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

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

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 this conditions before creating a free tier account.

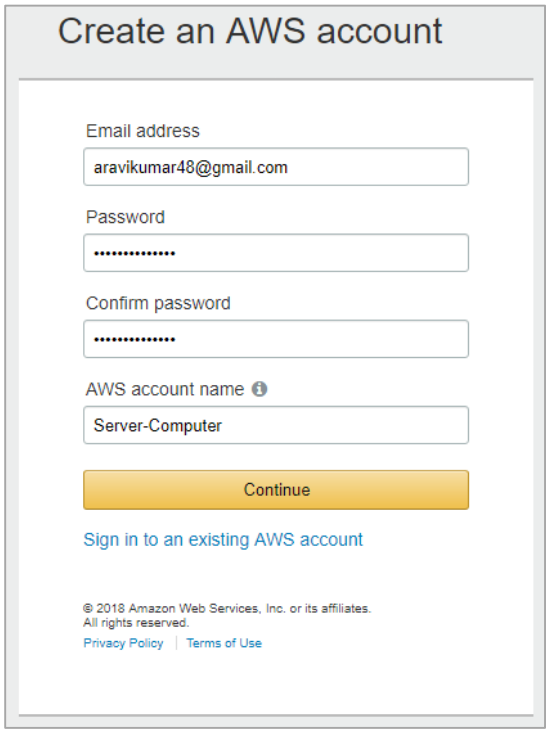
https://aws.amazon.com/free/

Prerequisites:

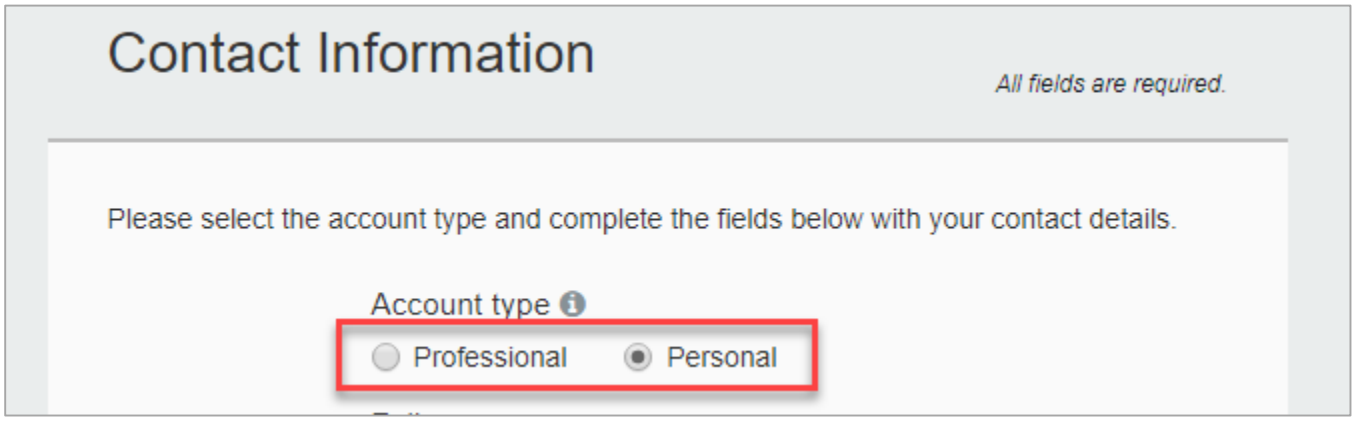
- Credit card with minimum 1$ available balance
- Reachable mobile number for verification

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

Click on 

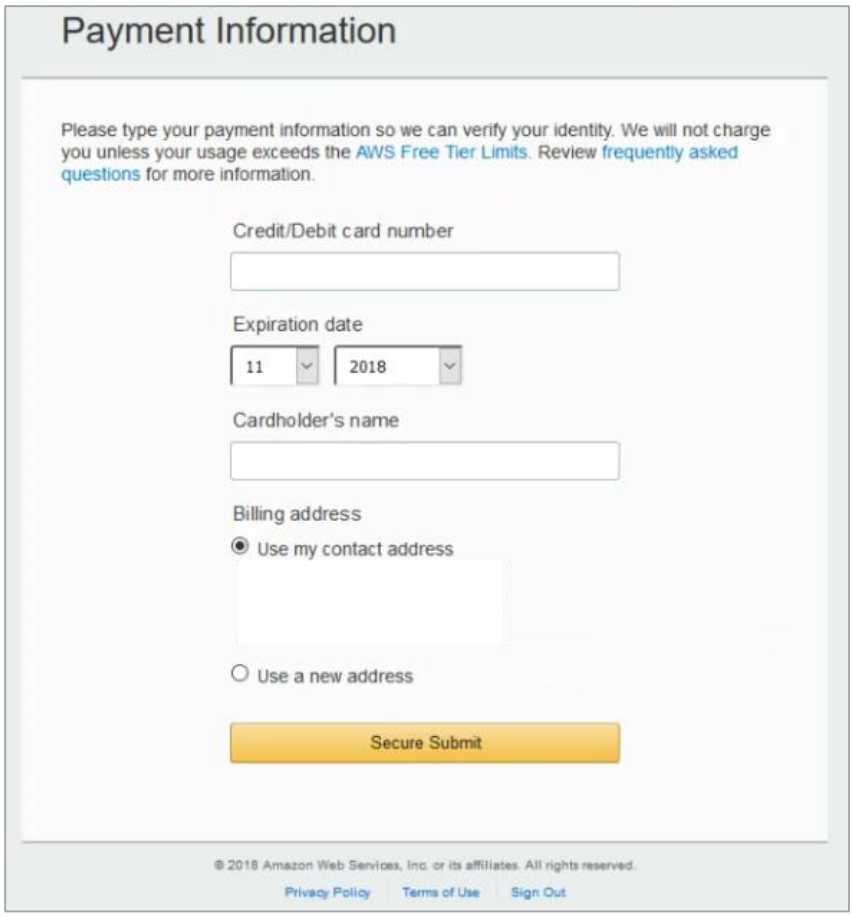


Fill the details example is shown above and **click continue**



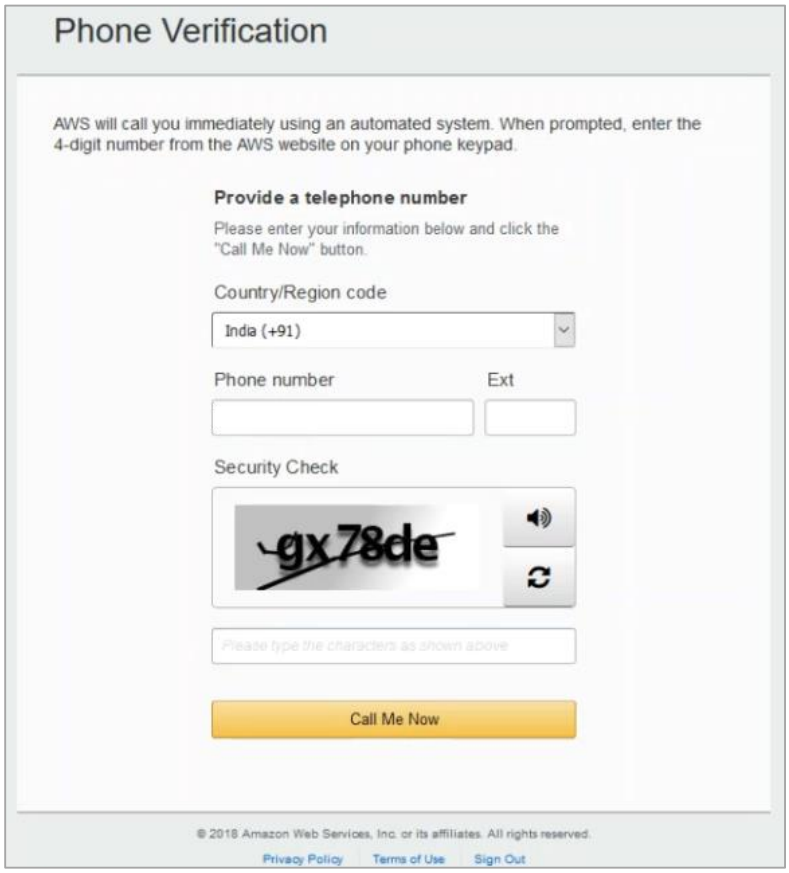
Click on radio button

- 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**

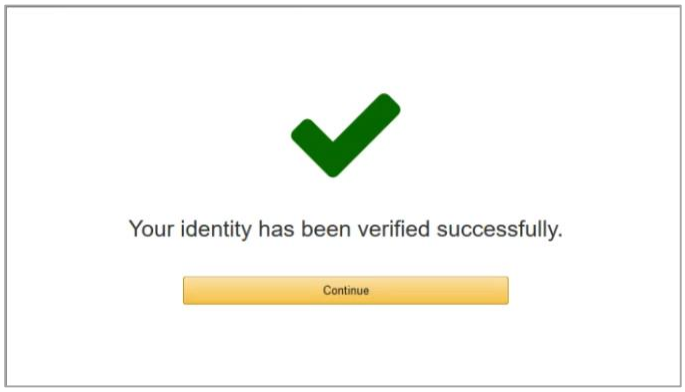


It will ask you to enter phone number, Security check then click on **Call Me Now**

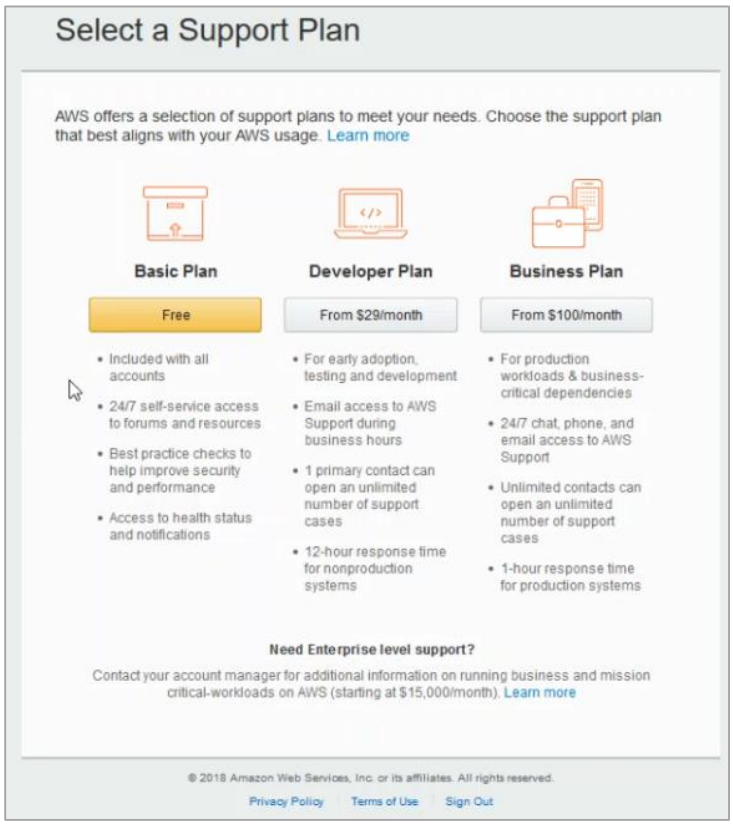
You will receive a call from AWS tele communication and ask you to enter the code displayed on screen.

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**

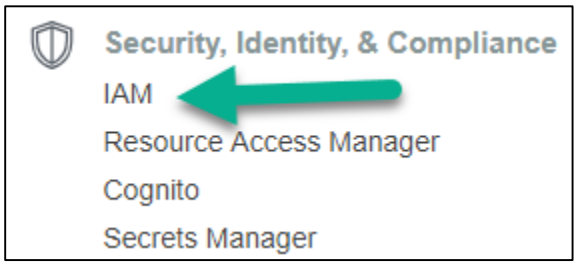
You successfully completed Free Tier Account Creation. Login and Enjoy AWS Free Tier.

AWS Console

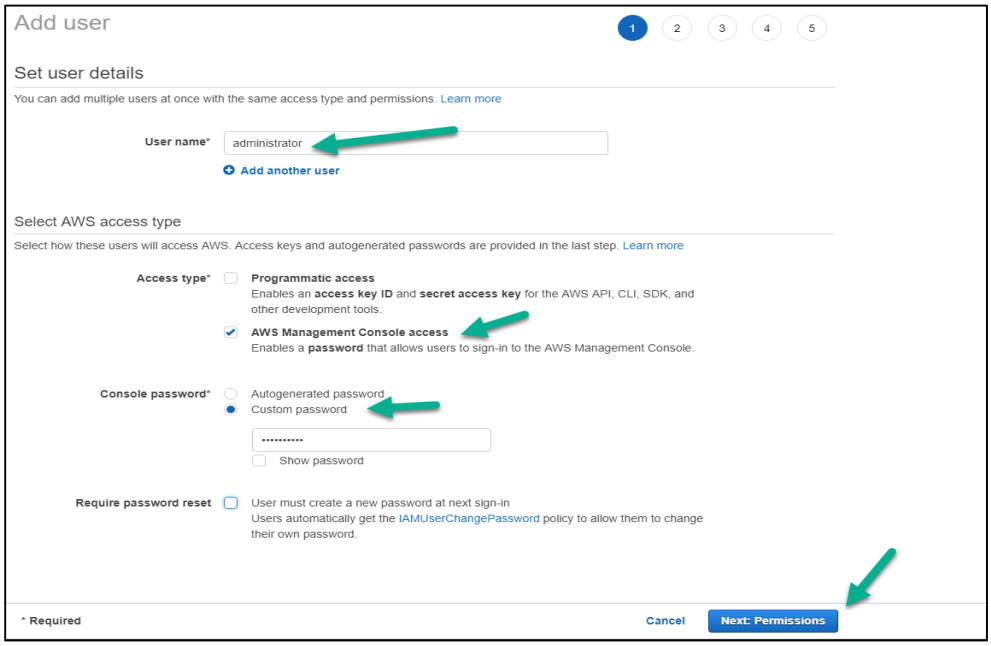
Provide your email address and password to **Sign In**

5. Enabling Multi-Factor Authentication to Secure Your Access

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



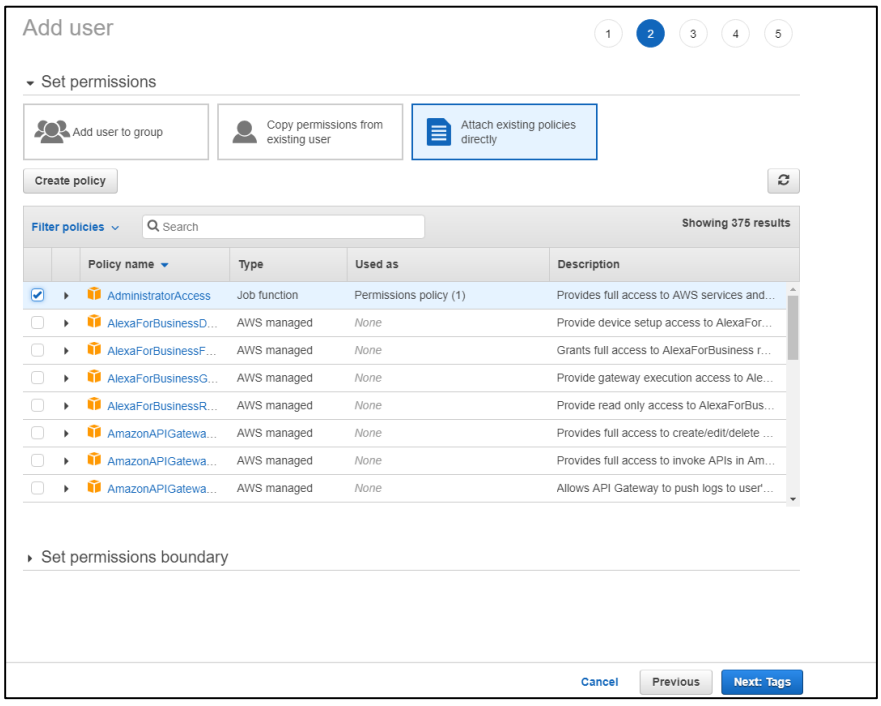
Click on Users → Add User



Provide user name, select access type

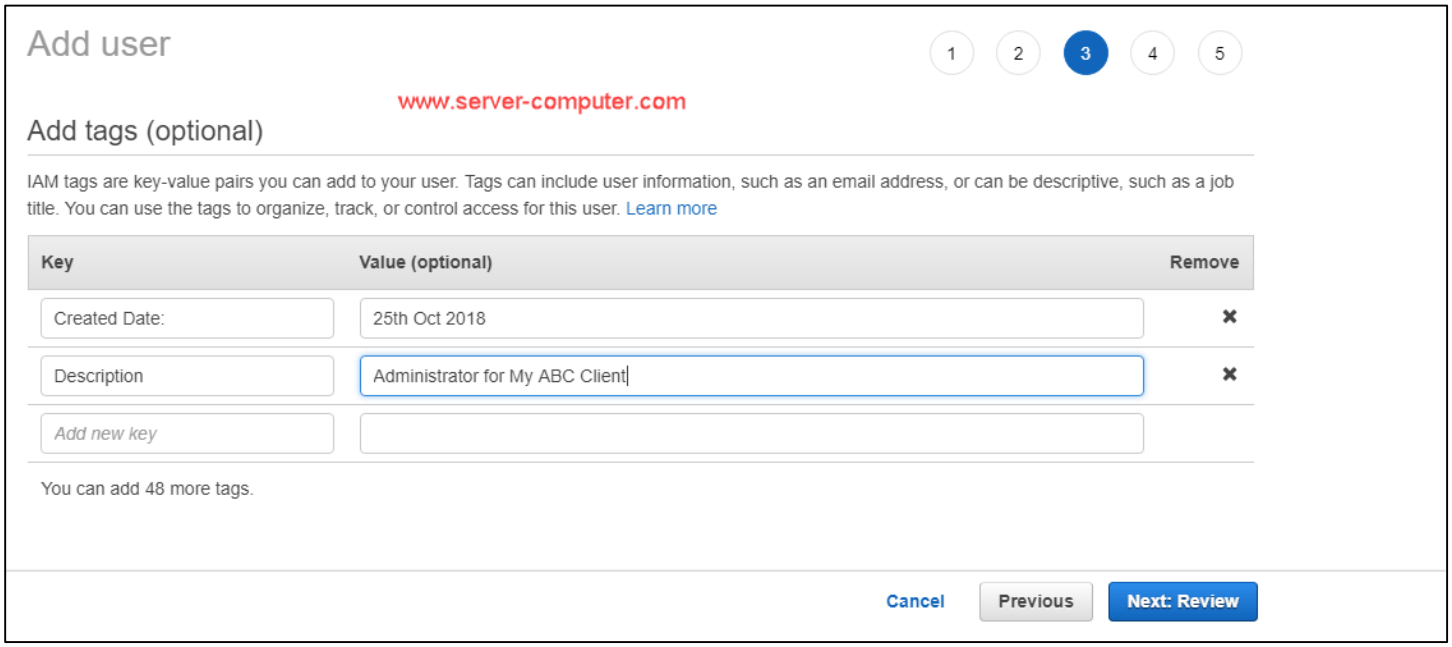
- Programmatic Access – Required for automation, run any operation using programs
- AWS Management Console Access – User will have web console access

Click **Next Permissions**

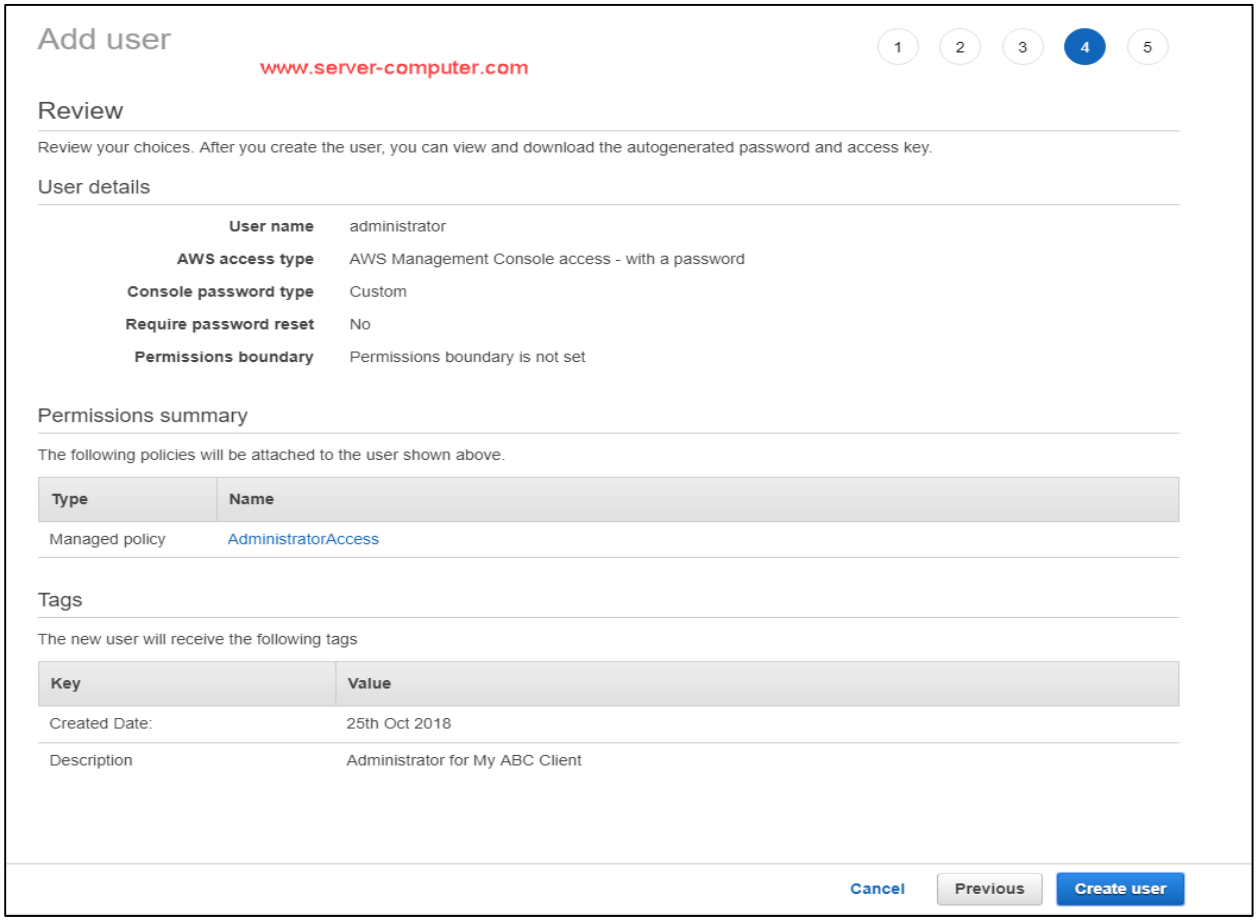


Click **Next: Tags**

Add tags whatever required to identify user



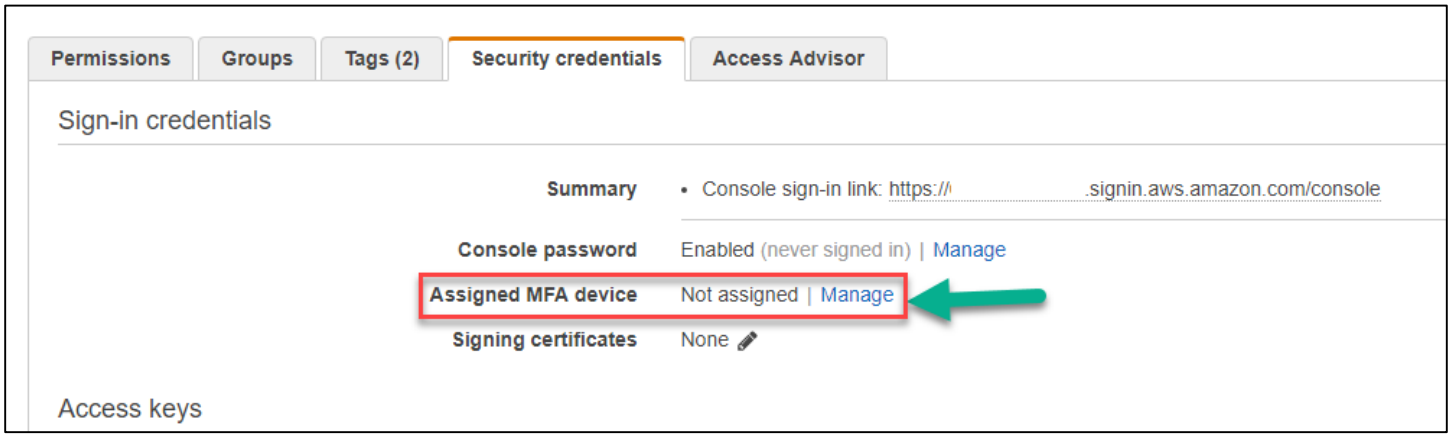
Click **Next: Review**



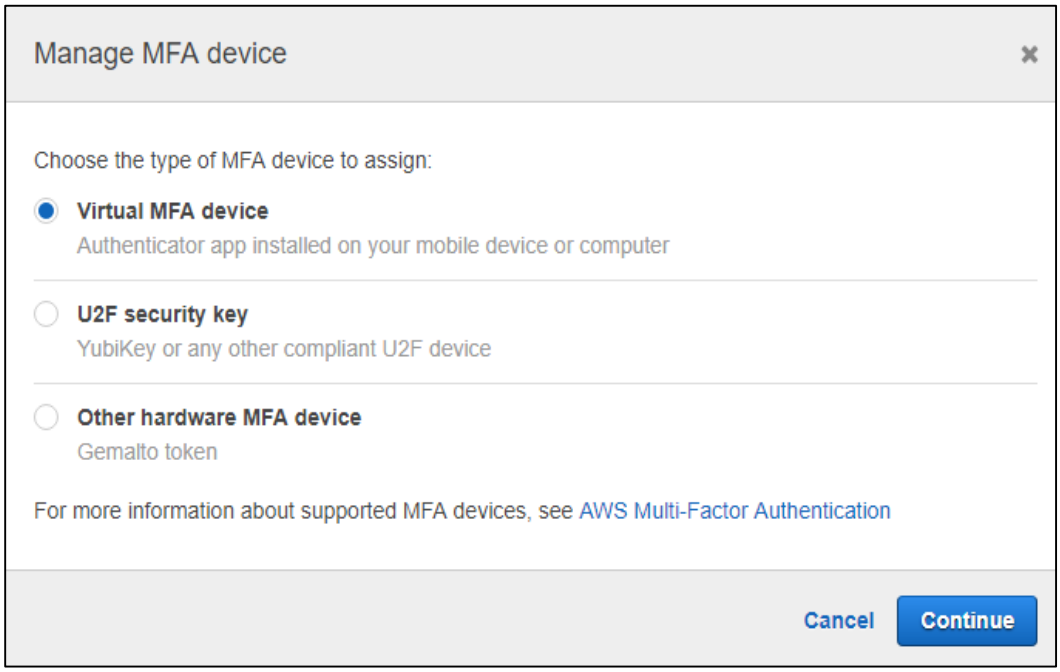
Click **Create User**

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



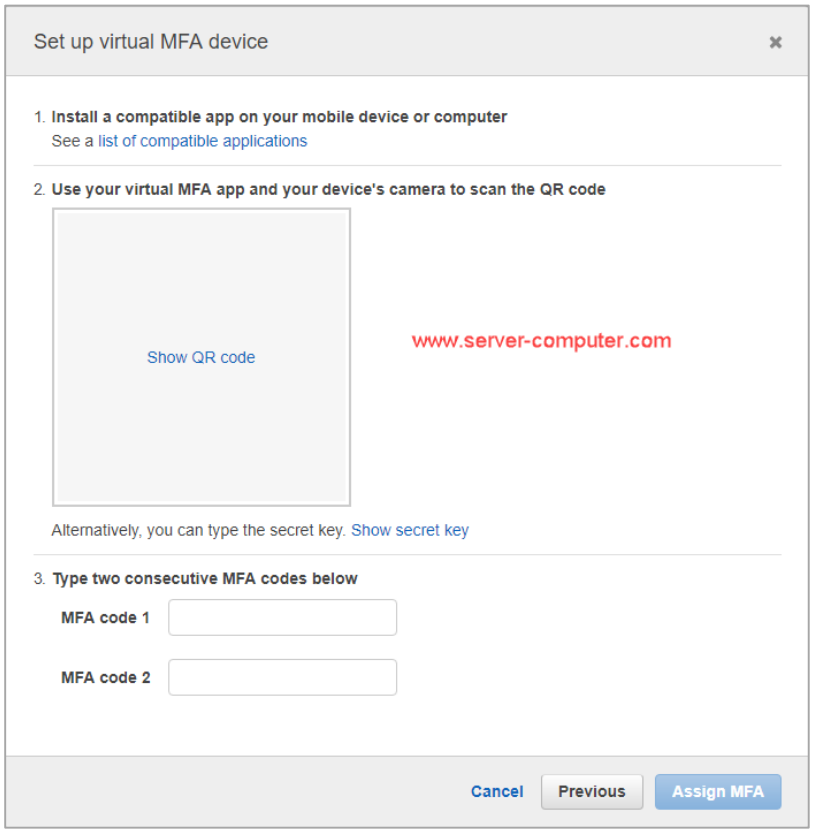
Click on Assigned MFA Device – Manage



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**



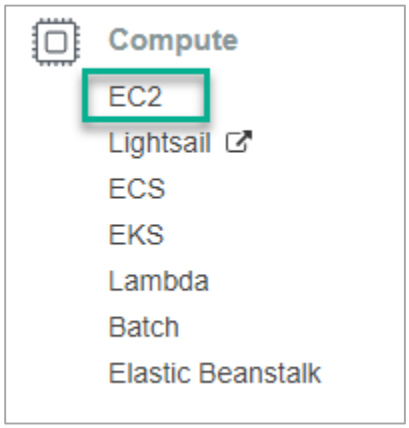
Click in Show QR Code and scan the same code from your Google authenticator App. It will generate 6 digit codes enter one code in first MFA code1 and second one in MFA Code 2 Click on **Assign MFA**

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

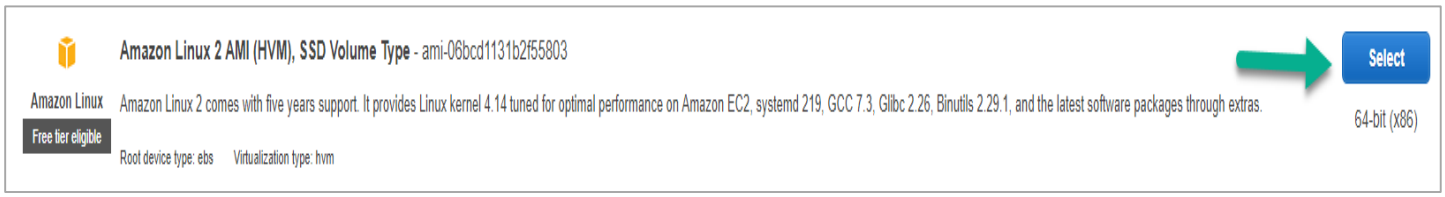
Here after if you want to login you have to enter credentials and MFA code to Login.

6. Creating First Linux Instance

Login to AWS console, services drop down click on EC2



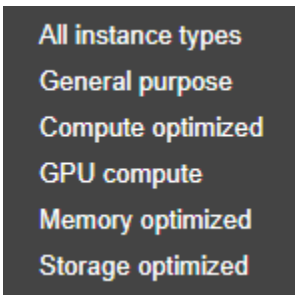
Click on **Launch instance**

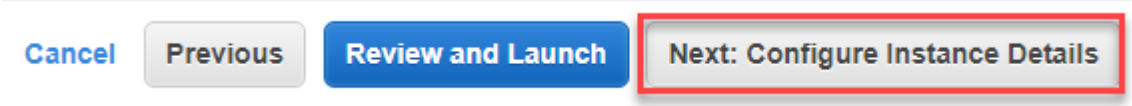


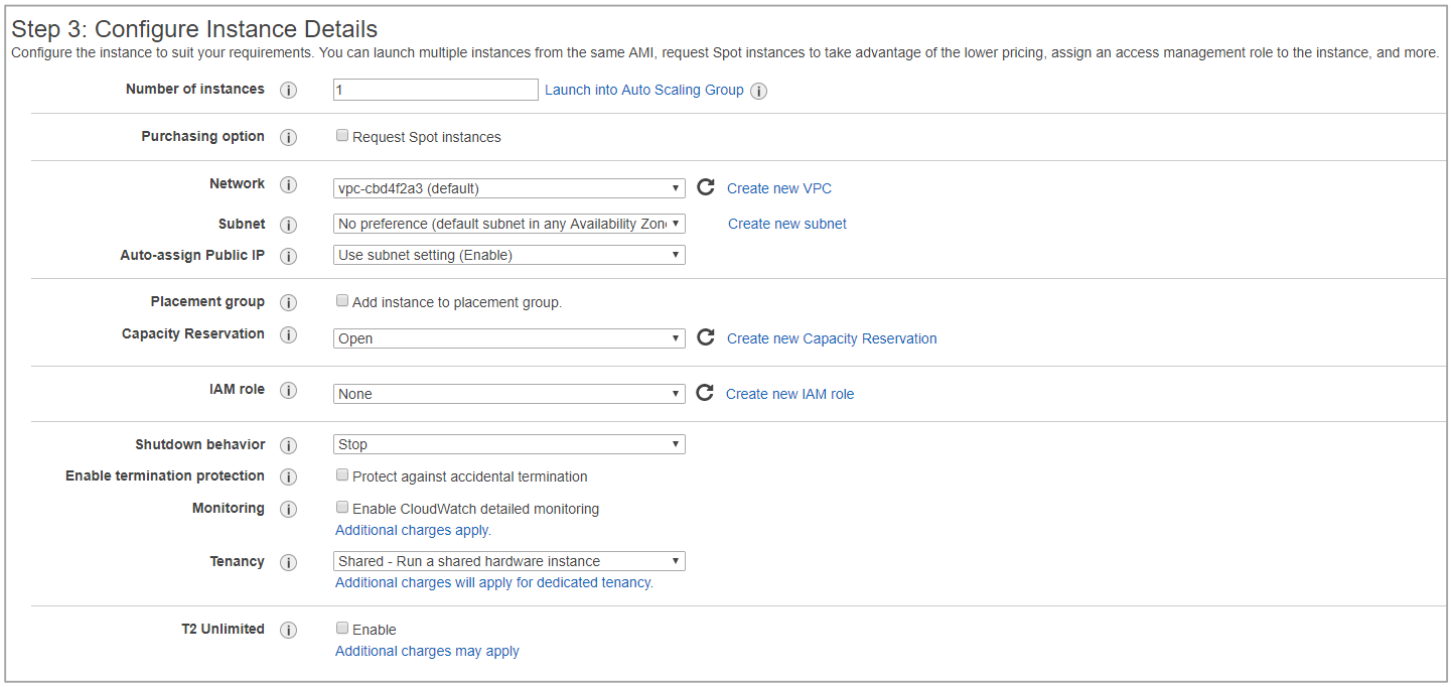
I am selecting Free Tier instance Amazon Linux

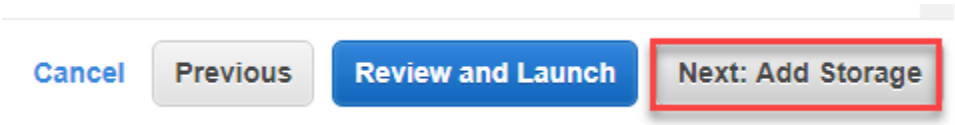
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

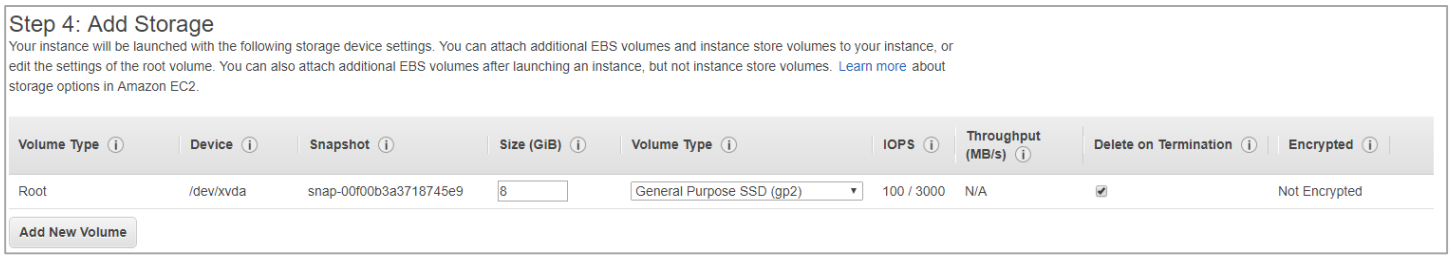
We have below types of instances



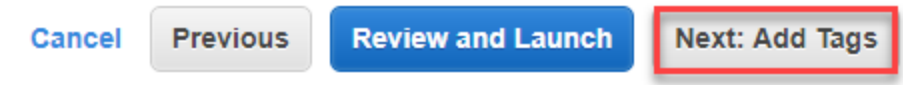




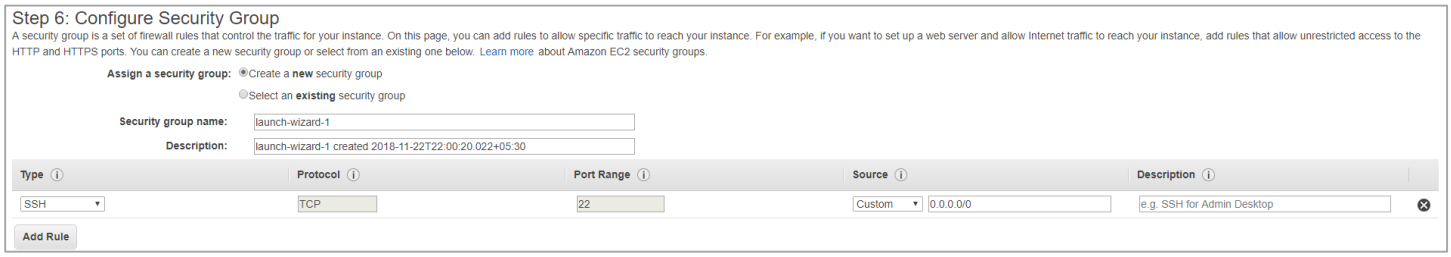




Add storage – EBS Elastic Block Storage volume will attached to your instance

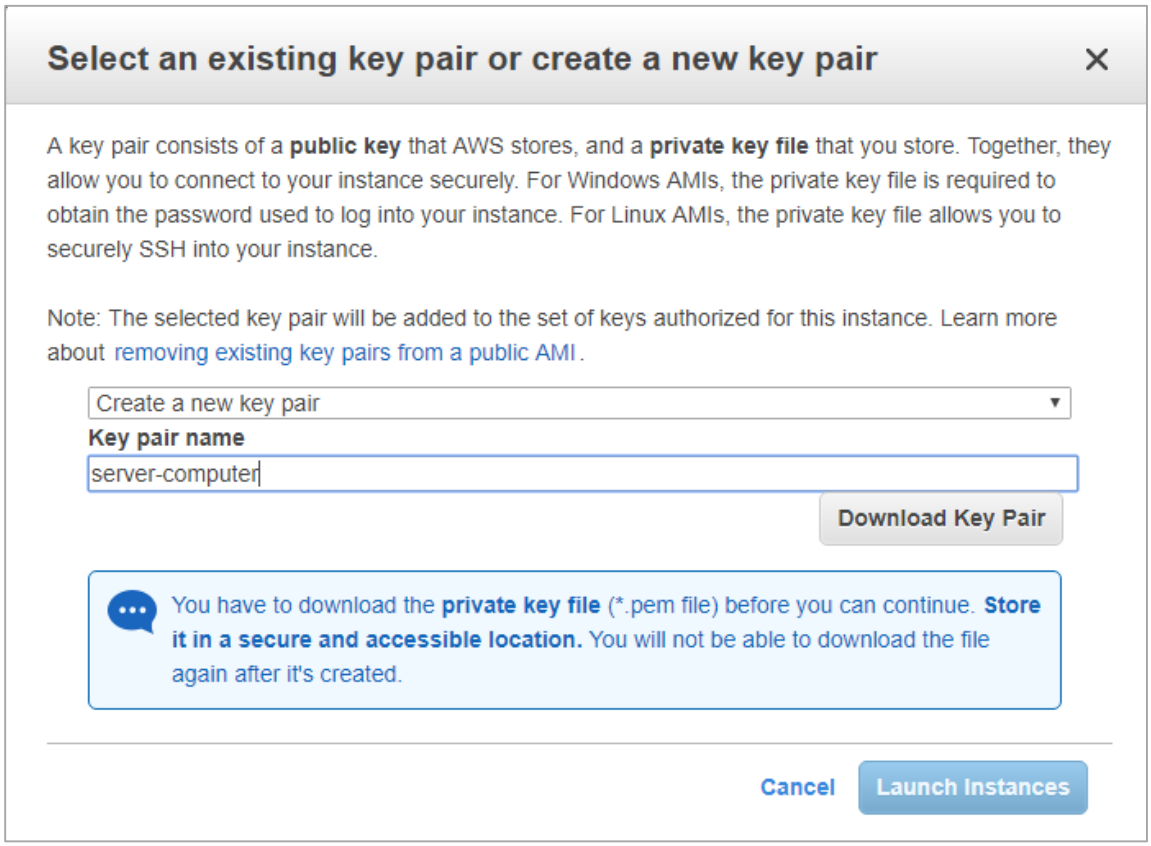


Tags to identify the details about instance (Production/Test/Dev/Client Name)



Using security group we can allow/deny any ports

Verify the details and click on **Launch**



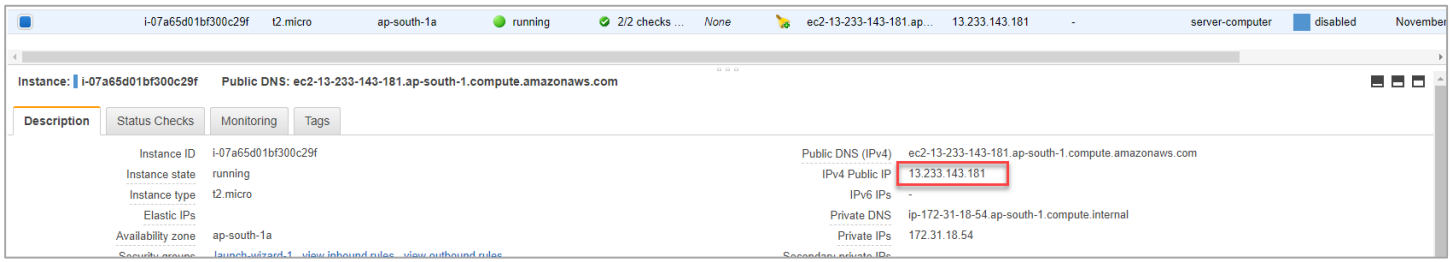
For the first time you **create a new key pair** and **Download Key Pair**

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

Launch Instances

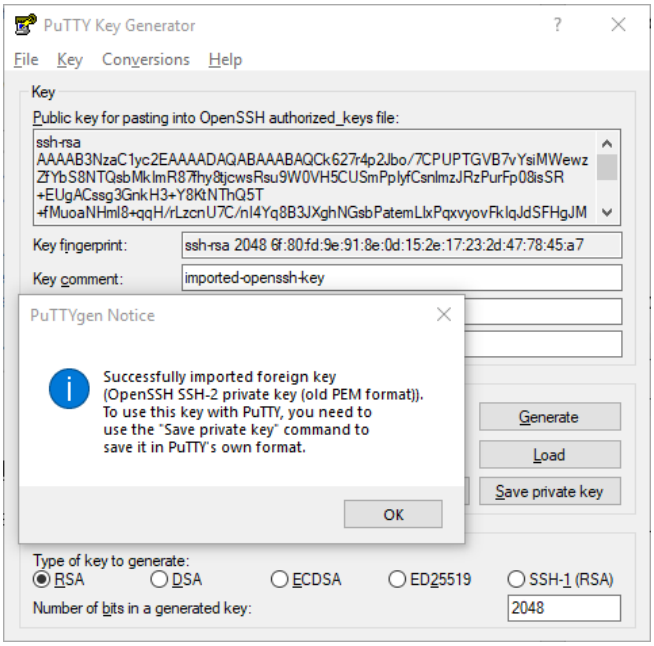
Go to EC2 → See the instances

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

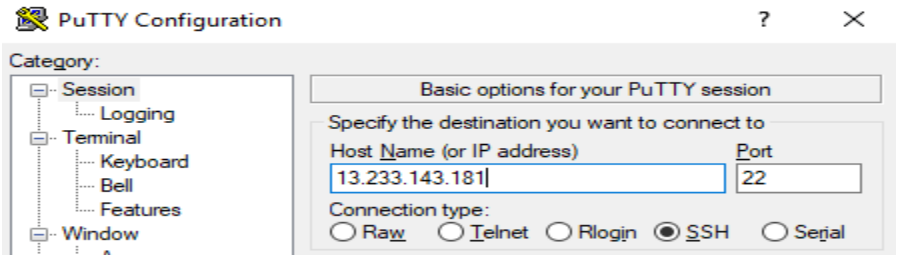


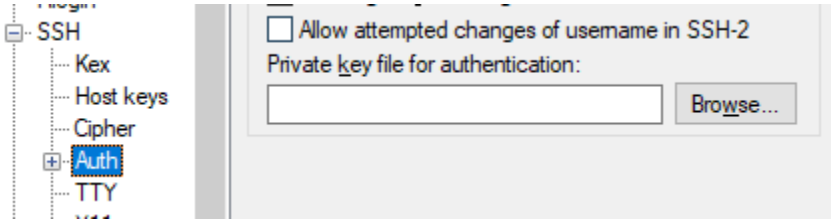
Click Ok.

Save Private Key

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

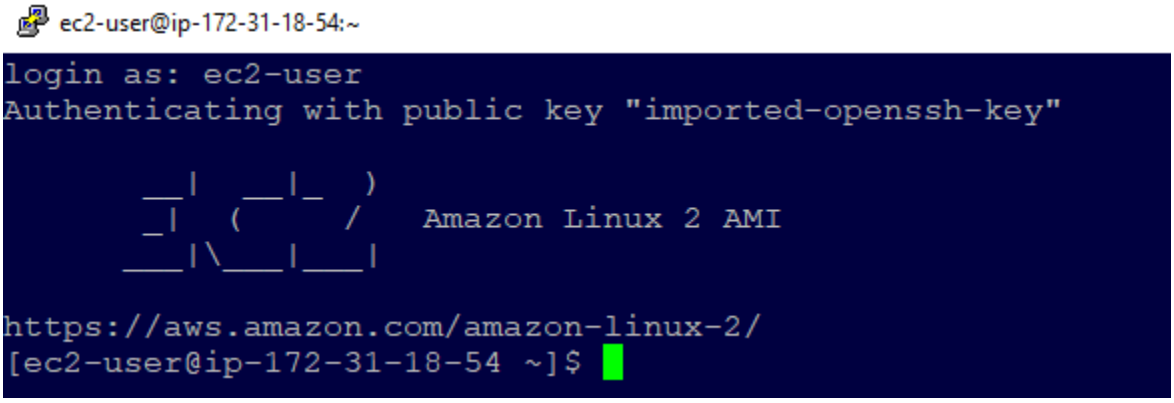
Open putty application and type IP address as shown below





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

```
sudo yum update

sudo yum install httpd

sudo service httpd start

sudo service httpd status

sudo chkconfig httpd on
```

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

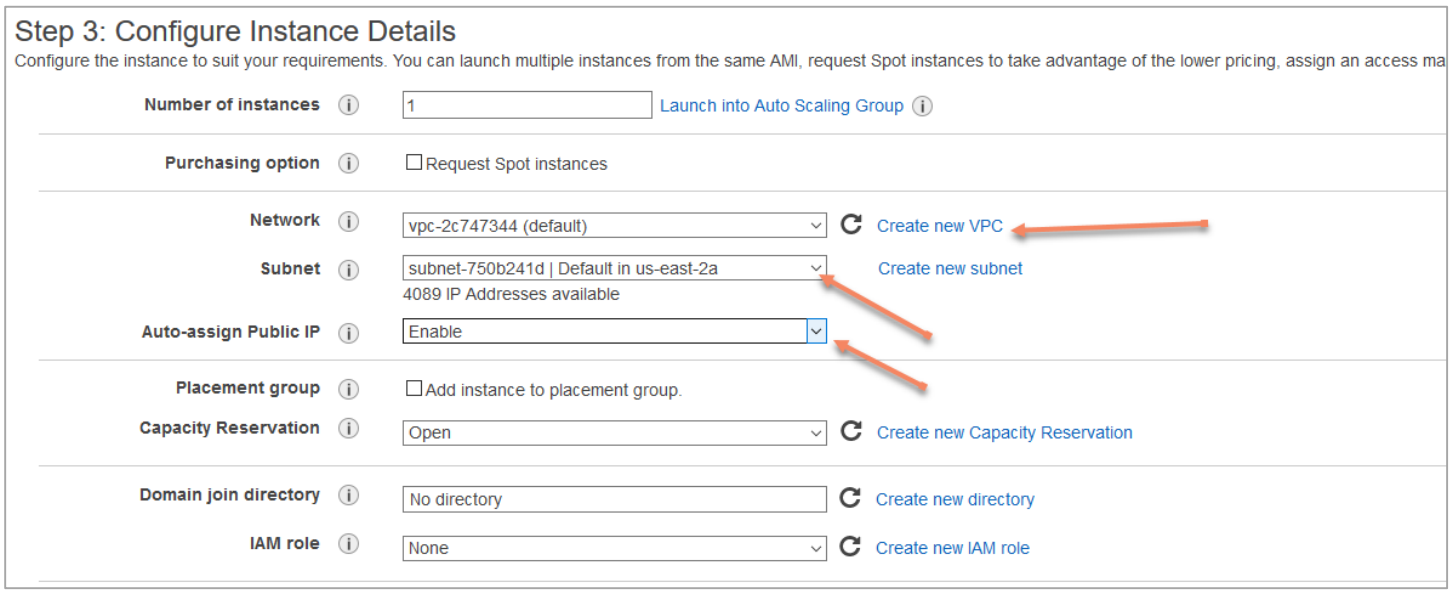
Open browser and type your instance public IP address you can access web-server test page.

7. 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** →



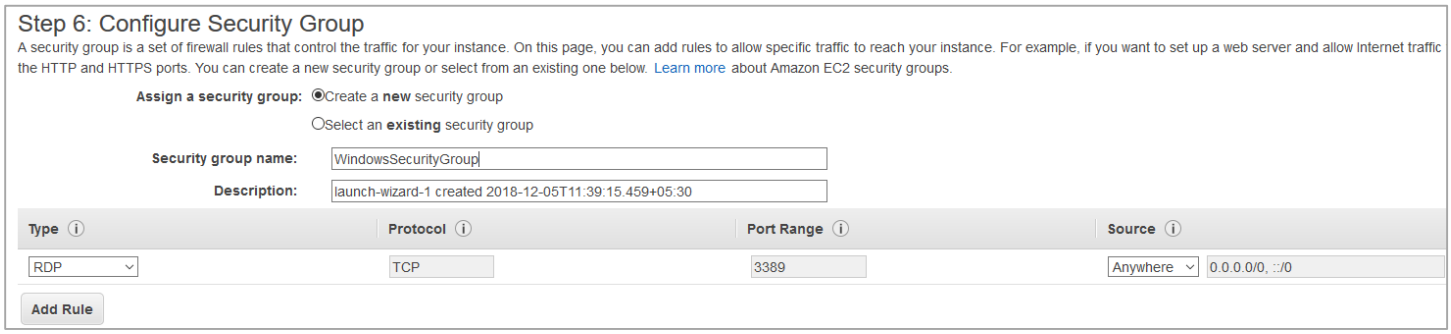
Select VPC, subnet and enable Public IP address.

Click **Next: Add Storage**

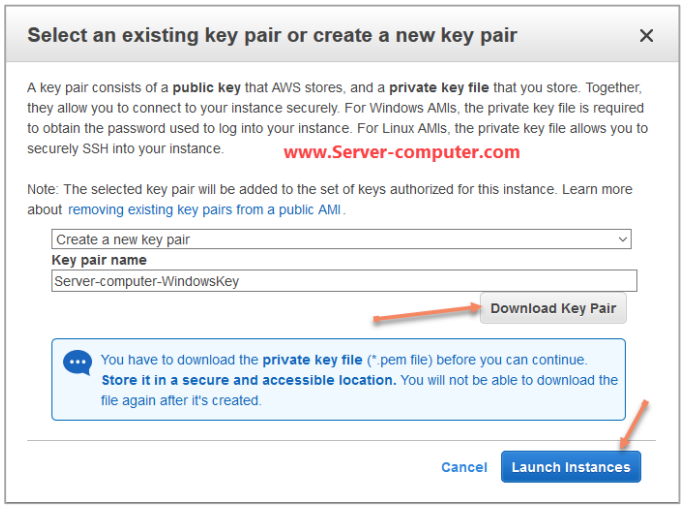
Click **Next: Add Tags**

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

Click **Next: Configure Security Group**



Click **Review and Launch**

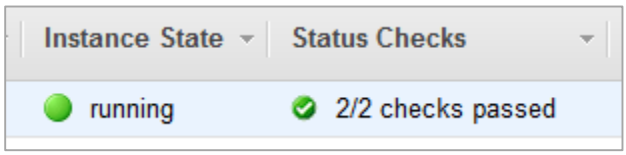


Download Key Pair and **Launch Instance**

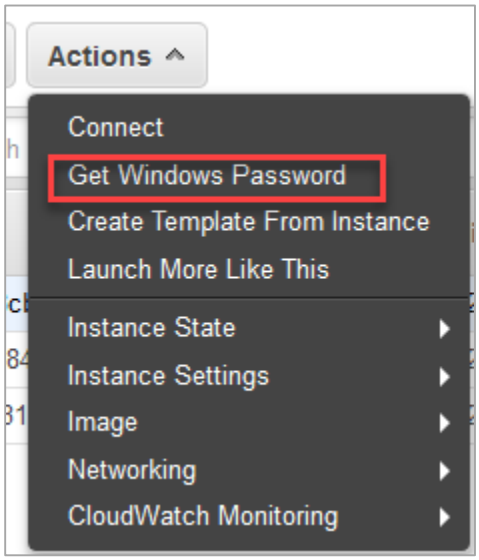
Note: Wait 4 Minutes instance to launch

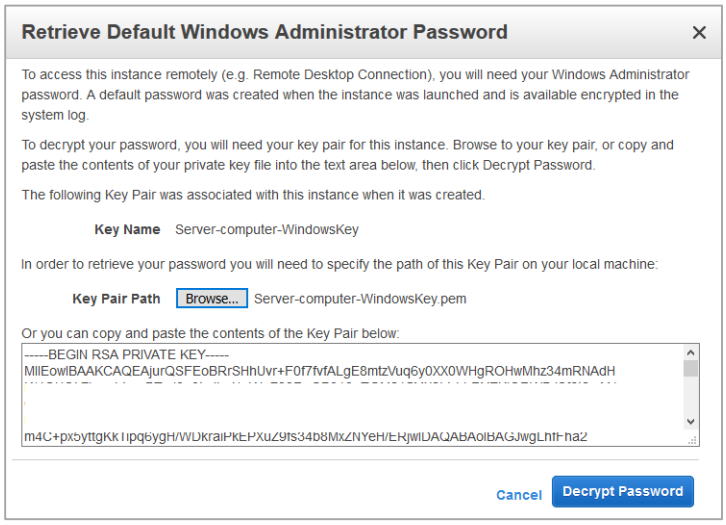
It should display the following:

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

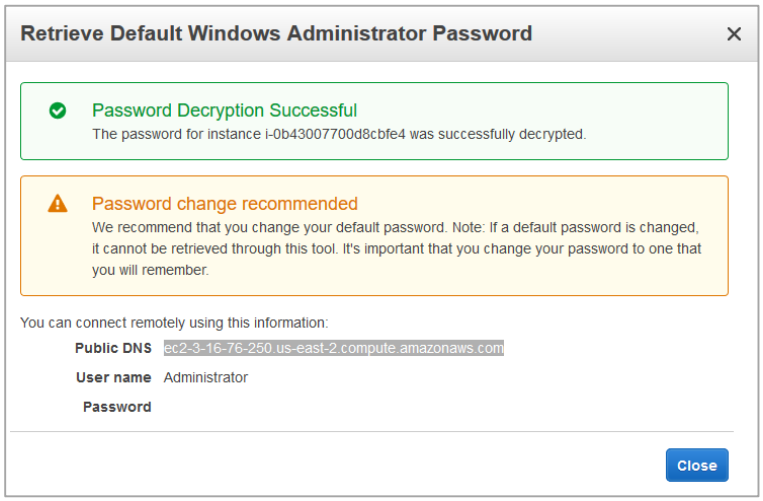


Select instance you have launched → Actions



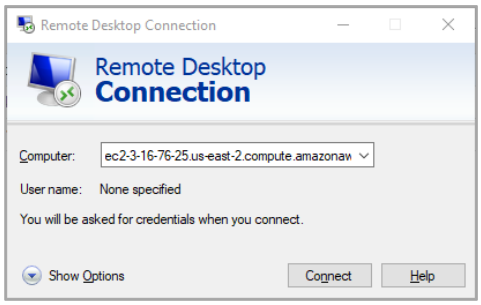


Browse server-computer-WindowsKey.pem file to decrypt and get password



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

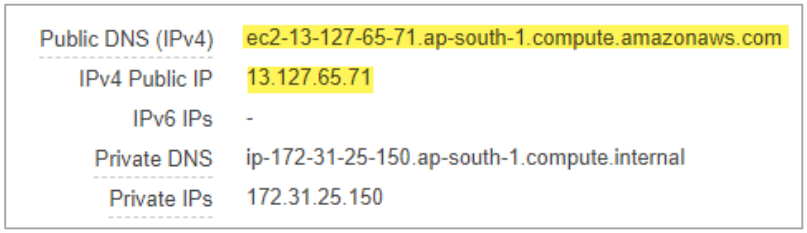
Go to your windows machine Start → Run → mstsc → Ok



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

8. 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



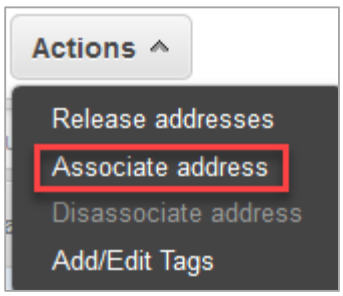
However, after stop and start of instance assigned public IP address will release to the amazon free pool

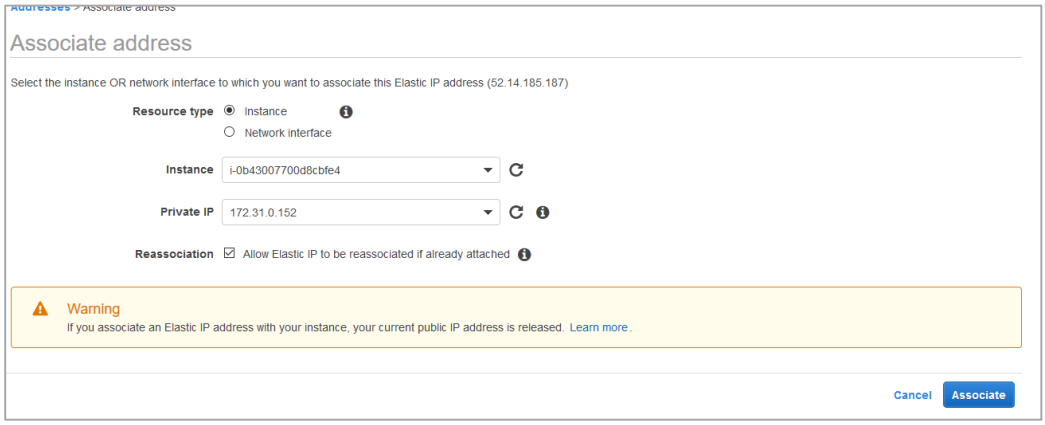
If would like to assign an static public address then navigate to Elastic IP's

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





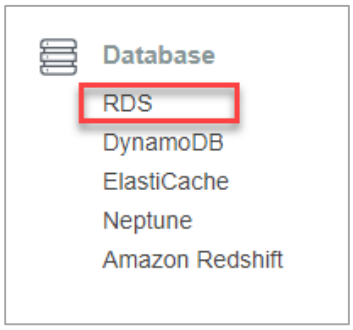
Select Instance ID check Instance ID before allocating. Click **Associate**

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.

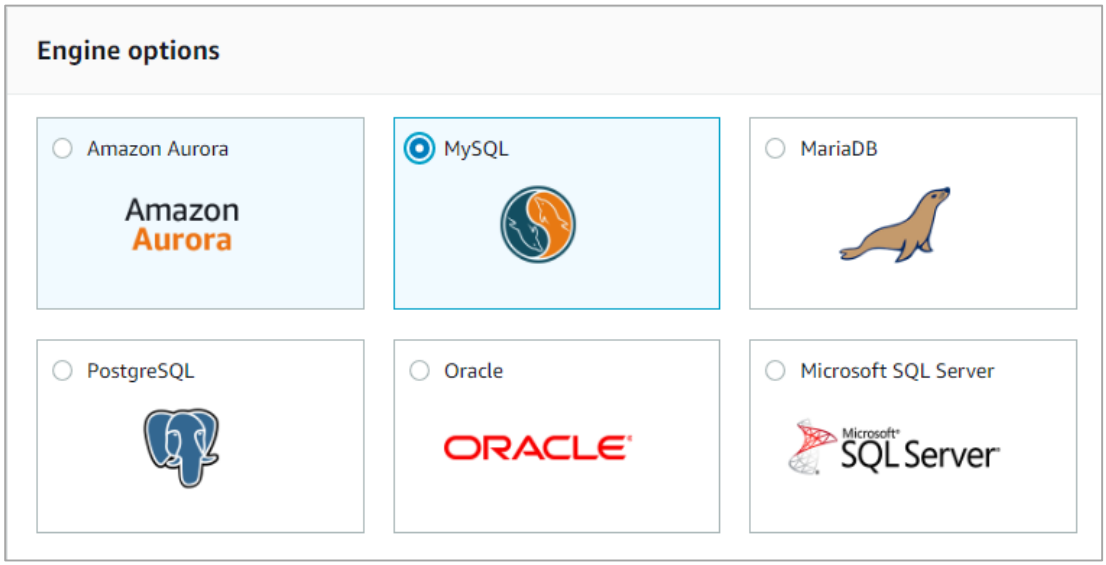
9. Launching RDS Instance

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



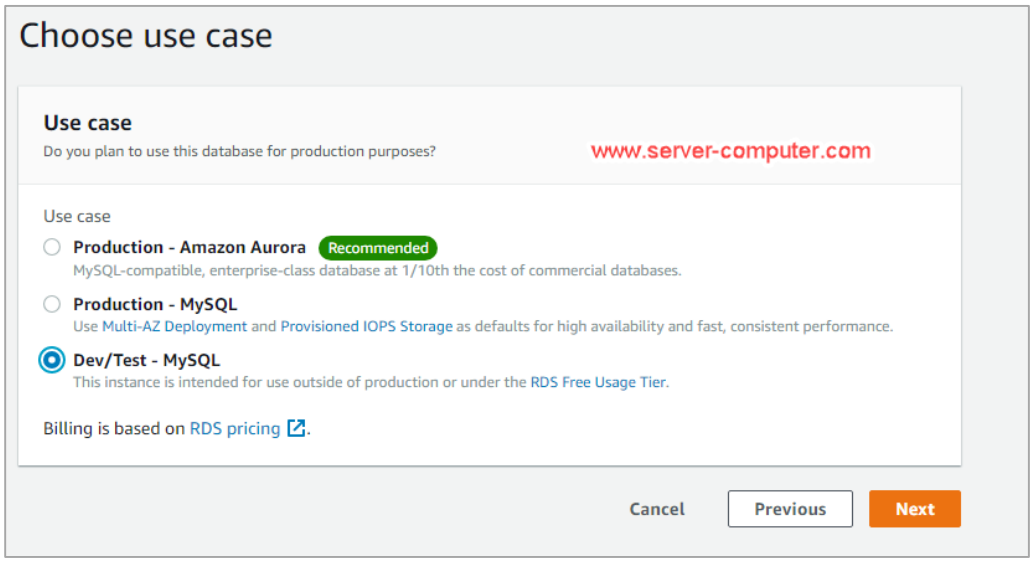
Now we are going to create a new Database instance with empty database

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

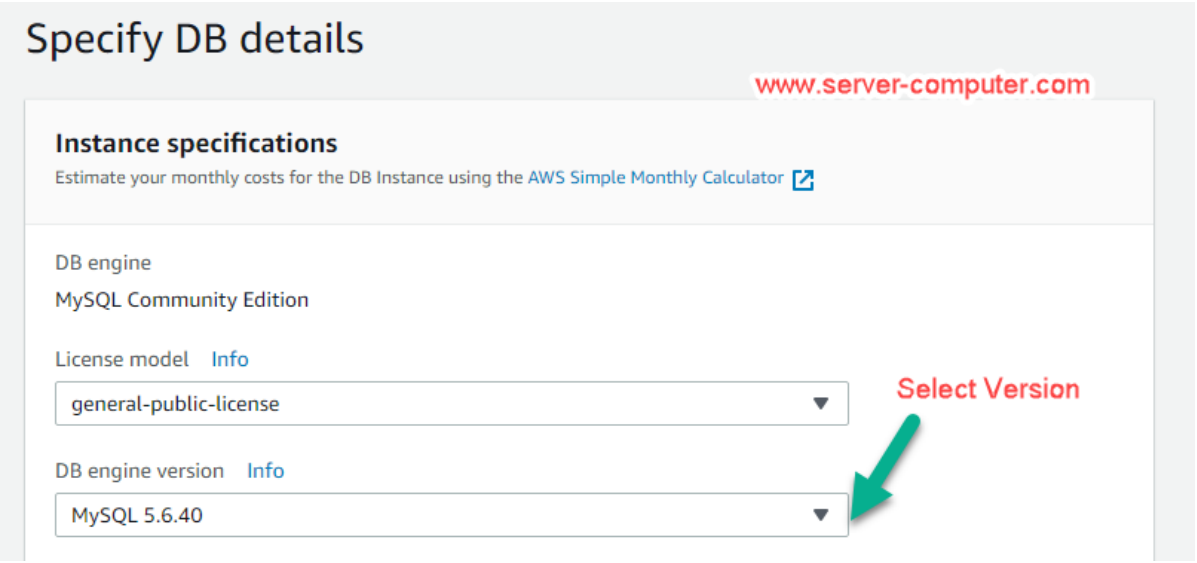


Select any one of the database engine, which you want to launch and Click **Next**

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

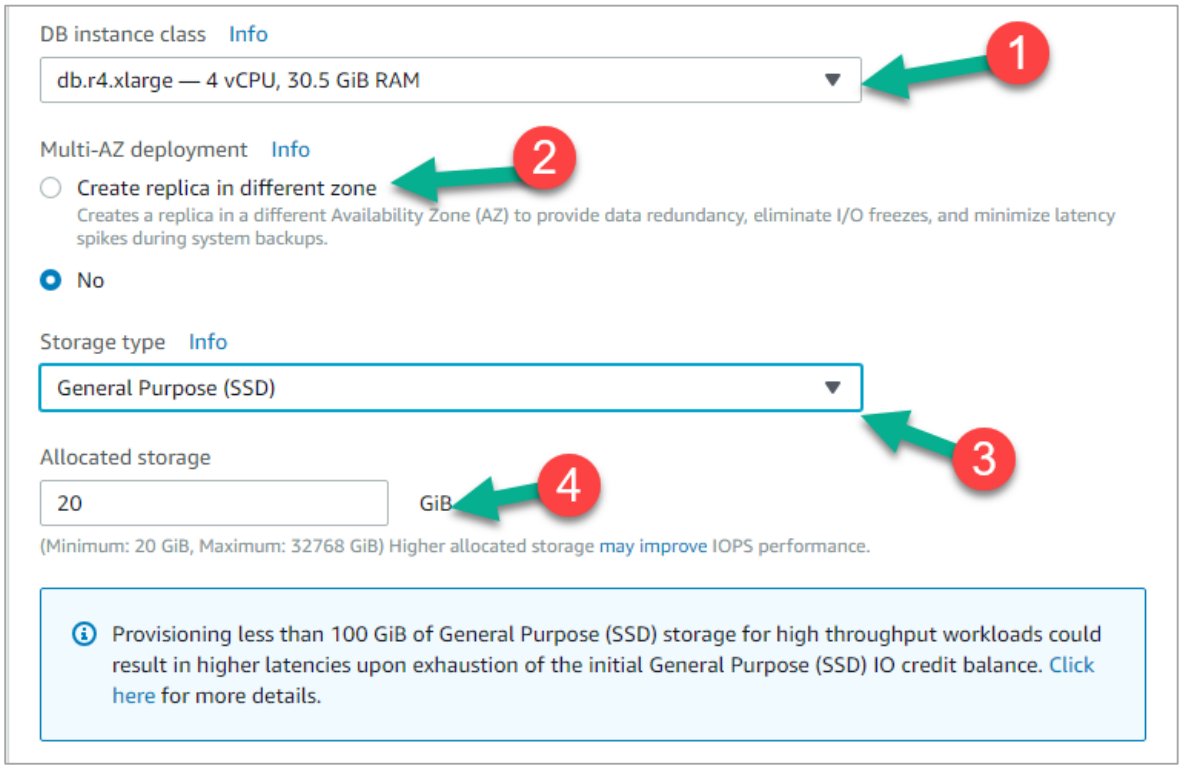


Choose appropriate usage of your instance. In this scenario, I am using Dev/Test instance Click **Next**



In drop down, select appropriate and required MySQL Version.

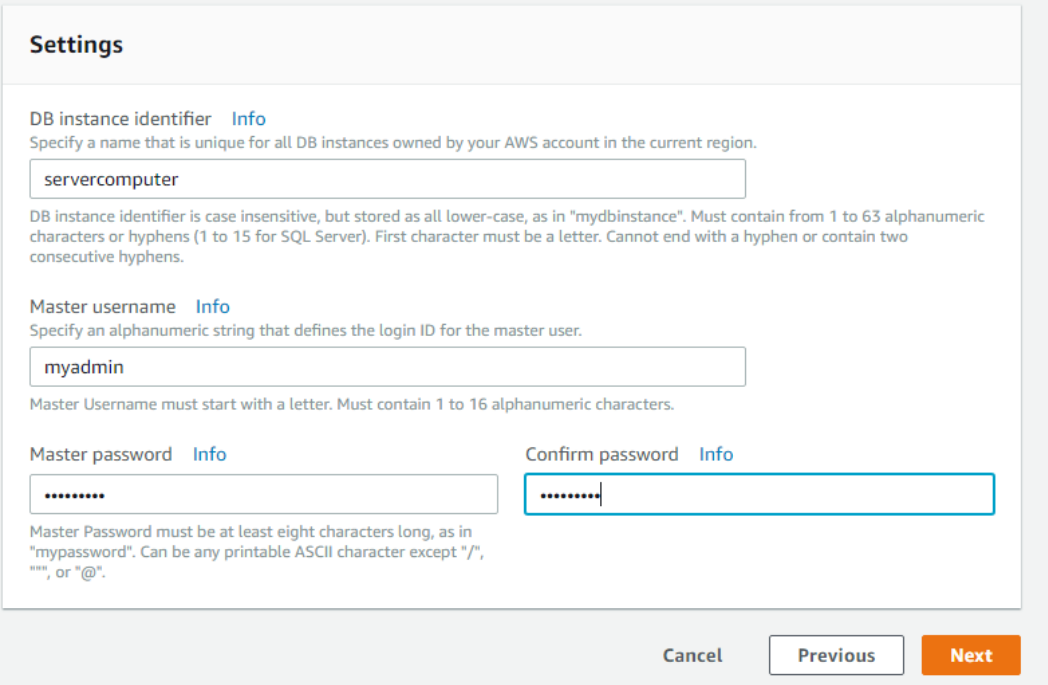
Note: If you select Free Tier. Selected version and options will overwritten free options.



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

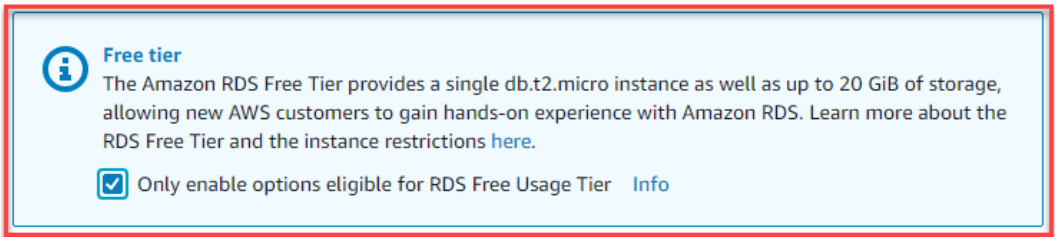
b. Provisioned IOPS is for most read/write operations

4. Size of the storage

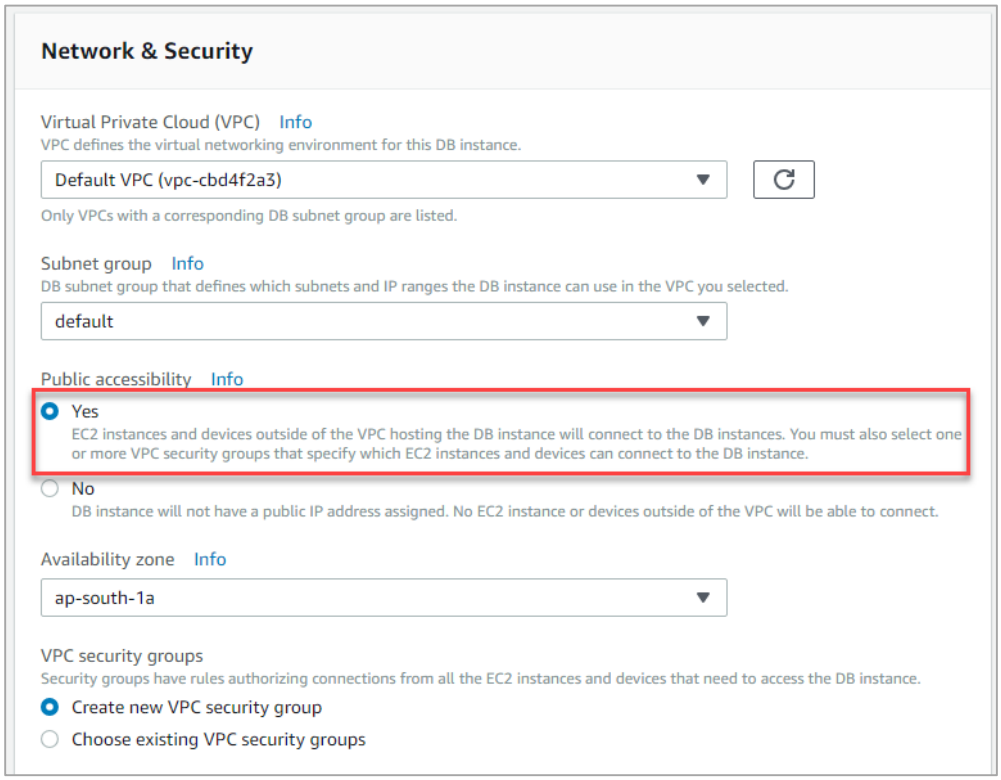


Provide

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



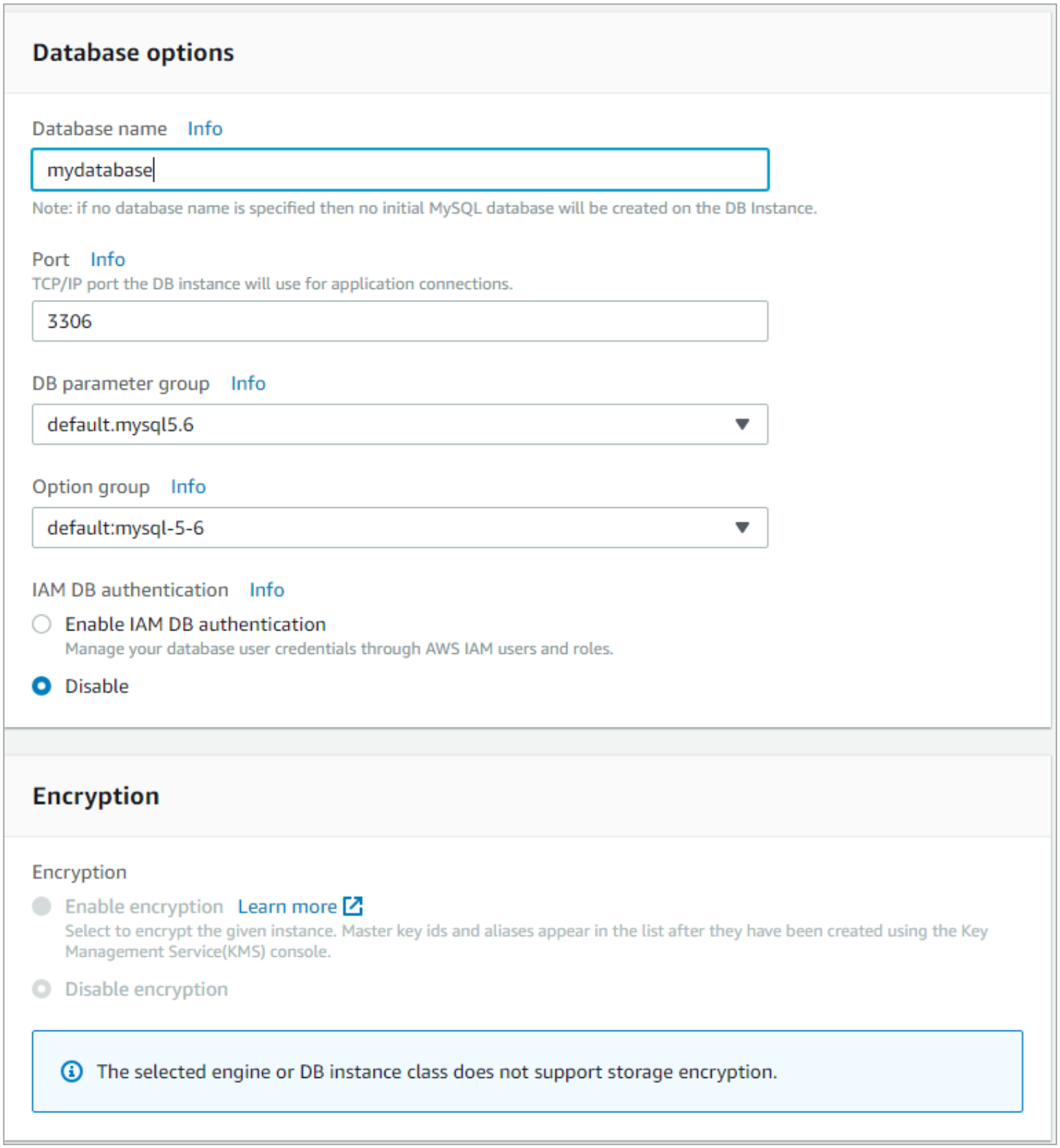
DO NOT FORGOT TO SELECT IF YOU'RE USING FREE TIER OTHERWISE YOU WILL BE CHARGED



Select appropriate VPC and Subnet group (If any)

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

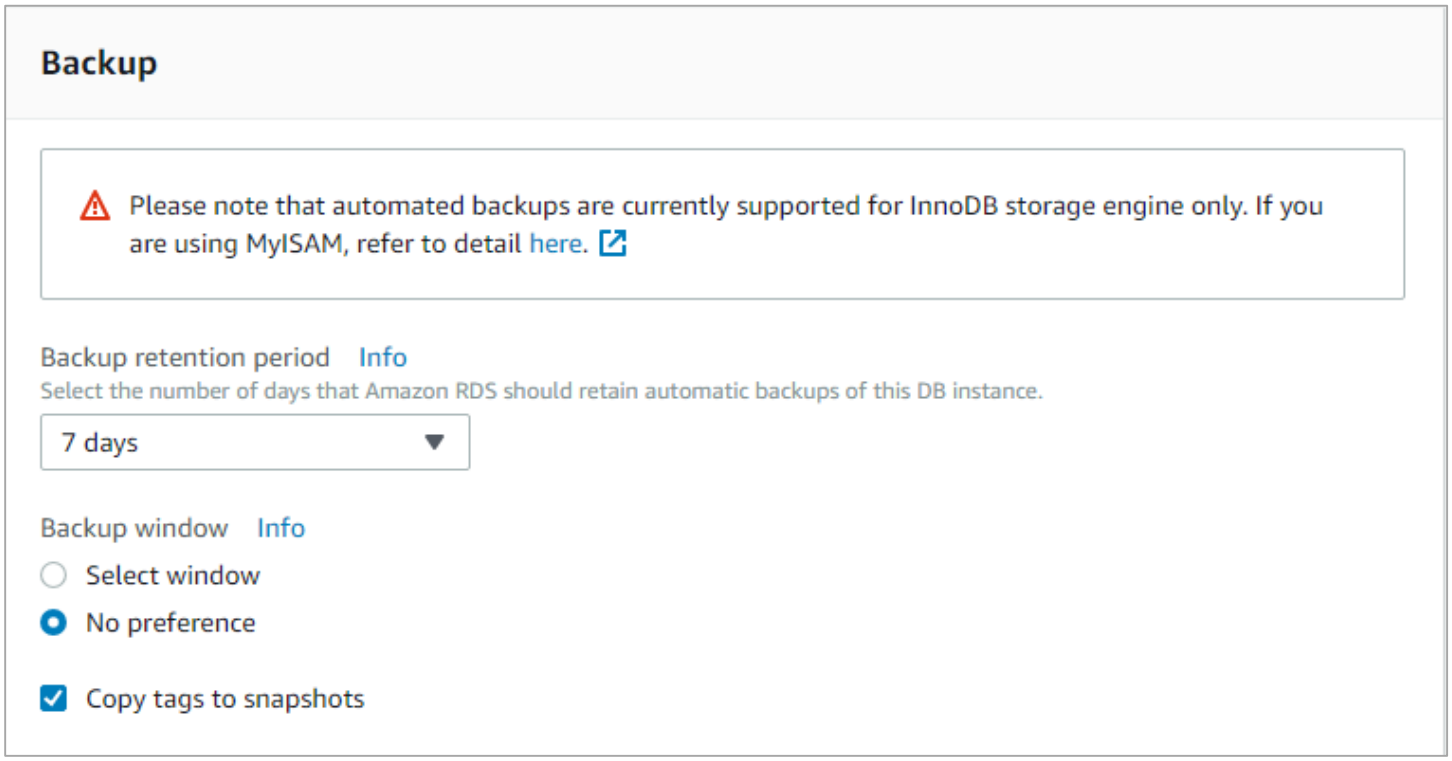
Choose existing VPC security groups if you have already or it will create new security group for this instance access.



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

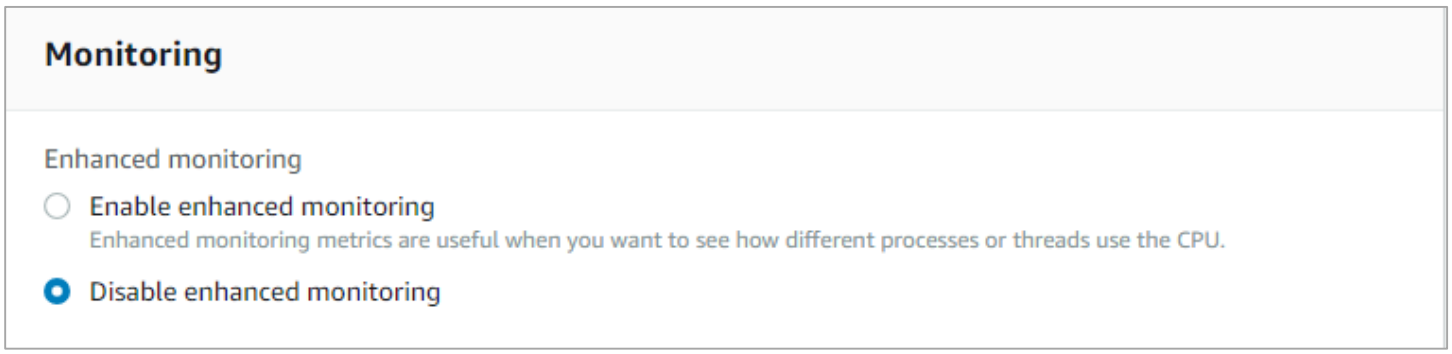
Enabling IAM DB Authentication. IAM Users also can access your instance based on IAM policies.

For free tier encryption option is disabled

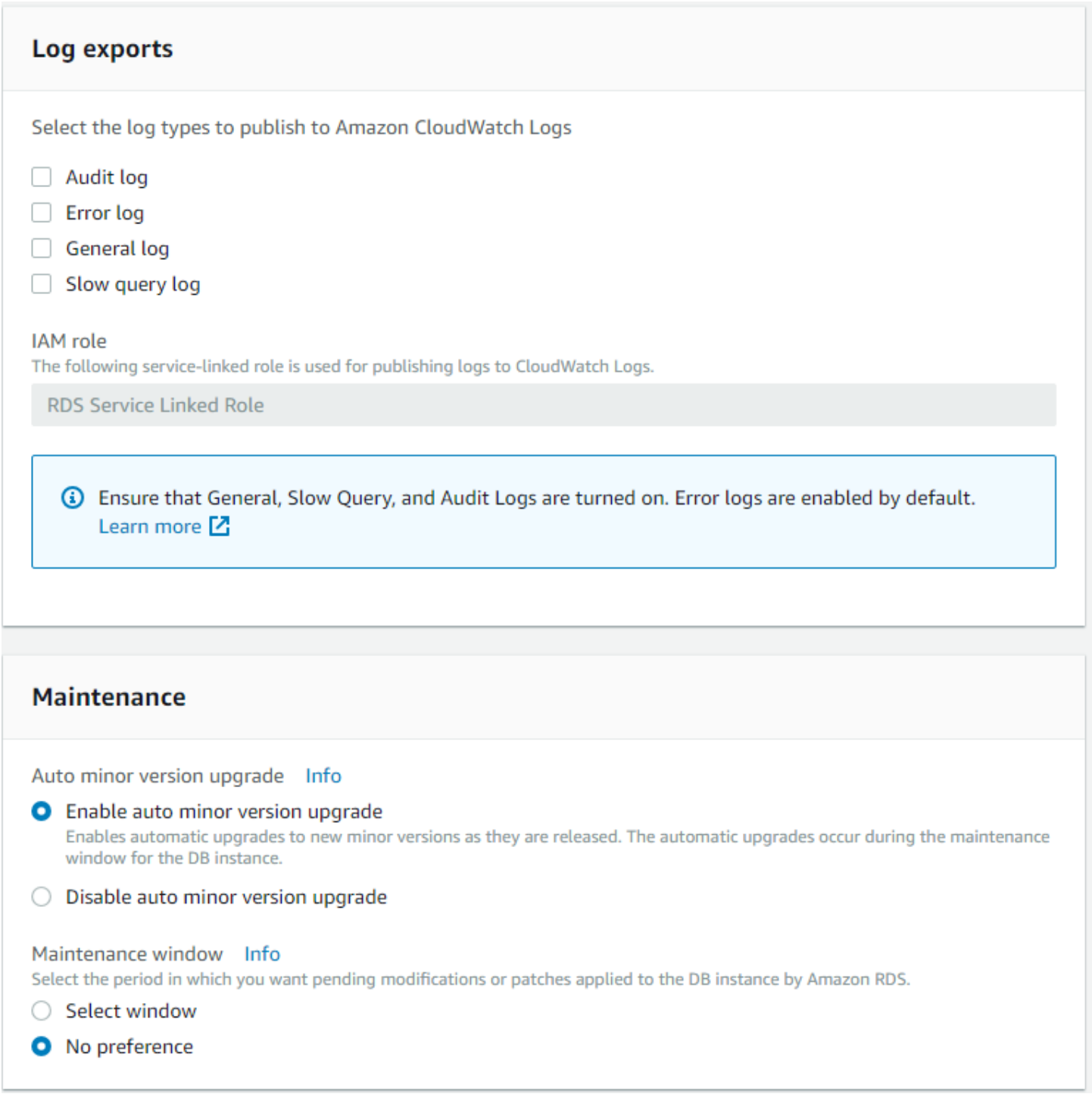


If you want database backups select, the retention max is **35 Days**

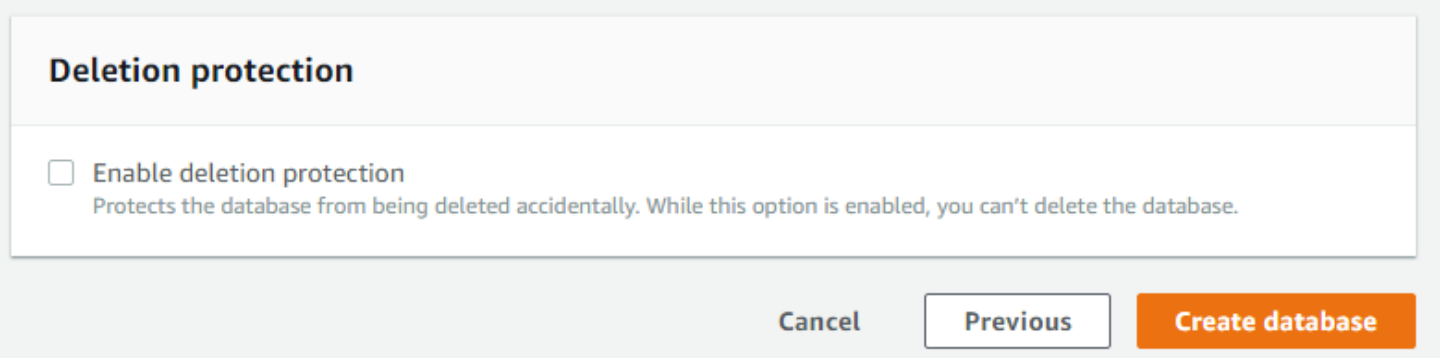
If you have particular backup window for database select it otherwise leave it default.



Enhanced monitoring will charged



Select the options you required



Enabling database protection, you cannot delete database

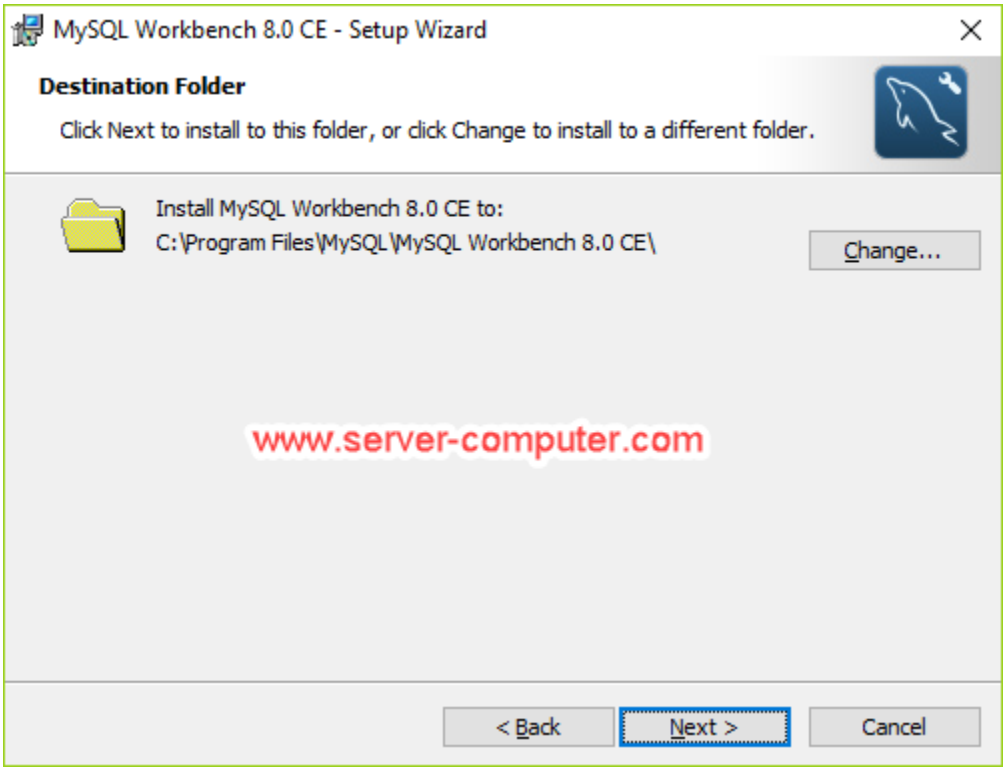
Click **Create Database**

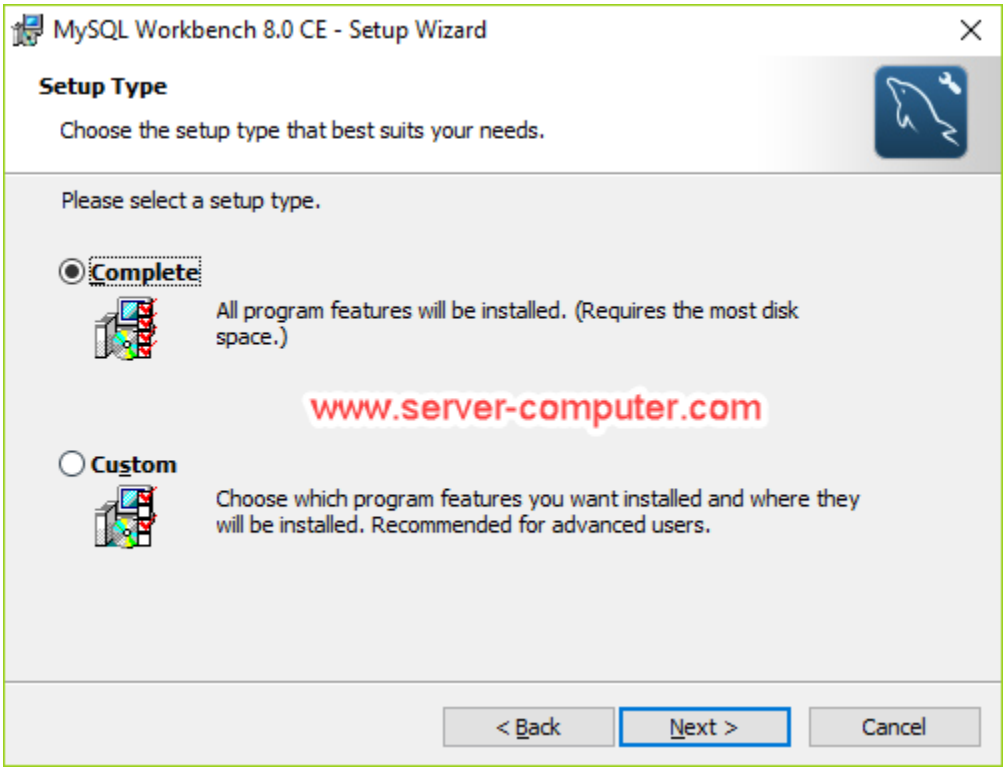
Note: Database instance creation will take at least 10minutes.

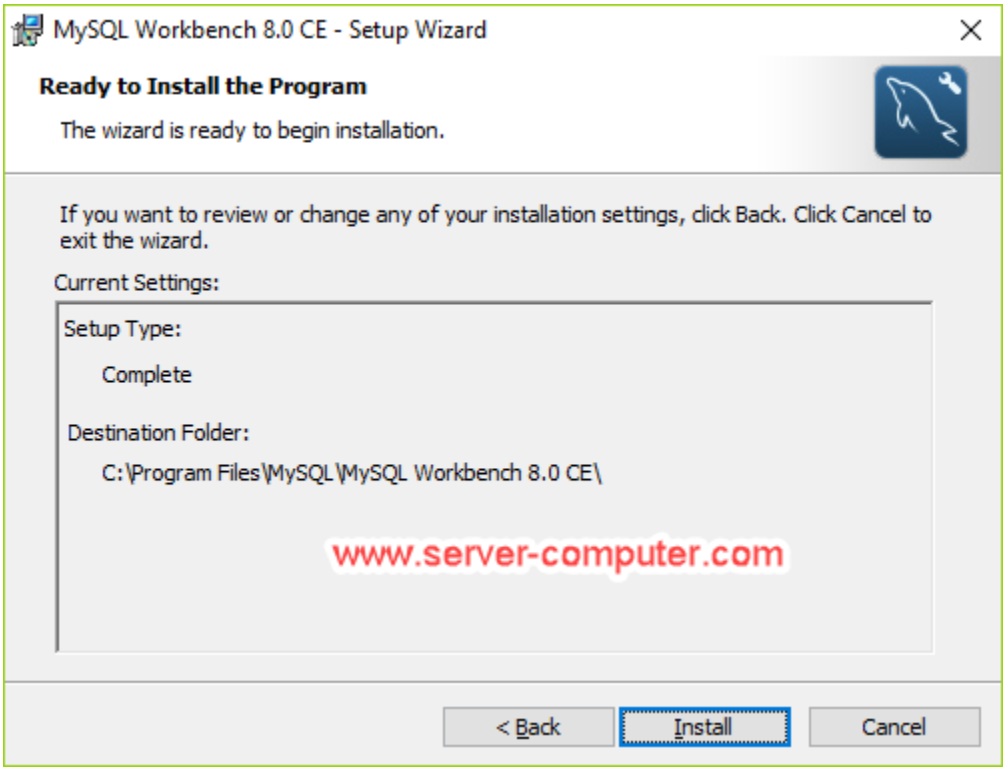
10.Accessing MySQL Instance Using Workbench

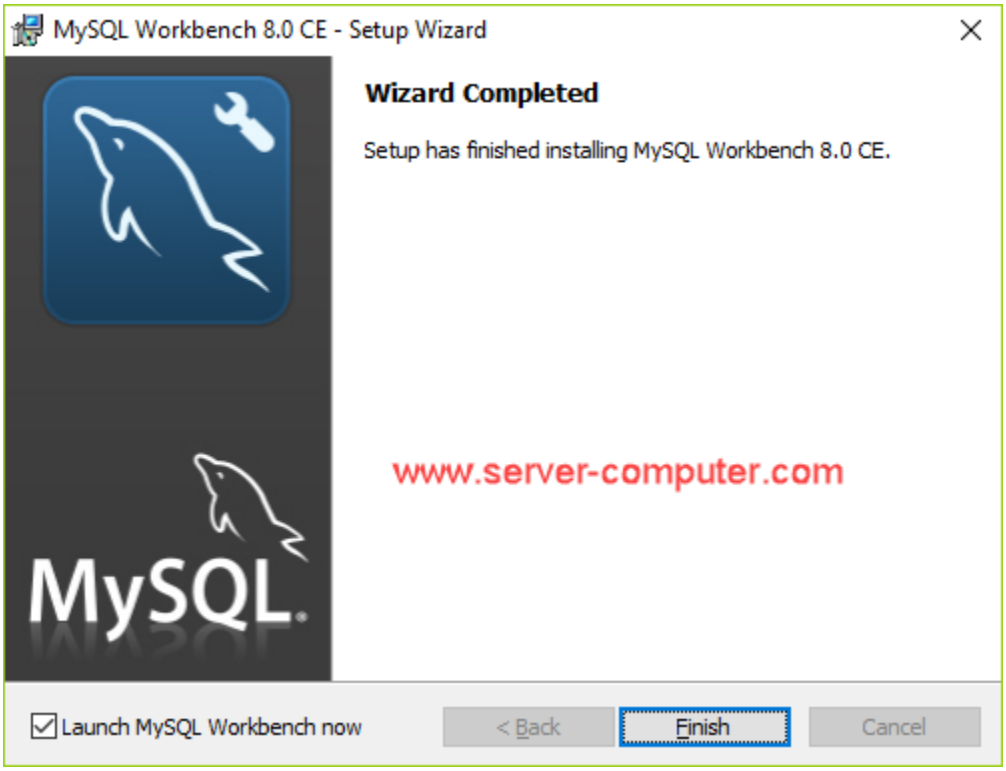
Download MySQL Workbench to access MySQL instance remotely

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

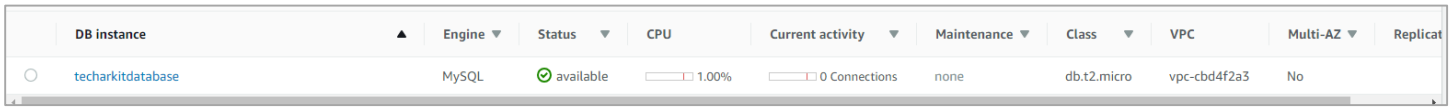








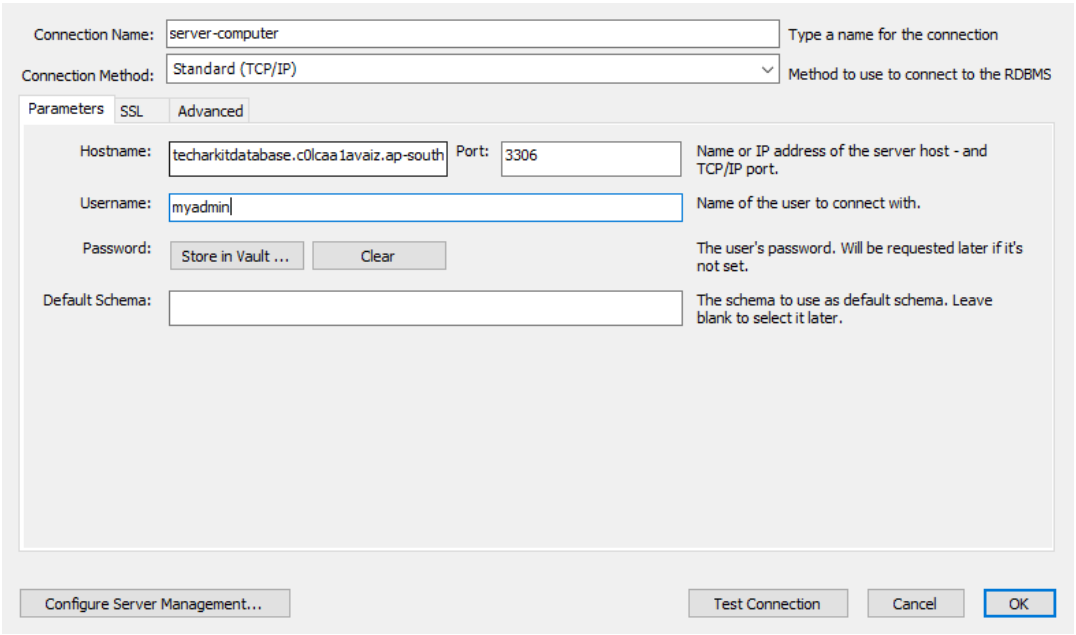
After successful creation you see like below



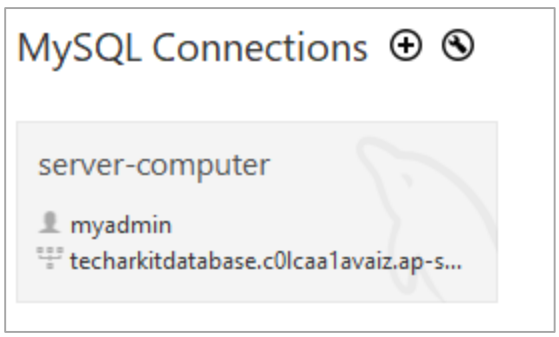
Click on Database name and come down copy the **Endpoint URL**

Open your MySQL workbench and create connection

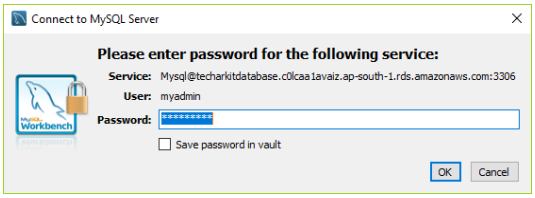
Click on Plus (+) sign to create a New MySQL Connection

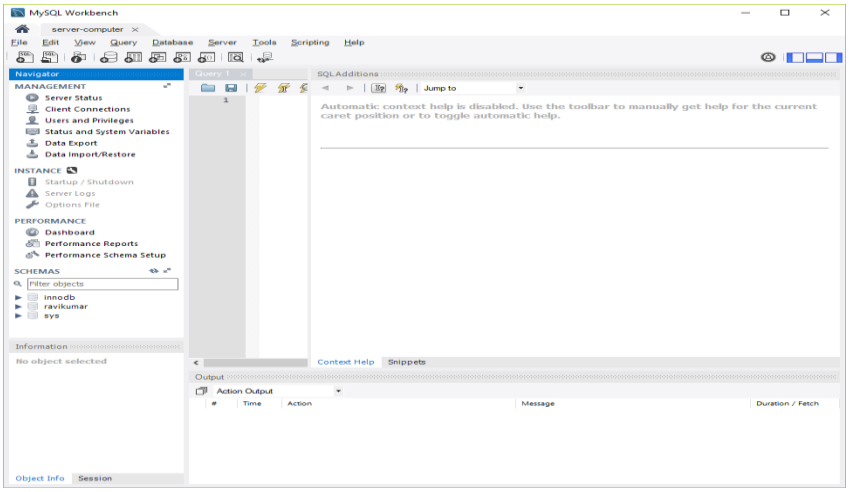


Click **OK**



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;
```

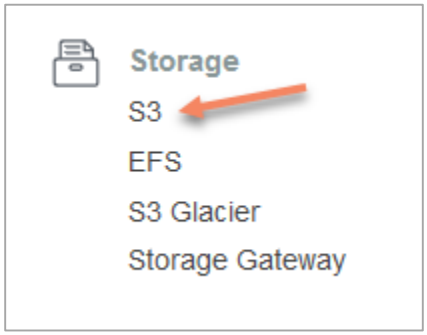
Create Table using below query

```
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;
```

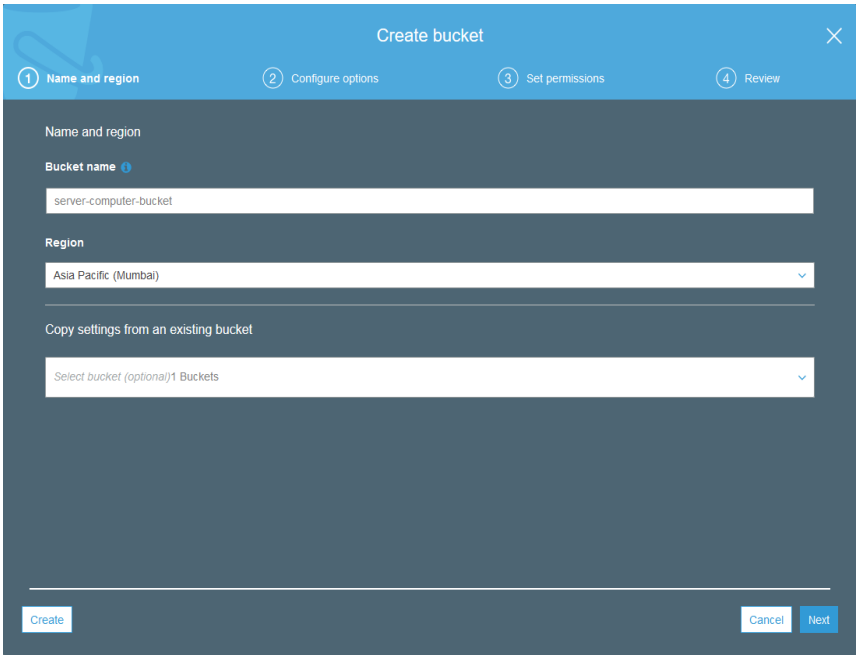
If you know much more database queries like select, insert and delete statement try doing more. Good Luck.

11. AWS S3 Bucket

Login to AWS Console and navigate to storage → S3

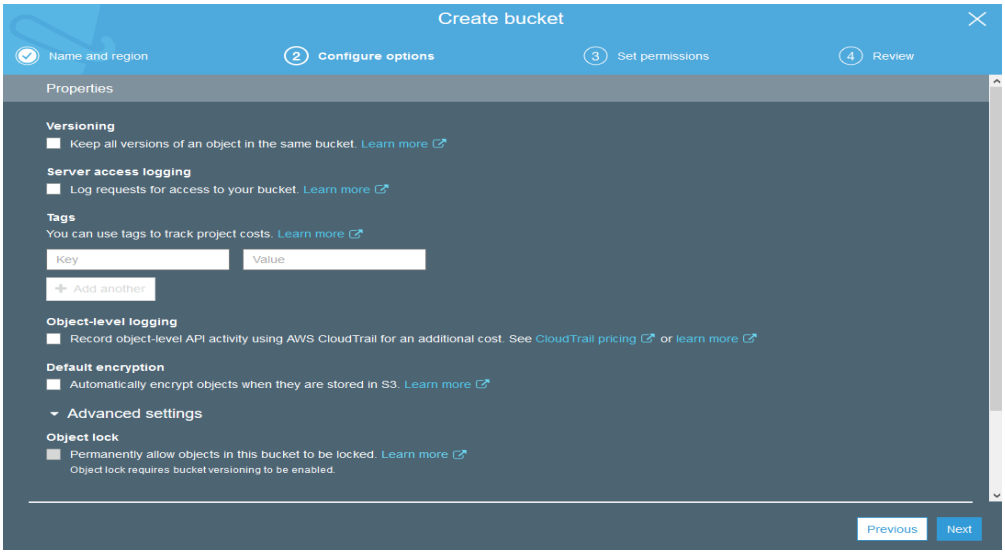


Click on 



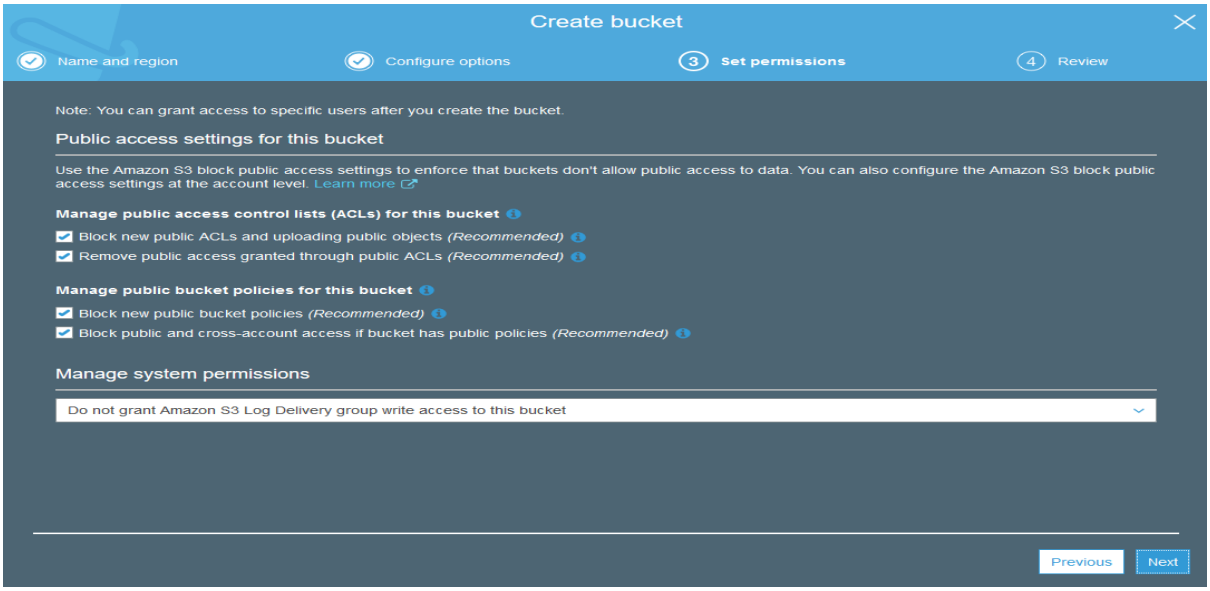
Provide bucket name, it should be a unique name. To Access your S3 bucket over internet it will create DNS entry.

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

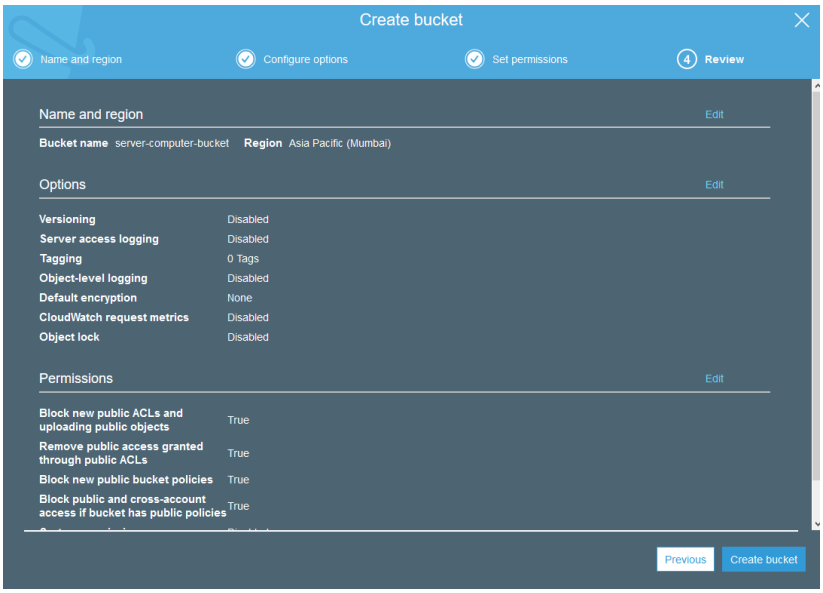
Click **Next**



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

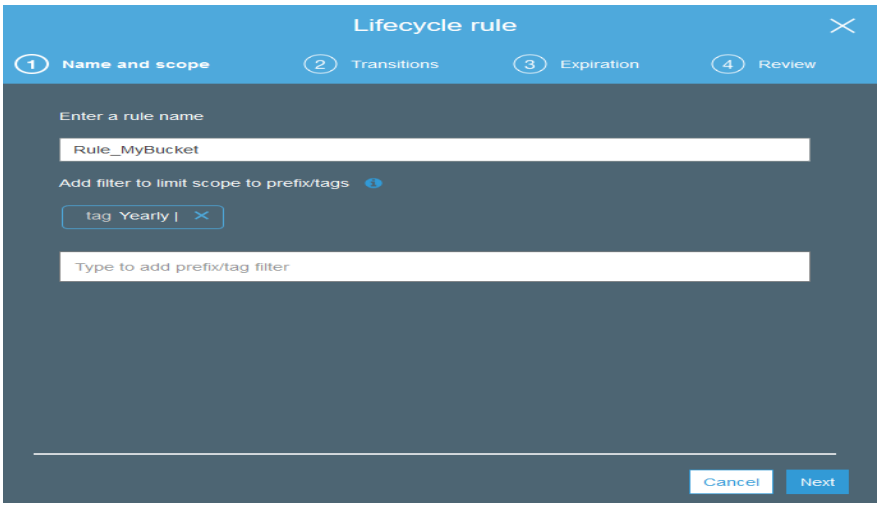
https://s3.ap-south-1.amazonaws.com/server-computer-bucket

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

11.1. AWS S3 Lifecycle Management

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

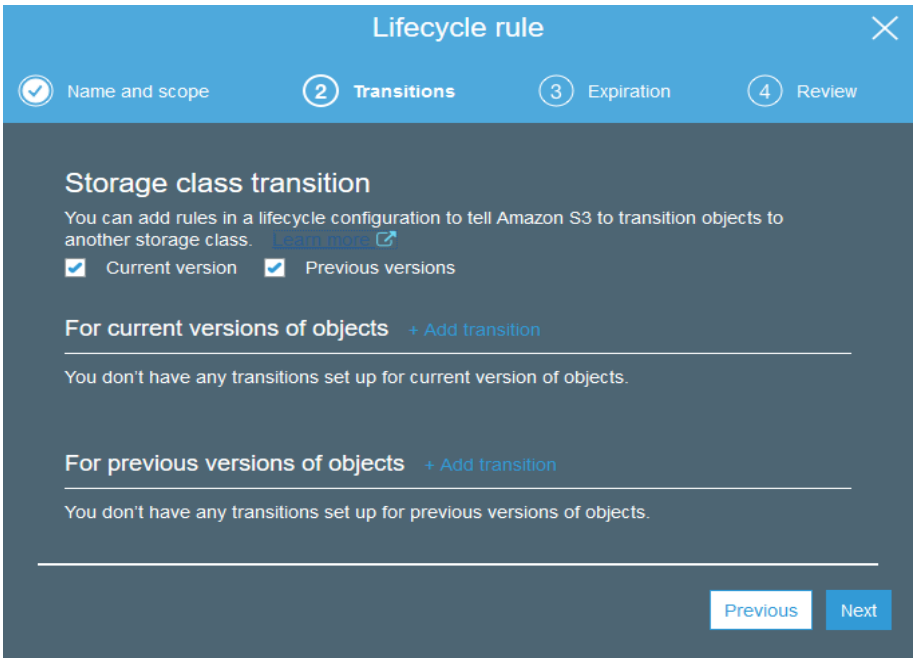
You can manage an objects lifecycle using this feature/rule, which defines



Enter Rule Name

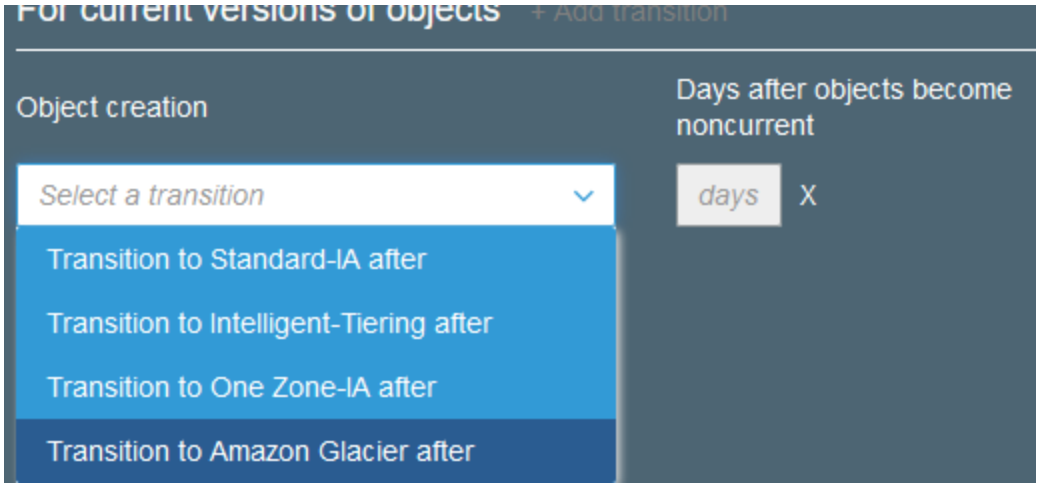
Tag Name if you do not want leave it blank

Click **Next**

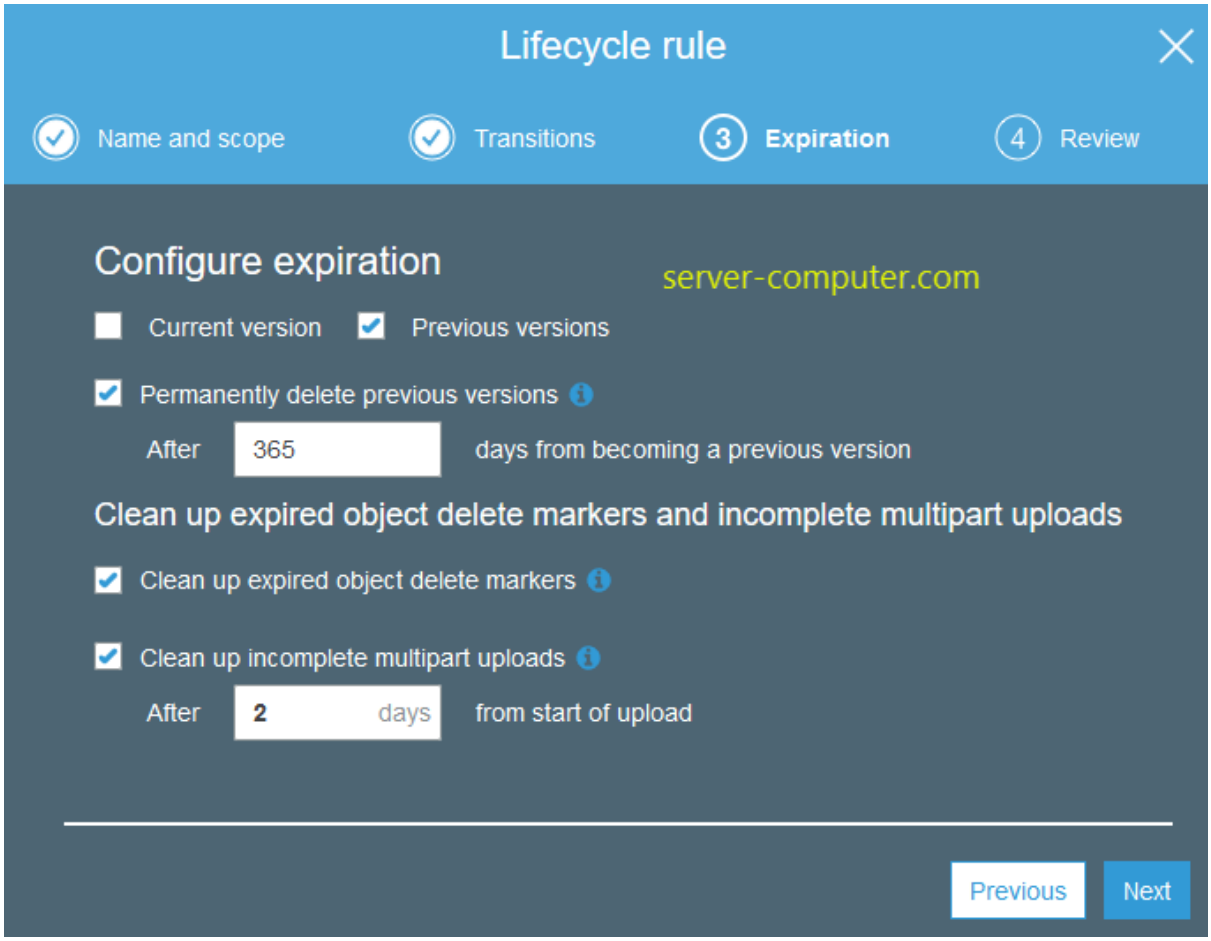


- Current Versions
- Previous Versions

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



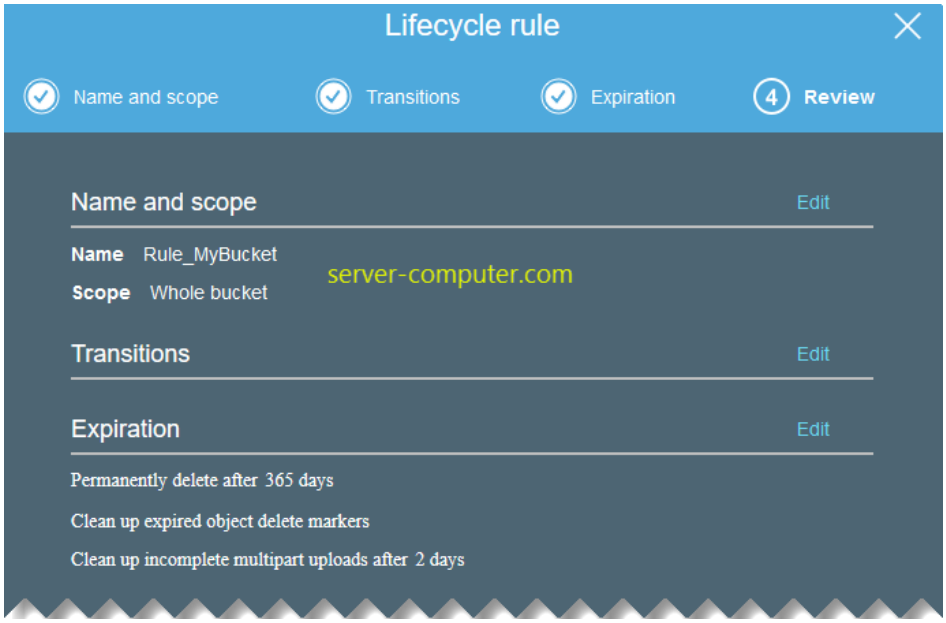
Click **Next**



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**

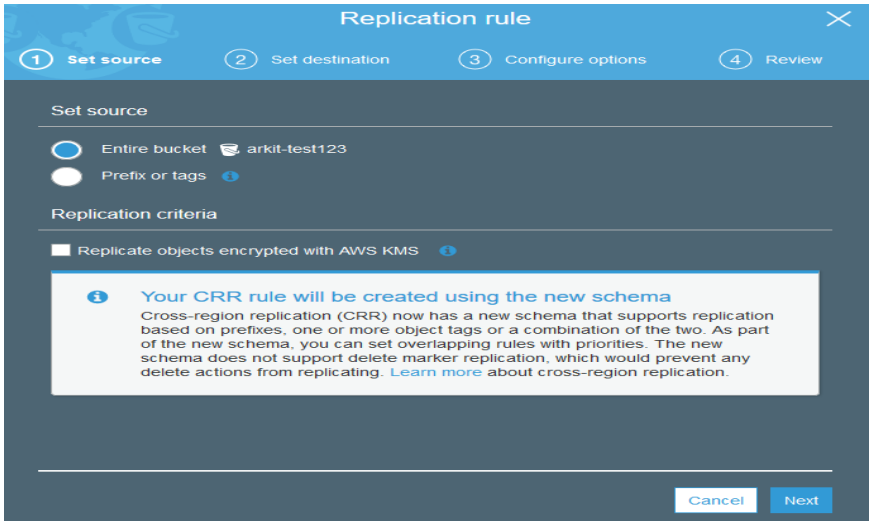


Click **Save.**

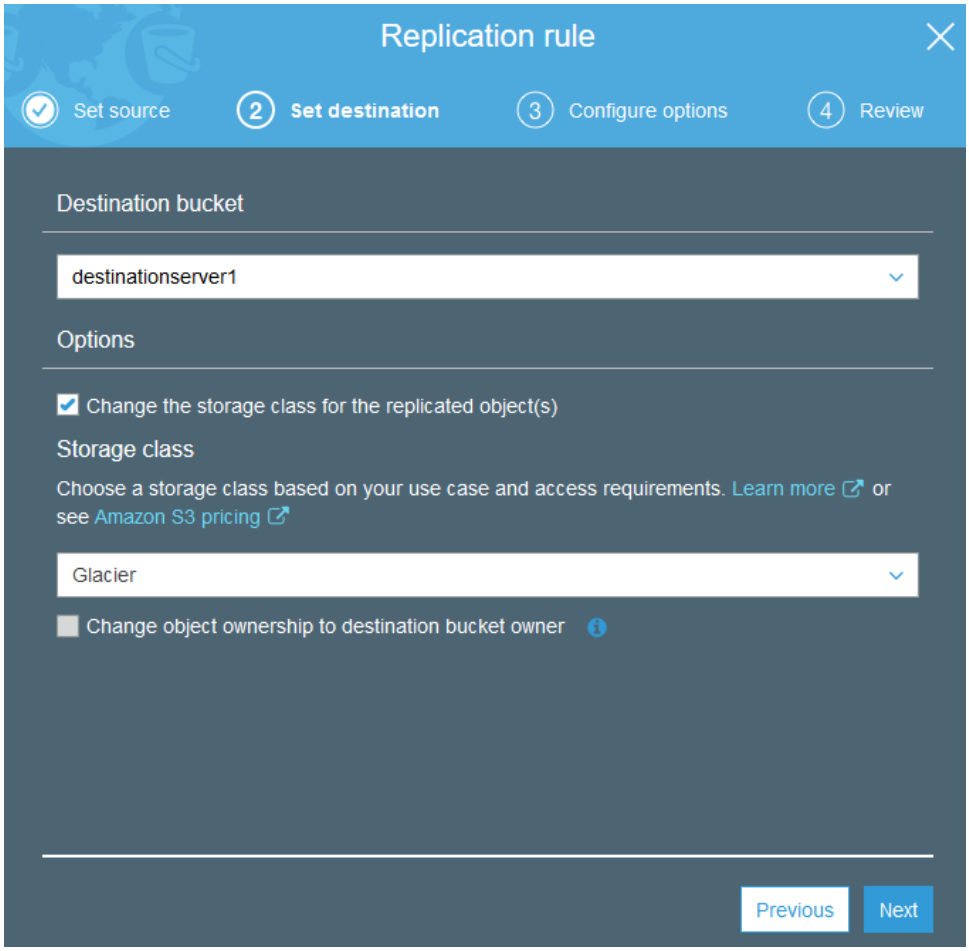
11.2. S3 Bucket Replication to Cross-Region

S3 bucket **Name → Management → Replication**

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



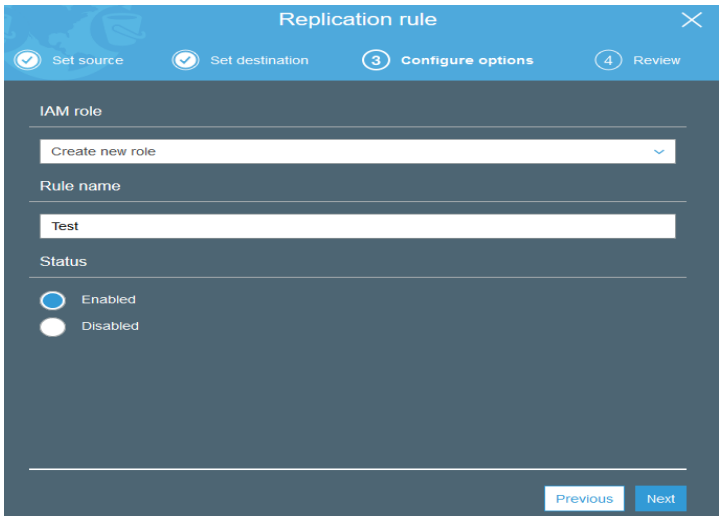
Click **Next**



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**

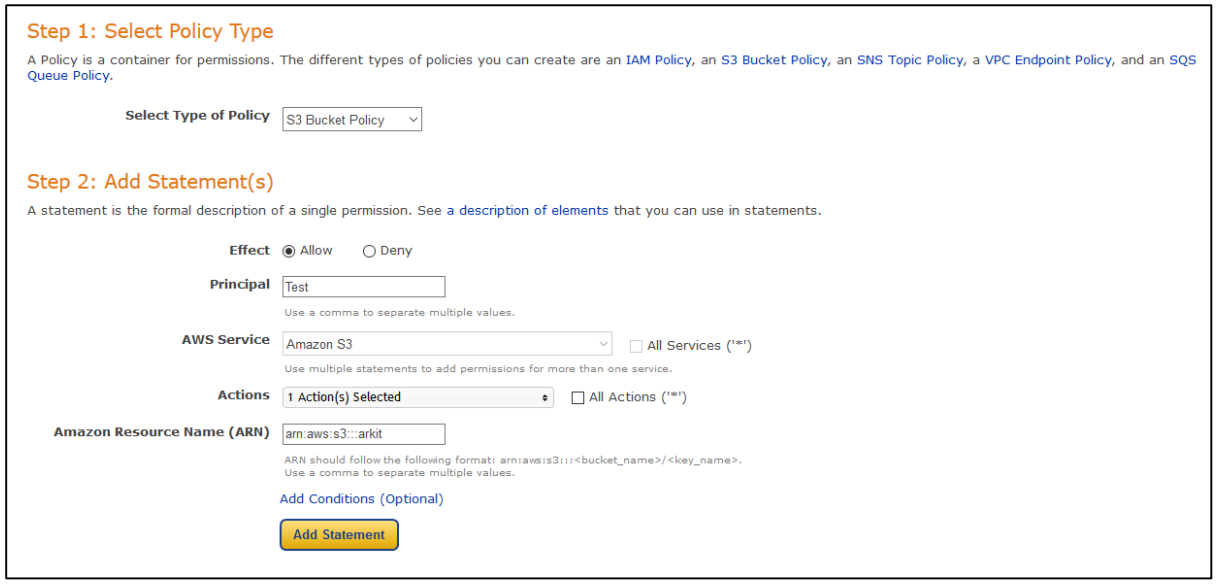
Review final and Click **Save**

11.3. 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



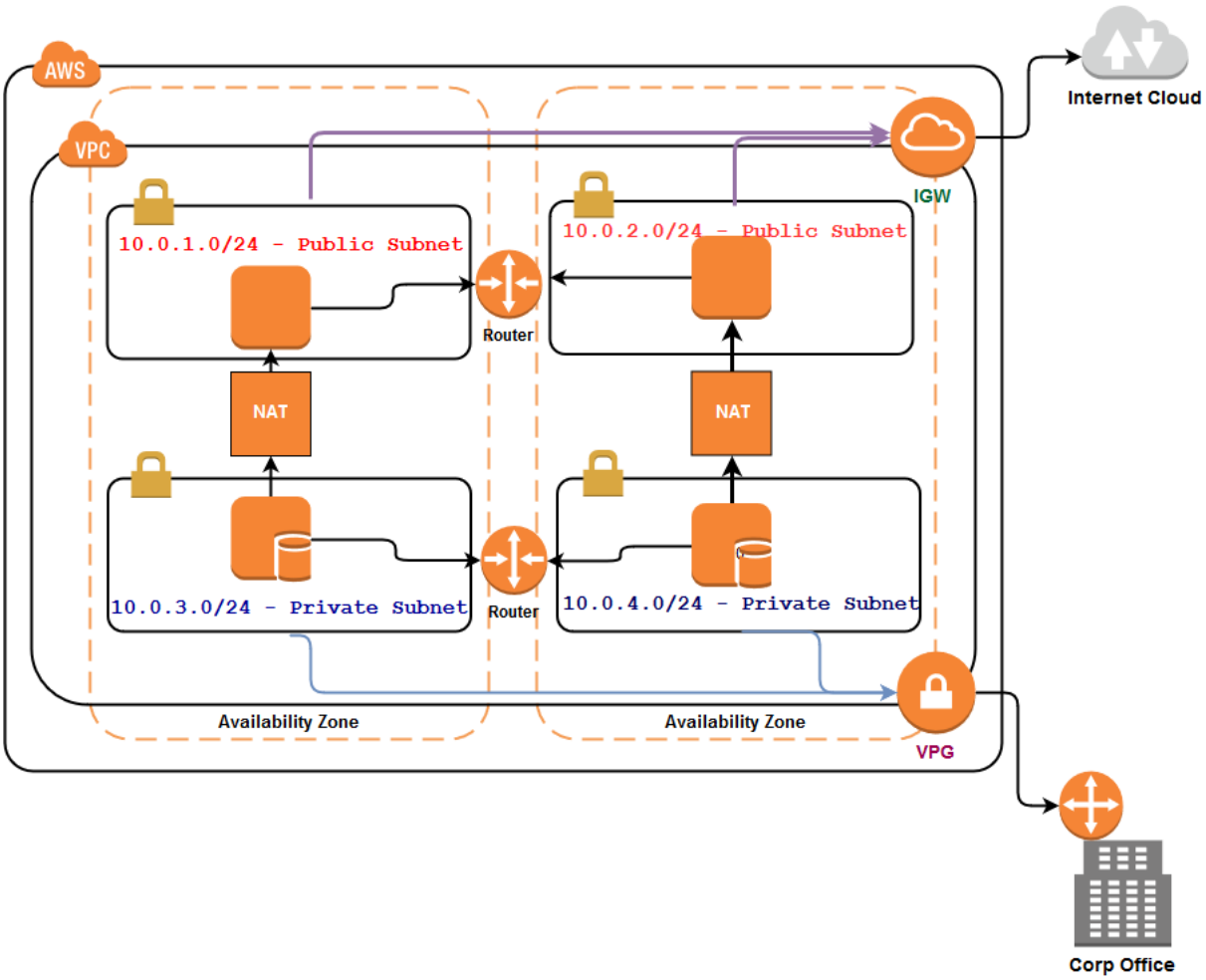
Add Statement and click on **Generate Policy**

```
{
  "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"
        ]
```

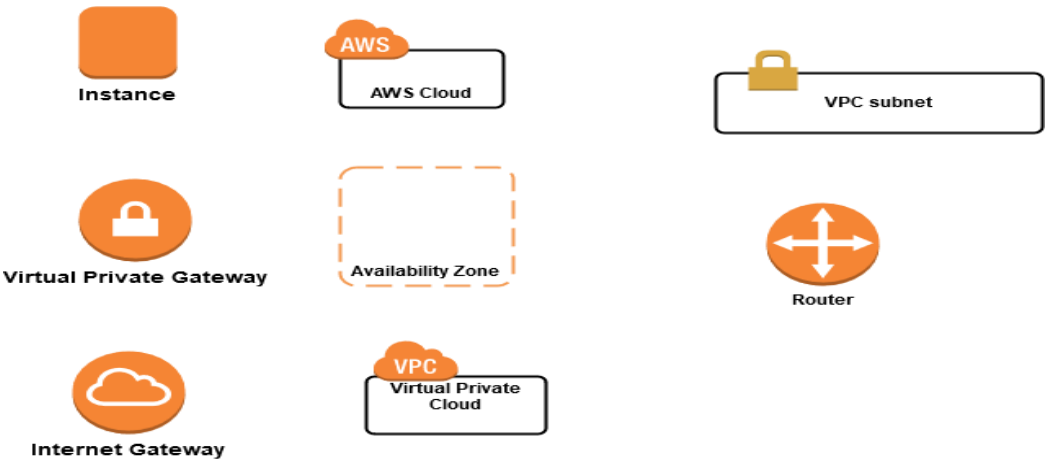
```
            }
        }
    ]
}
```

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

12. VPC – Virtual Private Cloud



Picture: 1.1 Typical VPC Example



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 2 web app servers and 2 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.