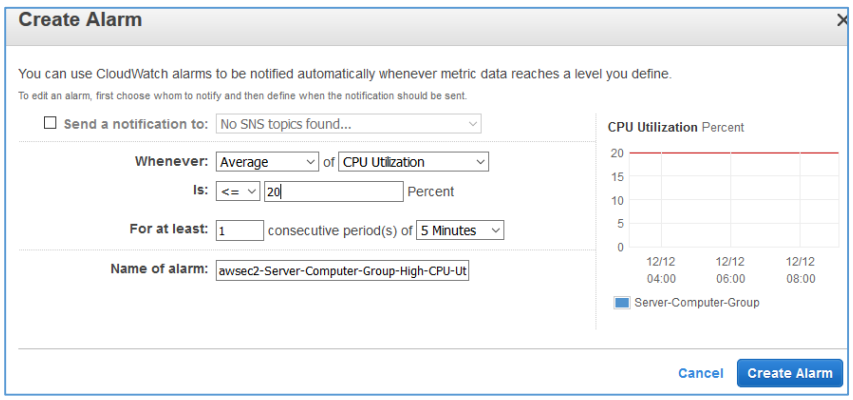
If you do not want to create scaling policy, select first radio button otherwise select use scaling policies button

Below are the conditions you can use for auto scaling EC2 instances

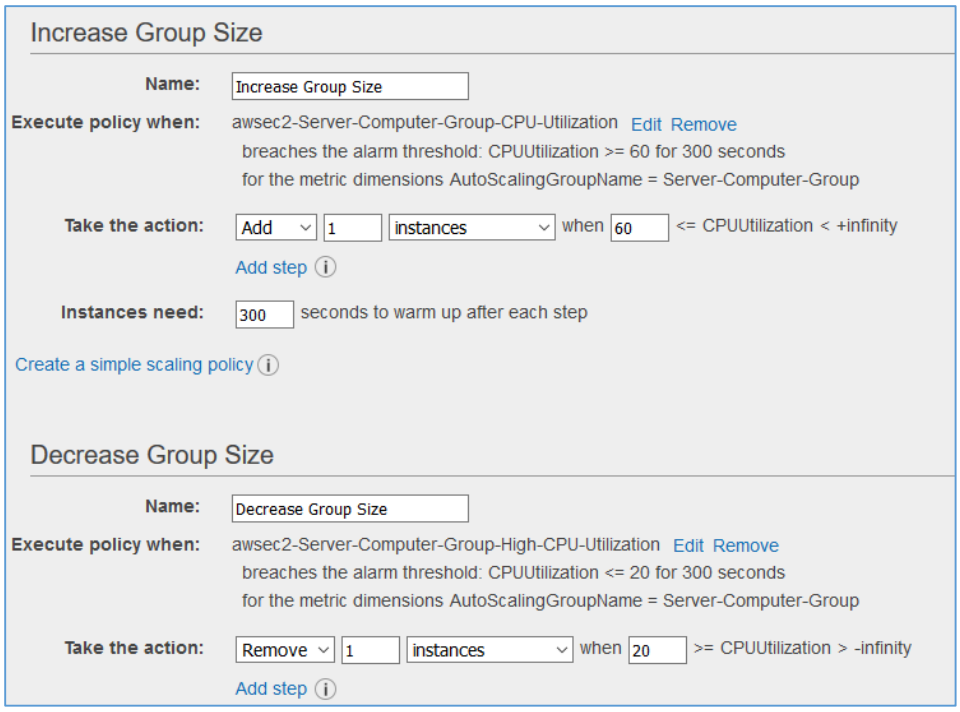




Created Auto increase group IF CPU Utilization is Greater than or equal to 60 for 5minutes add new EC2 instance to auto scaling group

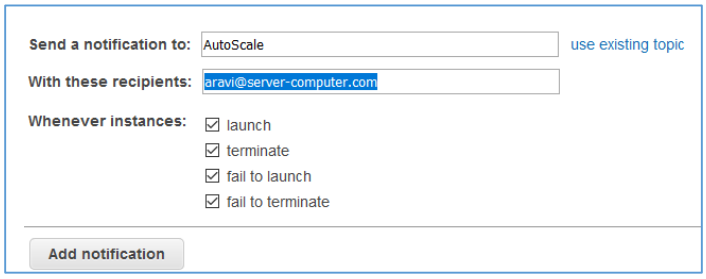


Create auto decrease group IF CPU Utilization is less than or equal to 20 for 5 minutes remove on EC2 instance from scaling group



Click **Next: Configure Notifications**

If you want notifications when auto scale triggers create notification

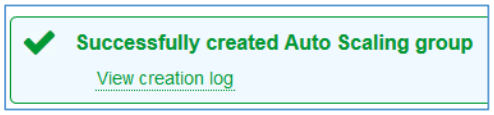


Click **Next: Configure Tags**

Add tags for recognizing auto scale instances

Click **review**

Click **Create Auto Scaling Group**



Now go back to instances you would see EC2 instances launched by auto scaling group configuration.

In order to create a CPU load to test auto scaling use below scripts

```
while true; do true; done &
dd if=/dev/zero of=/dev/null &
```

Execute above scripts multiple times in your EC2 instances, to create CPU Load is more than 60 percent for 5 minutes it will automatically launch another EC2 instance.

Wait for 5 Minutes and see

To scale down identify the background running jobs and kill them

```
jobs
fg <Job Number>
CTRL + C
```

OR

```
ps -aux |grep dd |awk '{print $2}' | xargs kill -9
ps -aux |grep bash |awk '{print $2}' | xargs kill -9
```

OR

```
kill -9 <PID>
```

Wait for 5 minutes EC2 instances will be terminated automatically which are launched using auto scale option.

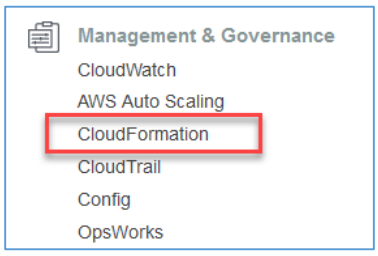
20. ClodFormation

AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment. CloudFormation allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts. This file serves as the single source of truth for your cloud environment.

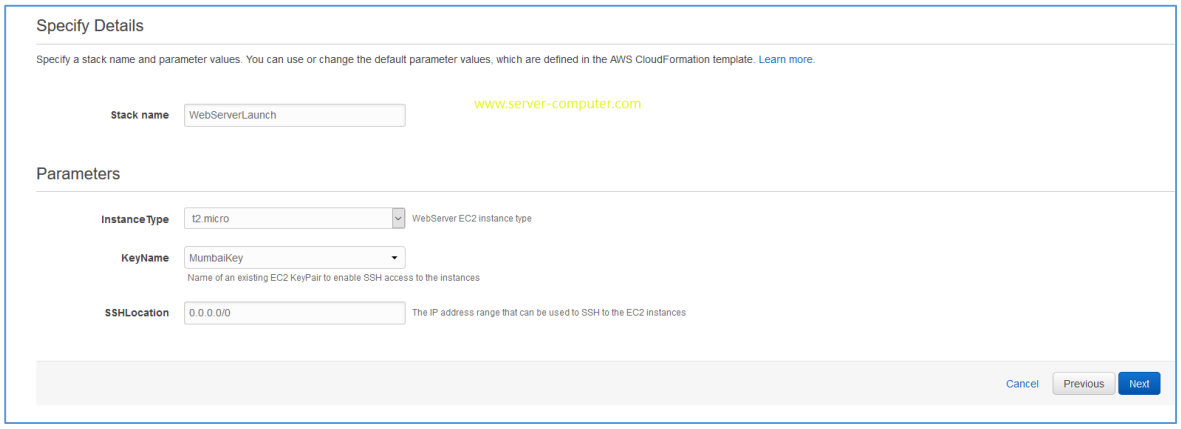
AWS CloudFormation is available at no additional charge, and you pay only for the AWS resources needed to run your applications.

"Cloud Infrastructure as Code"

Login to **AWS Web console →Services**

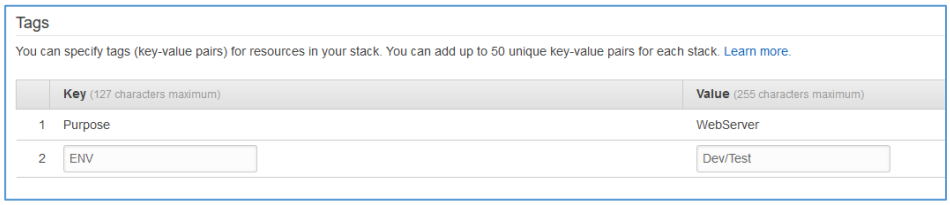


Create Stack

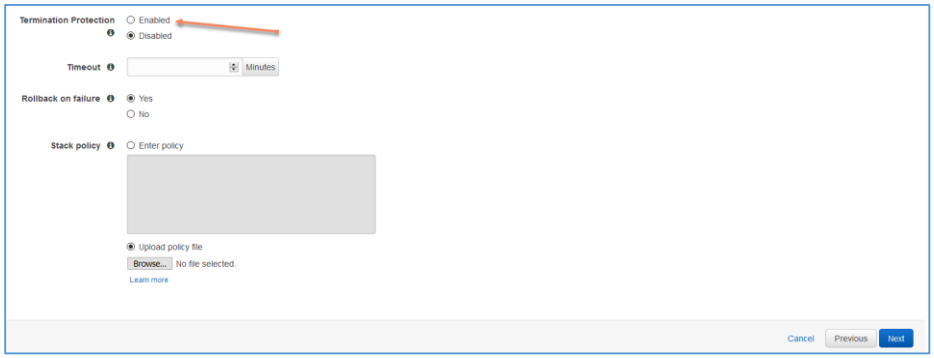


https://github.com/techtutorials/aws-lab-guide/blob/aws/LaunchEC2WebServer.template

Download and upload the template file Click **Next**



Add Tags



Click **Next**

Click on **Create**

It will create S3 bucket for CF template store and keeps your CloufFormation templates in it

If you delete CloufFormation, it will automatically delete associate stack/resources

21. Amazon FSx

Amazon FSx provides fully managed third-party file systems. Amazon FSx provides you with the native compatibility of third-party file systems with feature sets for workloads such as Windows-based storage, high-performance computing

(HPC), machine learning, and electronic design automation (EDA). You don't have to worry about managing file servers and storage, as Amazon FSx automates the time-consuming administration tasks such as hardware provisioning, software configuration, patching, and backups. Amazon FSx integrates the file systems with cloud-native AWS services, making them even more useful for a broader set of workloads.

Services → Storage → FSx → Create File system

Note: If you're looking for HPC High Performance computer then select FSX for Lustre

Click **Next**

File System name

Minimum 300GB and Max 65536GB

Default throughput 8MB/s you can also select different values of throughput

Select Network & Security

- ✓ VPC
- ✓ AZ
- ✓ Subnet
- ✓ Security Group

Windows Authentication

Note: Must be active directory or create new active directory in AWS

Encryption

Maintenance preferences

Select backup window time

Click **Next**

22. SQS – Simple Queue Service

Amazon SQS provides several advantages over building your own software for managing message queues or using commercial or open-source message queuing systems that require significant up-front time for development and configuration.

These alternatives require ongoing hardware maintenance and system administration resources. The complexity of configuring and managing these systems is compounded by the need for redundant storage of messages that ensures messages are not lost if hardware fails.

What can be used to communicate between components?

- ✓ Amazon Simple Queue Service (SQS)

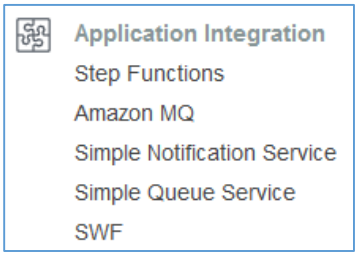
Standard Queue

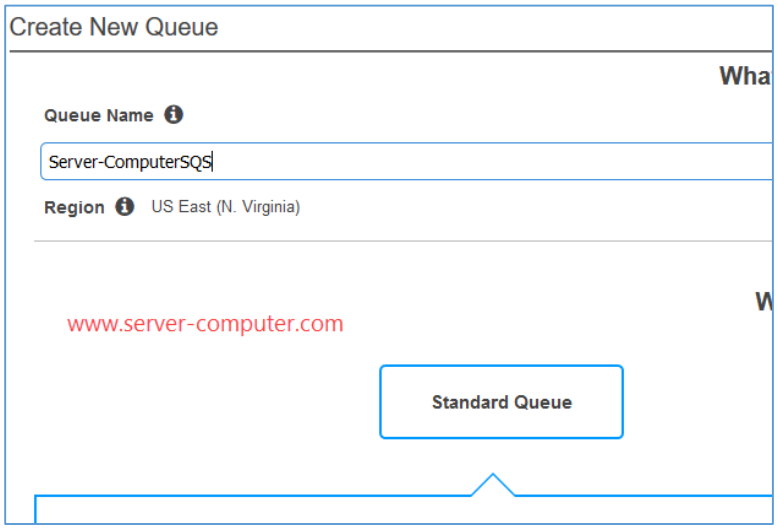
- At-Least-Once Delivery
- Best-Effort-Ordering

FIFO Queue

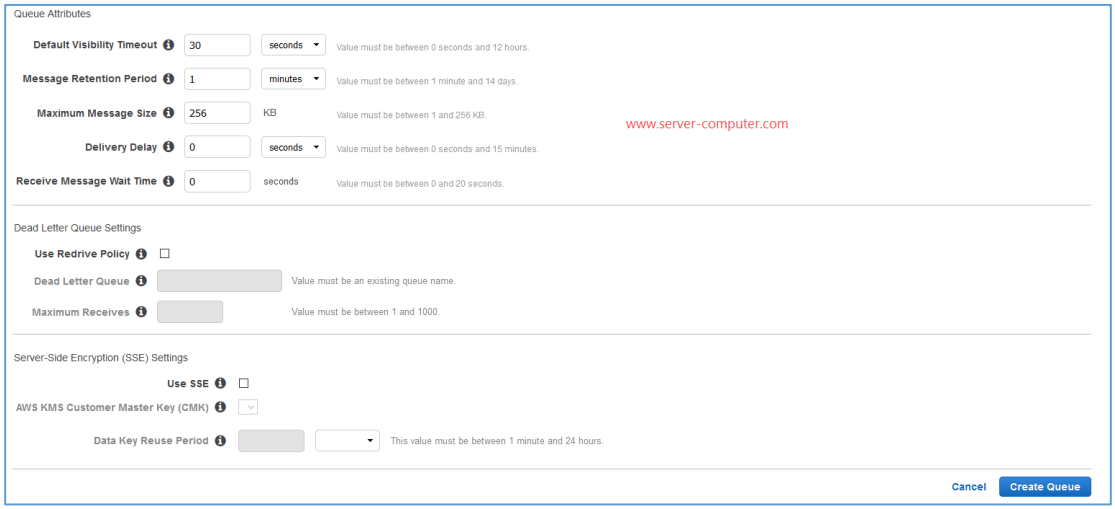
- Exactly-Once Processing
  - Duplicates are not introduces
- Limited Throughput
  - Up to 300 send, receive, delete per second

Services → Application Integration → Simple Queue Service





Provide queue name and Click **Configure Queue**

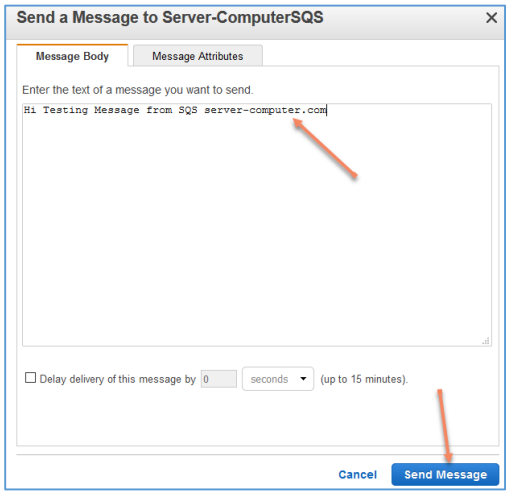


Click **Create Queue**

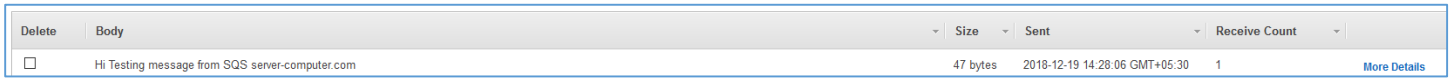
New queue created successfully. Now send message and poll to see the message queue

Select newly created queue name and Actions → send message

Write the message in message box example is shown in below screenshot Click **Send Message**



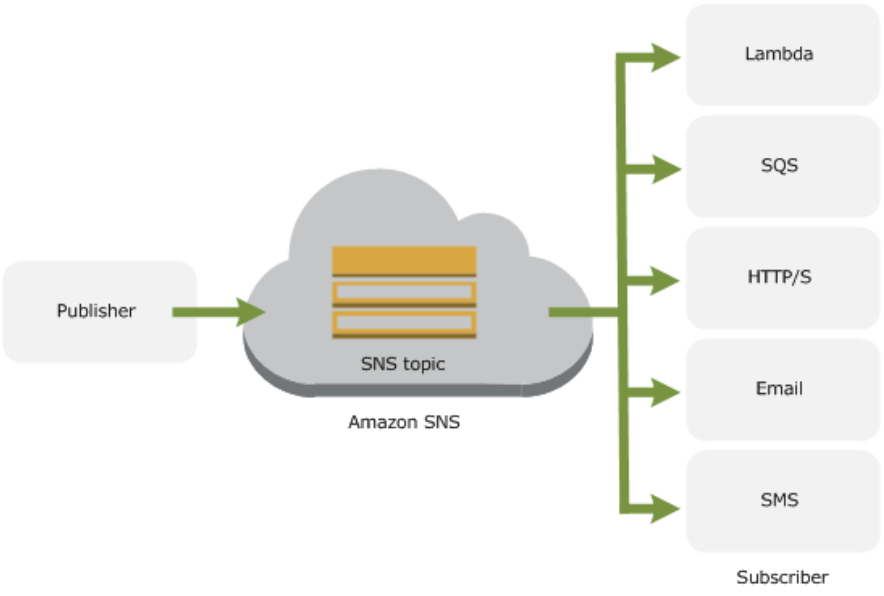
Close the popup window, select queue name **Actions → View/Delete messages → start polling for messages**



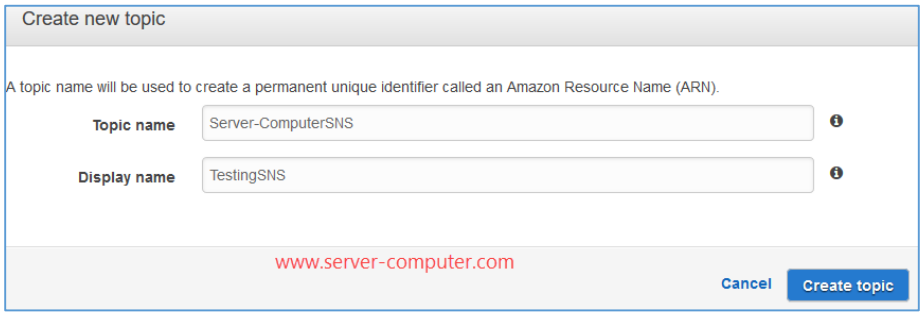
This scenario is only for testing SQS or practicing SQS. If you know use case or project, where you can integrate SQS try

23. SNS – Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers. Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel. Subscribers (i.e., web servers, email addresses, Amazon SQS queues, AWS Lambda functions) consume or receive the message or notification over one of the supported protocols (i.e., Amazon SQS, HTTP/S, email, SMS, Lambda) when they are subscribed to the topic.



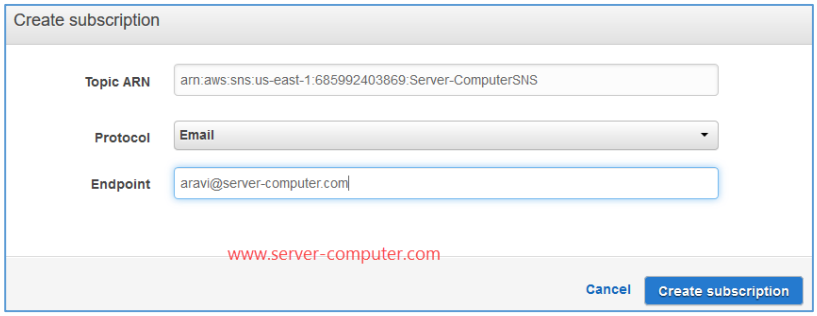
Services → Simple Notification Service → Create topic



Click **Create Topic**

Topic created successfully. Click on topic

Create subscription



Subscription will sent an email for verification after verification you will see subscription ID

Click **Publish to Topic**

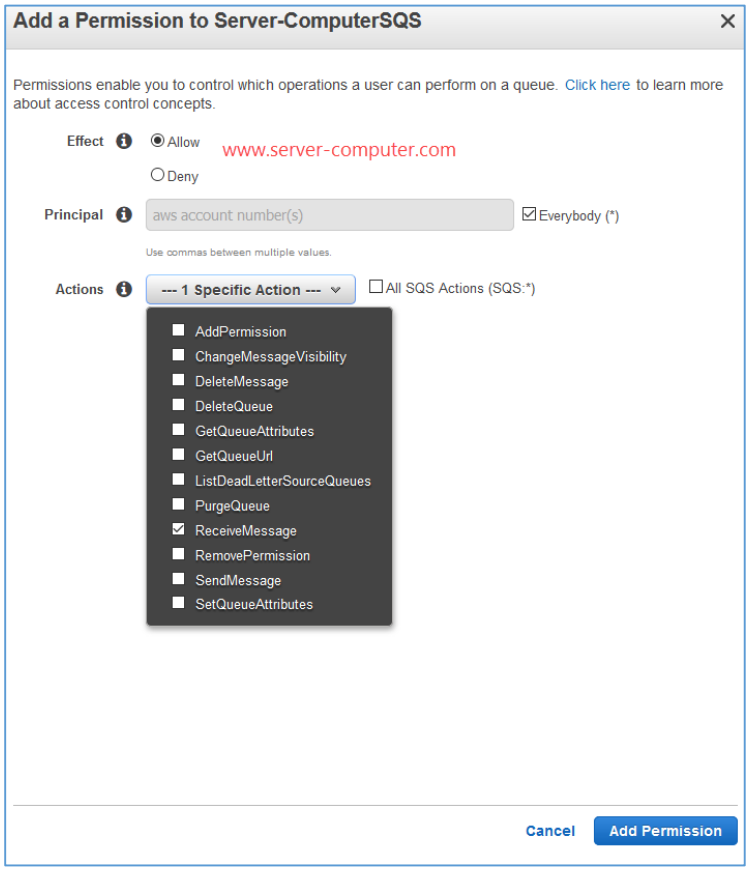
Write Subject and Message body click publish

All the subscribers will receive email immediately

Will do Flow as **SNS → SQS → Lambda function**

Go to SQS and provide permissions to SNS to send notifications using ARN value

Select SQS Queue and add permissions



Copy the ARN value from Details tab on SQS

arn:aws:sqs:us-east-1:585692493869:Server-ComputerSQS

Change back to SNS and create Subscription under topic

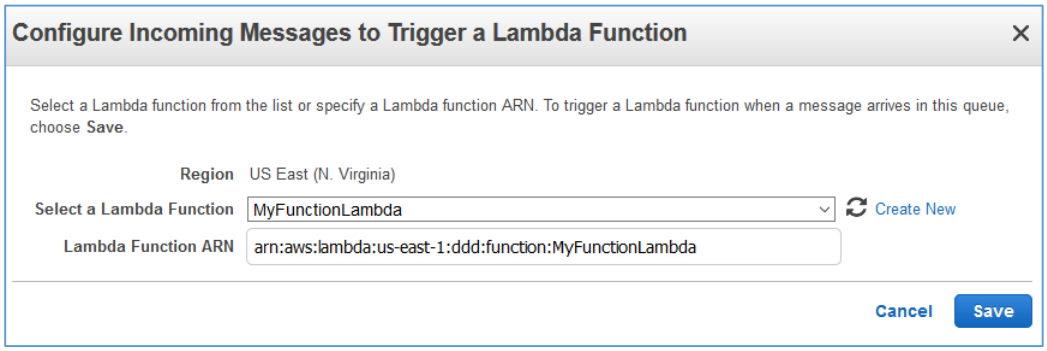
- ✓ Topic ARN : Autofil
- ✓ Protocol: Amazon SQS
- ✓ EndPoint: ARN Value copied from SQS

Send topic

Go back to SQS and View/Delete Messages → Start poling messages you can see the message from SNS

Similar to this create Lambda function, get ARN value from Lambda, and add to SQS for further triggers

Queue Actions → configure Trigger for Lambda Function



Go back to SNS and publish to topic

As soon as SNS trigs SQS will send message after words lambda will execute the defined function.

24. AWS CLI

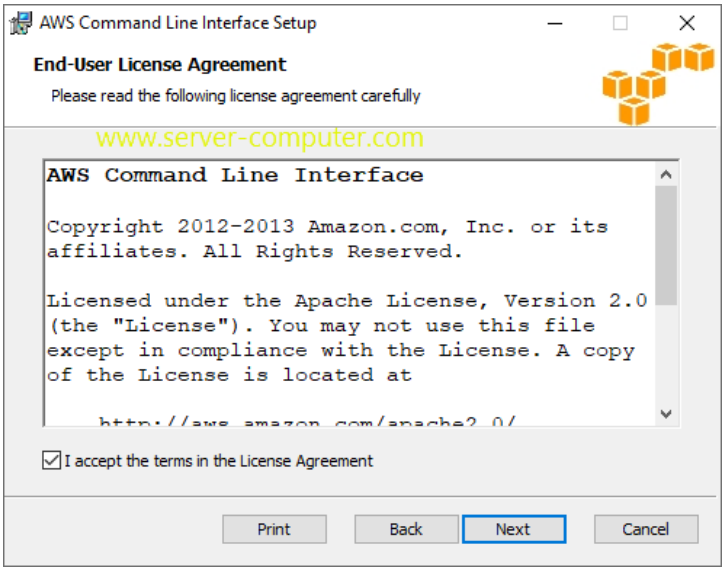
The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI introduces a new set of simple file commands for efficient file transfers to and from Amazon S3.

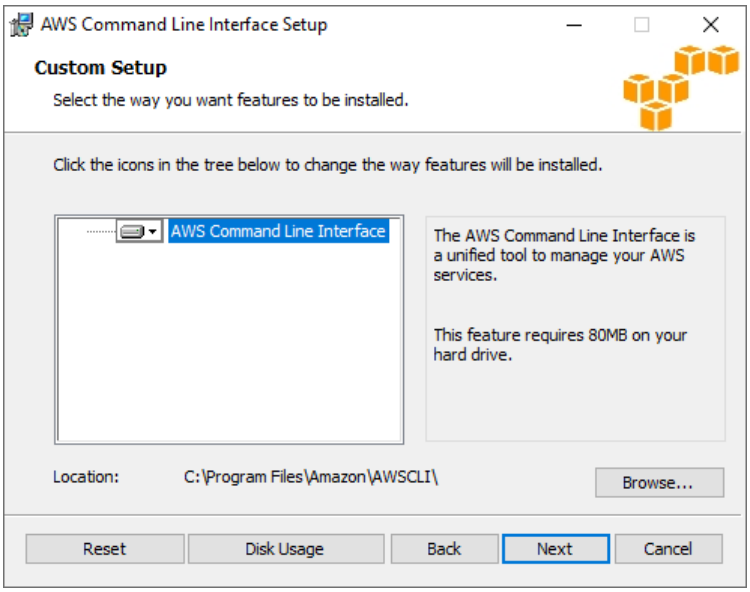
Aws cli configuration for Linux

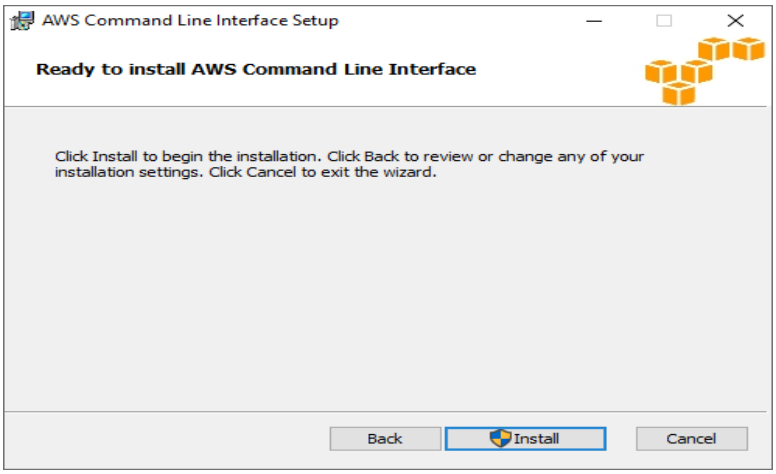
Download AWS CLI for Windows 64 Bit

You should require administrator privileges to install this package in windows machine

Double click on .msi file







Click **Install**

Click **Finish**

Login back to AWS Management console and create user with programmatic access **Refer Topic 5 download** ACCESS Key and secret key

Start Menu → Run → cmd

cd C:\Program Files\Amazon\AWSCLI\bin

Change your directory path to above mentioned

>aws configure

```
C:\Program Files\Amazon\AWSCLI\bin>aws configure
AWS Access Key ID [**********ULVQ]:
AWS Secret Access Key [**********XhN2]:
Default region name [ap-south-1]:
Default output format [None]:
```

Now successfully installed and configure aws cli, run few aws cli commands to manage AWS infrastructure

```
C:\Program Files\Amazon\AWSCLI\bin>aws s3 mb s3://servercomputerbucket
make_bucket: servercomputerbucket

C:\Program Files\Amazon\AWSCLI\bin>aws s3 ls
2018-12-18 08:31:47 arkit-test123
2018-12-20 16:57:01 servercomputerbucket

C:\Program Files\Amazon\AWSCLI\bin>aws s3 cp D:\Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf s3://servercomputerbucket
upload: D:\Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf to s3://servercomputerbucket/Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.
pdf

C:\Program Files\Amazon\AWSCLI\bin>aws s3 ls s3://servercomputerbucket
2018-12-20 16:58:39   25173965 Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf

C:\Program Files\Amazon\AWSCLI\bin>aws s3api delete-object --bucket servercomputerbucket --key Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf

C:\Program Files\Amazon\AWSCLI\bin>aws s3 ls s3://servercomputerbucket

C:\Program Files\Amazon\AWSCLI\bin>aws s3api delete-bucket --bucket servercomputerbucket --region ap-south-1

C:\Program Files\Amazon\AWSCLI\bin>aws s3 ls
2018-12-18 08:31:47 arkit-test123
```

Create S3 Bucket

Bin> `aws s3 mb s3://servercomputerbucket`

make_bucket: servercomputerbucket

List S3 buckets

Bin> `aws s3 ls`

2018-12-18 08:31:47 arkit-test123

2018-12-20 16:57:01 servercomputerbucket

Upload Object to S3 Bucket

Bin> `aws s3 cp D:\Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf s3://servercomputerbucket`

upload: D:\Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf to s3://servercomputerbucket/Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf

List Objects in S3 Bucket

Bin> `aws s3 ls s3://servercomputerbucket`

2018-12-20 16:58:39 25173965 Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf

Delete Object from S3 bucket

Bin> `aws s3api delete-object --bucket servercomputerbucket --key Red_Hat_Enterprise_Linux-7-System_Administrators_Guide-en-US.pdf`

Bin> `aws s3 ls s3://servercomputerbucket`

Delete S3 bucket

Bin> `aws s3api delete-bucket --bucket servercomputerbucket --region ap-south-1`

25. Creating EC2 Instance using AWS CLI

Before creating an EC2 instance using AWS CLI collect few details

- AMI ID
- Instance Type
- Key Name (If there is no Key Pair create one)
- Security Group ID
- Subnet ID

Bin> `aws ec2 run-instances --image-id ami-06bcd1131b2f55803 --count 1 --instance-type t2.micro --key-name KEYNAME --security-group-ids sg-857f92e9 --subnet-id subnet-51e5592c`

26. Few AWS Articles

- Mount S3 Bucket in Linux using S3FS
- Use S3 Bucket as Windows Local Drive
- AWS Basic Interview Questions and Answers
- AWS Certification course Content
- List all AWS Instances from All Regions
- How To create your First Free Tier AWS Account
- AWS Add New User Accounts with SSH Access Linux Instance
-

27. AWS Services and abbreviations

- S3 – Simple Storage
- EC2 – Elastic Compute Cloud

- EBS – Elastic Block Storage
- EFS – Elastic File System
- ECS – Elastic Container Service
- EKS – Elastic Container Service for Kubernetes
- RDS – Amazon Relational Database Service
- IAM – Identity, Access Management
- VPC – Virtual Private Cloud (isolated Network)
- ELB – Elastic Load Balancer
- EMR – Elastic MapReduce
- MSK – Managed Streaming for Kafka
- SQS – Amazon Simple Queue Service
- SNS – Amazon Simple Notification Service
- SES – Amazon Simple Email Service
- ECR –Amazon Elastic Container Registry
- SWF – Amazon Simple Workflow Service