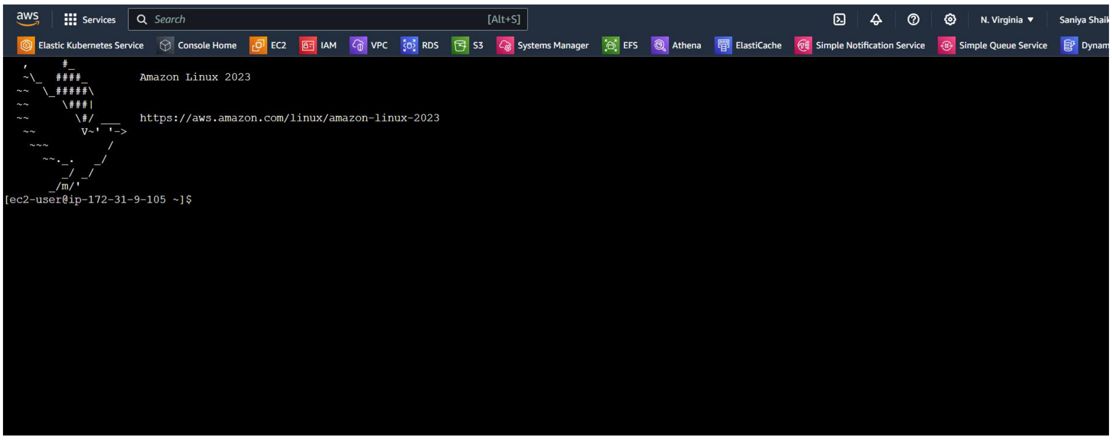
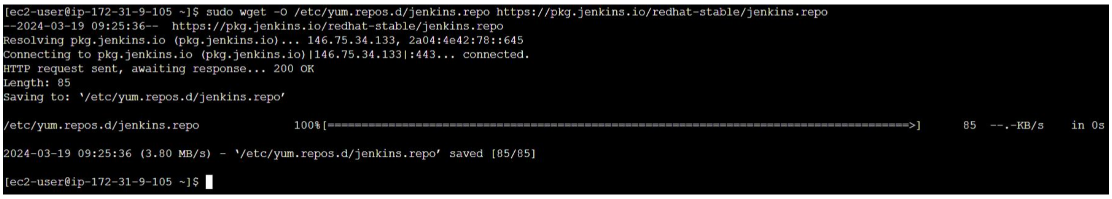
Jenkins_Assigment_01

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.

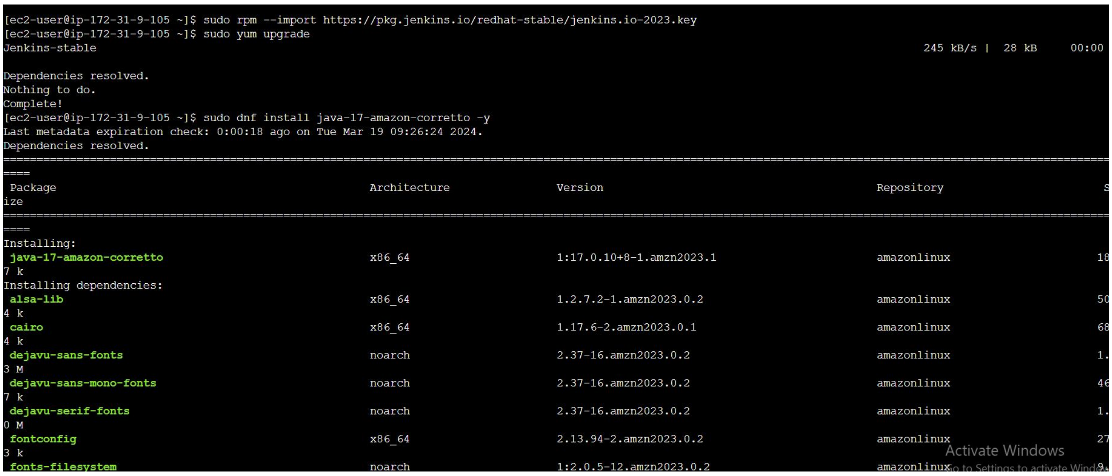
SOLUTION:-



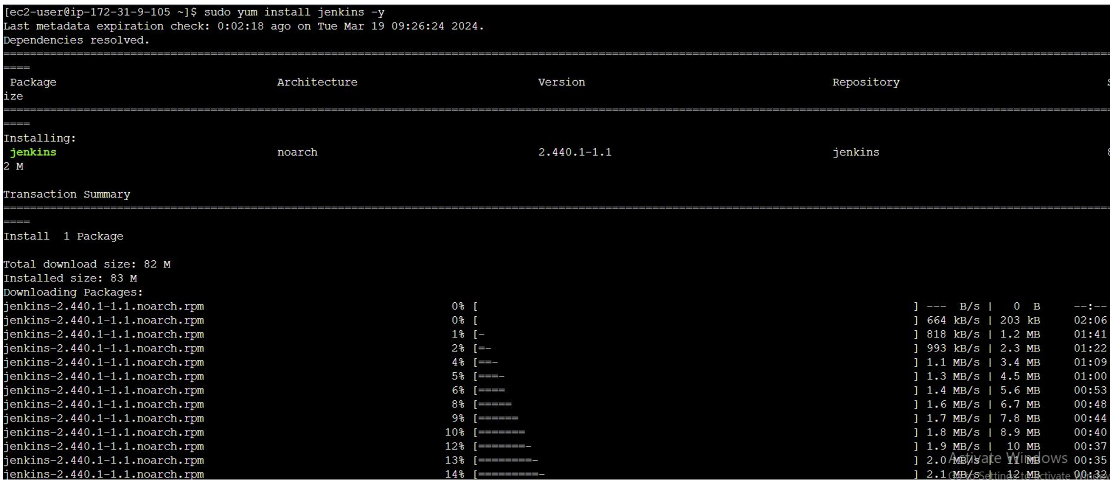
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-9-105 ~]$
```



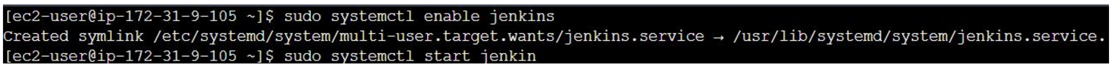
```
[ec2-user@ip-172-31-9-105 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-03-19 09:25:36--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo             100%[===============================================================================>]      85  --.-KB/s    in 0s

2024-03-19 09:25:36 (3.80 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[ec2-user@ip-172-31-9-105 ~]$ 
```



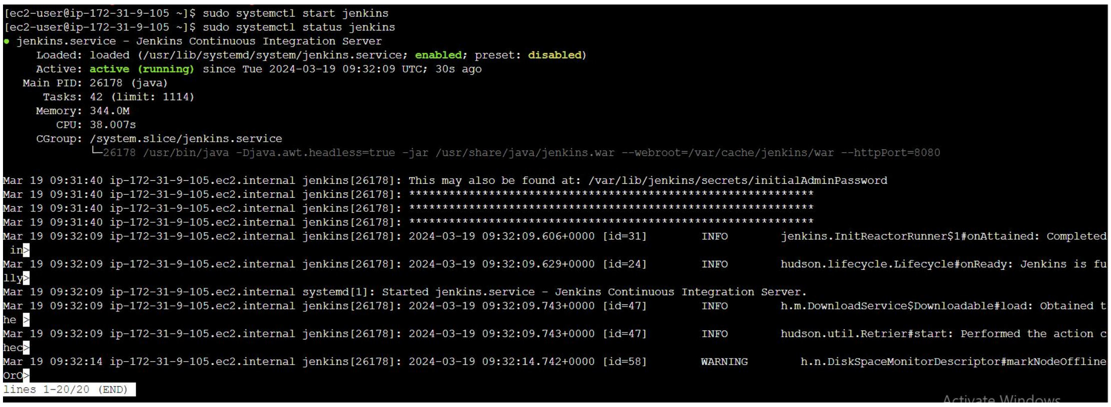
```
[ec2-user@ip-172-31-9-105 ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[ec2-user@ip-172-31-9-105 ~]$ sudo yum upgrade
Jenkins-stable                                                                                                                         245 kB/s |  28 kB     00:00

Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-9-105 ~]$ sudo dnf install java-17-amazon-corretto -y
Last metadata expiration check: 0:00:18 ago on Tue Mar 19 09:26:24 2024.
Dependencies resolved.
==================================================================================================================================================================
====
 Package                                            Architecture               Version                                      Repository                           S
ize
==================================================================================================================================================================
====
Installing:
 java-17-amazon-corretto                            x86_64                     1:17.0.10+8-1.amzn2023.1                     amazonlinux                          18
7 k
Installing dependencies:
 alsa-lib                                           x86_64                     1.2.7.2-1.amzn2023.0.2                       amazonlinux                          50
4 k
 cairo                                              x86_64                     1.17.6-2.amzn2023.0.1                        amazonlinux                          68
4 k
 dejavu-sans-fonts                                  noarch                     2.37-16.amzn2023.0.2                         amazonlinux                          1.
3 M
 dejavu-sans-mono-fonts                             noarch                     2.37-16.amzn2023.0.2                         amazonlinux                          46
7 k
 dejavu-serif-fonts                                 noarch                     2.37-16.amzn2023.0.2                         amazonlinux                          1.
0 M
 fontconfig                                         x86_64                     2.13.94-2.amzn2023.0.2                       amazonlinux                          27
3 k
 fonts-filesystem                                   noarch                     1:2.0.5-12.amzn2023.0.2                      amazonlinux                          9.
```



```
[ec2-user@ip-172-31-9-105 ~]$ sudo yum install jenkins -y
Last metadata expiration check: 0:02:18 ago on Tue Mar 19 09:26:24 2024.
Dependencies resolved.
================================================================================================================================================
====
 Package                               Architecture                         Version                                   Repository                            S
ize
================================================================================================================================================
====
Installing:
 jenkins                               noarch                               2.440.1-1.1                               jenkins                               8
2 M

Transaction Summary
================================================================================================================================================
====
Install  1 Package

Total download size: 82 M
Installed size: 83 M
Downloading Packages:
jenkins-2.440.1-1.1.noarch.rpm                                     0% [                                                                     ] ---  B/s |   0  B     --:--
jenkins-2.440.1-1.1.noarch.rpm                                     0% [                                                                     ] 664 kB/s | 203 kB     02:06
jenkins-2.440.1-1.1.noarch.rpm                                     1% [-                                                                    ] 818 kB/s | 1.2 MB     01:41
jenkins-2.440.1-1.1.noarch.rpm                                     2% [=-                                                                   ] 993 kB/s | 2.3 MB     01:22
jenkins-2.440.1-1.1.noarch.rpm                                     4% [==-                                                                  ] 1.1 MB/s | 3.4 MB     01:09
jenkins-2.440.1-1.1.noarch.rpm                                     5% [===-                                                                 ] 1.3 MB/s | 4.5 MB     01:00
jenkins-2.440.1-1.1.noarch.rpm                                     6% [====                                                                 ] 1.4 MB/s | 5.6 MB     00:53
jenkins-2.440.1-1.1.noarch.rpm                                     8% [=====                                                                ] 1.6 MB/s | 6.7 MB     00:48
jenkins-2.440.1-1.1.noarch.rpm                                     9% [======                                                               ] 1.7 MB/s | 7.8 MB     00:44
jenkins-2.440.1-1.1.noarch.rpm                                    10% [=======                                                              ] 1.8 MB/s | 8.9 MB     00:40
jenkins-2.440.1-1.1.noarch.rpm                                    12% [=======-                                                             ] 1.9 MB/s |  10 MB     00:37
jenkins-2.440.1-1.1.noarch.rpm                                    13% [========-                                                            ] 2.0 MB/s |  11 MB     00:35
jenkins-2.440.1-1.1.noarch.rpm                                    14% [=========-                                                           ] 2.1 MB/s |  12 MB     00:32
```



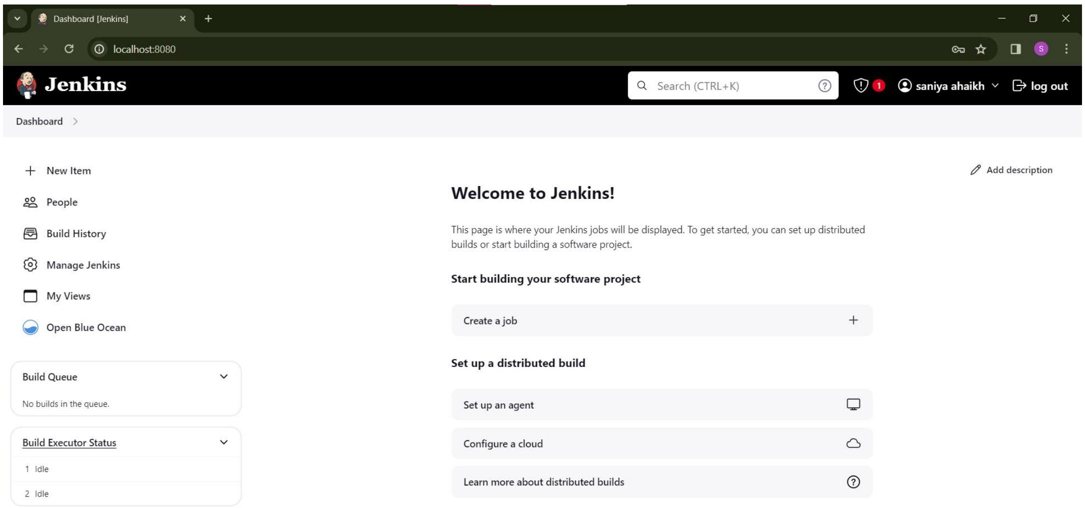
```
[ec2-user@ip-172-31-9-105 ~]$ sudo systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-9-105 ~]$ sudo systemctl start jenkin
```



```
[ec2-user@ip-172-31-9-105 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-9-105 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-03-19 09:32:09 UTC; 30s ago
   Main PID: 26178 (java)
      Tasks: 42 (limit: 1114)
     Memory: 344.0M
        CPU: 38.007s
     CGroup: /system.slice/jenkins.service
             └─26178 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 19 09:31:40 ip-172-31-9-105.ec2.internal jenkins[26178]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 19 09:31:40 ip-172-31-9-105.ec2.internal jenkins[26178]: *************************************************************
Mar 19 09:31:40 ip-172-31-9-105.ec2.internal jenkins[26178]: *************************************************************
Mar 19 09:31:40 ip-172-31-9-105.ec2.internal jenkins[26178]: *************************************************************
Mar 19 09:32:09 ip-172-31-9-105.ec2.internal jenkins[26178]: 2024-03-19 09:32:09.606+0000 [id=31]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed
 in>
Mar 19 09:32:09 ip-172-31-9-105.ec2.internal jenkins[26178]: 2024-03-19 09:32:09.629+0000 [id=24]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is fu
lly>
Mar 19 09:32:09 ip-172-31-9-105.ec2.internal systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 19 09:32:09 ip-172-31-9-105.ec2.internal jenkins[26178]: 2024-03-19 09:32:09.743+0000 [id=47]        INFO        h.m.DownloadService$Downloadable#load: Obtained t
he >
Mar 19 09:32:09 ip-172-31-9-105.ec2.internal jenkins[26178]: 2024-03-19 09:32:09.743+0000 [id=47]        INFO        hudson.util.Retrier#start: Performed the action c
hec>
Mar 19 09:32:14 ip-172-31-9-105.ec2.internal jenkins[26178]: 2024-03-19 09:32:14.742+0000 [id=58]        WARNING        h.n.DiskSpaceMonitorDescriptor#markNodeOffline
OrO>
lines 1-20/20 (END)
```

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.

SOLUTION:-



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

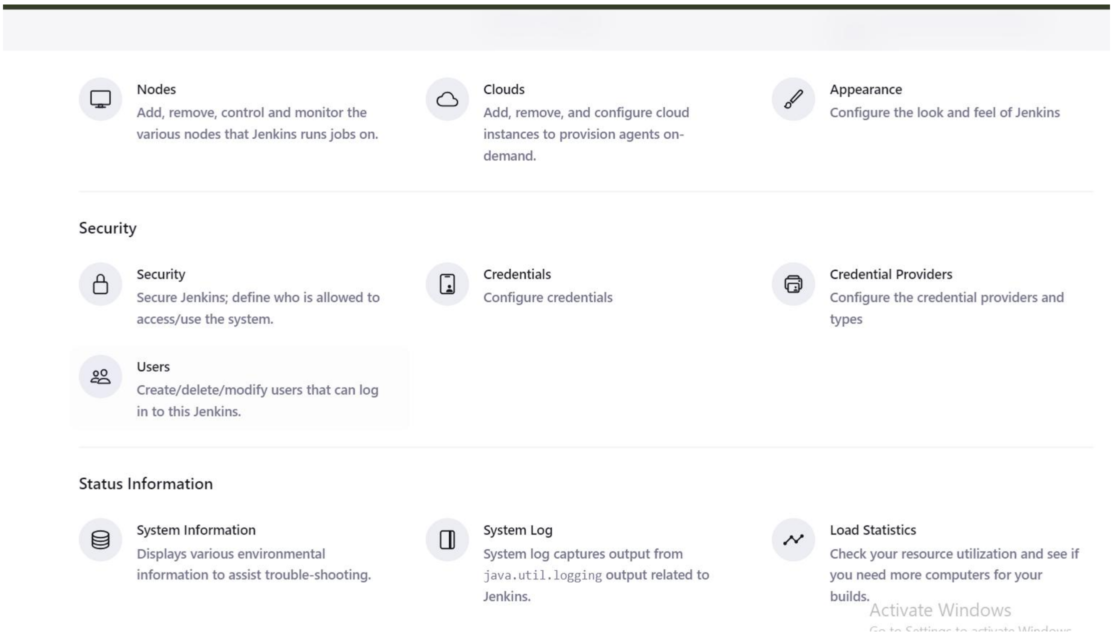
Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle



Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

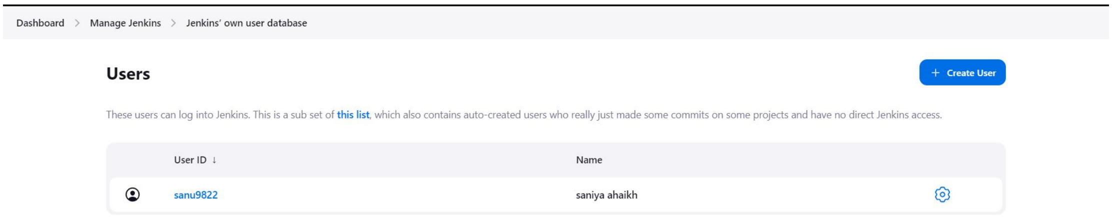
Users
Create/delete/modify users that can log in to this Jenkins.

Status Information

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

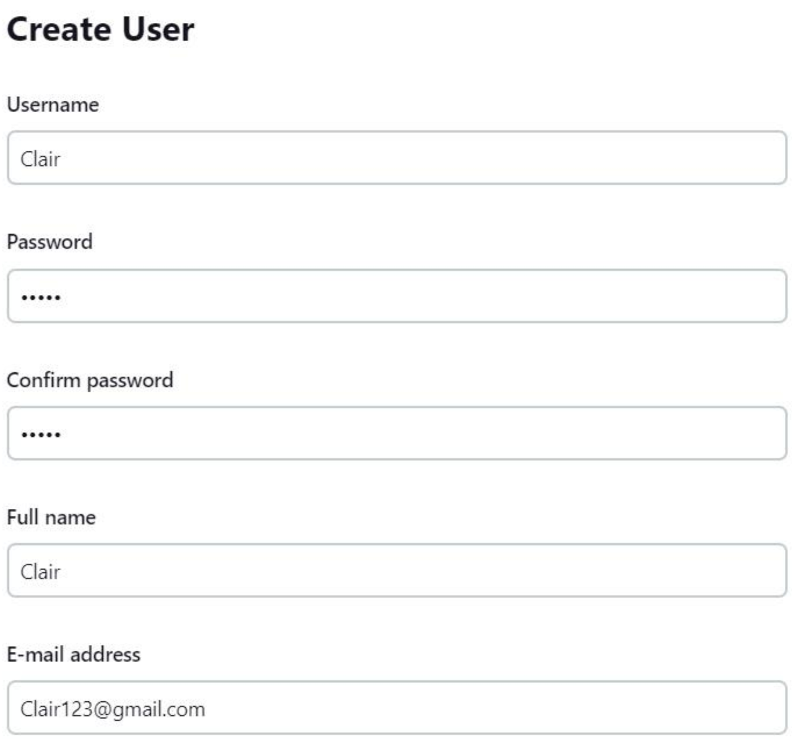


Users

These users can log into Jenkins. This is a sub set of this list, which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

	User ID ↓	Name	
	sanu9822	saniya ahaikh	

First User



Create User

Username: Clair

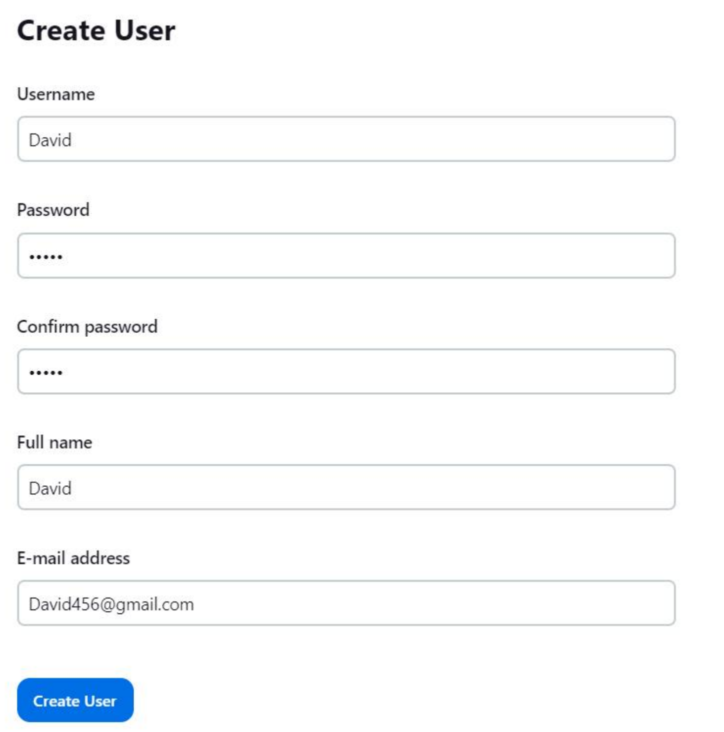
Password: •••••

Confirm password: •••••

Full name: Clair

E-mail address: Clair123@gmail.com

Second User



Create User

Username

David

Password

•••••

Confirm password

•••••

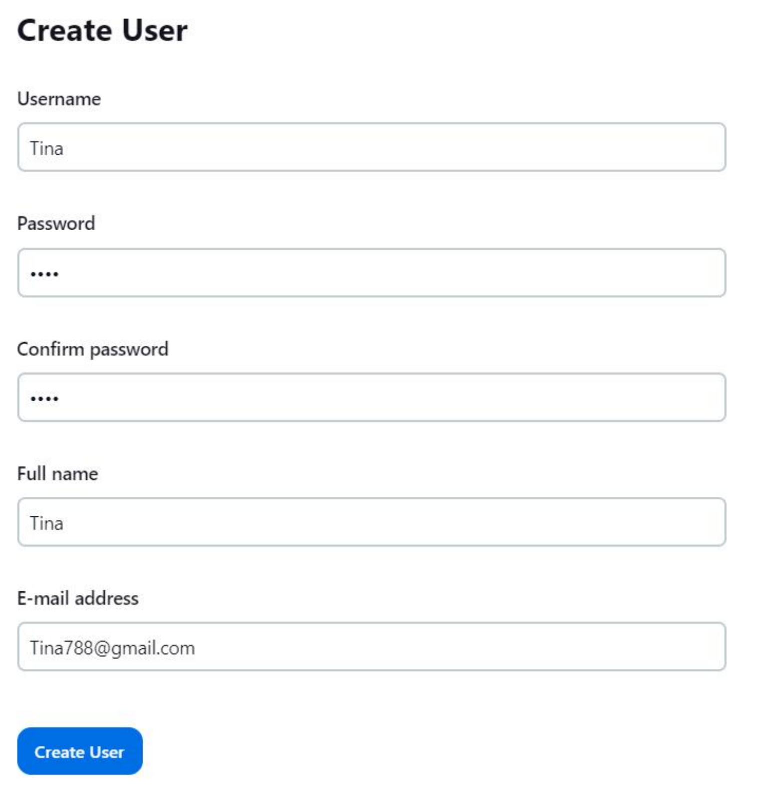
Full name

David

E-mail address

David456@gmail.com

Third User



Create User

Username

Tina

Password

••••

Confirm password

••••

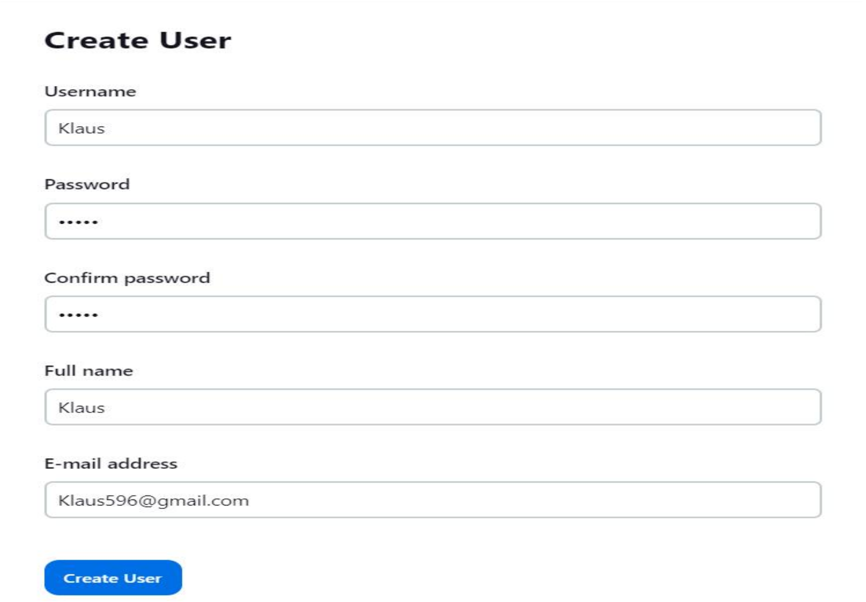
Full name

Tina

E-mail address

Tina788@gmail.com

Fourth User



Create User

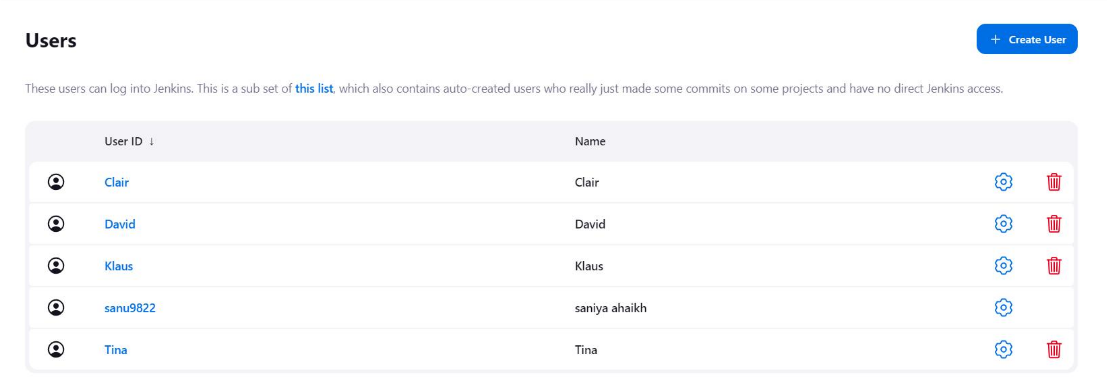
Username: Klaus

Password: •••••

Confirm password: •••••

Full name: Klaus

E-mail address: Klaus596@gmail.com

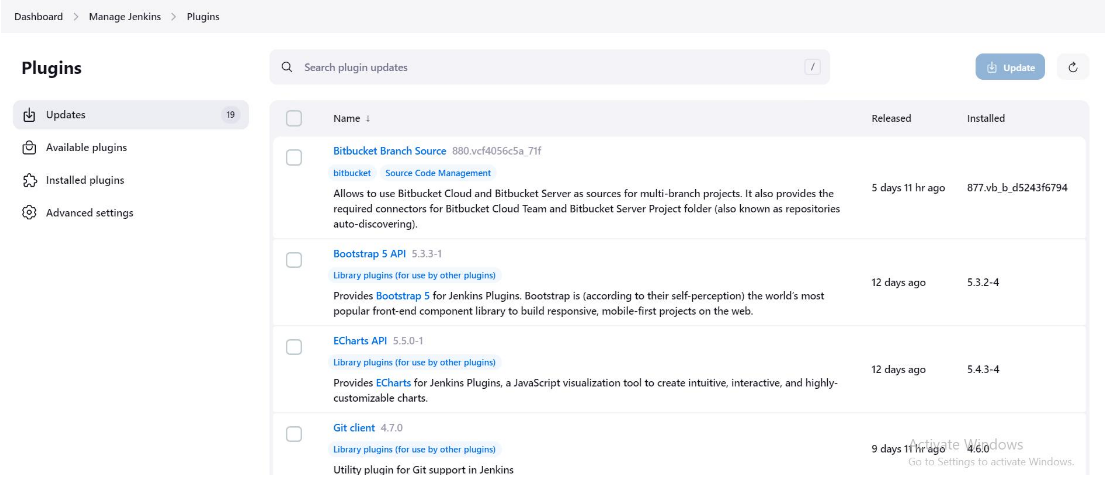


Users

These users can log into Jenkins. This is a sub set of this list, which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

User ID ↓	Name
Clair	Clair
David	David
Klaus	Klaus
sanu9822	saniya ahaikh
Tina	Tina

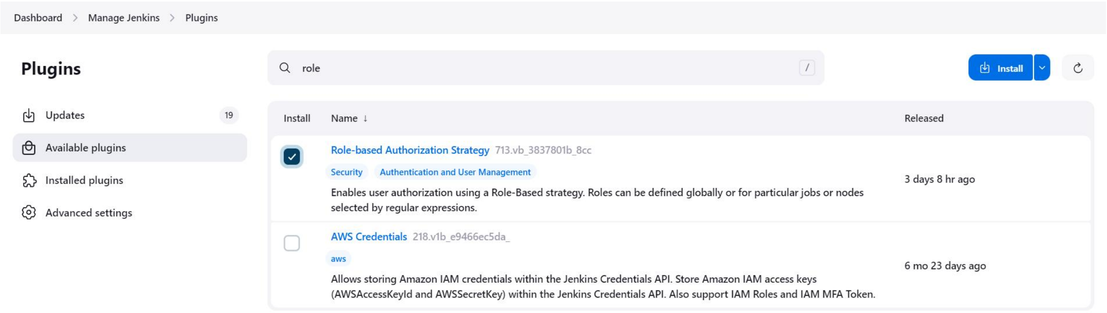
create 2 Jenkins roles named developers & delivery



Dashboard > Manage Jenkins > Plugins

Plugins

Name ↓	Released	Installed
Bitbucket Branch Source 880.vcf4056c5a_71f bitbucket Source Code Management Allows to use Bitbucket Cloud and Bitbucket Server as sources for multi-branch projects. It also provides the required connectors for Bitbucket Cloud Team and Bitbucket Server Project folder (also known as repositories auto-discovering).	5 days 11 hr ago	877.vb_b_d5243f6794
Bootstrap 5 API 5.3.3-1 Library plugins (for use by other plugins) Provides Bootstrap 5 for Jenkins Plugins. Bootstrap is (according to their self-perception) the world's most popular front-end component library to build responsive, mobile-first projects on the web.	12 days ago	5.3.2-4
ECharts API 5.5.0-1 Library plugins (for use by other plugins) Provides ECharts for Jenkins Plugins, a JavaScript visualization tool to create intuitive, interactive, and highly-customizable charts.	12 days ago	5.4.3-4
Git client 4.7.0 Library plugins (for use by other plugins) Utility plugin for Git support in Jenkins	9 days 11 hr ago	4.6.0



Dashboard > Manage Jenkins > Plugins

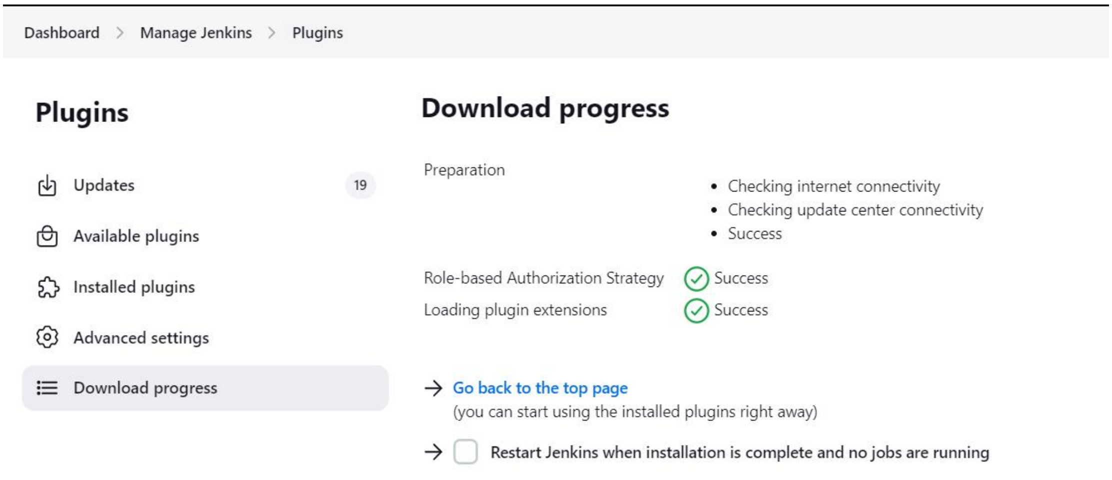
Plugins

- Updates 19
- Available plugins
- Installed plugins
- Advanced settings

Search: role

Install

Install	Name ↓	Released
☑	Role-based Authorization Strategy 713.vb_3837801b_8cc Security · Authentication and User Management Enables user authorization using a Role-Based strategy. Roles can be defined globally or for particular jobs or nodes selected by regular expressions.	3 days 8 hr ago
□	AWS Credentials 218.v1b_e9466ec5da_ aws Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.	6 mo 23 days ago



Dashboard > Manage Jenkins > Plugins

Plugins

- Updates 19
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

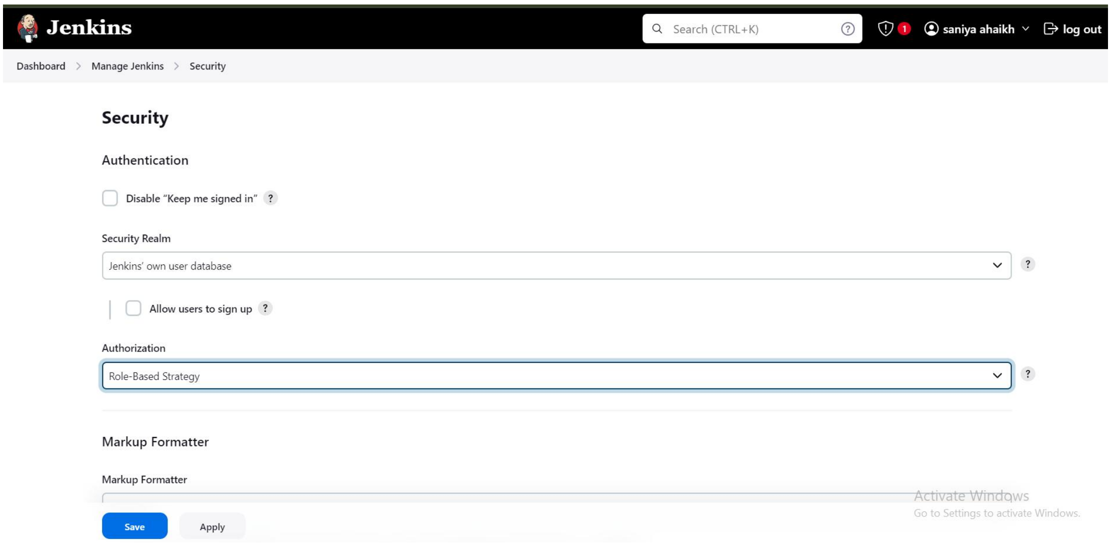
Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Role-based Authorization Strategy ✓ Success

Loading plugin extensions ✓ Success

→ Go back to the top page
(you can start using the installed plugins right away)

→ □ Restart Jenkins when installation is complete and no jobs are running



Jenkins | Search (CTRL+K) | 1 | saniya ahaikh | log out

Dashboard > Manage Jenkins > Security

Security

Authentication

□ Disable "Keep me signed in" ?

Security Realm: Jenkins' own user database

□ Allow users to sign up ?

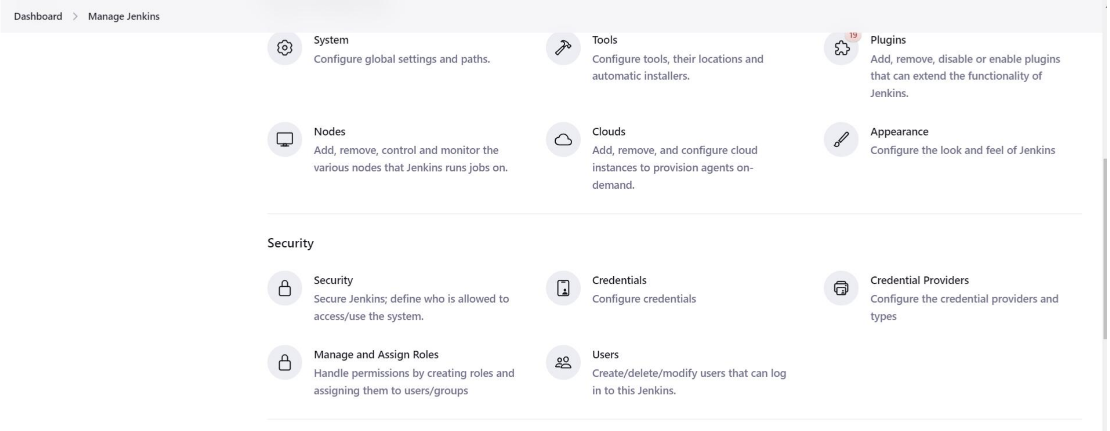
Authorization: Role-Based Strategy

Markup Formatter

Markup Formatter

Save | Apply

Activate Windows
Go to Settings to activate Windows.



Dashboard > Manage Jenkins

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

Security

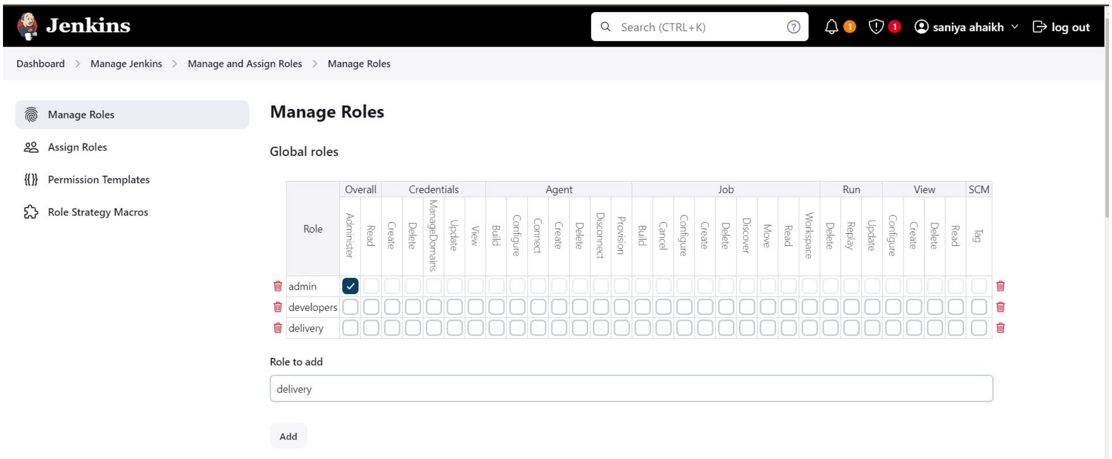
Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Manage and Assign Roles
Handle permissions by creating roles and assigning them to users/groups

Users
Create/delete/modify users that can log in to this Jenkins.



Dashboard > Manage Jenkins > Manage and Assign Roles > Manage Roles

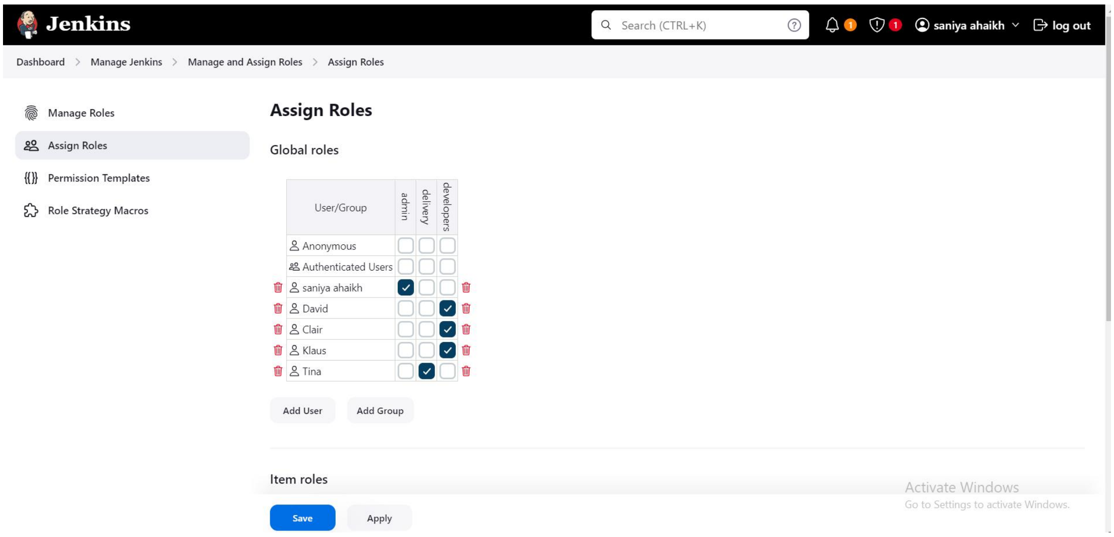
Manage Roles

Global roles

	Overall		Credentials					Agent							Job									Run			View				SCM
Role	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create	Delete	Read	Tag
admin	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
developers	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
delivery	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Role to add

delivery



Dashboard > Manage Jenkins > Manage and Assign Roles > Assign Roles

Assign Roles

Global roles

User/Group	admin	delivery	developers
Anonymous	□	□	□
Authenticated Users	□	□	□
saniya ahaikh	☑	□	□
David	□	□	☑
Clair	□	□	☑
Klaus	□	□	☑
Tina	□	☑	□

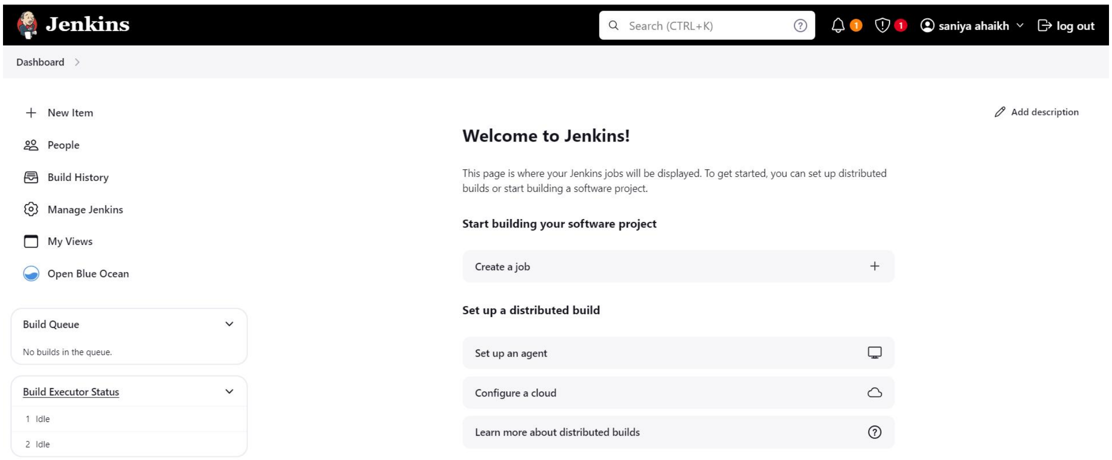
Item roles

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.

```
#!/bin/bash
# This is a simple "Hello World" script
echo "Hello World!"
```

Prepare well formatted document with screenshots.

SOLUTION:-



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

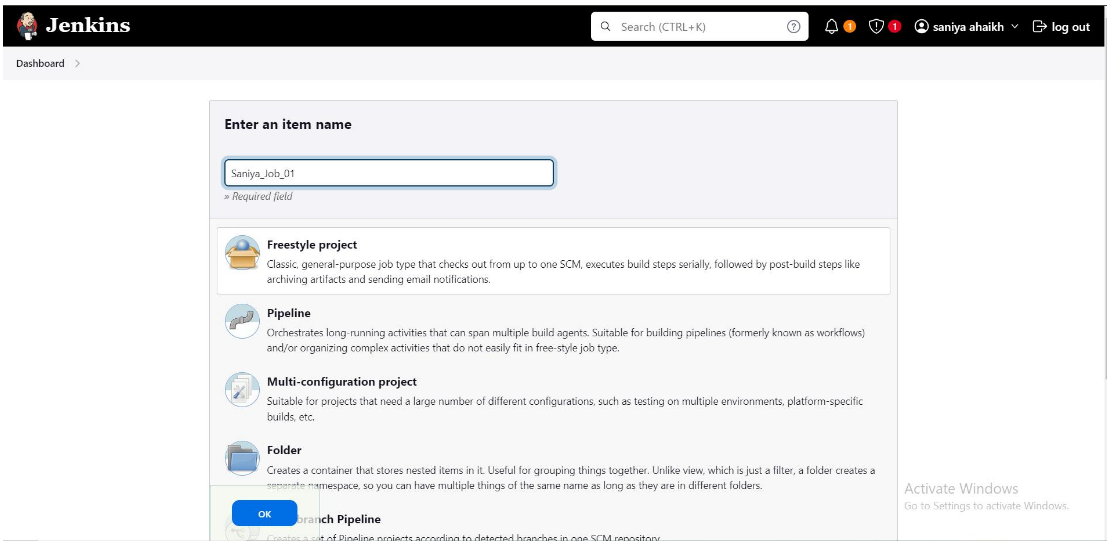
Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle



Enter an item name

Saniya_Job_01

» Required field

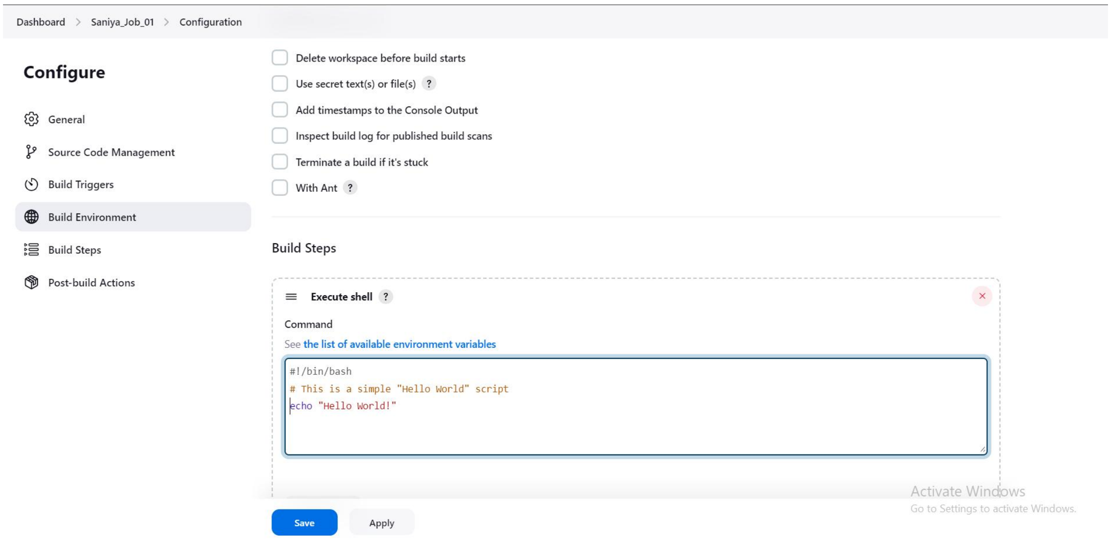
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.



☐ Delete workspace before build starts
☐ Use secret text(s) or file(s) ?
☐ Add timestamps to the Console Output
☐ Inspect build log for published build scans
☐ Terminate a build if it's stuck
☐ With Ant ?

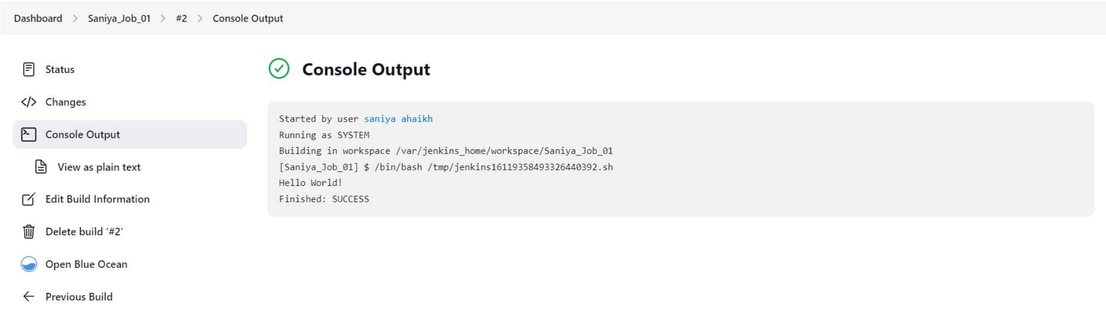
Build Steps

Execute shell ?

Command

See the list of available environment variables

```
#!/bin/bash
# This is a simple "Hello World" script
echo "Hello World!"
```



Console Output

```
Started by user saniya ahaikh
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Saniya_Job_01
[Saniya_Job_01] $ /bin/bash /tmp/jenkins16119358493326440392.sh
Hello World!
Finished: SUCCESS
```

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.

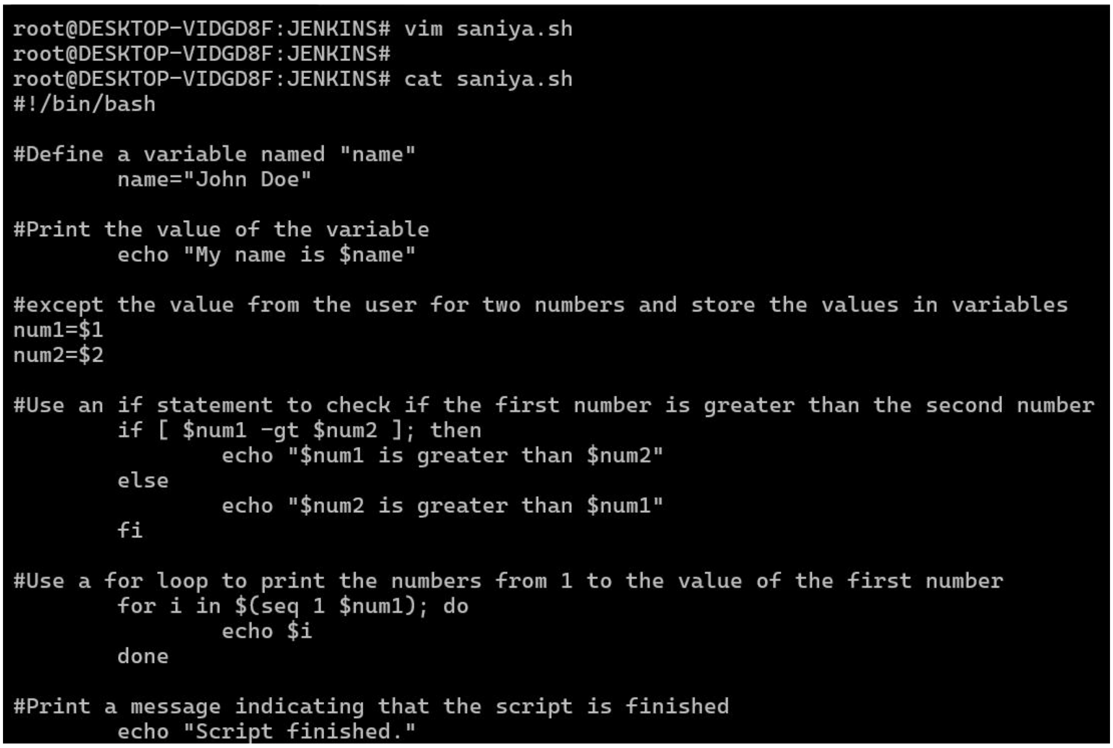
```
# sh your_script_name.sh 11 12
```

```
#] 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
```

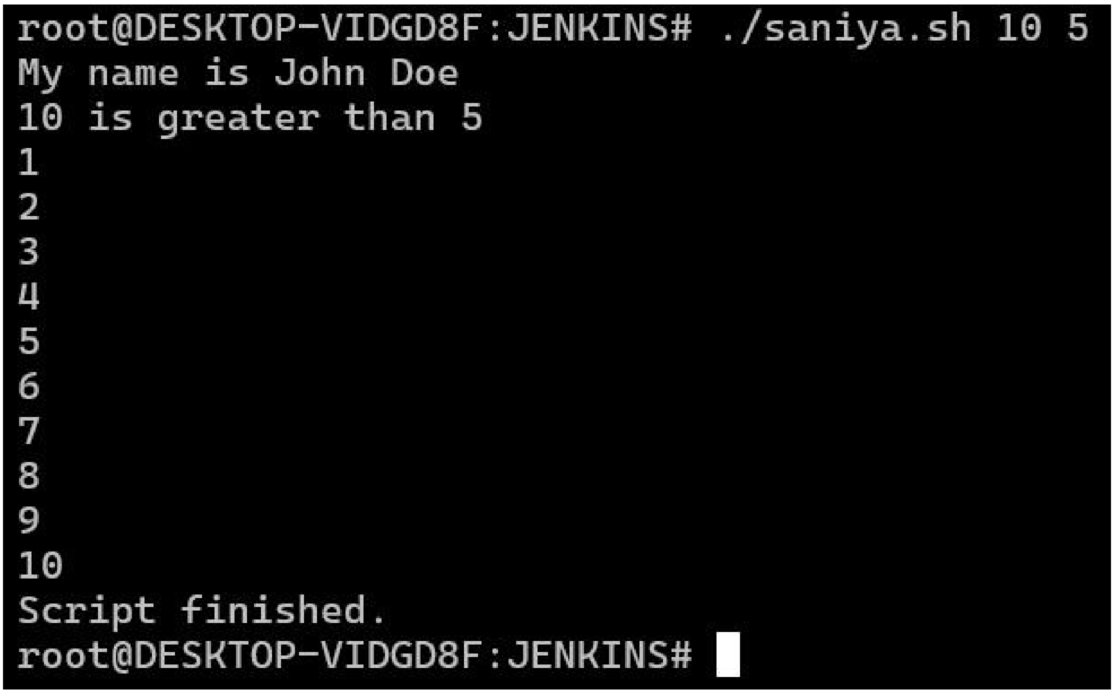
```
#Print a message indicating that the script is finished
        echo "Script finished."
```

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.

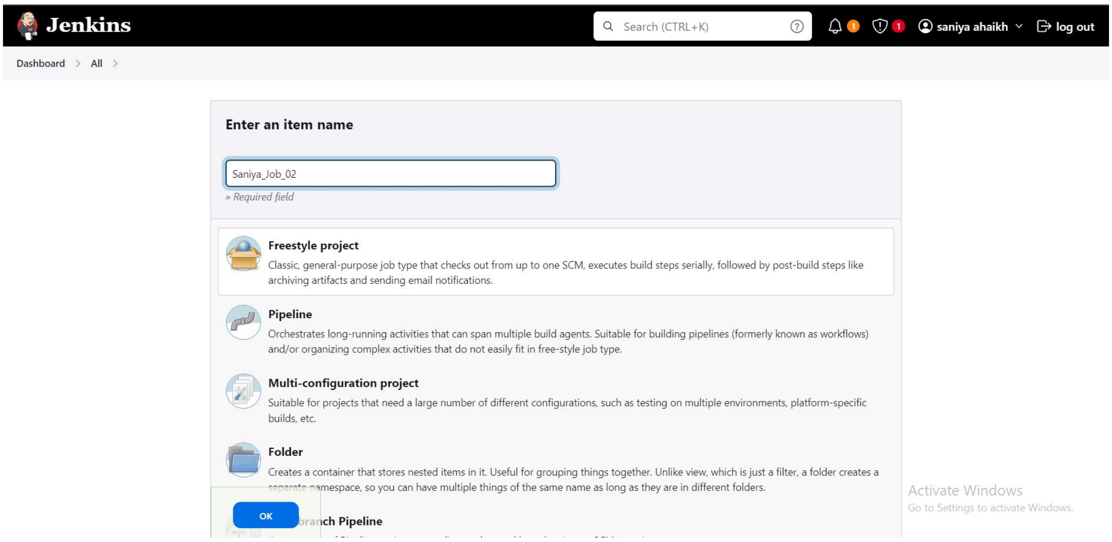
SOLUTION:-



```
root@DESKTOP-VIDGD8F:JENKINS# vim saniya.sh
root@DESKTOP-VIDGD8F:JENKINS#
root@DESKTOP-VIDGD8F:JENKINS# cat saniya.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."
```



```
root@DESKTOP-VIDGD8F:JENKINS# ./saniya.sh 10 5
My name is John Doe
10 is greater than 5
1
2
3
4
5
6
7
8
9
10
Script finished.
root@DESKTOP-VIDGD8F:JENKINS#
```



Enter an item name

Saniya_Job_02

» *Required field*

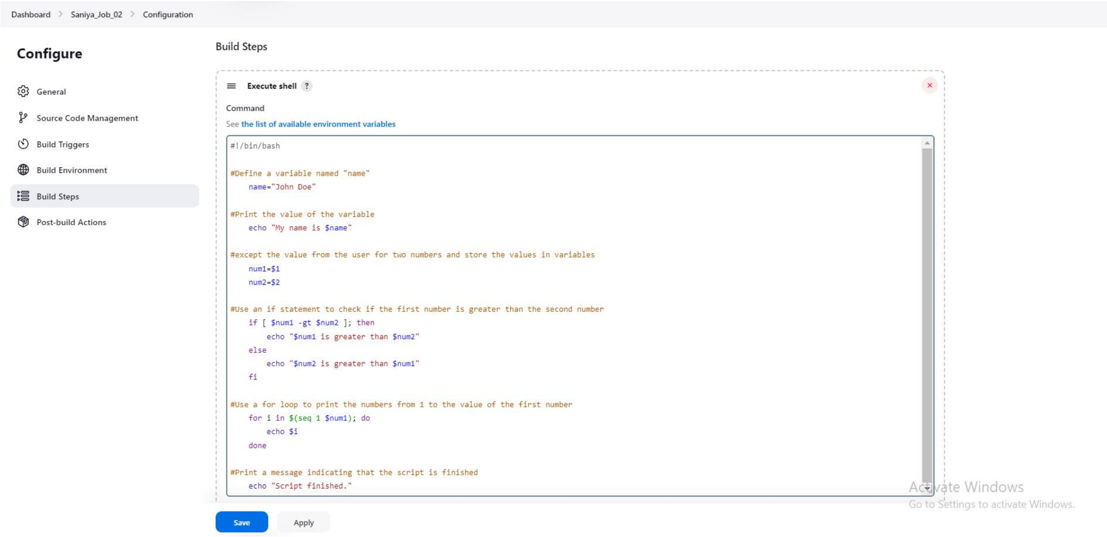
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

branch Pipeline



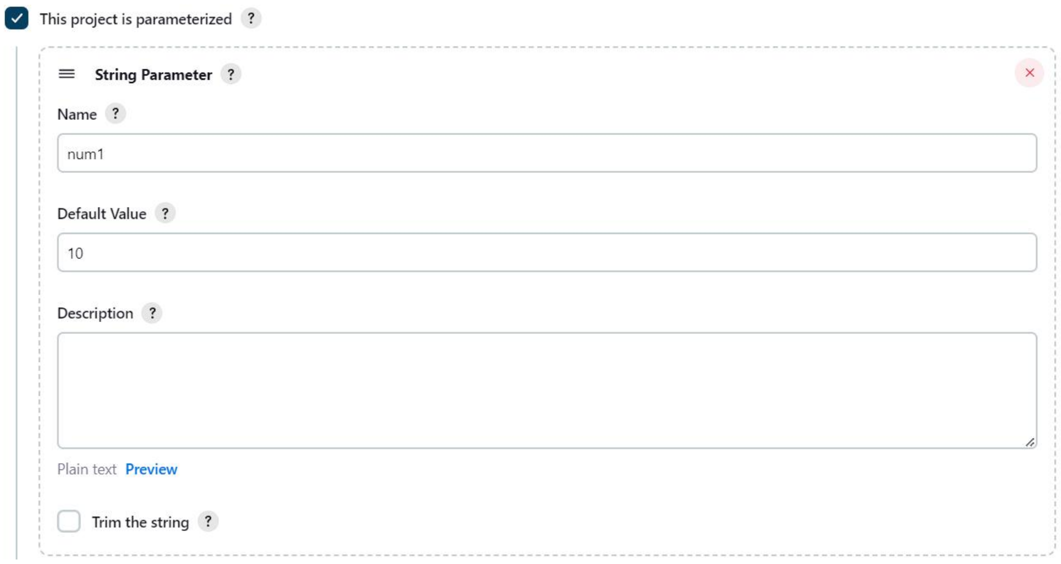
Build Steps

Execute shell

Command

See the list of available environment variables

```
#!/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."
```



☑ This project is parameterized ?

String Parameter ?

Name ?

num1

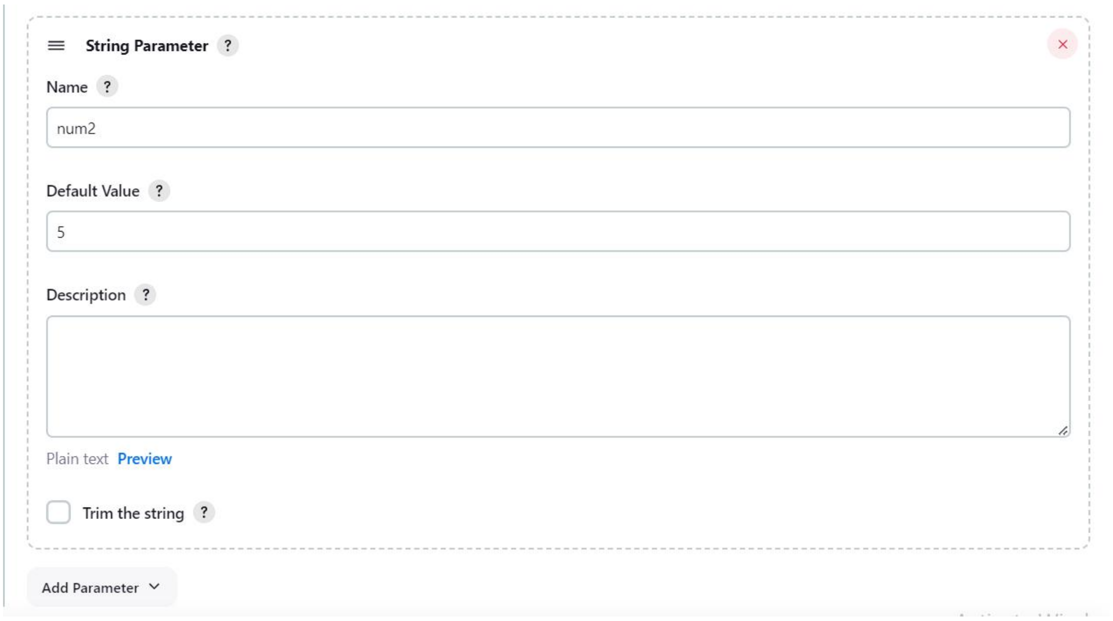
Default Value ?

10

Description ?

Plain text Preview

□ Trim the string ?



String Parameter ?

Name ?

num2

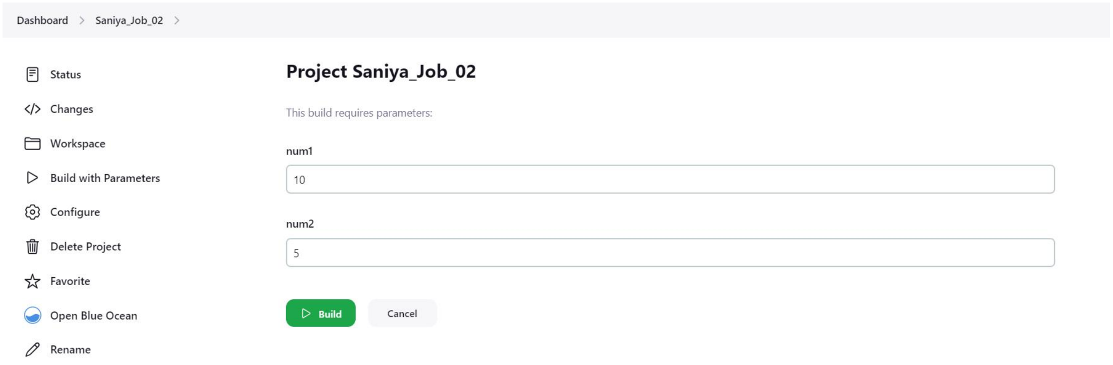
Default Value ?

5

Description ?

Plain text Preview

□ Trim the string ?



Project Saniya_Job_02

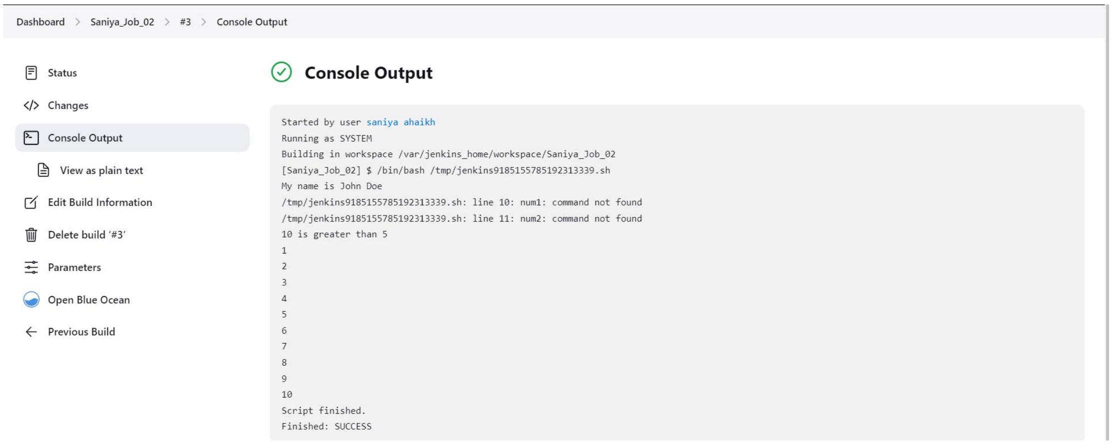
This build requires parameters:

num1

10

num2

5



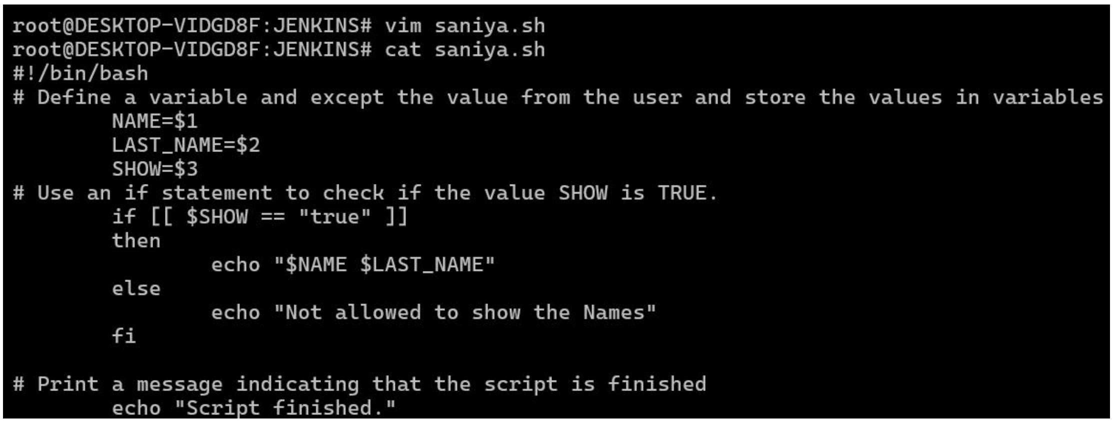
Console Output

```
Started by user saniya ahaikh
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Saniya_Job_02
[Saniya_Job_02] $ /bin/bash /tmp/jenkins9185155785192313339.sh
My name is John Doe
/tmp/jenkins9185155785192313339.sh: line 10: num1: command not found
/tmp/jenkins9185155785192313339.sh: line 11: num2: command not found
10 is greater than 5
1
2
3
4
5
6
7
8
9
10
Script finished.
Finished: SUCCESS
```

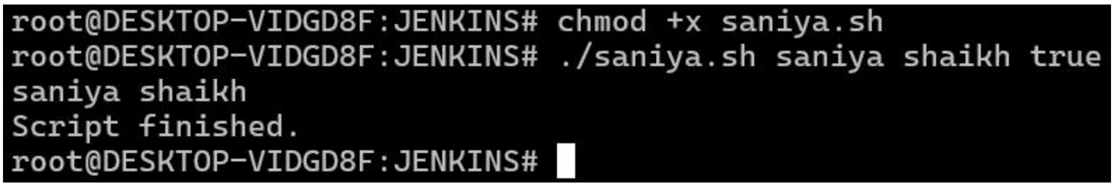
5. Create a Jenkins job named YOURNAME_Job_03 that runs a shell script on a local server using Jenkins parameters.
   The script should take in three parameters, liKE NAME, LAST_NAME, SHOW.
   Print the NAME and LAST_NAME if value of SHOW is true.

```
#!/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."
```

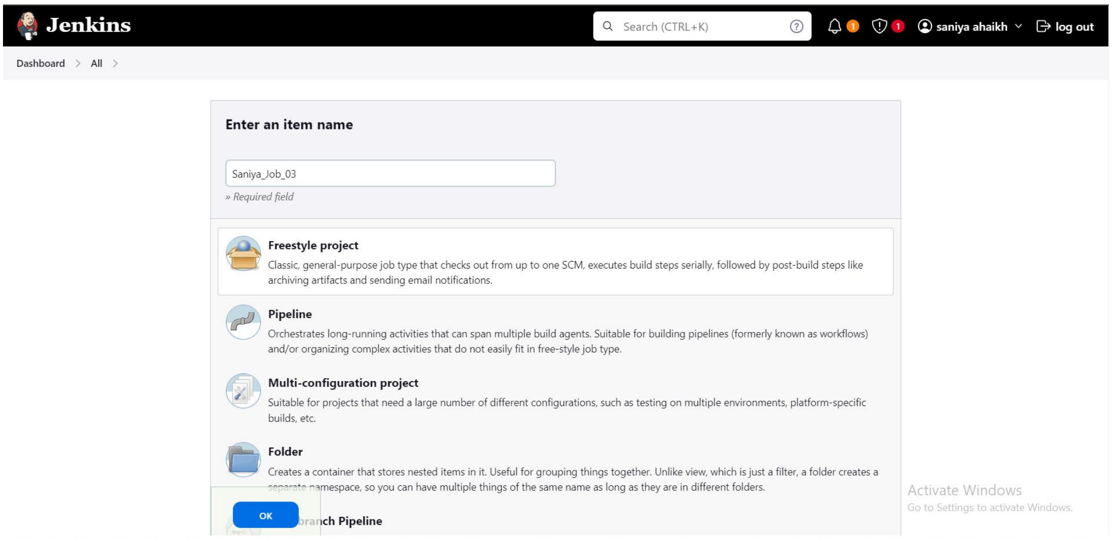
SOLUTION:-



```
root@DESKTOP-VIDGD8F:JENKINS# vim saniya.sh
root@DESKTOP-VIDGD8F:JENKINS# cat saniya.sh
#!/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."
```



```
root@DESKTOP-VIDGD8F:JENKINS# chmod +x saniya.sh
root@DESKTOP-VIDGD8F:JENKINS# ./saniya.sh saniya shaikh true
saniya shaikh
Script finished.
root@DESKTOP-VIDGD8F:JENKINS# 
```



Enter an item name

Saniya_Job_03

» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

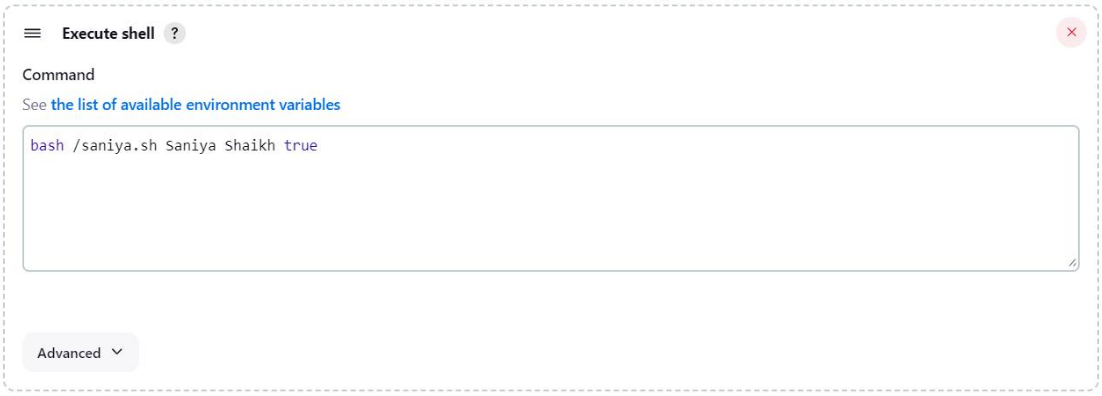
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

…ranch Pipeline

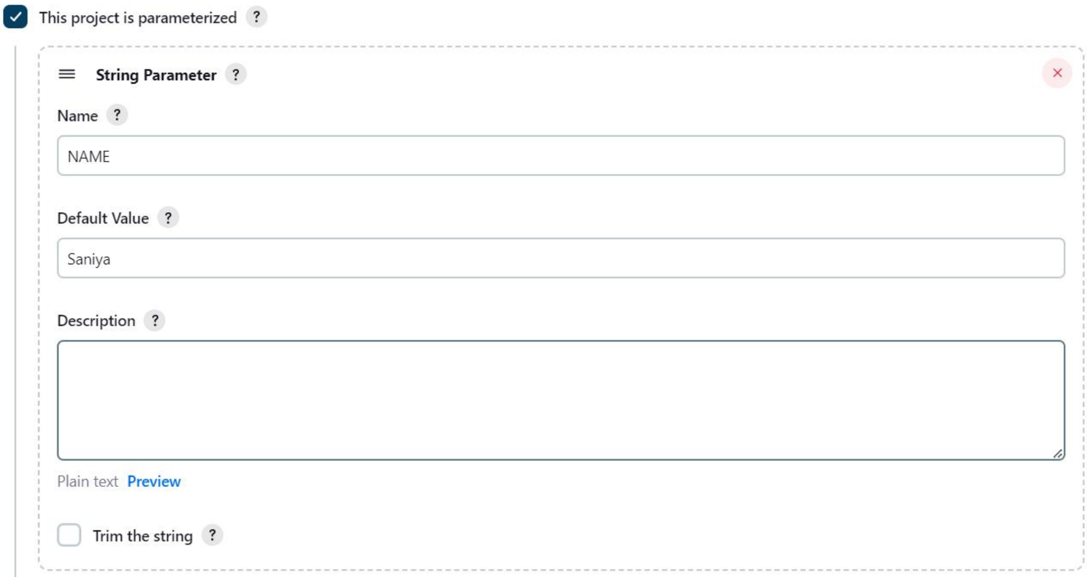
Activate Windows
Go to Settings to activate Windows.



Command

See the list of available environment variables

```
bash /saniya.sh Saniya Shaikh true
```



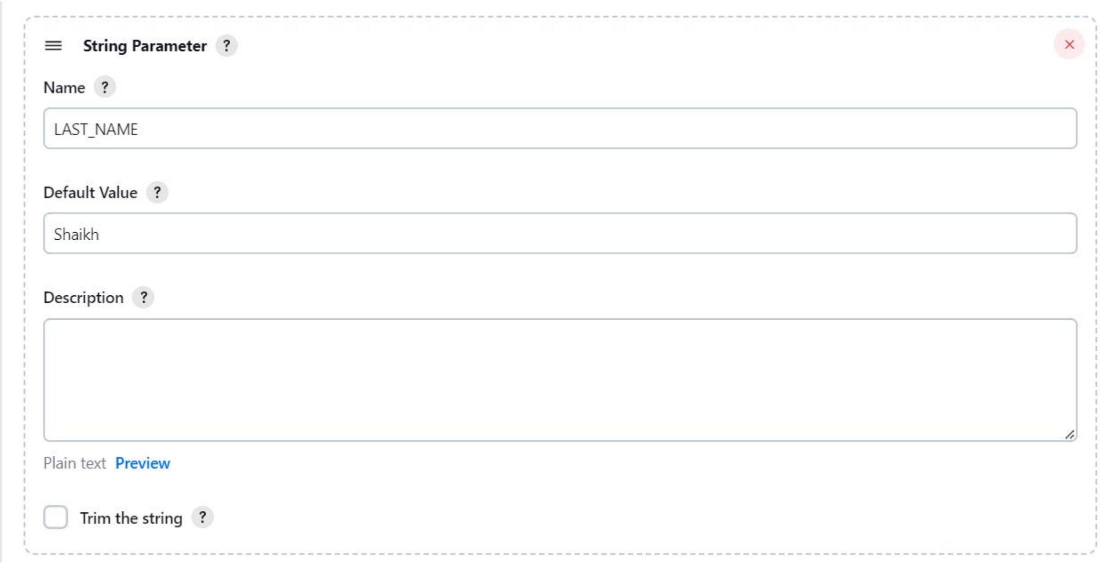
☑ This project is parameterized

Name: NAME

Default Value: Saniya

Description:

□ Trim the string

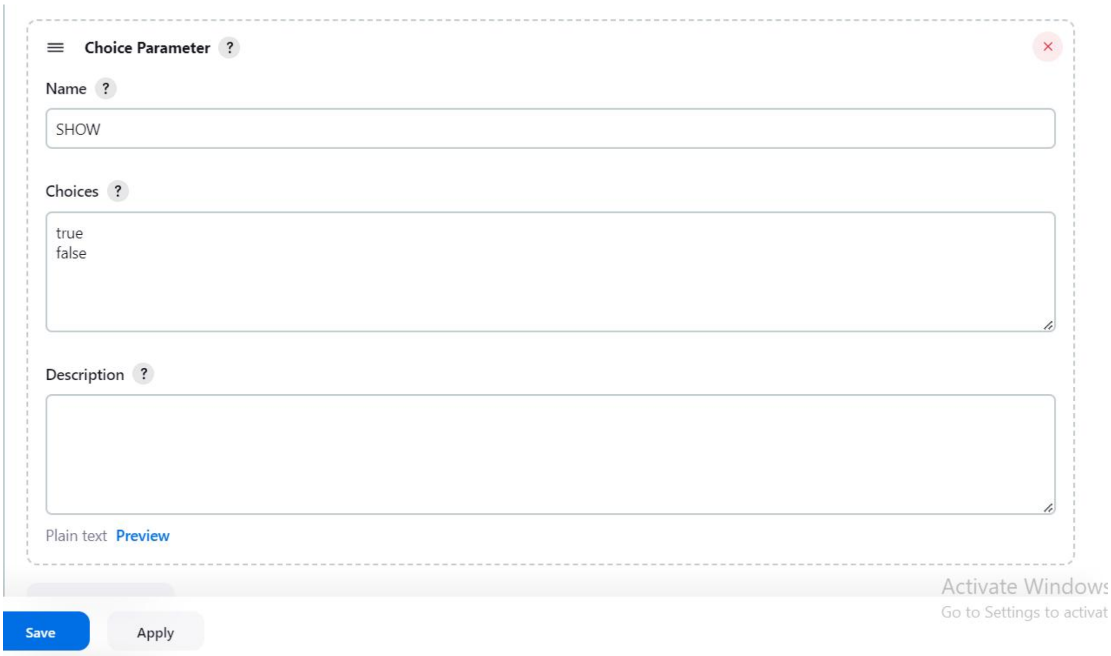


Name: LAST_NAME

Default Value: Shaikh

Description:

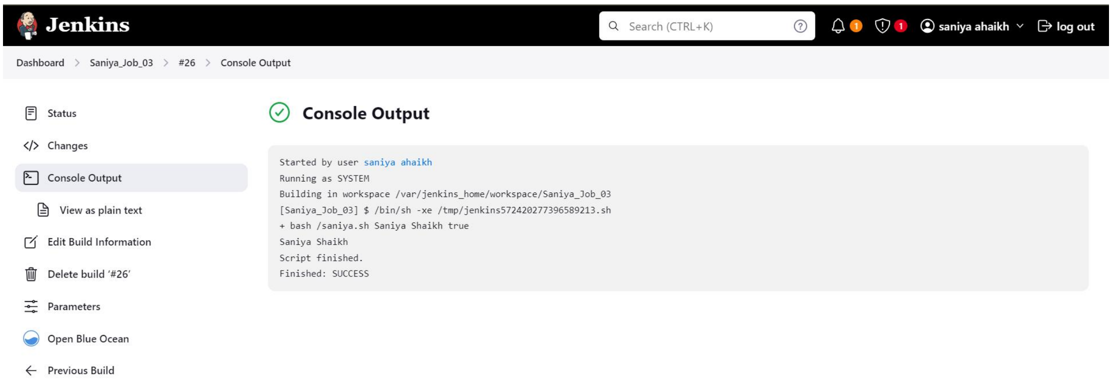
□ Trim the string



Name: SHOW

Choices:
true
false

Description:



Console Output

```
Started by user saniya ahaikh
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Saniya_Job_03
[Saniya_Job_03] $ /bin/sh -xe /tmp/jenkins5724202773965892l3.sh
+ bash /saniya.sh Saniya Shaikh true
Saniya Shaikh
Script finished.
Finished: SUCCESS
```