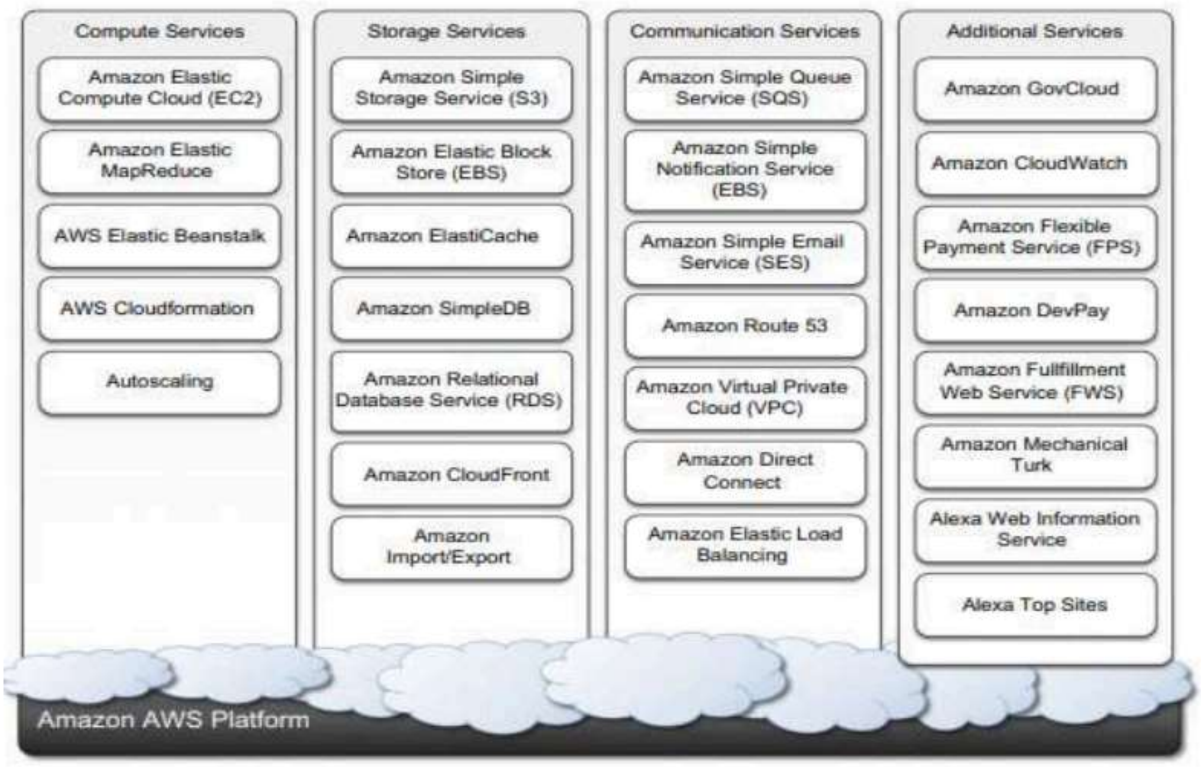
EX.NO 4: Deploy Java Web Application using Amazon-EC2

Date:

READING MATERIALS:

Amazon Web Services (AWS) is a platform that allows the development of flexible applications by providing solutions for elastic infrastructure scalability, messaging, and data storage. The platform is accessible through SOAP or RESTful Web service interfaces and provides a Web-based console where users can handle administration and monitoring of the resources required, as well as their expenses computed on a pay-as-you-go basis.

Amazon Webservice Ecosystem


Compute Services
Amazon Elastic Compute Cloud (EC2)
Amazon Elastic MapReduce
AWS Elastic Beanstalk
AWS Cloudformation
Autoscaling
Storage Services
Amazon Simple Storage Service (S3)
Amazon Elastic Block Store (EBS)
Amazon ElastiCache
Amazon SimpleDB
Amazon Relational Database Service (RDS)
Amazon CloudFront
Amazon Import/Export
Communication Services
Amazon Simple Queue Service (SQS)
Amazon Simple Notification Service (EBS)
Amazon Simple Email Service (SES)
Amazon Route 53
Amazon Virtual Private Cloud (VPC)
Amazon Direct Connect
Amazon Elastic Load Balancing
Additional Services
Amazon GovCloud
Amazon CloudWatch
Amazon Flexible Payment Service (FPS)
Amazon DevPay
Amazon Fullfillment Web Service (FWS)
Amazon Mechanical Turk
Alexa Web Information Service
Alexa Top Sites
Amazon AWS Platform

Above figure shows all the services available in the AWS ecosystem. At the base of the solution stack are services that provide raw compute and raw storage: Amazon ElastiCompute (EC2) and Amazon Simple Storage Service (S3). These are the two most popuservices, which are generally complemented with other offerings for building a complete system. At the higher level, Elastic MapReduce and AutoScaling provide additional capabilities for building smarter and more elastic computing systems. On the data side, Elastic Block Store (EBS), Amazon SimpleDB, Amazon RDS,and Amazon ElastiCache provide solutions for reliable data snapshots and the management of structured and semistructured data. Communication needs are covered at the networking level by AmazoVirtual Private Cloud (VPC), Elastic Load Balancing, Amazon Route 53, and Amazon Direct Connect. More advanced services for connecting applications are Amazon Simple Queue Service (SQS), Amazon Simple Notification Service (SNS), and Amazon Simple Email Service Compute services Compute services constitute the fundamental element of cloud computing systems. The fundamental service in this space is Amazon EC2, which delivers an IaaS solution that has served as a reference model for several offerings from other vendors in the same market segment. Amazon EC2 allows deploying servers in the form of virtual machines created as instances of a specific image. Images come with a preinstalled operating system and a software stack, and instances can be configured for memory, number of processors, and storage. Users are provided with credentials to remotely access the instance and further configure or install software if needed.

Amazon machine images

Amazon Machine Images (AMIs) are templates from which it is possible to create a virtual machine. They are stored in Amazon S3 and identified by a unique identifier in the form of ami-xxxxxx and a manifest XML file. An AMI contains a physical file system layout with a predefined operating system installed. These are specified by the Amazon Ramdisk Image (ARI, id: ari-yyyyyy) and the Amazon Kernel Image (AKI, id: aki-zzzzzz), which are part of the configuration of the template. AMIs are either created from scratch or "bundled" from existing EC2 instances.

EC2 instances

EC2 instances represent virtual machines. They are created using AMI as templates, which are specialized by selecting the number of cores, their computing power, and the installed memory. The processing power is expressed in terms of virtual cores and EC2 Compute Units (ECUs). The ECU is a measure of the computing power of a virtual core; it is used to express a predictable quantity of real CPU power that is allocated to an instance. By using compute units instead of real frequency values, Amazon can change over time the mapping of such units to the underlying real amount of computing power allocated, thus keeping the performance of EC2 instances consistent with standards set by the times.

The six major categories are:

- Standard instances: This class offers a set of configurations that are suitable for most applications.
- Micro instances: This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges in the workload.
- High-memory instances: This class targets applications that need to process huge workloads and require large amounts of memory.
- High-CPU instances: This class targets compute-intensive applications.
- Cluster Compute instances: This class is used to provide virtual cluster services.
- Cluster GPU instances: This class provides instances featuring graphic processing units (GPUs) and high compute power, large memory, and extremely high I/O and network performance. EC2 instances can be run either by using the command-line tools provided by Amazon, which connects the Amazon Web Service that provides remote access to the EC2 infrastructure, or via the AWS console, which allows the management of other services, such as S3. By default an EC2 instance is created with the kernel and the disk associated to the AMI. These define the architecture (32 bit or 64 bit) and the space of disk available to the instance. This is an ephemeral disk; once the instance is shut down, the content of the disk will be lost. Alternatively, it is possible to attach an EBS volume to the instance, the content of which will be stored in S3. If the default AKI and ARI are not suitable, EC2 provides capabilities to run EC2 instances by specifying a different AKI and ARI, thus giving flexibility in the creation of instances.

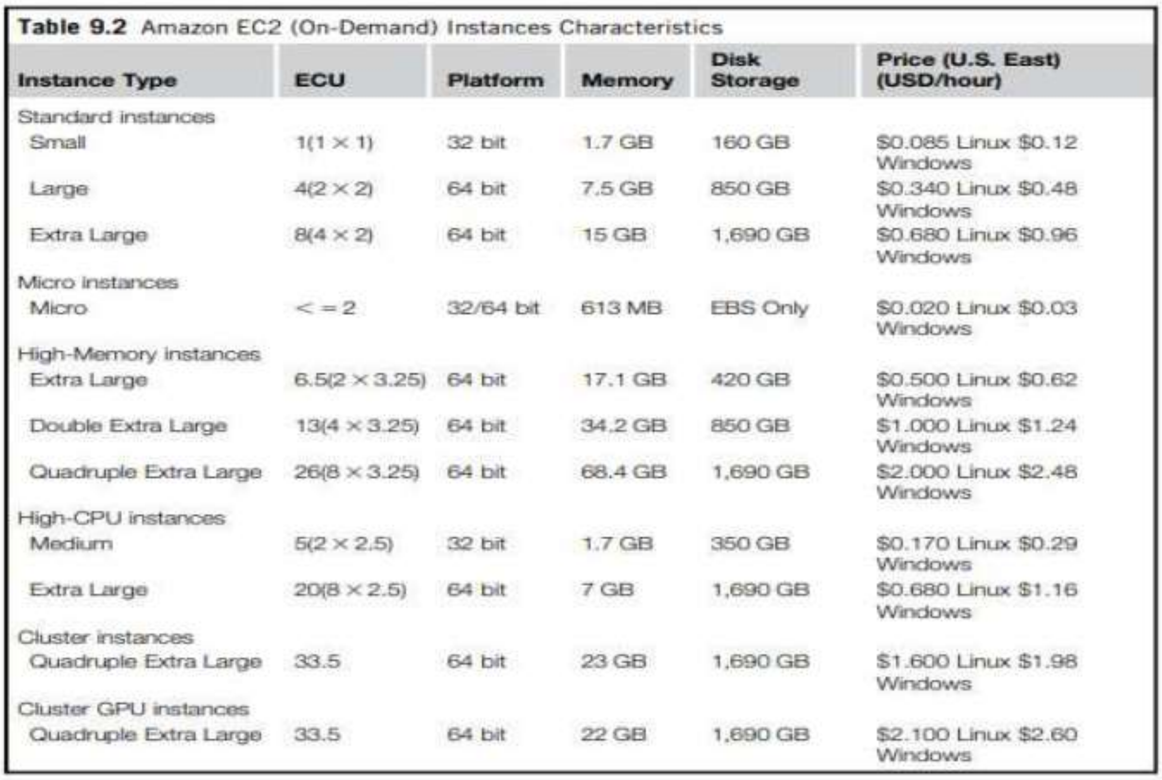


Table 9.2 Amazon EC2 (On-Demand) Instances Characteristics

Instance Type	ECU	Platform	Memory	Disk Storage	Price (U.S. East) (USD/hour)
Standard instances					
Small	1(1 × 1)	32 bit	1.7 GB	160 GB	$0.085 Linux $0.12 Windows
Large	4(2 × 2)	64 bit	7.5 GB	850 GB	$0.340 Linux $0.48 Windows
Extra Large	8(4 × 2)	64 bit	15 GB	1,690 GB	$0.680 Linux $0.96 Windows
Micro instances					
Micro	< = 2	32/64 bit	613 MB	EBS Only	$0.020 Linux $0.03 Windows
High-Memory instances					
Extra Large	6.5(2 × 3.25)	64 bit	17.1 GB	420 GB	$0.500 Linux $0.62 Windows
Double Extra Large	13(4 × 3.25)	64 bit	34.2 GB	850 GB	$1.000 Linux $1.24 Windows
Quadruple Extra Large	26(8 × 3.25)	64 bit	68.4 GB	1,690 GB	$2.000 Linux $2.48 Windows
High-CPU instances					
Medium	5(2 × 2.5)	32 bit	1.7 GB	350 GB	$0.170 Linux $0.29 Windows
Extra Large	20(8 × 2.5)	64 bit	7 GB	1,690 GB	$0.680 Linux $1.16 Windows
Cluster instances					
Quadruple Extra Large	33.5	64 bit	23 GB	1,690 GB	$1.600 Linux $1.98 Windows
Cluster GPU instances					
Quadruple Extra Large	33.5	64 bit	22 GB	1,690 GB	$2.100 Linux $2.60 Windows

EX.NO 4: Deploy Java Web Application using Amazon-EC2

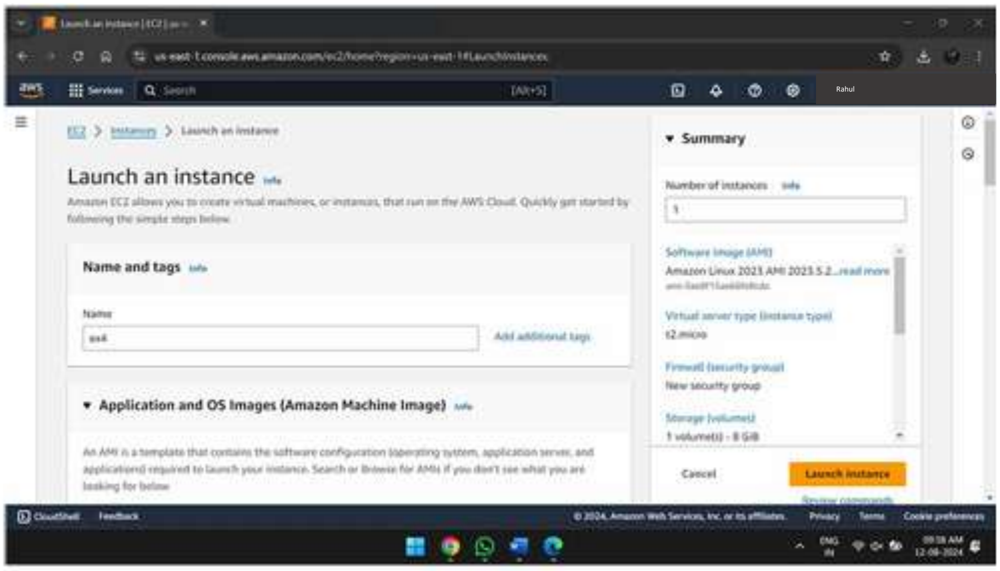
Date:

AIM:

To Deploy Java Web Application using Amazon-EC2.

PROCEDURE with SCREENSHOTS:

Step 1: Login into AWS and open EC2 service.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances

EC2 > Instances > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: ex4

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Summary

Number of instances: 1

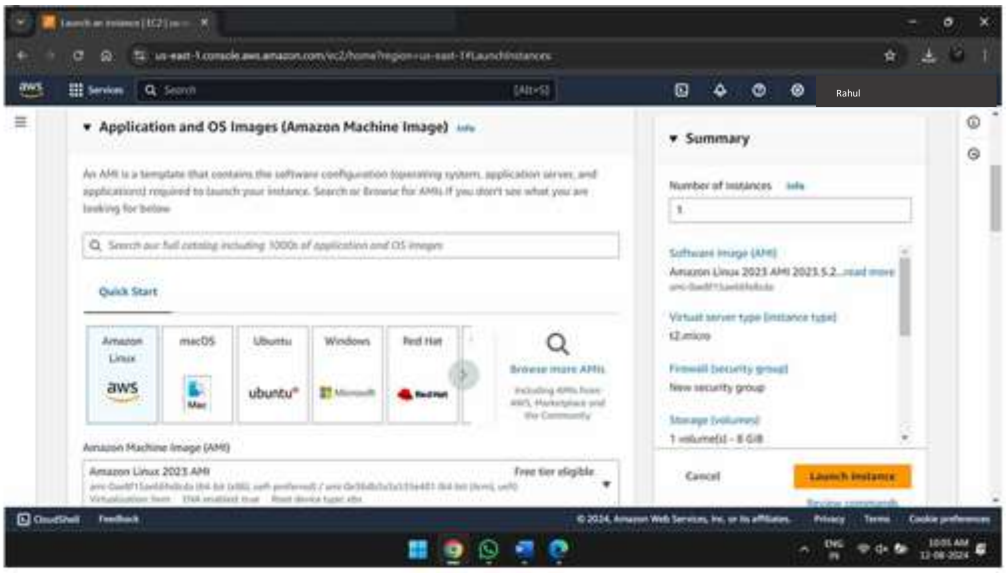
Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Step 2: Create an t2.micro Instance with Amazon Linux.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI): Amazon Linux 2023 AMI — Free tier eligible

Summary

Number of instances: 1

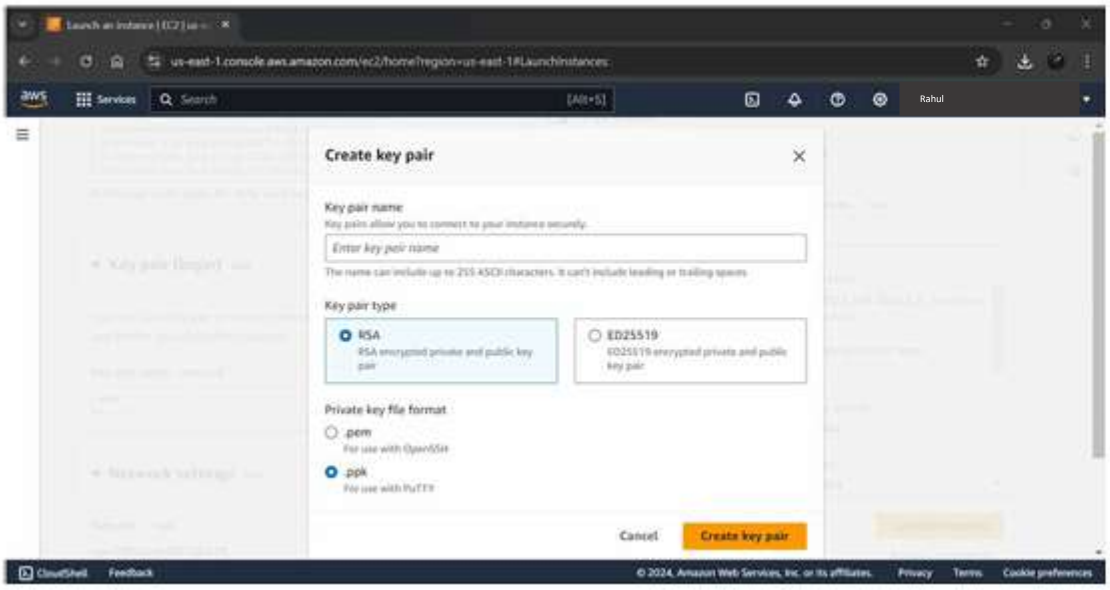
Software Image (AMI): Amazon Linux 2023 AMI 2023.5.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Step 3: Create key pair to be used with putty.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Create key pair

Key pair name
Key pairs allow you to connect to your instance securely.
Enter key pair name
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

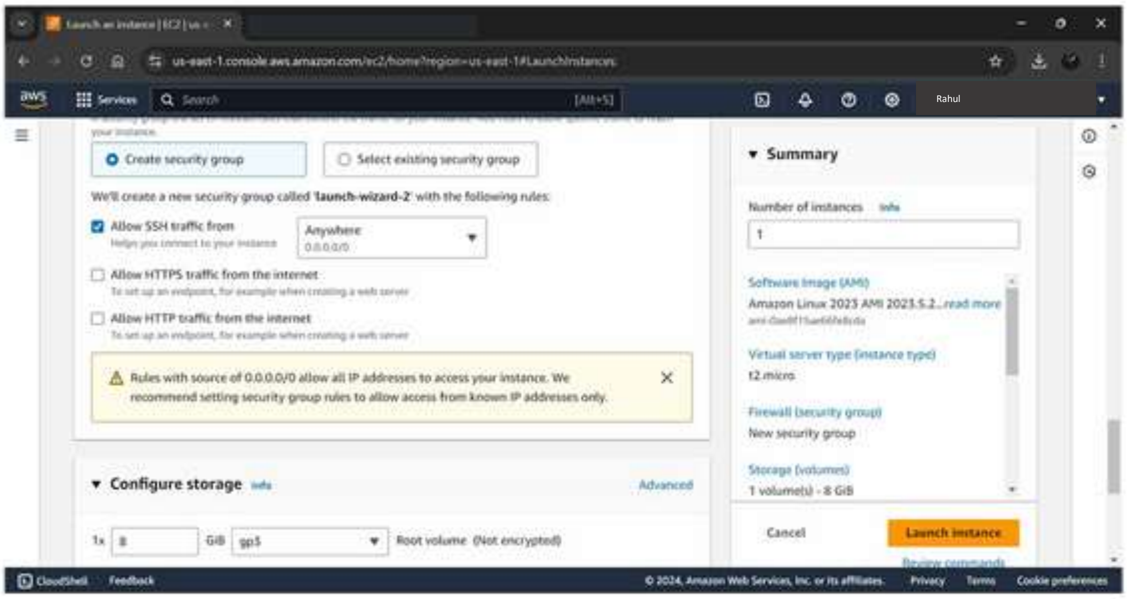
Key pair type
[✓ RSA] RSA encrypted private and public key pair
ED25519 ED25519 encrypted private and public key pair

Private key file format
.pem For use with OpenSSH
[✓ .ppk] For use with PuTTY

Cancel | Create key pair

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 4: Leave security group and storage settings as default.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

[✓ Create security group] Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

[✓] Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

[ ] Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

[ ] Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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.

Configure storage

1x 8 GiB gp3 Root volume (Not encrypted)

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...read more

Virtual server type (instance type)
t2.micro

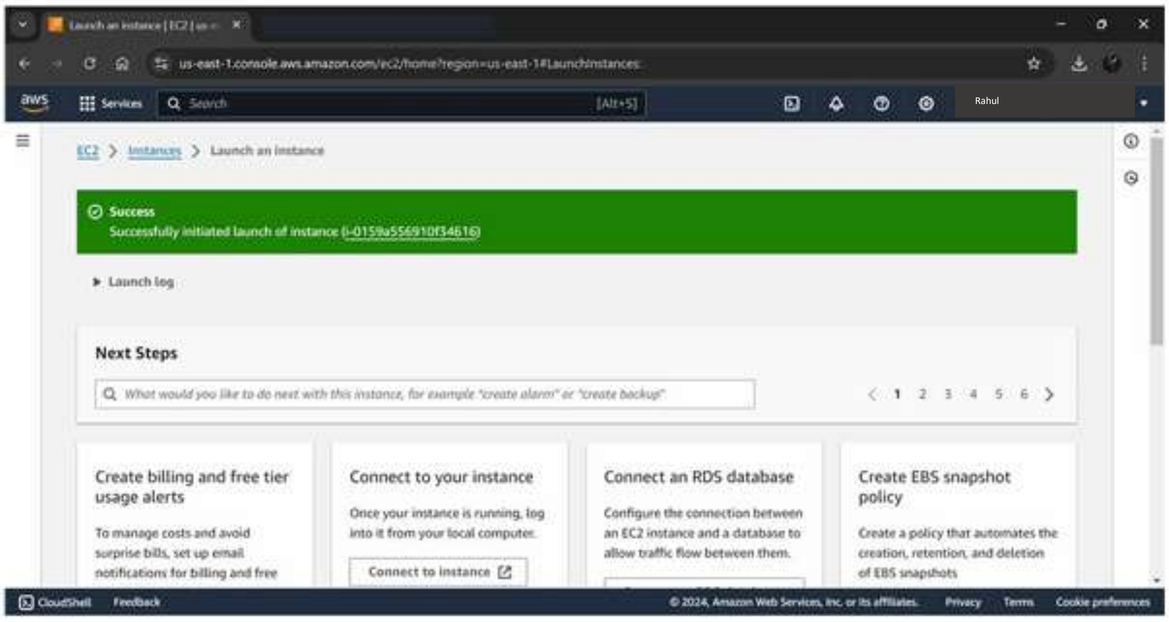
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel | Launch instance

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 5 : Click Launch Instance.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0159a556910f34616)

Launch log

Next Steps

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free

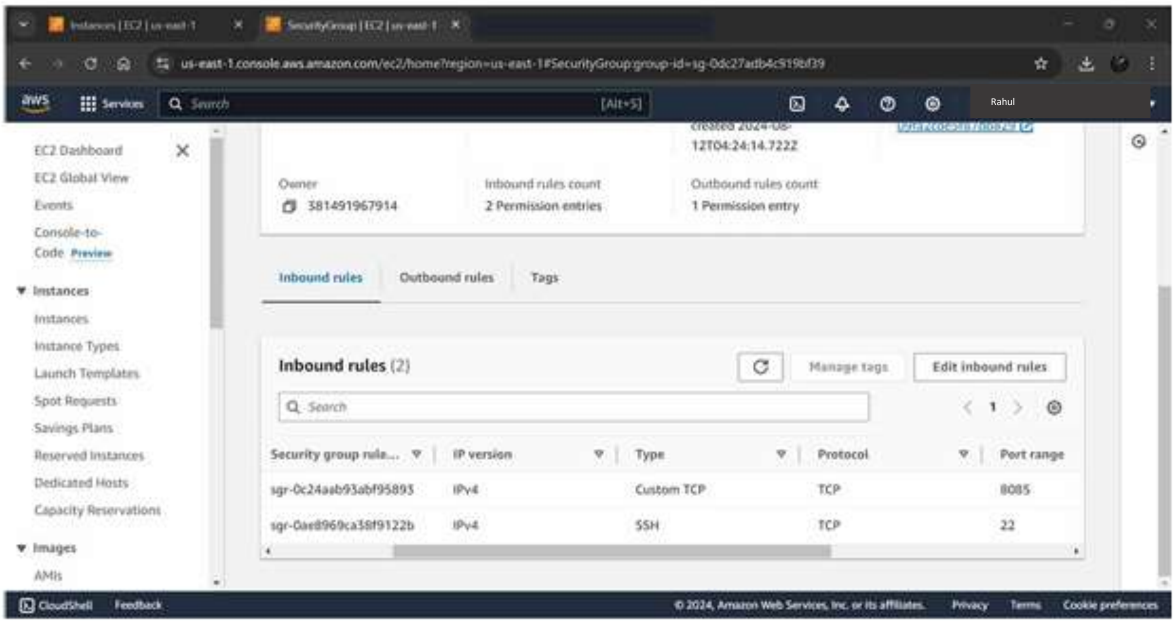
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 6 : Add Inbound rule of type Custom TCP with port range 8085.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:group-id=sg-0dc27adb4c919bf39

Created 2024-08-12T04:24:14.722Z

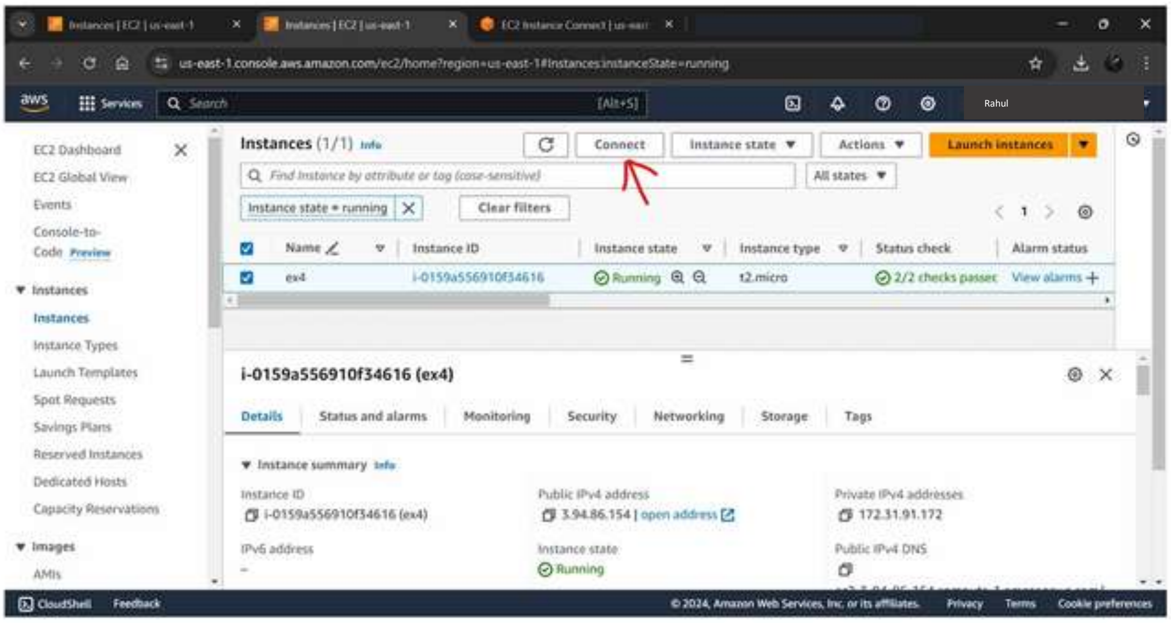
Owner	Inbound rules count	Outbound rules count
381491967914	2 Permission entries	1 Permission entry

Inbound rules (2)

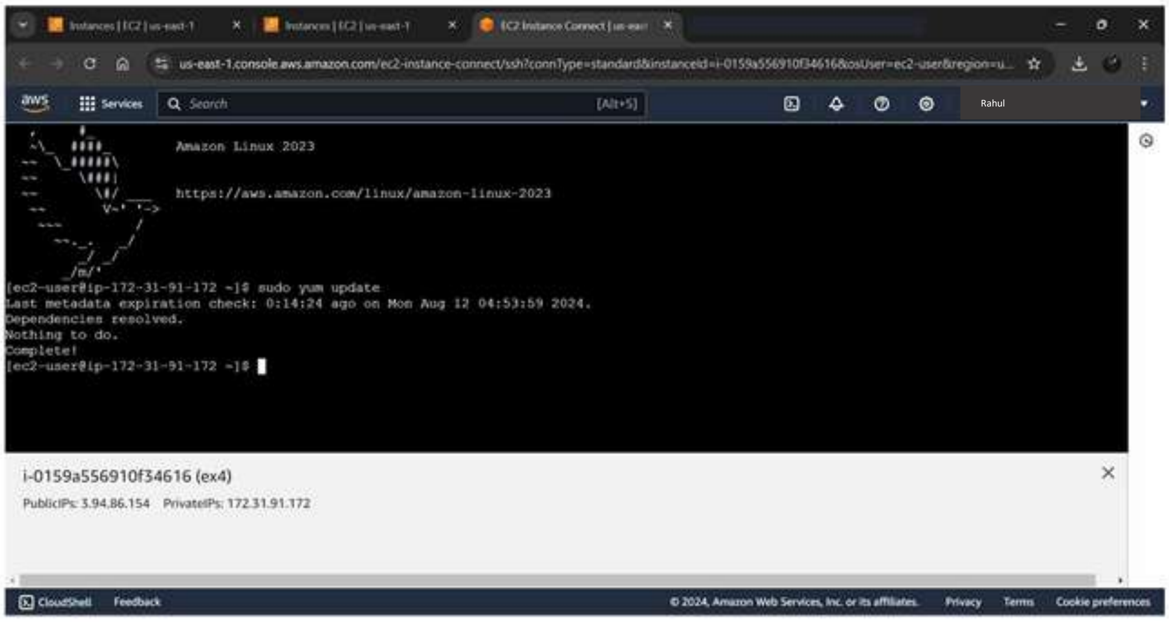
Security group rule...	IP version	Type	Protocol	Port range
sgr-0c24aab93abf95893	IPv4	Custom TCP	TCP	8085
sgr-0ae8969ca38f9122b	IPv4	SSH	TCP	22

© 2024, Amazon Web Services, Inc. or its affiliates.

Step 7 : Click connect to open console of the instance.



Step 8 : Update packet manager yum using sudo yum update.

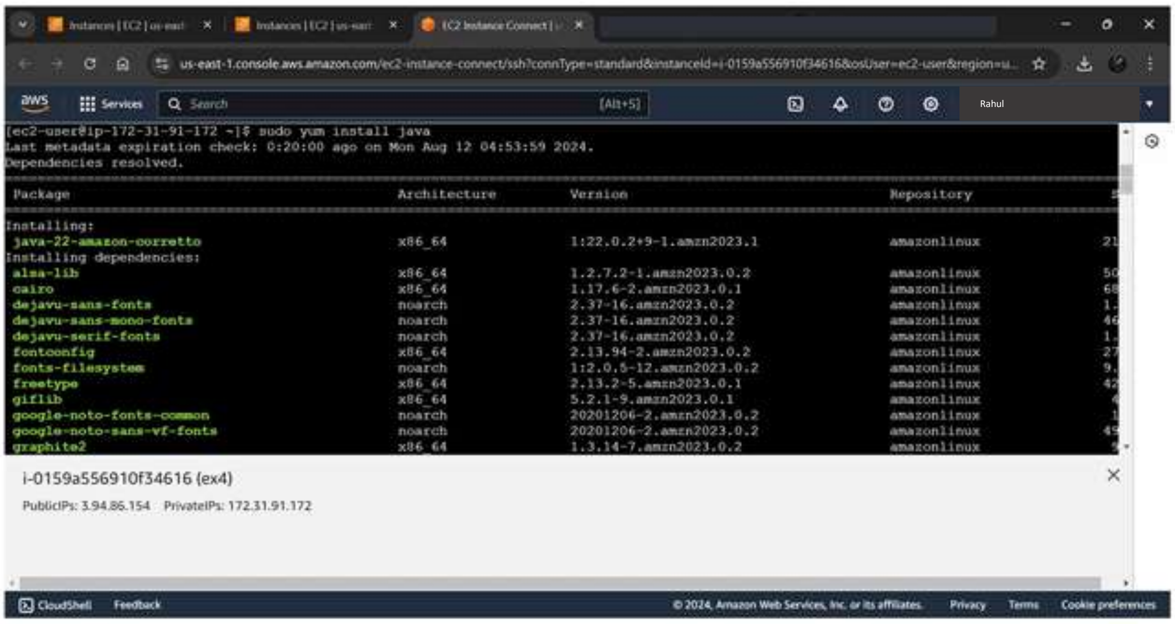


```
Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-91-172 ~]$ sudo yum update
Last metadata expiration check: 0:14:24 ago on Mon Aug 12 04:53:59 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-91-172 ~]$
```

i-0159a556910f34616 (ex4)

PublicIPs: 3.94.86.154 PrivateIPs: 172.31.91.172

Step 9 : Install Java using sudo yum install java.



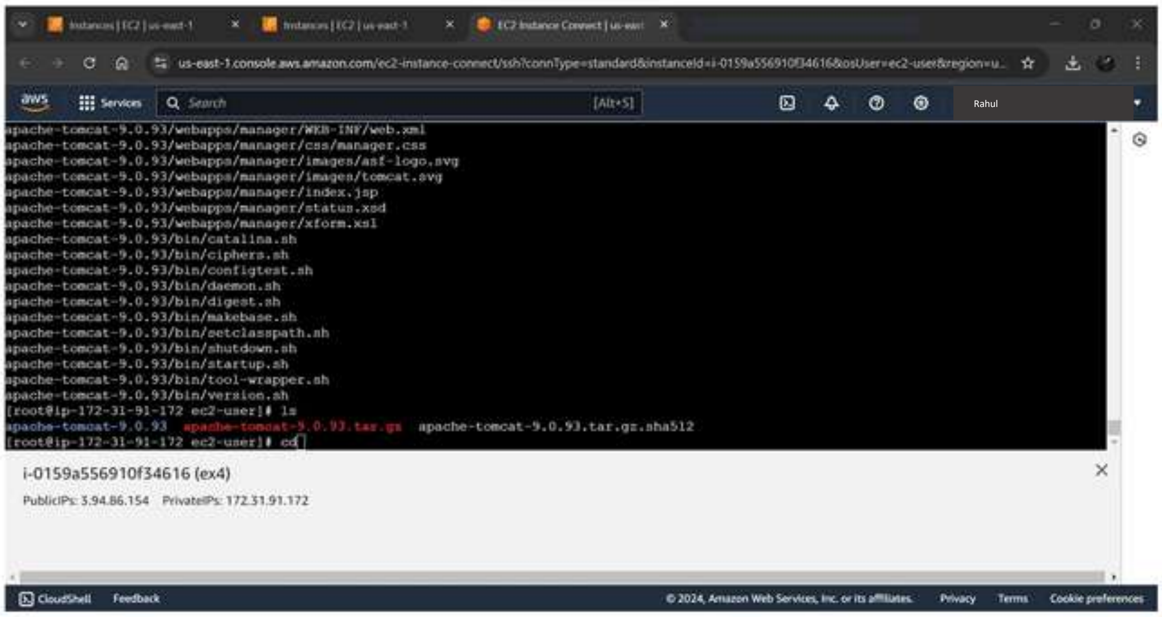
```
[ec2-user@ip-172-31-91-172 ~]$ sudo yum install java
Last metadata expiration check: 0:20:00 ago on Mon Aug 12 04:53:59 2024.
Dependencies resolved.
================================================================================
 Package                     Architecture   Version                      Repository
================================================================================
Installing:
 java-22-amazon-corretto     x86_64         1:22.0.2+9-1.amzn2023.1      amazonlinux
Installing dependencies:
 alsa-lib                    x86_64         1.2.7.2-1.amzn2023.0.2       amazonlinux
 cairo                       x86_64         1.17.6-2.amzn2023.0.1        amazonlinux
 dejavu-sans-fonts           noarch         2.37-16.amzn2023.0.2         amazonlinux
 dejavu-sans-mono-fonts      noarch         2.37-16.amzn2023.0.2         amazonlinux
 dejavu-serif-fonts          noarch         2.37-16.amzn2023.0.2         amazonlinux
 fontconfig                  x86_64         2.13.94-2.amzn2023.0.2       amazonlinux
 fonts-filesystem            noarch         1:2.0.5-12.amzn2023.0.2      amazonlinux
 freetype                    x86_64         2.13.2-5.amzn2023.0.1        amazonlinux
 giflib                      x86_64         5.2.1-9.amzn2023.0.1         amazonlinux
 google-noto-fonts-common    noarch         20201206-2.amzn2023.0.2      amazonlinux
 google-noto-sans-vf-fonts   noarch         20201206-2.amzn2023.0.2      amazonlinux
 graphite2                   x86_64         1.3.14-7.amzn2023.0.2        amazonlinux
```

i-0159a556910f34616 (ex4)

PublicIPs: 3.94.86.154 PrivateIPs: 172.31.91.172

Step 10: Install Tomcat Server using following commands. wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz

Step 11: Extract Tomcat File using tar xvfz apache-tomcat-9.0.93.tar.gz

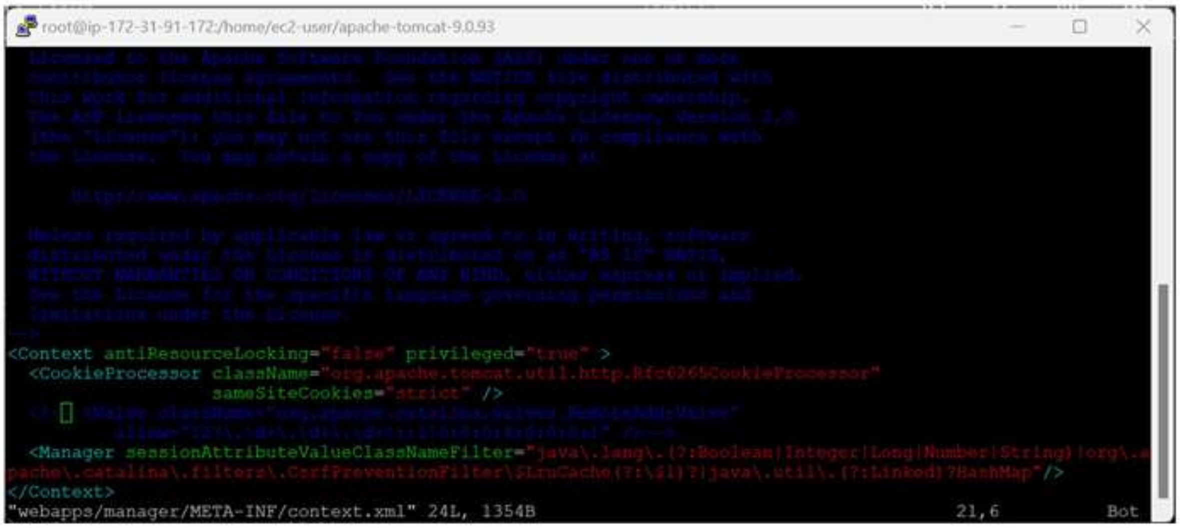


```
apache-tomcat-9.0.93/webapps/manager/WEB-INF/web.xml
apache-tomcat-9.0.93/webapps/manager/css/manager.css
apache-tomcat-9.0.93/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.93/webapps/manager/images/tomcat.svg
apache-tomcat-9.0.93/webapps/manager/index.jsp
apache-tomcat-9.0.93/webapps/manager/status.xsd
apache-tomcat-9.0.93/webapps/manager/xform.xsl
apache-tomcat-9.0.93/bin/catalina.sh
apache-tomcat-9.0.93/bin/ciphers.sh
apache-tomcat-9.0.93/bin/configtest.sh
apache-tomcat-9.0.93/bin/daemon.sh
apache-tomcat-9.0.93/bin/digest.sh
apache-tomcat-9.0.93/bin/makebase.sh
apache-tomcat-9.0.93/bin/setclasspath.sh
apache-tomcat-9.0.93/bin/shutdown.sh
apache-tomcat-9.0.93/bin/startup.sh
apache-tomcat-9.0.93/bin/tool-wrapper.sh
apache-tomcat-9.0.93/bin/version.sh
[root@ip-172-31-91-172 ec2-user]# ls
apache-tomcat-9.0.93  apache-tomcat-9.0.93.tar.gz  apache-tomcat-9.0.93.tar.gz.sha512
[root@ip-172-31-91-172 ec2-user]# cd
```

i-0159a556910f34616 (ex4)

PublicIPs: 3.94.86.154 PrivateIPs: 172.31.91.172

Step 12: Edit the context.xml file to comment default IP 127.0.0.1 Vi webapps/manager/META-INF/context.xml Comment the part .



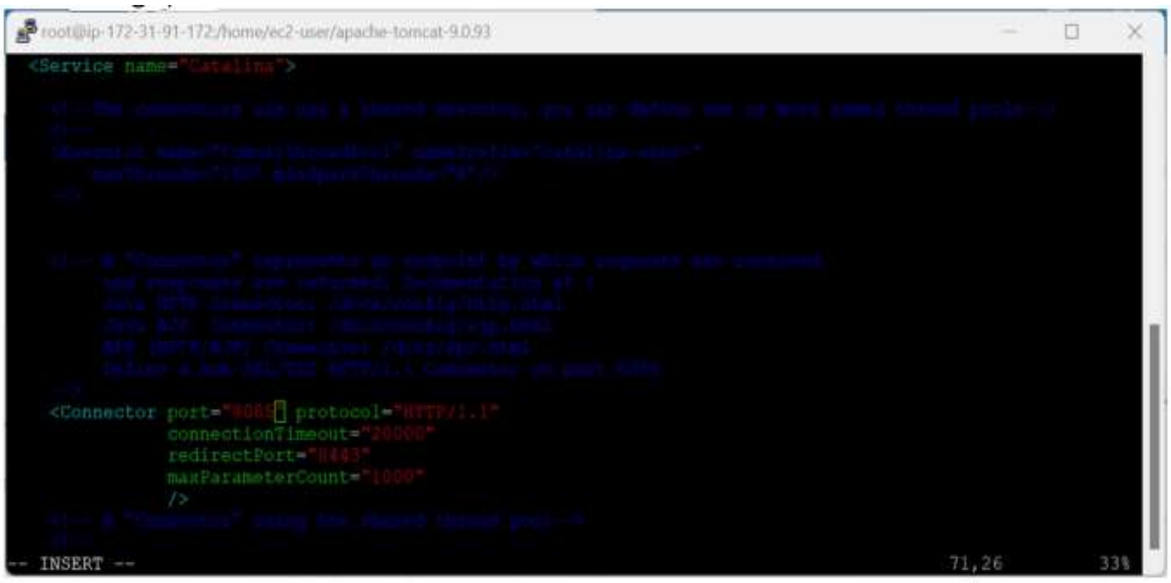
root@ip-172-31-91-172:/home/ec2-user/apache-tomcat-9.0.93

```
  licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<Context antiResourceLocking="false" privileged="true" >
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
  <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|String)|org\.a
pache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
</Context>
"webapps/manager/META-INF/context.xml" 24L, 1354B                          21,6          Bot
```

Step 13 : Give Credential to manager-gui of tomcat

Vi conf/tomcat-users.xml

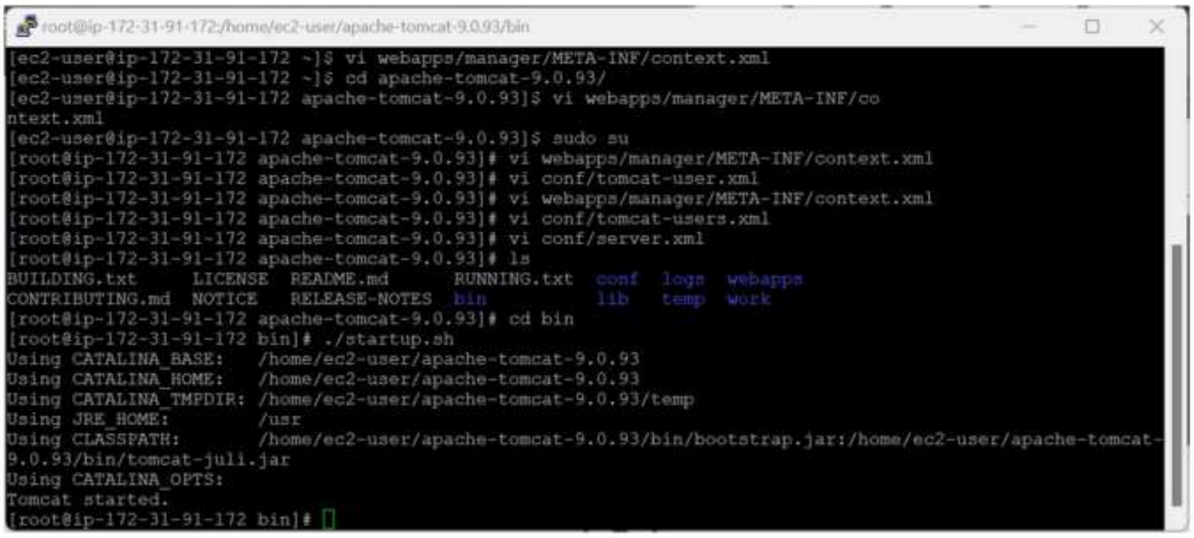
Change port 8080 to 8085



root@ip-172-31-91-172:/home/ec2-user/apache-tomcat-9.0.93

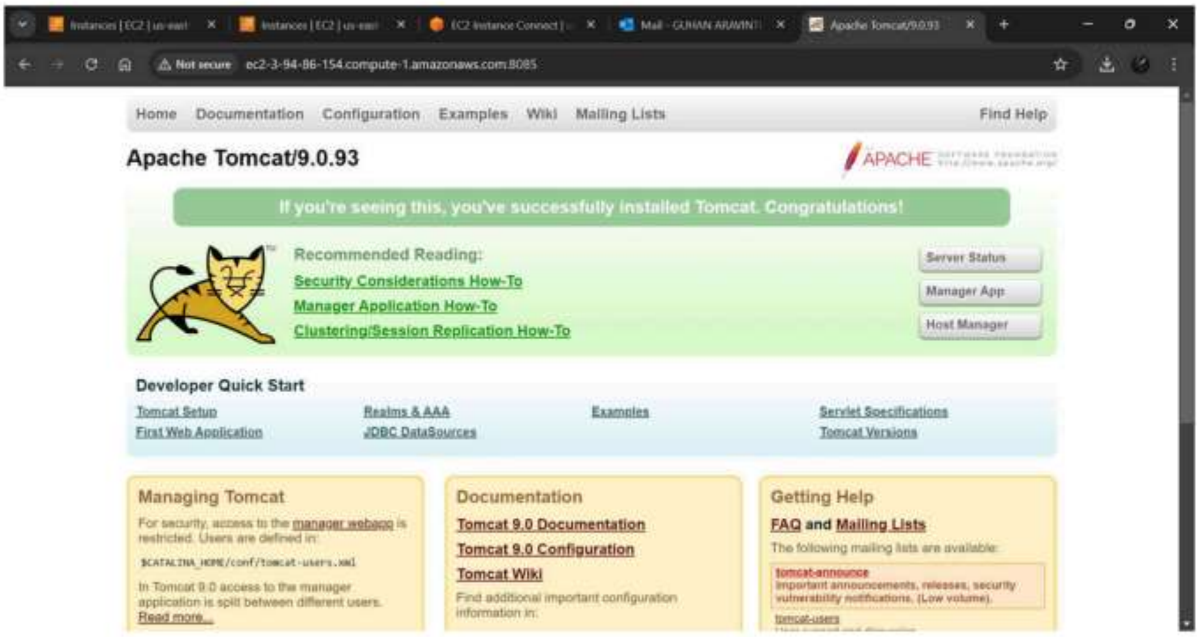
```
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one or more named thread pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
    -->


    <!-- 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="8085" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
               maxParameterCount="1000"
               />
    <!-- A "Connector" using the shared thread pool-->
    <!--
-- INSERT --                                                              71,26         33%
```

Step 14: Start the tomcat server.


root@ip-172-31-91-172:/home/ec2-user/apache-tomcat-9.0.93/bin

```
[ec2-user@ip-172-31-91-172 ~]$ vi webapps/manager/META-INF/context.xml
[ec2-user@ip-172-31-91-172 ~]$ cd apache-tomcat-9.0.93/
[ec2-user@ip-172-31-91-172 apache-tomcat-9.0.93]$ vi webapps/manager/META-INF/co
ntext.xml
[ec2-user@ip-172-31-91-172 apache-tomcat-9.0.93]$ sudo su
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# vi webapps/manager/META-INF/context.xml
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# vi conf/tomcat-user.xml
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# vi webapps/manager/META-INF/context.xml
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# vi conf/tomcat-users.xml
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# vi conf/server.xml
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  conf  logs  webapps
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  bin          lib   temp  work
[root@ip-172-31-91-172 apache-tomcat-9.0.93]# cd bin
[root@ip-172-31-91-172 bin]# ./startup.sh
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.93
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.93/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.93/bin/bootstrap.jar:/home/ec2-user/apache-tomcat-
9.0.93/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-91-172 bin]#
```

Step 15: Open Chrome tab and enter public ipv4 address followed by 8085 port of the Linux Instance.



Apache Tomcat/9.0.93

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Documentation

Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

Getting Help

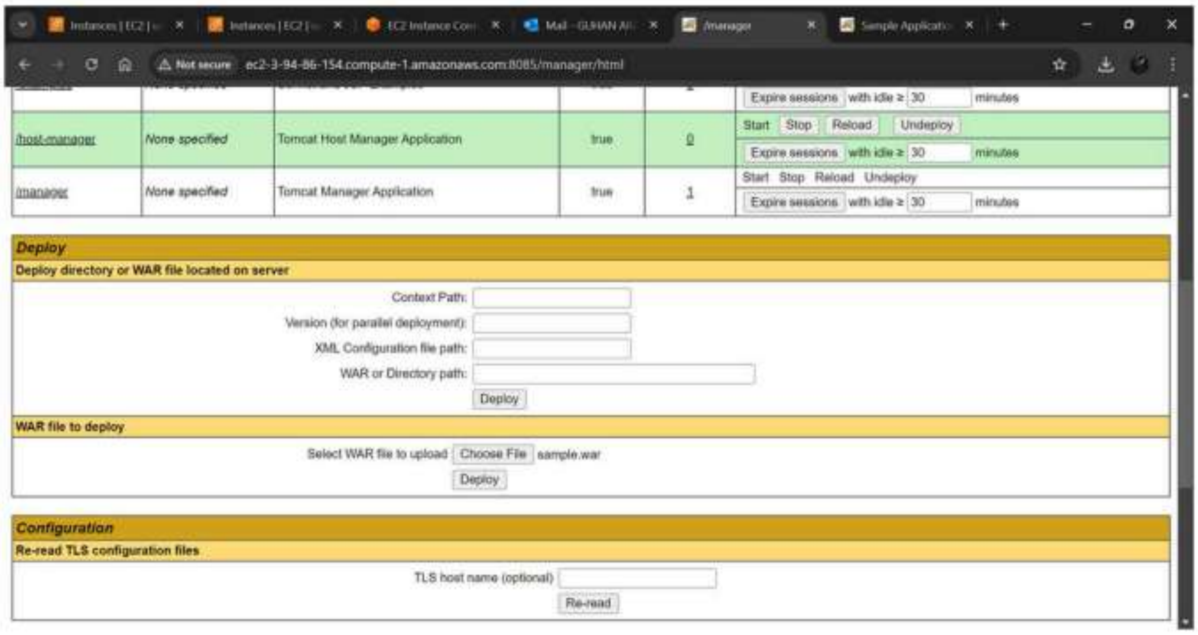
FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users

Step 16 : Download sample.war file and deploy using the dashboard.



ec2-3-94-86-154.compute-1.amazonaws.com:8085/manager/html

					Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy

WAR file to deploy

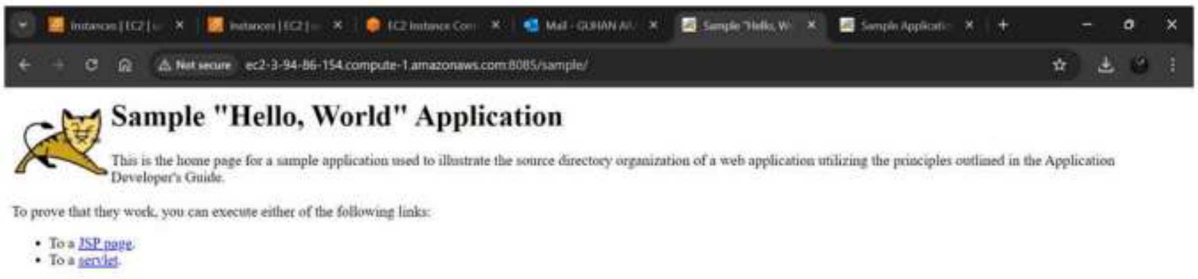
Select WAR file to upload Choose File sample.war
Deploy

Configuration

Re-read TLS configuration files

TLS host name (optional)
Re-read

Step 17 : After Deploying, Open the sample webpage in chrome tab.



ec2-3-94-86-154.compute-1.amazonaws.com:8085/sample/

Sample "Hello, World" Application

This is the home page for a sample application used to illustrate the source directory organization of a web application utilizing the principles outlined in the Application Developer's Guide.

To prove that they work, you can execute either of the following links:

- To a JSP page.
- To a servlet.

Evaluation by faculty	
Criteria	Marks
Preparation	/20
Program	/25
Output/Result	/20
Viva	/10
Total	/75
Faculty Signature with Date	

RESULT: