**Lesson 9 Demo 2**

**Continuous Monitoring on Docker with ELK Stack**



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 check the logs in Kibana

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



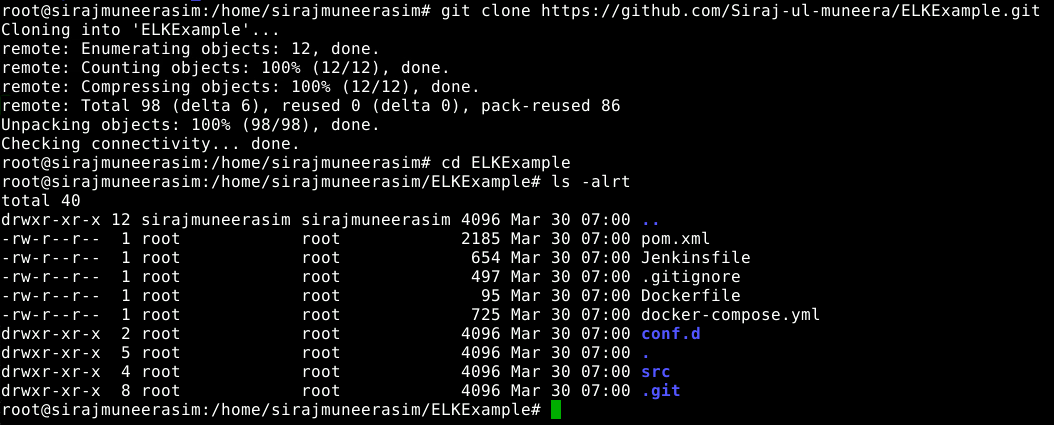
1. Download Docker compose file in one of the git repositories and follow the set of commands given below to initialize the ELK stack.

***sudo su***

***git clone https://github.com/Siraj-ul-muneera/ELKExample.git***

***cd ELKExample***

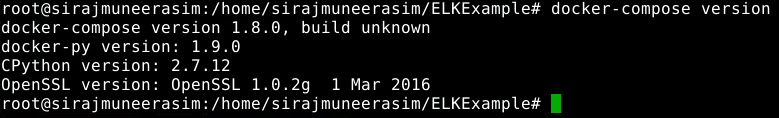
***ls -alrt***



1. Start the ELK stack using the docker-compose command. Usually, this binary is not installed on a server. So, follow the set of commands given below to install Docker Compose.

***apt install docker-compose***

***docker-compose version***



1. Before starting the ELK stack, run the command given below so that elastic search is configured properly.

***sysctl -w vm.max\_map\_count=262144***

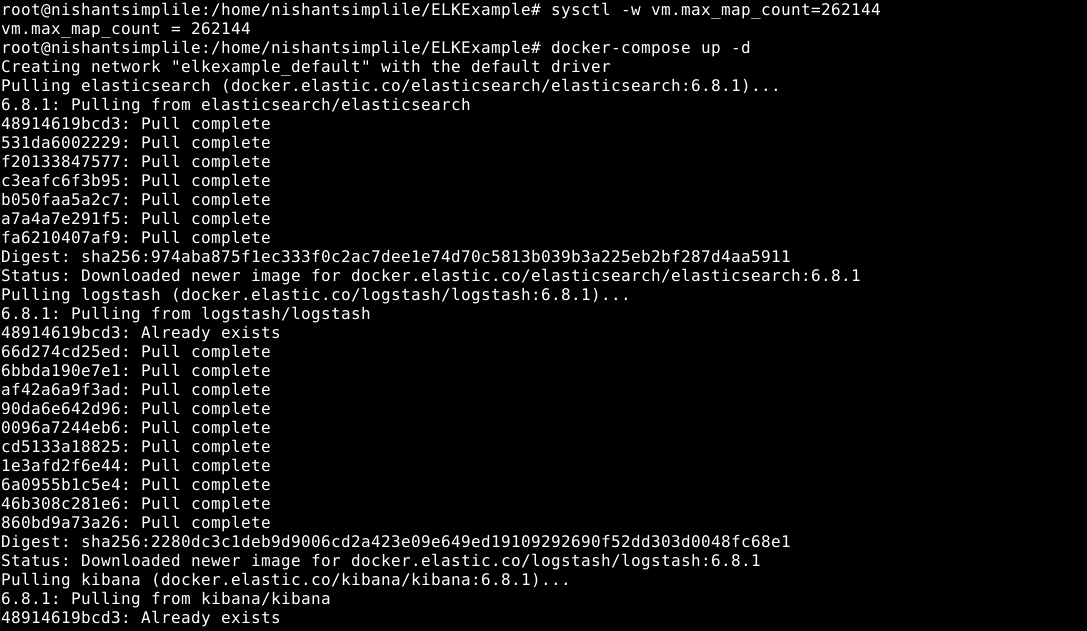
1. Run the docker-compose command to initialize the ELK stack.

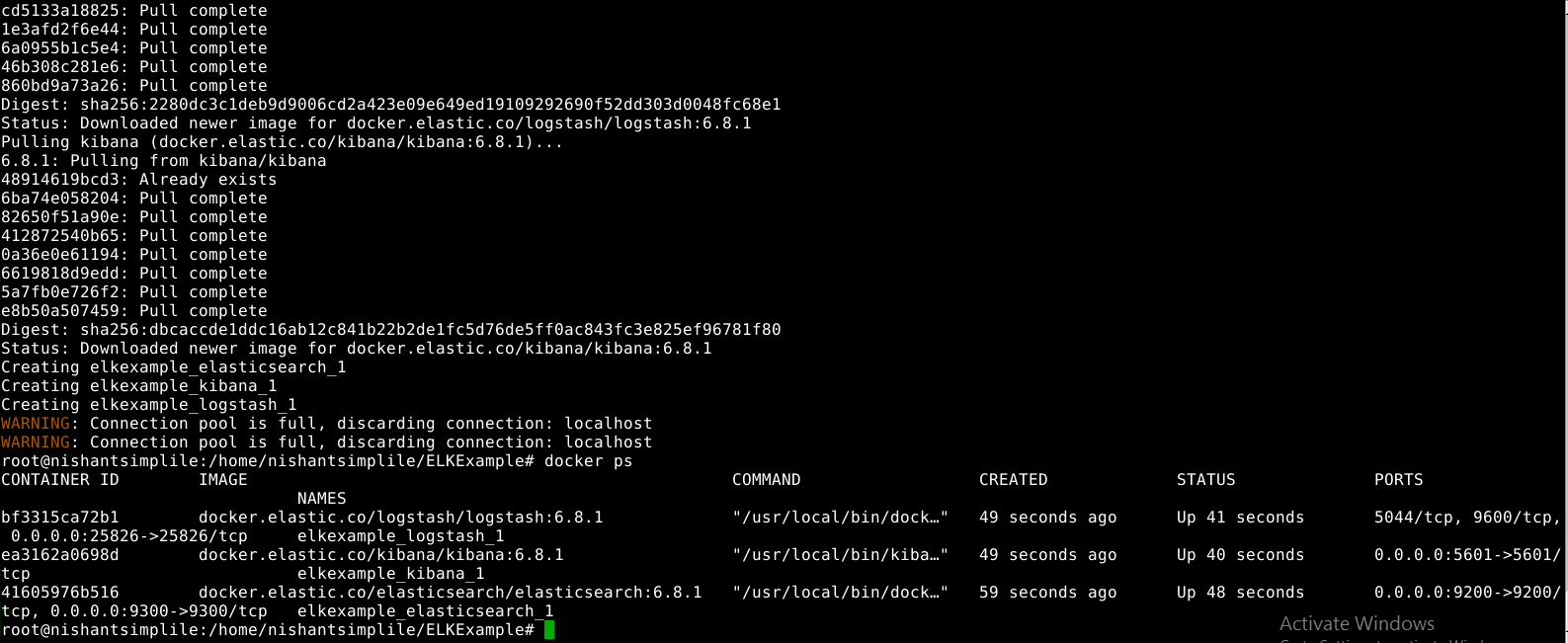
***sudo systemctl unmask docker***

***wget -qO- https://get.docker.com/ | sh  
  
sudo service docker start***

***docker-compose up -d***

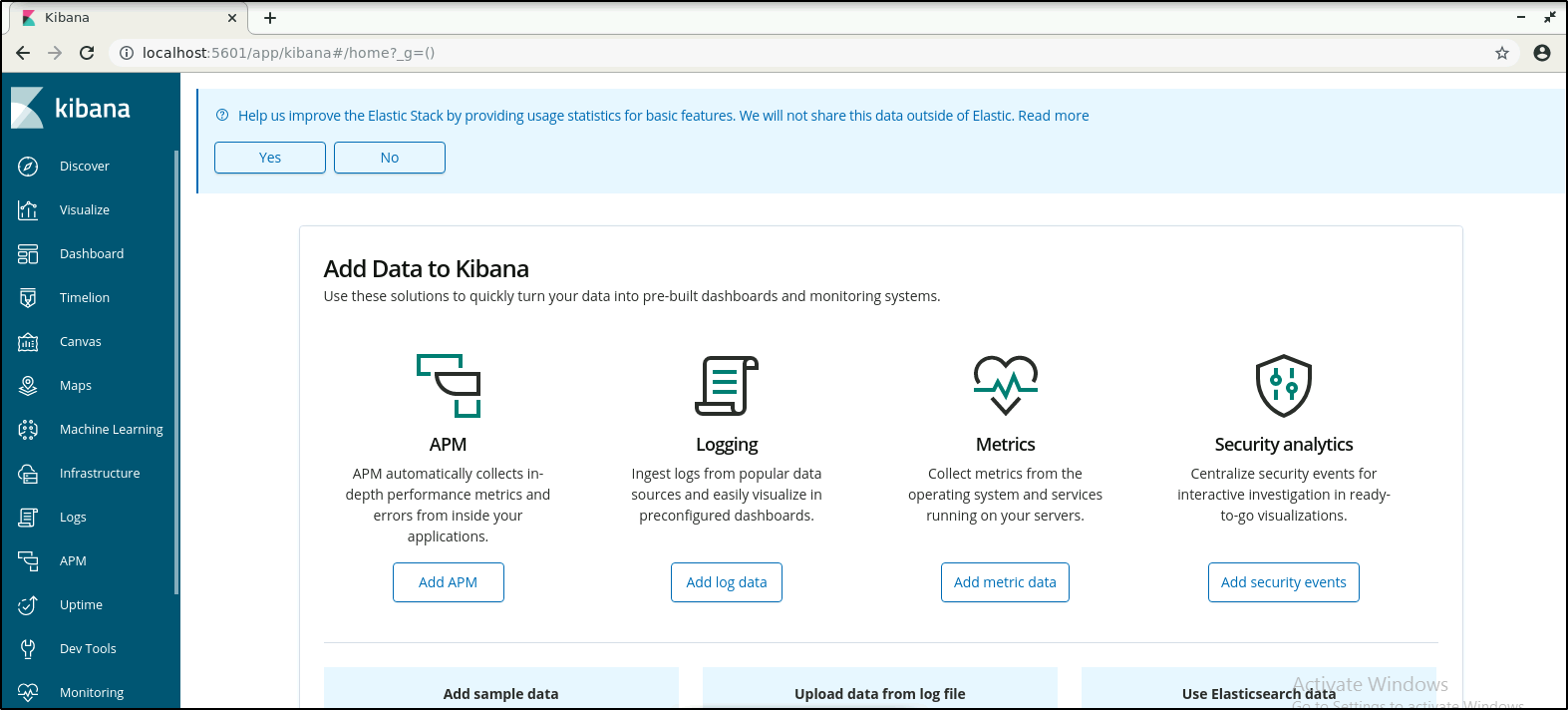
***docker ps***





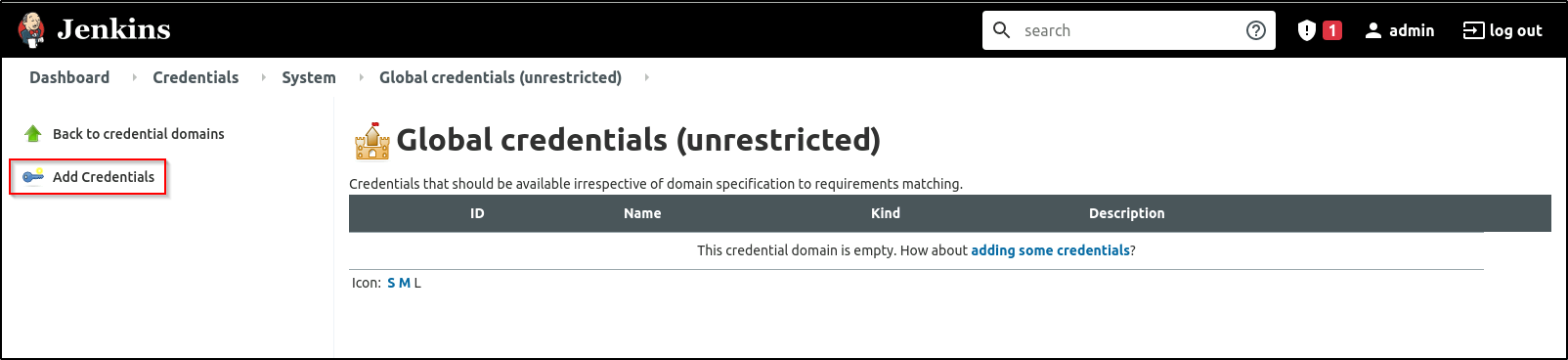
1. Open the Kibana URL using the public IP of the host and 5601 port to access the Kibana dashboard.

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



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

1. From the browser, navigate to <http://localhost:8080> and login to Jenkins.
2. Configure your Docker hub credentials in Jenkins. Go to *Manage Jenkins* -> *Manage Credentials* -> click on *Jenkins* link -> click on *Global credentials (unrestricted)* -> click on *Add Credentials* from the left pane.



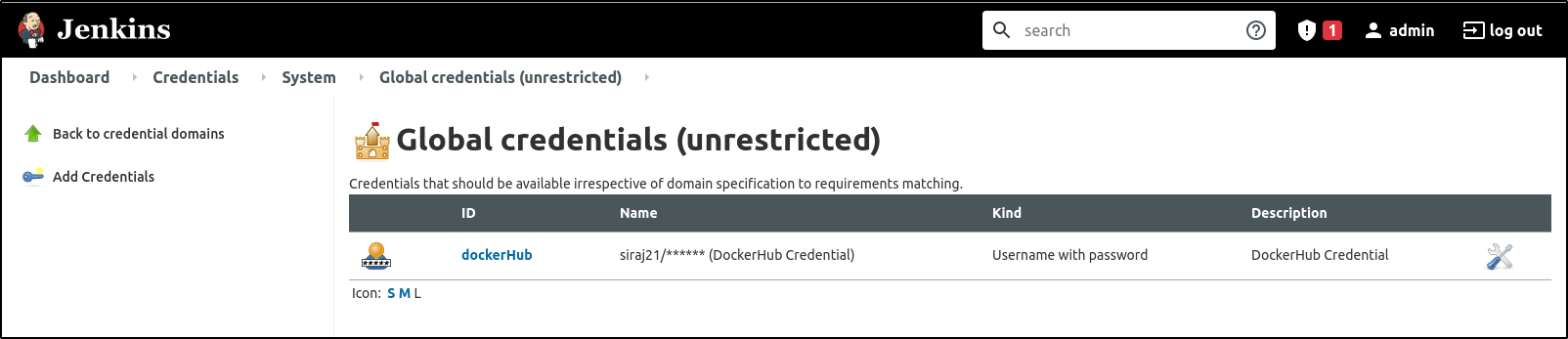
1. Add the details as shown below

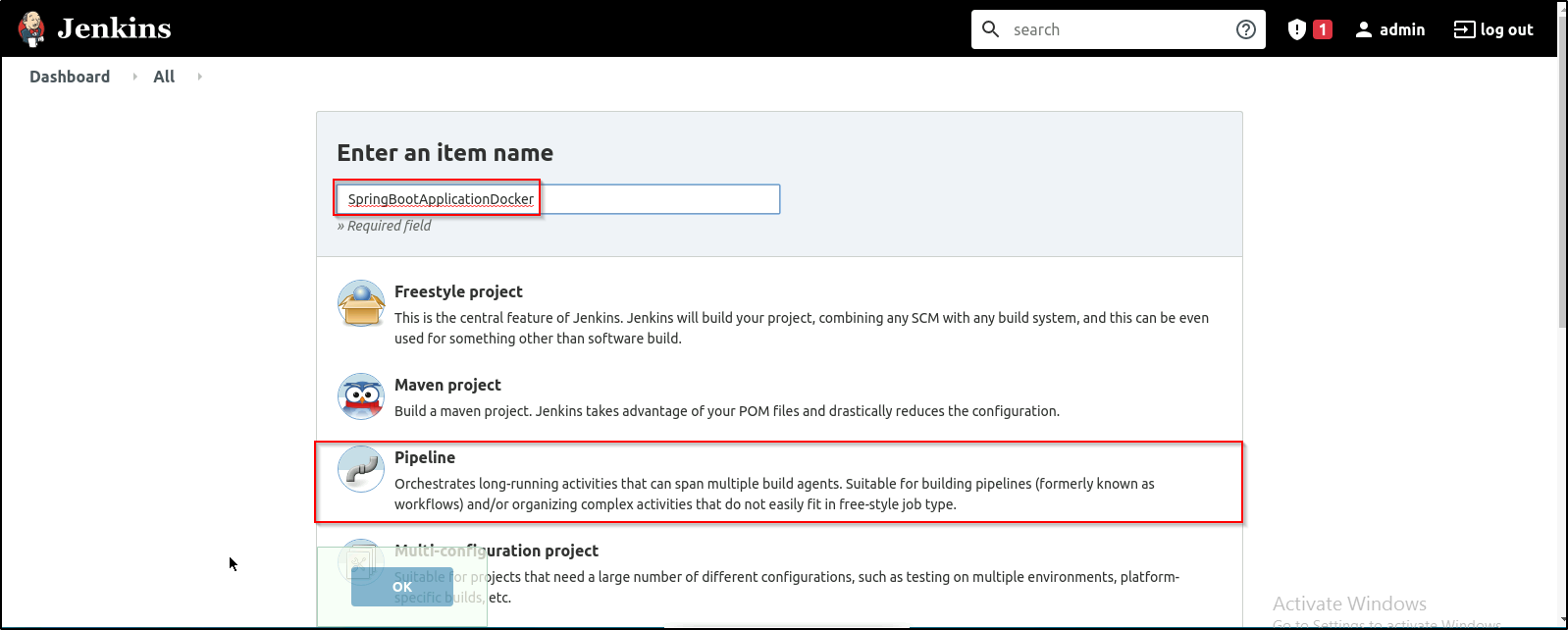
Username: *<Your\_DockerHub\_Username>*

Password: *<Your\_DockerHub\_Password>*



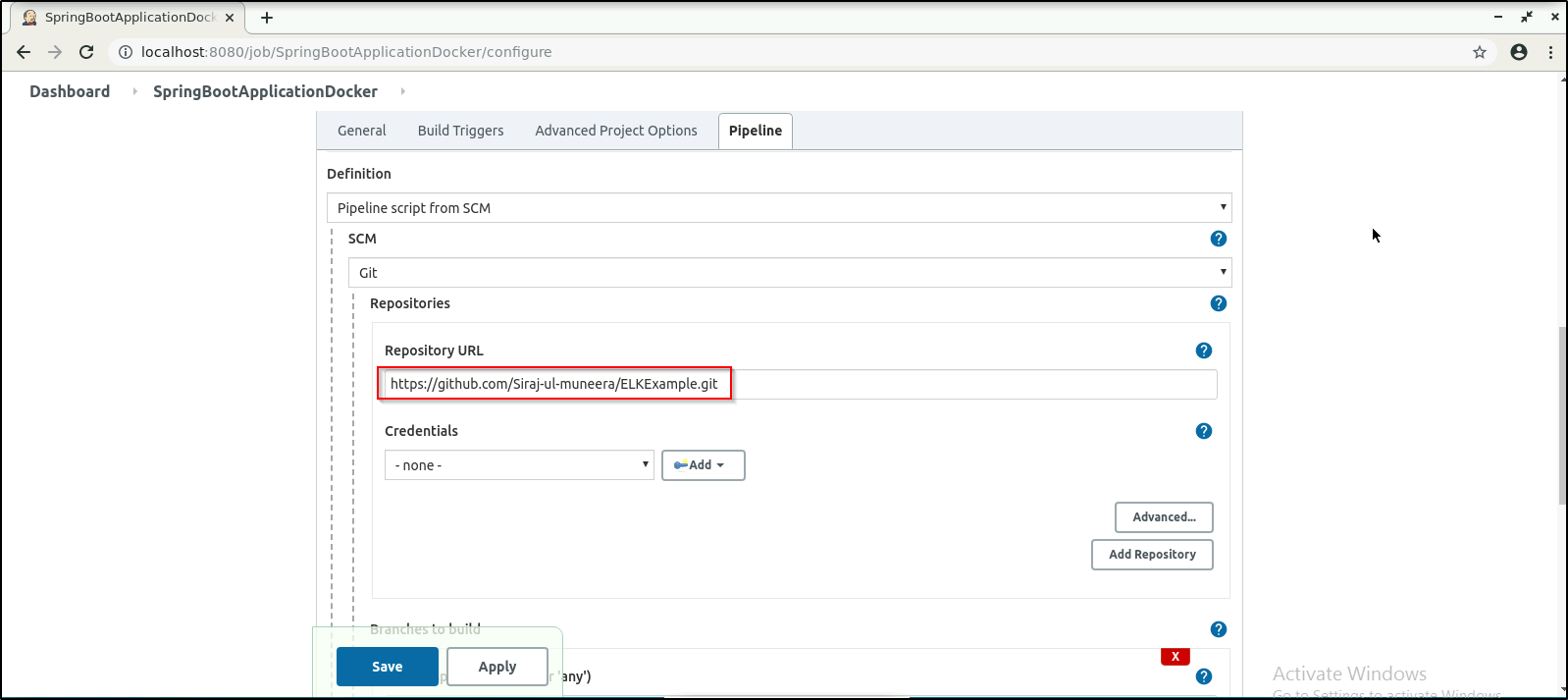
1. You should now see the credentials saved as shown below



1. Create a Jenkins pipeline job to fetch Jenkinsfile from the URL mentioned below. 
2. You can either use the below git repository or Fork it in your Github account and use it

***https://github.com/Siraj-ul-muneera/ELKExample.git***

1. Configure the job as shown in the screenshot below and then run the build.



1. Give 777 permission to the Docker sock file since we are running Docker command from a Jenkins user.

***chmod 777 /var/run/docker.sock***



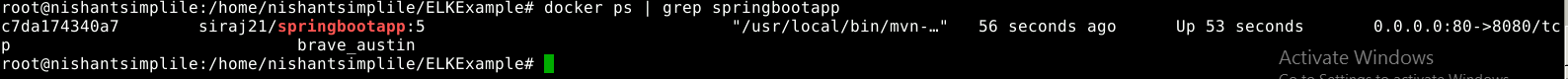
1. Build the Jenkins job to deploy the Docker container on the Docker host.
2. Jenkins pipeline will complete the build and the deployment process for the Spring Boot application





### We can see the Docker container deployed on the Docker host using the command:

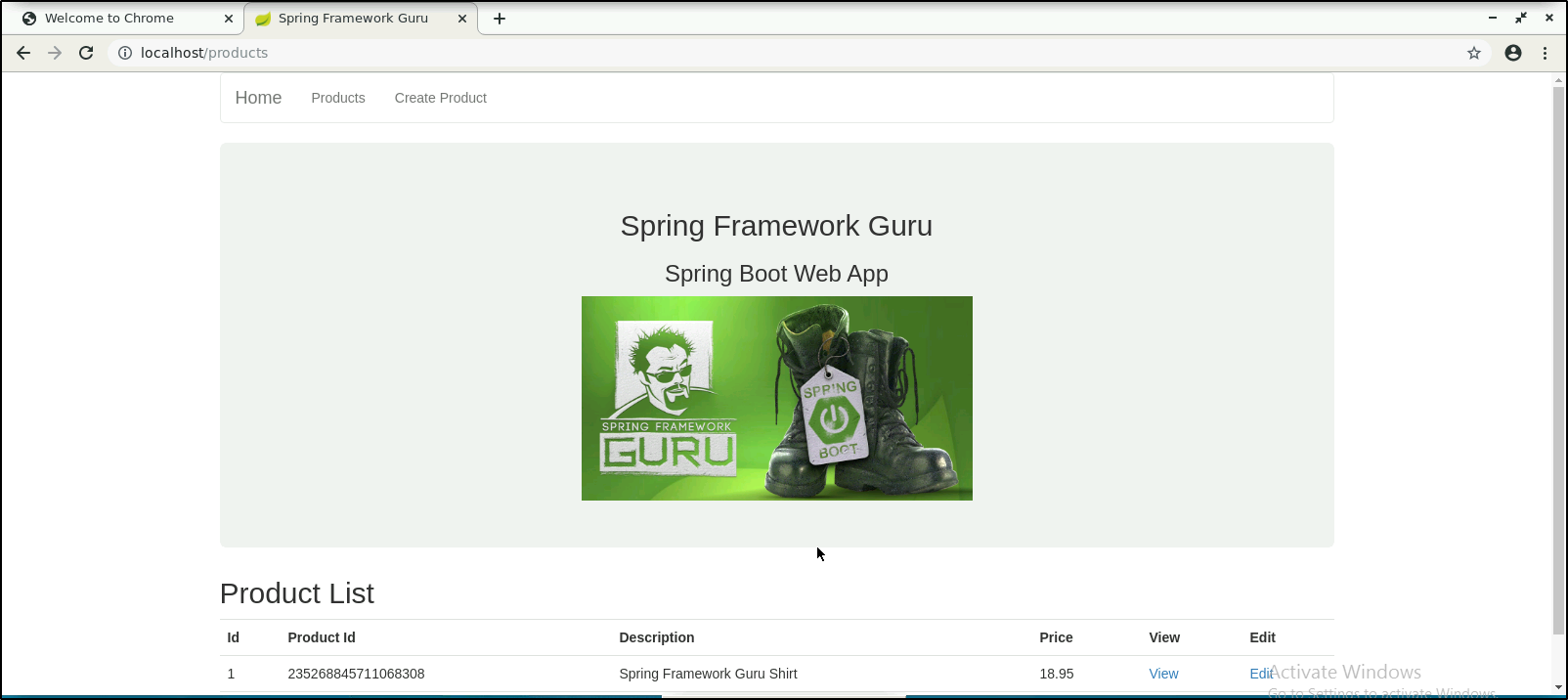
***docker ps | grep springbootapp***



**Step 3: Run the Spring Boot application and check the logs in Kibana**

1. Access the Spring Boot web application and perform some random activity so that the logs will be pushed to ELK stack.

***http://localhost:81***



1. Check the logs pushed to ELK stack in Kibana.
2. Navigate to the Kibana dashboard. Select *Index* *Management* from the navigation bar on the left. You can see the logs created.

