**Jenkins Assignment -01**

=================================================================================================================================

1. Install Jenkins on one of the server on your setup (Local VM / Ec2 / WSL).

Once Jenkins is installed , try install plugins with and without restart option.

Please verify plugins installed successfully.

Solution:-create AWS ec2 then create SG. Add inbound rule , and then select **SSH**. instance and connect.

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

sudo yum upgrade

sudo dnf install java-17-amazon-corretto -y

sudo yum install jenkins -y

sudo systemctl enable jenkins

sudo systemctl start jenkins

sudo systemctl status jenkins

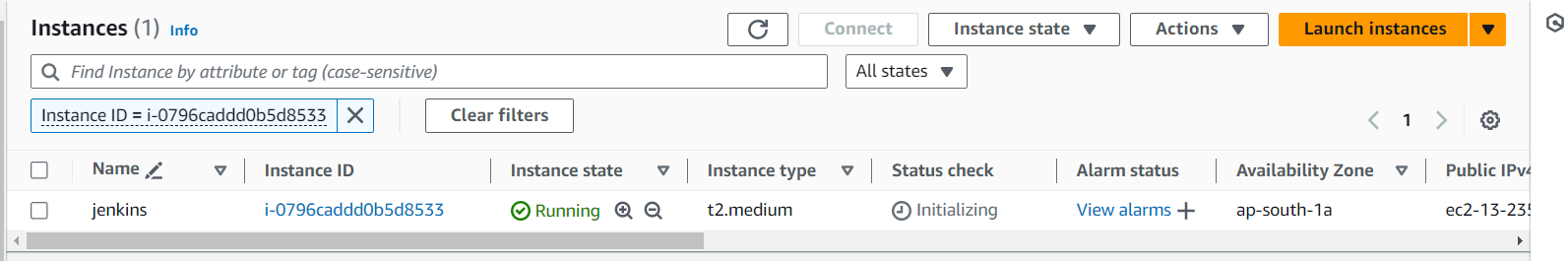
Connect to http:// public DNS:8080 from your browser

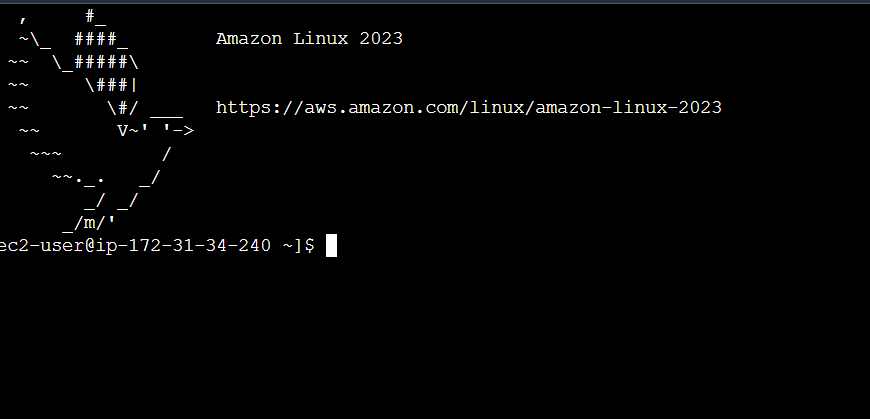
After that put the password from

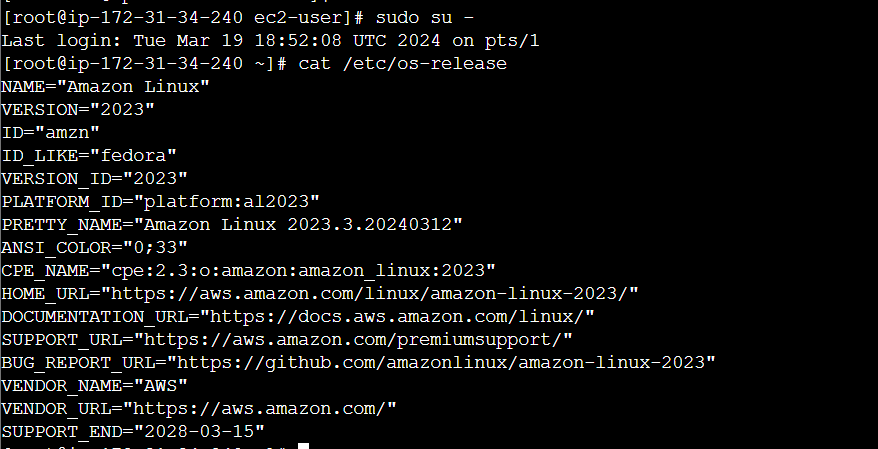
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

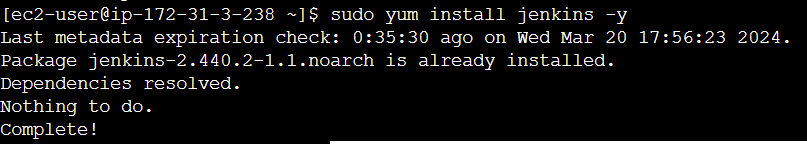
practical

------------



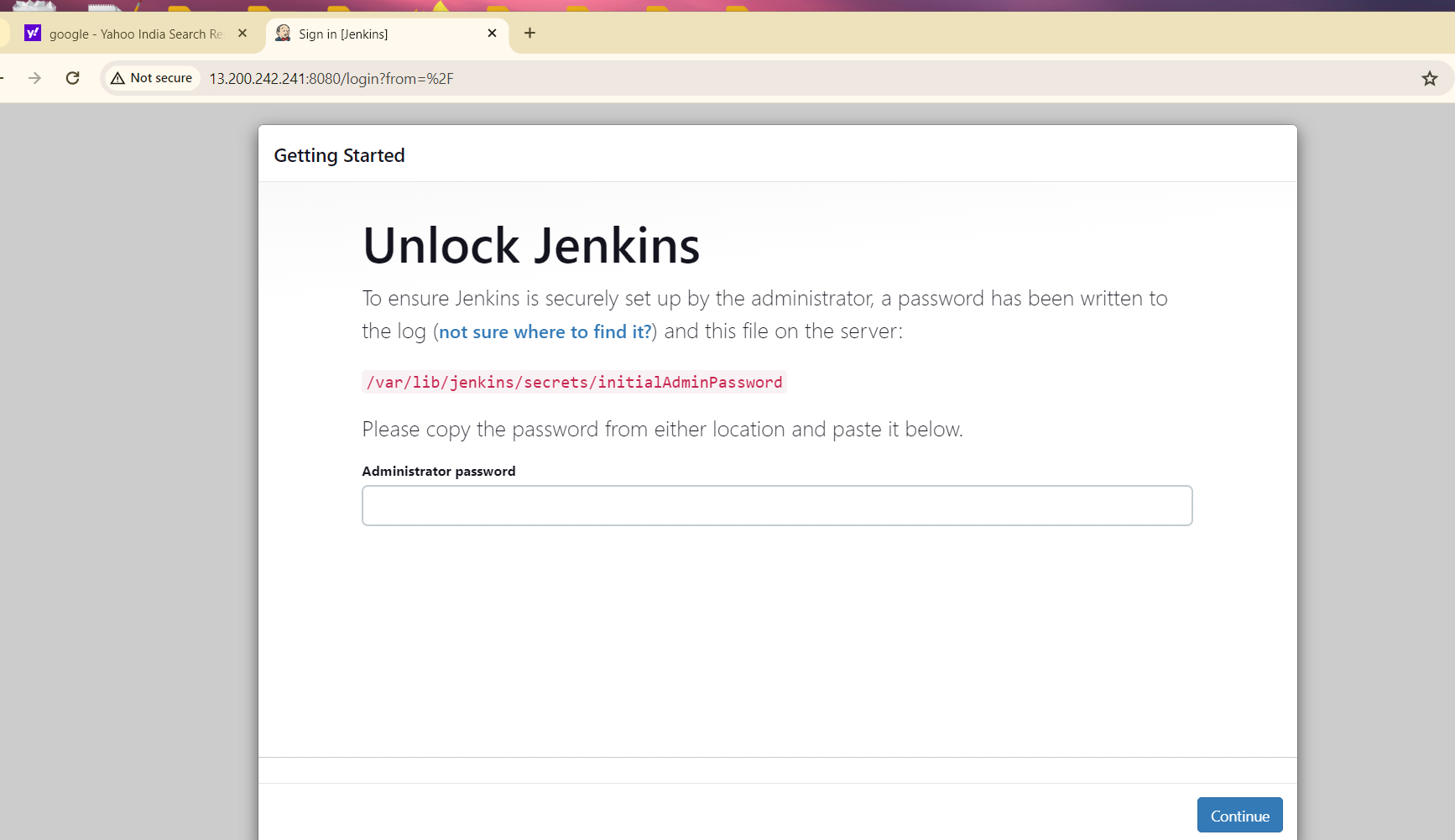


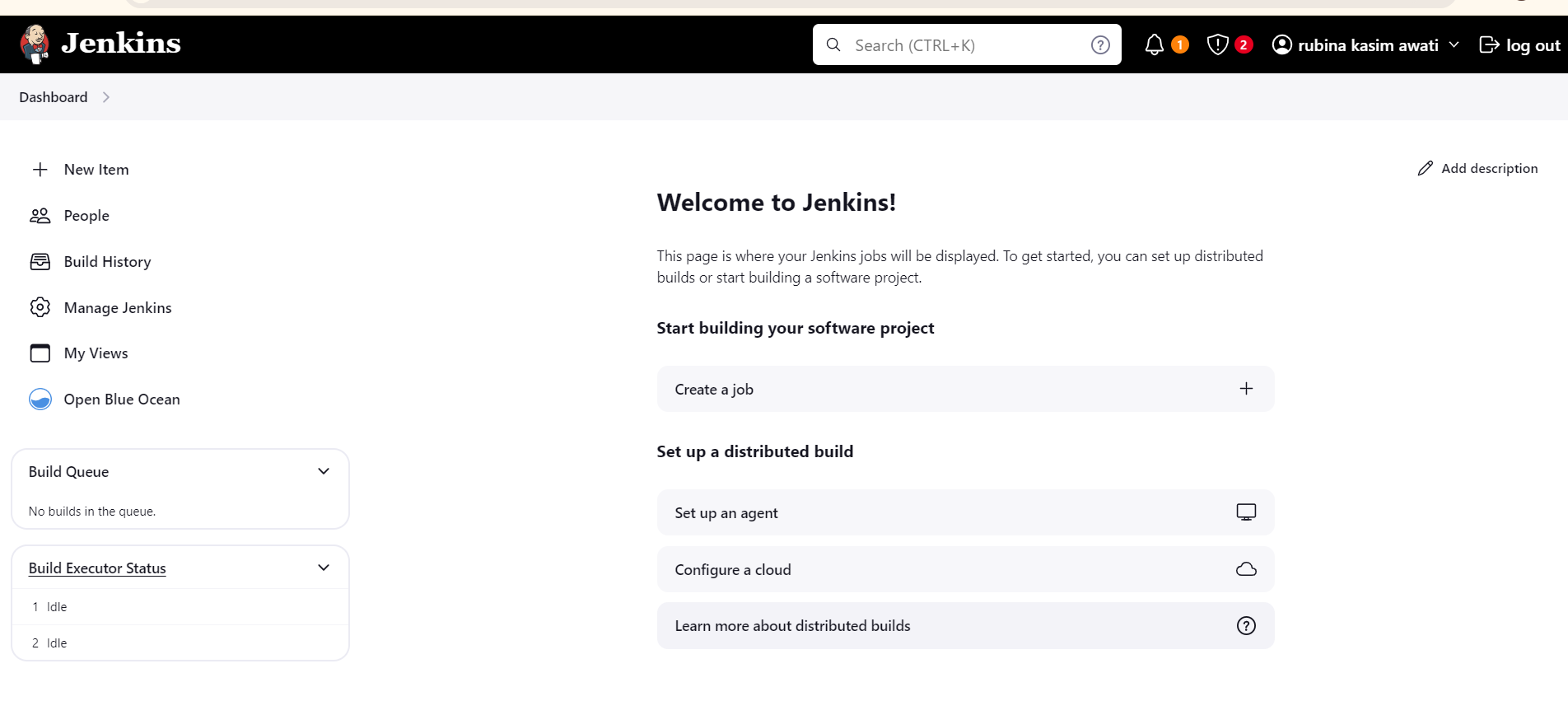










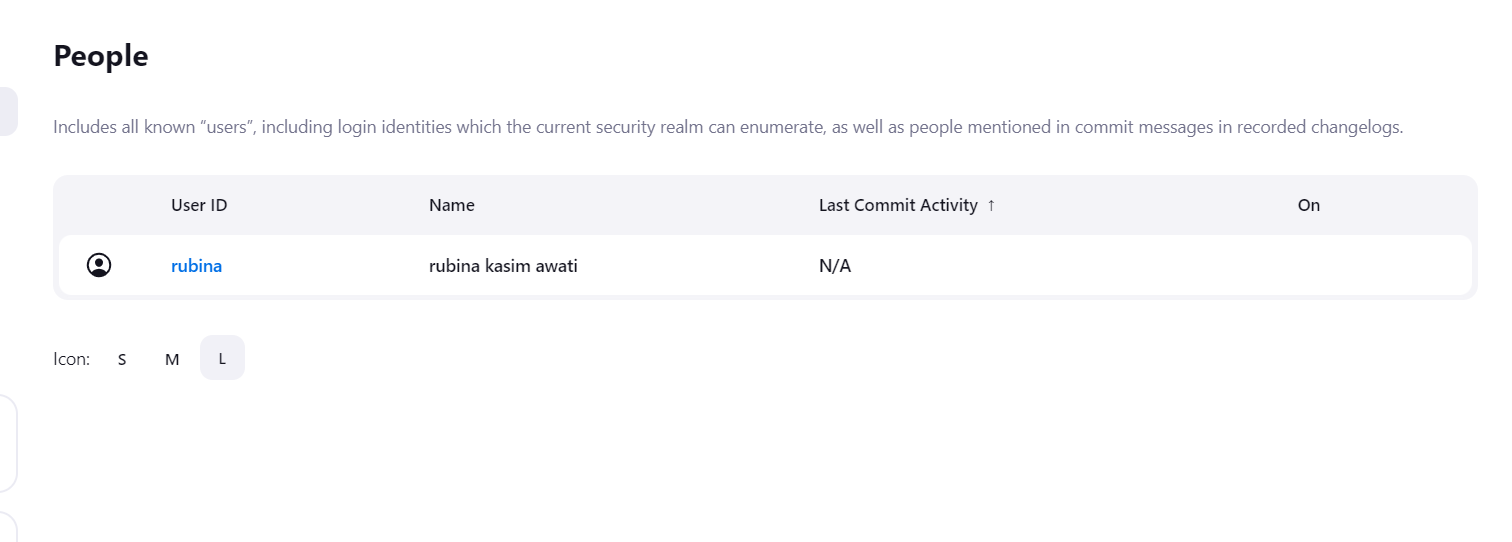


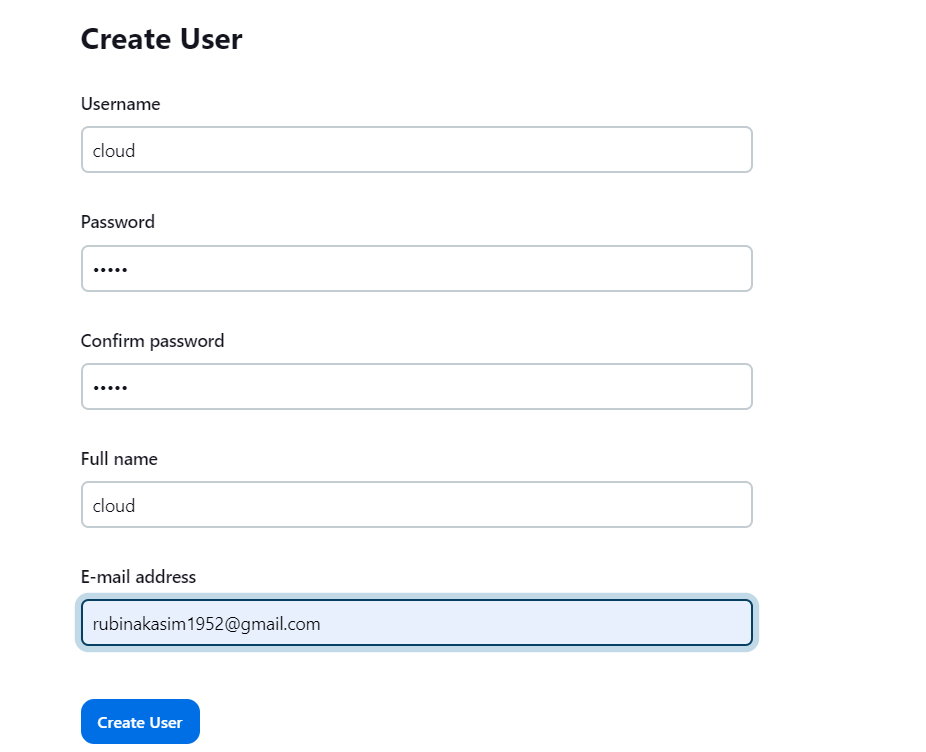
=================================================================================================================================

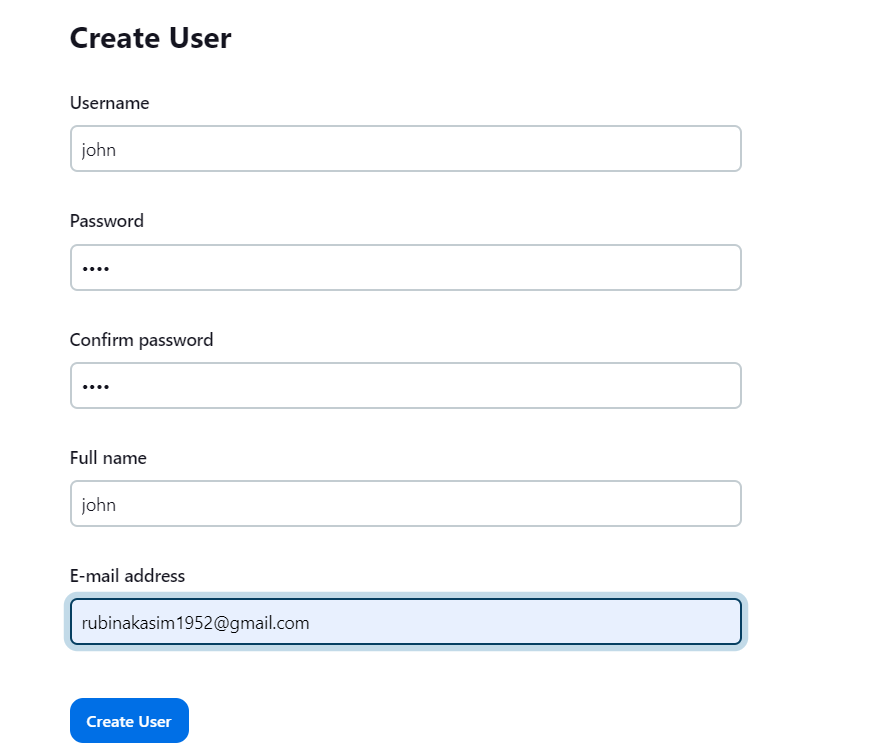
2. Create any 4 local Jenkins users on your Jenkins server. Also create 2 Jenkins roles named developers & delivery.

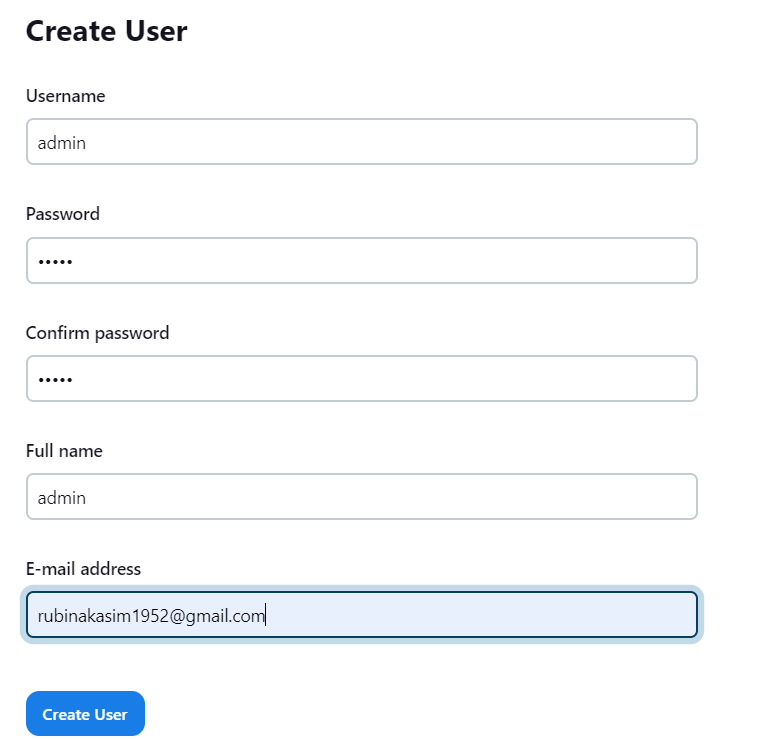
Once roles are created, assign developers role to 3 users and delivery role to project Manager user.

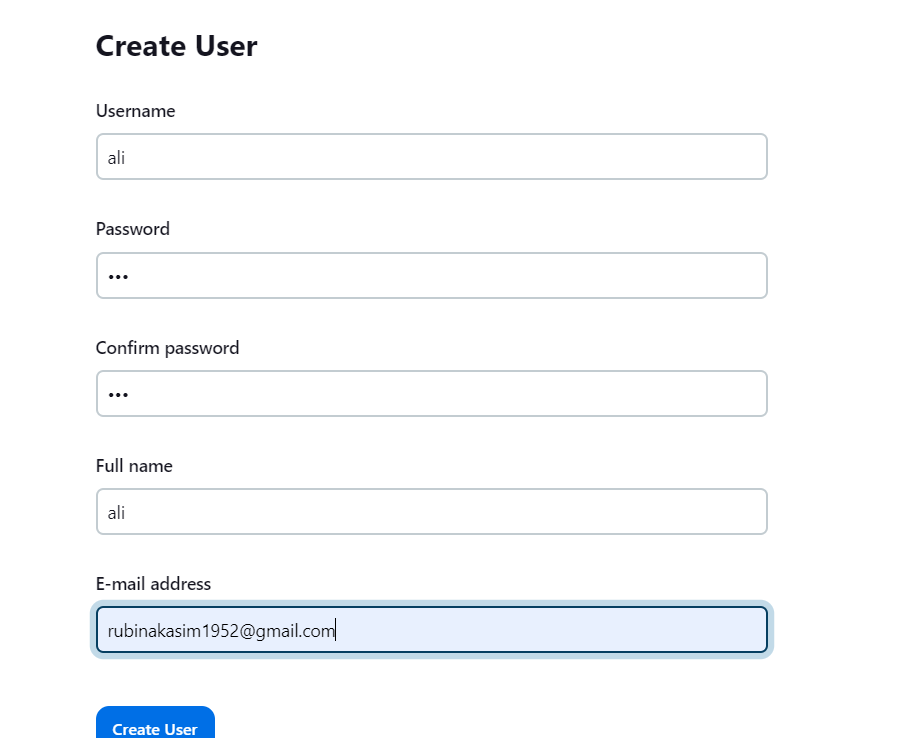
Please take screenshots and prepare well formatted document of your understanding.

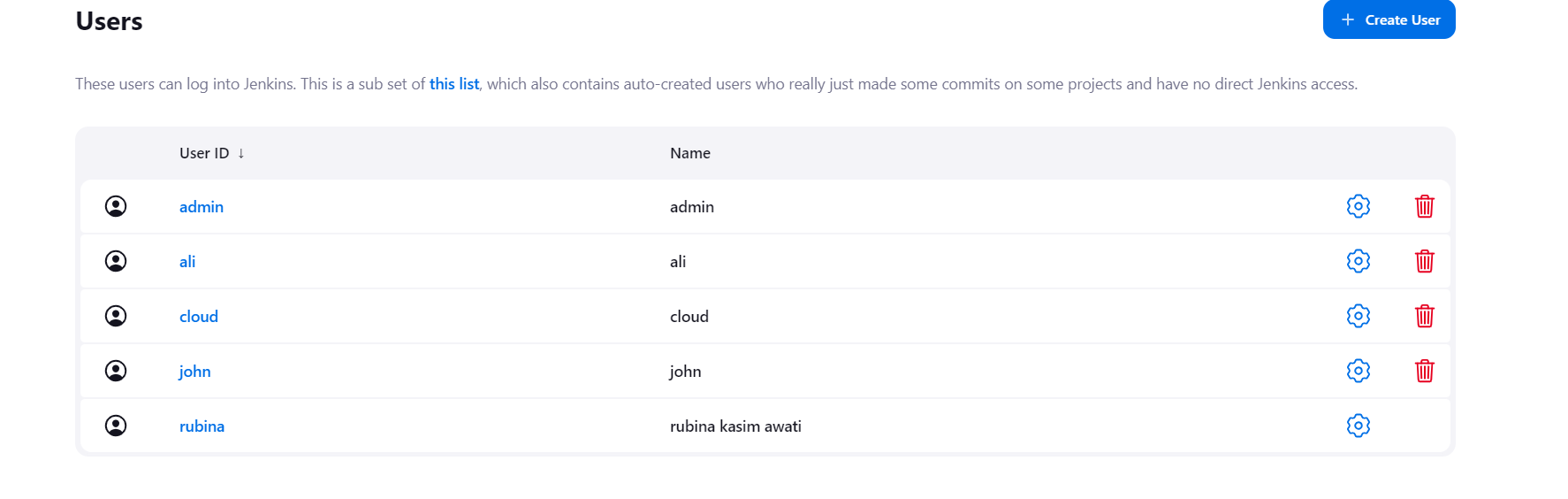


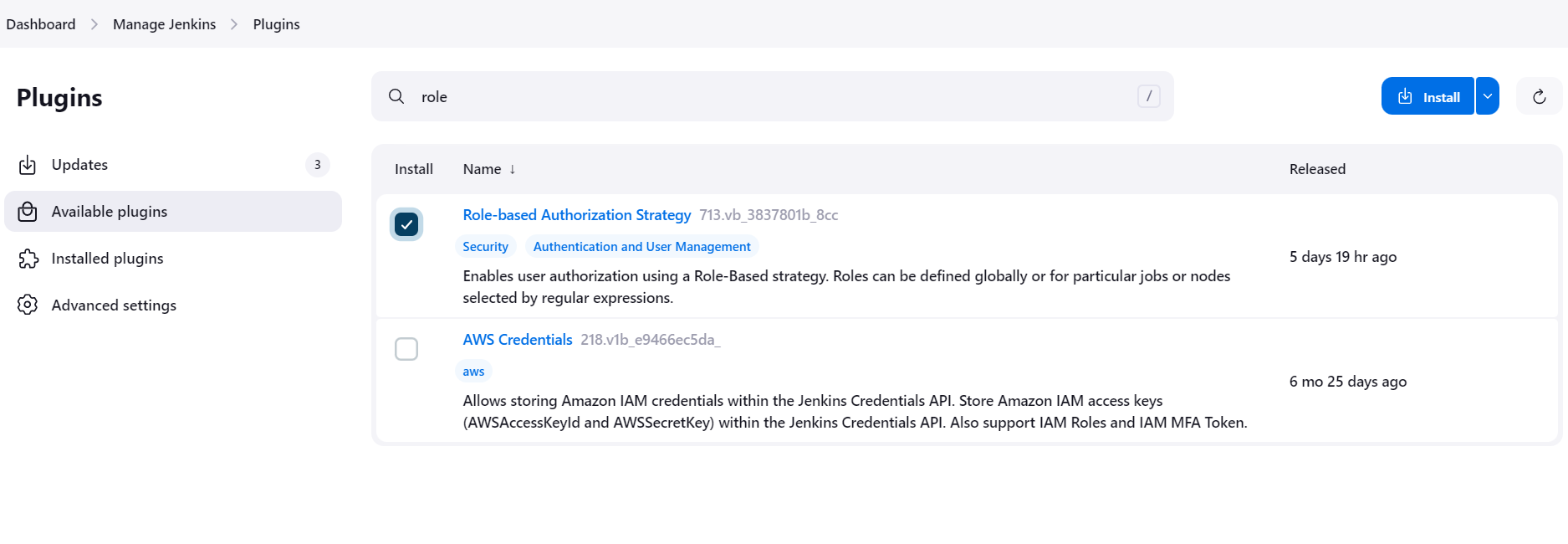


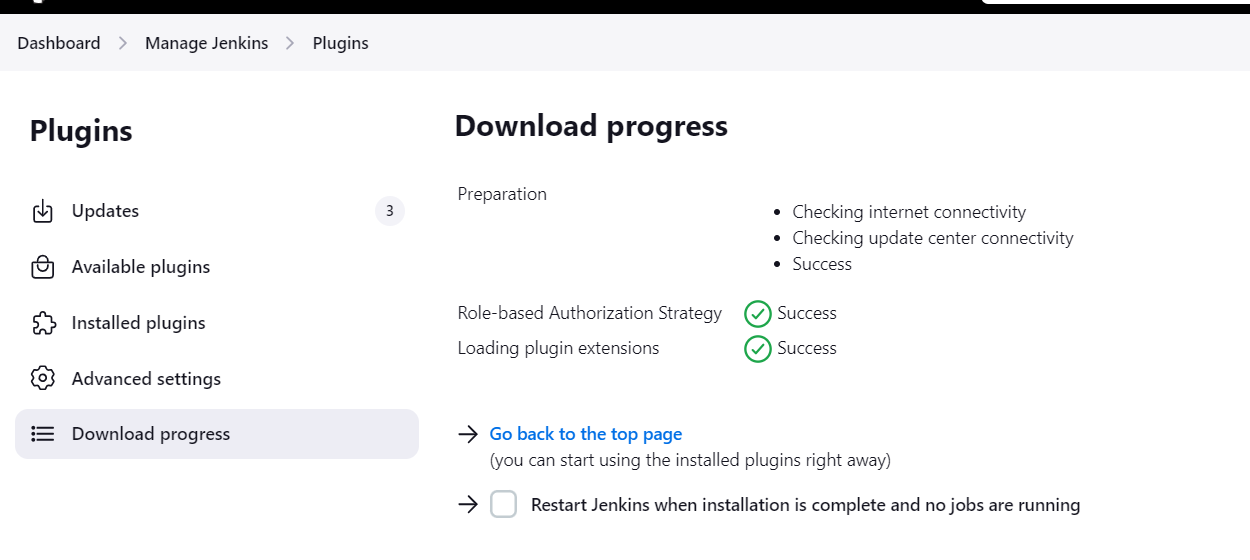


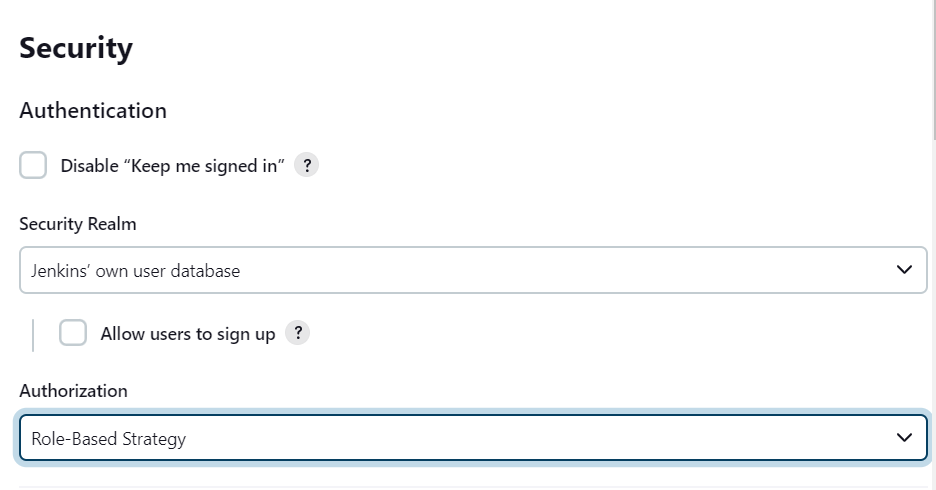




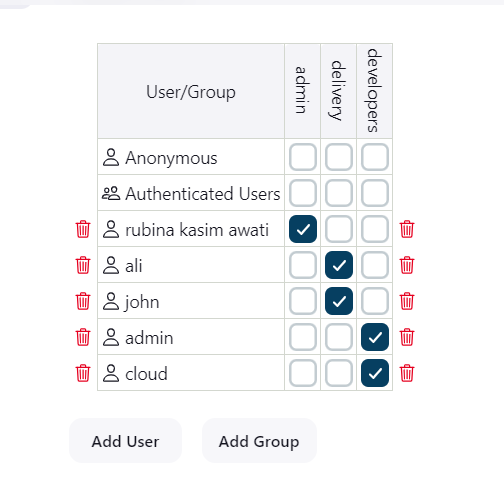










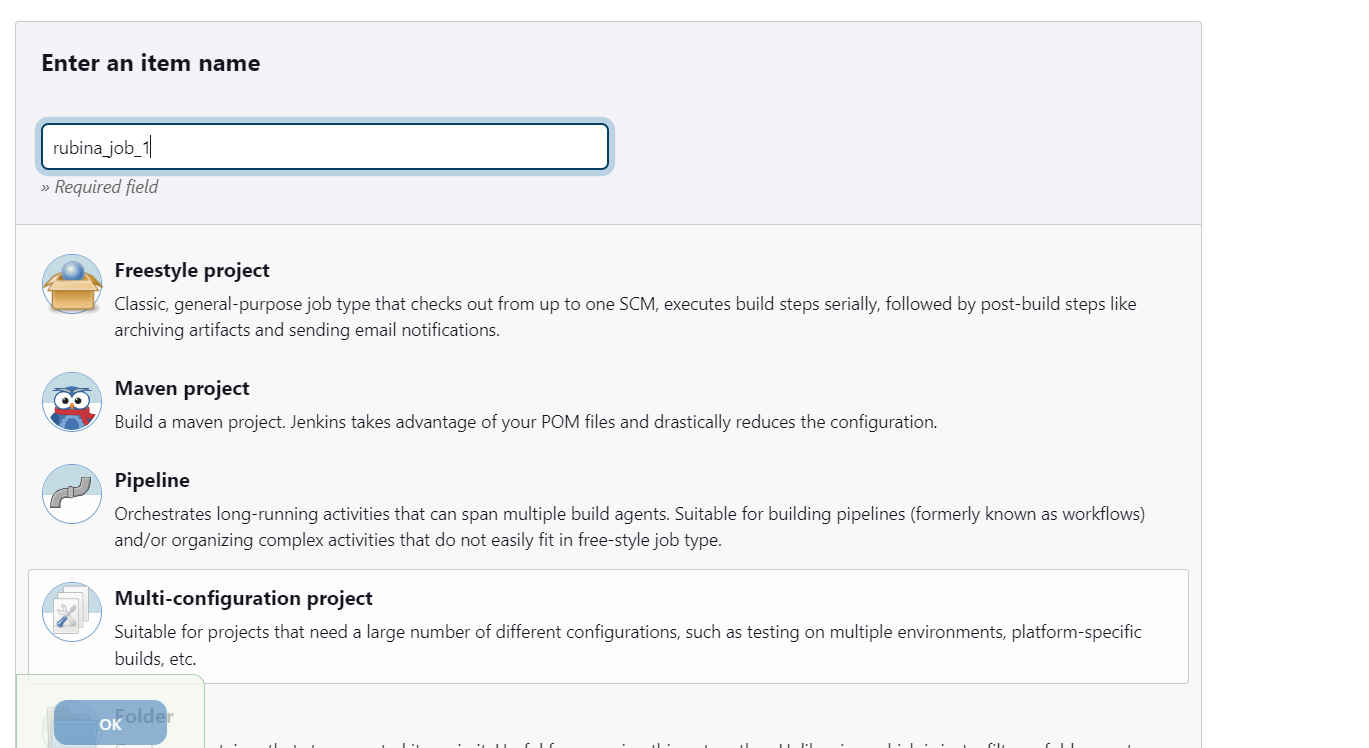


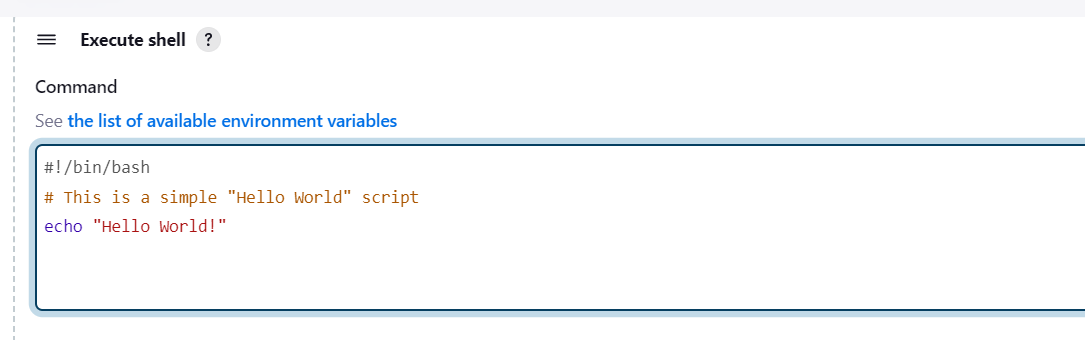
===============================================================================================================================================================================================================================

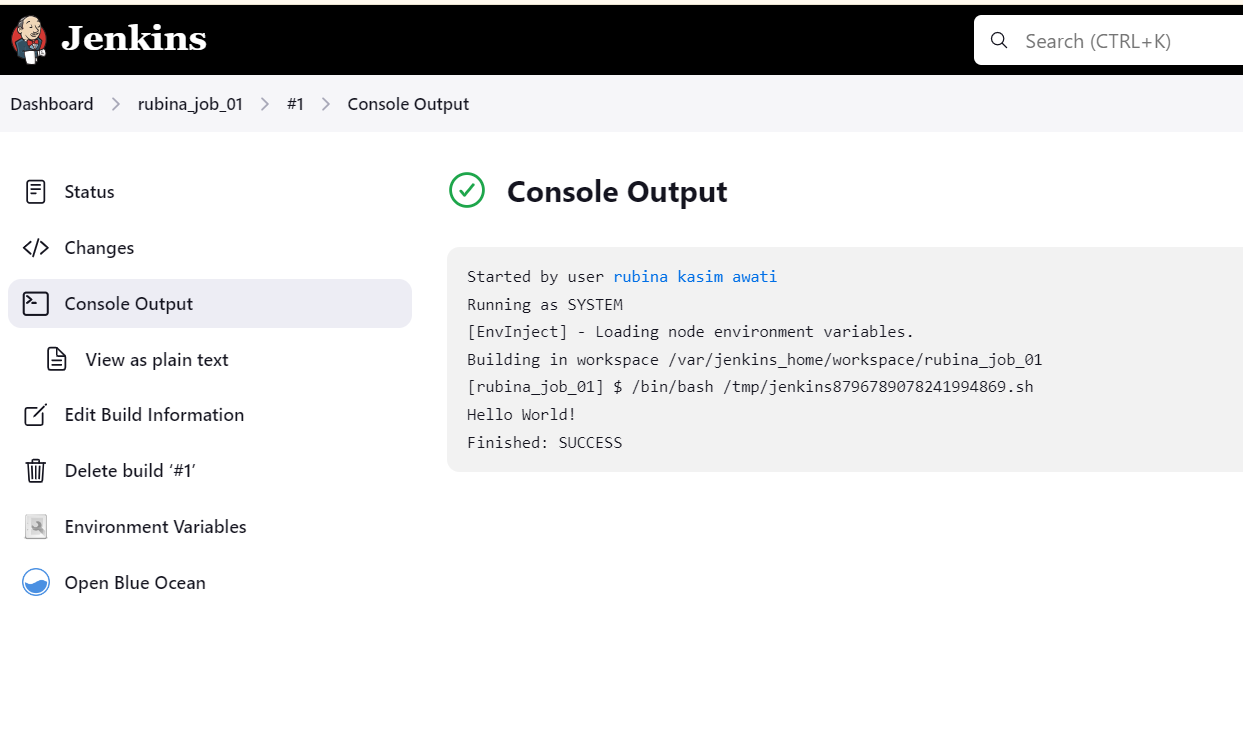
3. Create a Jenkins job named YOURNAME\_Job\_01 on a Jenkins Server.

This job should run below given shell script in the job.

Please check the console output of job and make sure it is successful.







==========================================================================================================================================================================================

4. Create a Jenkins job named YOURNAME\_Job\_02 that runs a shell script on a local server Jenkins.

The script should take in two parameters, e.g. num1, num2 from Jenkins.

This shell is taking 2 command line arguments as numbers.

Try to execute script locally to understand it more.

# sh your\_script\_name.sh 11 12

#] vim your\_script\_name.sh

#!/bin/bash

#Define a variable named "name"

name="John Doe"

#Print the value of the variable

echo "My name is $name"

#except the value from the user for two numbers and store the values in variables

num1 = $1

num2 = $2

#Use an if statement to check if the first number is greater than the second number

if [ $num1 -gt $num2 ]; then

echo "$num1 is greater than $num2"

else

echo "$num2 is greater than $num1"

fi

#Use a for loop to print the numbers from 1 to the value of the first number

for i in $(seq 1 $num1); do

echo $i

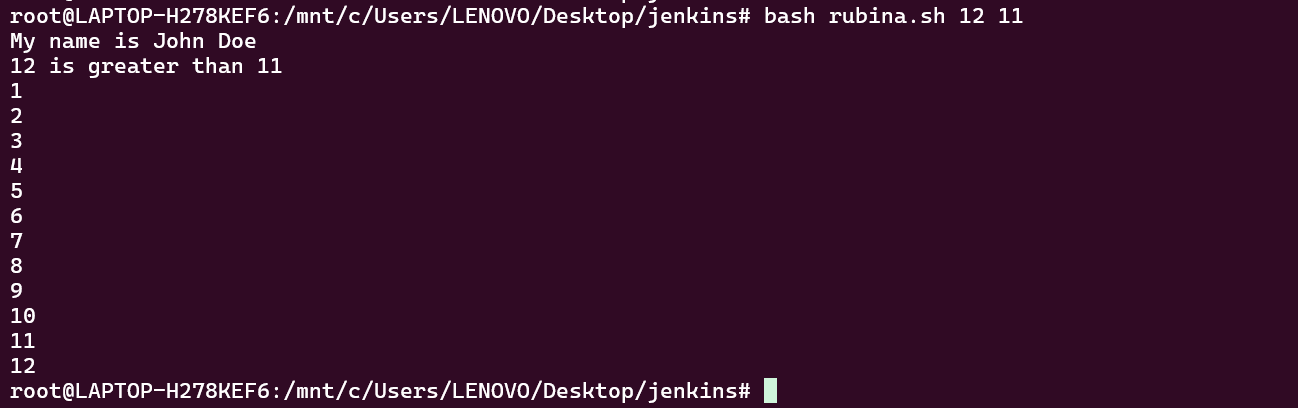
done

#Print a message indicating that the script is finished

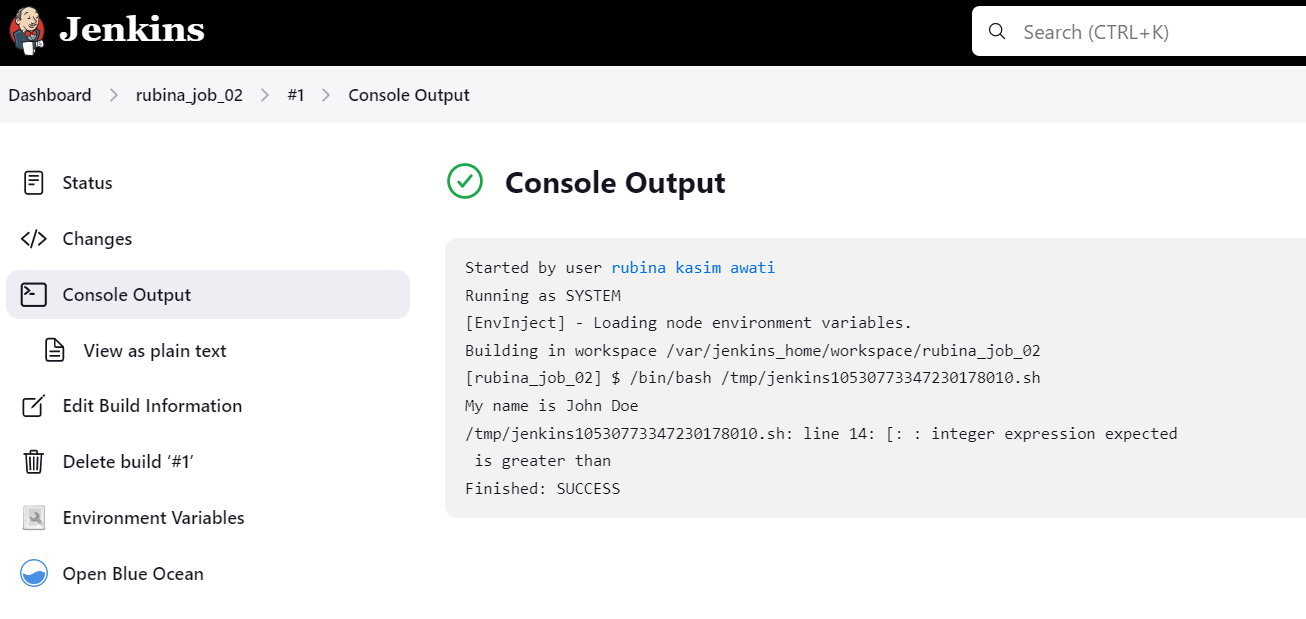
echo "Script finished."

Once script is tested locally , create 2 parameters in Jenkins and pass those parameters to the shell script.

Run the Jenkins job and check the console out for detailed job logs.







=============================================================================================================================================================

5. Create a Jenkins job named YOURNAME\_Job\_03 that runs a shell script on a local server using Jenkins parameters.

The script should take in three parameters, liKE NAME, LAST\_NAME, SHOW.

Print the NAME and LAST\_NAME if value of SHOW is true.

#!/bin/bash

# Define a variable and except the value from the user and store the values in variables

NAME=$1

LAST\_NAME=$2

SHOW=$3

# Use an if statement to check if the value SHOW is TRUE.

if [[ $SHOW == "true" ]]

then

echo " $NAME $LAST\_NAME"

else

echo " Not allowed to show the Names "

fi

# Print a message indicating that the script is finished

echo "Script finished."

