



Deploying your First Java Application to AWS-2 (AWS EC2)

Hands-On Workshop | Digital Summit '18

Miracle Innovation Labs

Miracle Software Systems, Inc.

Deploying your First Java Application to AWS-2 (AWS EC2)

Introduction

The goal of this document is to show you how to create an EC2 instance, install and configure all the necessary software for deploying and running a sample application onto a web server.

This guide was prepared by [Miracle's Innovation Labs!](#)

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the workshop (Both PC and MAC are compatible). The following pre-requisites will help you to make the workshop experience easier.

- AWS account
- Download and install PuTTY
- Download and install Tomcat

Technology Involved

- AWS account
- Java
- Apache Tomcat
- PuTTY (for windows)
- Git

Lab Steps

Let us get start the Workshop!

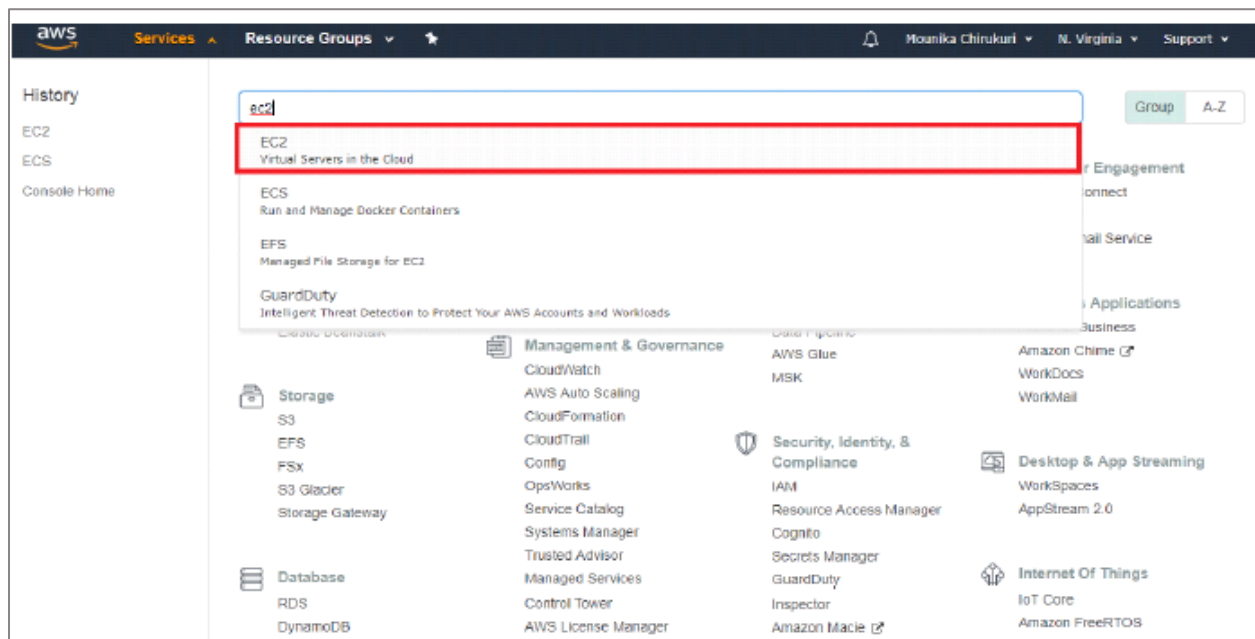
We will show you how to login into AWS Management Console and spin EC2 instances which are virtual machines. These virtual machines are managed by AWS and are called Elastic Compute Cloud. While spinning the instances, we will perform security group configurations which acts as firewall at instance level, generate private key which will be used to login to the instance.

Installing Java which is a basic requirement for running Apache Tomcat Server.

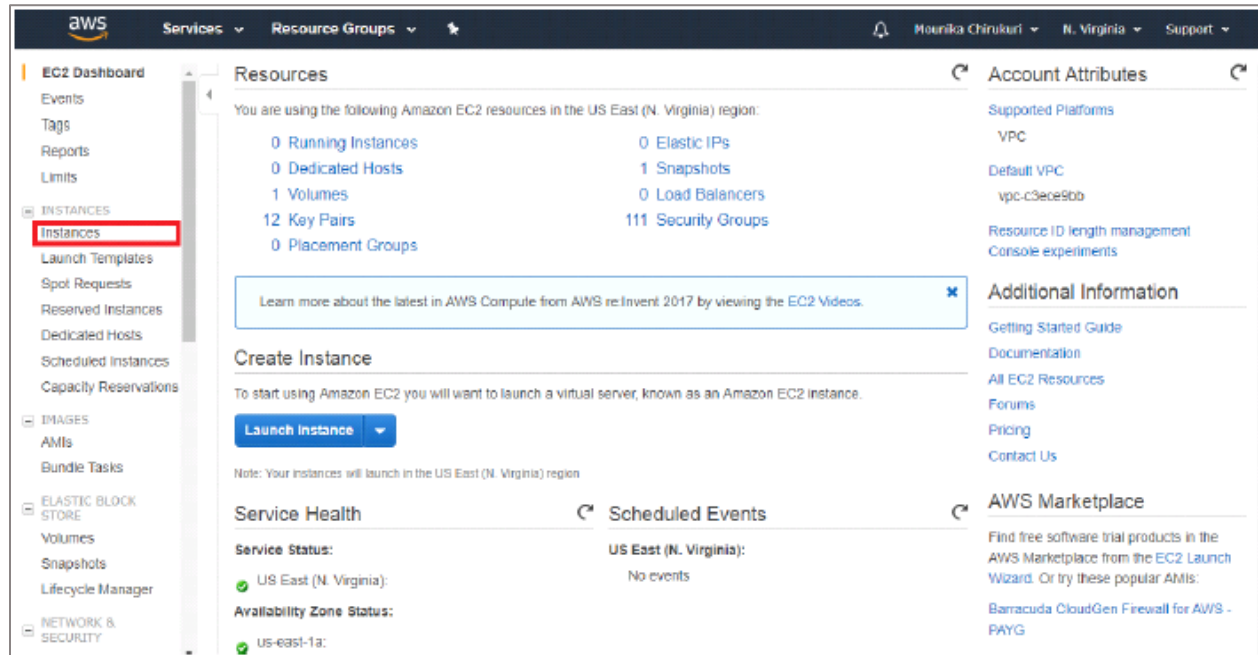
Configure Apache Tomcat on port 80, login user configurations with specific username and password. Once basic configurations are done, we will show you how to deploy the application onto Web Server Apache Tomcat.

Step #1 | AWS EC2 Instance Creation

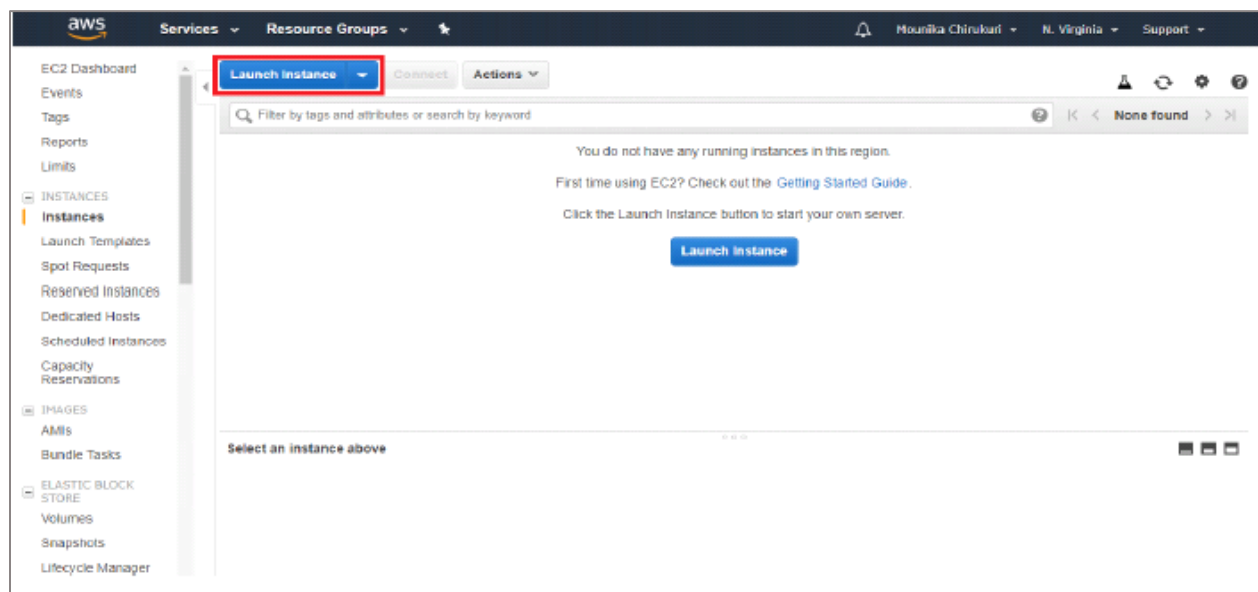
Goto AWS console and enter EC2 in the search bar as shown below and select the EC2 from the list of AWS services.



Now click on **Instances** which is at the left side menu.



Click on **Launch Instance**.



Click on **Select** for the **Amazon Linux 2 AMI (HVM)** option.

The screenshot shows the AWS console interface for Step 1: Choose an Amazon Machine Image (AMI). The top navigation bar includes the AWS logo, Services, Resource Groups, and user information. The main header shows the progress: 1. Choose AMI (active), 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. Below the header, there's a search bar and a 'Quick Start' section. On the left, there's a sidebar with 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. The main content area lists three AMIs: Amazon Linux 2 AMI (HVM), SSD Volume Type; Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type; and SUSE Linux Enterprise Server 15 (HVM), SSD Volume Type. Each AMI has a 'Select' button. The 'Select' button for the Amazon Linux 2 AMI is highlighted with a red box.

Select Instance type as **t2.micro** which is eligible for free tier and click on **Next: Configure Instance Details**.

The screenshot shows the AWS console interface for Step 2: Choose an Instance Type. The top navigation bar is the same as in Step 1. The main header shows the progress: 1. Choose AMI, 2. Choose Instance Type (active), 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. Below the header, there's a 'Filter by' section with 'All instance types', 'Current generation', and 'Show/Hide Columns'. The main content area shows a table of instance types. The 't2.micro' instance type is selected and highlighted with a red box. Below the table, there are buttons: 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Instance Details' (highlighted with a red box).

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

In this page, you can configure Number of instances, VPC, Subnet, Public IP etc., for now leave the default values and click on **Next: Add Storage**.

Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: ☐ Request Spot instances

Network: vpc-c3ace1b0 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: ☐ Add instance to placement group

Capacity Reservation: Open [Create new Capacity Reservation](#)

IAM role: None [Create new IAM role](#)

Shutdown behavior: Stop

Enable termination protection: ☐ Protect against accidental termination

[Cancel](#) [Previous](#) [Review and Launch](#) **[Next: Add Storage](#)**

In this page you can configure volume and for now leave the default values and click on **Next: Add Tags**.

Step 4: Add Storage
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-05c1b4e03900ec07b	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

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

[Cancel](#) [Previous](#) [Review and Launch](#) **[Next: Add Tags](#)**

Click on **Add Tag** to give custom name for the instance.

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

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances ⓘ	Volumes ⓘ
This resource currently has no tags			
Choose the Add tag button or click to add a Name tag . Make sure your IAM policy includes permissions to create tags.			

Add Tag (Up to 50 tags maximum)

Enter the below values.

Key : Name

Value :<your-default-name>

Click on **Next : Configure Security Group**

aws Services Resource Groups

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

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances ⓘ	Volumes ⓘ
Name	MyTomcatAndJenkins	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) **Next: Configure Security Group**

Select **Create a new security Group** and enter name for security group and click on **Add Rule** to open ports.

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

Step 6: Configure Security Group

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

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

Security group name:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere 0.0.0.0/0 ::0	e.g. SSH for Admin Desktop

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

Select **HTTP** and **Source** as **Anywhere**, and click on **Review and Launch**.

aws Services Resource Groups Mounika Chirukuri N. Virginia Support

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

Step 6: Configure Security Group

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

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

Security group name:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere 0.0.0.0/0 ::0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0 ::0	e.g. SSH for Admin Desktop

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

If you want to modify anything or review, you can check here and click on **Launch**.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, MyTomcatAndJenkins_SG, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

▼ **AMI Details** [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-009d6802948d06e52

Free tier eligible Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

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

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

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

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

You need to create a key pair for your instance to SSH into instance. For that select **Create a new key pair**.

Key pair name : <your-key-pair-name>

Click on **Download Key Pair**.

Note - If you forget to download this **.pem** file, you will not be able to SSH to this instance.

Select an existing key pair or create a new key pair

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

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

Create a new key pair

Key pair name
TomcatAndJenkinsKey

Download Key Pair

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

Cancel

Launch Instances

Click on **Launch Instances**.

Select an existing key pair or create a new key pair

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

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

Create a new key pair

Key pair name
TomcatAndJenkinsKey

Download Key Pair

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

Cancel

Launch Instances

Click on **View Instances** to navigate to EC2 home page.

Launch Status

✓ **Your instances are now launching**
The following instance launches have been initiated: [i-0940d0a5cbd37d860](#) [View launch log](#)

ℹ **Get notified of estimated charges**
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

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

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

▼ Here are some helpful resources to get you started

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

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes. (Additional charges may apply)
- Manage security groups

[View Instances](#)

After navigating to EC2 home page, select your instance and check for the **IPv4 Public IP** which is at under **Description** tab.

EC2 Dashboard

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
MyTomcatAn...	i-0940d0a5cbd37d860	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-18-206-186-27.co...	18.206.186.27

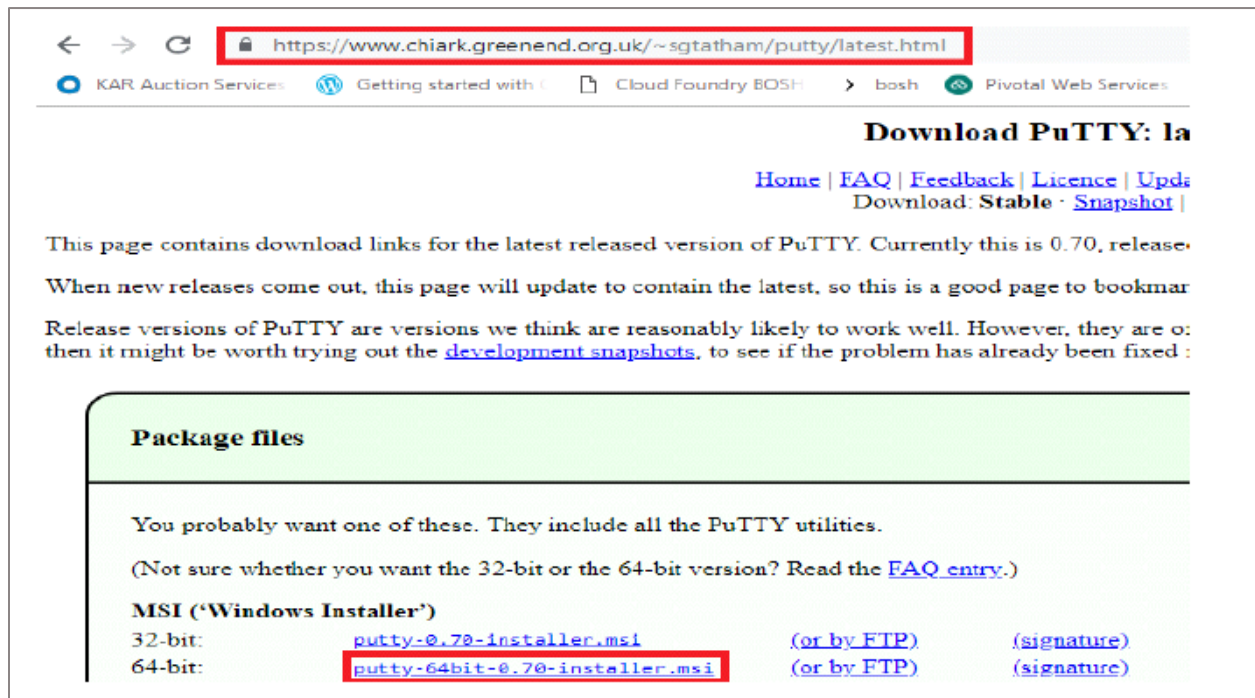
Instance: [i-0940d0a5cbd37d860](#) (MyTomcatAnJenkins_SG) Public DNS: [ec2-18-206-186-27.compute-1.amazonaws.com](#)

Description Status Checks Monitoring Tags

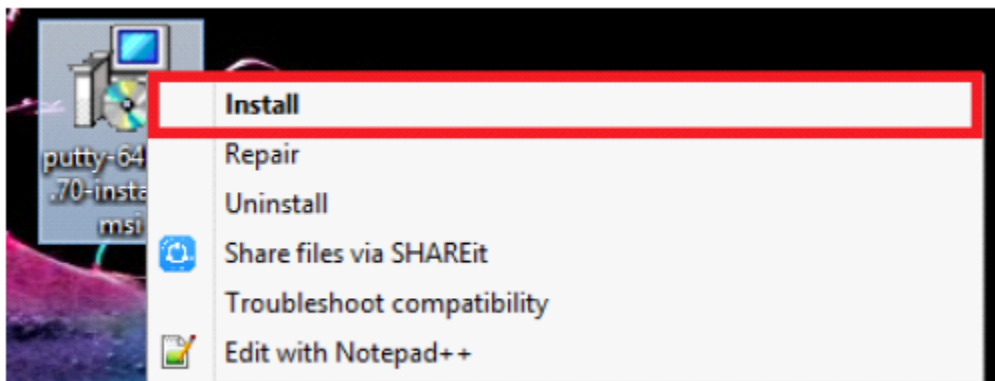
Instance ID	i-0940d0a5cbd37d860	Public DNS (IPv4)	ec2-18-206-186-27.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.206.186.27
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	-	Private DNS	ip-172-31-46-155.ec2.internal
Availability zone	us-east-1b	Private IPs	172.31.46.155
Security groups	MyTomcatAnJenkins_SG. view inbound rules . view outbound rules	Secondary private IPs	-
Scheduled events	No scheduled events	VPC ID	vpc-c1e698b0
AMI ID	ami-2020-10-11-1114-x06_64-gp2 (ami-009d002940d06e52)	Subnet ID	subnet-6208463f
Platform	-	Network interfaces	eth0

Step #2 | Installing Putty

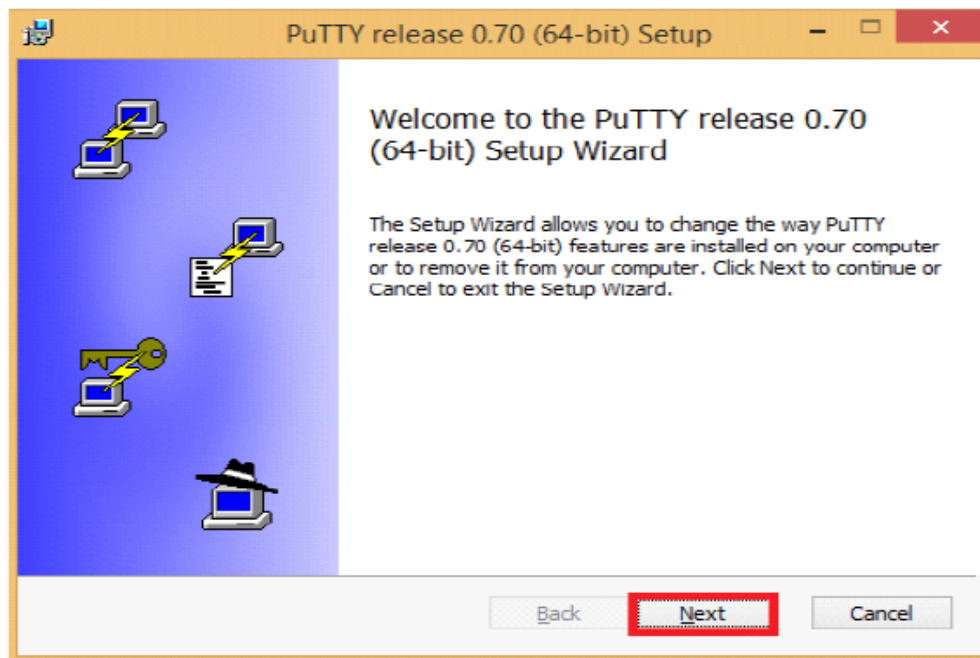
Goto <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> and download **putty msi installer** by clicking on the download link.



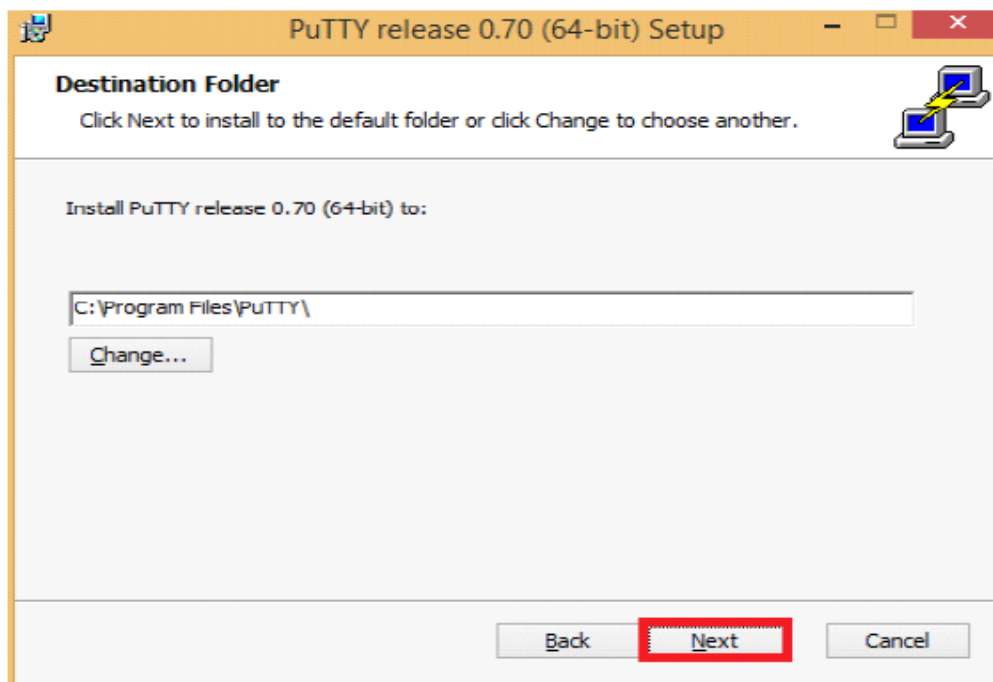
Right click on the Installer and select **Install** option.



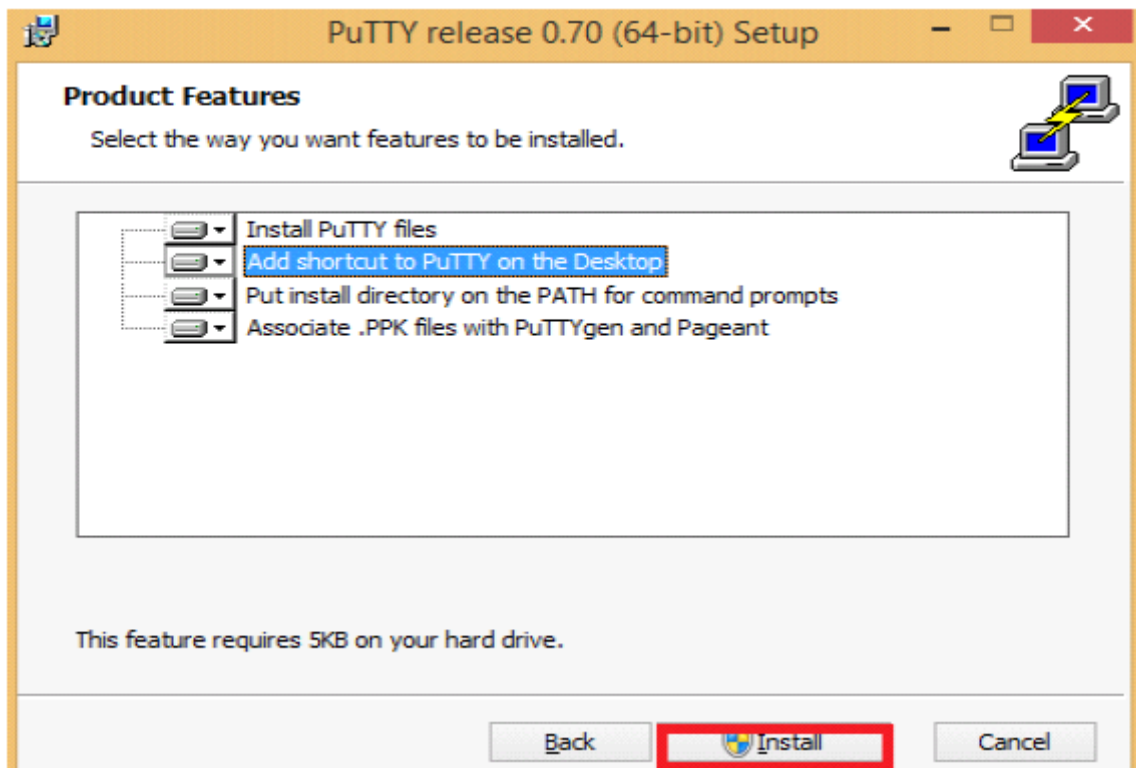
After selecting install option, the setup page will be displayed. Click on **Next**.



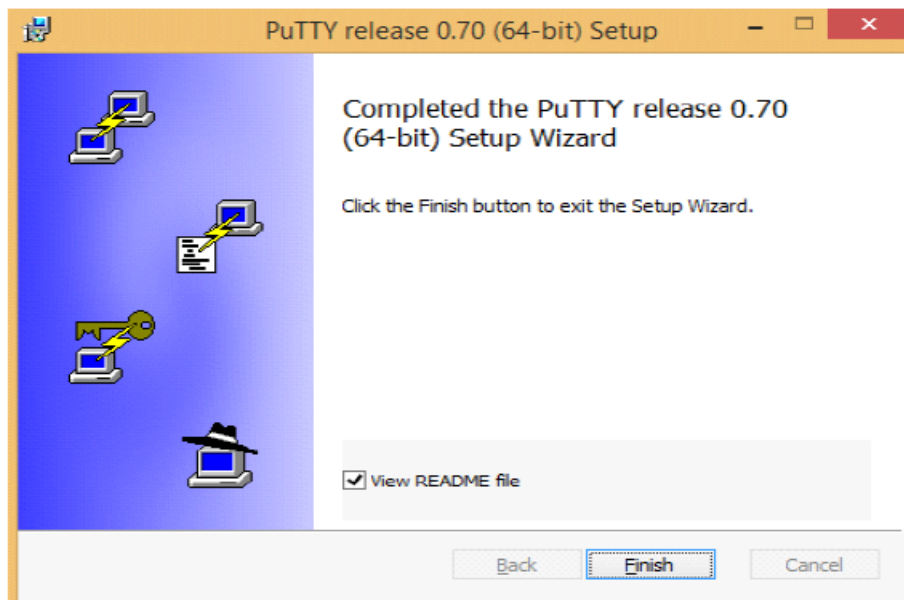
Select the location where putty has to be installed and click on **Next**.



Now select install option to install the application as shown below.



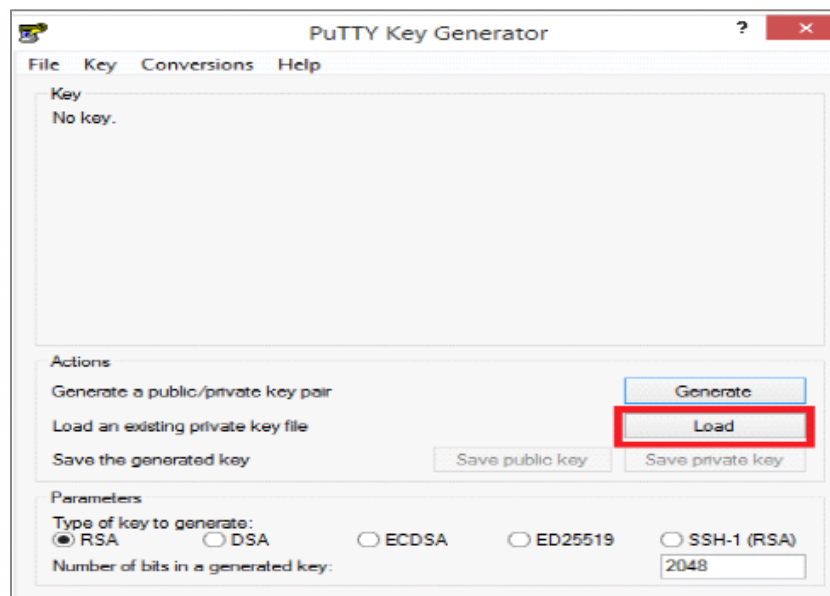
Select **Install** option it will prompt you to confirm installation with yes or no. Select **yes** to continue. Once the installation is completed, click on Finish.



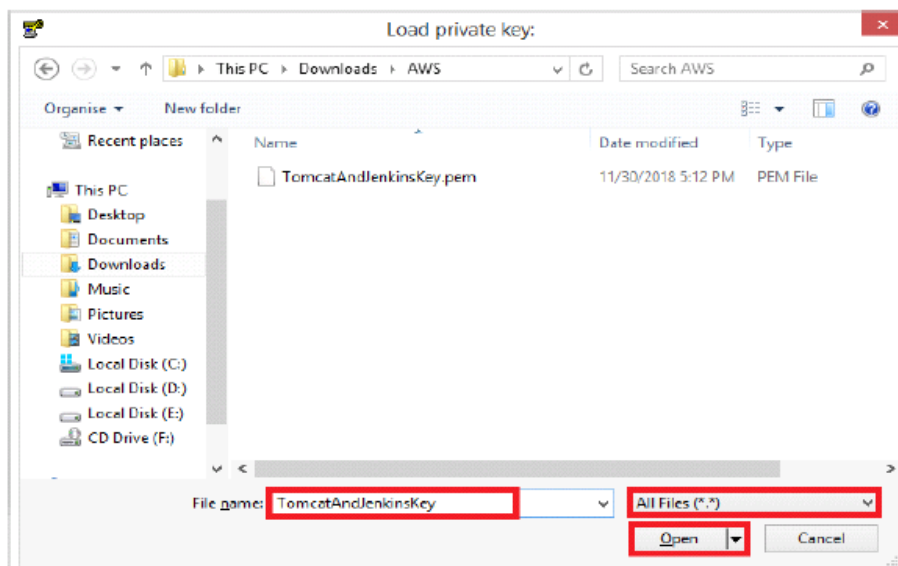
Step #3 | Conversion of .pem to .ppk

If you are using windows to connect to the instance, you need to install PuTTYgen and convert .pem file to .ppk file as shown below.

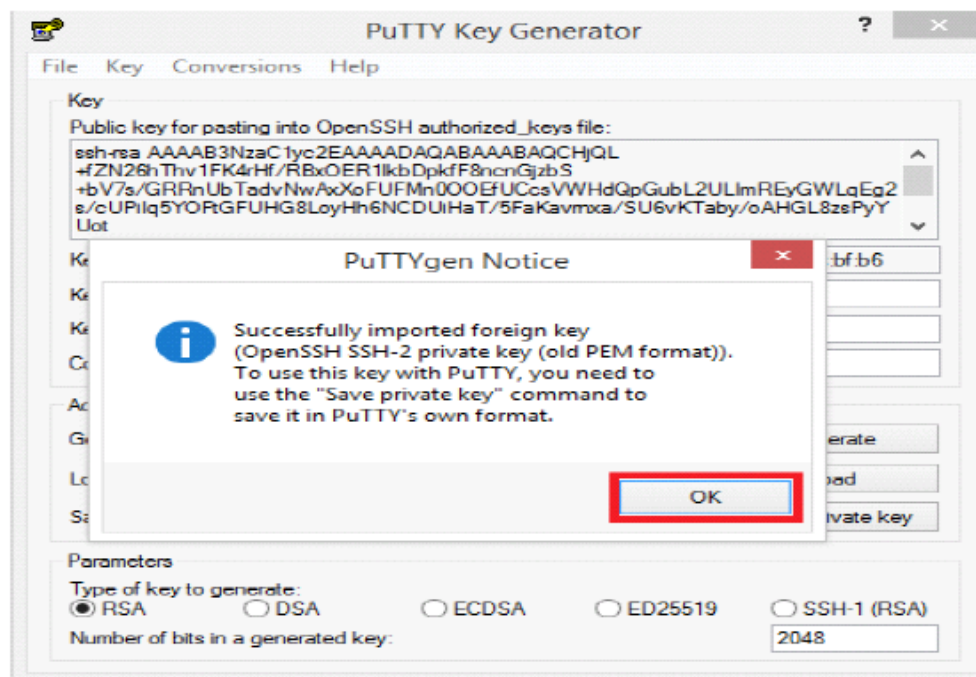
Open PuTTYgen and click on **Load**.



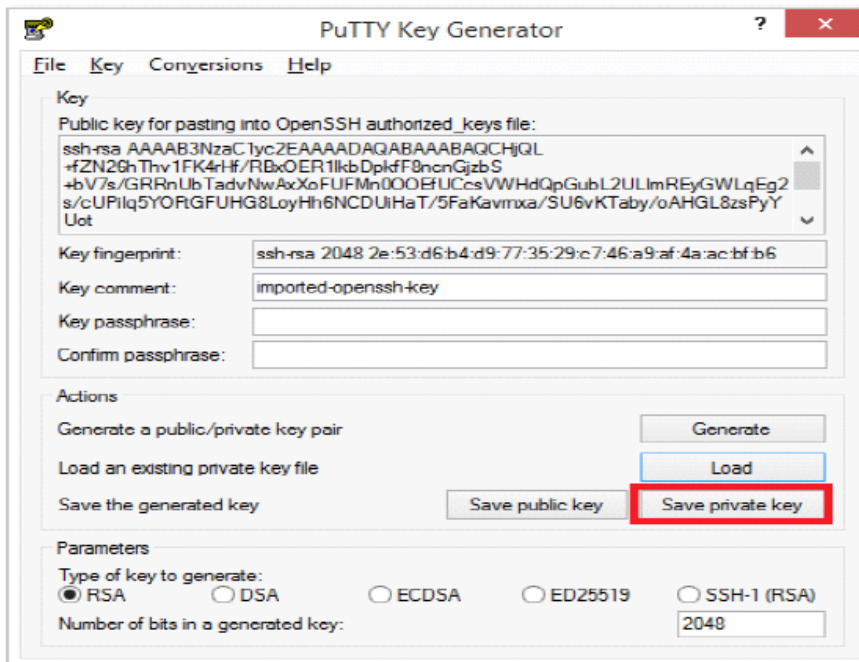
Search with your .pem file and change the file type to **All Files** as shown below and click on **Open**.



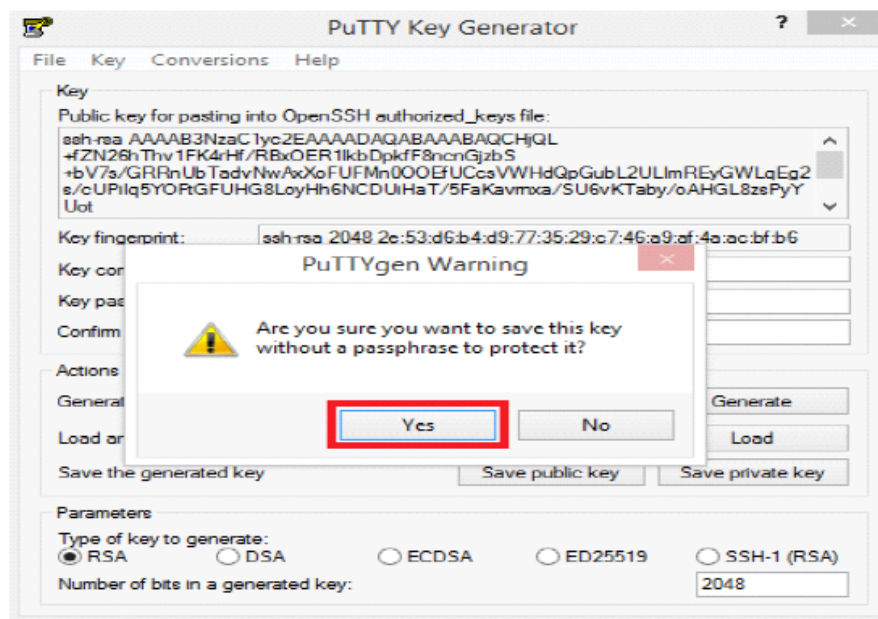
A popup is displayed on successful upload. Click on **OK** to continue.



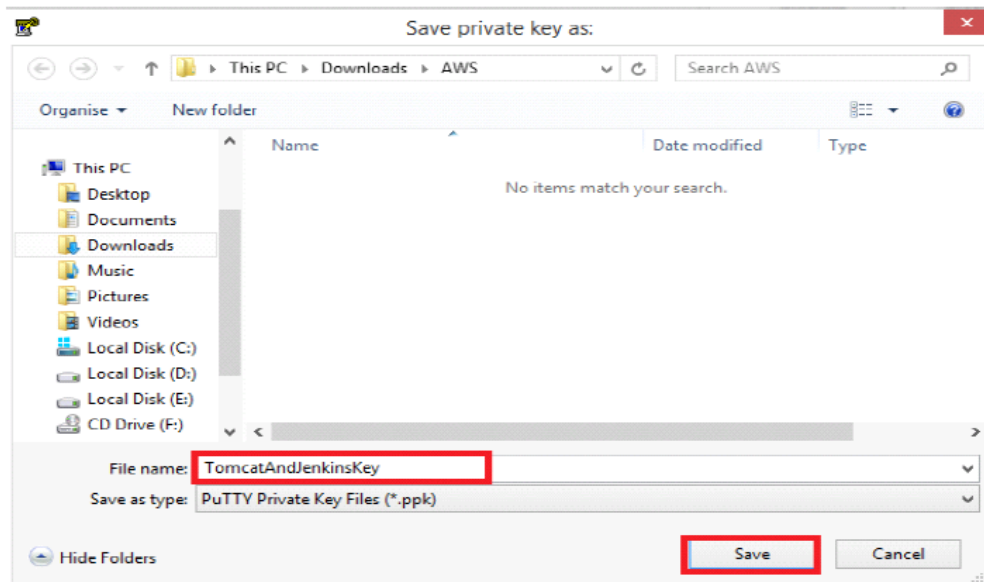
Click on Save private key to get .ppk file to your local machine.



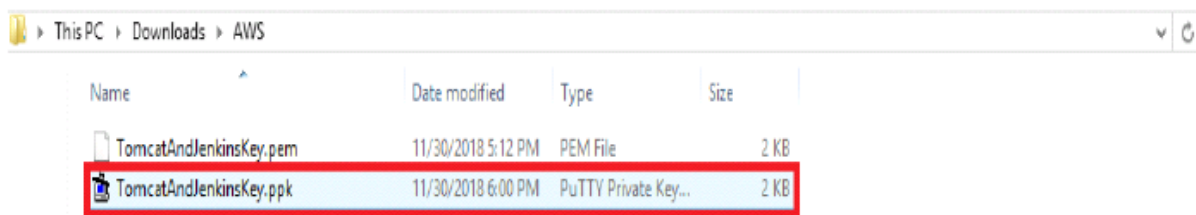
Confirmation popup is displayed before saving the .ppk file. Click on **Yes**.



Enter name for .ppk file and click on Save.



After downloading .ppk into your local machine, the file is as shown below.

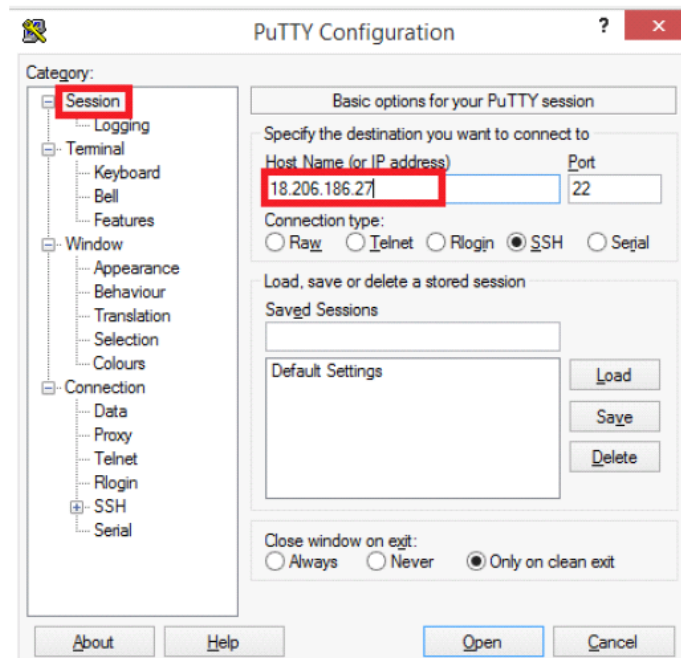


Step # 4| SSH to your Instance

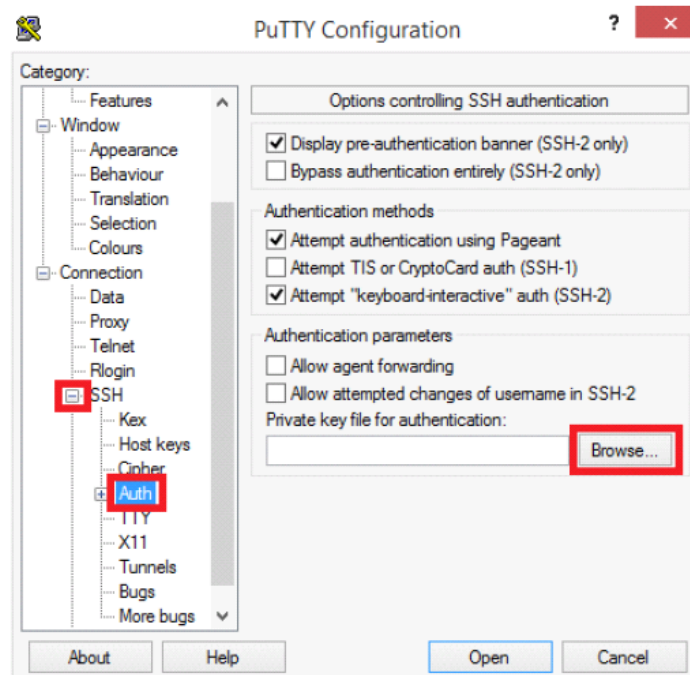
To connect to your instance, install PuTTY.

Open PuTTY and click on Session which is at the left side menu and enter Public IPV4 of the instance in Host Name (or IP address).

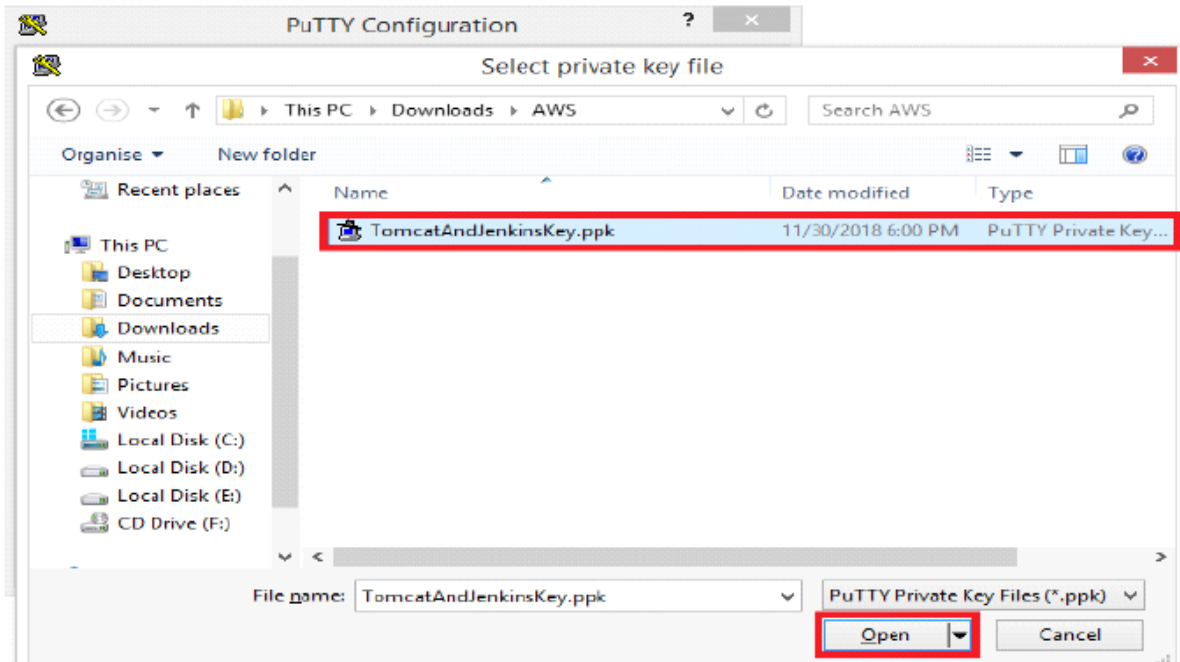
Note - Here Host Name means IPV4 Public IP which is discussed in #AWS EC2 instance creation.



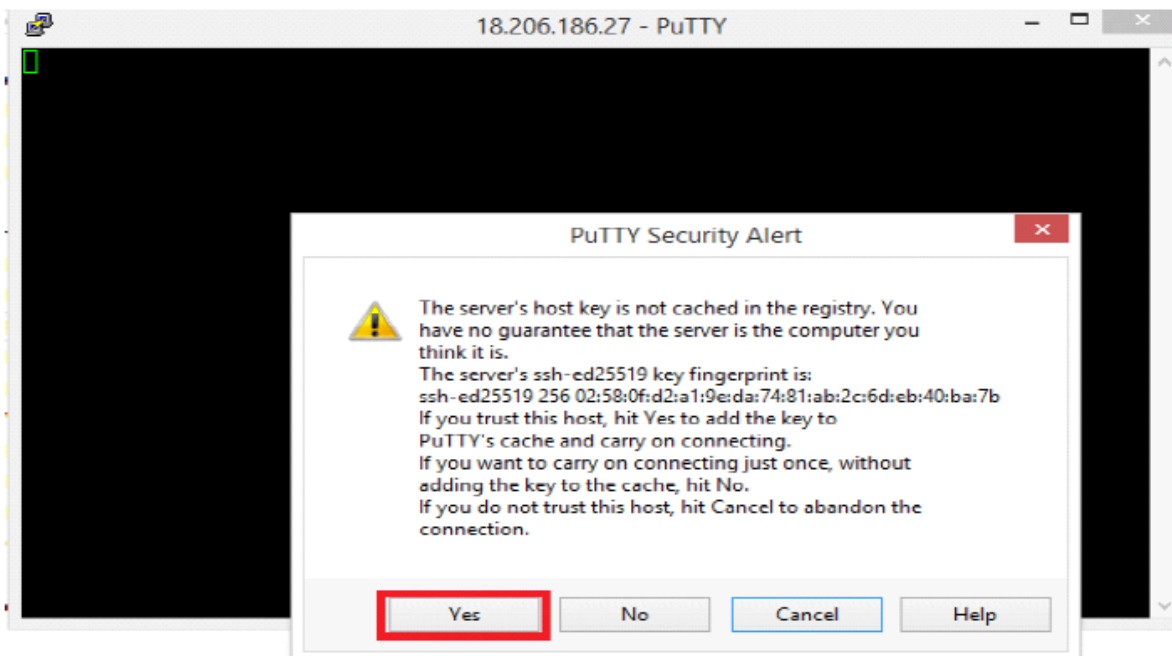
Click on **+SSH** in the left menu. Thereafter, click on **Auth**. You can select your .ppk file by clicking on **Browse**.



Select your .ppk file and click on **Open**.



A security alert will be displayed, click on **Yes**.

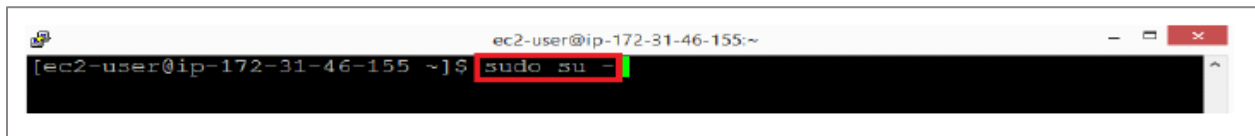


Enter **ec2-user** as user name as it is a Linux instance.



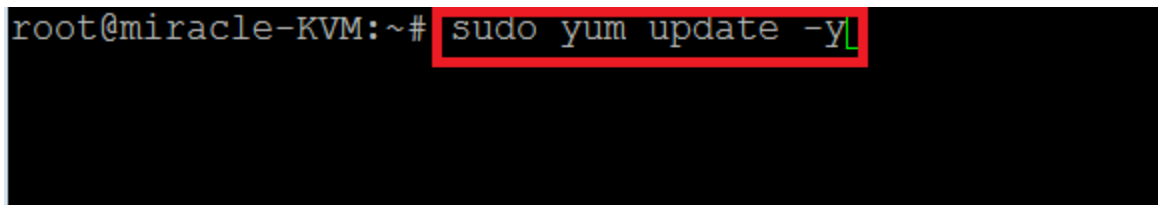
```
login as: ec2-user
```

To switch to root user enter **sudo su -**



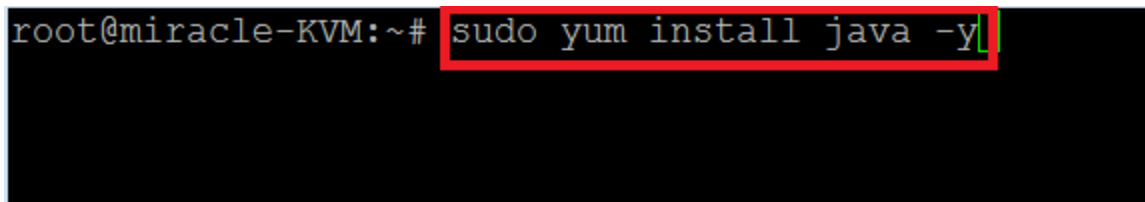
```
ec2-user@ip-172-31-46-155:~$ sudo su -
```

To check for updates, enter **sudo yum update -y**



```
root@miracle-KVM:~# sudo yum update -y
```

Install Java by using this command: **sudo yum install java -y**



```
root@miracle-KVM:~# sudo yum install java -y
```

Check the version of Java by giving this command **java -version** as shown below

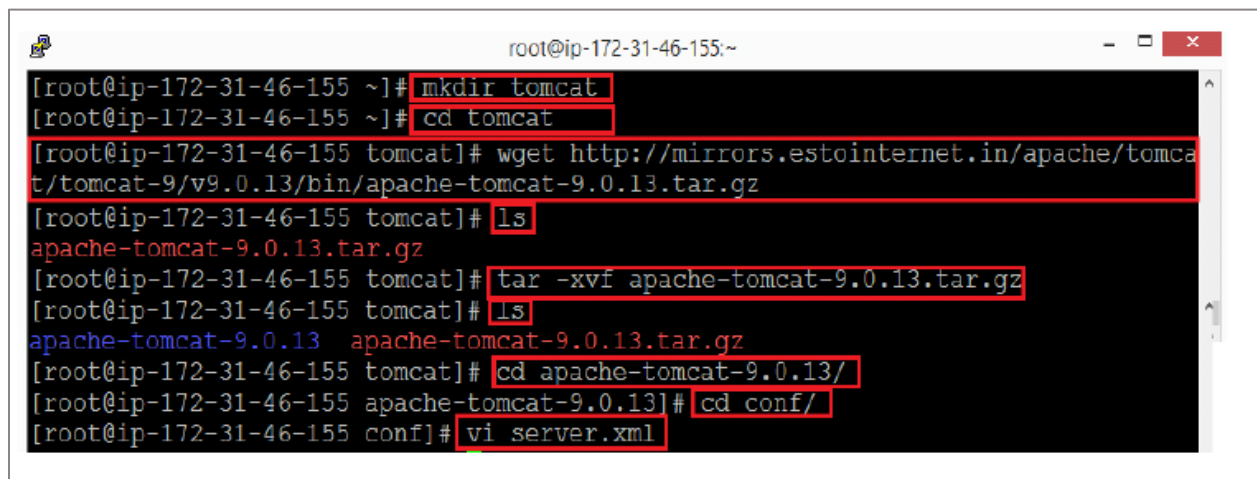


```
ec2-user@ip-172-31-46-155:~$ java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-b12)
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@ip-172-31-46-155 ~]#
```


Step #5 | Tomcat Installation

To place Tomcat on your instance execute the following commands,

- Create folder : **mkdir folderName**(give any folderName name you wish to have)
Example: **mkdir tomcat**
- Set the path for your folder : **cd folderName**
- Download the Tomcat file from official tomcat website,
Wget <http://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.13/bin/apache-tomcat-9.0.13.tar.gz>
- Check whether tomcat tar file in that path by giving this command : **ls**
- Unzip the tar file : **tar -xvf apache-tomcat-9.0.13.tar.gz**
- Check whether unzipped tomcat file got placed in that path by giving this command : **ls**
- Goto that unzipped folder : **cd apache-tomcat-9.0.13**
- Goto conf folder : **cd conf**
- Open server.xml to change the port number of tomcat: **vi server.xml**

A terminal window screenshot showing the installation of Tomcat. The window title is 'root@ip-172-31-46-155:~'. The commands and their outputs are as follows:

```
[root@ip-172-31-46-155 ~]# mkdir tomcat
[root@ip-172-31-46-155 ~]# cd tomcat
[root@ip-172-31-46-155 tomcat]# wget http://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.13/bin/apache-tomcat-9.0.13.tar.gz
[root@ip-172-31-46-155 tomcat]# ls
apache-tomcat-9.0.13.tar.gz
[root@ip-172-31-46-155 tomcat]# tar -xvf apache-tomcat-9.0.13.tar.gz
[root@ip-172-31-46-155 tomcat]# ls
apache-tomcat-9.0.13  apache-tomcat-9.0.13.tar.gz
[root@ip-172-31-46-155 tomcat]# cd apache-tomcat-9.0.13/
[root@ip-172-31-46-155 apache-tomcat-9.0.13]# cd conf/
[root@ip-172-31-46-155 conf]# vi server.xml
```

After opening **server.xml**, click on **insert** option on your keyboard and search for **<connector port>** by pressing the down arrow. Change the port to **80** and save the file by clicking the **esc** key on your keyboard and enter : **wq**

Note - Here, you can give any port number for Tomcat server. To access easily we gave it as 80


```
root@ip-172-31-46-155:~/tomcat/apache-tomcat-9.0.13/conf
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
-->
-- INSERT --
```

Open tomcat-users.xml : **vi tomcat-users.xml**

```
root@ip-172-31-46-155:~/tomcat/apache-tomcat-9.0.13/conf
[root@ip-172-31-46-155 conf]# vi tomcat-users.xml
```

Make sure that the below lines are added before **</tomcat-users>**

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="tomcat" roles="manager-gui,manager-
script"/>
```

Save the file by hitting ESC key and enter : **wq**

```
root@ip-172-31-46-155:~/tomcat/apache-tomcat-9.0.13/conf
<!--
NOTE: The sample user and role entries below are intended for use with the
examples web application. They are wrapped in a comment and thus are ignored
when reading this file. If you wish to configure these users for use with the
examples web application, do not forget to remove the <!-- .. --> that surrounds
them. You will also need to set the passwords to something appropriate.
-->

<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="tomcat" password="tomcat" roles="manager-gui,manager-script"/>

</tomcat-users>
-- INSERT --
```

Restart the Tomcat Server by giving restart command or by shut down and start again. To do that execute the below commands,

- Go back from the current folder : **cd ..**
- Go to bin folder : **cd bin**
- Shutdown Tomcat server : **sh shutdown.sh**
- Start Tomcat server : **sh startup.sh**

```
root@ip-172-31-46-155:~/tomcat/apache-tomcat-9.0.13/bin
[root@ip-172-31-46-155 conf]# cd ..
[root@ip-172-31-46-155 apache-tomcat-9.0.13]# cd bin/
[root@ip-172-31-46-155 bin]# sh shutdown.sh
Using CATALINA_BASE:   /root/tomcat/apache-tomcat-9.0.13
Using CATALINA_HOME:   /root/tomcat/apache-tomcat-9.0.13
Using CATALINA_TMPDIR: /root/tomcat/apache-tomcat-9.0.13/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /root/tomcat/apache-tomcat-9.0.13/bin/bootstrap.jar:/root
/tomcat/apache-tomcat-9.0.13/bin/tomcat-juli.jar
[root@ip-172-31-46-155 bin]#
```

You can see **Tomcat started** message on the screen.

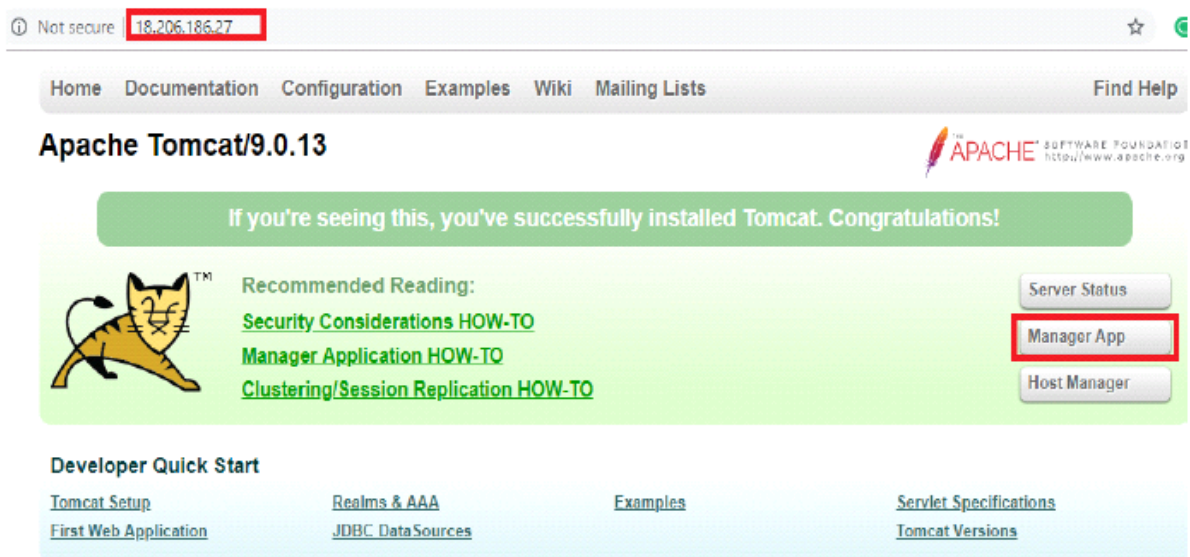
```
root@ip-172-31-46-155:~/tomcat/apache-tomcat-9.0.13/bin
[root@ip-172-31-46-155 bin]# sh startup.sh
Using CATALINA_BASE:   /root/tomcat/apache-tomcat-9.0.13
Using CATALINA_HOME:   /root/tomcat/apache-tomcat-9.0.13
Using CATALINA_TMPDIR: /root/tomcat/apache-tomcat-9.0.13/temp
Using JRE_HOME:        /usr
Using CLASSPATH:        /root/tomcat/apache-tomcat-9.0.13/bin/bootstrap.jar:/root
/tomcat/apache-tomcat-9.0.13/bin/tomcat-juli.jar
Tomcat started.
[root@ip-172-31-46-155 bin]#
```

To access your application through Tomcat do the following process.

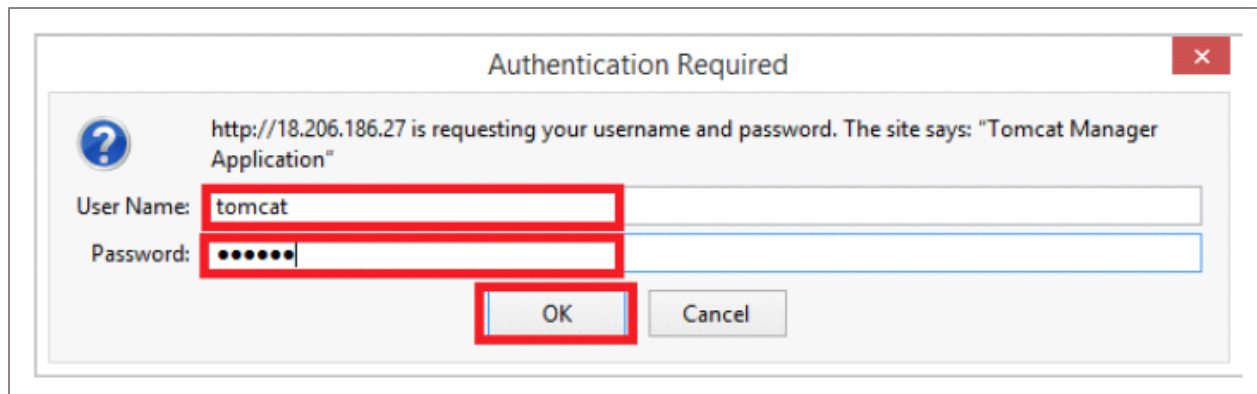
Step #6 | Login to Tomcat Manager Console

Enter your instance public IPV4 address in browser and hit enter. For example: 18.206.186.27

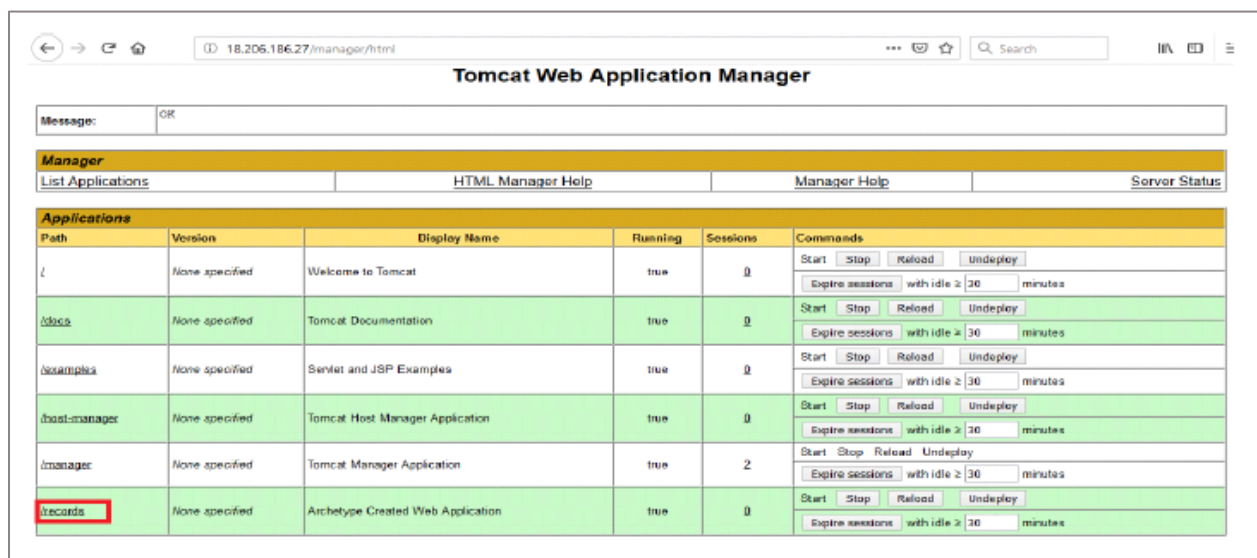
Click on **Manager App** and it asks for Username and Password.



Enter User Name and Password and click on **OK**.



After opening **ManagerApp** page, you will get the list of all deployed contexts as shown below.



If you get the 403 access denied error after opening the Tomcat Manager App do the following steps,

- Give the command to go the previous directory **cd ..**
- Goto the directory by using the command **cd webapps/manager/META-INF**
- List the files in that directory using **ls**
- Open the file **context.xml** by using this command **vi context.xml**
- Click the **insert** option on your keyboard
- Change the allow value as **\d+\. \d+\. \d+\. \d+** within the **double quotes ("")**. Save the file by selecting the **esc** key on your keyboard and enter : **wq**

- Come back to your Apache folder by giving the command **cd ..** (until you get the Apache folder)
- Open bin folder by using **cd bin**
- Once shut down your Tomcat Server with **sh shutdown.sh**
- Start Tomcat by using command **sh startup.sh**
- After completion of above procedure you have to open the manager app and you can deploy the application

For any questions regarding the lab please feel free to reach out to innovation@miraclesoft.com. We hope you enjoyed this!