AMAZON-AWS WAR DEPLOYMENT

Version 1.0

User Manual

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Document Version** | **Updated By** | **Reviewed By** | **Release Date** |
| 1 | 1.0.1 | Vishnu Moorthy K | Raja | 21-May-16 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

***Amazon-AWS War Deployment***

**Description:**

Amazon Web Services (AWS), is a subsidiary of Amazon.com, which offers a suite of cloud computing services that make up an on-demand computing platform. These services operate from 12 geographical regions across the world. The most central and best-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3". AWS now has more than 70 services that span a wide range including compute, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools and tools for the Internet of things. Amazon markets AWS as a service to provide large computing capacity quicker and cheaper than a client company building an actual physical server farm.

**Procedures:**

* Account Login/Registration
* EC2(Elastic Compute Cloud)
* Elastic IP
* Security Groups
* Configure PUTTY with SSH
* Install Required Software
* Route 53
* Record Sets
* SES (Simple Email Service)
* S3 (Simple Storage Solution)
* War Deployment
* Jar Deployment
* Putty Commands
* Issue Analysis

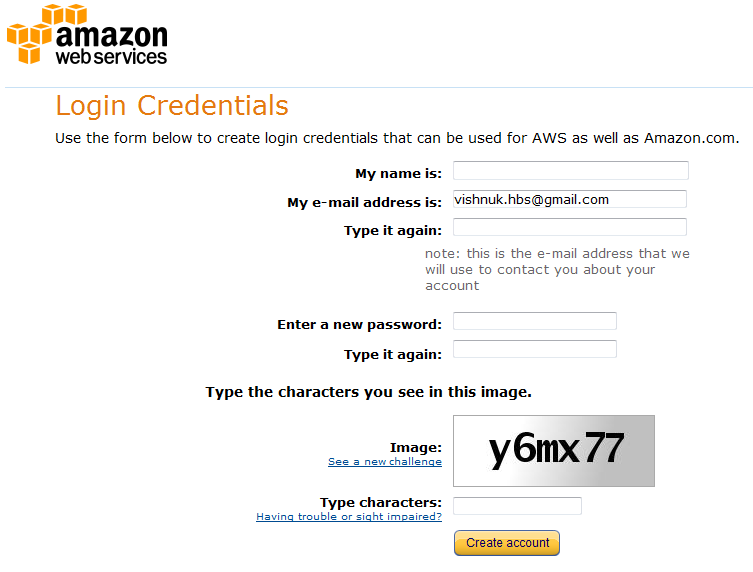
**Account Login/Registration;**

* Login/Register an Account with AMAZON-AWS from this URL <https://console.aws.amazon.com>

**Amazon Login/Registration Image:**

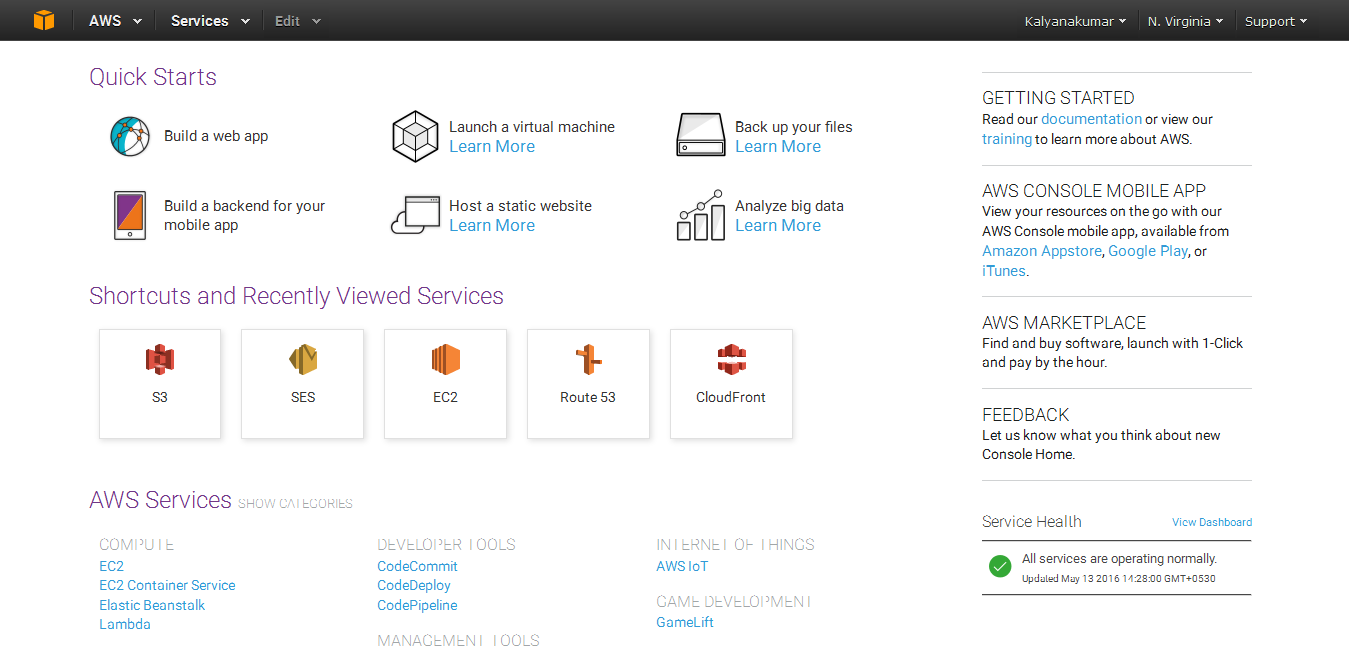


**Amazon Registration Image:**



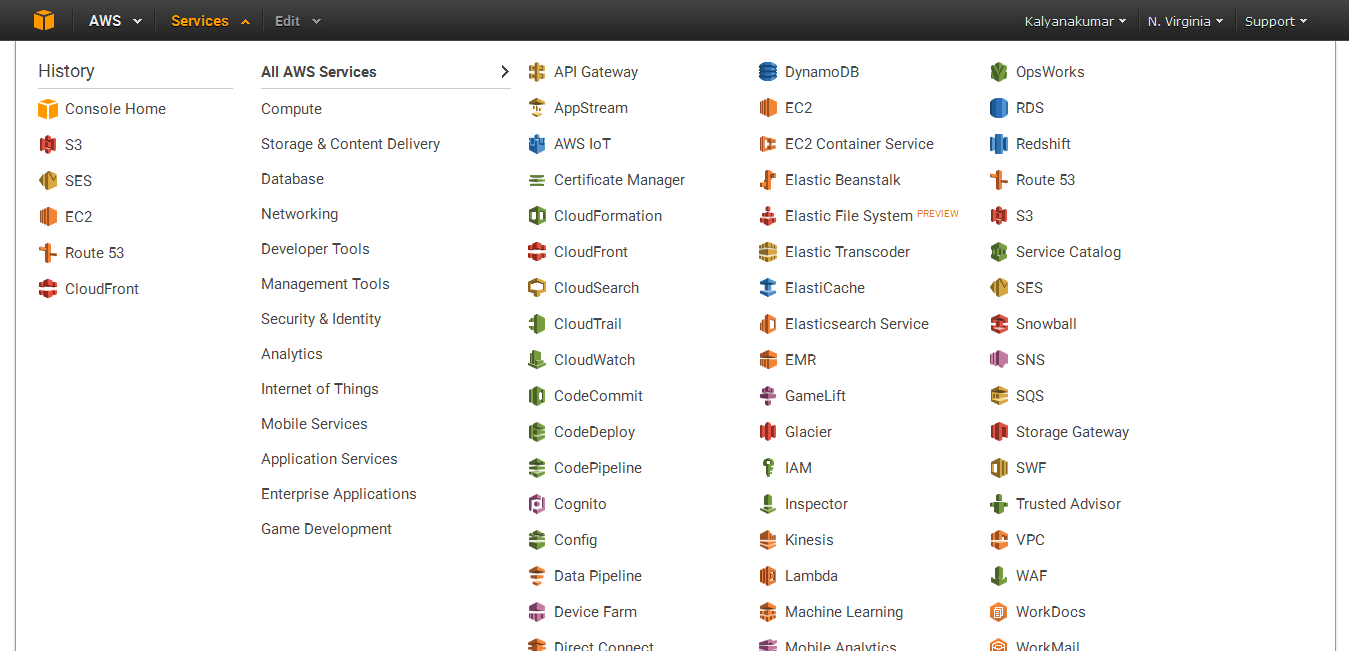
* After create Account provide the Credit Card information to complete the Registration.
* Login to AWS.

**Amazon Dashboard Image:**



* Services Menu

**Amazon Service Menu Image:**

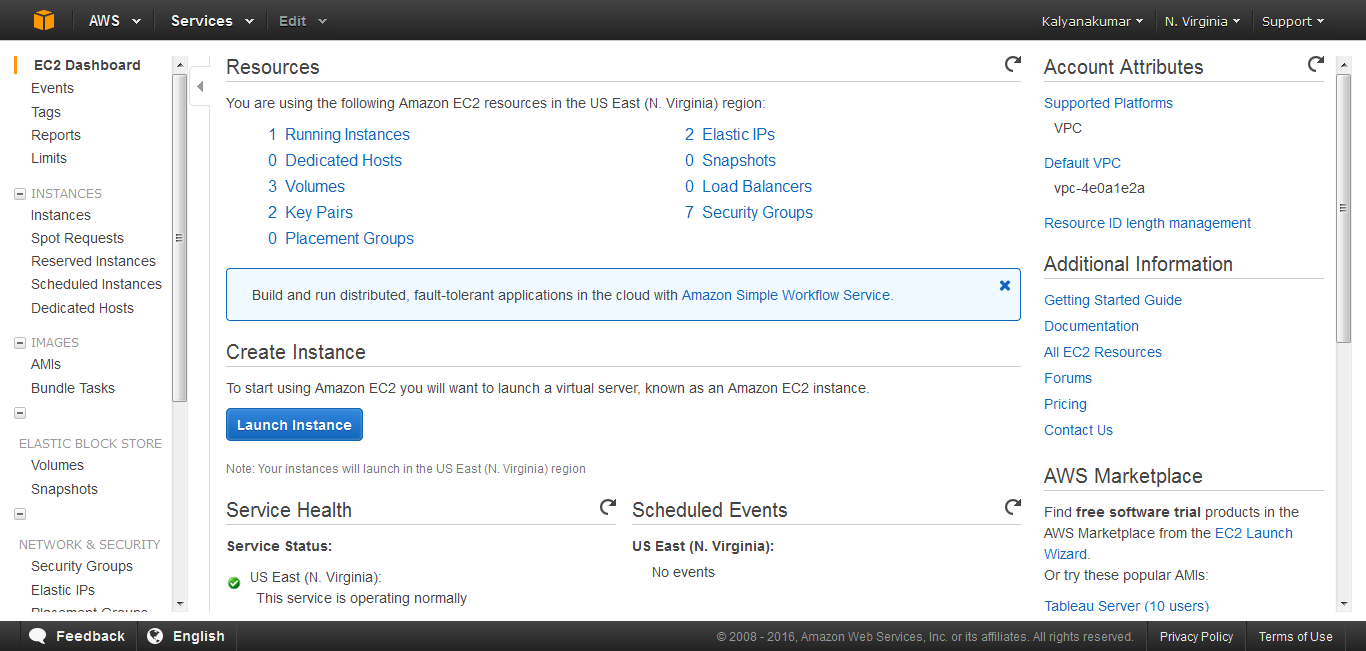


**EC2:**

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale [cloud computing](https://aws.amazon.com/what-is-cloud-computing/) easier for developers.

Amazon EC2’s simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon’s proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

**EC2 Instance Dashboard Image:**

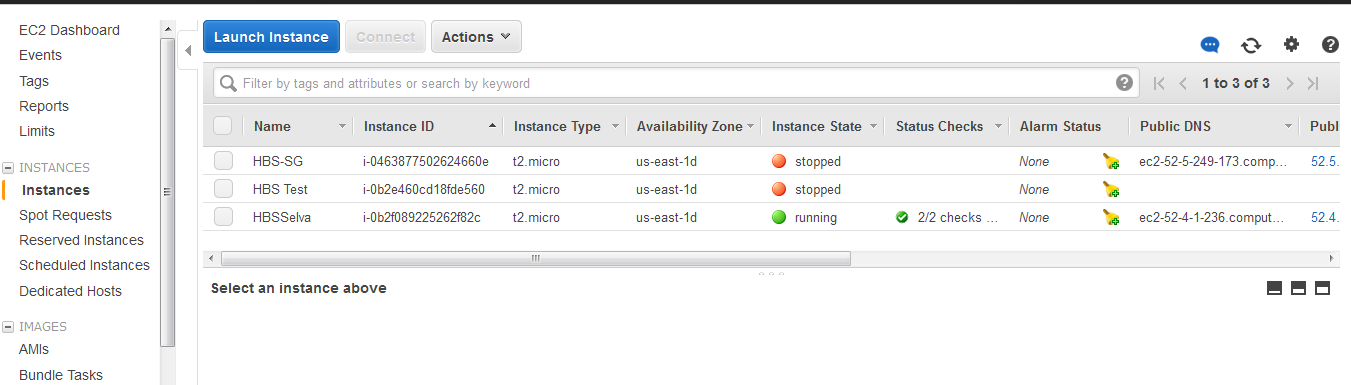


**Instances:**

**Create New Instance**

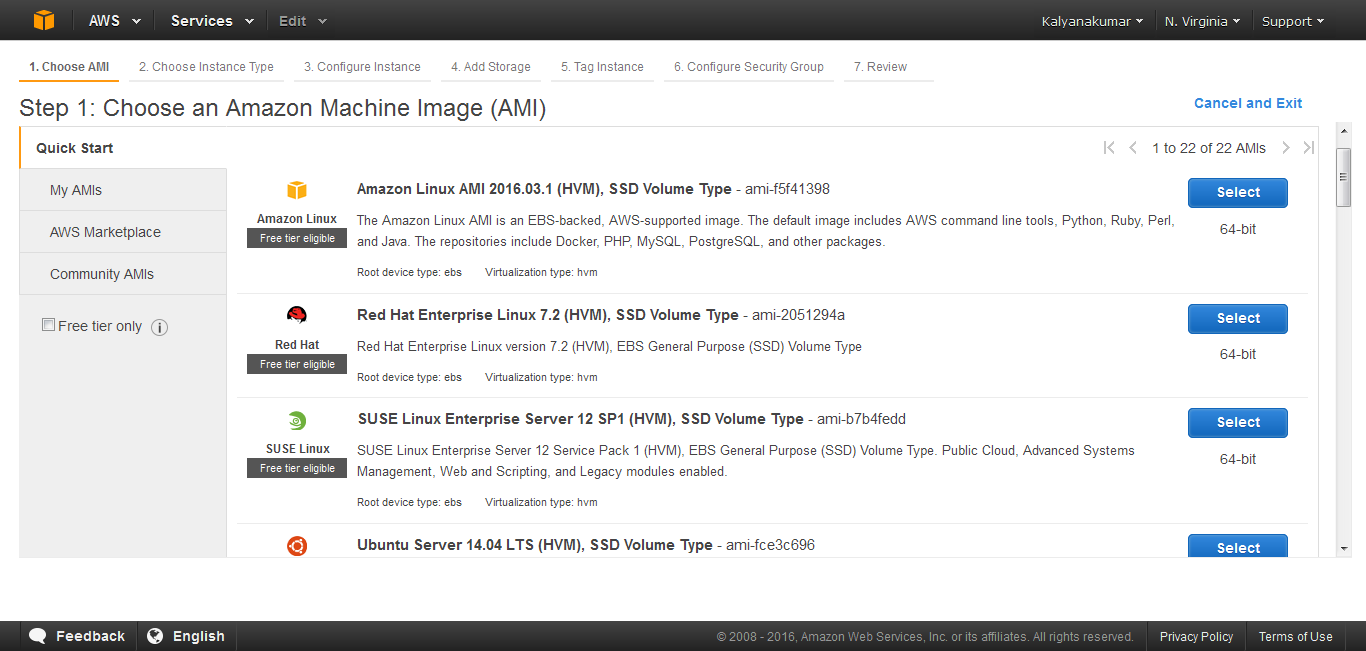
* Click “Launch Instance” button to create the new instance.

**Instances Table Image:**

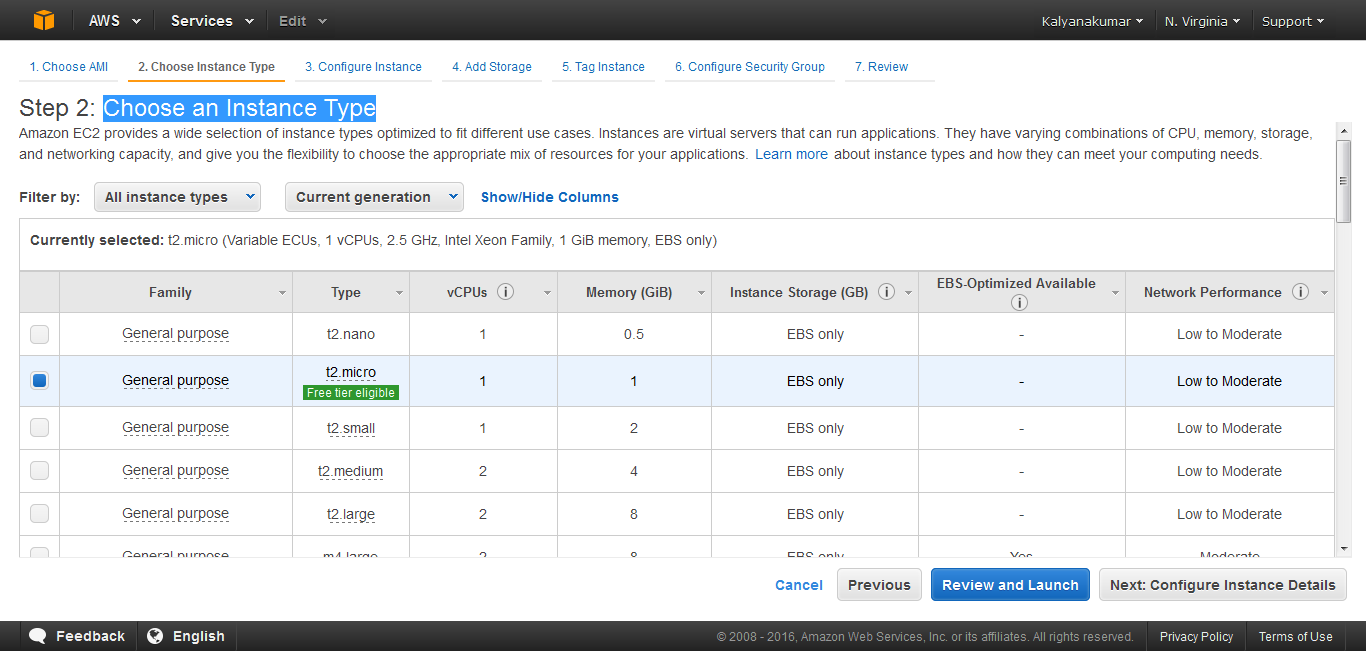


* Choose an Amazon Machine Image (AMI) –(Red Hat)

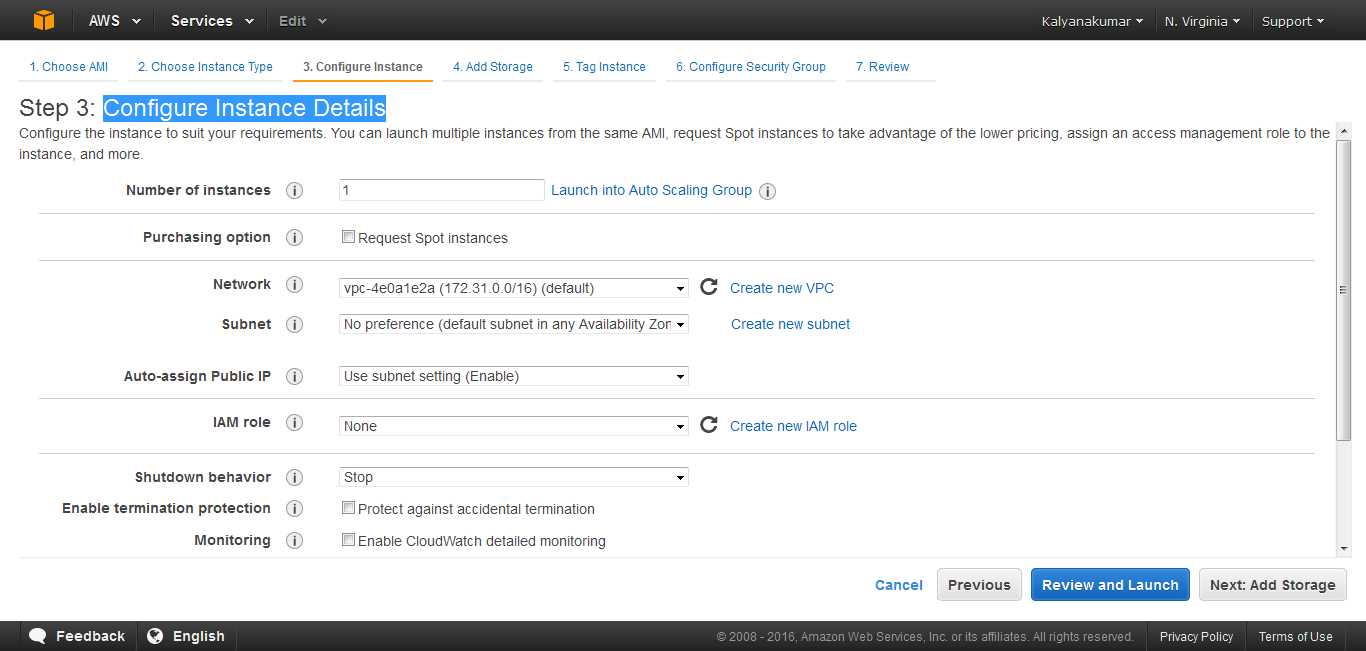
**Create EC2 Instance Images:**



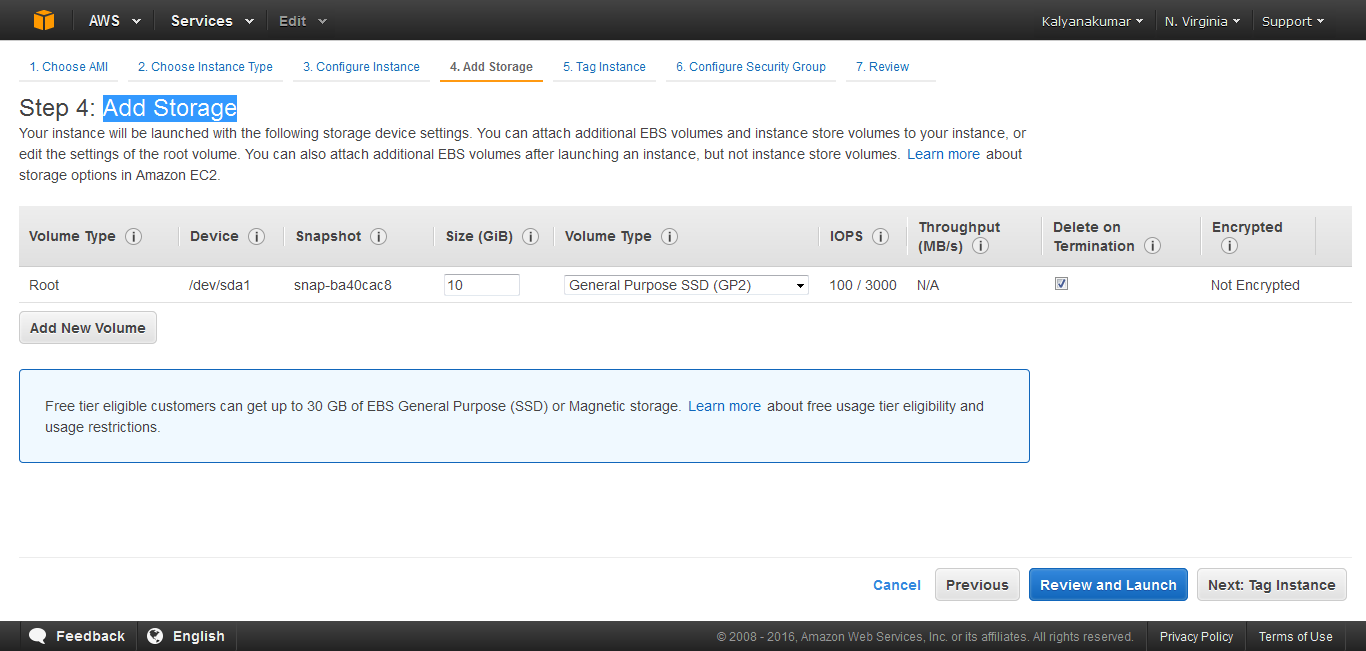
* Choose an Instance Type



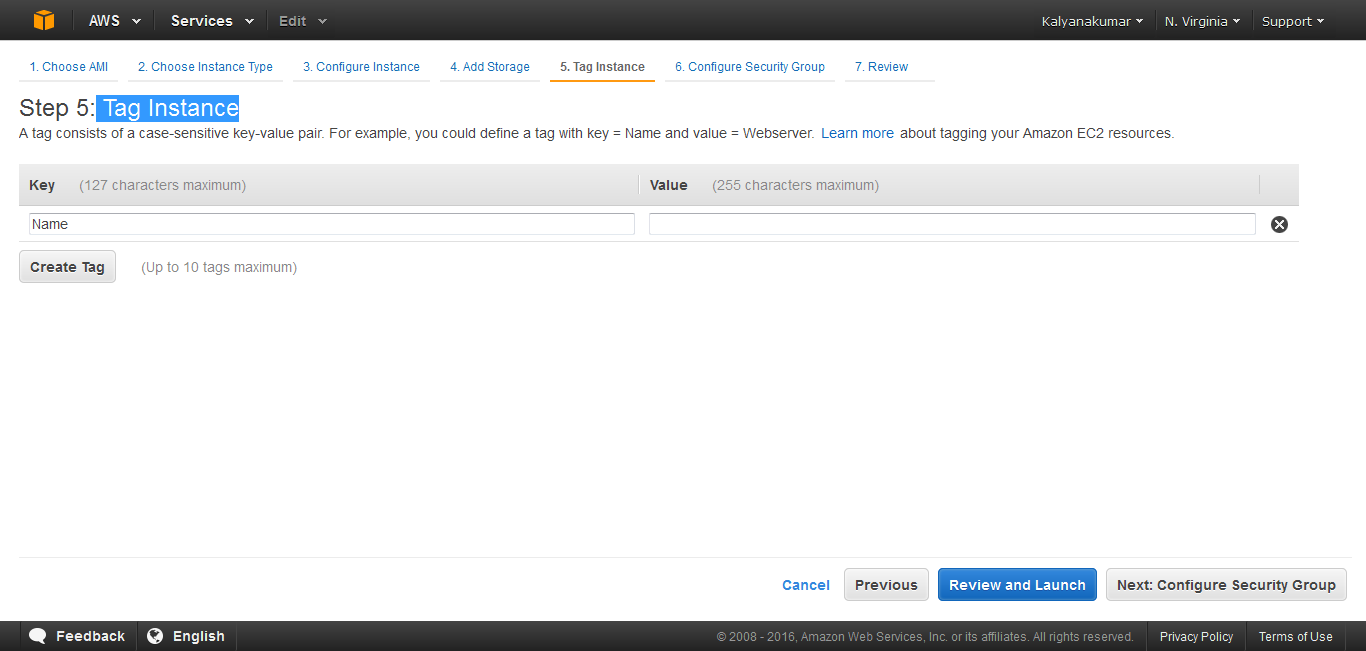
* Configure Instance Details



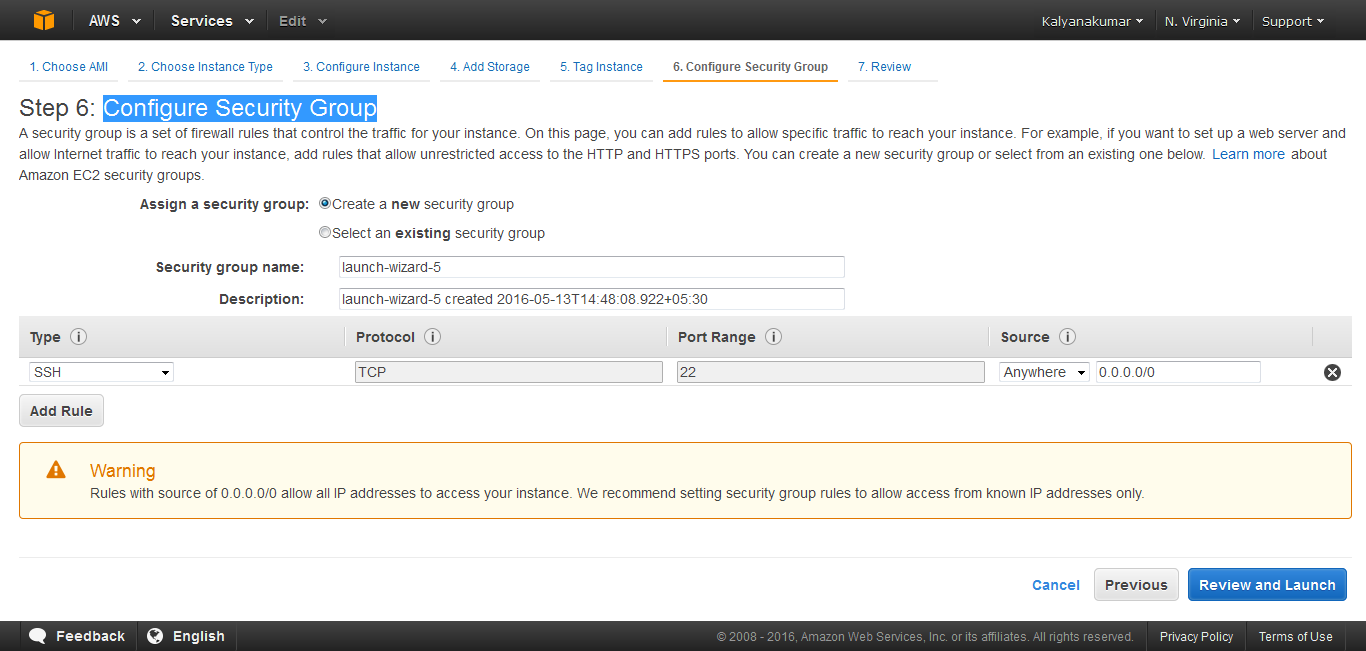
* Add Storage



* Tag Instance



* Configure Security Group

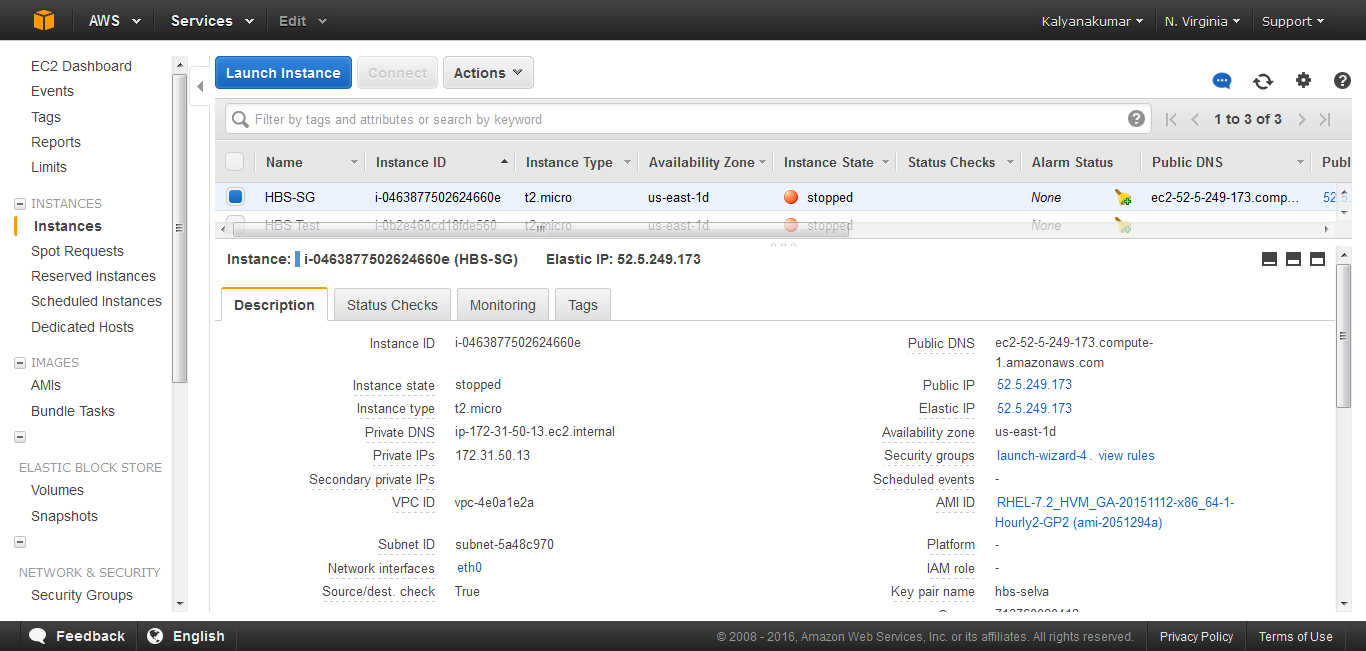


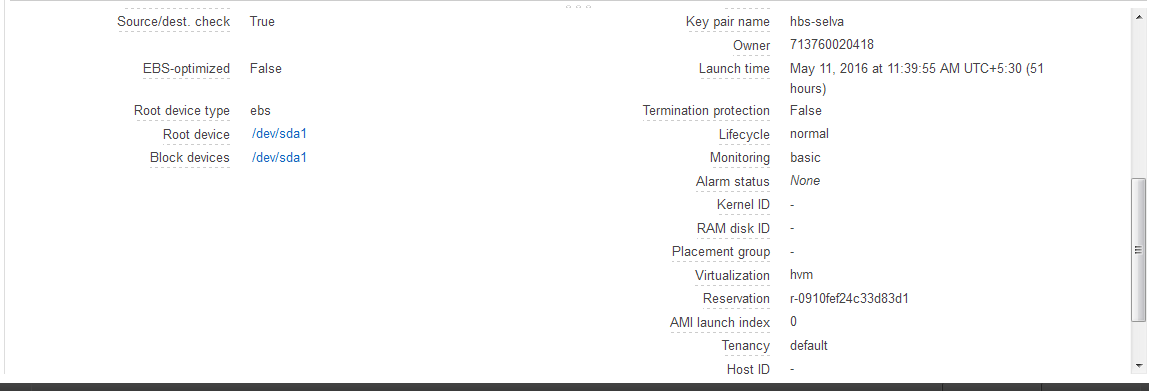
* Review & Launch
* Download the key file (filename.pem) while launch the instance, which is used for connecting putty with corresponding EC2 instance.

**View the Selected Instance:**

* Instance Description

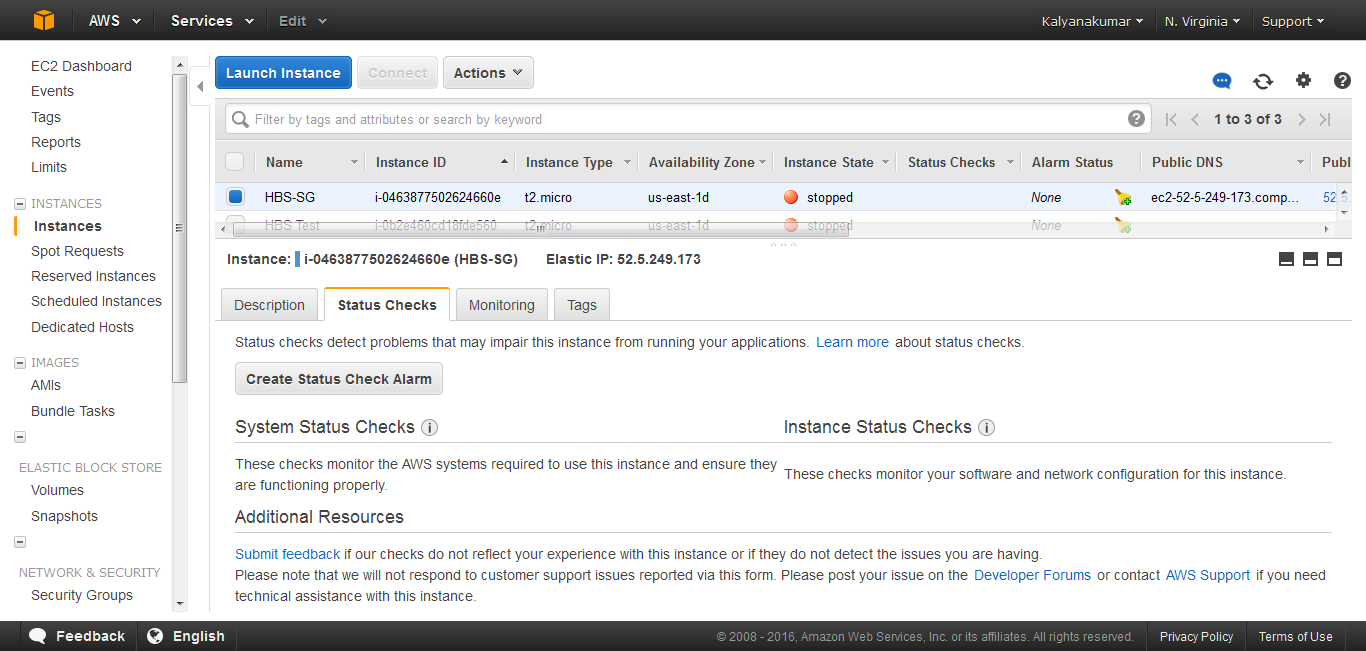
**Instance Description Image:**





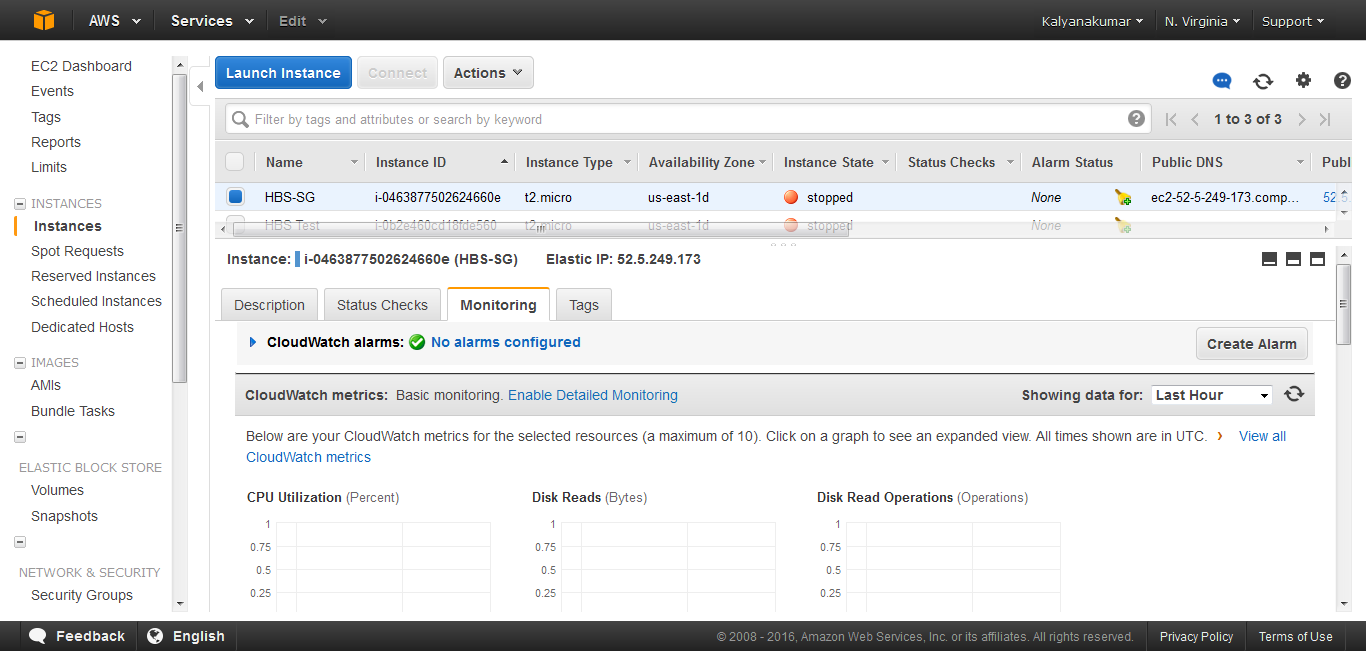
* Check Status

**Instance Check Status Image:**



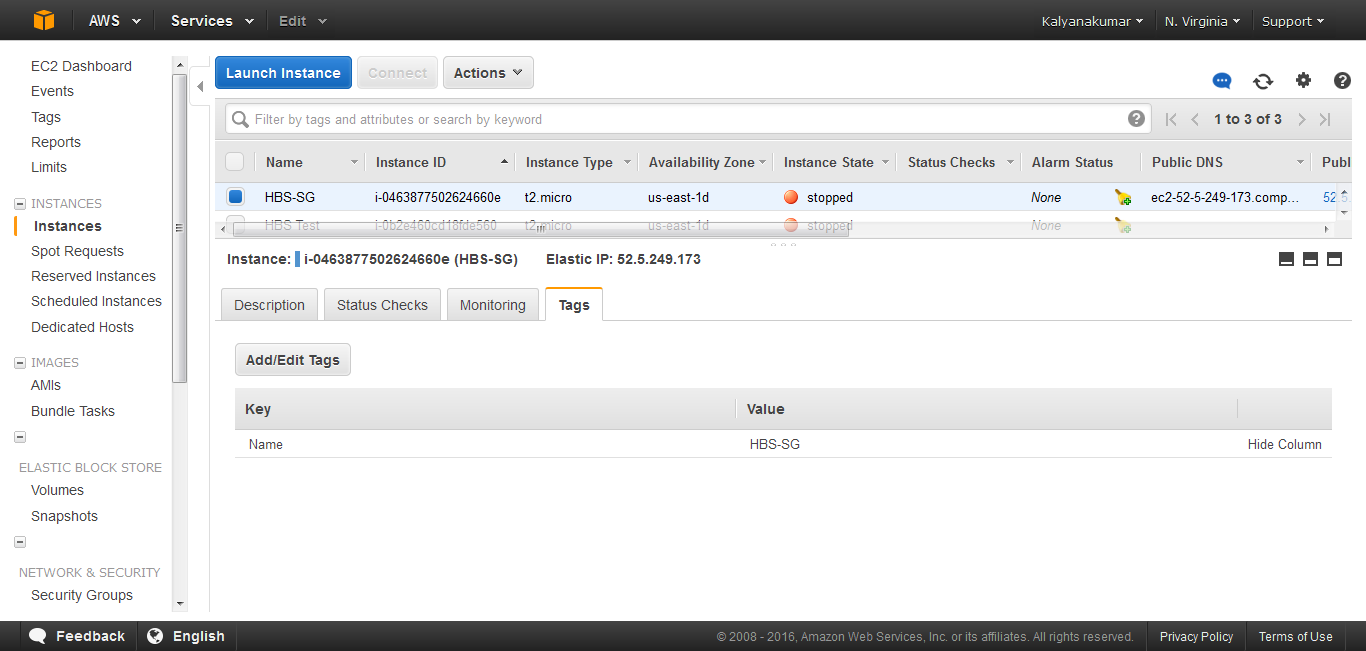
* Monitoring

**Instance Monitoring Image:**



* Tags

**Instance Tag Image:**



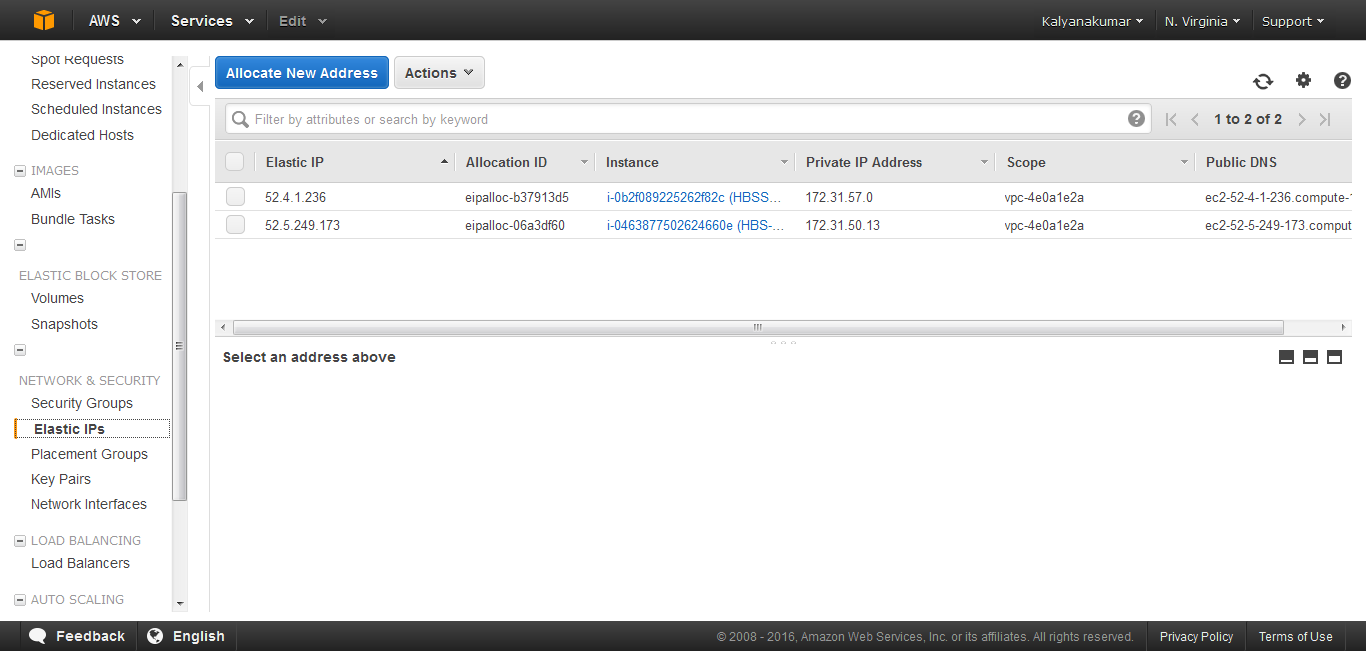
**Elastic IP:**

An Elastic IP address is a static IP address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

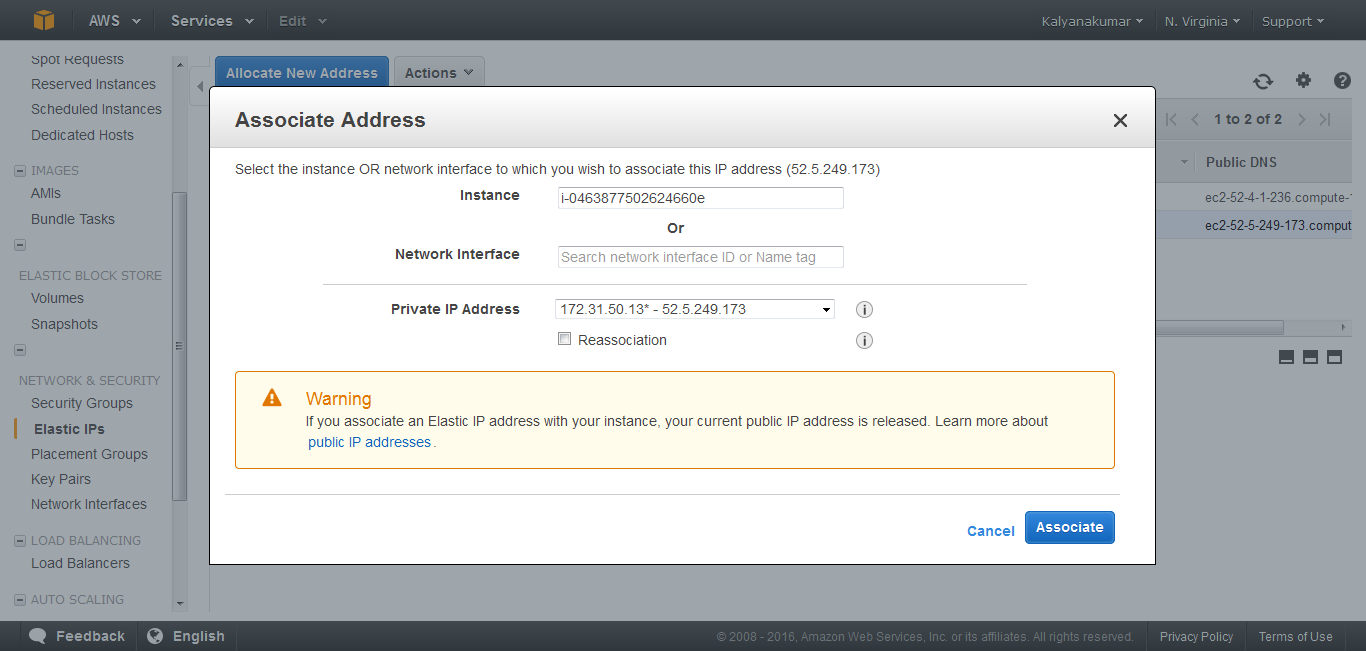
An Elastic IP address is a public IP address, which is reachable from the Internet. If your instance does not have a public IP address, you can associate an Elastic IP address with your instance to enable communication with the Internet; for example, to connect to your instance from your local computer.

* Elastic IP is under the Network & Security
* Click “Allocate New Address” to create the new Elastic IP Address
* Choose Elastic IP Address to Associate With Corresponding EC2 Instance Id

**Elastic IP Image:**



**Associate Elastic IP with Instance Image:**



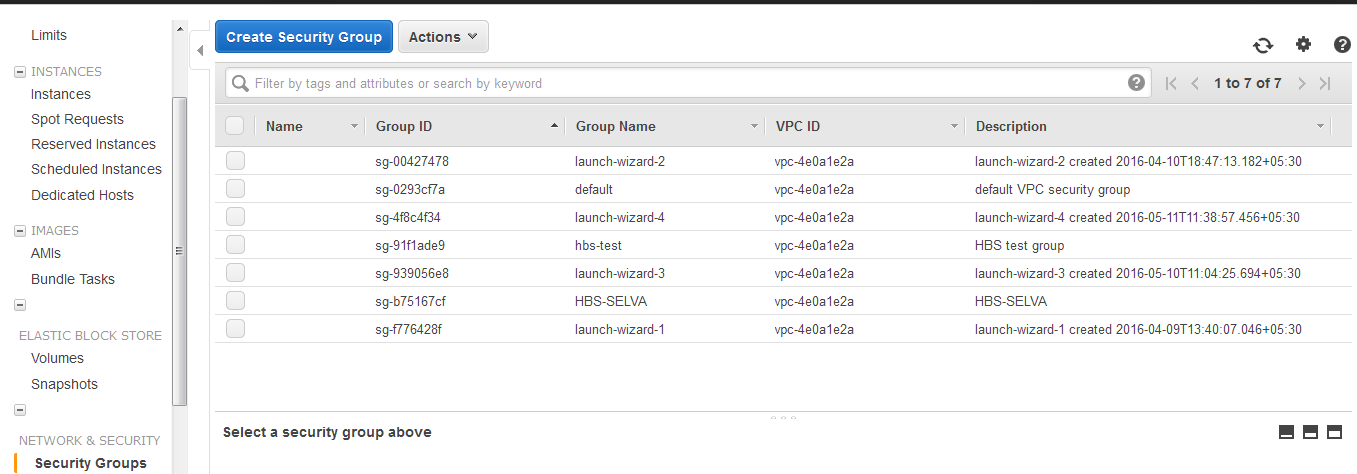
**Security Group:**

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

## Create Security Group:

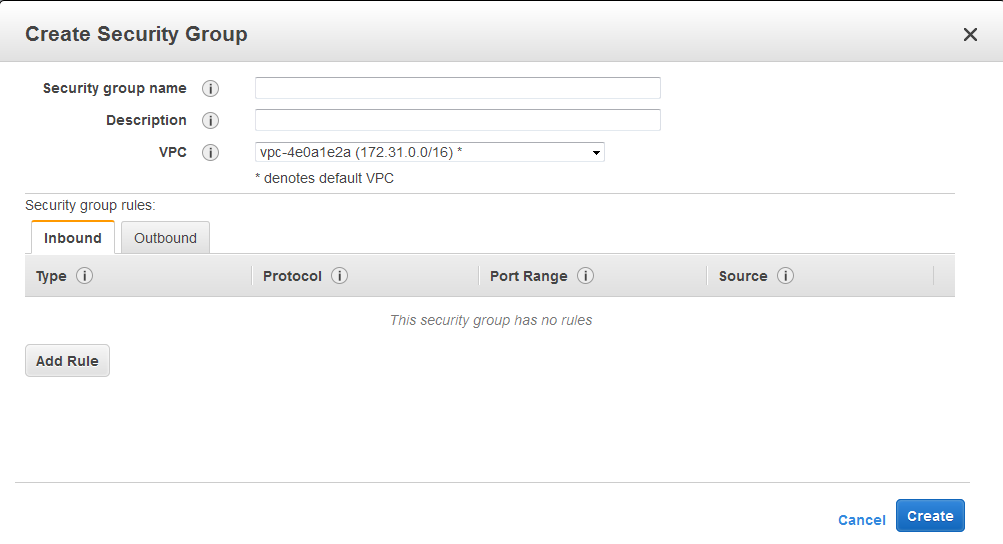
* Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
* In the navigation pane, choose Security Groups under Network & Security.
* Choose Create Security Group.

**EC2-Instance Security Group Image:**



* Specify a name and description for the security group. For VPC, choose No VPC to create a security group for EC2-Classic, or choose a VPC ID to create a security group for that VPC.

**Create Security Group Image:**

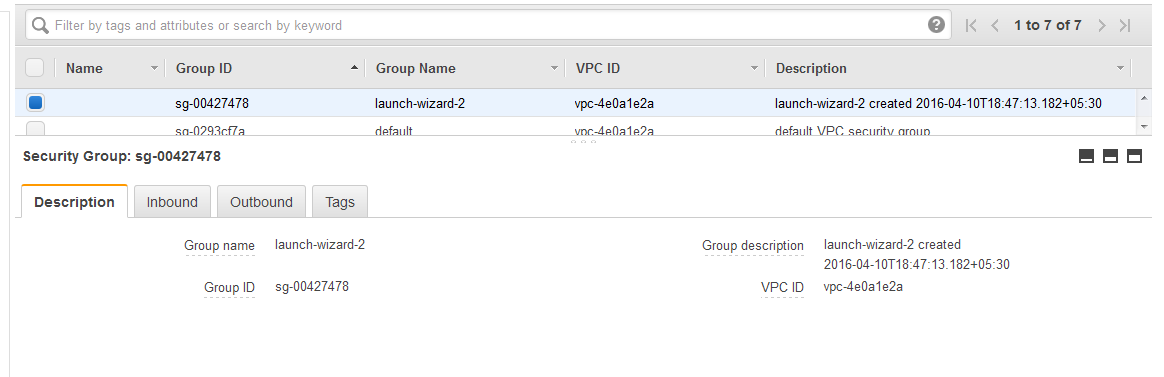


* You can start adding rules, or you can choose Create to create the security group now (you can always add rules later). For more information about adding rules, see [Adding Rules to a Security Group](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html#adding-security-group-rule).

**Security Groups Configurations:**

Select anyone Security Group from the list of security groups.

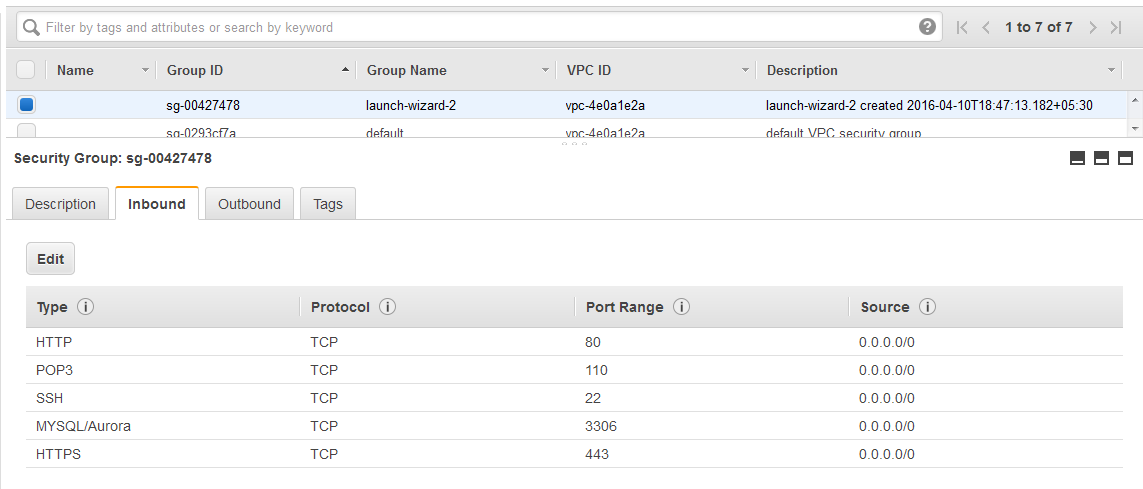
**Security Group Descriptions Image:**



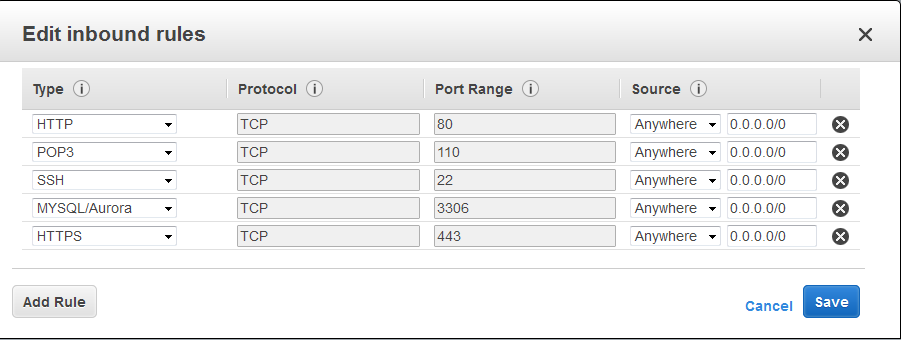
**Inbound:**

* You can allow web servers to receive all inbound HTTP and HTTPS traffic. On the Inbound tab, choose Edit. In the dialog, choose Add Rule. Select HTTP from the Type list, and leave the source as Anywhere (0.0.0.0/0). Add a similar rule for the HTTPS protocol.

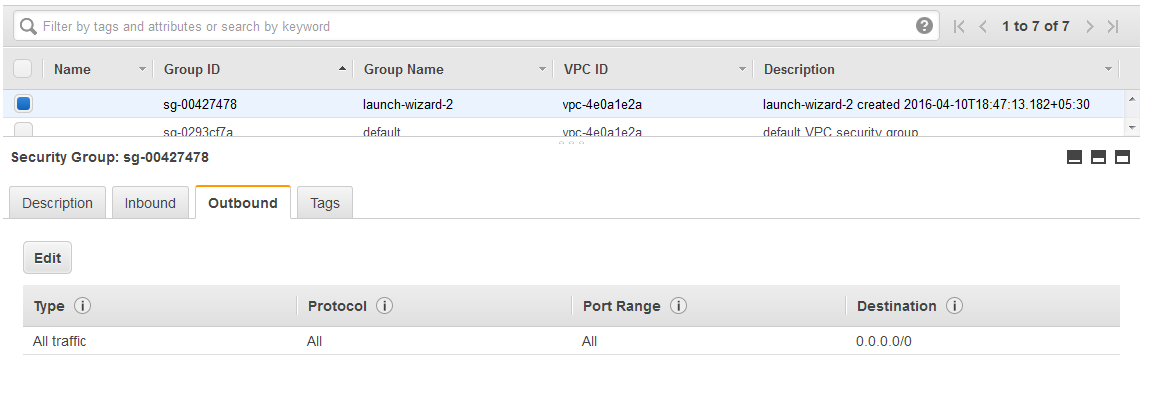
**Security Group Inbound Image:**



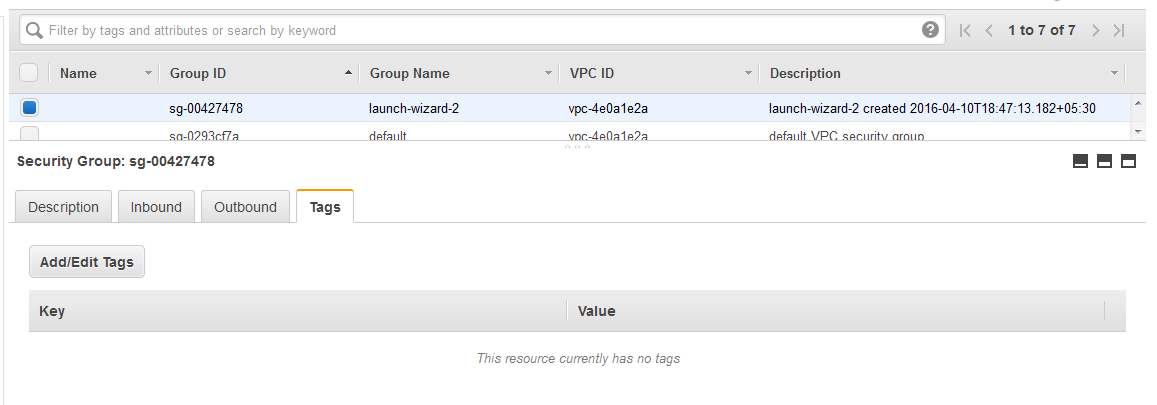
**Security Group Edit Inbound Image:**



**Security Group Outbound Image:**



**Security Group Tags Image:**



**References:**

[**http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html**](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html)

**Configure Putty With SSH:**

After you launch your instance, you can connect to it and use it the way that you'd use a computer sitting in front of you.

**Note**

After you launch an instance, it can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks - you can view this information in the Status Checks column on the Instances page.

The following instructions explain how to connect to your instance using PUTTY, a free SSH client for Windows. If you receive an error while attempting to connect to your instance, see [Troubleshooting Connecting to Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html).

**Prerequisites:**

Before you connect to your Linux instance using PUTTY, complete the following prerequisites:

* **Install PUTTY -** Download and install PUTTY from the PUTTY download page. If you already have an older version of PUTTY installed, we recommend that you download the latest version. Be sure to install the entire suite.
* **Get the ID of the instance** - You can get the ID of your instance using the Amazon EC2 console (from the Instance ID column). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.
* **Get the public DNS name of the instance** - You can get the public DNS for your instance using the Amazon EC2 console (check the Public DNS column; if this column is hidden, click the Show/Hide icon and select Public DNS). If you prefer, you can use the describe-instances (AWS CLI) or ec2-describe-instances (Amazon EC2 CLI) command.
* **Locate the private key** - You'll need the fully-qualified path of the .pem file for the key pair that you specified when you launched the instance.
* **Enable inbound SSH traffic from your IP address to your instance** - Ensure that the security group associated with your instance allows incoming SSH traffic from your IP address. For more information, see Authorizing Network Access to Your Instances.

**Important**

Your default security group does not allow incoming SSH traffic by default.

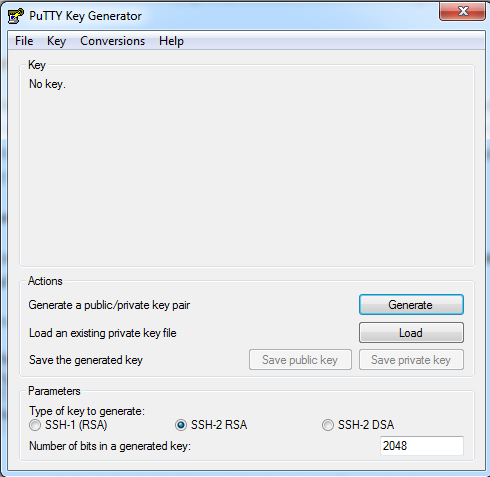
**Converting Your Private Key Using PUTTYgen:**

PUTTY does not natively support the private key format (.pem) generated by Amazon EC2. PUTTY has a tool named PUTTYgen, which can convert keys to the required PUTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PUTTY.

**To convert your private key**

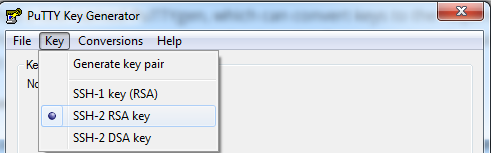
* **Start PUTTYgen** - (for example, from the Start menu, click All Programs > PUTTY > PUTTYgen).

**Putty Key Generator Image**:



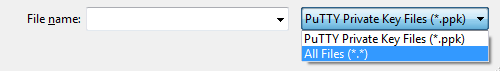
* Under **Type of key** to generate, select **SSH-2 RSA**.

**Putty Key Type Image**:



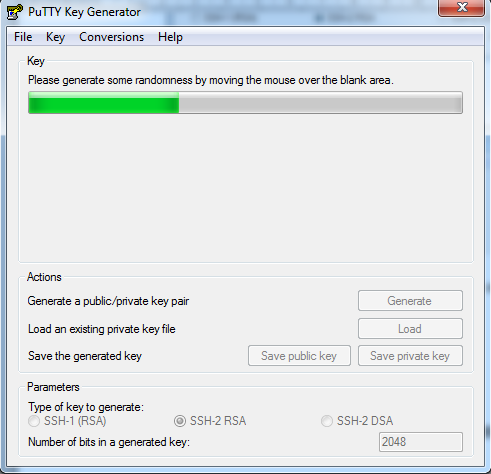
* Click **Load**. By default, PUTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.

**Upload instance pem file image:**

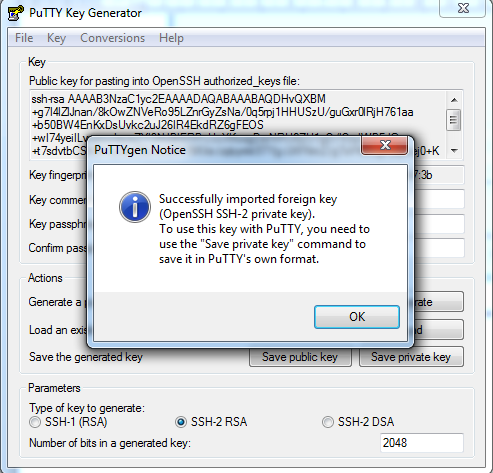


* Select your .pem file for the key pair that you specified when you launch your instance, and then click Open. Click OK to dismiss the confirmation dialog box.
* Click **Generate** button to generate the public key .pem file, while generation mover to be over that blank area.

**Generate Private Key Image:**



**Save Private Key Image:**



* Click **Save private key** to save the key in the format that PUTTY can use. PUTTYgen displays a warning about saving the key without a passphrase. Click Yes.

**Note**

A passphrase on a private key is an extra layer of protection, so even if your private key is discovered, it can't be used without the passphrase. The downside to using a passphrase is that it makes automation harder because human intervention is needed to log on to an instance, or copy files to an instance.

* Specify the same name for the key that you used for the key pair (for example, my-key-pair). PUTTY automatically adds the .ppk file extension.

Your private key is now in the correct format for use with PUTTY. You can now connect to your instance using PUTTY's SSH client.

**Starting a PUTTY Session:**

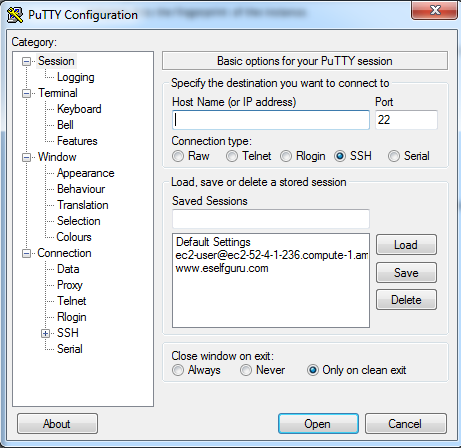
Use the following procedure to connect to your Linux instance using PUTTY. You'll need the .ppk file that you created for your private key. If you receive an error while attempting to connect to your instance, see Troubleshooting Connecting to Your Instance.

**To start a PUTTY session**

(Optional) You can verify the RSA key fingerprint on your instance by using one of the following commands on your local system (not on the instance). This is useful if you've launched your instance from a public AMI from a third party. Locate the SSH HOST KEY FINGERPRINTS section, and note the RSA fingerprint (for example, 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f) and compare it to the fingerprint of the instance.

* **Start PUTTY** (from the Start menu, click All Programs > PUTTY > PUTTY).

**Putty Image:**

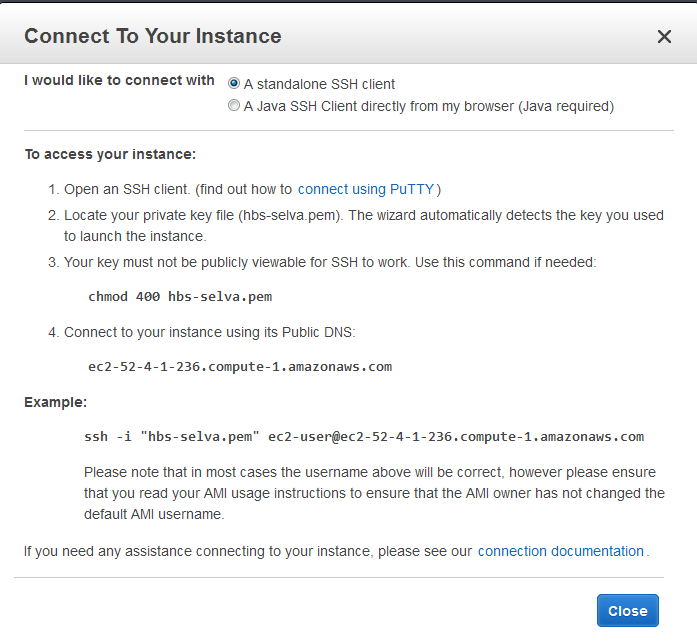


* In the **Category pane**, select **Session** and complete the following fields:
* In the Host Name box, enter user\_name@public\_dns\_name. Be sure to specify the appropriate user name for your AMI. For example:

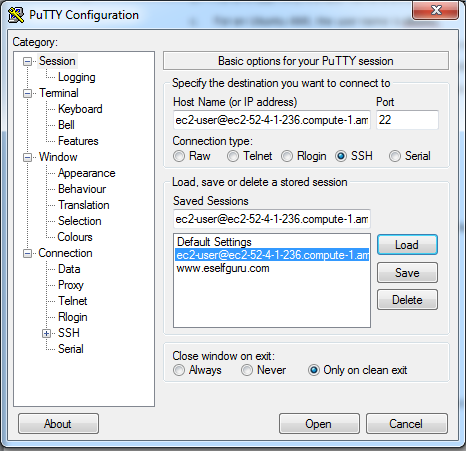
1. For an Amazon Linux AMI, the user name is **ec2-user**.
2. For a RHEL5 AMI, the user name is either **root** or **ec2-user**.
3. For an Ubuntu AMI, the user name is **ubuntu**.
4. For a Fedora AMI, the user name is either **fedora** or **ec2-user**.
5. For SUSE Linux, the user name is either **root** or **ec2-user**.
6. Otherwise, if ec2-user and root don't work, check with the AMI provider

* Under **Connection type**, select **SSH**.
* Ensure that **Port is 22**.
* Get the Host Name or IP Address from EC2 Instance – Right Click the Instance -> Connect

**EC2 Instance Connection Image:**

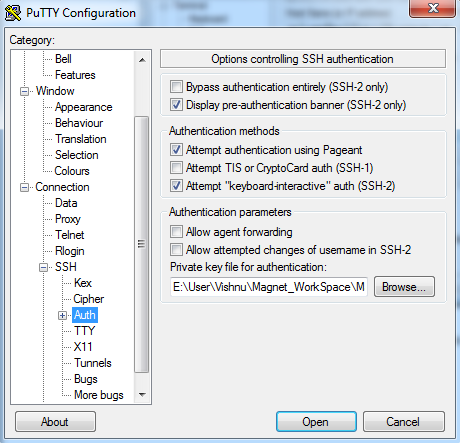


**Load & Open the Putty Session Image:**



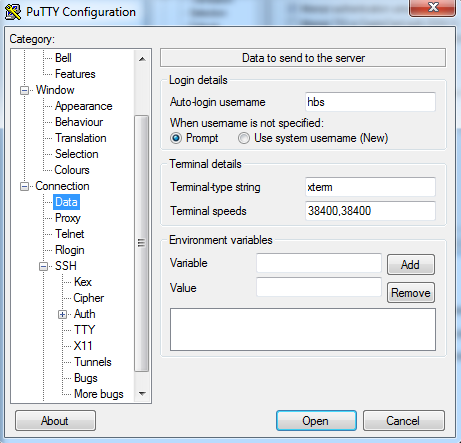
* In the Category pane, **Connection -> SSH -> Select Auth**. Complete the following:

**Browse the private key to open the putty:**



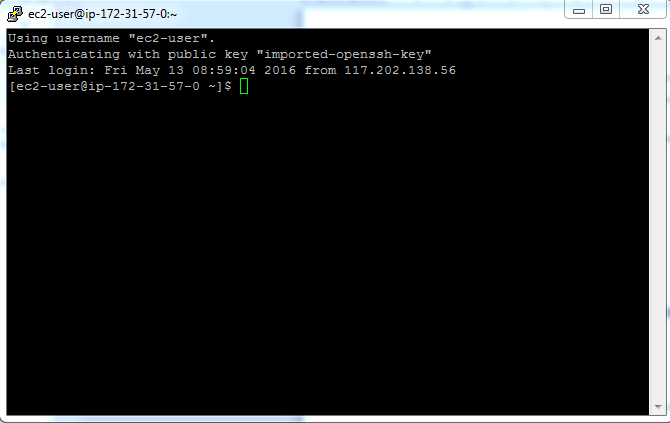
* Click **Browse**.
* Select the **.ppk file** that you generated for your key pair, and then click Open.
* (Optional) If you plan to start this session again later, you can save the session information for future use. Select Session in the Category tree, enter a name for the session in Saved Sessions, and then click Save.
* In the Category pane, **Connection -> Select Data,** Type the **Auto Login Username**.

**Putty login details Auto-Login username Image:**



* Click **Open** to start the PUTTY session.
* If this is the first time you have connected to this instance, PUTTY displays a security alert dialog box that asks whether you trust the host you are connecting to.
* (Optional) Verify that the fingerprint in the security alert matches the fingerprint that you obtained in step 1. If these fingerprints don't match, someone might be attempting a "man-in-the-middle" attack. If they match, continue to the next step.
* **Click Yes**. A window opens and you are connected to your instance.

**Putty with Instance Image:**



**Note**

If you specified a passphrase when you converted your private key to PUTTY's format, you must provide that passphrase when you log in to the instance.

If you receive an error while attempting to connect to your instance, [Troubleshooting Connecting to Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html).

**References:**

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/PUTTY.html>

**Install Required Software:**

For war deployment we need the following software,

* JDK development version (It installed by defiantly)
* Apache Tomcat
* Apache Httpd
* Database – MairaDB

**Apache Tomcat:**

**What is Apache Tomcat?**

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the [Java Community Process](http://jcp.org/en/introduction/overview).

The Apache Tomcat software is developed in an open and participatory environment and released under the [Apache License version 2](http://www.apache.org/licenses). The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](http://tomcat.apache.org/getinvolved.html).

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations.

**Install Tomcat:**

* Now you are ready to install Tomcat 7. Run the following command to install the Tomcat package:

|  |
| --- |
| $ sudo yum install tomcat |

* Answer y at the confirmation prompt to install tomcat. This will install Tomcat 7 and its dependencies, such as Java, and it will also create the tomcat user.
* Most of the important Tomcat files will be located in /usr/share/tomcat. If you already have a Tomcat application that you want to run, you can place it in the /usr/share/tomcat/webapps directory, configure Tomcat, and restart the Tomcat service. In this tutorial, however, we will install a few additional packages that will help you manage your Tomcat applications and virtual hosts.

## Start Tomcat:

* To put our changes into effect, restart the Tomcat service:

|  |
| --- |
| $ sudo service tomcat start |

* If you started the service earlier for some reason, run the restart command instead:

|  |
| --- |
| $ sudo service tomcat restart |

## Stop Tomcat:

* To put our changes into effect, Stop the Tomcat service:

|  |
| --- |
| $ sudo service tomcat stop |

## Check Tomcat Status:

* To check Tomcat service,

|  |
| --- |
| $ sudo service tomcat status |

### Enable Tomcat Service:

* If you want Tomcat to run every time the server is booted up, you will need to enable the service:

|  |
| --- |
| $ sudo systemctl enable tomcat |

* Now we're ready to access the web interface.

**Access the Web Interface:**

Now that Tomcat is up and running, let's access the web management interface in a web browser. You can do this by accessing the public IP address of the server, on port 8080:

Open in web browser:

http://server\_IP\_address:8080

This “server\_IP\_address” may the ec2-instance public dns or elastic ip.

**References:**

<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-7-on-centos-7-via-yum>

**Apache Httpd Server:**

**What is Apache Httpd Server?**

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

**Install Apache Httpd Server:**

* Now you are ready to install Apache Httpd server. Run the following command to install the Httpd package:

|  |
| --- |
| $ sudo yum install httpd |

* Answer y at the confirmation prompt to install httpd. This will apache Httpd and its dependencies.
* Most of the important Apache Httpd files will be located in /etc/httpd/.

**Start Httpd:**

* To put our changes into effect, start the Httpd service:

|  |
| --- |
| $ sudo service httpd start |

* If you started the service earlier for some reason, run the restart command instead:

|  |
| --- |
| $ sudo service httpd restart |

## Stop Httpd:

* To put our changes into effect, Stop the Tomcat service:

|  |
| --- |
| $ sudo service httpd stop |

## Check Httpd Status:

* To check Httpd service status,

|  |
| --- |
| $ sudo service httpd status |

### Verify the Ports:

### To verify the port 80 is open or not using the following command,

|  |
| --- |
| $ netstat -tulpn | grep :80 |

### Then it’s output should be like, given below,

|  |
| --- |
| (No info could be read for "-p": geteuid()=1000 but you should be root.)  tcp6 0 0 :::8080 :::\* LISTEN -  tcp6 0 0 :::80 :::\* LISTEN -  tcp6 0 0 127.0.0.1:8005 :::\* LISTEN - |

* For mapping the Tomcat port 8080 with http port 80, we need add **virtual host** in **/ect/httpd/conf/httpd.conf** file.
* Your server is started, but you need to configure and use Apache as per your requirement. **/etc/httpd/conf/httpd.conf** (RHEL/Cent OS/Fedora Core Linux) or **/etc/apache2/httpd.conf** (Debian / Ubuntu Linux) configuration file. Use a text editor such as **vi** to edit a file.

**References:**

<http://www.cyberciti.biz/faq/linux-install-and-start-apache-httpd/>

**MariaDB Server:**

**What is MariaDB?**

MariaDB is one of the most popular database servers in the world. It’s made by the original developers of MySQL and guaranteed to stay open source.

MariaDB turns data into structured information in a wide array of applications, ranging from banking to websites. It is an enhanced, drop-in replacement for MySQL. MariaDB is used because it is fast, scalable and robust, with a rich ecosystem of storage engines, plugins and many other tools make it very versatile for a wide variety of use cases.

MariaDB is developed as open source software and as a relational database it provides an SQL interface for accessing data. The latest versions of MariaDB also include GIS and JSON features.

**Check Installed MariaDB Server:**

* First We need to check the installed MariaDB services in our instances like below

|  |
| --- |
| $ yum list installed | grep maria |

### Then it’s output should be like, given below,

|  |
| --- |
| mariadb.x86\_64 1:5.5.47-1.el7\_2 @rhui-REGION-rhel-server-releases  mariadb-libs.x86\_64 1:5.5.47-1.el7\_2 @rhui-REGION-rhel-server-releases  mariadb-server.x86\_64 1:5.5.47-1.el7\_2 @rhui-REGION-rhel-server-releases |

* But initially the **mariadb-server.x86\_64** is not installed, so we need to install the MariaDB Server.

**Install MariaDB Server:**

* Now we are ready to MariaDB server. Run the following command to install the MariaDB Server package:

|  |
| --- |
| $ sudo yum install mariadb-server |

* Answer y at the confirmation prompt to install MariaDB Server. This will install the MariaDB Server and its dependencies.

**Start MariaDB Server:**

* To put our changes into effect, start the MariaDB service:

|  |
| --- |
| $ sudo service mariadb start |

* If you started the service earlier for some reason, run the restart command instead:

|  |
| --- |
| $ sudo service mariadb restart |

## Stop MariaDB Server:

* To put our changes into effect, Stop the MairaDB service:

|  |
| --- |
| $ sudo service mariadb stop |

## Check MariaDB Server Status:

* To check MariaDB service status,

|  |
| --- |
| $ sudo service mariadb status |

### Enable MariaDB Server:

### To **Auto Start** MariaDB service, execute the following command,

|  |
| --- |
| $ sudo systemctl enable mariadb.service |

### Configure MySQL:

### After Start the MariaDB server, we need to configure the MySQL Secure Installation, execute the following command,

|  |
| --- |
| $ /usr/bin/mysql\_secure\_installation |

### Then it’s output should be like, given below, and it asks the following settings to configure

### Current Password of Root – Initially there is no password for root user

1. **Set root password? [Y/n]** – If we need new password, then we change the password here.
2. **Remove anonymous users? [Y/n]** – It asks Yes/No to remove the anonymous users.

## Disallow root login remotely? [Y/n] – It asks Yes/No to disallow the root login remotely.

## Reload privilege tables now? [Y/n] y – It asks Yes/No to reload the privilege tables now.

|  |
| --- |
| /usr/bin/mysql\_secure\_installation: line 379: find\_mysql\_client: command not foundNOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDBSERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!In order to log into MariaDB to secure it, we'll need the currentpassword for the root user. If you've just installed MariaDB, andyou haven't set the root password yet, the password will be blank,so you should just press enter here.Enter current password for root (enter for none):OK, successfully used password, moving on...Setting the root password ensures that nobody can log into the MariaDBroot user without the proper authorisation.Set root password? [Y/n] yNew password:Re-enter new password:Password updated successfully!Reloading privilege tables..... Success!By default, a MariaDB installation has an anonymous user, allowing anyoneto log into MariaDB without having to have a user account created forthem. This is intended only for testing, and to make the installationgo a bit smoother. You should remove them before moving into aproduction environment.Remove anonymous users? [Y/n] y... Success!Normally, root should only be allowed to connect from 'localhost'. Thisensures that someone cannot guess at the root password from the network.Disallow root login remotely? [Y/n] n... skipping.By default, MariaDB comes with a database named 'test' that anyone canaccess. This is also intended only for testing, and should be removedbefore moving into a production environment.Remove test database and access to it? [Y/n] n... skipping.Reloading the privilege tables will ensure that all changes made so farwill take effect immediately.Reload privilege tables now? [Y/n] y... Success!Cleaning up...All done! If you've completed all of the above steps, your MariaDBinstallation should now be secure.Thanks for using MariaDB! |

**Access MariaDB Monitor:**

### To access the MariaDB monitor, execute the below command

|  |
| --- |
| $ mysql -u root -p |

### Then it shows an output should be like, given below, & it asks root password to authenticate,

|  |
| --- |
| Enter password:  Welcome to the MariaDB monitor. Commands end with ; or \g.  Your MariaDB connection id is 35  Server version: 5.5.47-MariaDB MariaDB Server  Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement. |

**SQL Queries to Initial Configuration:**

* List the all databases

|  |
| --- |
| MariaDB [(none)]> show databases;  +--------------------+  | Database |  +--------------------+  | information\_schema |  | mysql |  | performance\_schema |  +--------------------+  3 rows in set (0.00 sec) |

### To configure the MySQL with MairaDB

|  |
| --- |
| MariaDB [(none)]> use mysql  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  Database changed  MariaDB [mysql]> |

### To list the users of mysql, execute the below queries,

|  |
| --- |
| MariaDB [mysql]> SELECT User FROM mysql.user;  +------+  | User |  +------+  | root |  | root |  | root |  | root |  +------+  4 rows in set (0.00 sec) |

|  |
| --- |
| MariaDB [mysql]> SELECT User,Host,Password FROM mysql.user;    5 rows in set (0.00 sec) |

|  |
| --- |
| MariaDB [mysql]> SELECT \* FROM mysql.user;    5 rows in set (0.00 sec) |

* To Create a new User for Database,
* For localhsot (private) user, execute the below command,

|  |
| --- |
| MariaDB [mysql]> CREATE USER 'hbs'@'localhost' IDENTIFIED BY 'xxxxxxxx';  Query OK, 0 rows affected (0.00 sec)4 rows in set (0.00 sec) |

* For remote access (public) user, execute the below command,

|  |
| --- |
| MariaDB [mysql]> CREATE USER 'visnu'@'%' IDENTIFIED BY 'xxxxxxxx';  Query OK, 0 rows affected (0.00 sec)4 rows in set (0.00 sec) |
| ‘xxxxxxx’ means password i.e ‘magnet’ |

### Assign Privileges for the User;

* To assign the Privileges for the user, execute the below command to assign all privileges to the user,

|  |
| --- |
| MariaDB [mysql]> GRANT ALL PRIVILEGES ON \*.\* TO 'hbs'@'localhost';  Query OK, 0 rows affected (0.00 sec)  MariaDB [mysql]> GRANT ALL PRIVILEGES ON \*.\* TO 'visnu'@'%';  Query OK, 0 rows affected (0.00 sec) |

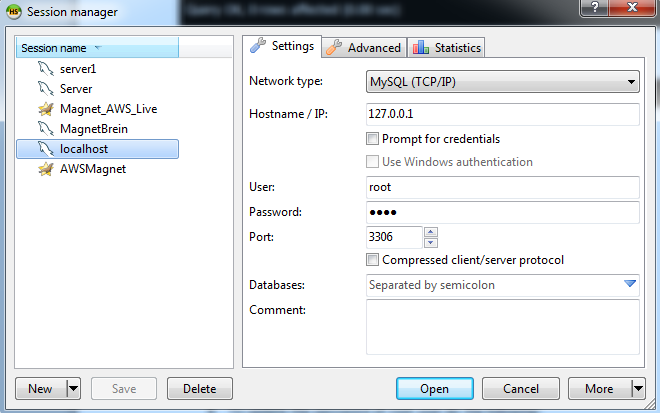
* Again, we must flush privileges for these changes to happen:

|  |
| --- |
| MariaDB [mysql]> FLUSH PRIVILEGES; |

### Remotely Access the MariaDB Server:

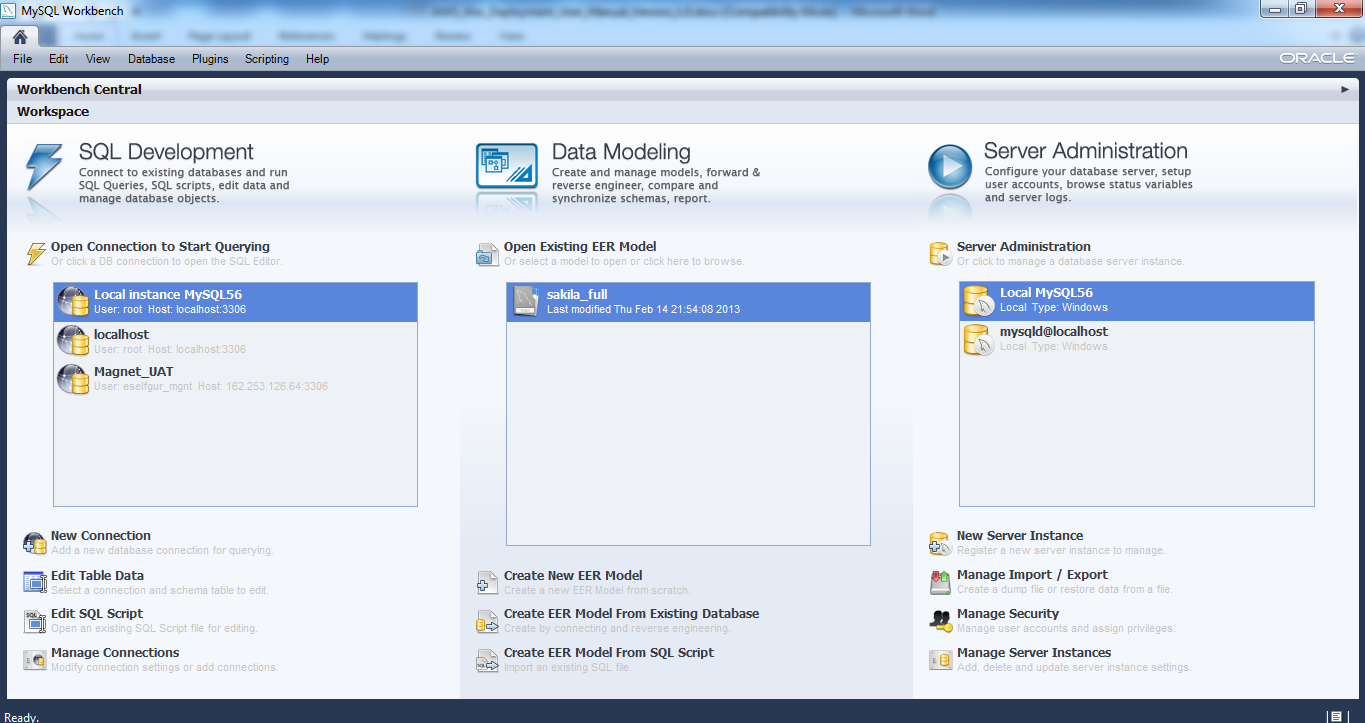
* We can remotely access the MariaDB Server using MySQL Wrokbench or HeidiSQL.
* For HeidiSQL : Start -> HeidiSQL -> HeidiSQL

**HeidiSQL Image:**



* For MySQL Workbench : Start -> MySQL -> MySQL Workbench 5.2 CE

**MySQL Workbench Image:**



* To Create a New Connection in remotely, we need the below information
* Connection Name
* Hostname/IP – AWS EC2 instance public dns
* Username – Created User in MairaDB Server
* Password – User Password
* Port – Commonly 3306
* Next, from the AWS Management Console find the Security Group that you assigned to your instance during set-up and add 'MySQL' to the group. You can also manually add port 3306 and Assign Anywhere. Finally Save.

**References:**

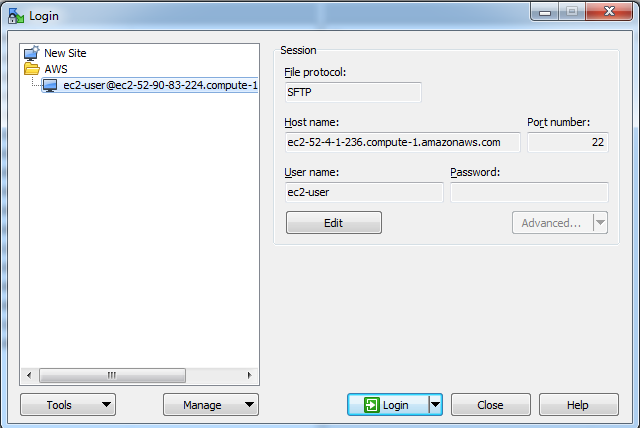
<https://ask.fedoraproject.org/en/question/43459/how-to-start-mysql-mysql-isnt-starting/>

<https://www.digitalocean.com/community/tutorials/how-to-secure-mysql-and-mariadb-databases-in-a-linux-vps>

**Transfer Files:**

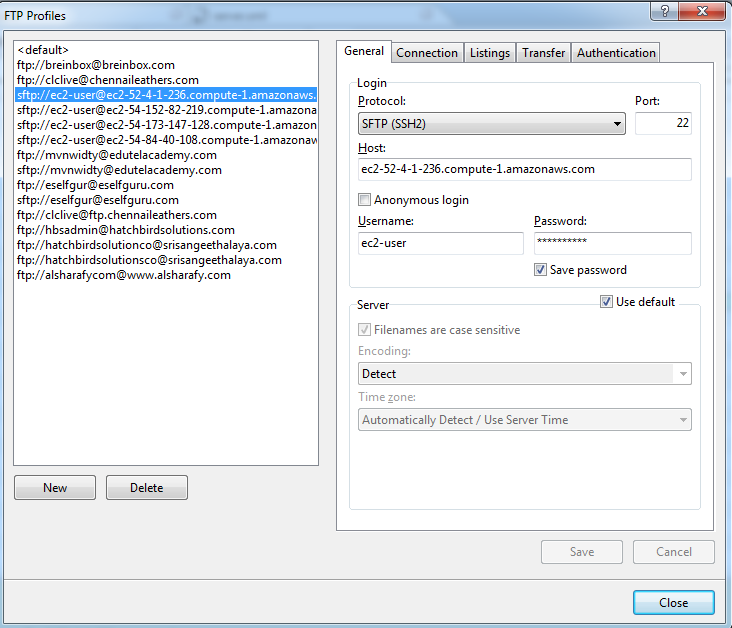
* We can transfer the files from our local system to EC2-Instance, we need to use WinSCP/Beyond Comparer Tools.
* Using WinSCP

**Connect WinSCP Image:**



* Using Beyond comparer

**Connect Beyond Comparer Image:**



**Route 53:**

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS. You can use Amazon Route 53 to configure DNS health checks to route traffic to healthy endpoints or to independently monitor the health of your application and its endpoints. Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally through a variety of routing types, including Latency Based Routing, Geo DNS, and Weighted Round Robin—all of which can be combined with DNS Failover in order to enable a variety of low-latency, fault-tolerant architectures. Using Amazon Route 53 Traffic Flow’s simple visual editor, you can easily manage how your end-users are routed to your application’s endpoints—whether in a single AWS region or distributed around the globe. Amazon Route 53 also offers Domain Name Registration – you can purchase and manage domain names such as example.com and Amazon Route 53 will automatically configure DNS settings for your domains.

**Route 53 Dashboard Image:**

## 

## Amazon Route 53 performs three main functions:

## Domain registration – Amazon Route 53 lets you register domain names such as example.com.

## Domain Name System (DNS) service – Amazon Route 53 translates friendly domains names like www.example.com into IP addresses like 192.0.2.1. Amazon Route 53 responds to DNS queries using a global network of authoritative DNS servers, which reduces latency.

## Health checking – Amazon Route 53 sends automated requests over the Internet to your application to verify that it's reachable, available, and functional.

## You can use any combination of these functions. For example, you can use Amazon Route 53 as both your registrar and your DNS service, or you can use Amazon Route 53 as the DNS service for a domain that you registered with another domain registrar.

## Create New Hosted Zones:

## Click the “Create Hosted Zone” button to create the new Hosted Zone for the registered Domain.

## Hosted Zone Image:

## 

## For Domain Name, enter the name of your domain.

## Optional: For Comment, enter a comment about the hosted zone.

## Click Create.

## Create Hosted Zone Image:

## 

## Create Record Sets:

## When you create basic resource record sets, you specify the following values:

## Name

## Type

## Alias

## TTL (Time to Live)

## Value

## Routing Policy

## We need to create the Record sets for the corresponding Hosted Zone

## Created Hosted Zone Image:

## 

## Click the “Go to Record Sets” button to open the Record Set page

## List of Created Record Sets Image:

## 

## Click “Create Record Set” button to create a new record set

## Create New Record Set Image:

## 

## Name:

## Enter the name of the domain or subdomain for which you're creating the resource record set. The default value is the name of the hosted zone. If you're creating a resource record set that has the same name as the hosted zone, don't enter a value (for example, an @ symbol) in the Name field.

## For information about how to specify characters other than a-z, 0-9, and - (hyphen) and how to specify internationalized domain names, see DNS Domain Name Format.

## You can use an asterisk (\*) character in the name. DNS treats the \* character either as a wildcard or as the \* character (ASCII 42), depending on where it appears in the name. For more information, see Using an Asterisk (\*) in the Names of Hosted Zones and Resource Record Sets.

## Important

## You can't use the \* wildcard for resource records sets that have a type of NS.

## Types:

## The DNS record type. For more information, see Supported DNS Resource Record Types.

## Select the value for Type based on how you want Amazon Route 53 to respond to DNS queries.

## Available Types,

## A – IPV4 Address

## CNAME – Canonical Name

## MX – Mail Exchange

## AAAA – IPV6 Address

## TXT – Text

## PTR – Pointer

## SRV – Service Locator

## SRF – Sender Policy Framework

## NS – Name Server

## SOA – Start Of Authority

## Alias:

## We should Select No.

**TTL (Time to Live):**

## The amount of time, in seconds, that you want DNS recursive resolvers to cache information about this resource record set. If you specify a longer value (for example, 172800 seconds, or two days), you pay less for Amazon Route 53 service because recursive resolvers send requests to Amazon Route 53 less often. However, it takes longer for changes to the resource record set (for example, a new IP address) to take effect because recursive resolvers use the values in their cache for longer periods instead of asking Amazon Route 53 for the latest information.

## If you're associating this resource record set with a health check, we recommend that you specify a TTL of 60 seconds or less so clients respond quickly to changes in health status.

## Value:

## Enter a value that is appropriate for the value of Type. For all types except CNAME, you can enter more than one value. Enter each value on a separate line

## A — IPv4 address - An IP address in IPv4 format, which is an Elastic IP of corresponding ec2 instance, for example, 52.4.1.236.

**A Record Set Images:**

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\autoconfig.brein.sg.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\autodiscover.brein.sg.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\mail.brein.sg.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\brein.sg_A.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\magnet.brein.sg.png

## AAAA — IPv6 address - An IP address in IPv6 format, for example, 2001:0db8:85a3:0:0:8a2e:0370:7334.

## CNAME — Canonical name - The fully qualified domain name (for example, www.example.com) that you want Amazon Route 53 to return in response to DNS queries for this resource record set. A trailing dot is optional; Amazon Route 53 assumes that the domain name is fully qualified. This means that Amazon Route 53 treats www.example.com (without a trailing dot) and www.example.com. (with a trailing dot) as identical.

## CNAME Record Set Image:

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\_domainkey.brein.sg.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\ftp.brein.sg.png E:\HBS-Vishnu\Amazon_Web\12_May_2016\imap.brein.sg.png E:\HBS-Vishnu\Amazon_Web\12_May_2016\m3lgddudovuxekzgigeplq2ledrtuzb3._domainkey.brein.sg.png

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\smtp.brein.sg.png E:\HBS-Vishnu\Amazon_Web\12_May_2016\pop.brein.sg.png

## MX — Mail exchange - A priority and a domain name that specifies a mail server, for example, 10 mailserver.example.com.

## [priority] [mailserver.example.com]

## MX Record Set Image:

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\mx.brein.sg.png

## NS — Name server - The domain name of a name server, for example, ns1.example.com.

## NS Record Set Image:

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\brein.sg_NS.png

## PTR — Pointer - The domain name that you want Amazon Route 53 to return.

## SOA — Start of Authority - Basic DNS information about the domain. For more information, see [The Start of Authority (SOA) Resource Record Set](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/SOA-NSrecords.html#SOArecords)

## SOA Record Set Image:

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\brein.sg_SOA.png

## SPF — Sender Policy Framework - An SPF record enclosed in quotation marks, for example, "v=spf1 ip4:192.168.0.1/16-all". SPF records are not recommended. For more information, see [Supported DNS Resource Record Types](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/ResourceRecordTypes.html).

## SRV — Service locator - An SRV record. For information about SRV record format, refer to the applicable documentation. The format of an SRV record is:

## [priority] [weight] [port] [server host name]

## For example: 1 10 5269 xmpp-server.example.com.

## SRV Record Set Image:

## E:\HBS-Vishnu\Amazon_Web\12_May_2016\_autodiscover_tcp.brein.sg.png

## TXT — Text - A text record. Enclose text in quotation marks, for example, "Sample Text Entry".

## TXT Record Set Image:E:\HBS-Vishnu\Amazon_Web\12_May_2016\_amazonses.brein.sg.png

## Routing Policy:

## We should Select Simple.

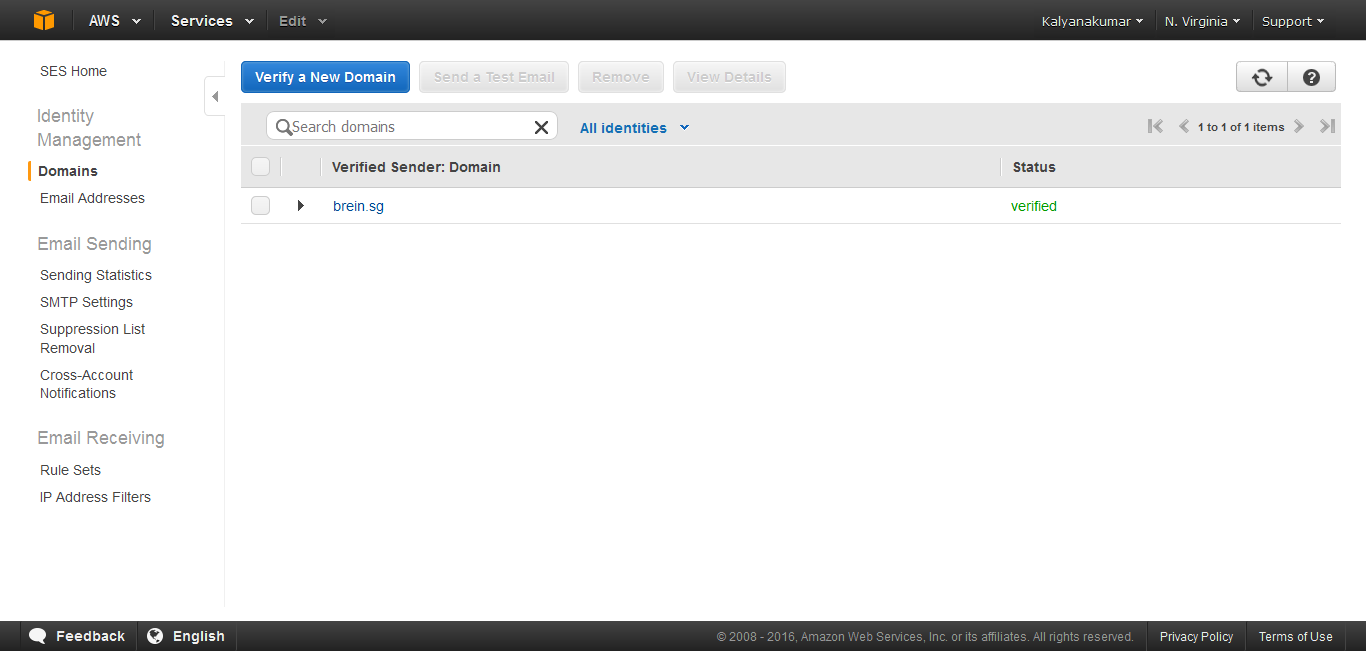
## References:

## <http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-values-basic.html>

**SES (Simple Email Service):**

Amazon Simple Email Service (Amazon SES) is a cost-effective email service built on the reliable and scalable infrastructure that Amazon.com developed to serve its own customer base. With Amazon SES, you can send transactional email, marketing messages, or any other type of high-quality content to your customers. You can also use Amazon SES to receive messages and deliver them to an Amazon S3 bucket, call your custom code via an AWS Lambda function, or publish notifications to Amazon SNS. With Amazon SES, you have no required minimum commitments – you pay as you go, and you only pay for what you use.

**SES Dashboard Image:**



**War Deployment:**

* Get ready the war file from development team.
* Initially Stops the Tomcat & Httpd Server
* Create the database with the corresponding instance.
* Connect the EC2 Instance with WinSCP or Beyond Comparer to move the war file from local system to Instance tomcat location (/usr/share/tomcat/webapps/) and rename the war file name as ROOT.war.
* Once the war is ready in webapps folder, than starts the Httpd & Tomcat server
* Before that, we need to do some configuration in both server conf files
* In **Httpd Server conf** files do the following steps.
* Open the **\etc\httpd\conf\httpd.conf** file.
* Add the below coding as end of the file and Save it.
* The “**DocumentRoot**” points the corresponding war file location in tomcat webapps
* The “**ServerName**” points the corresponding mapping domain
* The “**SeverAlias**” points the corresponding mapping domain’s alias name
* The “**ProxyPass**” & “**ProxyPassReverse**” points the **http://localhost:8080**

|  |
| --- |
| <VirtualHost \*:80>  ServerAdmin webmaster@localhost  DocumentRoot /usr/share/tomcat/webapps/ROOT  ServerName www.brein.sg  ServerAlias brein.sg  LogLevel warn  <Location />  Order allow,deny  Allow from all  </Location>  ProxyRequests Off  ProxyPreserveHost On  <Proxy \*>  Order deny,allow  Allow from all  </Proxy>  ProxyPass / http://localhost:8080/ retry=0 timeout=5  ProxyPassReverse / http://localhost:8080/  </VirtualHost> |

* In **Tomcat Server conf** files
* Open the **/usr/share/tomcat/conf/server.xml** file.
* Find the **8080 port connector** and change its configuration below like and save it.

|  |
| --- |
| <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" proxyName="brein.sg" proxyPort="80"/> |

## Here I additionally add the proxyName & proxyPort

## The “proxyName” points the corresponding mapping domain alias name.

## The “proxyPort” points the corresponding EC2 instance http port (80).

## After complete these changes, we should start the both Httpd and Tomcat servers.

## Finally our project works well.

## For file uploading and downloading, we need the change the permission as 777 for the below directories,

## Repository Location (/home/ec2-user/Repository Location)

## Content Session folder (/usr/share/tomcat/webapps/ROOT/content)

**Jar Deployment:**

## Here we need to deploy the following executable jars,

## Reading Resumes from emails (hbsap-mail-reader-1.0.1.jar)

## Sending Emails to corresponding recipients (hbsap-message-sender-1.0.1.jar)

## Required Files & Procedures:

## Create Base Folder under /home/ec2-user

## The Base Folder must contains the following files/folders

## “properties” folder – which contains what are the property files need for the corresponding jar (Database property, Log 4j property)

## “lib” folder – which contains the what are the dependency jars needed for the core

## “Executable Jar” – which performs the corresponding functionality using crontab

## The crontab concepts briefly explain in another document. Please [Click Here](CronTab/AWS_CronTab.docx) to Check.

## Putty Commands:

## Use sudo before the commands if not execute.

|  |  |
| --- | --- |
| **COMMANDS** | **DESCRIPTIONS** |
| $ sudo su - | Access Root Login |
| $ pwd | Present Working Directory |
| $ ls | List the files and folders of the current directory |
| $ ls –ltr | List the files and folders of the current directory with its permissions |
| $ cd | Change Directory |
| $ cd foldername | Change Directory to the given folder name |
| $ cd ../ | Change Directory to one level back |
| $ cp source destination | Copy the source file/folder to the destination directory |
| $ mv source destination | Move the source file/folder to the destination directory |
| $ cat filename | View the given file name |
| $ vi filename | Edit the given file name with editor |
| Press [Esc] & Type :wq | Save & Exit the Editor |
| Press [Esc] & Type :q | Exit the Editor |
| Press [Esc] & Type :q! | Exit without saving |
| $ history | View the previously used commands |
| $ clear | Clear the screen |
| $ exit | Exit the session/Close the Terminal |
| $ chmod | Change the permissions of files or directories |
| $ chmod permission filename | Change the permission(Like 777, 655, 766, …) for the given file or directory |
| $ chmod -R permission filename | Change the permission(Like 777, 655, 766, …) for the given file or directory recursively |
| $ chmod o+x /home/ec2-user/RepositoryLocation | Change the permission for the repository location for upload and download the files from application. |
| rm filename/foldername | Delete the given file or folder name |
| rm –rf filename/foldername | Delete the given file or folder name as recursively |
|  |  |
|  |  |

**Issue Analysis:**

**Issue -1:**

|  |  |
| --- | --- |
| File Permissions issues while download and upload the files from application | |
| File |  |
| Solution | 1. Change the Repository location file permission using the below command   ***$ chmod o+x /home/ec2-user/RepositoryLocation***   1. Change the Content session folder permission under the /usr/share/tomcat/webapps/Root, using the below command   ***$ chmod -R 777 /usr/share/tomcat/webapps/Root/content*** |
| Reference | Check under Putty Commands Title |

**Issue -2:**

|  |  |
| --- | --- |
| Mapping domain with the aws EC2 Instance | |
| File |  |
| Solution | Add virtual host in the file /etc/Httpd/conf/httpd.conf |
| Reference | Check under War Development Title |

**Issue -3:**

|  |  |
| --- | --- |
| Permission denied: AH00957: HTTP: attempt to connect to 127.0.0.1:8080 (localhost) failed | |
| File | /etc/httpd/logs/error.log |
| Solution | setsebool -P httpd\_can\_network\_connect 1 |
| Reference | <https://wiki.apache.org/httpd/13PermissionDenied> |

**Issue -4:**

|  |  |
| --- | --- |
| Connection refused: proxy: HTTP: attempt to connect to 127.0.0.1:8080 (localhost) failed | |
| File | /etc/httpd/logs/error.log |
| Solution | 1. Add Proxy Name & Proxy Port in connector port 8080 in /usr/shared/tomcat/conf/server.xml 2. Add “retry=0 timeout=5” in virtual host in /etc/httpd/conf/httpd.conf |
| Reference | <https://velenux.wordpress.com/2012/10/11/solving-apache-503-service-temporarily-unavailable-error/> |

**Issue -5:**

|  |  |
| --- | --- |
| Cache Problem | |
| File |  |
| Solution | 1. Open putty and login as root user 2. Delete the Catalina folder under /usr/share/tomcat/work |
| Reference |  |

Issue -6

|  |  |
| --- | --- |
| **[proxy\_http:error] [pid 14614] (70007)The timeout specified has expired:** | |
| File | etc/httpd/conf/httpd.conf |
| Solution | Find out : var/log/httpd/error.log   1. etc/httpd/conf/httpd.conf under the line Added this   Timeout 600  ProxyTimeout 600 |
| Reference | http://stackoverflow.com/questions/31975951/proxy-ajperror-70007the-timeout-specified-has-expired |

**[proxy\_http:error] [pid 14614] (70007)The timeout specified has expired:**

<http://stackoverflow.com/questions/31975951/proxy-ajperror-70007the-timeout-specified-has-expired>

---------------------------------------------------------------------------------------------------------------------