

## Lesson 09 Demo 03

# Implementing Continuous Monitoring on Docker with ELK Stack

**Objective:** To demonstrate continuous monitoring on Docker with ELK stack for efficient log management and analysis

**Tools required:** Jenkins, Docker, and ELK stack

**Prerequisites:** You need to have Docker and Jenkins installed, along with a set of required plugins such as Docker, Docker pipeline, and build pipeline. If you don't have Jenkins installed, refer to the Demo 01 of Lesson 05. For Docker installation instructions, refer to the Demo 01 of Lesson 07 or the lab guide.

Additionally, you need to have a Docker Hub account to proceed with this demo. If you don't have one, you can create it by visiting <https://hub.docker.com/>

Steps to be followed:

1. Set up ELK stack on Docker
2. Configure Jenkins pipeline for Docker build and deployment
3. Run the Spring Boot application and verify the logs in Kibana

### Step 1: Set up ELK stack on Docker

1.1 Switch to the root user using the following command:

**sudo su**

```
labsuser@ip-172-31-41-35:~$ sudo su
root@ip-172-31-41-35:/home/labsuser#
```

- 1.2 Execute the following command to clone the repository from GitHub using Git:  
**git clone https://github.com/Siraj-ul-muneera/ELKExample.git**

```
labsuser@ip-172-31-41-35:~$ sudo su
root@ip-172-31-41-35:/home/labsuser# git clone https://github.com/Siraj-ul-muneera/ELKExample.git
Cloning into 'ELKExample'...
remote: Enumerating objects: 104, done.
remote: Total 104 (delta 0), reused 0 (delta 0), pack-reused 104
Receiving objects: 100% (104/104), 1.02 MiB | 6.97 MiB/s, done.
Resolving deltas: 100% (34/34), done.
root@ip-172-31-41-35:/home/labsuser#
```

- 1.3 Navigate to the **ELKExample** directory using the following command:  
**cd ELKExample**

```
labsuser@ip-172-31-41-35:~$ sudo su
root@ip-172-31-41-35:/home/labsuser# git clone https://github.com/Siraj-ul-muneera/ELKExample.git
Cloning into 'ELKExample'...
remote: Enumerating objects: 104, done.
remote: Total 104 (delta 0), reused 0 (delta 0), pack-reused 104
Receiving objects: 100% (104/104), 1.02 MiB | 6.97 MiB/s, done.
Resolving deltas: 100% (34/34), done.
root@ip-172-31-41-35:/home/labsuser# cd ELKExample
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

- 1.4 List the contents of the **ELKExample** directory using the following command:  
**ls -alrt**

```
Resolving deltas: 100% (34/34), done.
root@ip-172-31-41-35:/home/labsuser# cd ELKExample
root@ip-172-31-41-35:/home/labsuser/ELKExample# ls -alrt
total 40
drwxr-x--- 23 labsuser labsuser 4096 Feb 15 12:31 ..
-rw-r--r--  1 root    root    2185 Feb 15 12:31 pom.xml
-rw-r--r--  1 root    root    725 Feb 15 12:31 docker-compose.yml
drwxr-xr-x  2 root    root    4096 Feb 15 12:31 conf.d
-rw-r--r--  1 root    root    626 Feb 15 12:31 Jenkinsfile
-rw-r--r--  1 root    root    95 Feb 15 12:31 Dockerfile
-rw-r--r--  1 root    root    497 Feb 15 12:31 .gitignore
drwxr-xr-x  5 root    root    4096 Feb 15 12:31 .
drwxr-xr-x  4 root    root    4096 Feb 15 12:31 src
drwxr-xr-x  8 root    root    4096 Feb 15 12:31 .git
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

- 1.5 Execute the following command to install Docker Compose using the **apt** package manager:

**sudo apt install docker-compose**

```
root@ip-172-31-41-35:/home/labsuser/ELKExample# ls -alrt
total 40
drwxr-x--- 23 labsuser labsuser 4096 Feb 15 12:31 ..
-rw-r--r--  1 root      root      2185 Feb 15 12:31 pom.xml
-rw-r--r--  1 root      root      725 Feb 15 12:31 docker-compose.yml
drwxr-xr-x  2 root      root      4096 Feb 15 12:31 conf.d
-rw-r--r--  1 root      root      626 Feb 15 12:31 Jenkinsfile
-rw-r--r--  1 root      root      95 Feb 15 12:31 Dockerfile
-rw-r--r--  1 root      root      497 Feb 15 12:31 .gitignore
drwxr-xr-x  5 root      root      4096 Feb 15 12:31 .
drwxr-xr-x  4 root      root      4096 Feb 15 12:31 src
drwxr-xr-x  8 root      root      4096 Feb 15 12:31 .git
root@ip-172-31-41-35:/home/labsuser/ELKExample# apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-compose is already the newest version (1.29.2-1).
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

- 1.6 Check the version of Docker Compose installed on your system by running the following command:

**docker-compose version**

```
root@ip-172-31-41-35:/home/labsuser/ELKExample# apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-compose is already the newest version (1.29.2-1).
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
root@ip-172-31-41-35:/home/labsuser/ELKExample# docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
CPython version: 3.10.12
OpenSSL version: OpenSSL 3.0.2 15 Mar 2022
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

- 1.7 Set the **vm.max\_map\_count** kernel parameter to **262144** using the following command:  
**sysctl -w vm.max\_map\_count=262144**

```
root@ip-172-31-41-35:/home/labsuser/ELKExample# apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker-compose is already the newest version (1.29.2-1).
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 74 not upgraded.
root@ip-172-31-41-35:/home/labsuser/ELKExample# docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
CPython version: 3.10.12
OpenSSL version: OpenSSL 3.0.2 15 Mar 2022
root@ip-172-31-41-35:/home/labsuser/ELKExample# sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

- 1.8 Start Docker Compose services in detached mode using the following command:  
**docker-compose up -d**

```
root@ip-172-31-41-35:/home/labsuser/ELKExample# docker-compose up -d
Pulling elasticsearch (docker.elastic.co/elasticsearch/elasticsearch:6.8.1)...
6.8.1: Pulling from elasticsearch/elasticsearch
48914619bcd3: Pull complete
531da6002229: Pull complete
f20133847577: Pull complete
c3eafc6f3b95: Pull complete
b050faa5a2c7: Pull complete
a7a4a7e291f5: Pull complete
fa6210407af9: Pull complete
Digest: sha256:974aba875f1ec333f0c2ac7dee1e74d70c5813b039b3a225eb2bf287d4aa5911
Status: Downloaded newer image for docker.elastic.co/elasticsearch/elasticsearch:6.8.1
Pulling kibana (docker.elastic.co/kibana/kibana:6.8.1)...
6.8.1: Pulling from kibana/kibana
48914619bcd3: Already exists
6ba74e058204: Pull complete
82650f51a90e: Pull complete
412872540b65: Pull complete
0a36e0e61194: Pull complete
6619818d9edd: Pull complete
5a7fb0e726f2: Pull complete
e8b50a507459: Pull complete
Digest: sha256:dbcaccde1ddc16ab12c841b22b2de1fc5d76de5ff0ac843fc3e825ef96781f80
Status: Downloaded newer image for docker.elastic.co/kibana/kibana:6.8.1
Pulling logstash (docker.elastic.co/logstash/logstash:6.8.1)...
6.8.1: Pulling from logstash/logstash
48914619bcd3: Already exists
66d274cd25ed: Pull complete
6bbda190e7e1: Pull complete
af42a6a9f3ad: Pull complete
90da6e642d96: Pull complete
```

```
Pulling kibana (docker.elastic.co/kibana/kibana:6.8.1)...
6.8.1: Pulling from kibana/kibana
48914619bcd3: Already exists
6ba74e058204: Pull complete
82650f51a90e: Pull complete
412872540b65: Pull complete
0a36e0e61194: Pull complete
6619818d9edd: Pull complete
5a7fb0e726f2: Pull complete
e8b50a507459: Pull complete
Digest: sha256:dbcaccde1ddc16ab12c841b22b2de1fc5d76de5ff0ac843fc3e825ef96781f80
Status: Downloaded newer image for docker.elastic.co/kibana/kibana:6.8.1
Pulling logstash (docker.elastic.co/logstash/logstash:6.8.1)...
6.8.1: Pulling from logstash/logstash
48914619bcd3: Already exists
66d274cd25ed: Pull complete
6bbda190e7e1: Pull complete
af42a6a9f3ad: Pull complete
90da6e642d96: Pull complete
0096a7244eb6: Pull complete
cd5133a18825: Pull complete
1e3afd2f6e44: Pull complete
6a0955b1c5e4: Pull complete
46b308c281e6: Pull complete
860bd9a73a26: Pull complete
Digest: sha256:2280dc3c1deb9d9006cd2a423e09e649ed19109292690f52dd303d0048fc68e1
Status: Downloaded newer image for docker.elastic.co/logstash/logstash:6.8.1
Creating elkexample_elasticsearch_1 ... done
Creating elkexample_logstash_1 ... done
Creating elkexample_kibana_1 ... done
root@ip-172-31-41-35:/home/labsuser/ELKExample#
```

## 1.9 Use the following command to list all running Docker containers:

### docker ps

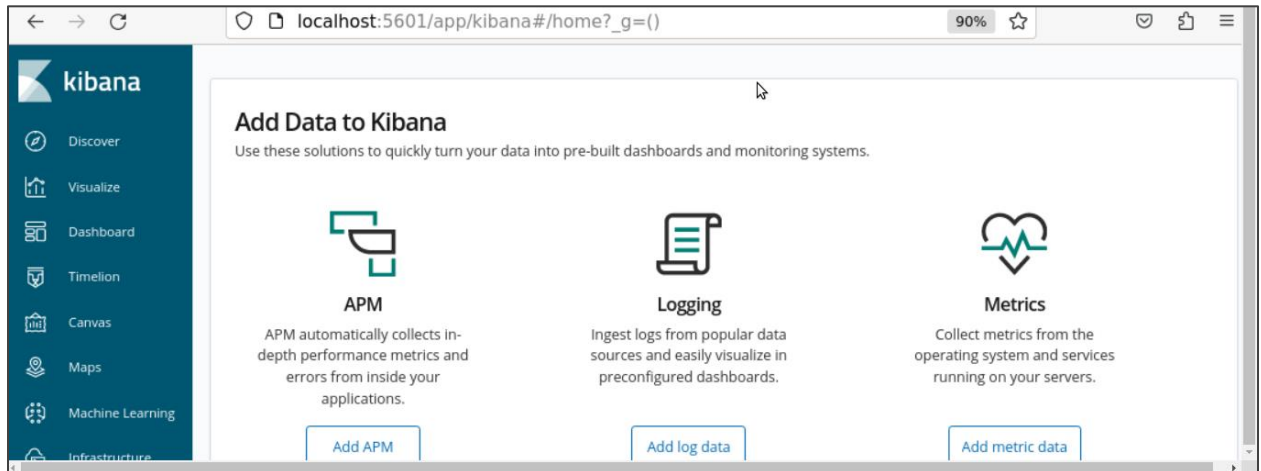
```
e8b50a507459: Pull complete
Digest: sha256:dbcaccde1ddc16ab12c841b22b2de1fc5d76de5ff0ac843fc3e825ef96781f80
Status: Downloaded newer image for docker.elastic.co/kibana/kibana:6.8.1
Pulling logstash (docker.elastic.co/logstash/logstash:6.8.1)...
6.8.1: Pulling from logstash/logstash
48914619bcd3: Already exists
66d274cd25ed: Pull complete
6bbda190e7e1: Pull complete
af42a6a9f3ad: Pull complete
90da6e642d96: Pull complete
0096a7244eb6: Pull complete
cd5133a18825: Pull complete
1e3afd2f6e44: Pull complete
6a0955b1c5e4: Pull complete
46b308c281e6: Pull complete
860bd9a73a26: Pull complete
Digest: sha256:2280dc3c1deb9d9006cd2a423e09e649ed19109292690f52dd303d0048fc68e1
Status: Downloaded newer image for docker.elastic.co/logstash/logstash:6.8.1
Creating elkexample_elasticsearch_1 ... done
Creating elkexample_logstash_1 ... done
Creating elkexample_kibana_1 ... done
root@ip-172-31-45-41:/home/labsuser/ELKExample# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
b83b30a8d5a4	docker.elastic.co/logstash/logstash:6.8.1	"/usr/local/bin/dock..."	About a minute ago	Up About a minute	5044/tcp, 9600
/tcp, 0.0.0.0:25826->25826/tcp, :::25826->25826/tcp	elkexample_logstash_1				
b615335ecbf6	docker.elastic.co/kibana/kibana:6.8.1	"/usr/local/bin/kiba..."	About a minute ago	Up About a minute	0.0.0.0:5601->
5601/tcp, :::5601->5601/tcp	elkexample_kibana_1				
6af6b67fb491	docker.elastic.co/elasticsearch/elasticsearch:6.8.1	"/usr/local/bin/dock..."	About a minute ago	Up About a minute	0.0.0.0:9200->
9200/tcp, ::9200->9200/tcp, 0.0.0.0:9300->9300/tcp, :::9300->9300/tcp	elkexample_elasticsearch_1				

```
root@ip-172-31-45-41:/home/labsuser/ELKExample#
```

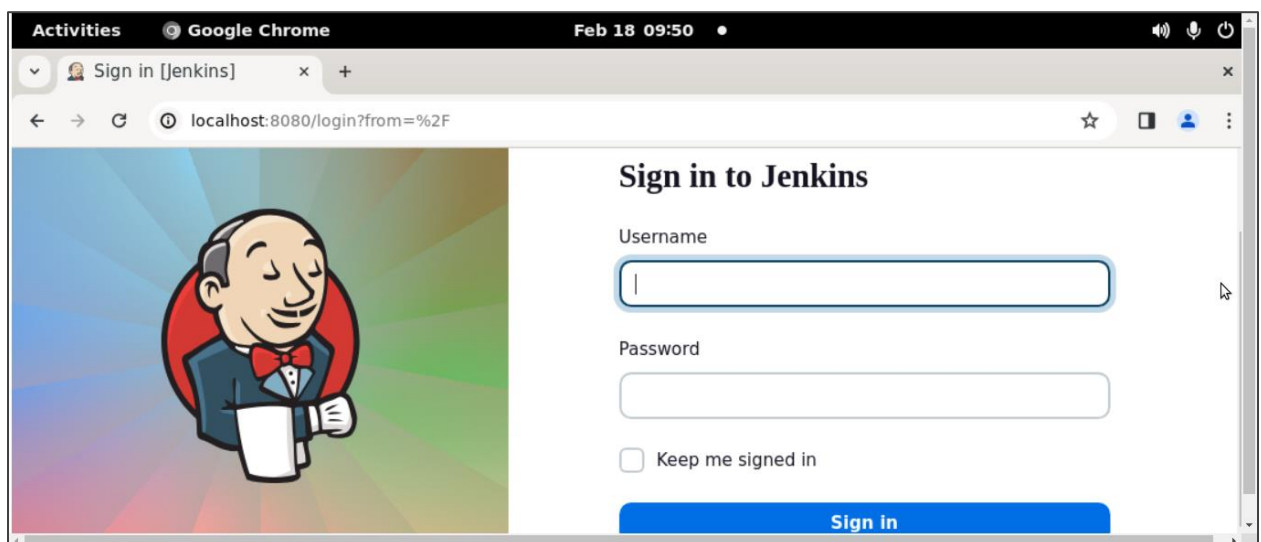
1.10 Access the Kibana dashboard using the public IP of the host and port 5601 using the following link as shown in the screenshot below:

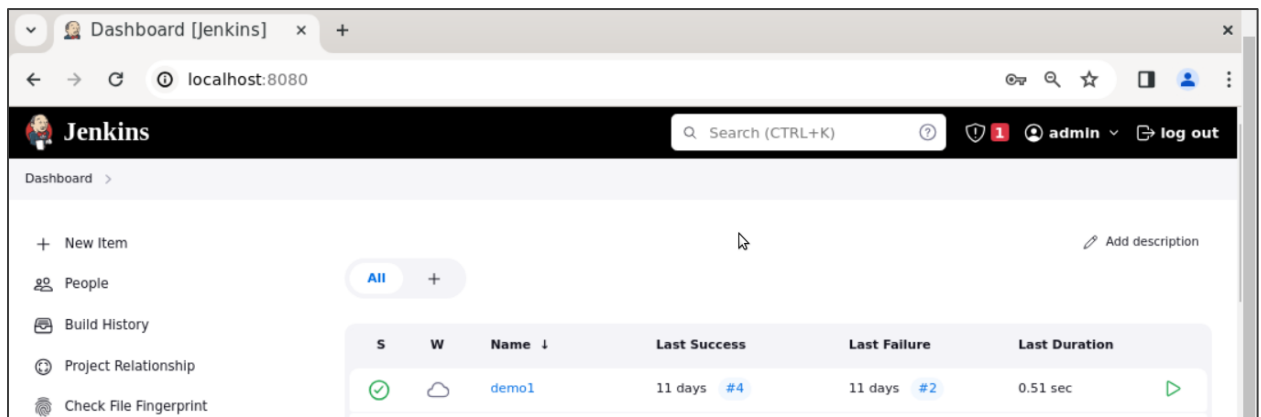
**<http://localhost:5601/app/kibana>**



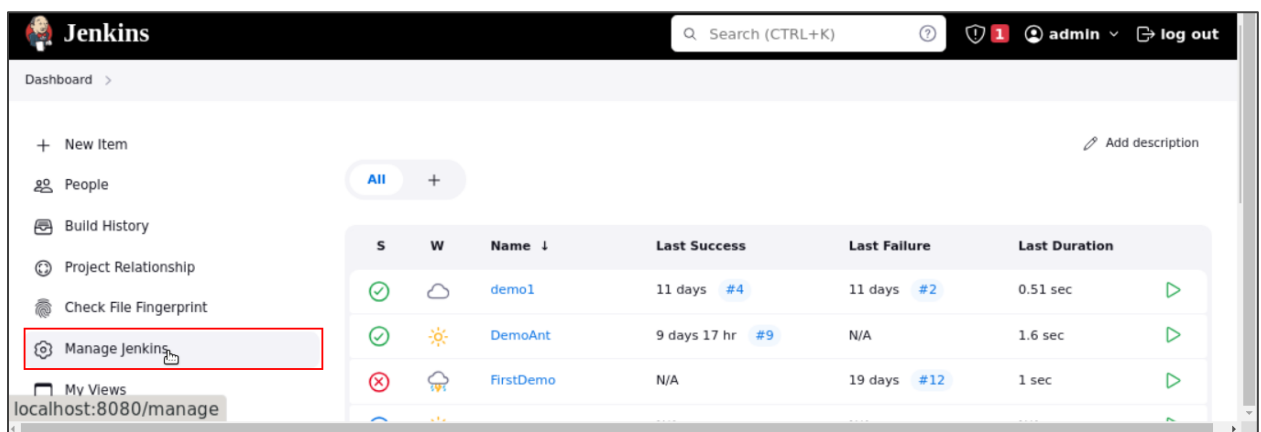
## Step 2: Configure Jenkins pipeline for Docker build and deployment

2.1 Navigate to **<http://localhost:8080>** and log in to Jenkins

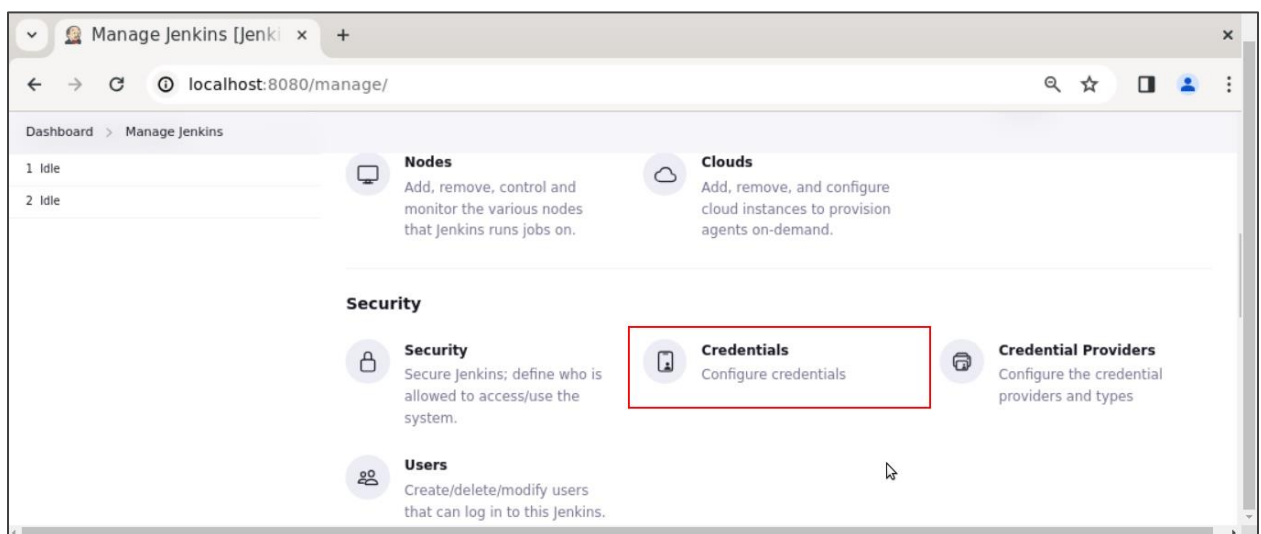




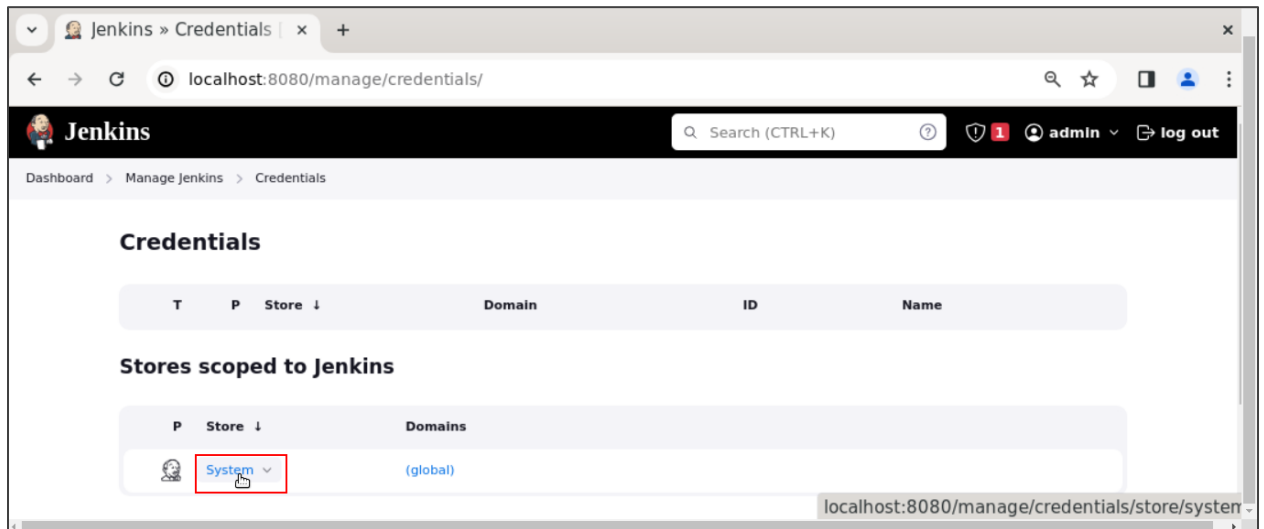
2.2 Click on **Manage Jenkins** as shown in the screenshot below:



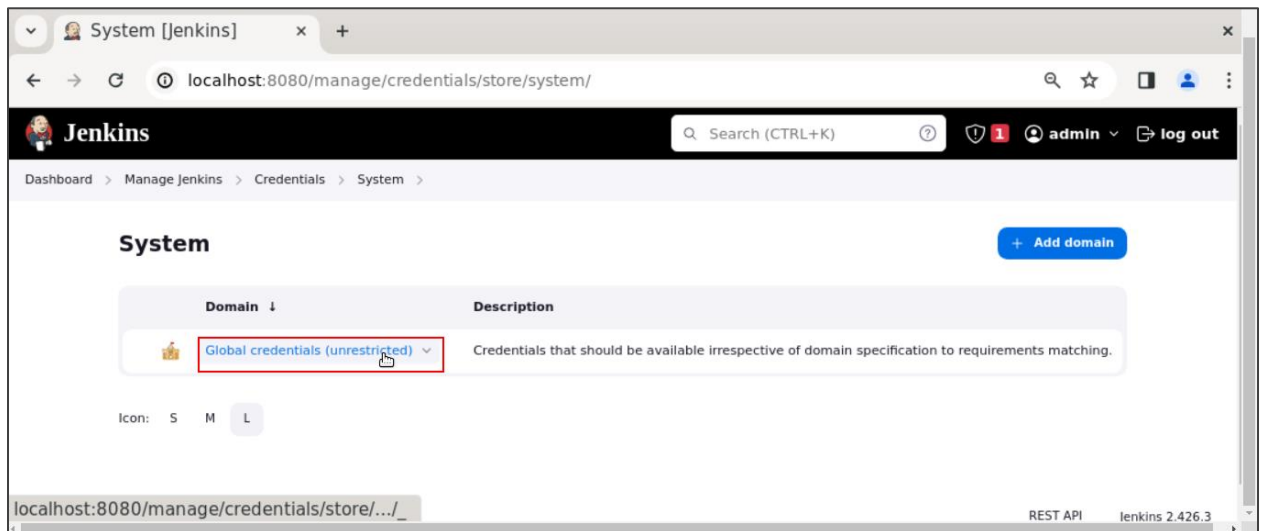
2.3 Scroll to the **Security** section and click on **Credentials** as shown in the screenshot below:



2.4 In the **Store** section, click on **System** as shown in the screenshot below:

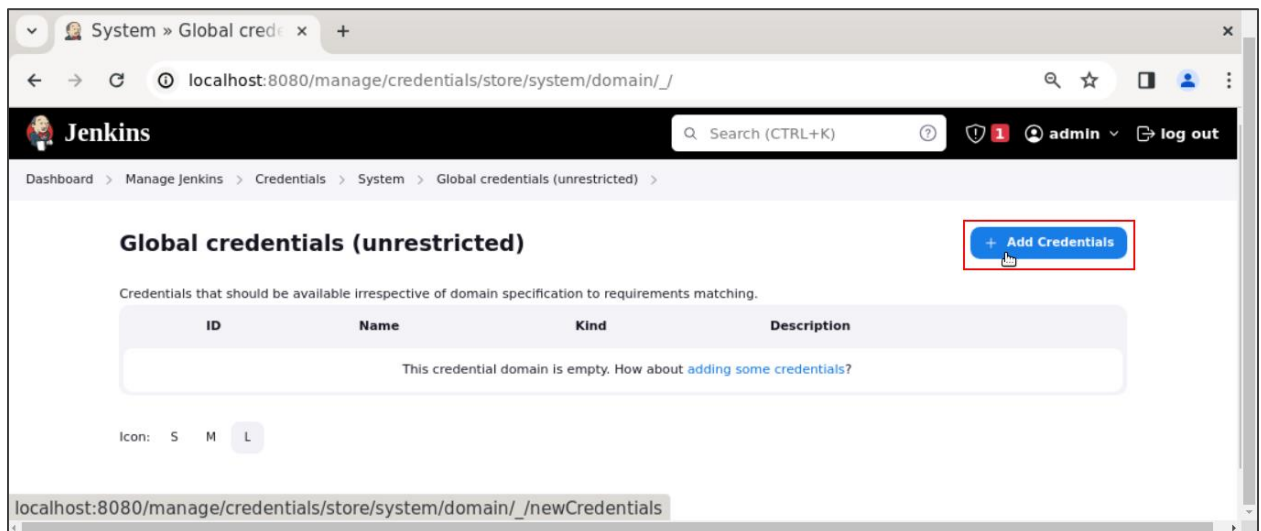


2.5 In the **Domain** section, click on **Global credentials (unrestricted)** as shown in the screenshot below:

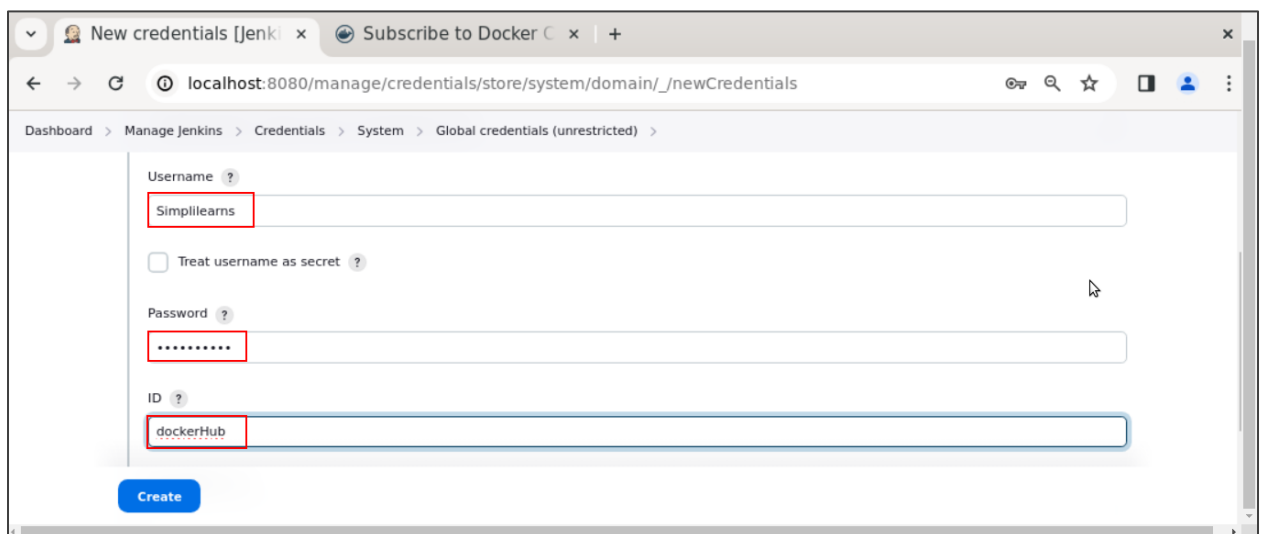




2.6 Click on the **Add Credentials** button as shown in the screenshot below:



2.7 Add the details such as **Username**, **Password**, **ID**, and **Description**, and click on the **Create** button as shown in the screenshots below:



**Note:** Add the Docker Hub username in the **Username** field and Docker Hub password in the **Password** field

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

.....

ID ?  
dockerHub

Description ?  
DockerHub Credentials

Create

REST API Jenkins 2.426.3

You can now see the successfully added credentials as shown in the screenshot below:

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

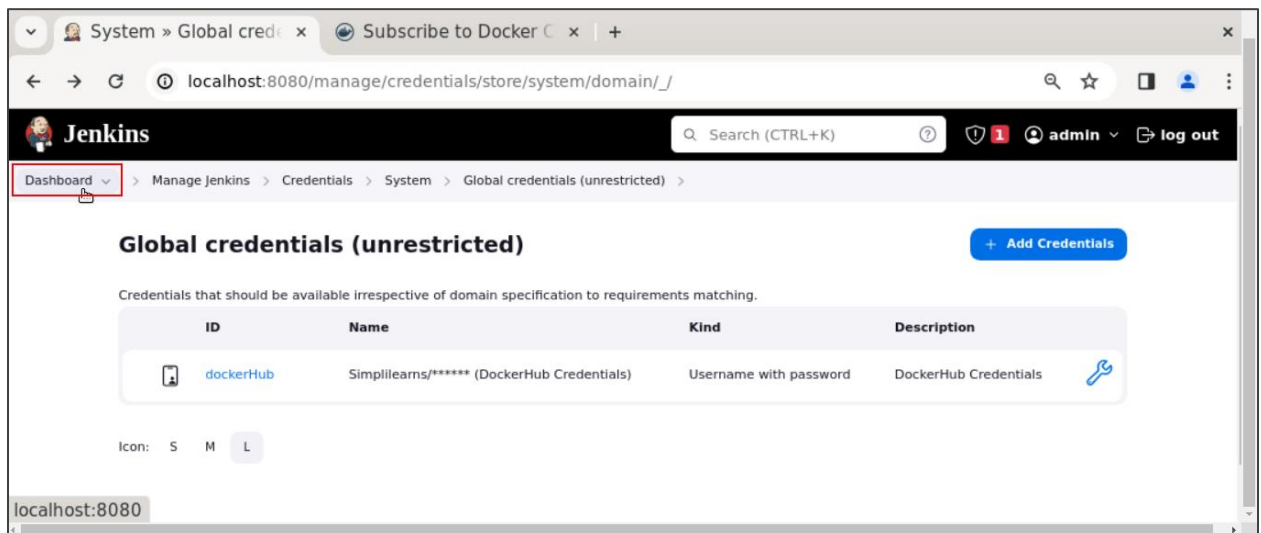
**Global credentials (unrestricted)** + Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

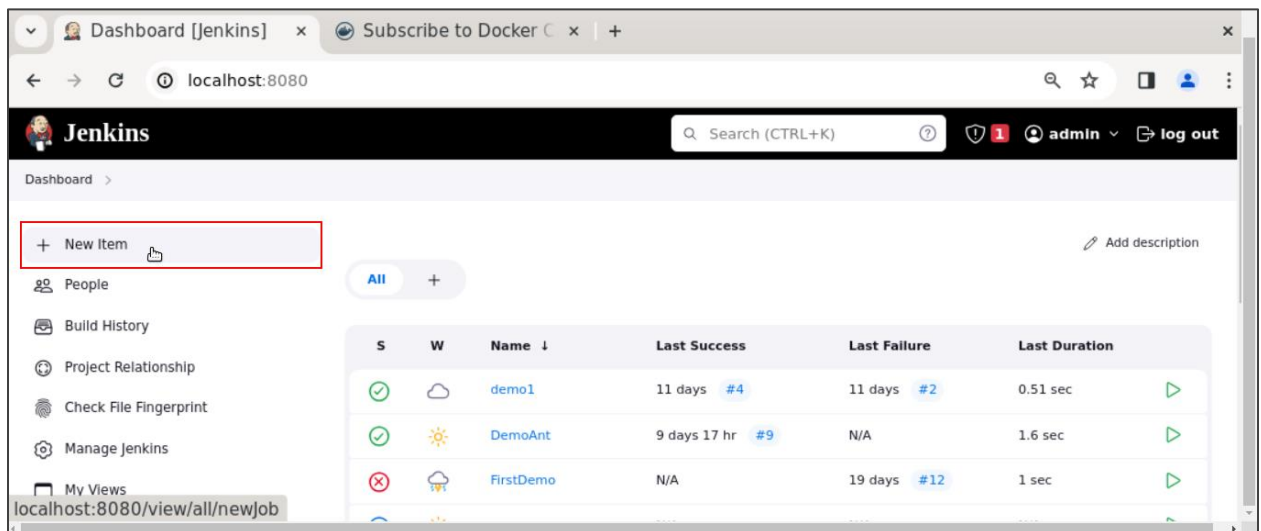
ID	Name	Kind	Description
dockerHub	Simplilearns/***** (DockerHub Credentials)	Username with password	DockerHub Credentials

Icon: S M L

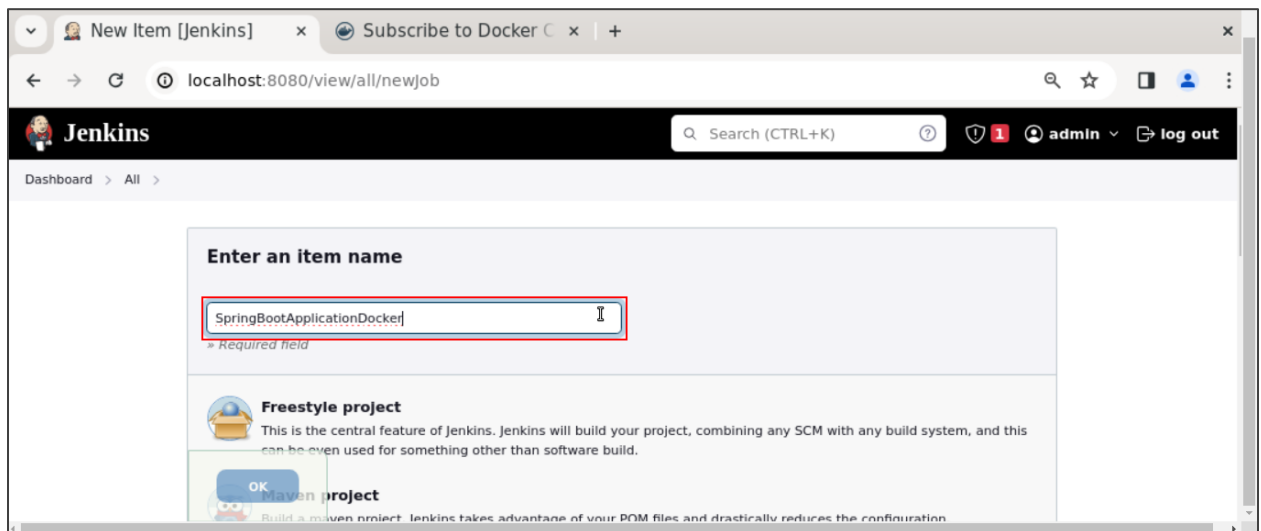
2.8 Click on **Dashboard** as shown in the screenshot below:



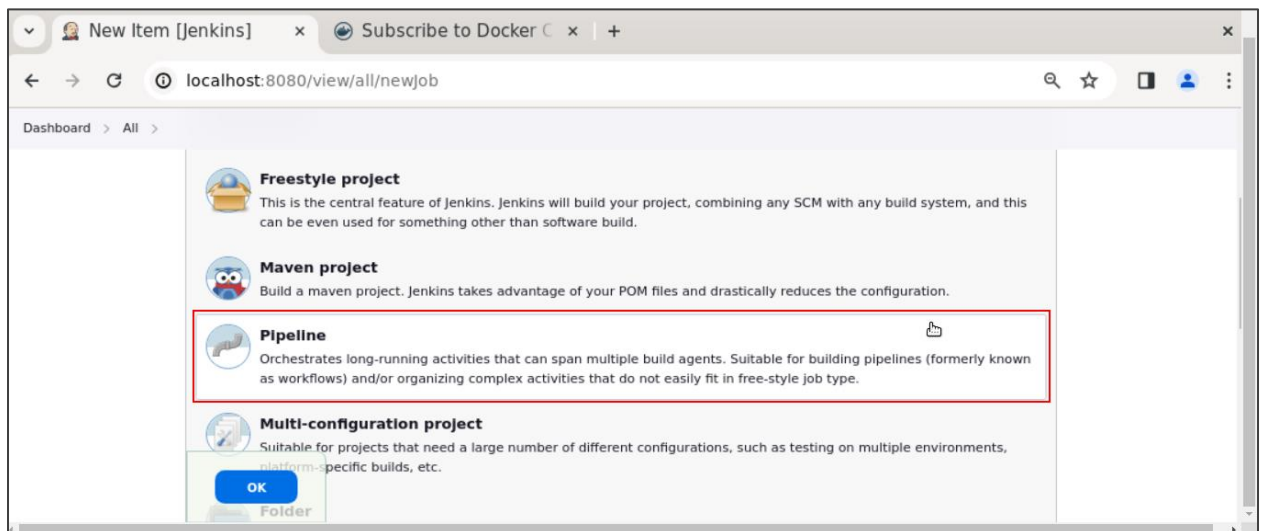
2.9 Click on **New item** to create a Jenkins pipeline job



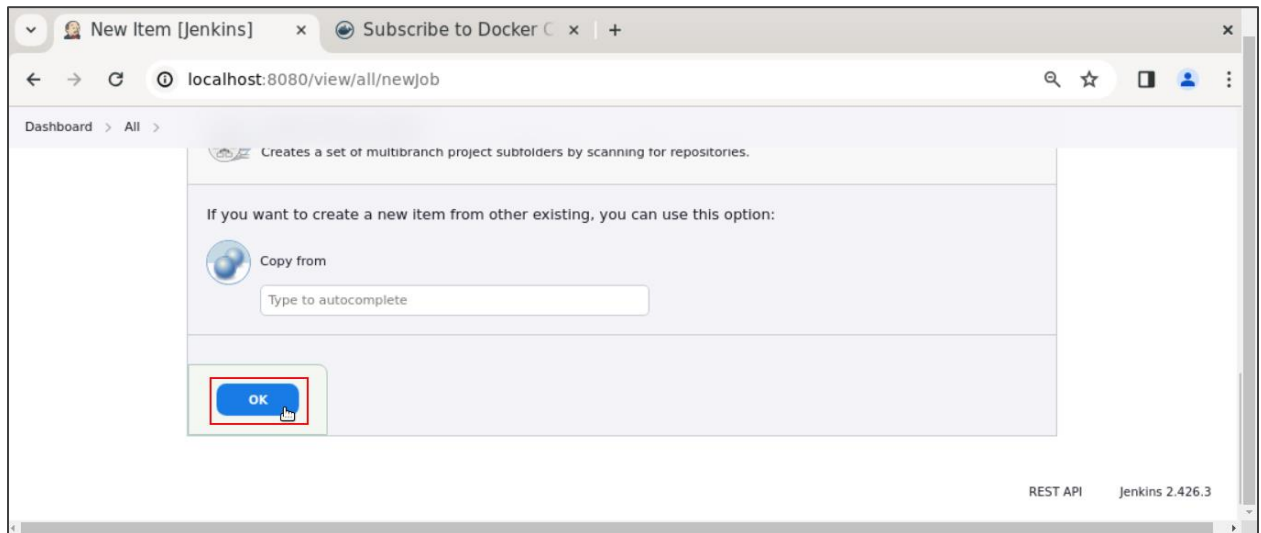
2.10 Add the name for the job as shown in the screenshot below:



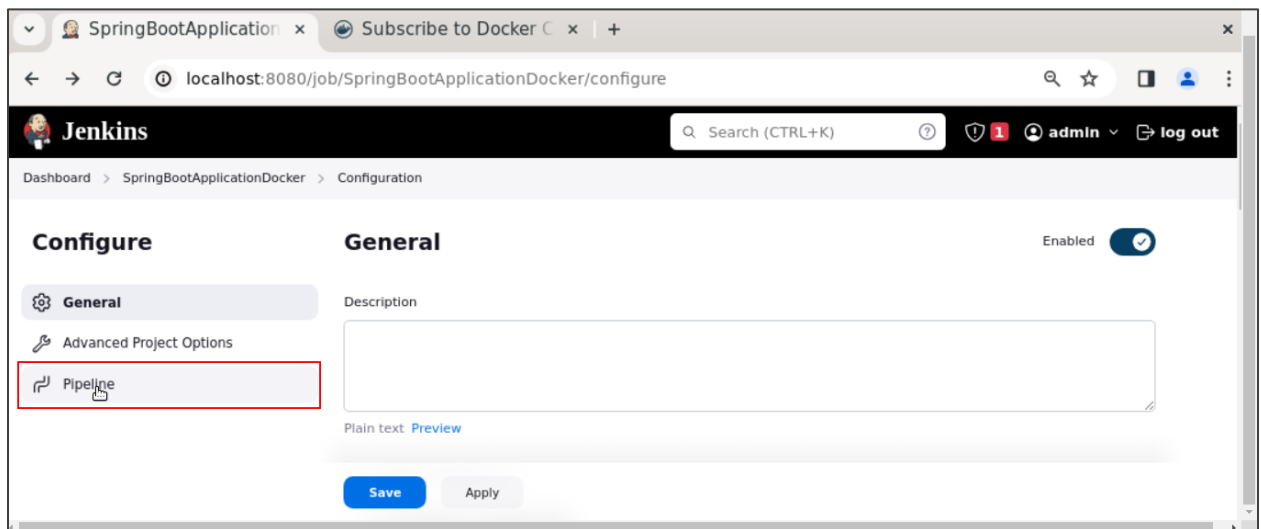
2.11 Select the **Pipeline** option as shown in the screenshot below:



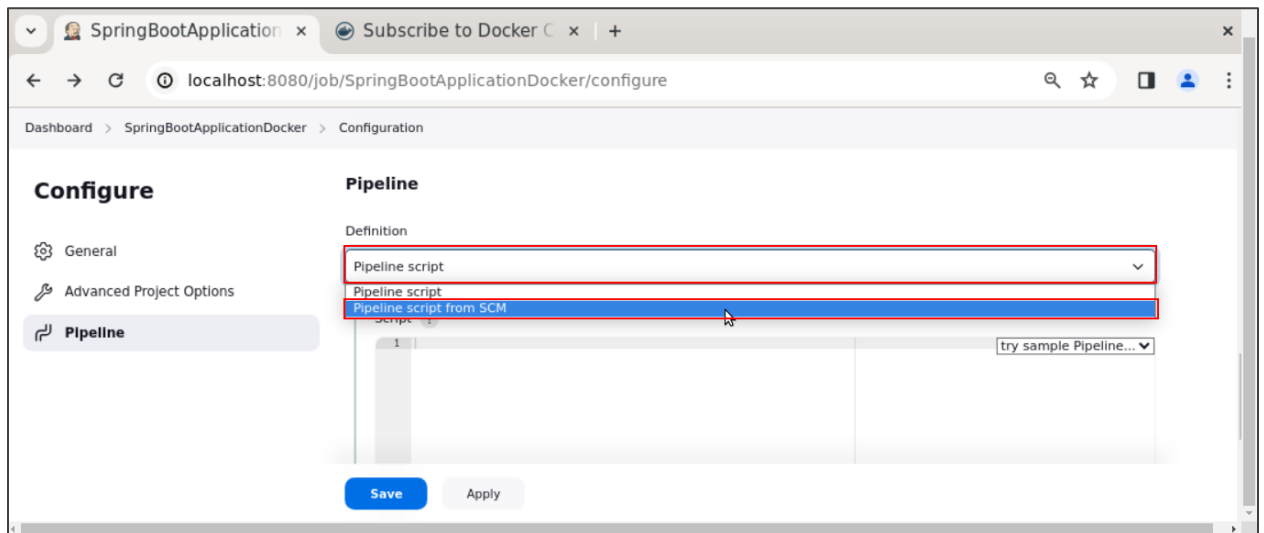
2.12 Click on the **OK** button as shown in the screenshot below:



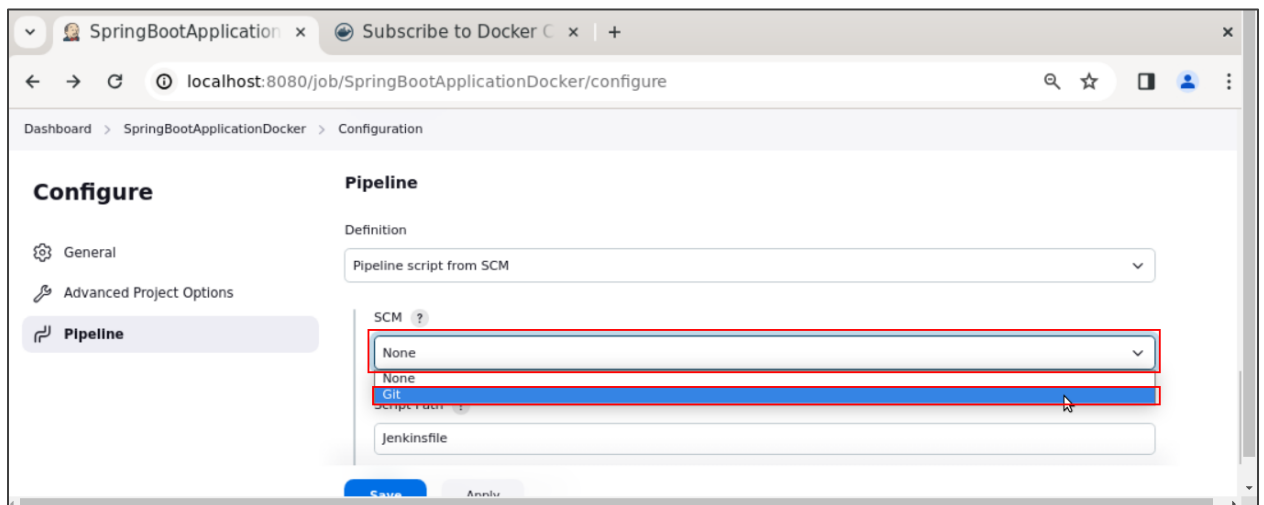
2.13 In the **Configure** section, click on **Pipeline** as shown in the screenshot below:



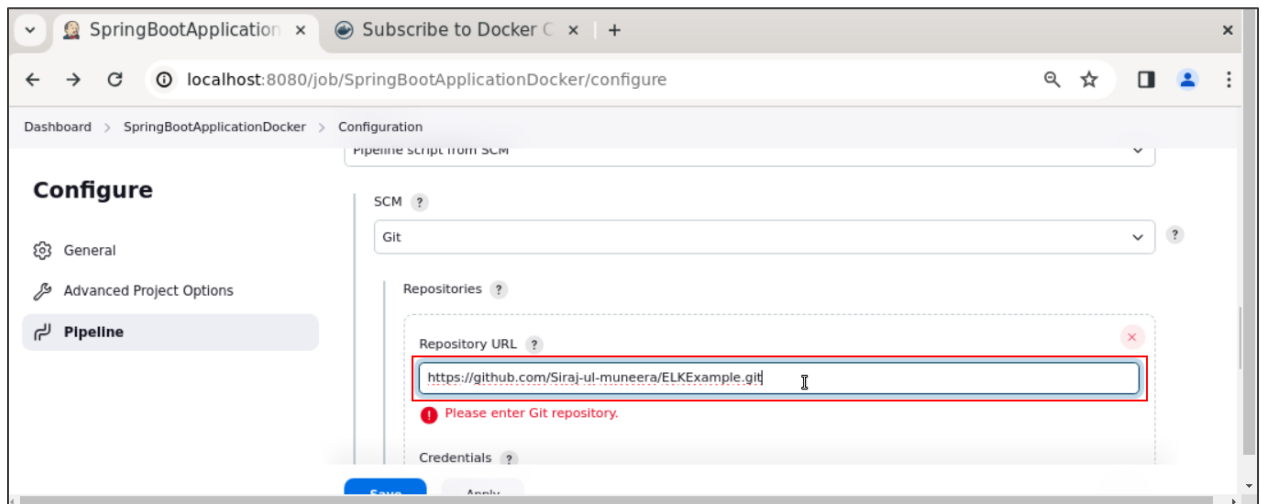
2.14 In the **Definition** section, click on the **Pipeline script** dropdown and select the **Pipeline script with SCM** option as shown in the screenshot below:



2.15 In the **SCM** section, click on the dropdown named **None** and select the **Git** option as shown in the screenshot below:

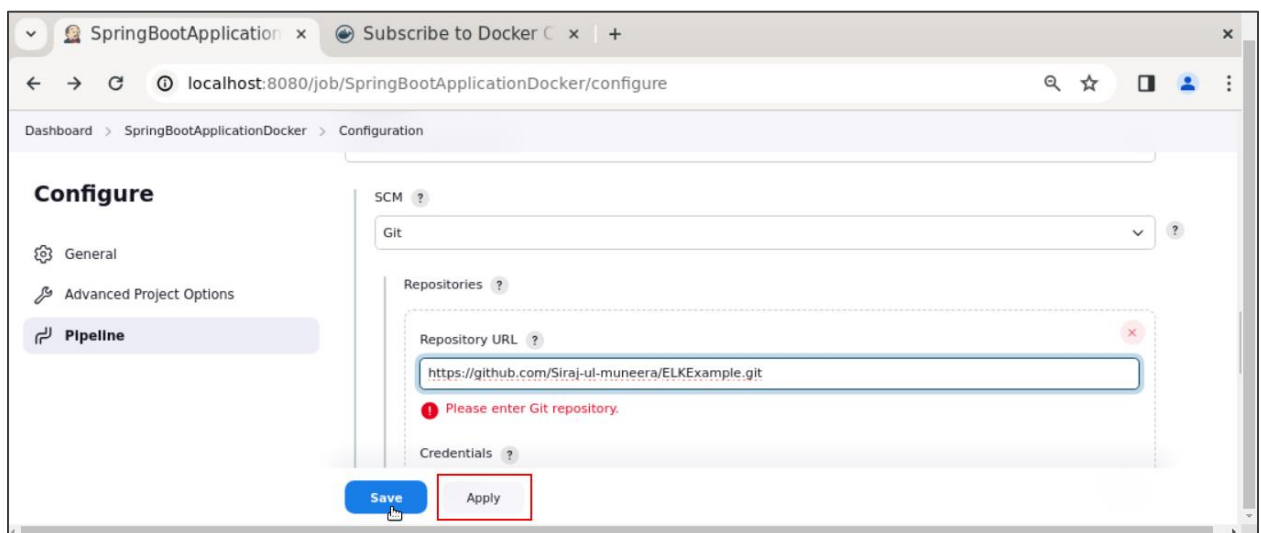


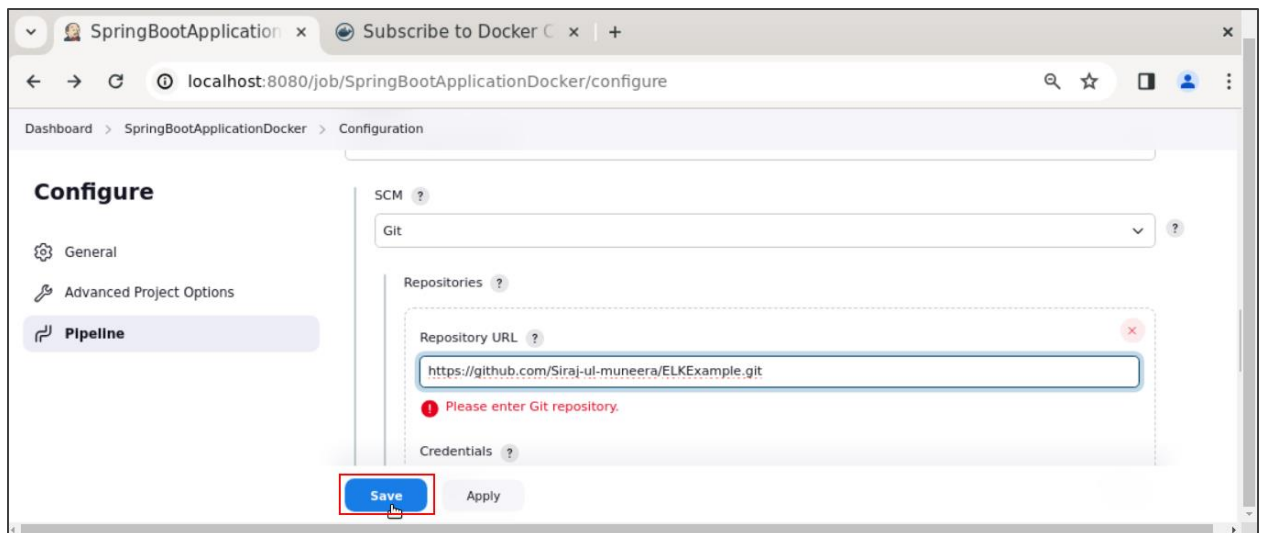
2.16 In the **Repository URL**, add the following Git repository link:  
**<https://github.com/Siraj-ul-muneera/ELKExample.git>**



**Note:** You can either use the provided repository directly or fork it into your GitHub account for customization.

2.17 Click on the **Apply** button, and then click on the **Save** button as shown in the screenshot below:

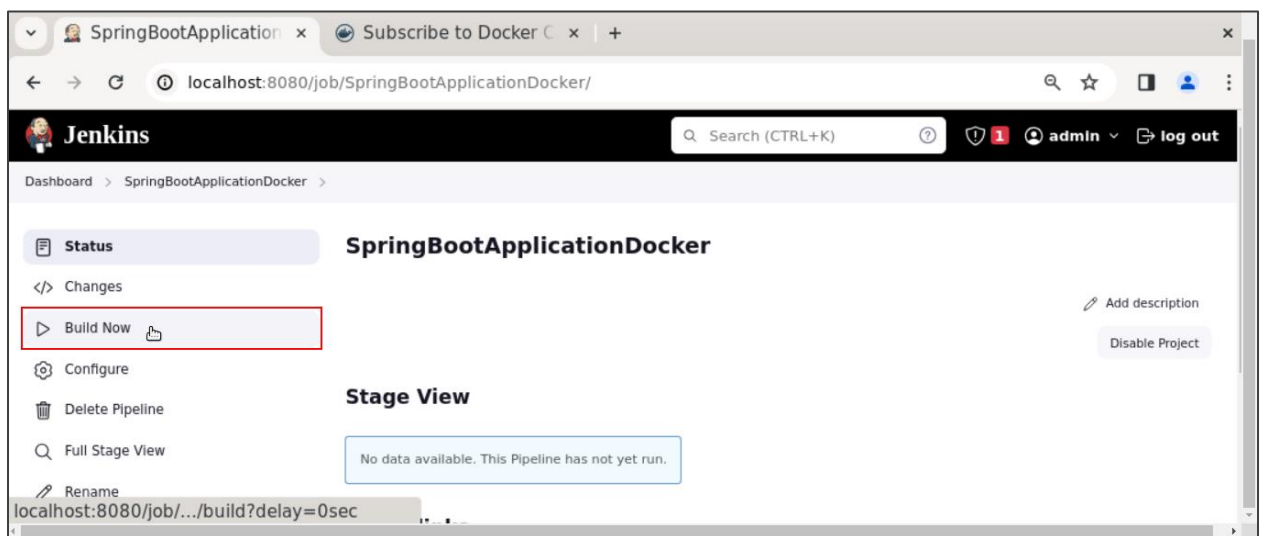




2.18 Set permissions to 777 for the Docker sock file using the following command:  
**sudo chmod 777 /var/run/docker.sock**

```
root@ip-172-31-39-225:/home/labsuser/ELKExample# sudo chmod 777 /var/run/docker.sock
root@ip-172-31-39-225:/home/labsuser/ELKExample#
```

2.19 Click on **Build Now**, and then click on **Full Stage View** to build and view the output



**Note:** Once the build and deployment are successful, you may also see the Docker container deployed on Docker host using **docker ps | grep springbootapp** command.



## Console Output

```

Started by user admin
Obtained Jenkinsfile from git https://github.com/Siraj-ul-muneera/ELKExample.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/SpringBootApponDocker
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clone repository)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> /bin/git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/SpringBootApponDocker/.git # timeout=10
Fetching changes from the remote Git repository
> /bin/git config remote.origin.url https://github.com/Siraj-ul-muneera/ELKExample.git # timeout=10
Fetching upstream changes from https://github.com/Siraj-ul-muneera/ELKExample.git
> /bin/git --version # timeout=10
> git --version # 'git version 2.32.0'
> /bin/git fetch --tags --force --progress -- https://github.com/Siraj-ul-muneera/ELKExample.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> /bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 732blacbbe38fe837aebbfba6f54a537d1c8fa29 (refs/remotes/origin/master)
> /bin/git config core.sparsecheckout # timeout=10
> /bin/git checkout -f 732blacbbe38fe837aebbfba6f54a537d1c8fa29 # timeout=10
Commit message: "Update Jenkinsfile"
> /bin/git rev-list --no-walk 732blacbbe38fe837aebbfba6f54a537d1c8fa29 # timeout=10
    
```

Activate Windows  
Go to Settings to activate Windows

**Note:** If Console output is a **failure**, follow the below mentioned steps:

- Log in to your GitHub account
- Fork the repository <https://github.com/Siraj-ul-muneera/ELKExample.git>
- Open **Jenkinsfile** and replace **dockerhubaccountid** with your Docker Hub **Username**
- Save and copy the URL
- Paste the URL while creating Jenkins pipeline job

## Step 3: Run the Spring Boot application and verify the logs in Kibana


3.1 Access the Spring Boot web application and perform an activity to generate logs for the ELK stack using the <http://localhost:81> link

Id	Product Id	Description	Price	View	Edit
1	235268845711068308	Spring Framework Guru Shirt	18.95	<a href="#">View</a>	<a href="#">Edit</a>

3.2 Navigate to the Kibana dashboard, click on **Management**, and then select **Index Management** from the left navigation bar to view the logs pushed to the ELK stack

Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. [Read more](#)

Yes No

 **Elasticsearch**  
[Index Management](#)  
Index Lifecycle Policies  
Rollup Jobs  
Cross Cluster Replication  
Remote Clusters  
License Management  
7.0 Upgrade Assistant

### Index management

Update your Elasticsearch indices individually or in bulk.

☐ Include rollout indices
☐ Include system indices

<input type="checkbox"/> Name	Health	Status	Primaries	Replicas	Docs count	Storage size
<input type="checkbox"/> logstash-2021.03.29	● yellow	open	5	1	53	134.2kb

Rows per page: 10 ▾

By following these steps, you have successfully implemented continuous monitoring on Docker with ELK stack for streamlined log management and analysis.