EXERCISE 09 – SETUP OF CONTINUOUS DELIVERY

SAP Partner Workshop	**30 min**

Description

In this exercise, you'll learn how

- To do setup of Continuous Delivery

For further reading on SAP Cloud SDK, click link below.
https://www.sap.com/germany/developer/topics/s4hana-cloud-sdk.html

Target group

- Developers
- People interested in learning about S/4HANA extension and SAP Cloud SDK

Goal

The goal of this exercise is to build a basic extension app on Cloud Foundry using Business Partner Example.

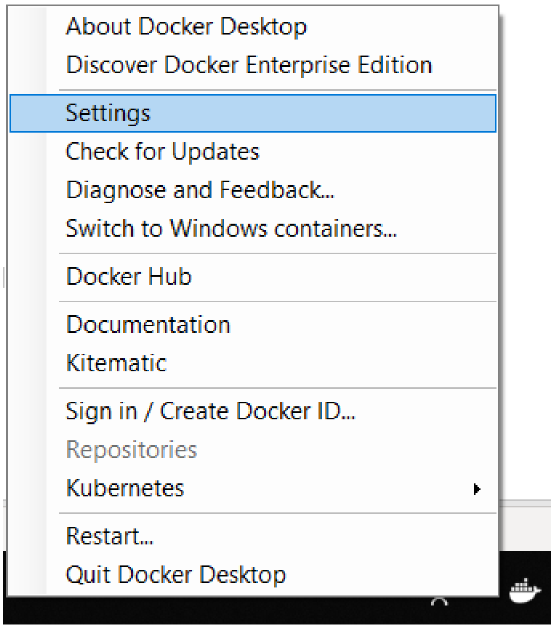
Prerequisites

Here below are prerequisites for this exercise.

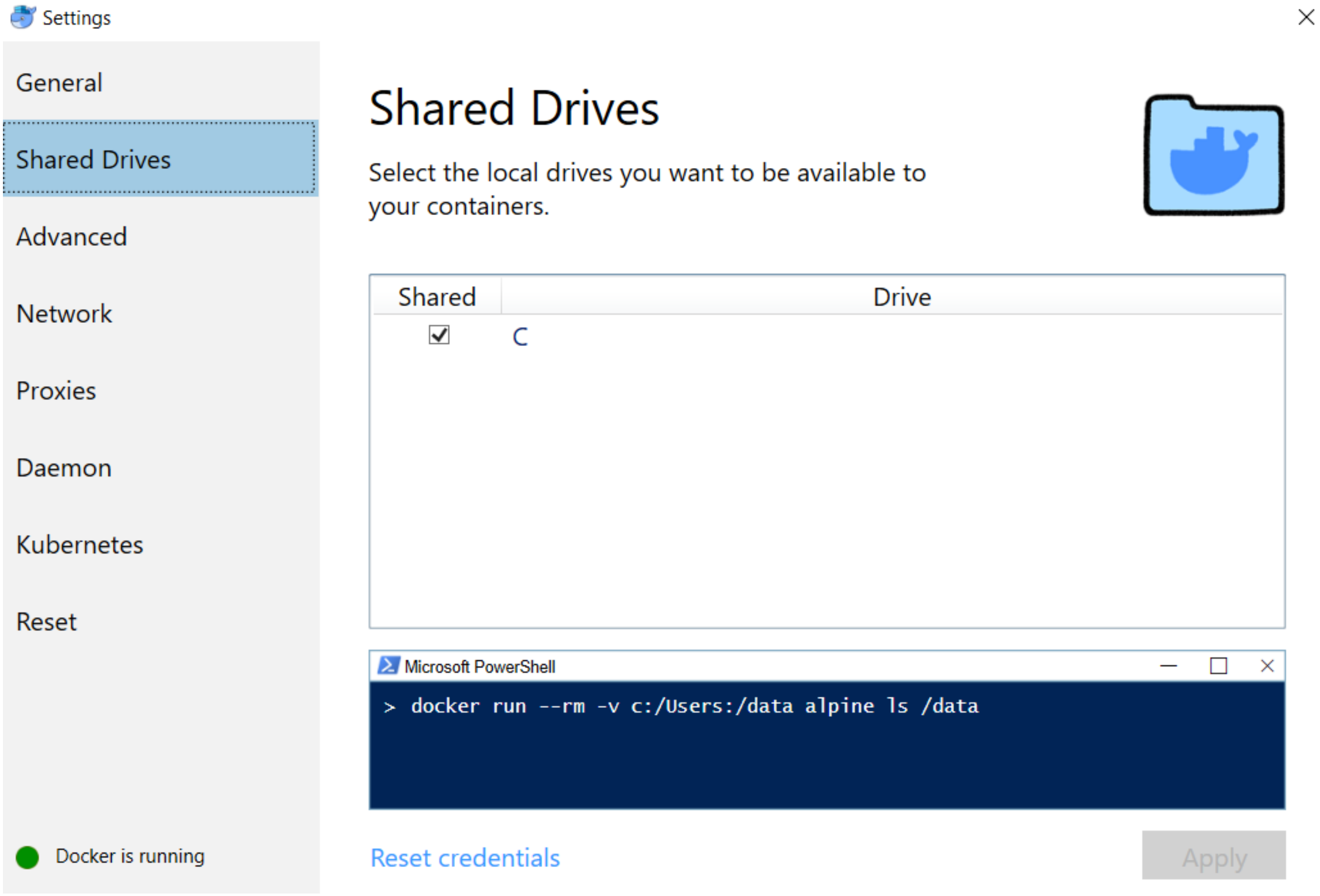
- Install Docker - https://docs.docker.com/docker-for-windows/install/

Step 1 – Run and Configure Docker

Run the docker. Go to Settings as below.



Share “C:\” drive as below.



Shared Drives

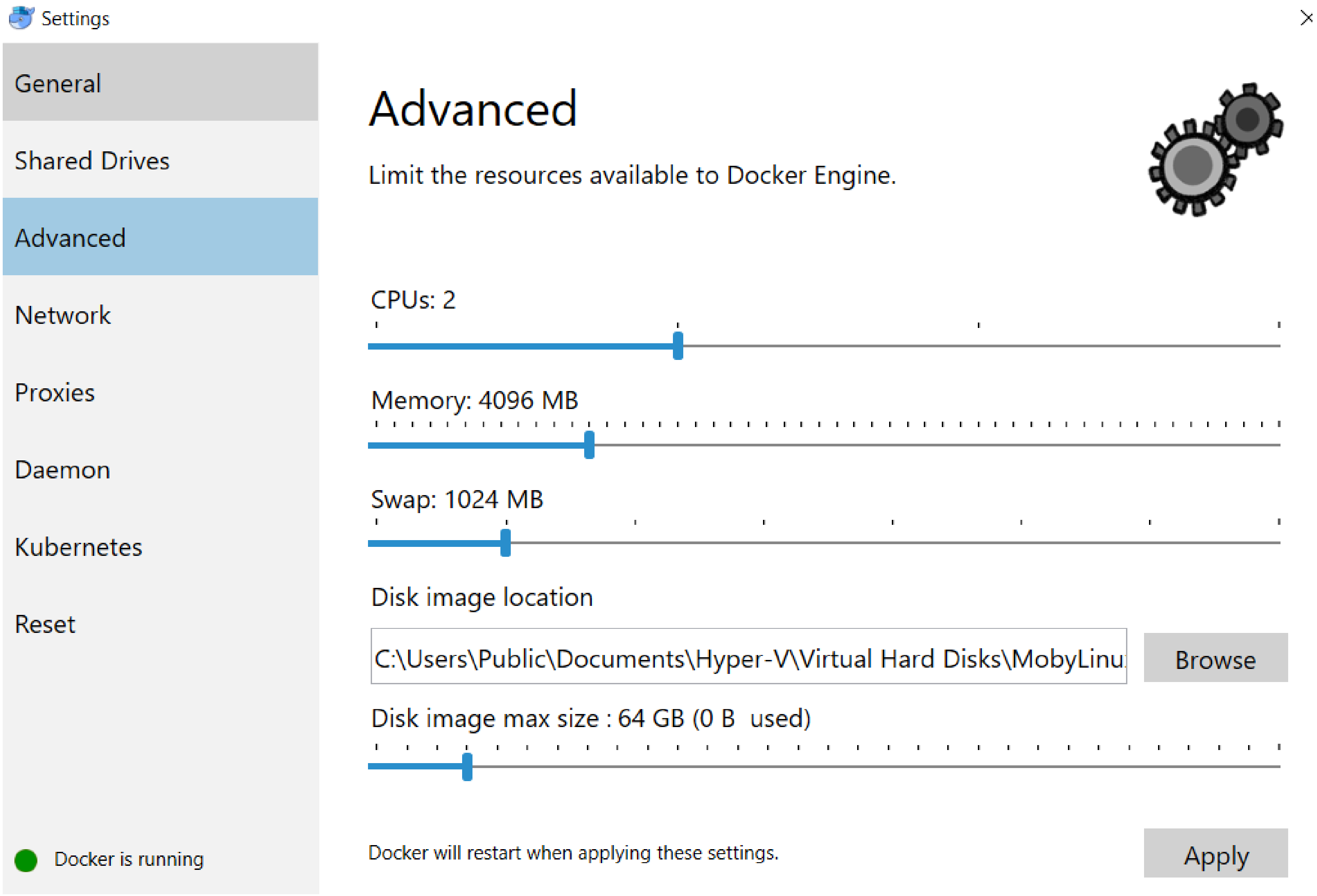
Select the local drives you want to be available to your containers.

Shared	Drive
☑	C

Microsoft PowerShell

```
> docker run --rm -v c:/Users:/data alpine ls /data
```

Go to **Advanced** and increase the memory to 4096 MB.



Advanced

Limit the resources available to Docker Engine.

CPUs: 2

Memory: 4096 MB

Swap: 1024 MB

Disk image location

C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\MobyLinu

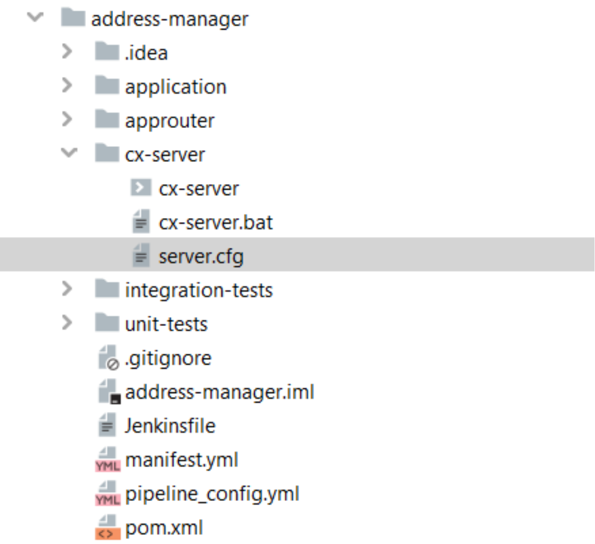
Disk image max size : 64 GB (0 B used)

Docker will restart when applying these settings.

Click on **Apply.**

Step 2 – Run Jenkins Server

The lifecycle of the Cx Server is maintained by a script called *cx-server*. It can be found in the same named folder on the root of each SAP Cloud SDK project archetype.



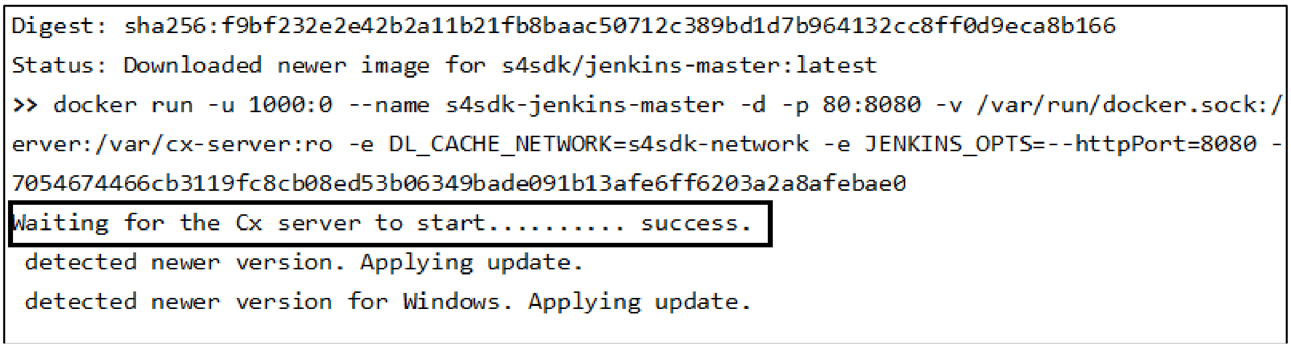
```
address-manager
  .idea
  application
  approuter
  cx-server
    cx-server
    cx-server.bat
    server.cfg
  integration-tests
  unit-tests
  .gitignore
  address-manager.iml
  Jenkinsfile
  manifest.yml
  pipeline_config.yml
  pom.xml
```

Note: If you may use the project from previous exercise or download a project sample from here.

Run the below command to start Jenkins server locally.

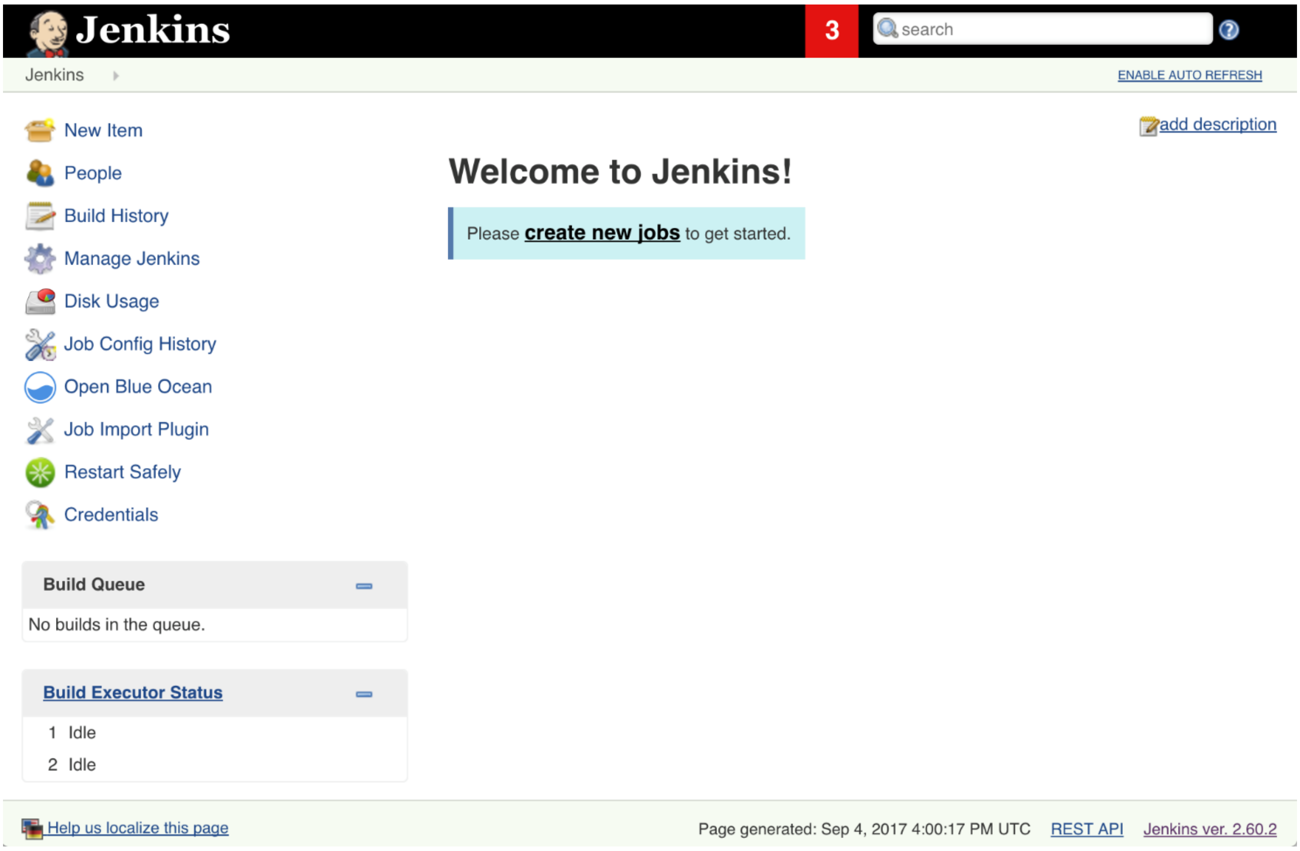
```
cd <your project location>\address-manager\cx-server
cx-server start
```

The output should be as below.



```
Digest: sha256:f9bf232e2e42b2a11b21fb8baac50712c389bd1d7b964132cc8ff0d9eca8b166
Status: Downloaded newer image for s4sdk/jenkins-master:latest
>> docker run -u 1000:0 --name s4sdk-jenkins-master -d -p 80:8080 -v /var/run/docker.sock:/
erver:/var/cx-server:ro -e DL_CACHE_NETWORK=s4sdk-network -e JENKINS_OPTS=--httpPort=8080 -
7054674466cb3119fc8cb08ed53b06349bade091b13afe6ff6203a2a8afebae0
Waiting for the Cx server to start.......... success.
 detected newer version. Applying update.
 detected newer version for Windows. Applying update.
```

Open a browser and type “localhost” and enter. You should get the Jenkins home page as below.



Welcome to Jenkins!

Please **create new jobs** to get started.

No builds in the queue.

1 Idle
2 Idle

Page generated: Sep 4, 2017 4:00:17 PM UTC

For stopping your Cx Server instance, you can use the command *cx-server stop*. It will stop the Cx Server container in a safe manner by waiting for all jobs to finish before shutdown.

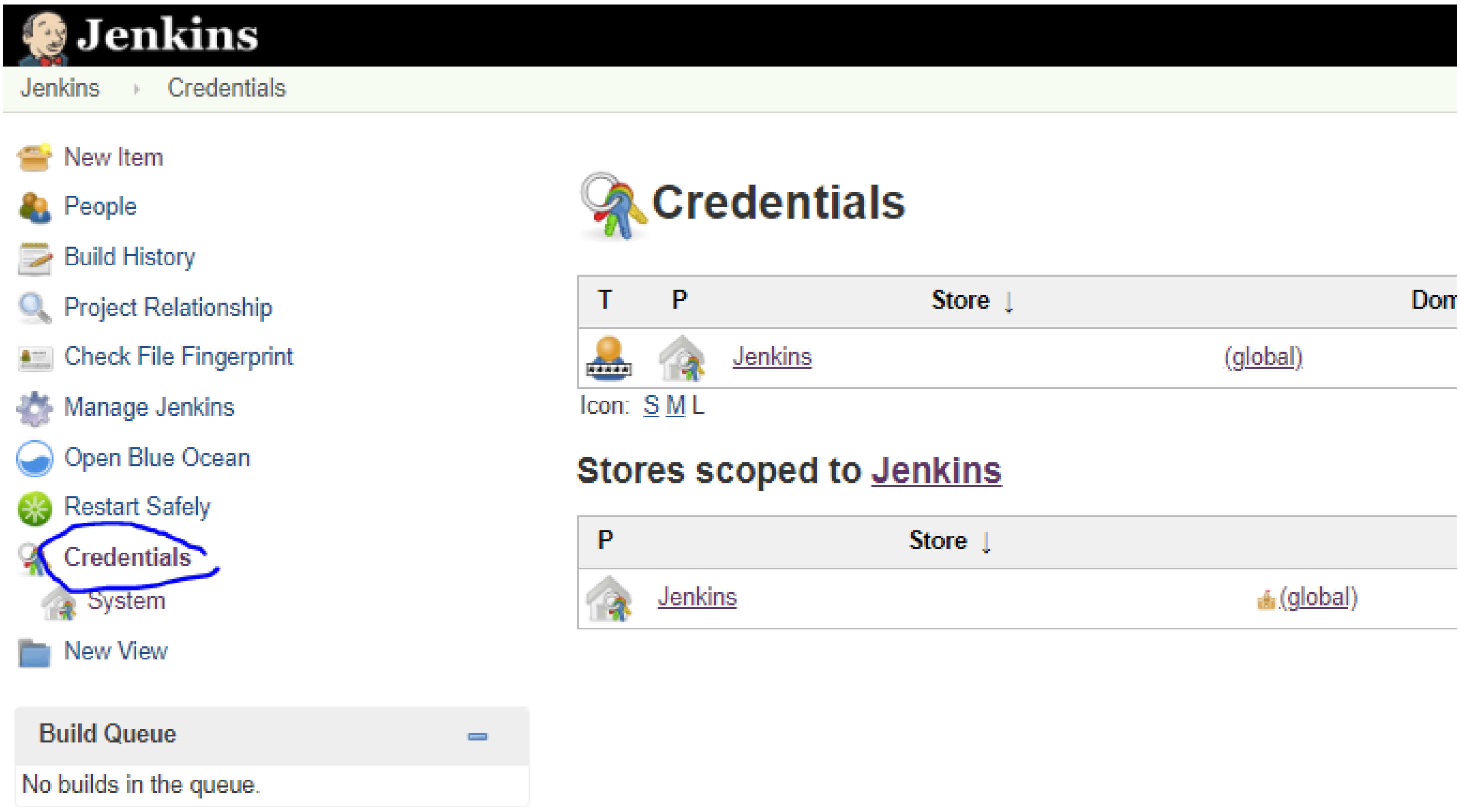
Step 3 – Create Credential in Jenkins

To create a job in Jenkins, we will use the source code repository from Github.com. You can use any code repository.

For this exercise, you may use this repository - https://github.com/rajagupta20/address-manager-for-pipeline

Signup at https://github.com and create your user.

First create a credential for GitHub as below.



Credentials

T	P	Store ↓	Dom
		Jenkins	(global)

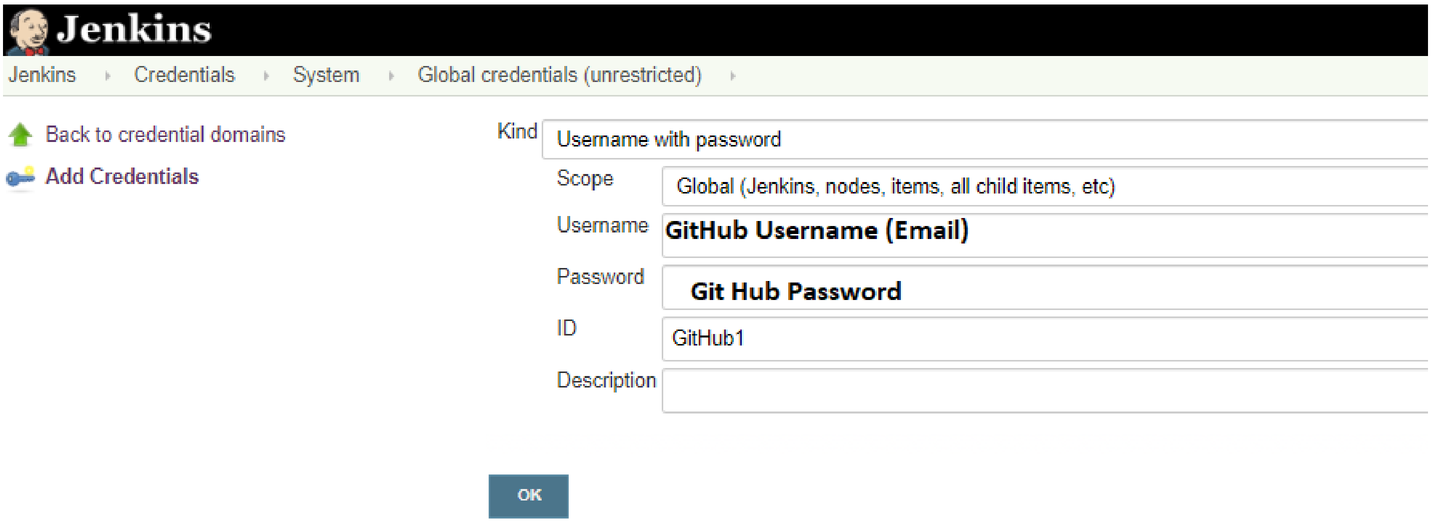
Icon: S M L

Stores scoped to Jenkins

P	Store ↓	
	Jenkins	(global)

Build Queue

No builds in the queue.



Jenkins › Credentials › System › Global credentials (unrestricted) ›

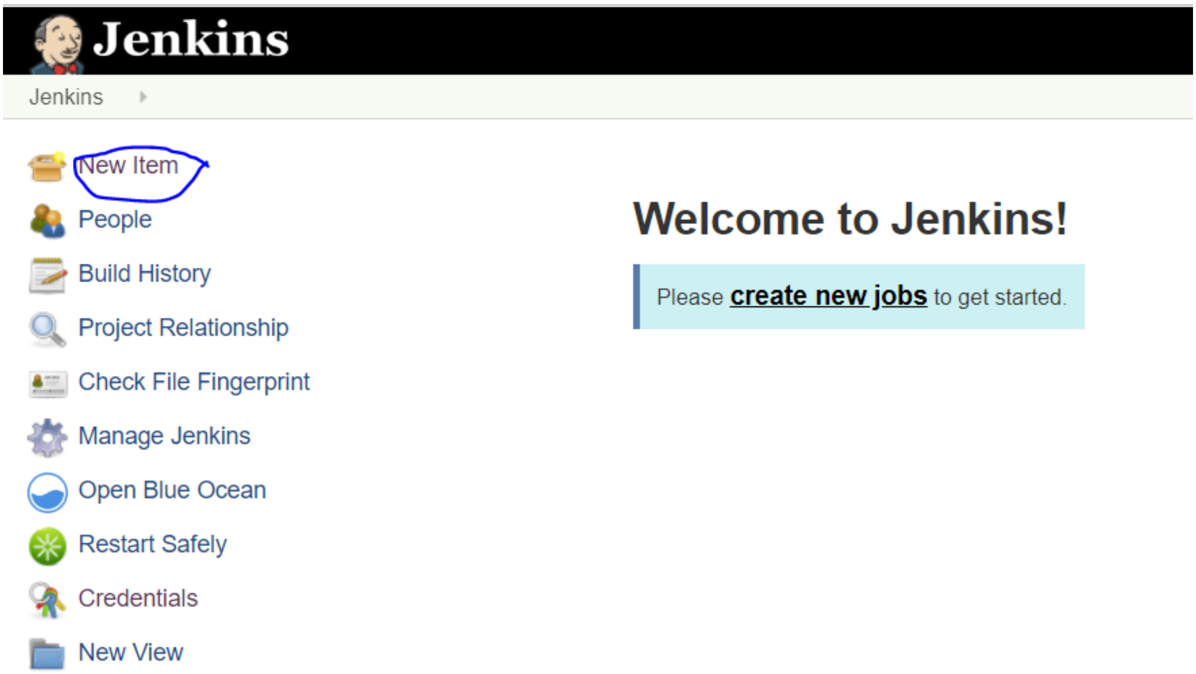
Back to credential domains

Add Credentials

Kind	Username with password
Scope	Global (Jenkins, nodes, items, all child items, etc)
Username	GitHub Username (Email)
Password	Git Hub Password
ID	GitHub1
Description	

Step 4 – Create a new Job and build Pipeline

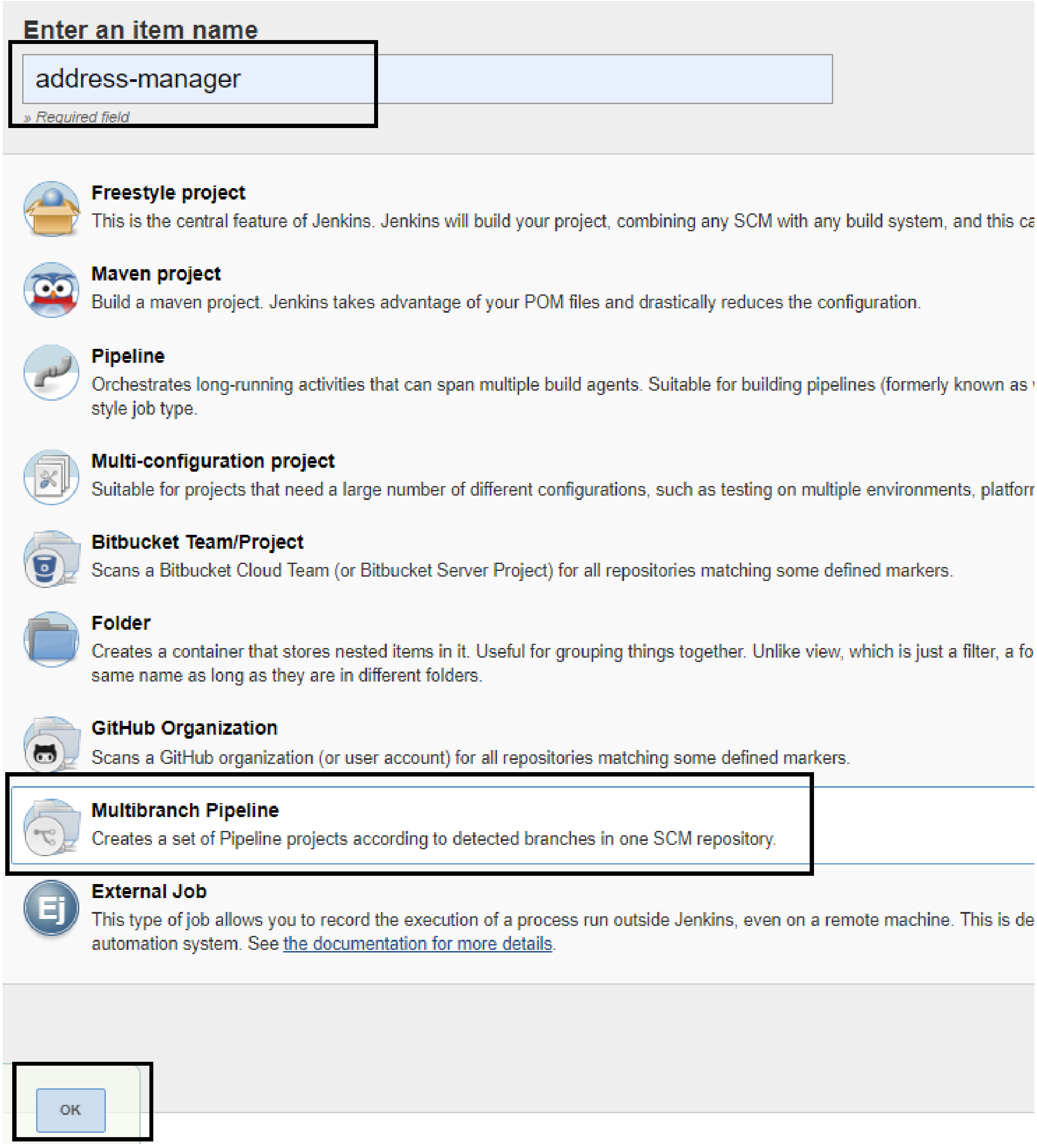
Create a new job as below.



Jenkins ›

Welcome to Jenkins!

Please **create new jobs** to get started.



Enter an item name

address-manager

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this ca

Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as style job type.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platforr

Bitbucket Team/Project

Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a fo same name as long as they are in different folders.

GitHub Organization

Scans a GitHub organization (or user account) for all repositories matching some defined markers.

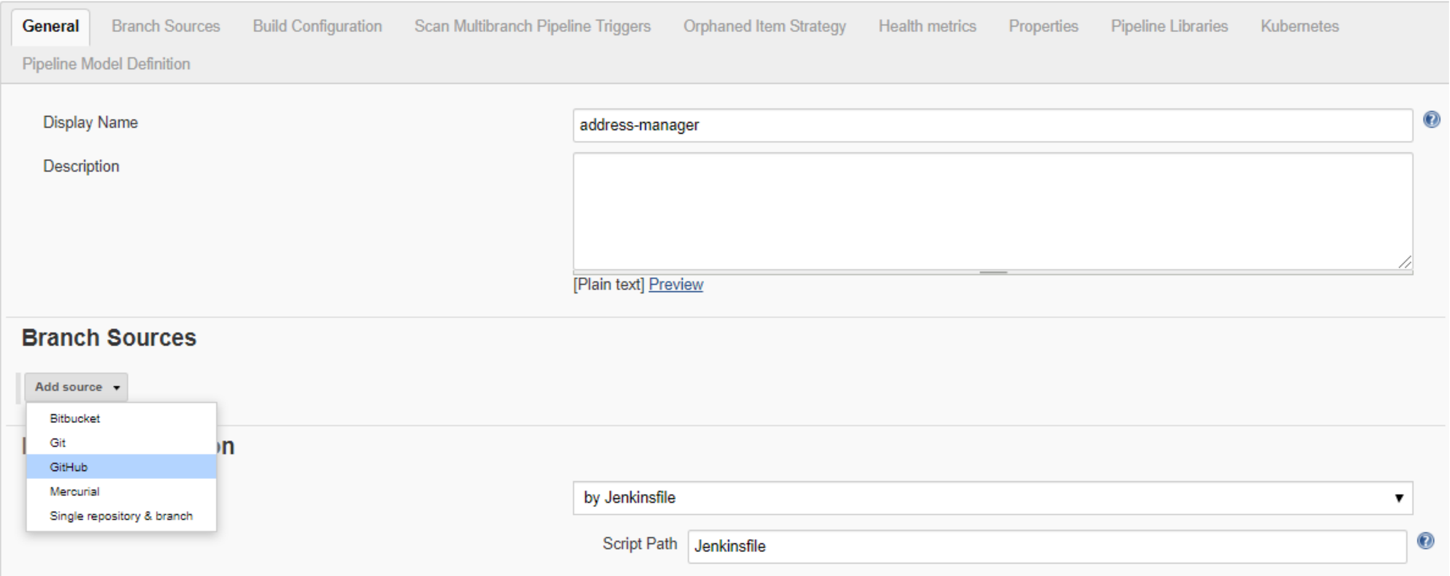
Multibranch Pipeline

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

External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is de automation system. See the documentation for more details.

Enter **Branch Source** as GitHub.



Display Name: address-manager

Description

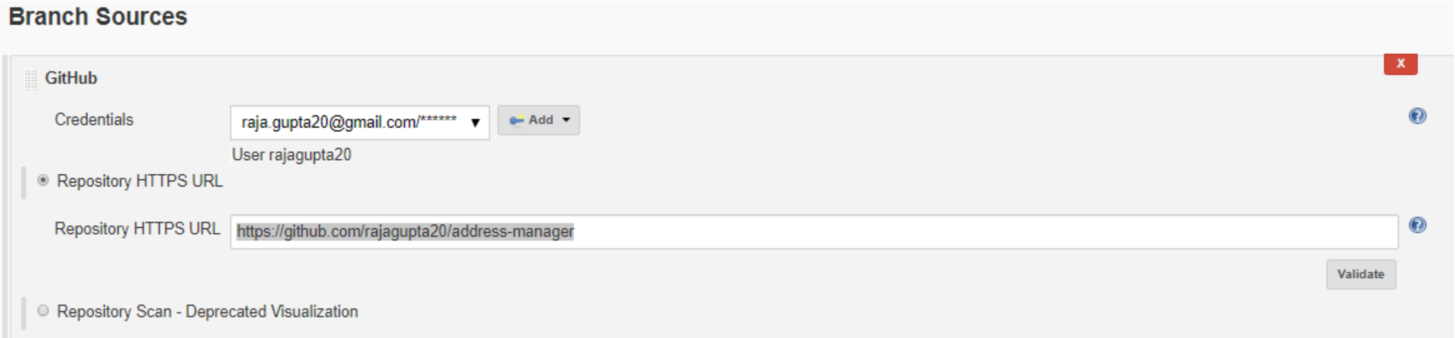
Branch Sources

by Jenkinsfile

Script Path: Jenkinsfile

Select the credential you created in previous step.

Enter the repository URL as https://github.com/rajagupta20/address-manager



Branch Sources

GitHub

Credentials: raja.gupta20@gmail.com/******

User rajagupta20

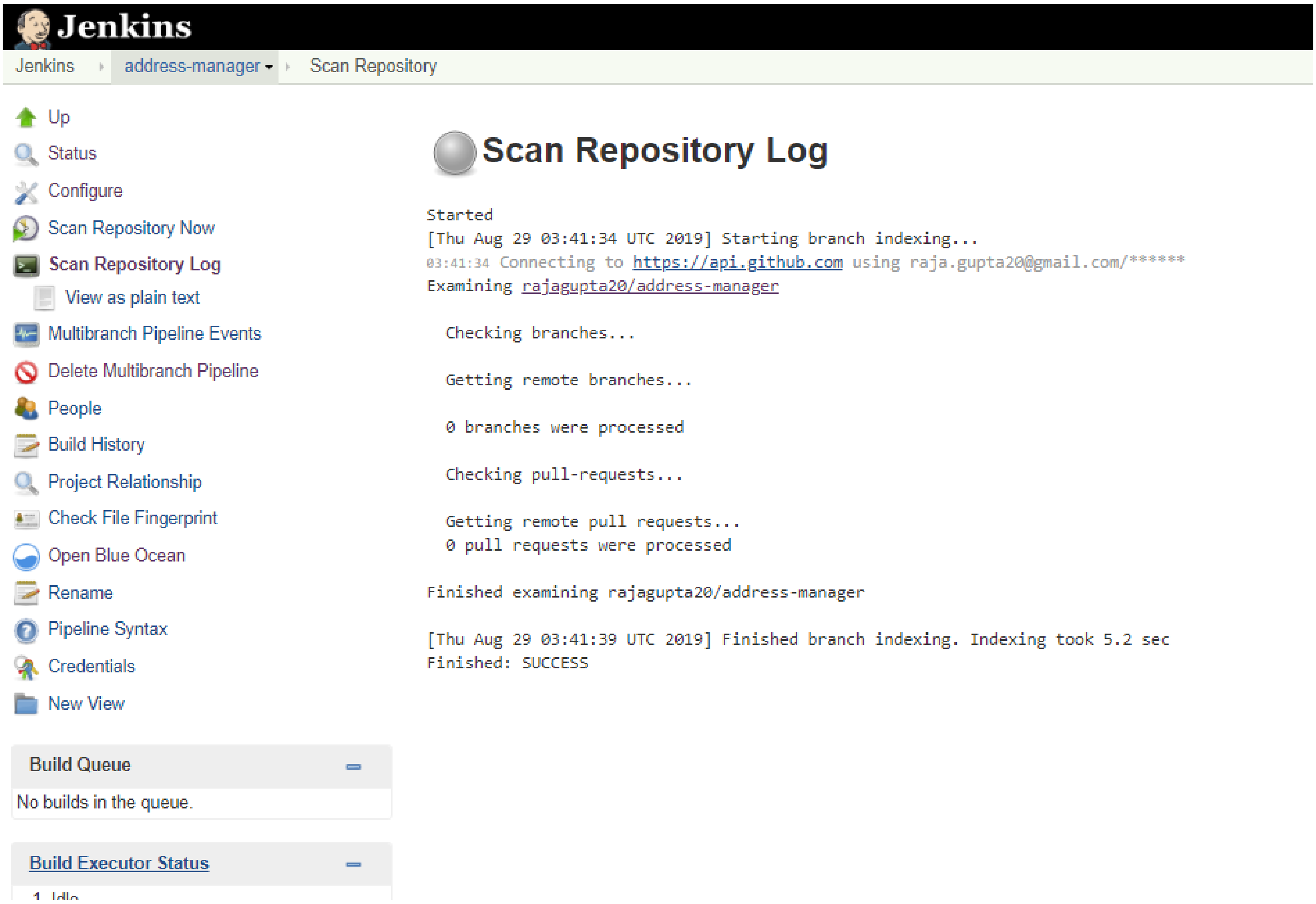
● Repository HTTPS URL

Repository HTTPS URL: https://github.com/rajagupta20/address-manager

○ Repository Scan - Deprecated Visualization

Click on **Save**.

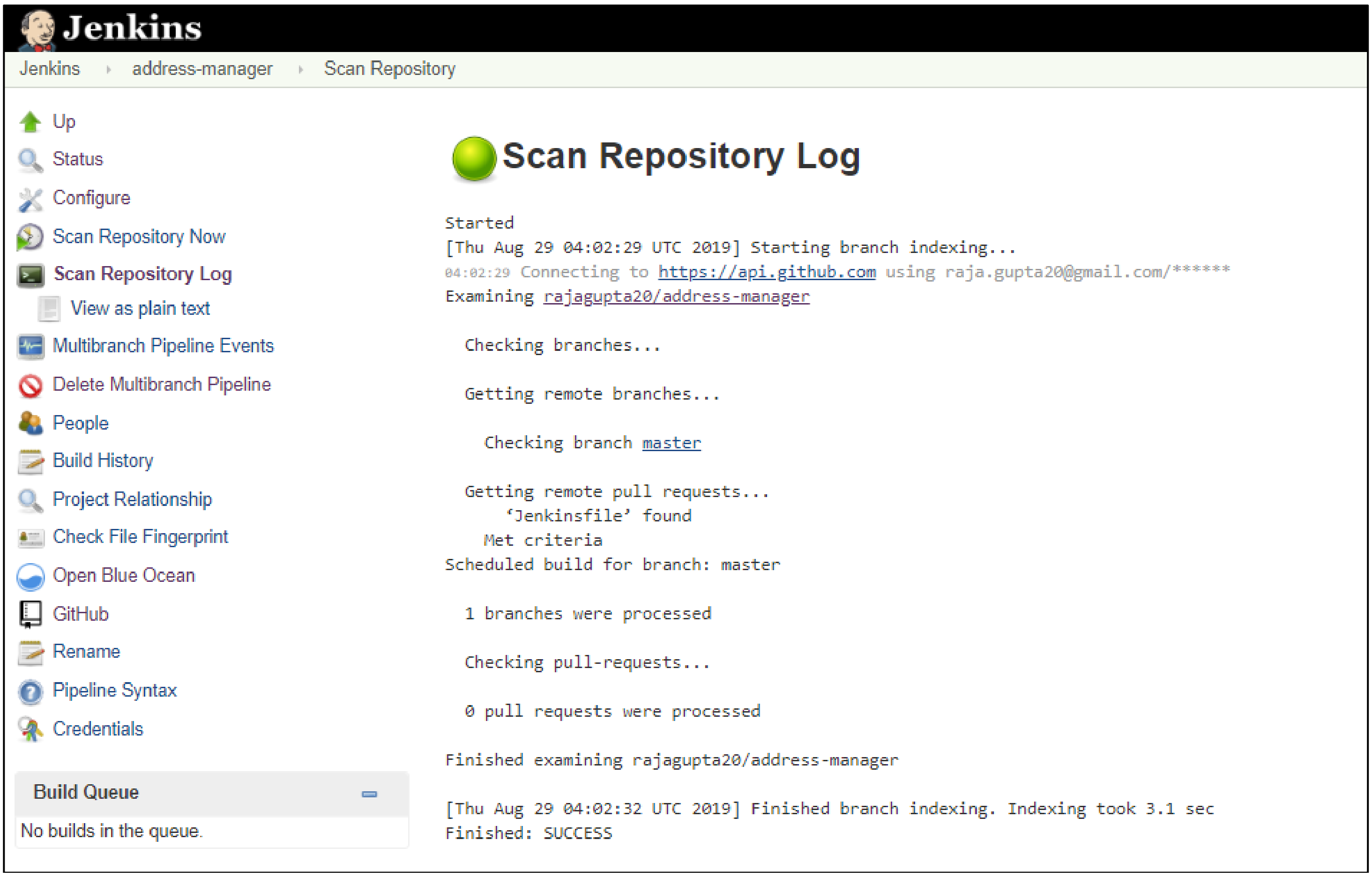
You will see the build screen as below.



Scan Repository Log

```
Started
[Thu Aug 29 03:41:34 UTC 2019] Starting branch indexing...
03:41:34 Connecting to https://api.github.com using raja.gupta20@gmail.com/******
Examining rajagupta20/address-manager

  Checking branches...

  Getting remote branches...

  0 branches were processed

  Checking pull-requests...

  Getting remote pull requests...
  0 pull requests were processed

Finished examining rajagupta20/address-manager

[Thu Aug 29 03:41:39 UTC 2019] Finished branch indexing. Indexing took 5.2 sec
Finished: SUCCESS
```

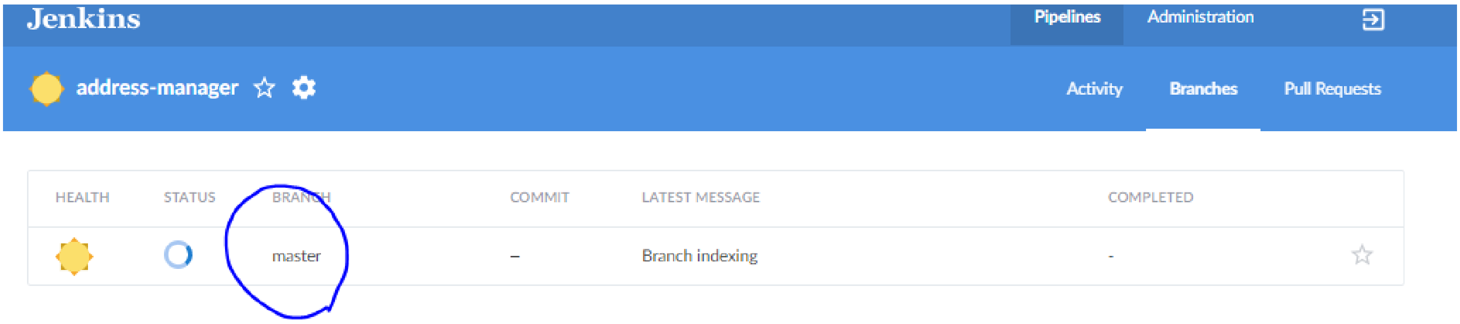
Finally the job should be created as below.



Scan Repository Log

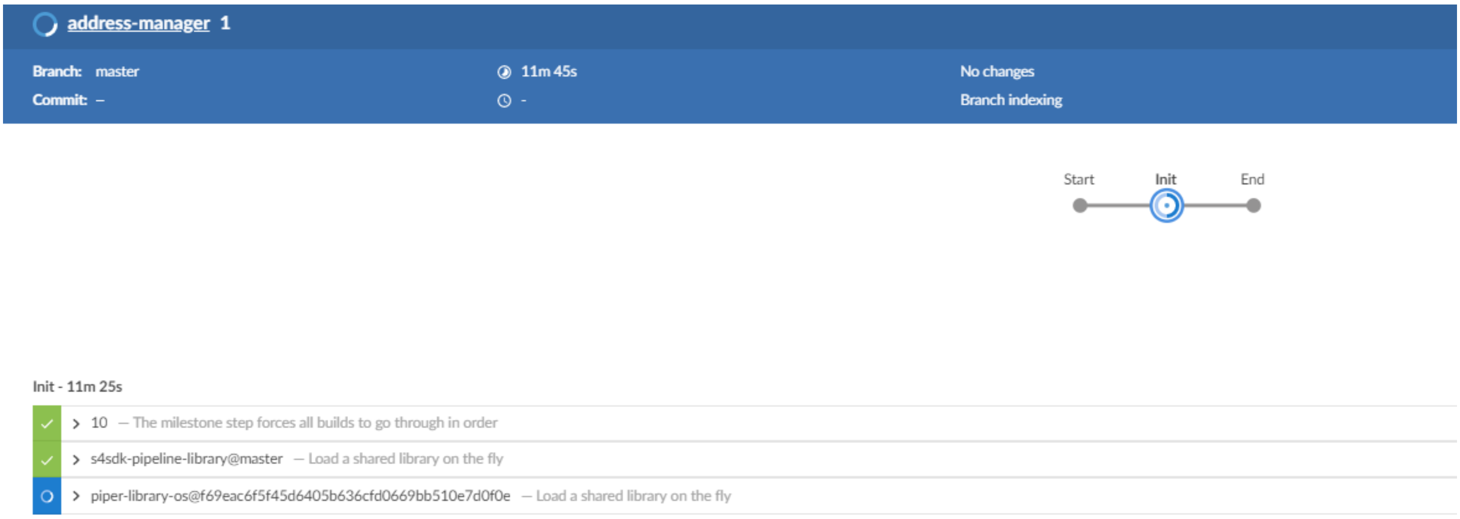
```
Started
[Thu Aug 29 04:02:29 UTC 2019] Starting branch indexing...
04:02:29 Connecting to https://api.github.com using raja.gupta20@gmail.com/******
Examining rajagupta20/address-manager

  Checking branches...

  Getting remote branches...

    Checking branch master

  Getting remote pull requests...
      'Jenkinsfile' found
    Met criteria
Scheduled build for branch: master

  1 branches were processed

  Checking pull-requests...

  0 pull requests were processed

Finished examining rajagupta20/address-manager

[Thu Aug 29 04:02:32 UTC 2019] Finished branch indexing. Indexing took 3.1 sec
Finished: SUCCESS
```

Click on the **Open Blue Ocean**. It should open the screen as below. Click on the branch.



HEALTH	STATUS	BRANCH	COMMIT	LATEST MESSAGE	COMPLETED
☀	○	master	–	Branch indexing	-

It will show the pipeline build information as below.



address-manager 1

Branch: master
Commit: –

11m 45s
-

No changes
Branch indexing

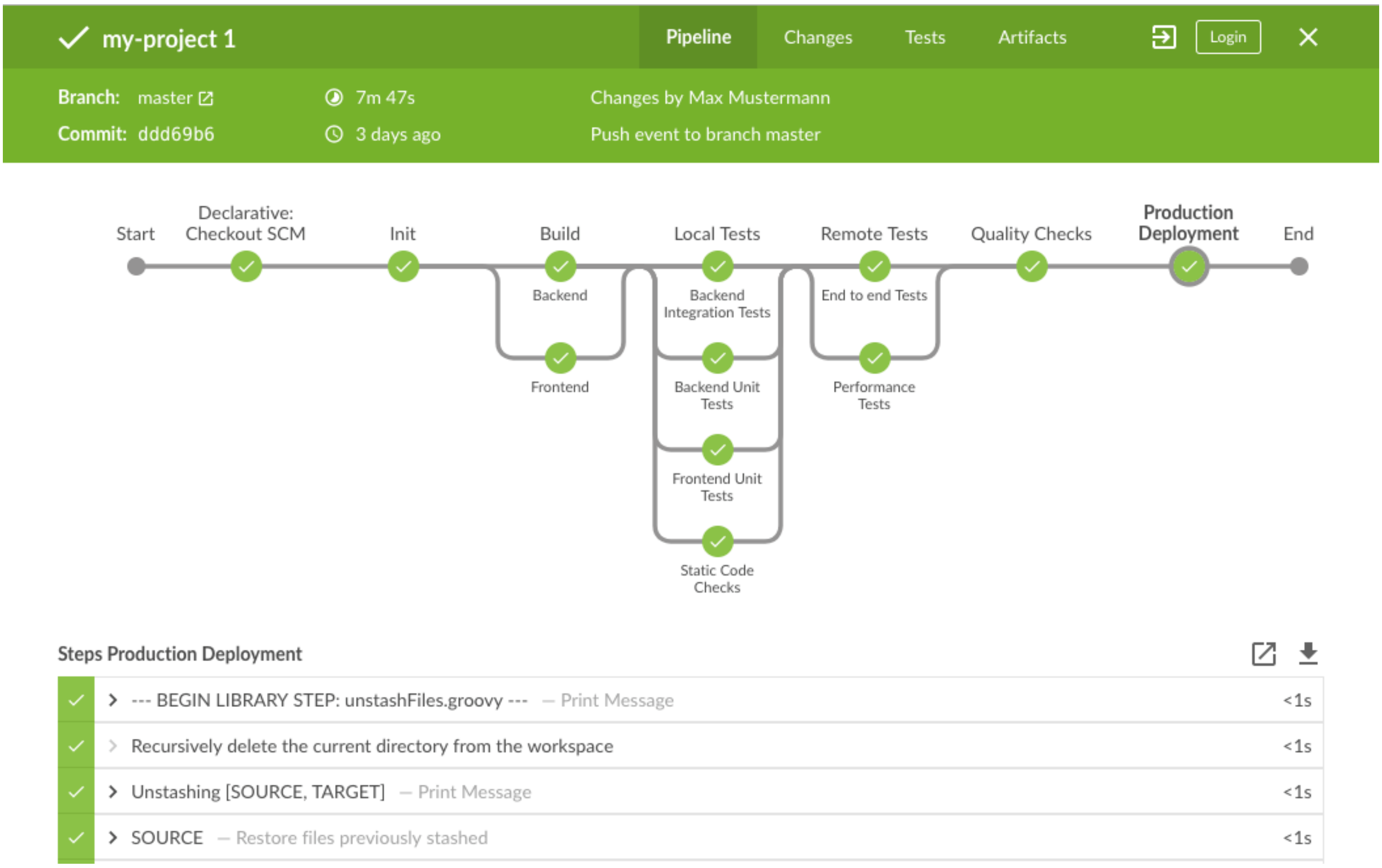
Init - 11m 25s

✓ 10 — The milestone step forces all builds to go through in order

✓ s4sdk-pipeline-library@master — Load a shared library on the fly

○ piper-library-os@f69eac6f5f45d6405b636cfd0669bb510e7d0f0e — Load a shared library on the fly

Finally you should get the complete pipeline built as below.



my-project 1

Branch: master
Commit: ddd69b6

7m 47s
3 days ago

Changes by Max Mustermann
Push event to branch master

Steps Production Deployment

✓ --- BEGIN LIBRARY STEP: unstashFiles.groovy --- — Print Message <1s

✓ Recursively delete the current directory from the workspace <1s

✓ Unstashing [SOURCE, TARGET] — Print Message <1s

✓ SOURCE — Restore files previously stashed <1s

Congratulation! You have successfully setup Continuous Delivery.