Name :- Komal Mhetre

Roles :- DevOps Engineer

Task :- Upload artifacts to nexus repository

Nexus

Introduction: -

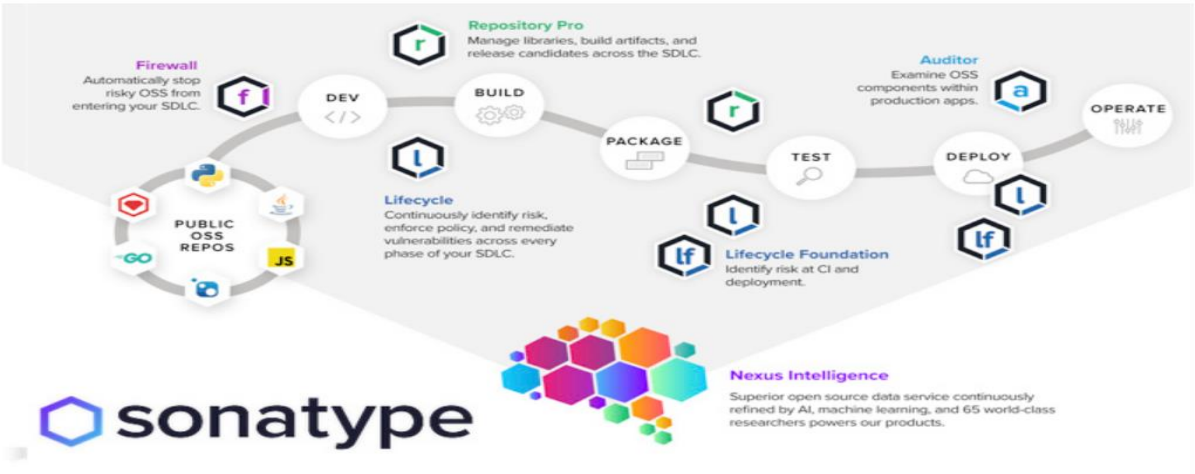
Basically the software is nexus and sonatype is company.

Sonatype Nexus is a repository manager used in software development for storing and managing binary artifacts. It provides a centralized platform for storing dependencies, libraries, and other artifacts used in the software development lifecycle. Nexus helps in ensuring that developers have access to the necessary components and dependencies required for building, testing, and deploying applications.

below are some key features and functionalities of Sonatype Nexus :-

1. **Artifact Management**:- Nexus allows organizations to store and manage various types of artifacts such as JARs, WARs, ZIPs, and Docker images. It supports multiple repository formats including Maven, npm, NuGet, PyPI, and others.
2. **Dependency Management**:- Developers can easily manage project dependencies by configuring Nexus repositories to proxy external repositories such as Maven Central, npm registry, or other public repositories. This helps in reducing build times and ensures consistency and reliability of dependencies.
3. **Proxying and Caching**:- Nexus can act as a proxy server for external repositories, caching artifacts locally to improve build performance and reduce external dependencies. This also provides control over the artifacts being used within the organization.
4. **Security and Access Control**:- Nexus offers robust security features including user authentication, role-based access control, and fine-grained permissions management. This ensures that only authorized users have access to specific artifacts and repositories.
5. **Lifecycle Management**:- Nexus facilitates the management of artifacts throughout their lifecycle, from development through testing and deployment. It supports features such as staging repositories for promoting artifacts through different environments and lifecycle stages.
6. **Integration with CI/CD Tools**:- Nexus seamlessly integrates with popular Continuous Integration and Continuous Deployment (CI/CD) tools such as Jenkins, TeamCity, and Bamboo, enabling automated build and deployment pipelines.
7. **Monitoring and Reporting**:- Nexus provides visibility into artifact usage, download statistics, and repository health through comprehensive monitoring and reporting features. This helps organizations in tracking dependencies, identifying bottlenecks, and optimizing resource usage.
8. **High Availability and Scalability**:- Nexus can be deployed in a high availability configuration with clustering support to ensure reliability and scalability, particularly in enterprise environments with large-scale artifact management requirements.

Overall, Sonatype Nexus plays a critical role in improving the efficiency, reliability, and security of software development processes by providing a centralized repository management solution for managing dependencies and artifacts.



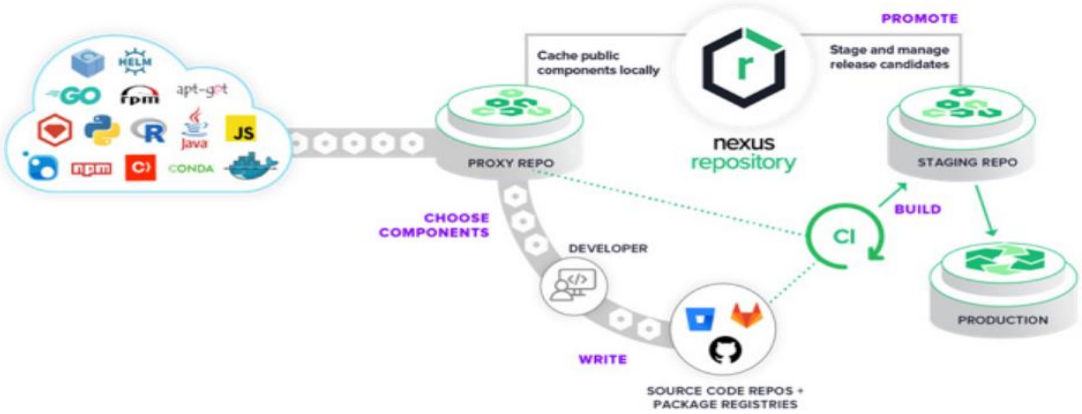
Nexus Repository

- Nexus by sonatype is an open source repository manager for build artifacts.
- Single source of truth for all of your components, binaries, and build artifacts.
- It organizes, stores and distributed artifacts needed for development.
- With nexus, developers can completely control access to, and deployment of, every artifact in an organization from a single location, making it easier to distribute software.
- Store and distribute Maven/Java, npm, Helm, Docker, API, Go, R and More.

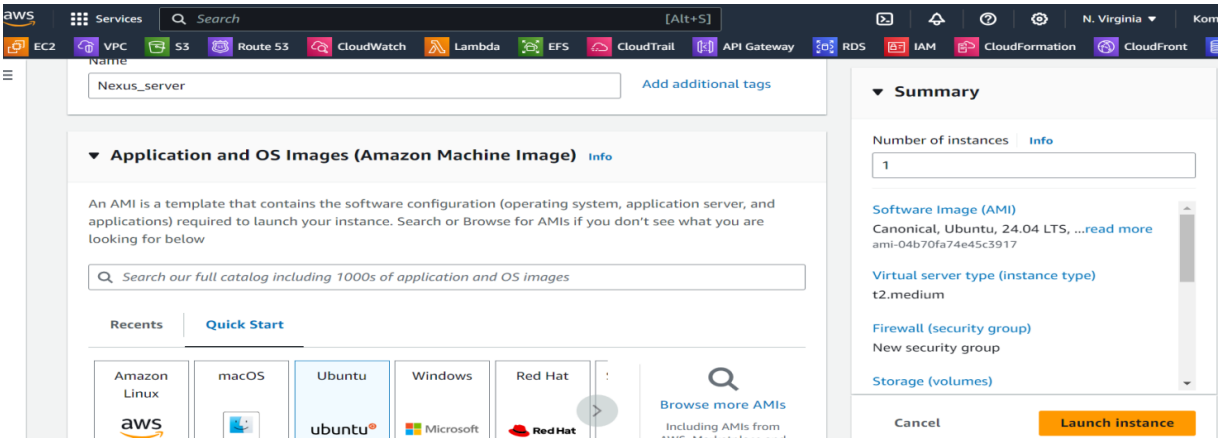
There are three types of repositories in Nexus for each Build/Package Manager Tool. (Maven, Npm etc.)

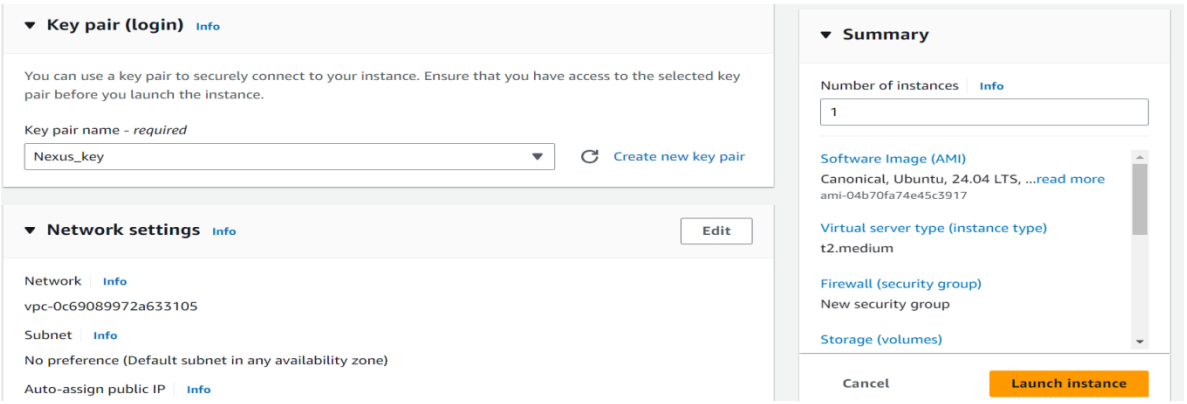
1. **Hosted Repositories:-** These are repositories where you can deploy your own artifacts. They are typically used for hosting internally developed libraries, components, or packages.
2. **Proxy Repositories:-** Proxy repositories cache artifacts from remote repositories such as Maven Central, npm registry, or PyPI. They help improve build performance by providing local access to frequently used dependencies.

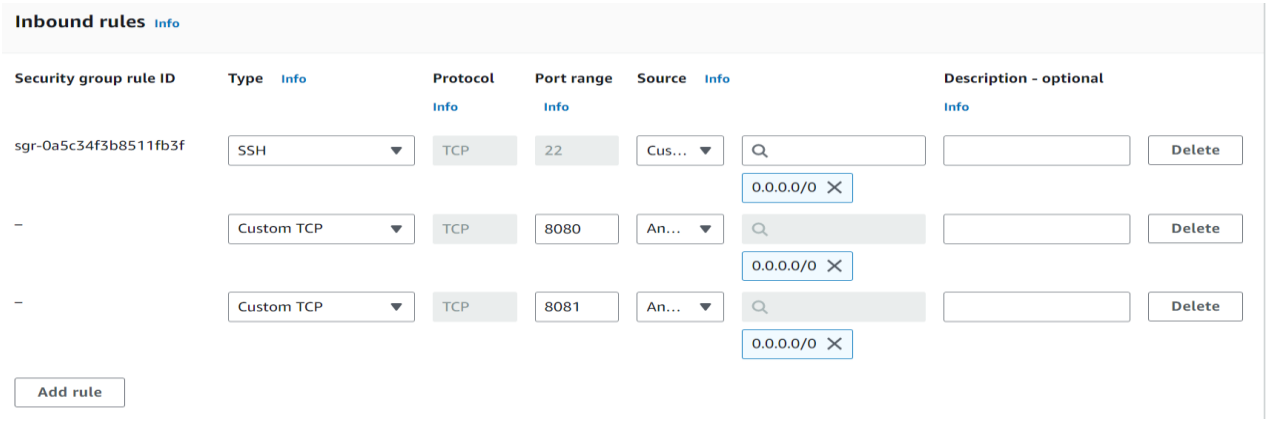
3. **Group Repositories:-** Group repositories combine multiple hosted and proxy repositories into a single virtual repository. This allows users to access artifacts from multiple sources through a single URL.



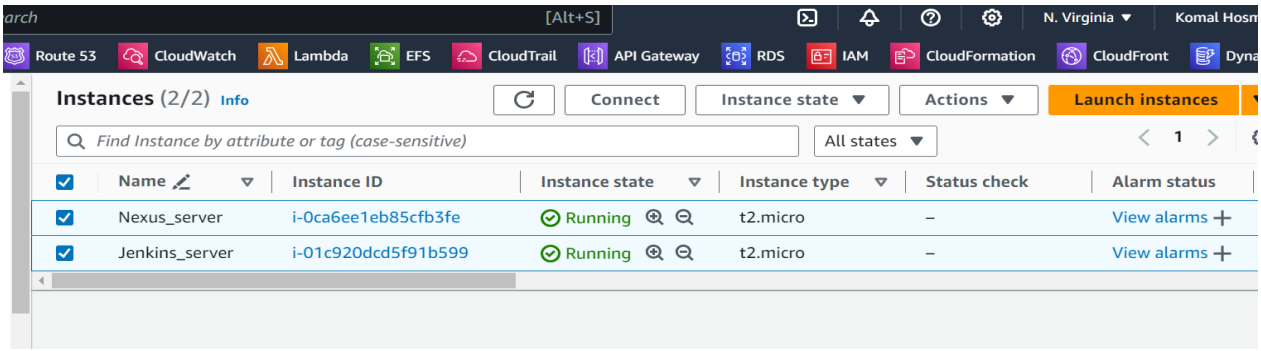
1. First, launch an instance and select the t2.medium instance type for it.







2. You need to perform the same process on the Jenkins server as well. You have to add the port numbers for Nexus and Jenkins.



3. Below are the commands to install Sonatype Nexus on your instance.

Nexus-Installation

sudo apt update -y

sudo apt-get install openjdk-8-jre-headless

cd /opt/

sudo wget http://download.sonatype.com/nexus/3/latest-unix.tar.gz

sudo tar -zxvf latest-unix.tar.gz

sudo mv nexus-3.67.1-01 nexus

sudo adduser nexus

sudo chmod 755 /etc/sudoers

sudo vi /etc/sudoers

jenkins ALL=(ALL) NOPASSWD:ALL
nexus ALL=()ALL) NOPASSWD:ALL

sudo chown -R nexus:nexus /opt/nexus

sudo chown -R nexus:nexus /opt/sonatype-work

sudo vi /opt/nexus/bin/nexus.rc

sudo vi /opt/nexus/bin/nexus.vmoptions

sudo vi /etc/systemd/system/nexus.service
in this directory...

Copy the below content.

[Unit]
Description=nexus service
After=network.target

[Service]
Type=forking
LimitNOFILE=65536
User=nexus
Group=nexus
ExecStart=/opt/nexus/bin/nexus start
ExecStop=/opt/nexus/bin/nexus stop
User=nexus
Restart=on-abort
[Install]
WantedBy=multi-user.target

Now Start Nexus

sudo systemctl enable nexus

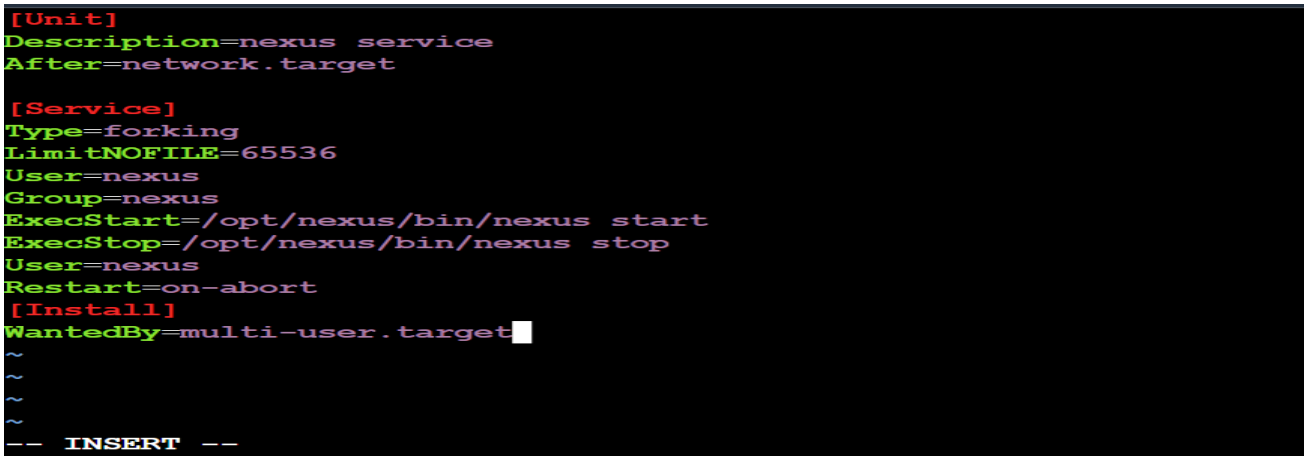
sudo systemctl start nexus

sudo systemctl status nexus

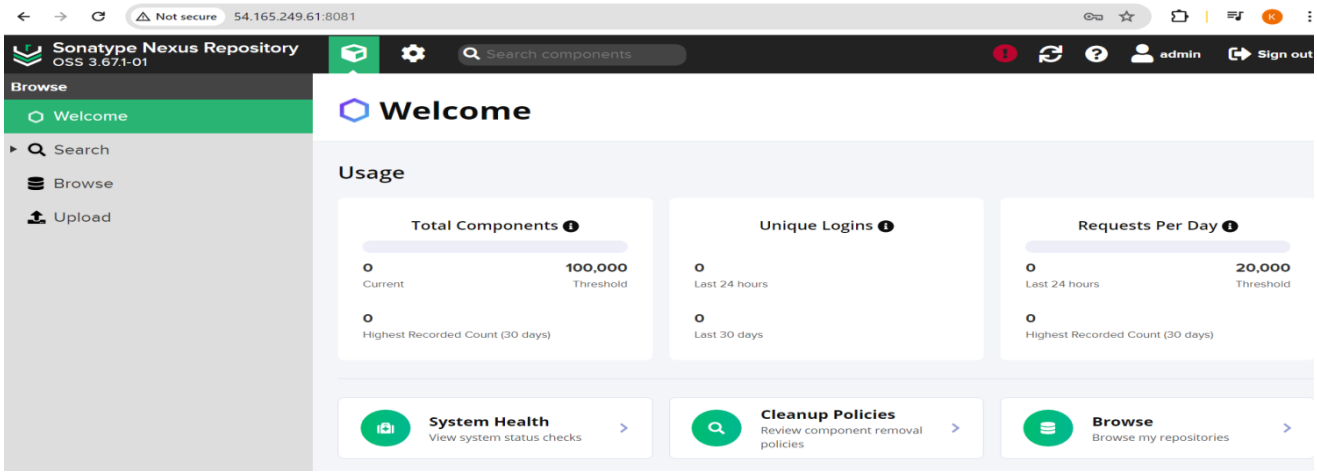
```
ubuntu@ip-172-31-61-129:/opt$ sudo adduser nexus
info: Adding user `nexus' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `nexus' (1001) ...
info: Adding new user `nexus' (1001) with group `nexus (1001)' ...
info: Creating home directory `/home/nexus' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
```

4. #run_as-user=”nexus”

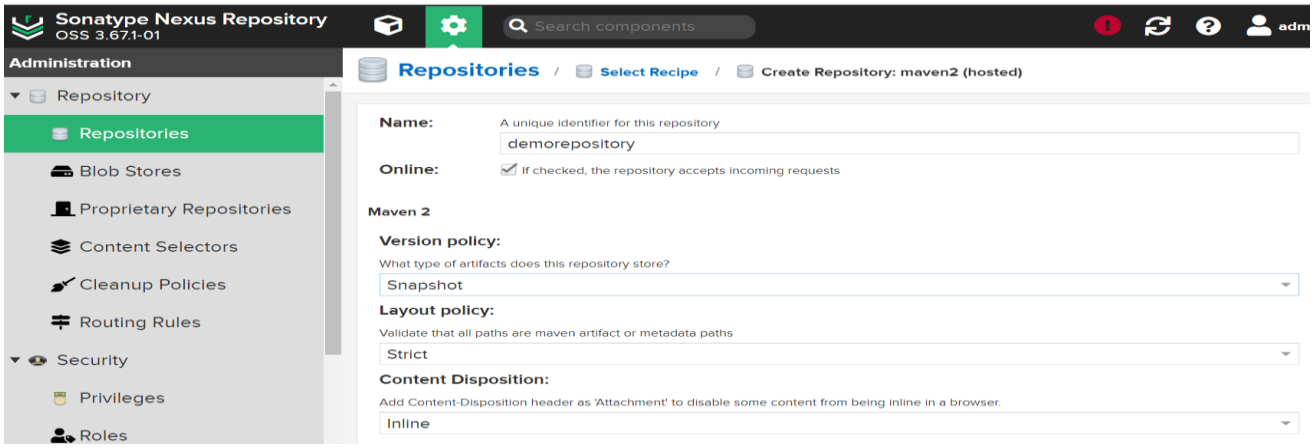
5. Configure nexus to run as a service.

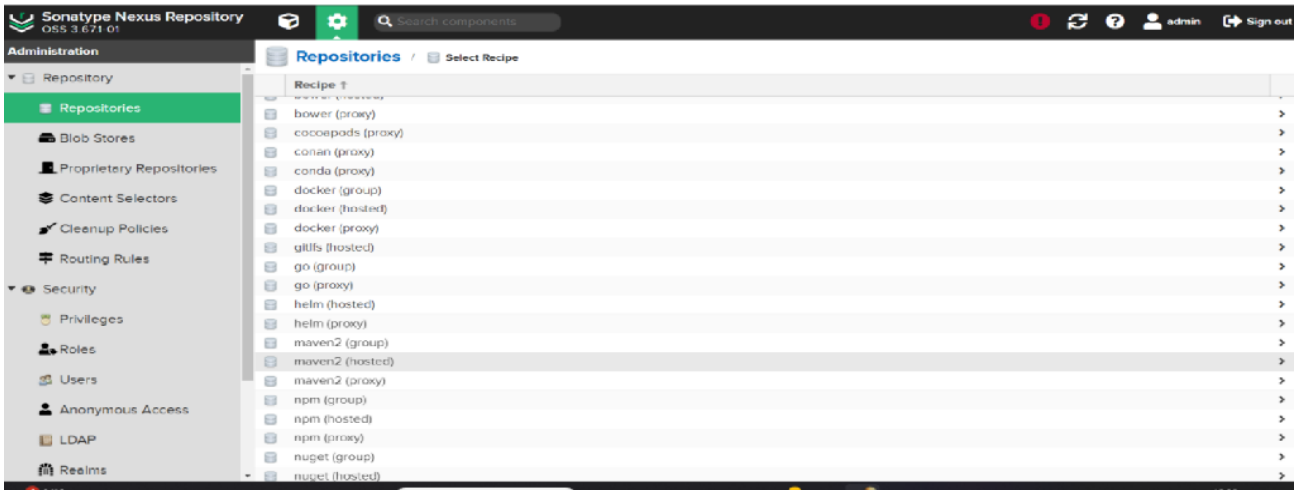


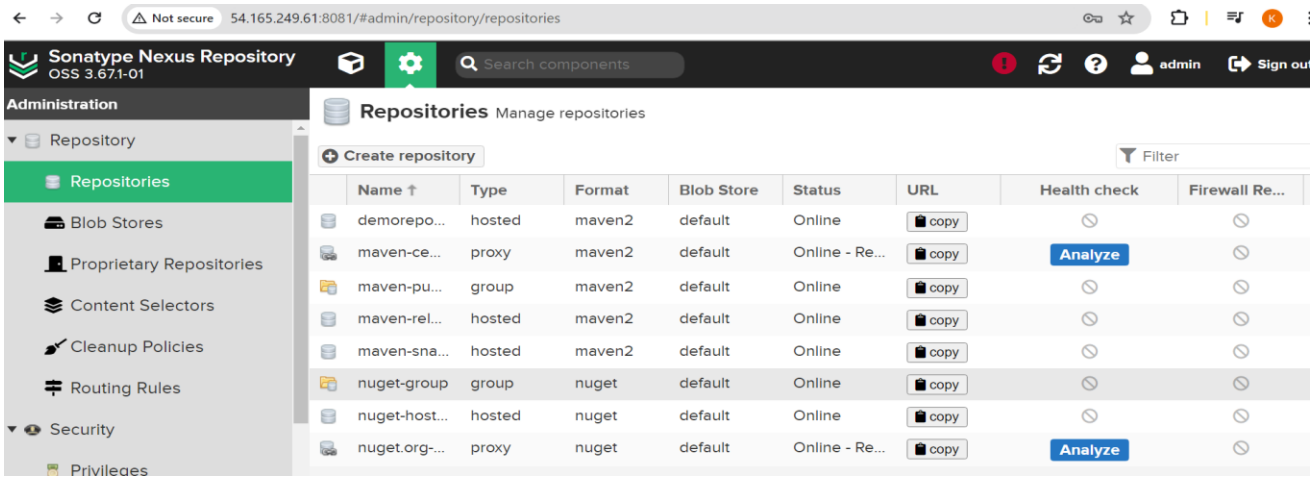
6. Now you can access Sonatype-Nexus at https://localhost:8081 in your browser.



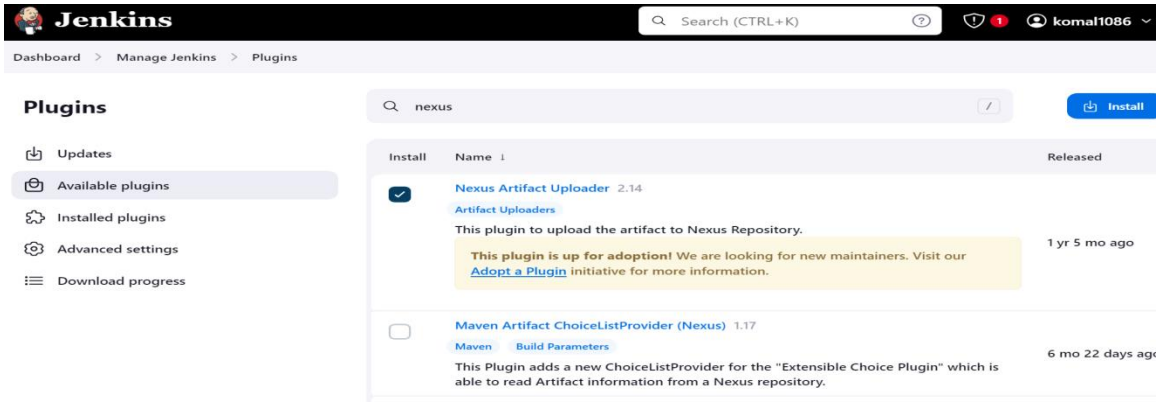
7. Create a repository in the nexus server.
   Follow the below steps:
   a. Go to the Nexus server
   b. create a repository
   c. And select 'Maven2 hosted'.
   d. Enter repository name and create repository.



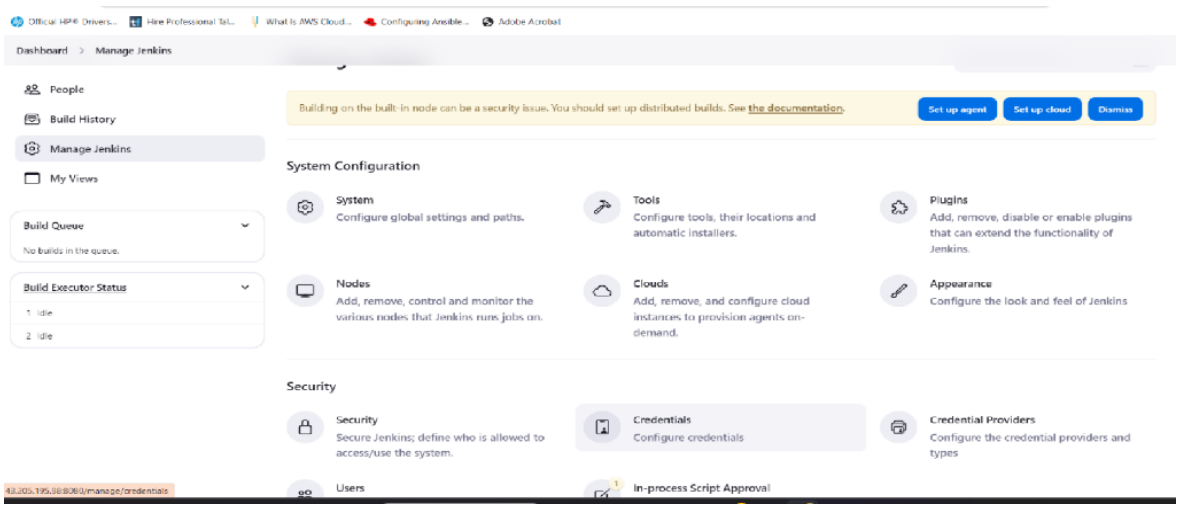


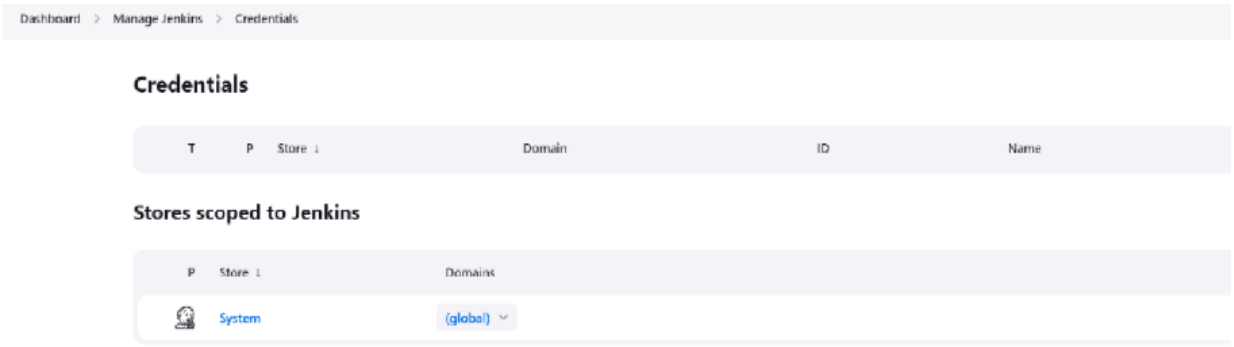


8. Next, add the credentials of Nexus in Jenkins.
   Go to the available plugins and install the Nexus Artifact Uploader.

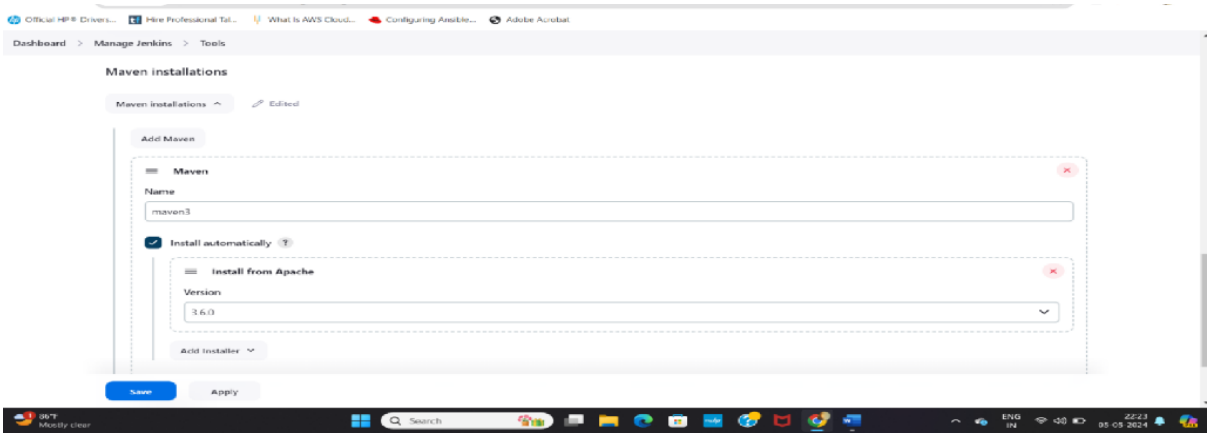


9. Then go to 'Manage Jenkins', click on 'Credentials', select 'Global', and click on 'Add Credentials'.

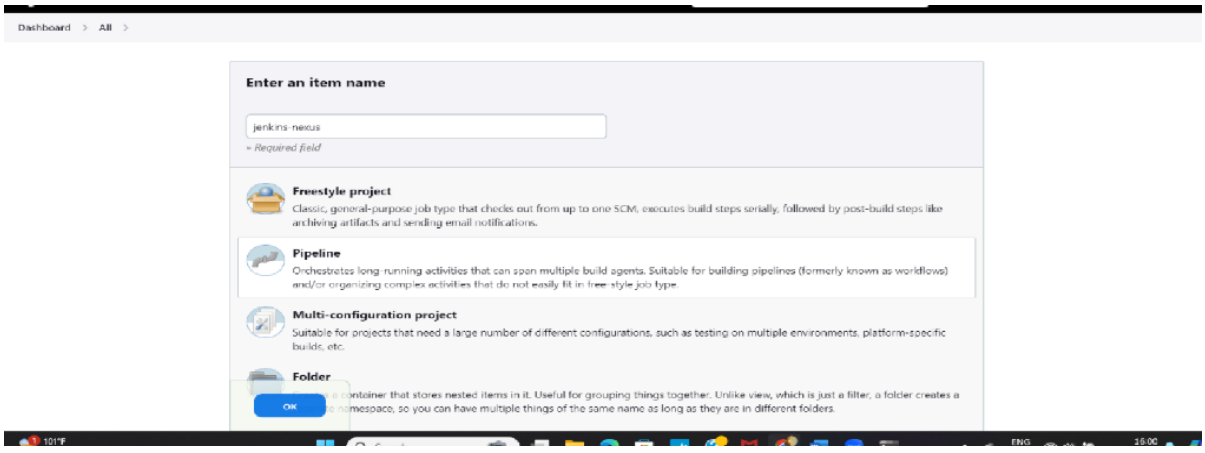




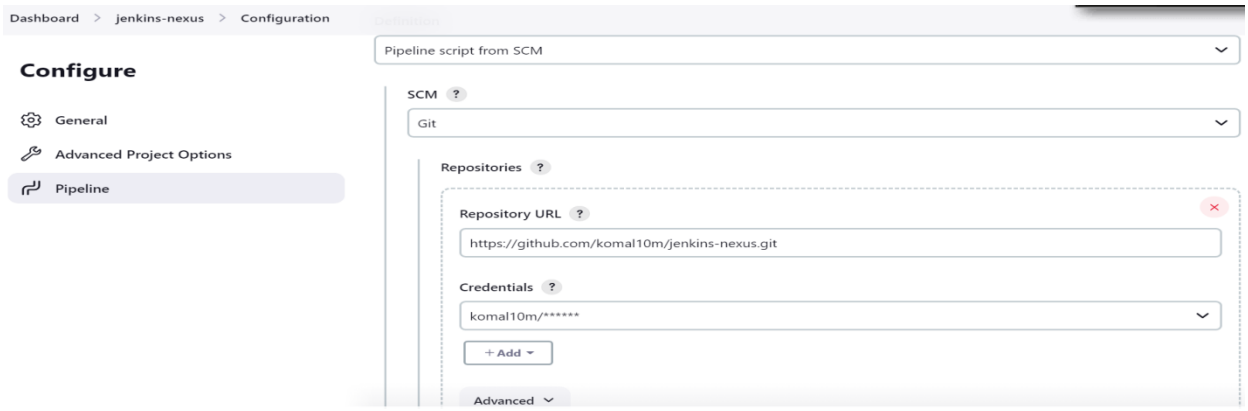
10. Go to 'Manage Jenkins', then 'Tools', and finally, add Maven.

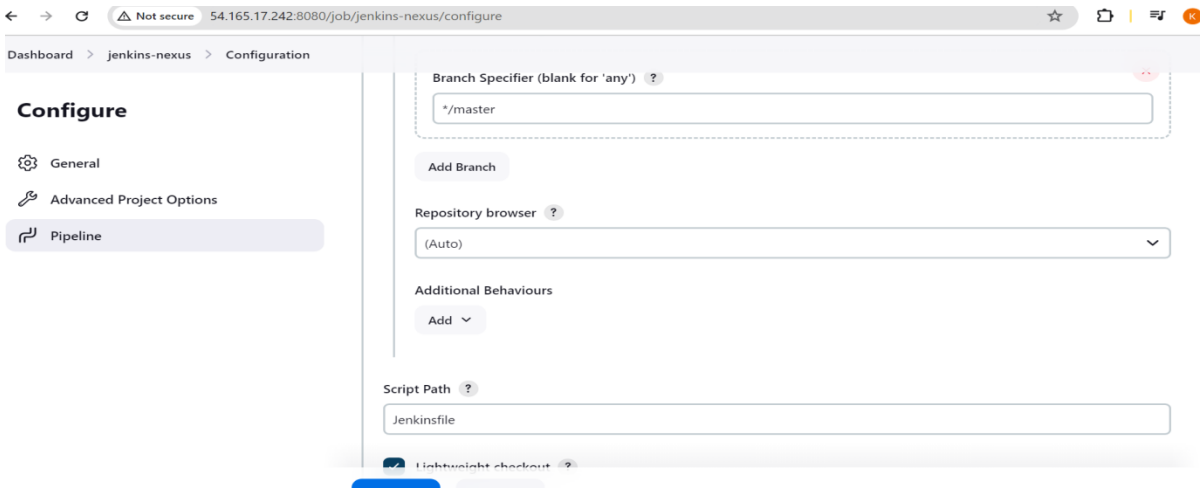


11. To create a pipeline, navigate to 'New Item', provide a name for the pipeline, select 'Pipeline', and then click Ok.

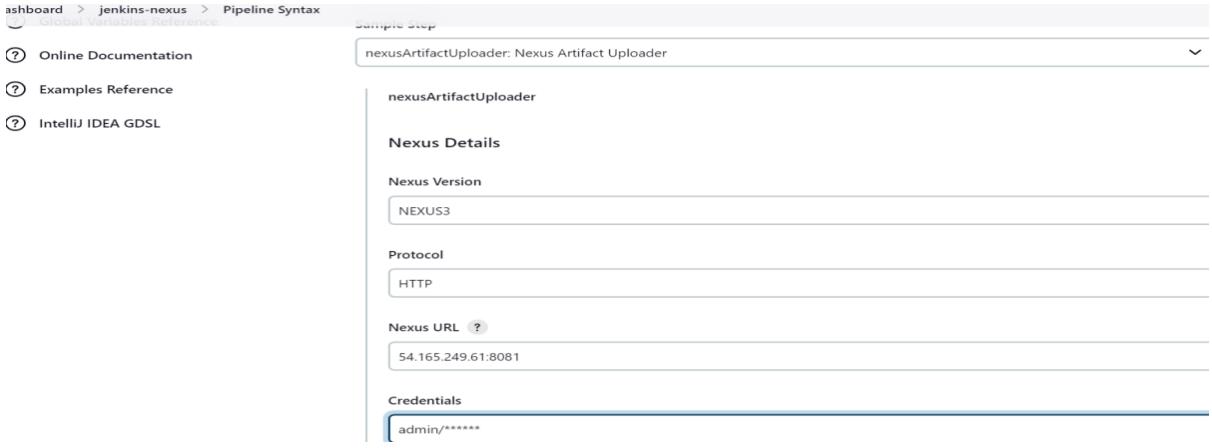


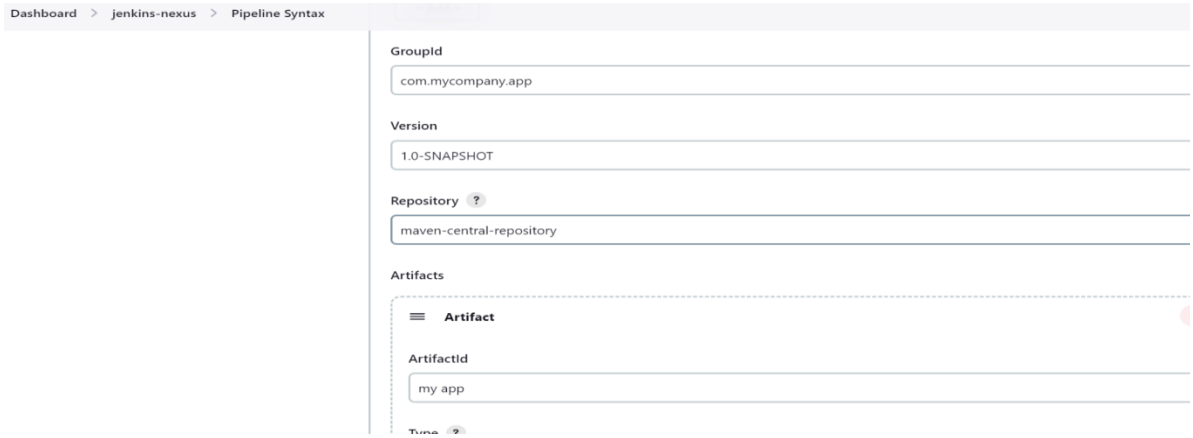
12. Next, go to the pipeline and select 'Pipeline script from SCM'



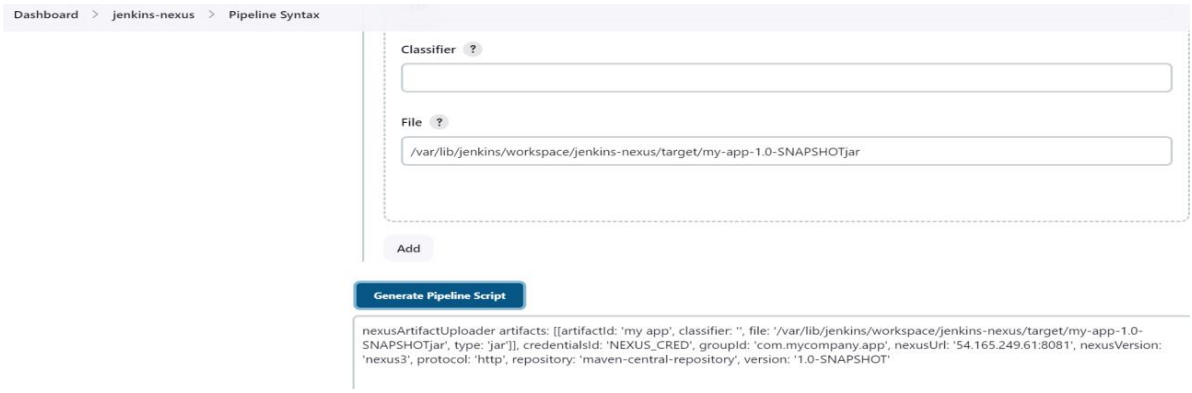


13. Select 'Pipeline Syntax', input the required data, and generate the script.

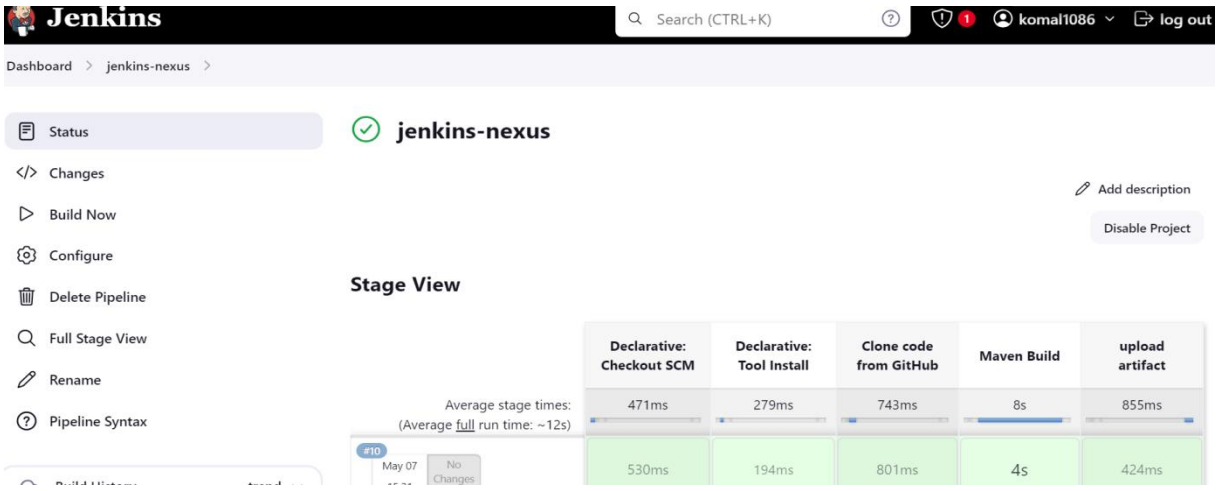




14. Copy the script and add the 'upload to artifact' stage.



15. Click to build now. And see the console output.



```
Downloaded: http://54.165.249.61:8081/repository/demorepository/com/mycompany/app/my-app/1.0-SNAPSHOT/maven-
metadata.xml (596 B at 18 kB/s)
Uploading: http://54.165.249.61:8081/repository/demorepository/com/mycompany/app/my-app/1.0-SNAPSHOT/my-app-1.0-
20240507.095114-2.jar
Uploaded: http://54.165.249.61:8081/repository/demorepository/com/mycompany/app/my-app/1.0-SNAPSHOT/my-app-1.0-
20240507.095114-2.jar (2.6 kB at 35 kB/s)
Uploading: http://54.165.249.61:8081/repository/demorepository/com/mycompany/app/my-app/1.0-SNAPSHOT/maven-
metadata.xml
Uploaded: http://54.165.249.61:8081/repository/demorepository/com/mycompany/app/my-app/1.0-SNAPSHOT/maven-
metadata.xml (596 B at 8.1 kB/s)
Uploading artifact my-app-1.0-SNAPSHOT.jar completed.
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

16. Go to the Nexus server and see the result.