Jenkins: Introduction - Understanding continuous integration

- Developers pushes software to a repository
- Operations builds and deploys the application to one or many environments like testing staging)
- QA team performs performance tests and releases it to production use

- Continuous Integration (CI) is the process of **automating** the build and testing of code every time a team member **commits** changes to Version Control System

Jenkins: Introduction - Understanding continuous integration

1. Developers pushes software to a repository
2. Operations builds and deploys the application to one or many environments like testing staging)
3. QA team performs performance tests and releases it to production use

Can automate most of the repetitive tasks! This facilitates continuous integration!

Jenkins: Introduction - Understanding continuous integration

A Build Pipeline Components

1. Unit Test
2. Acceptance Test
3. Packaging
4. Reporting
5. Deployment
6. Notification/Alerts

Can automate most of the repetitive tasks! This facilitates continuous integration!

Jenkins: Introduction - Understanding continuous integration

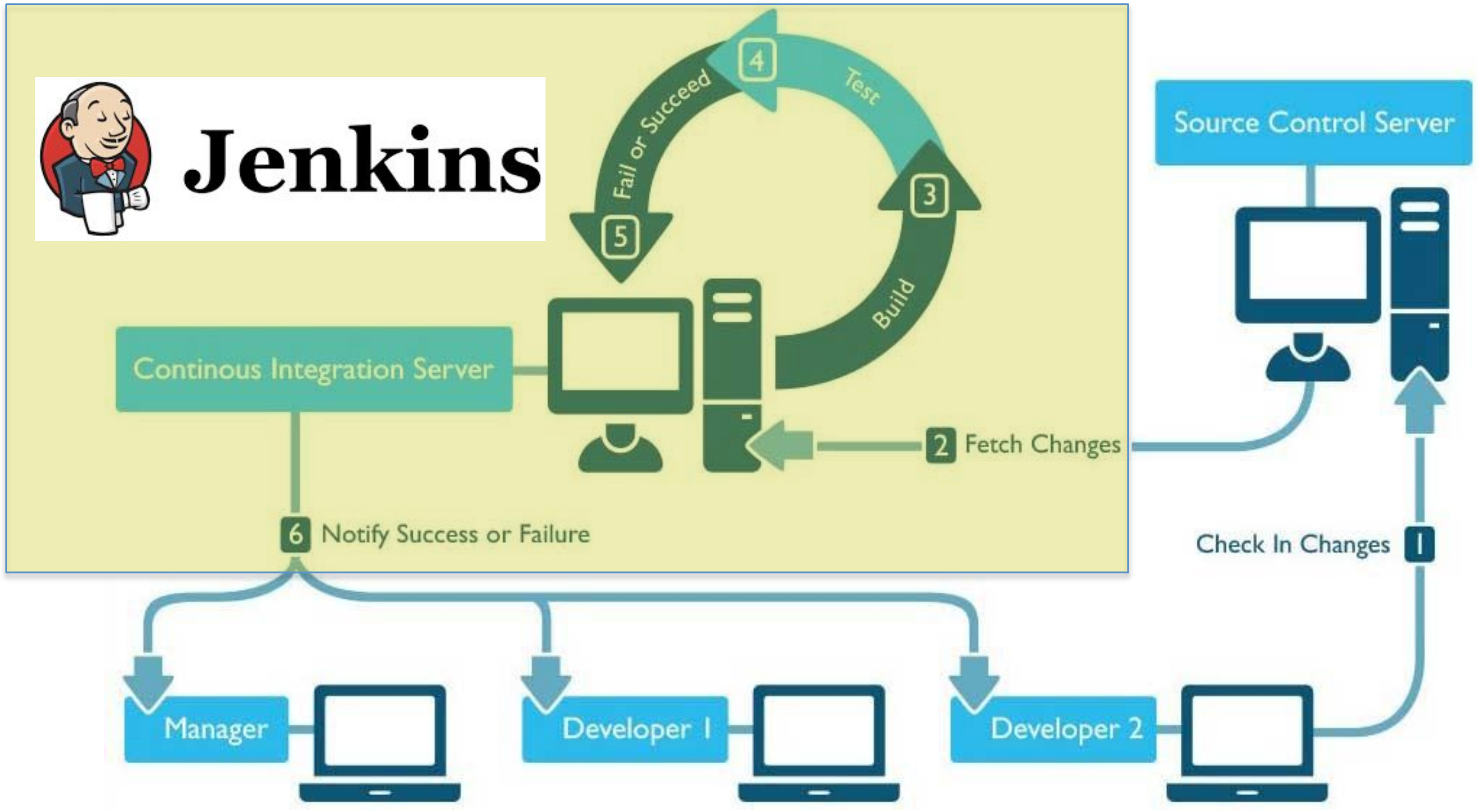
TeamCity

BAMBOO

Travis CI

Jenkins

Jenkins: Introduction - Understanding continuous integration

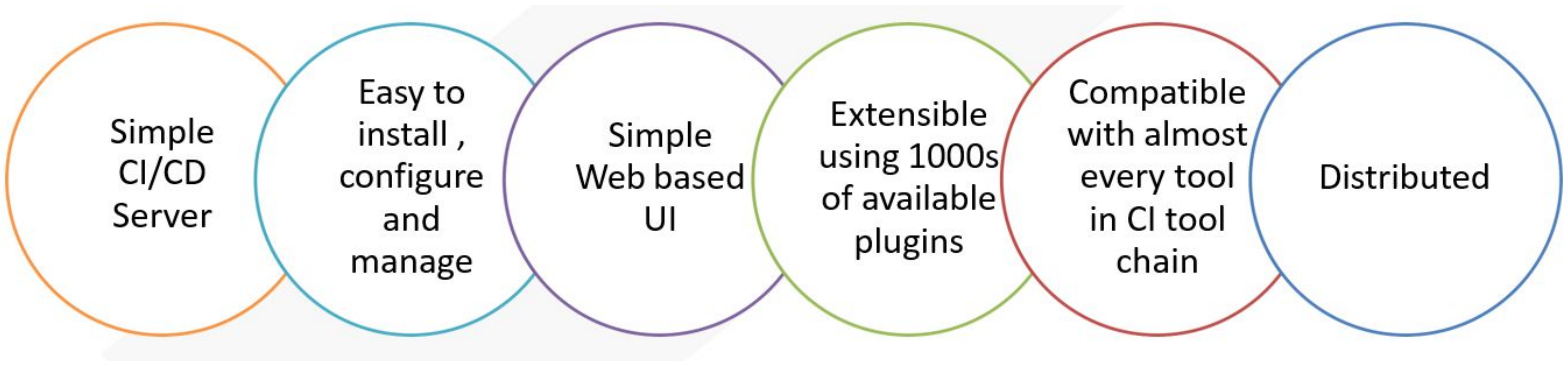


Jenkins: Introduction - Introduction about Jenkins

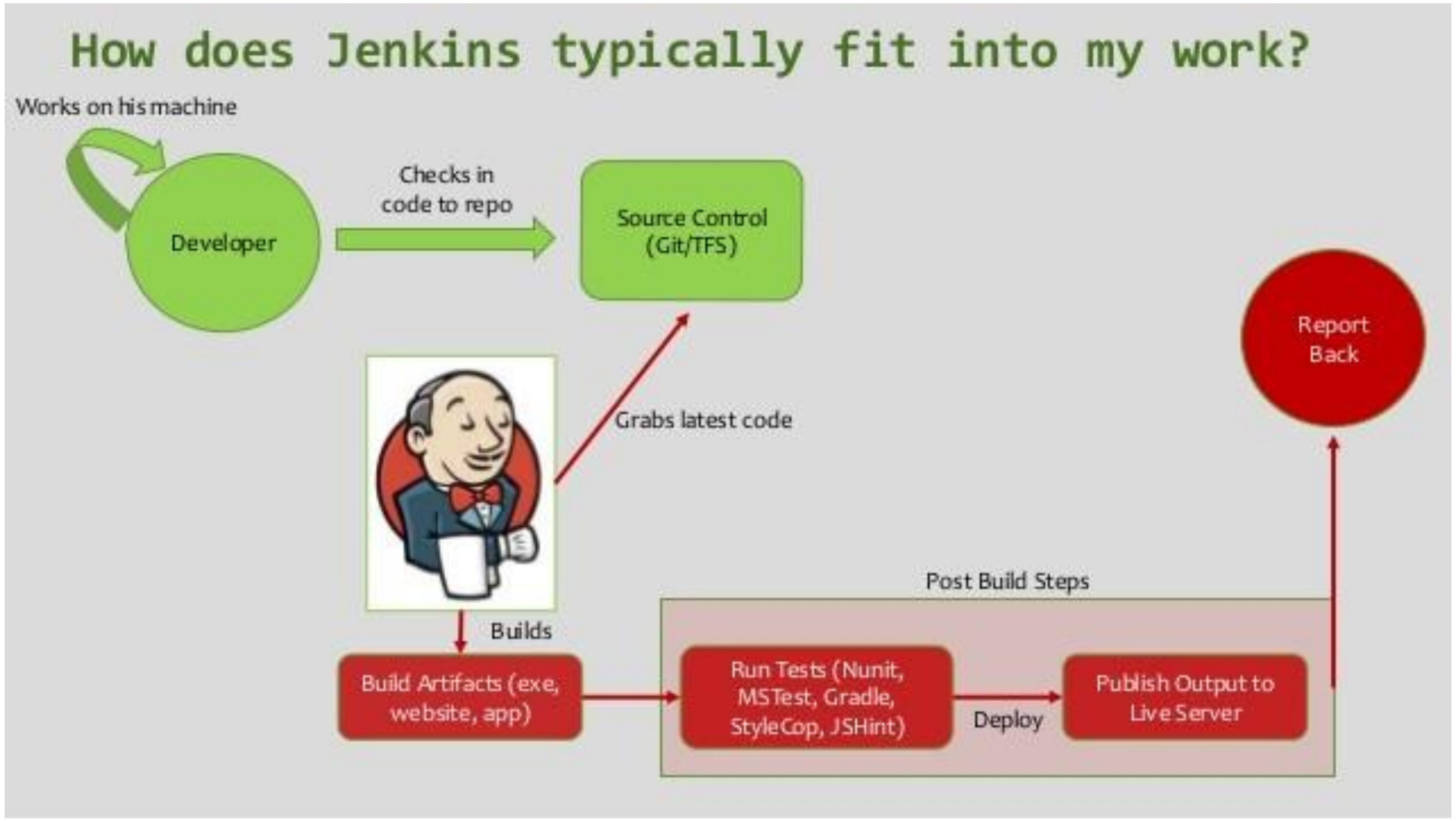
Jenkins is an open source automation server written in Java. Jenkins helps to automate the non-human part of software development process, with continuous integration and facilitates technical aspects of continuous delivery.

It is a server-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and can execute Apache Ant, Apache Maven and sbt based projects as well as arbitrary shell scripts and Windows batch commands.

Jenkins: Features


Simple CI/CD Server
Easy to install , configure and manage
Simple Web based UI
Extensible using 1000s of available plugins
Compatible with almost every tool in CI tool chain
Distributed

Jenkins: Introduction - Introduction about Jenkins


How does Jenkins typically fit into my work?
Works on his machine
Developer
Checks in code to repo
Source Control (Git/TFS)
Grabs latest code
Report Back
Builds
Post Build Steps
Build Artifacts (exe, website, app)
Run Tests (Nunit, MSTest, Gradle, StyleCop, JSHint)
Deploy
Publish Output to Live Server

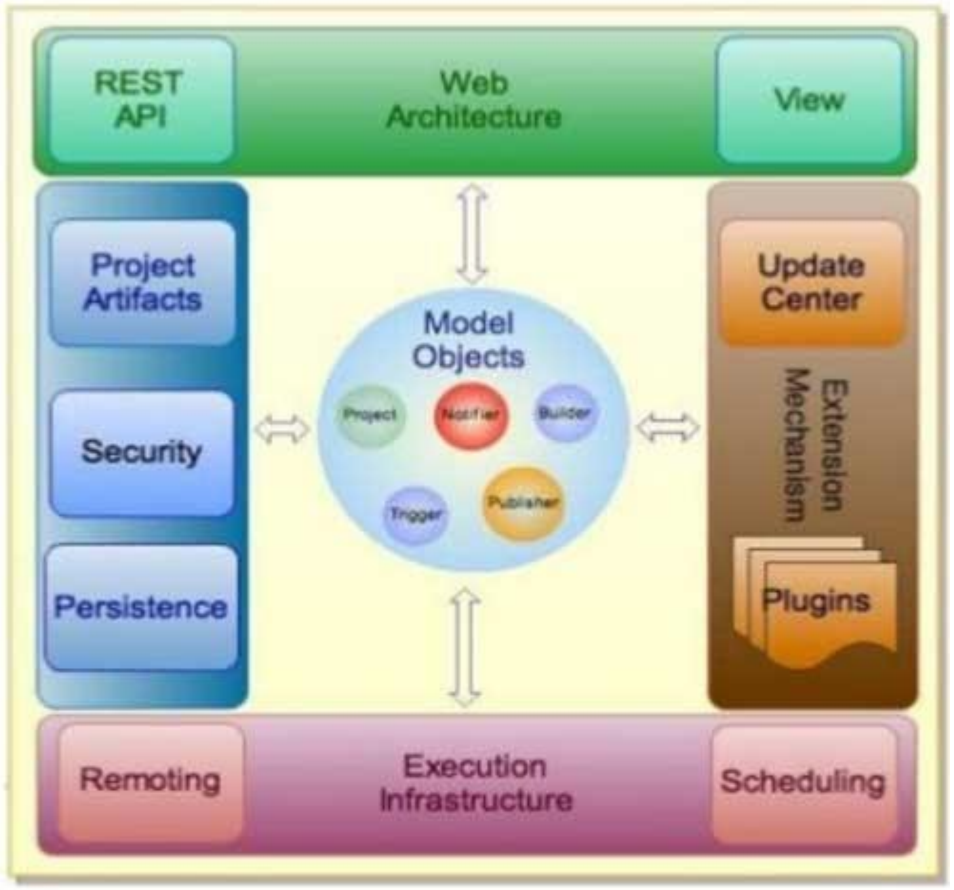
Jenkins: Introduction - Build Lifecycle

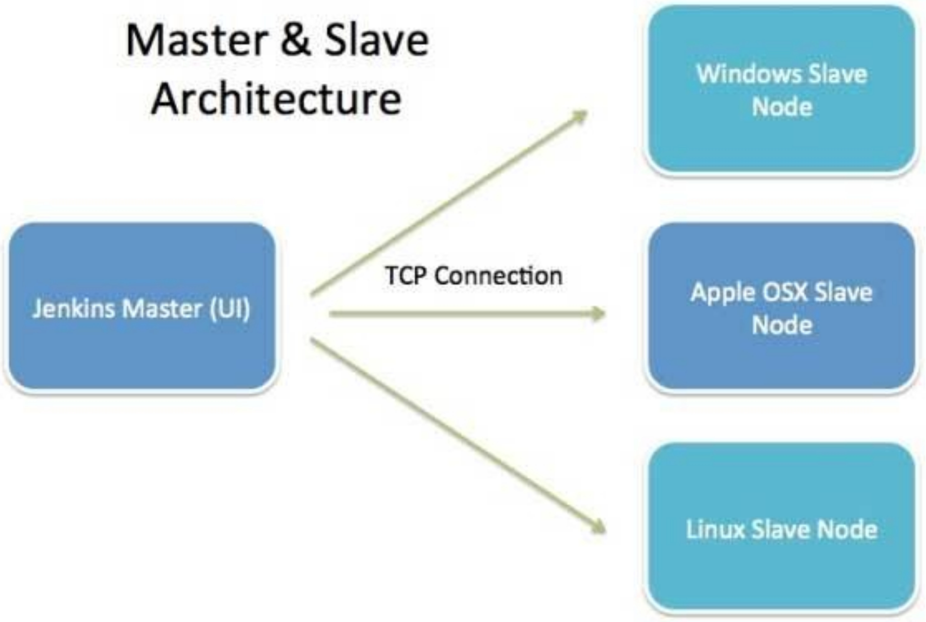
A build lifecycle is the process of building and distributing a particular artifact (project).

Build Lifecycle phases:

- **validate** - validates the project
- **compile** - compiles the source code of the project
- **test** - tests the compiled source code using a suitable unit testing framework.
- **package** - take the compiled code and package it in a distributable format
- **verify** – run integration tests
- **install** - install the package into the local repository (may be for dependencies)
- **deploy** - copies the final package to the remote repository for sharing with others

Jenkins: Introduction - Jenkins Architecture





Jenkins: Introduction - Jenkins Architecture

Master:
- Schedule Build Job
- Dispatches Builds to the Slave for Actual job Execution
- Monitoring the Slave and recording the build Results

Slave :
- Execute Builds jobs dispatched by master

Jenkins: Installation - Obtaining and installing Jenkins

Url: https://jenkins.io/doc/book/getting-started/installing/

Debian/Ubuntu

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/
sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins
```

MacOS [installer package also available]

```
brew install jenkins
```

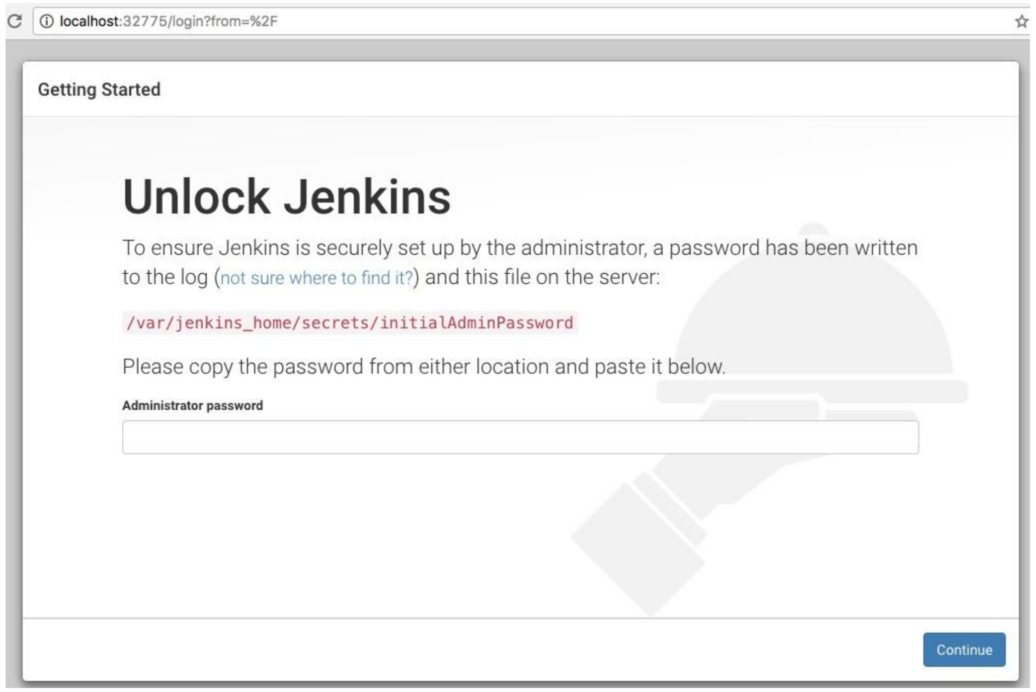
Docker

```
docker pull jenkins/jenkins
```

Windows

Using installer package

Jenkins: Installing


localhost:32775/login?from=%2F

Getting Started

Unlock Jenkins

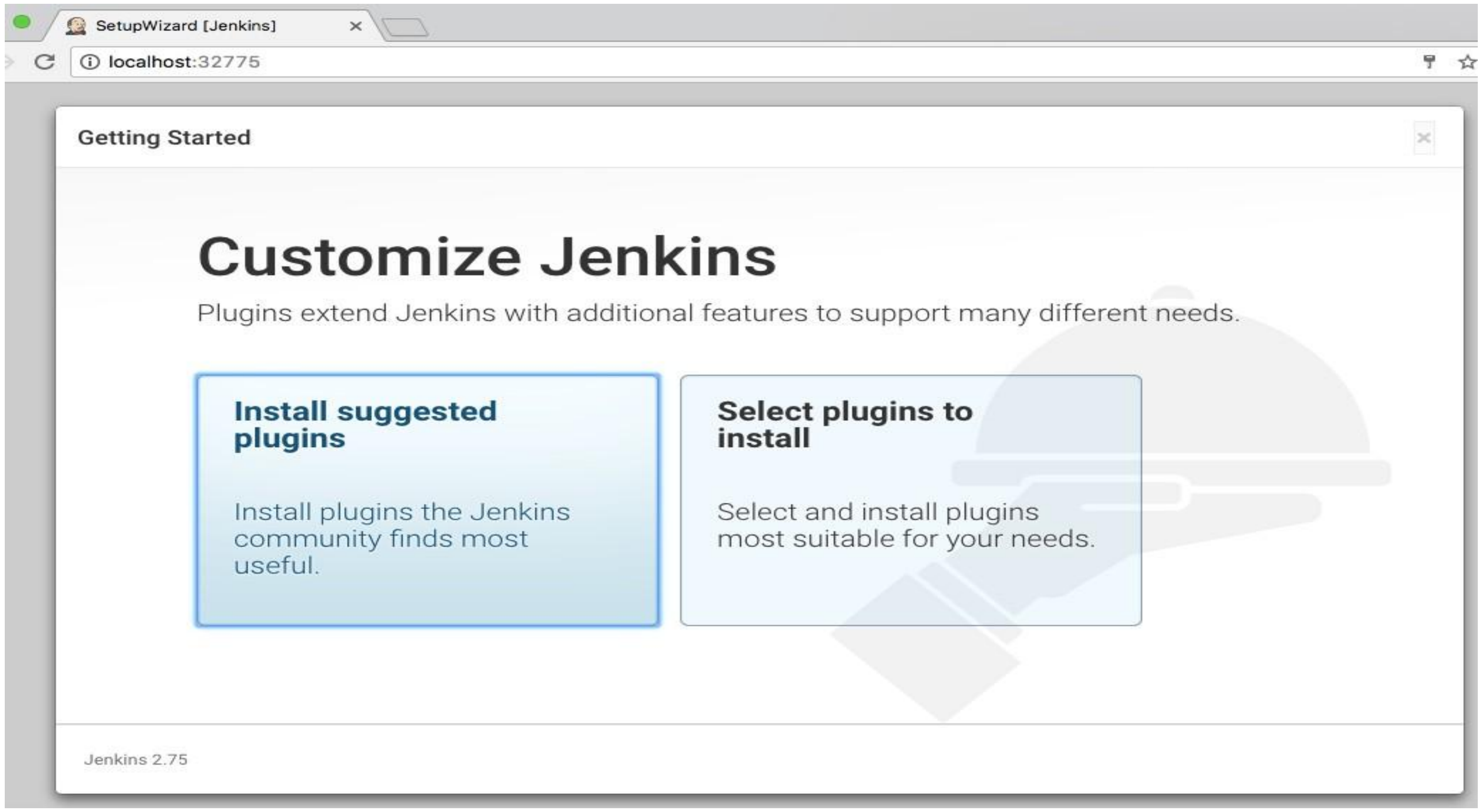
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/var/jenkins_home/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Jenkins: Installation - Docker + Jenkins: Browsing



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

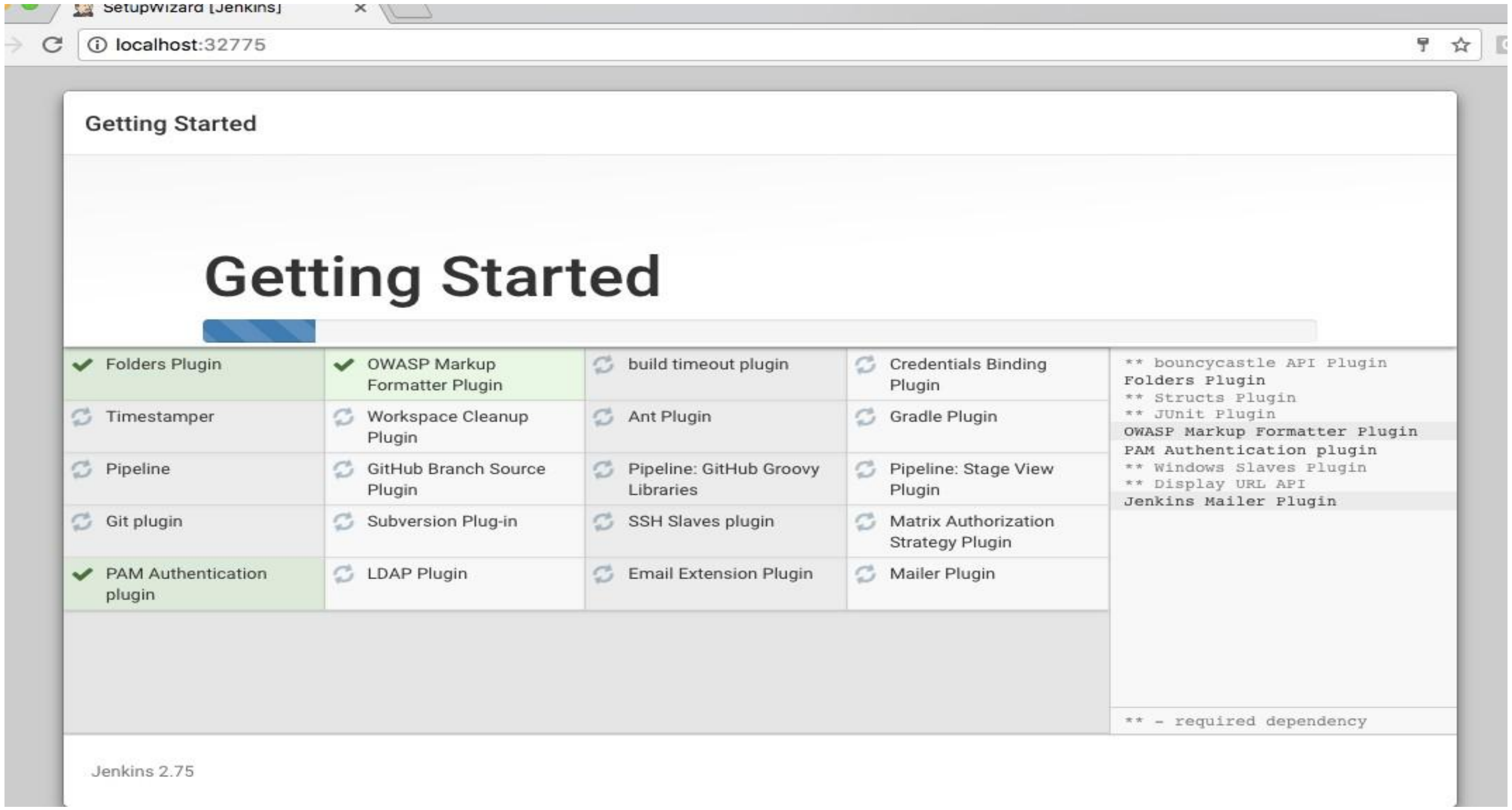
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.75

Jenkins: Installation – Install Suggested Plugins



Getting Started

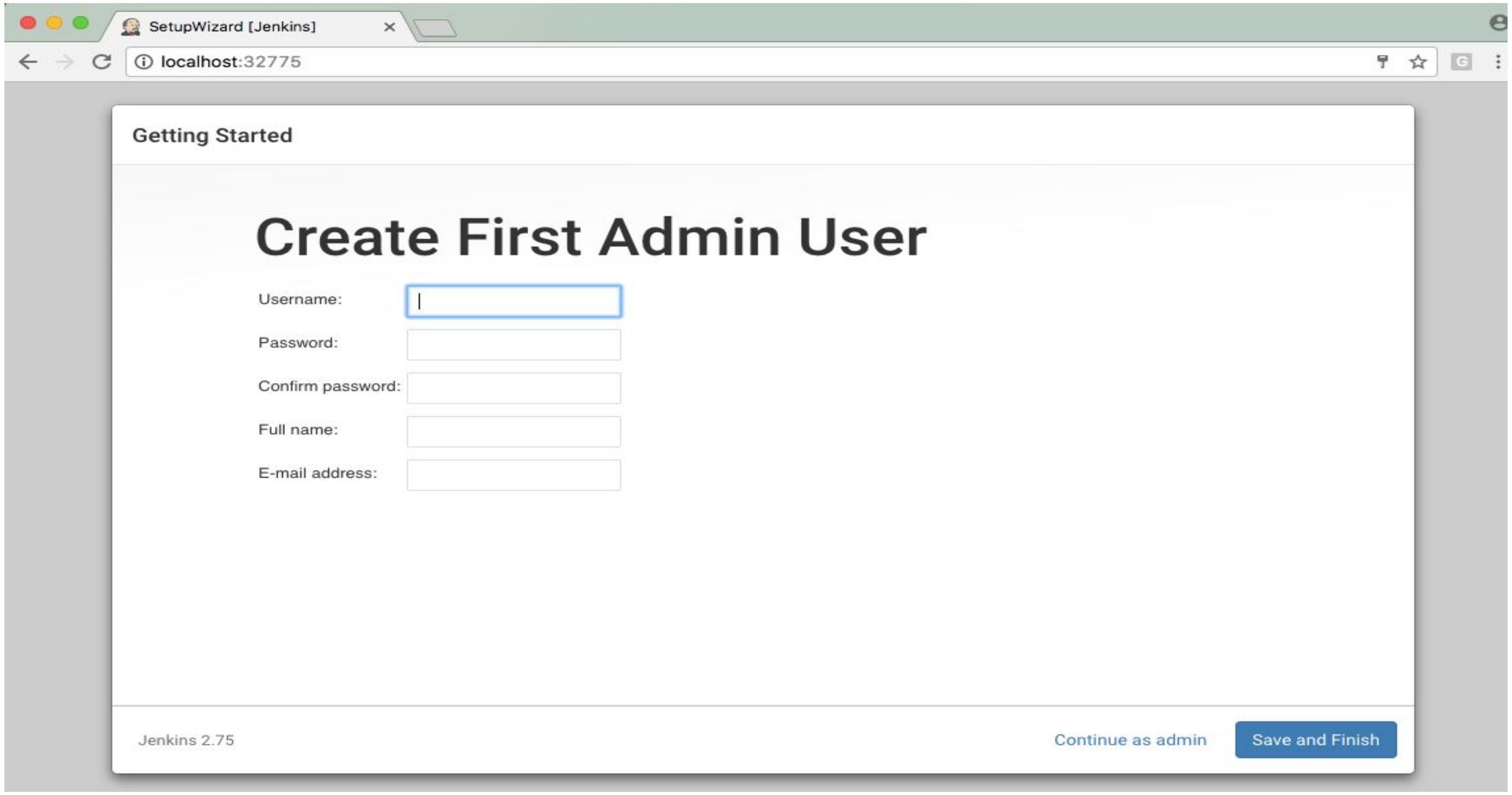
✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	build timeout plugin	Credentials Binding Plugin
Timestamper	Workspace Cleanup Plugin	Ant Plugin	Gradle Plugin
Pipeline	GitHub Branch Source Plugin	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View Plugin
Git plugin	Subversion Plug-in	SSH Slaves plugin	Matrix Authorization Strategy Plugin
✓ PAM Authentication plugin	LDAP Plugin	Email Extension Plugin	Mailer Plugin

```
** bouncycastle API Plugin
Folders Plugin
** Structs Plugin
** JUnit Plugin
OWASP Markup Formatter Plugin
PAM Authentication plugin
** Windows Slaves Plugin
** Display URL API
Jenkins Mailer Plugin
```

** - required dependency

Jenkins 2.75

Jenkins: Installation – Create First Admin User


SetupWizard [Jenkins]

localhost:32775

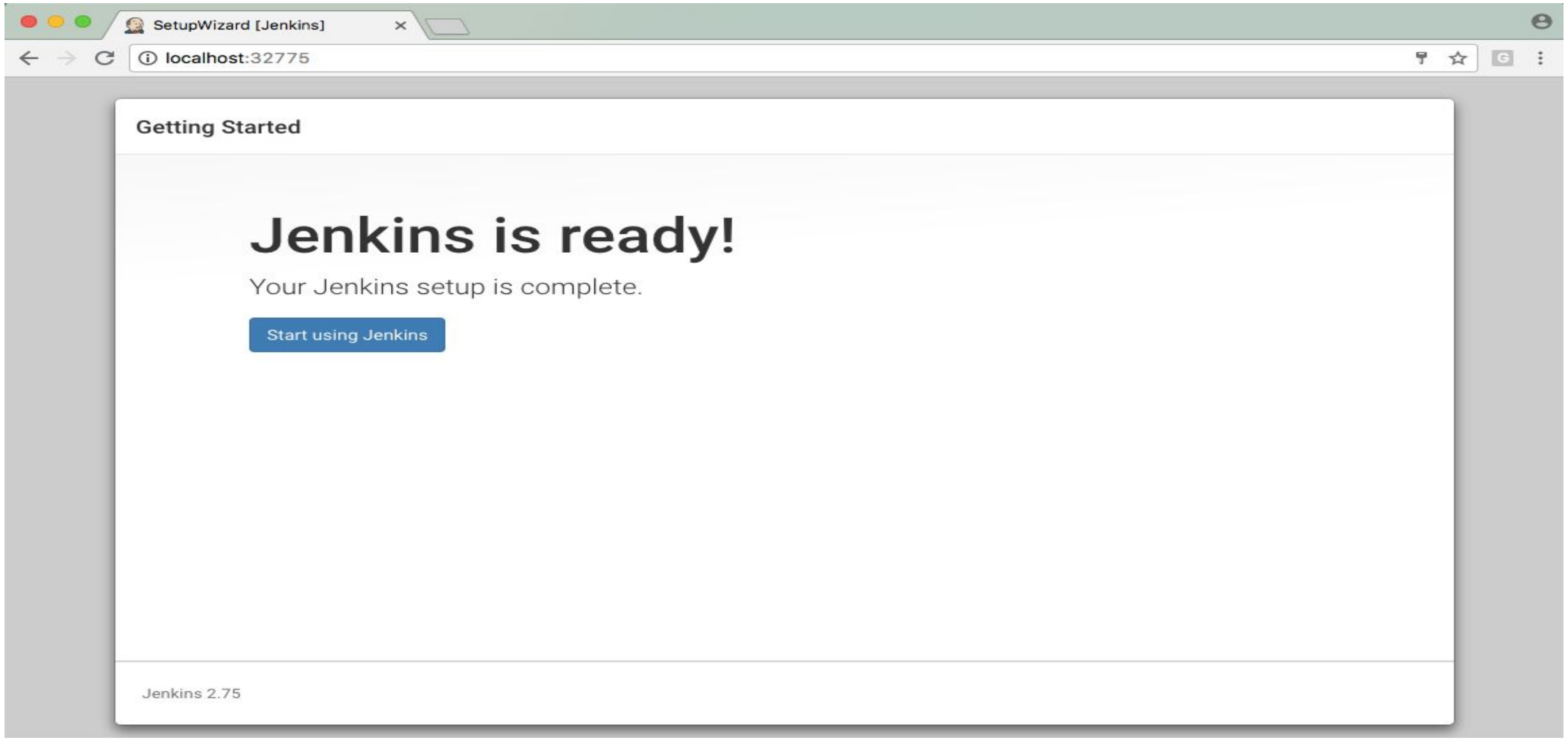
Getting Started

Create First Admin User

Username:	
Password:	
Confirm password:	
Full name:	
E-mail address:	

Jenkins 2.75

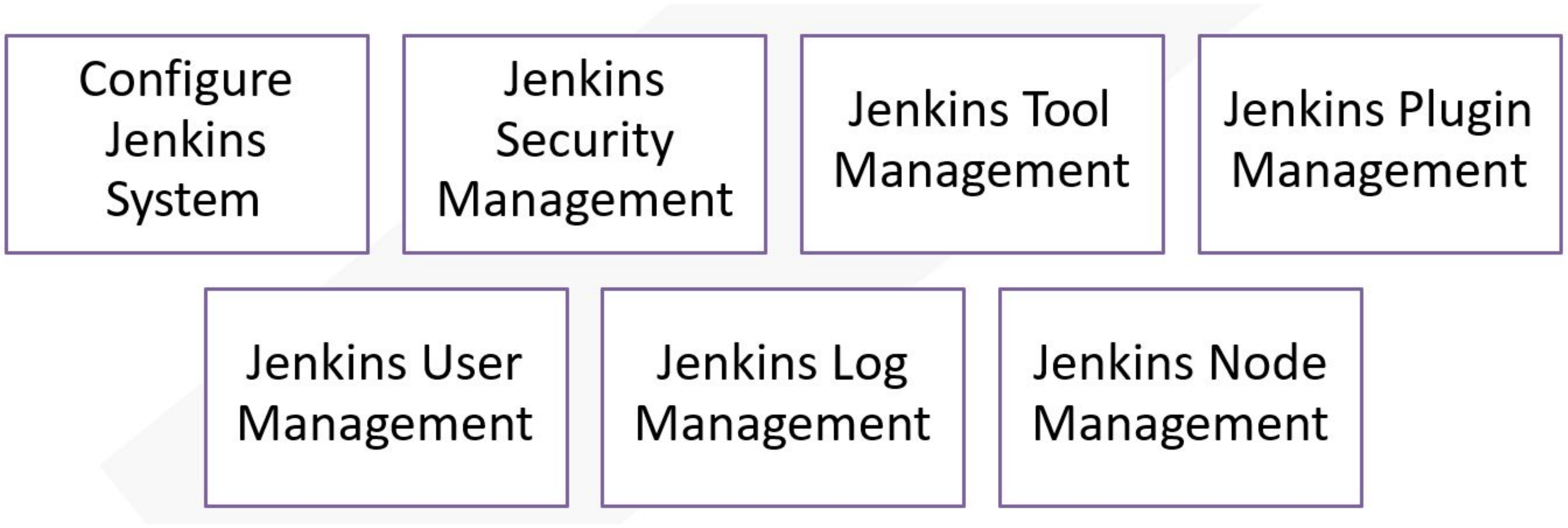
Jenkins: Installation – Ready to View Jenkins Dashboard



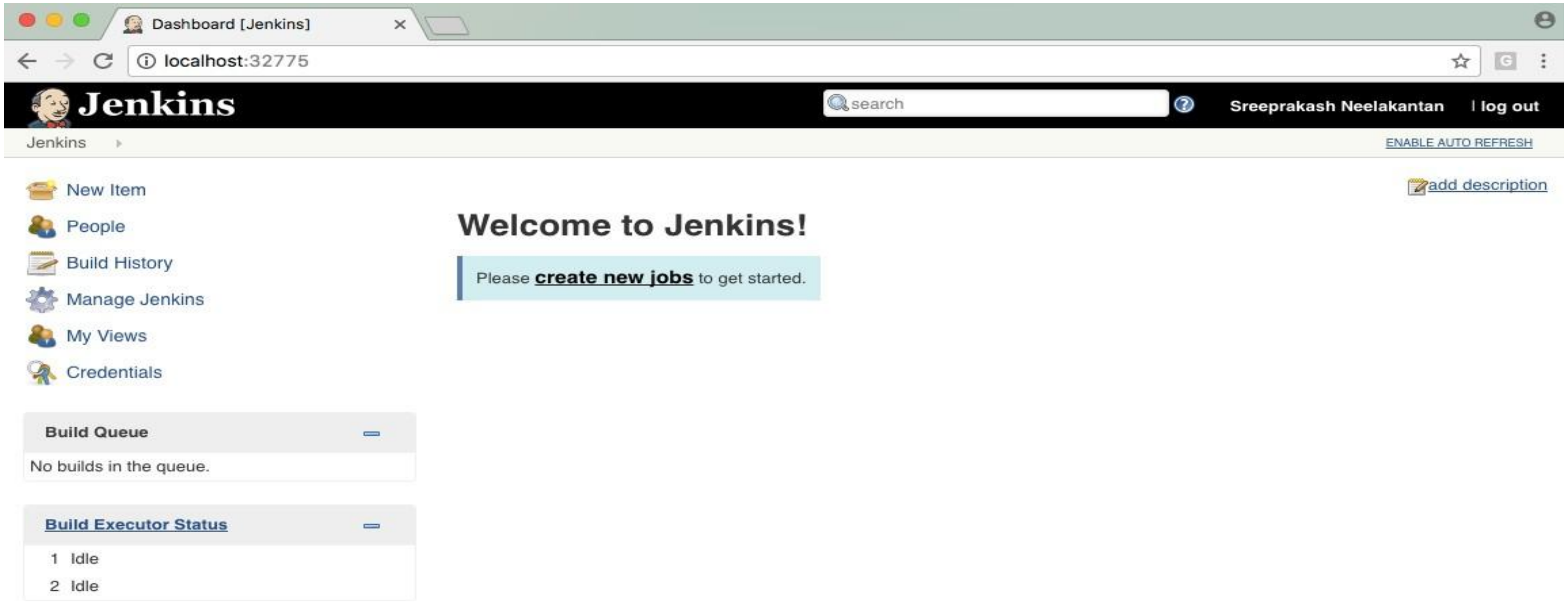
Jenkins is ready!

Your Jenkins setup is complete.

Jenkins: Management



Jenkins: Installation – Ready to View Jenkins Dashboard



Welcome to Jenkins!

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

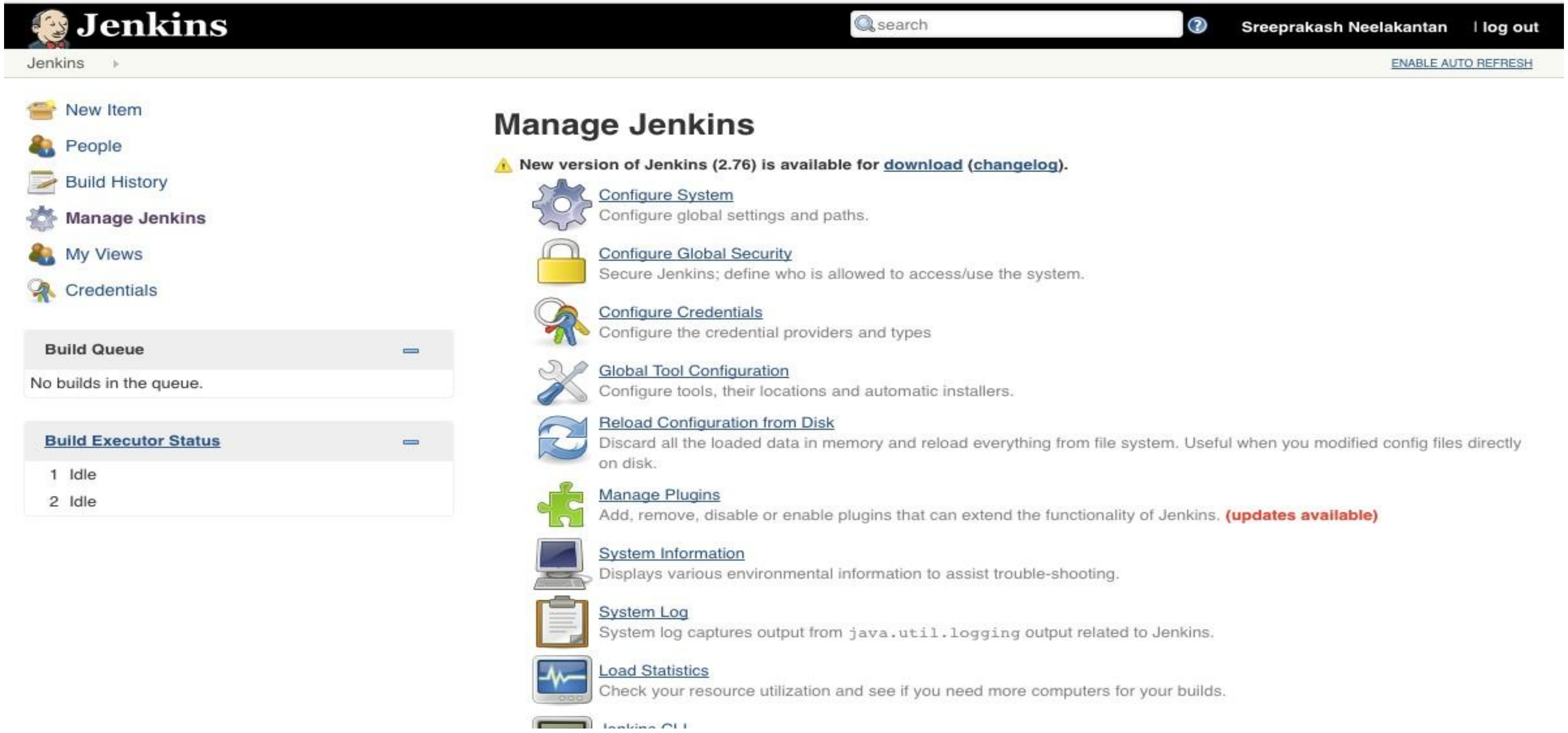
Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Jenkins: Securing Jenkins



Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Manage Jenkins

New version of Jenkins (2.76) is available for download (changelog).

Configure System
Configure global settings and paths.

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

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. **(updates available)**

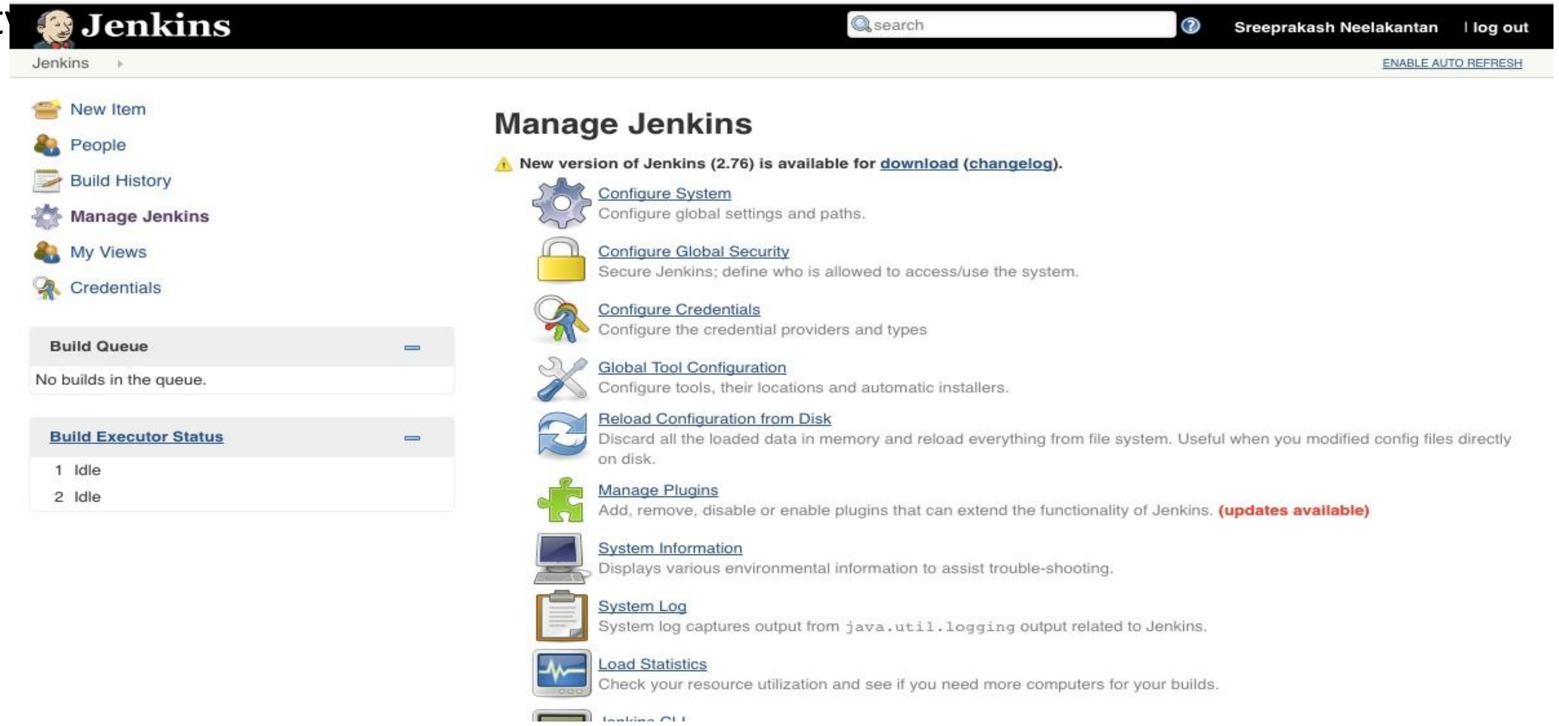
System Information
Displays various environmental information to assist trouble-shooting.

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.

Jenkins: Securing Jenkins - Authentication

1. Configure Global Securit



Jenkins | search | Sreeprakash Neelakantan | log out

Jenkins ▸ | ENABLE AUTO REFRESH

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Manage Jenkins

New version of Jenkins (2.76) is available for download (changelog).

Configure System
Configure global settings and paths.

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

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

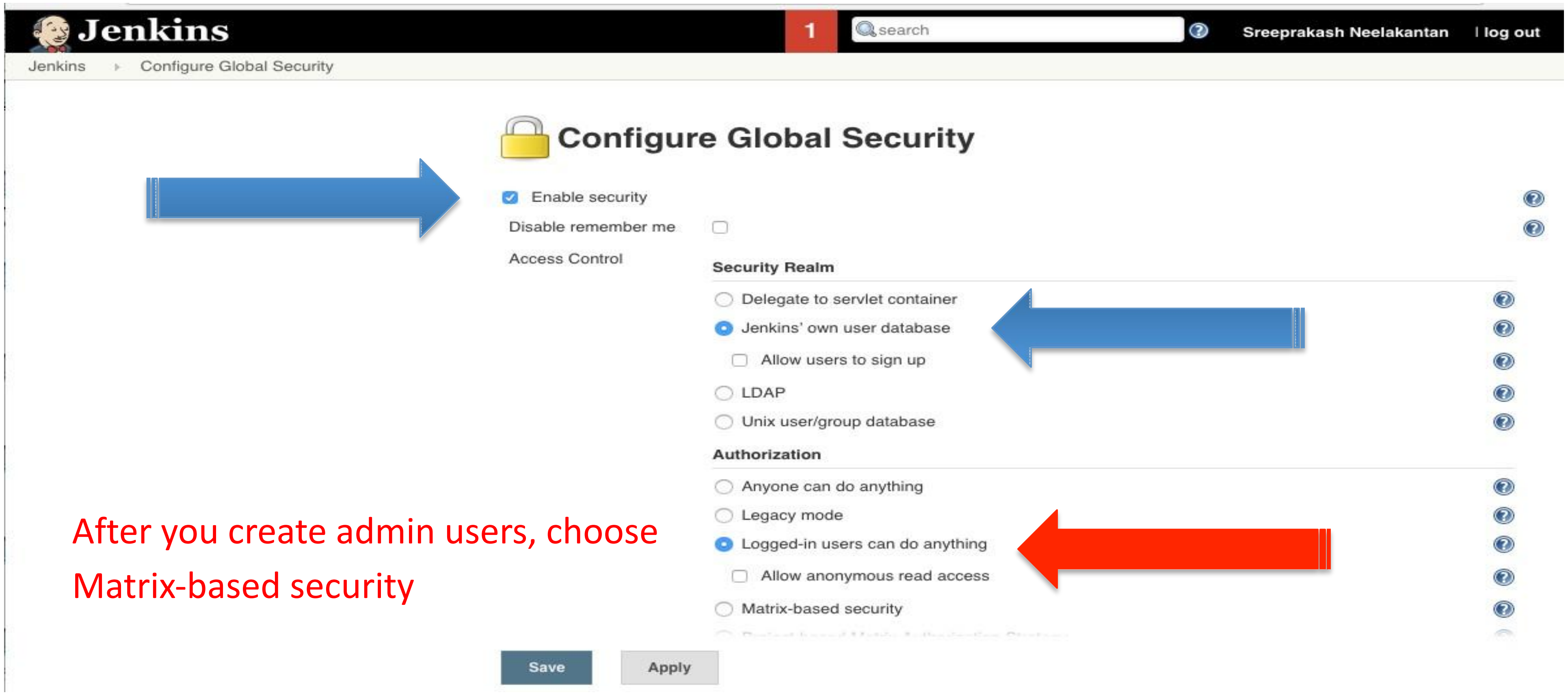
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. **(updates available)**

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.

Jenkins: Securing Jenkins - Authentication



Configure Global Security

- [✓] Enable security
- Disable remember me [ ]
- Access Control

Security Realm

- ( ) Delegate to servlet container
- (●) Jenkins' own user database
  - [ ] Allow users to sign up
- ( ) LDAP
- ( ) Unix user/group database

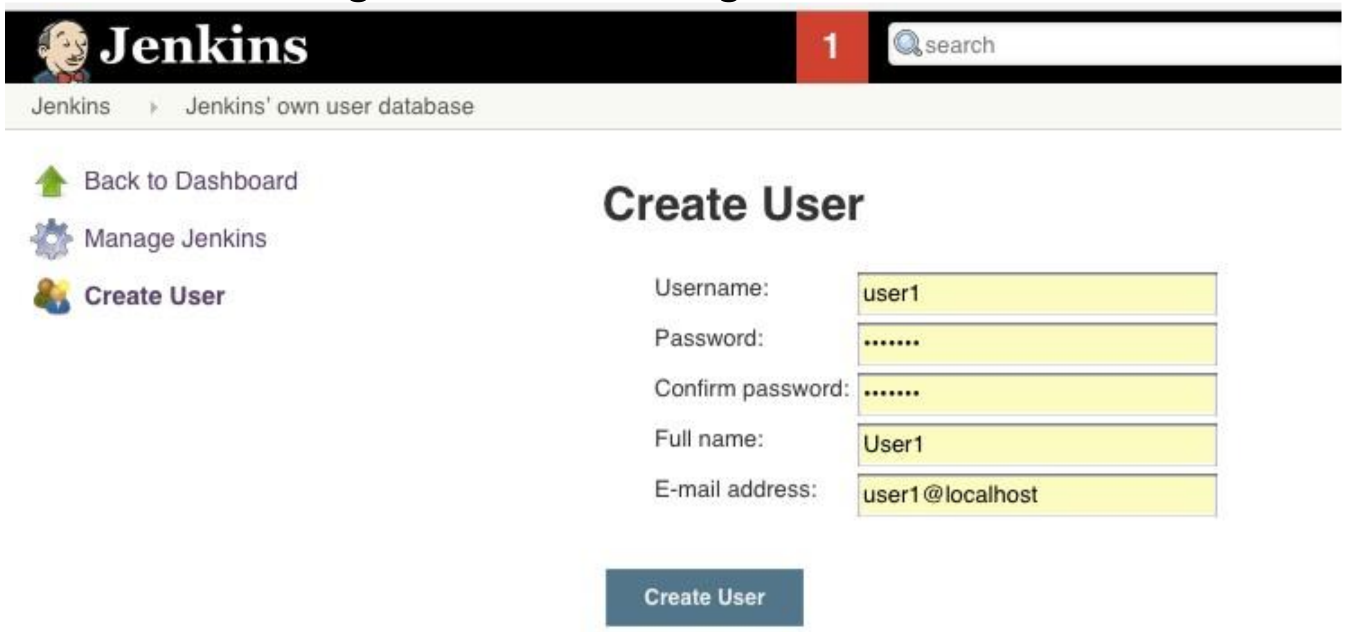
Authorization

- ( ) Anyone can do anything
- ( ) Legacy mode
- (●) Logged-in users can do anything
  - [ ] Allow anonymous read access
- ( ) Matrix-based security

After you create admin users, choose Matrix-based security

Jenkins: Securing Jenkins - Creating users

1. Login as Admin -> Manage Jenkins -> Manage Users -> Create User

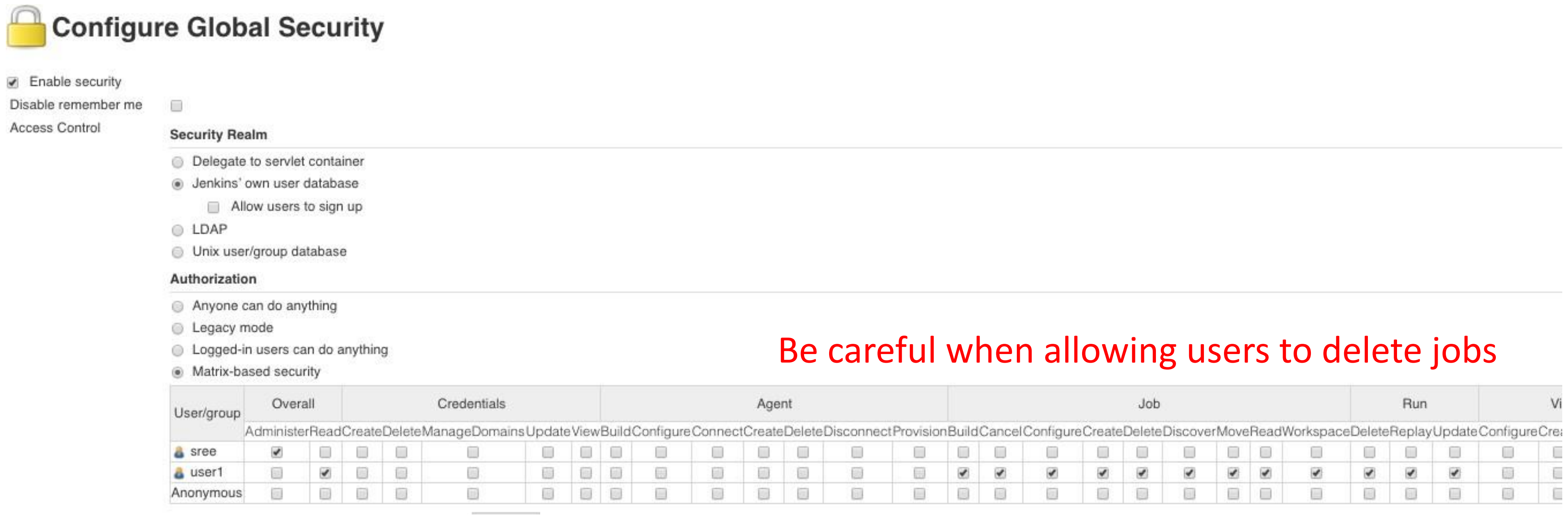


Create User

Username:	user1
Password:	•••••••
Confirm password:	•••••••
Full name:	User1
E-mail address:	user1@localhost

Jenkins: Securing Jenkins - Creating users

1. Login as Admin -> Manage Jenkins -> Configure Global Security -> Matrix-based
2. Allow only essential features for the users. Remember to check admin for at least one.



Configure Global Security

[✓ Enable security]

Disable remember me ☐

Access Control

Security Realm

- ○ Delegate to servlet container
- [✓ Jenkins' own user database]
  - ☐ Allow users to sign up
- ○ LDAP
- ○ Unix user/group database

Authorization

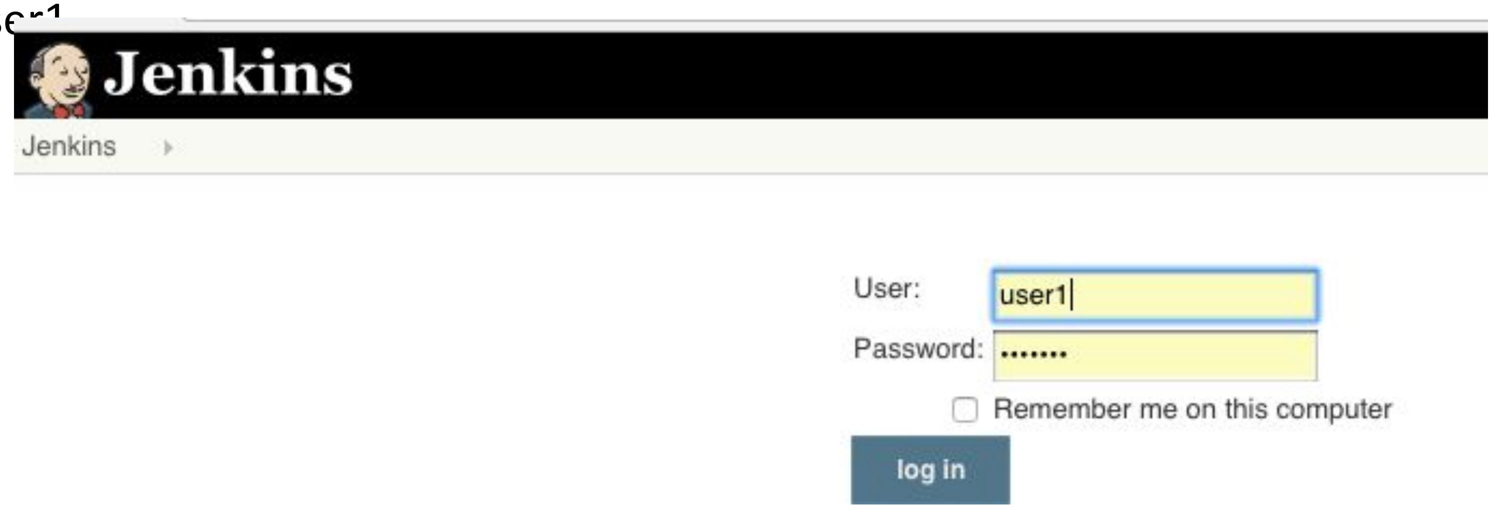
- ○ Anyone can do anything
- ○ Legacy mode
- ○ Logged-in users can do anything
- [✓ Matrix-based security]

Be careful when allowing users to delete jobs

User/group	Overall		Credentials					Agent							Job									Run			Vi	
	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	Crea
sree	✓	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
user1	☐	✓	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	☐	☐
Anonymous	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Jenkins: Securing Jenkins - Creating users

1. Logout and Login as User1



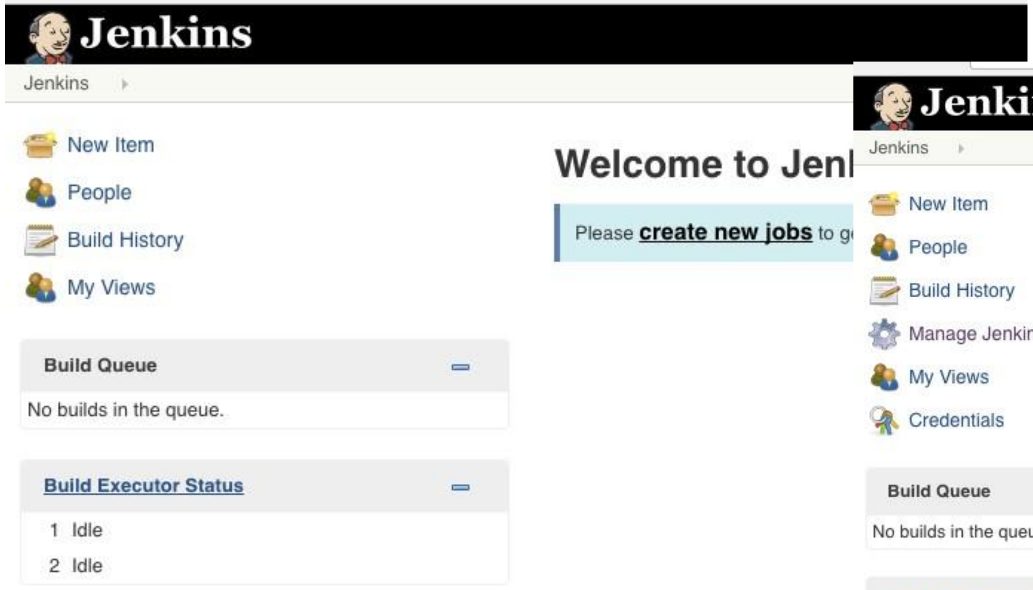
User: user1

Password: •••••••

Jenkins: Securing Jenkins - Creating users

1. User1 logged in (see the difference between the admin user and user1)

User1



Jenkins

Jenkins

- New Item
- People
- Build History
- My Views

Welcome to Jen

Please **create new jobs** to g

Build Queue

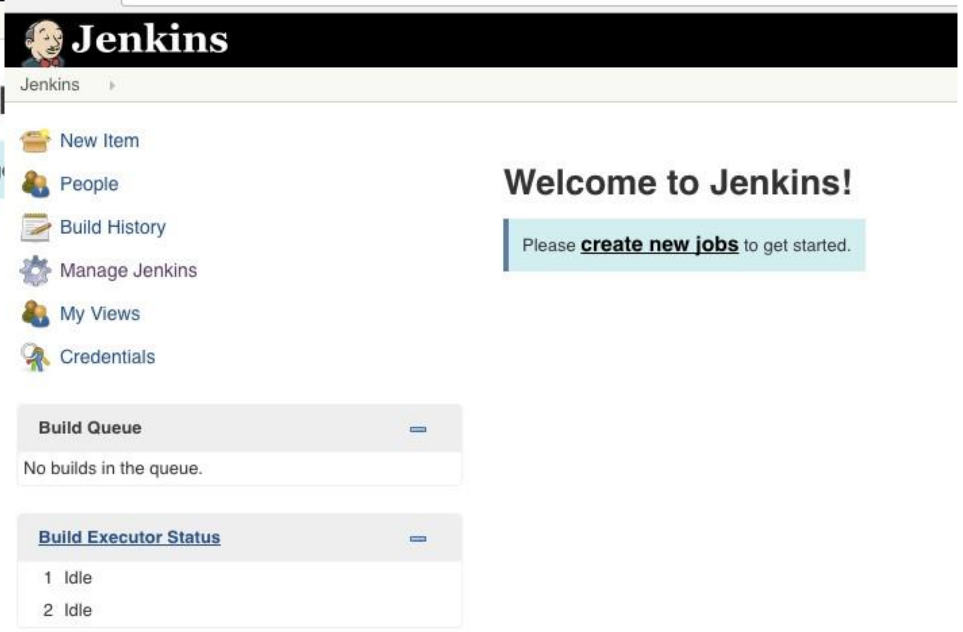
No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Admin



Jenkins

Jenkins

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

Welcome to Jenkins!

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

Build Queue

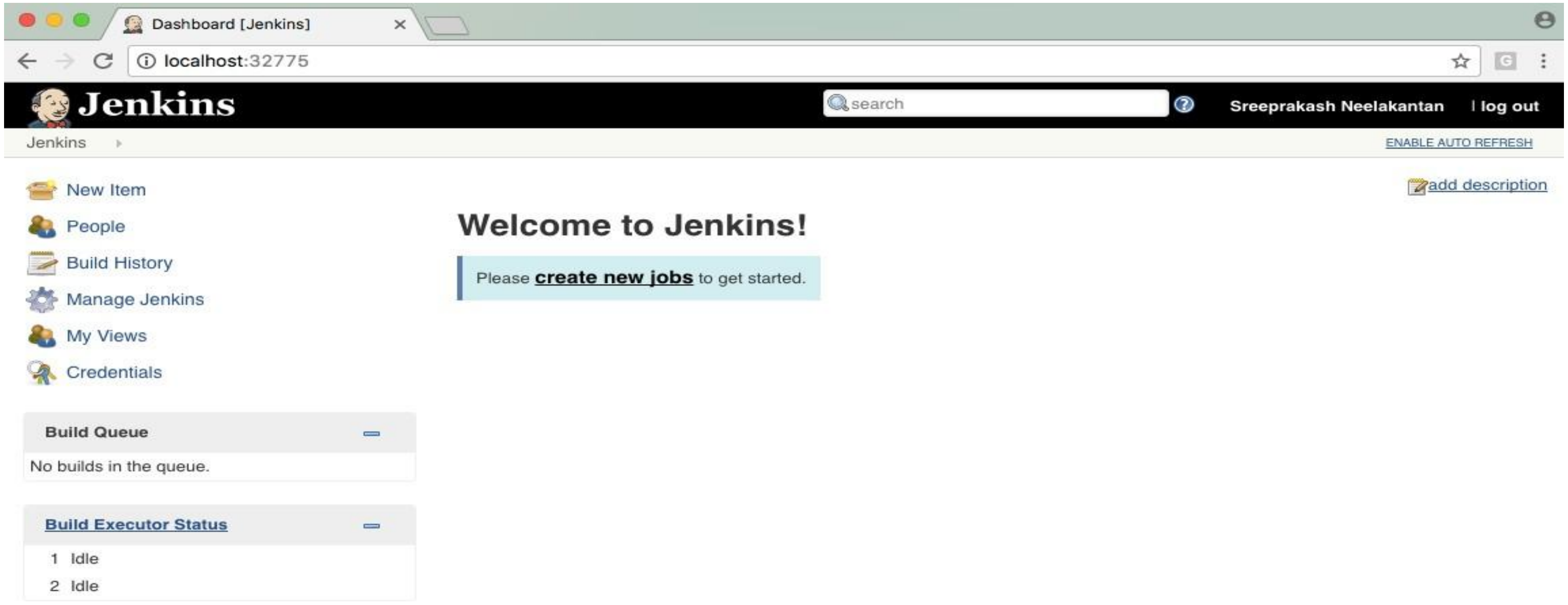
No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Jenkins: Installation - Exploring Jenkins Dashboard



Welcome to Jenkins!

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

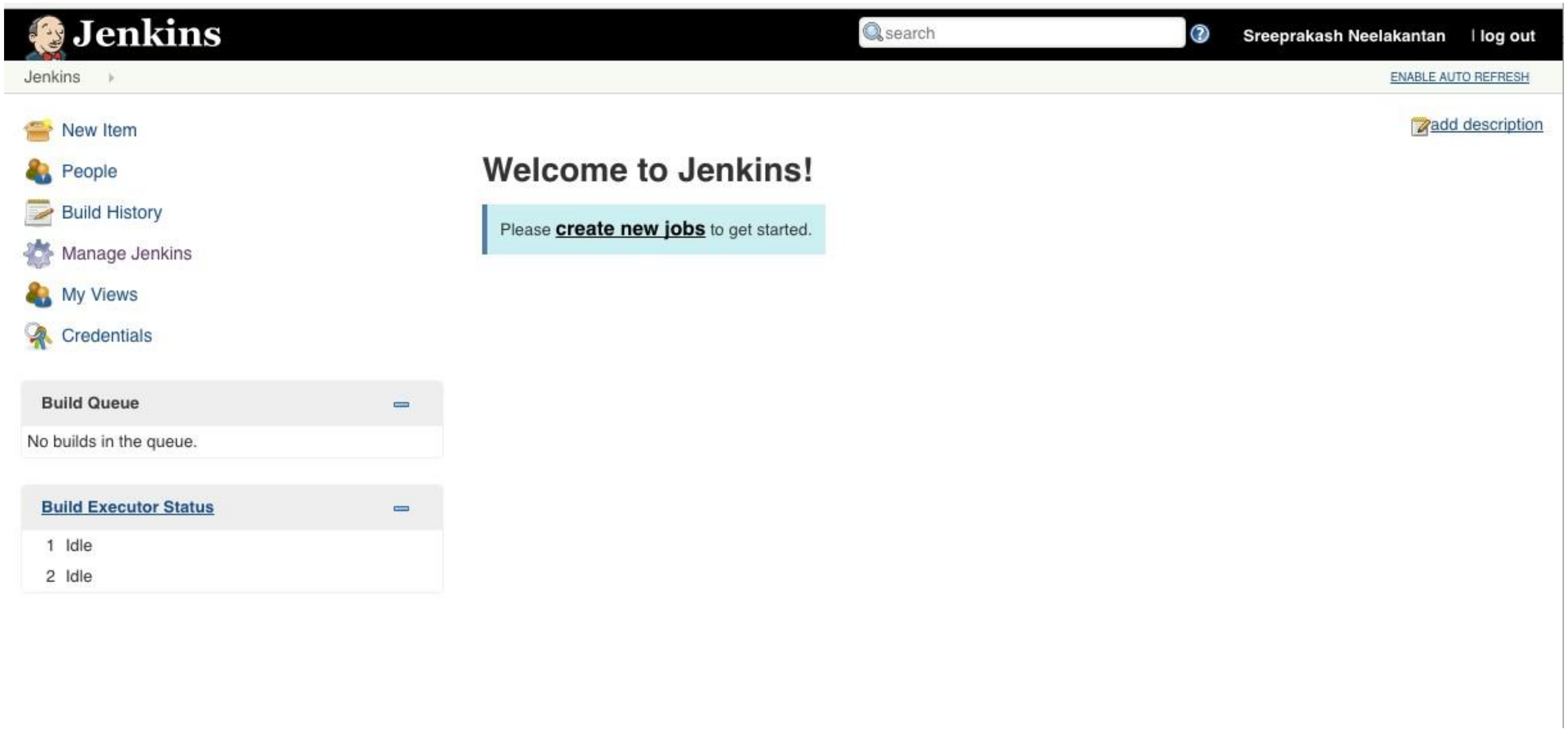
Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Jenkins: Jobs - Creating Jobs



Welcome to Jenkins!

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

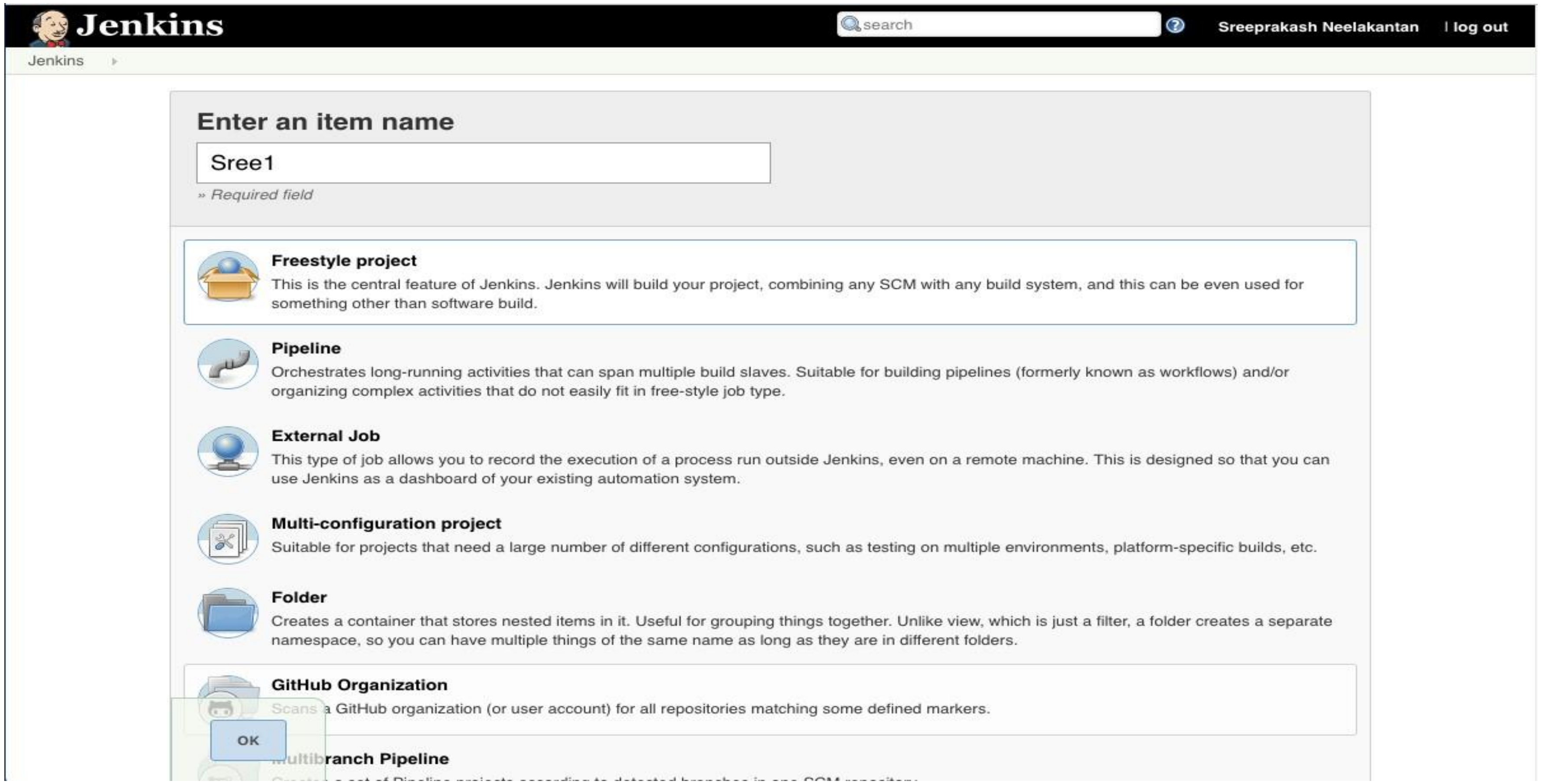
Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Jenkins: Jobs - Creating Jobs



Enter an item name

Sree1

» 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 can be even used for something other than software build.

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

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 designed so that you can use Jenkins as a dashboard of your existing automation system.

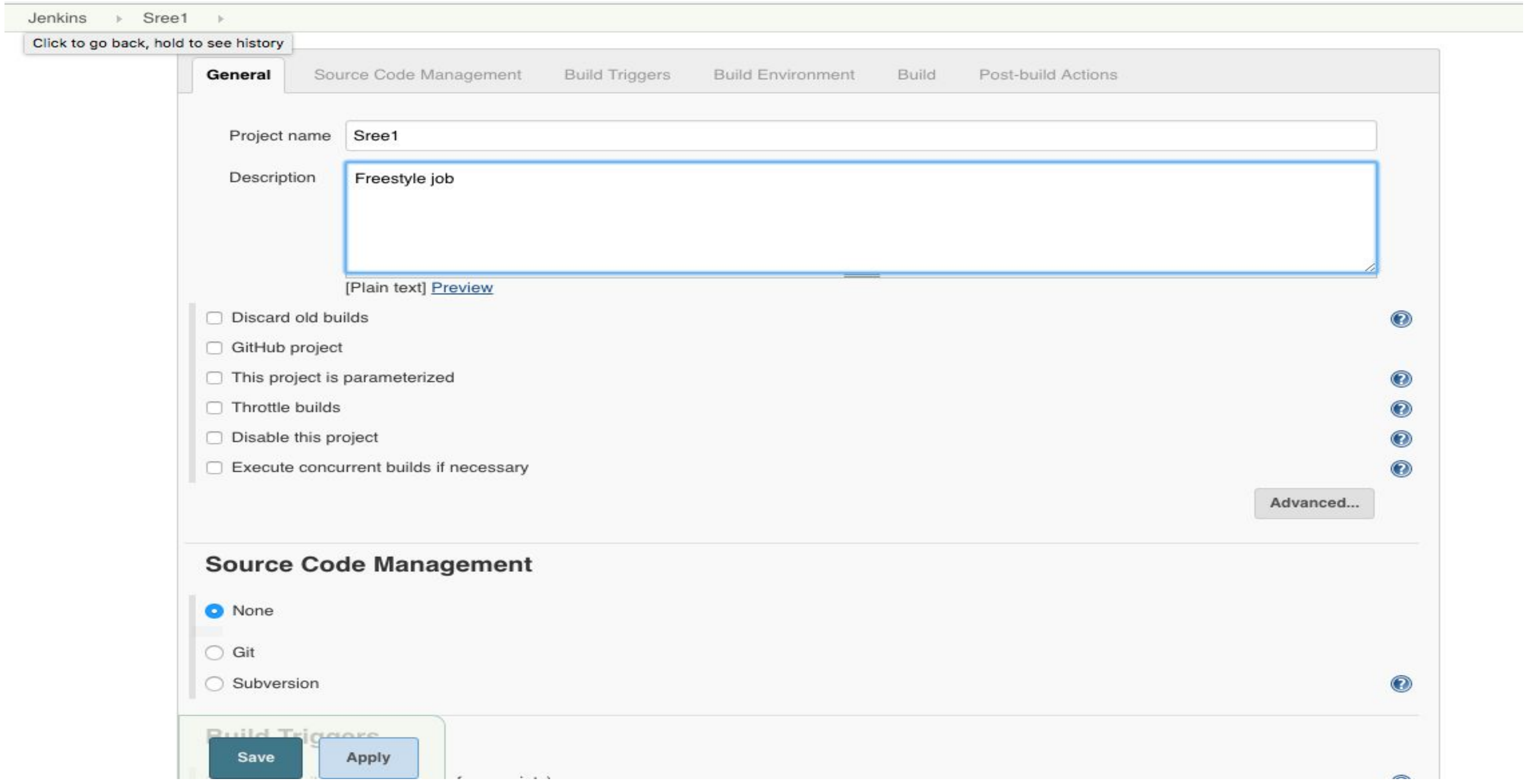
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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.

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

Multibranch Pipeline

Jenkins: Jobs - Creating Jobs



Project name: Sree1

Description: Freestyle job

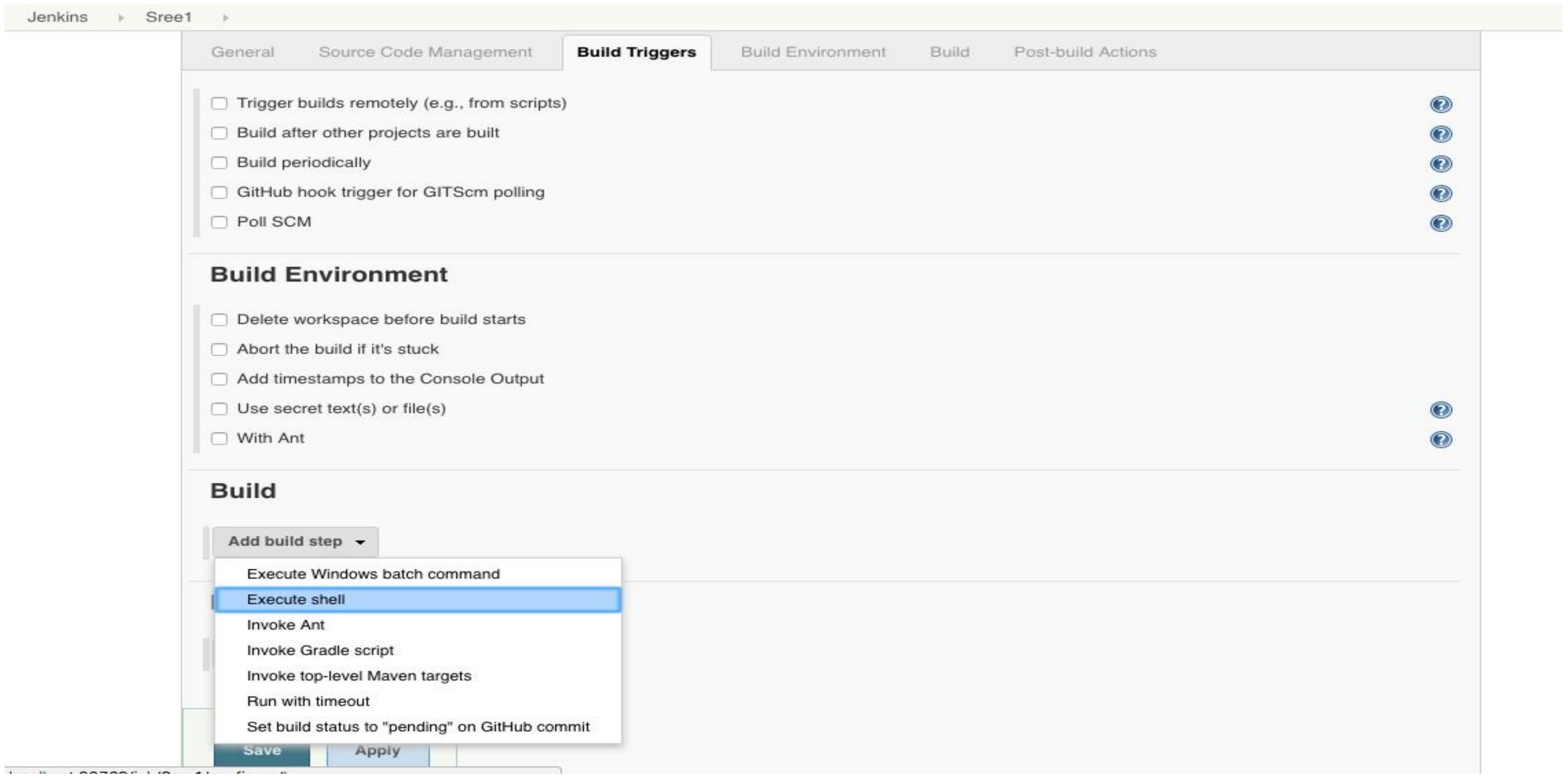
[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project
- [ ] Execute concurrent builds if necessary

Source Code Management

[✓ None] Git Subversion

Jenkins: Jobs - Creating Jobs



Jenkins ▸ Sree1 ▸

General | Source Code Management | **Build Triggers** | Build Environment | Build | Post-build Actions

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM

Build Environment

- ☐ Delete workspace before build starts
- ☐ Abort the build if it's stuck
- ☐ Add timestamps to the Console Output
- ☐ Use secret text(s) or file(s)
- ☐ With Ant

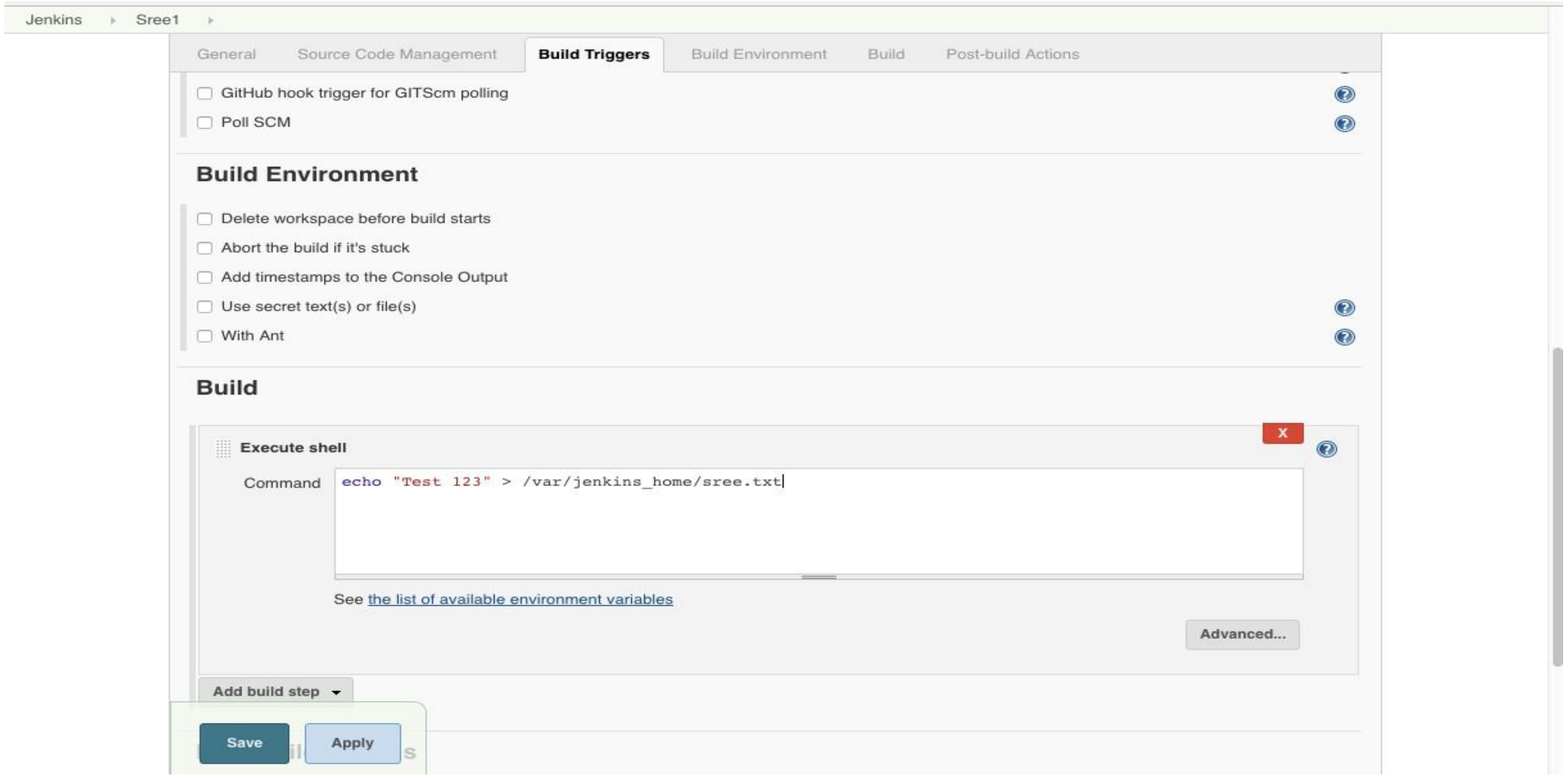
Build

Add build step ▾

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Save | Apply

Jenkins: Jobs - Creating Jobs



- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Use secret text(s) or file(s)
- [ ] With Ant

Build

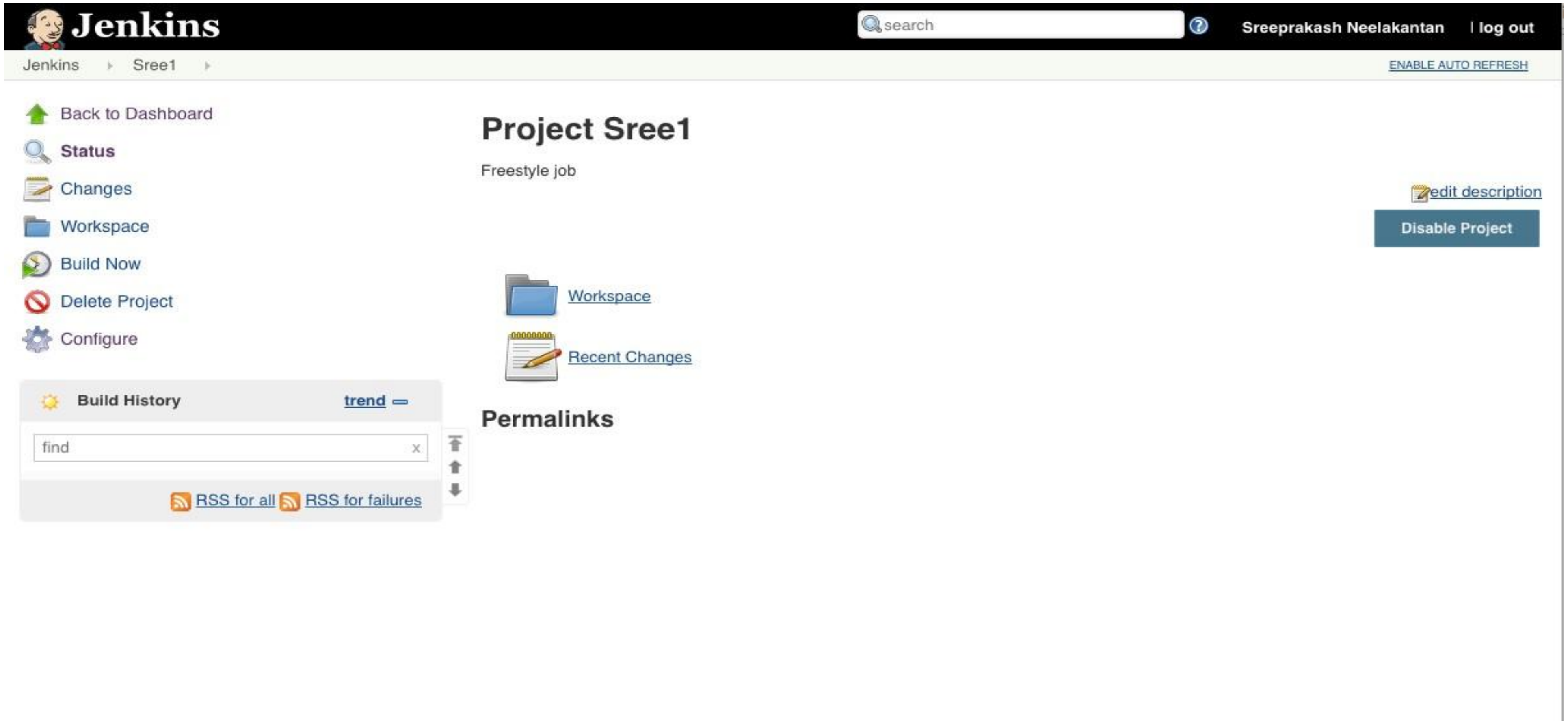
Execute shell

Command

```
echo "Test 123" > /var/jenkins_home/sree.txt
```

See the list of available environment variables

Jenkins: Jobs - Running the Jobs – Jobs Details

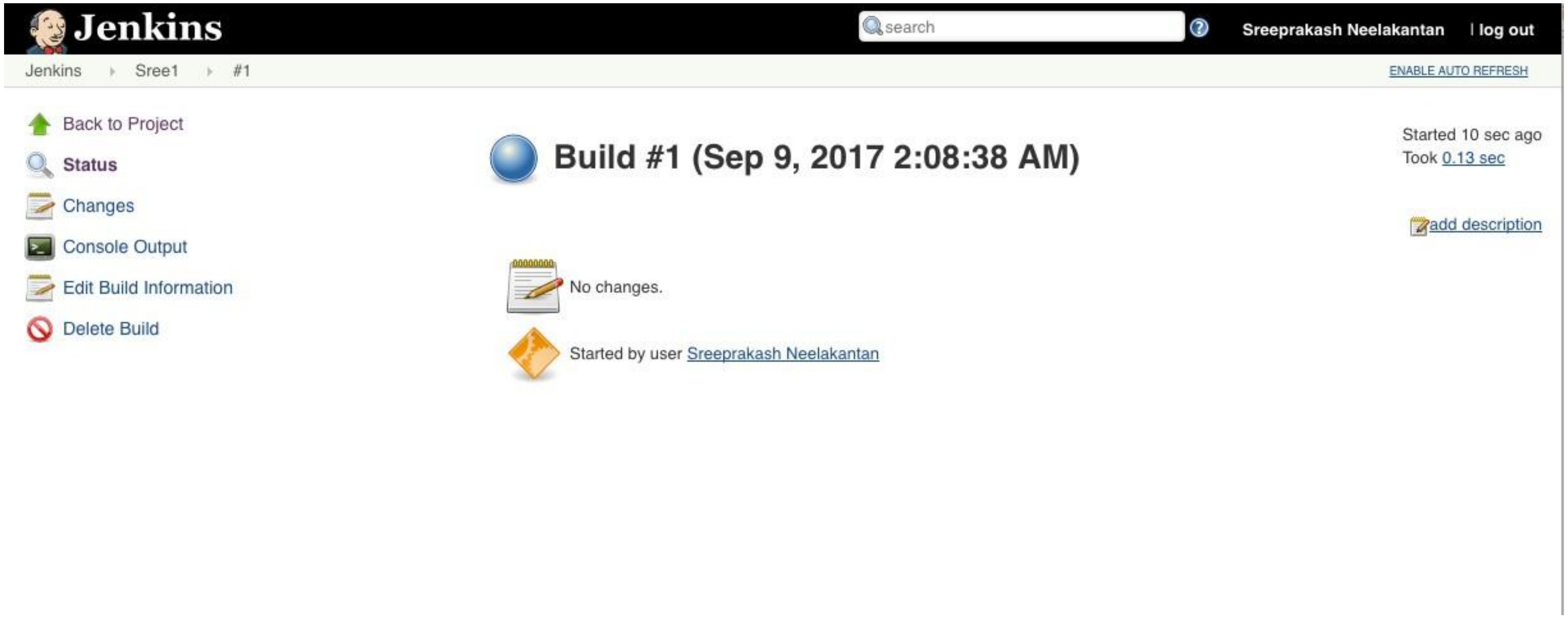


Project Sree1

Freestyle job

Permalinks

Jenkins: Jobs - Running the Jobs – Build Details



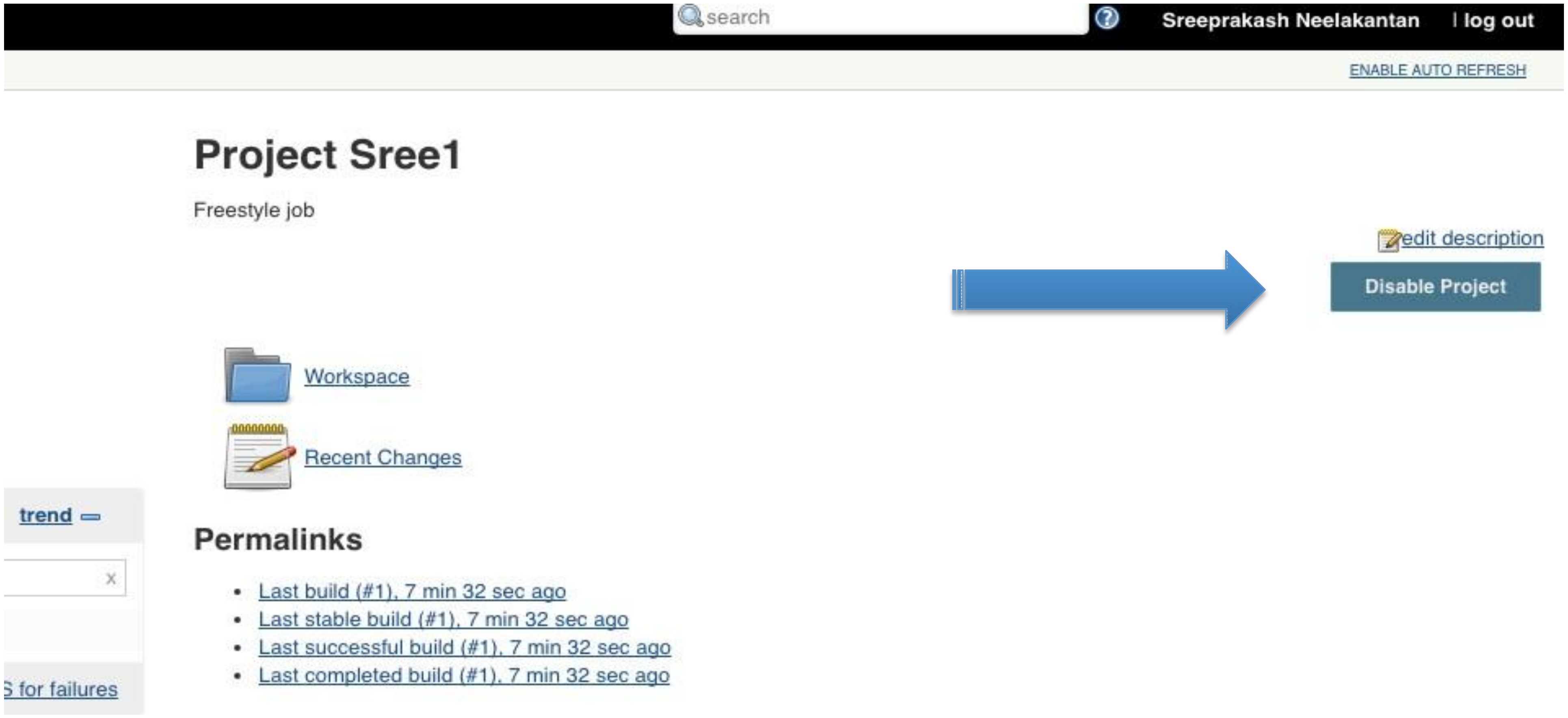
Build #1 (Sep 9, 2017 2:08:38 AM)

Started 10 sec ago
Took 0.13 sec

No changes.

Started by user Sreeprakash Neelakantan

Jenkins: Jobs - Disabling and Enabling jobs



Project Sree1

Freestyle job

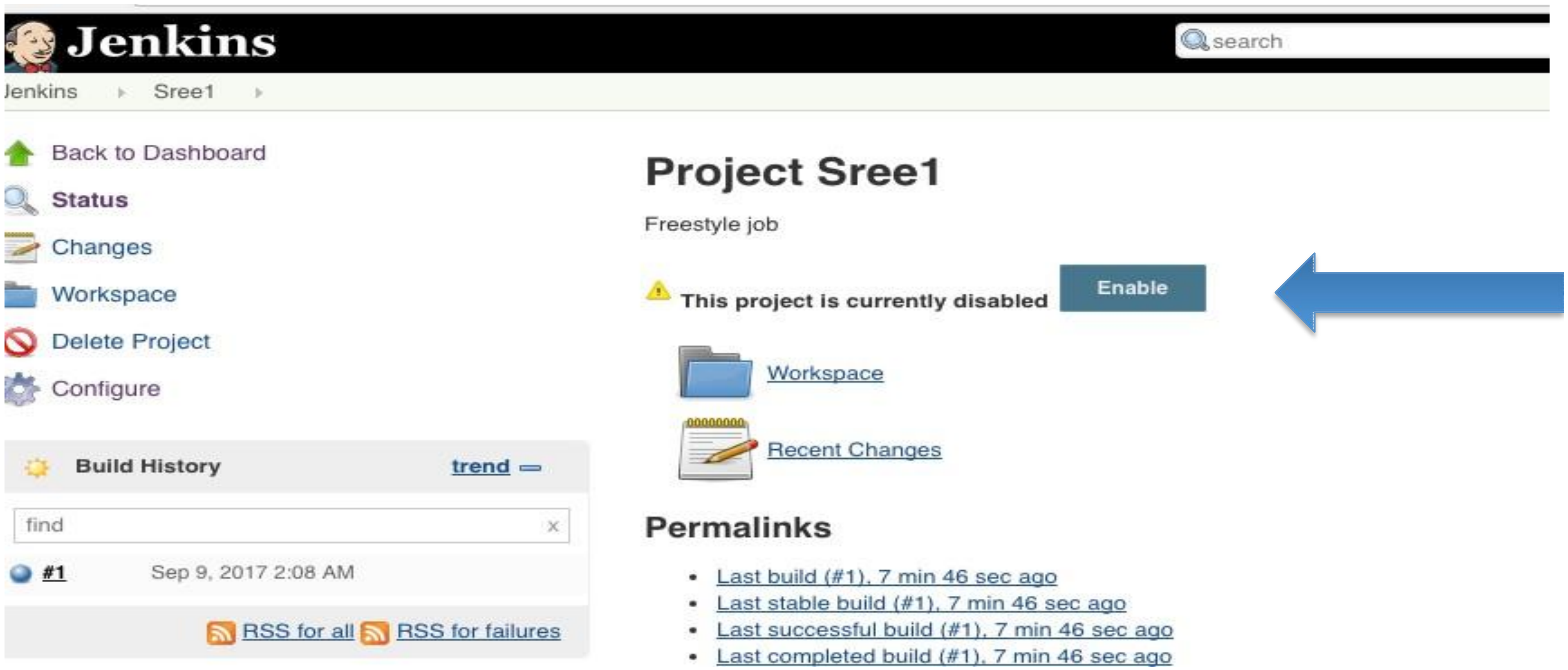
Workspace

Recent Changes

Permalinks

- Last build (#1), 7 min 32 sec ago
- Last stable build (#1), 7 min 32 sec ago
- Last successful build (#1), 7 min 32 sec ago
- Last completed build (#1), 7 min 32 sec ago

Jenkins: Jobs - Disabling and Enabling jobs



Jenkins › Sree1 ›

search

Back to Dashboard
Status
Changes
Workspace
Delete Project
Configure

Build History trend

find

#1 Sep 9, 2017 2:08 AM

RSS for all RSS for failures

Project Sree1

Freestyle job

This project is currently disabled Enable

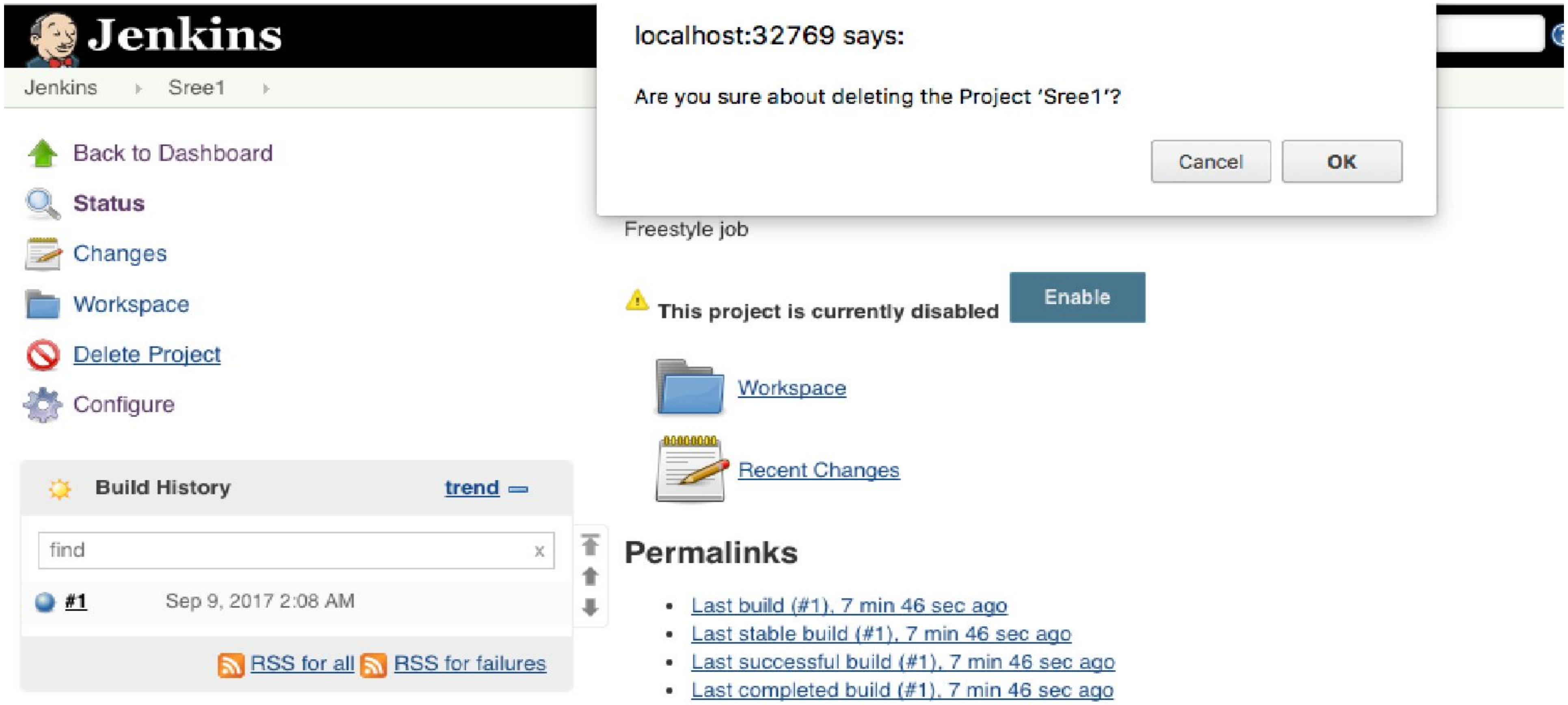
Workspace

Recent Changes

Permalinks

- Last build (#1), 7 min 46 sec ago
- Last stable build (#1), 7 min 46 sec ago
- Last successful build (#1), 7 min 46 sec ago
- Last completed build (#1), 7 min 46 sec ago

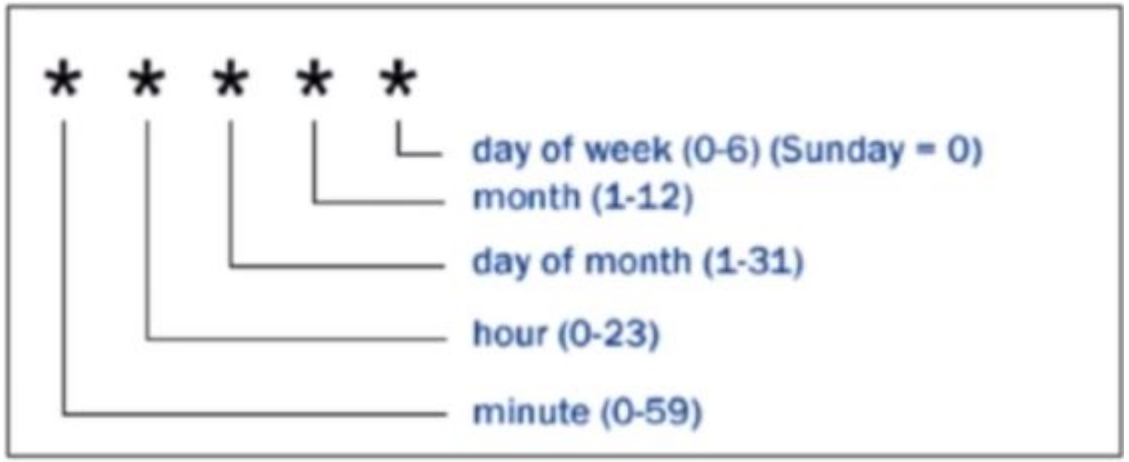
Jenkins: Jobs - Deleting jobs


localhost:32769 says:

Are you sure about deleting the Project 'Sree1'?

Jenkins: Schedule Build

In cron, Each line consists of 5 fields separated by tab or Whitespace



To specify the Multiple Values for one field

- -> all values
- M-N A range of values
- A,B,Z Enumerates multiple values

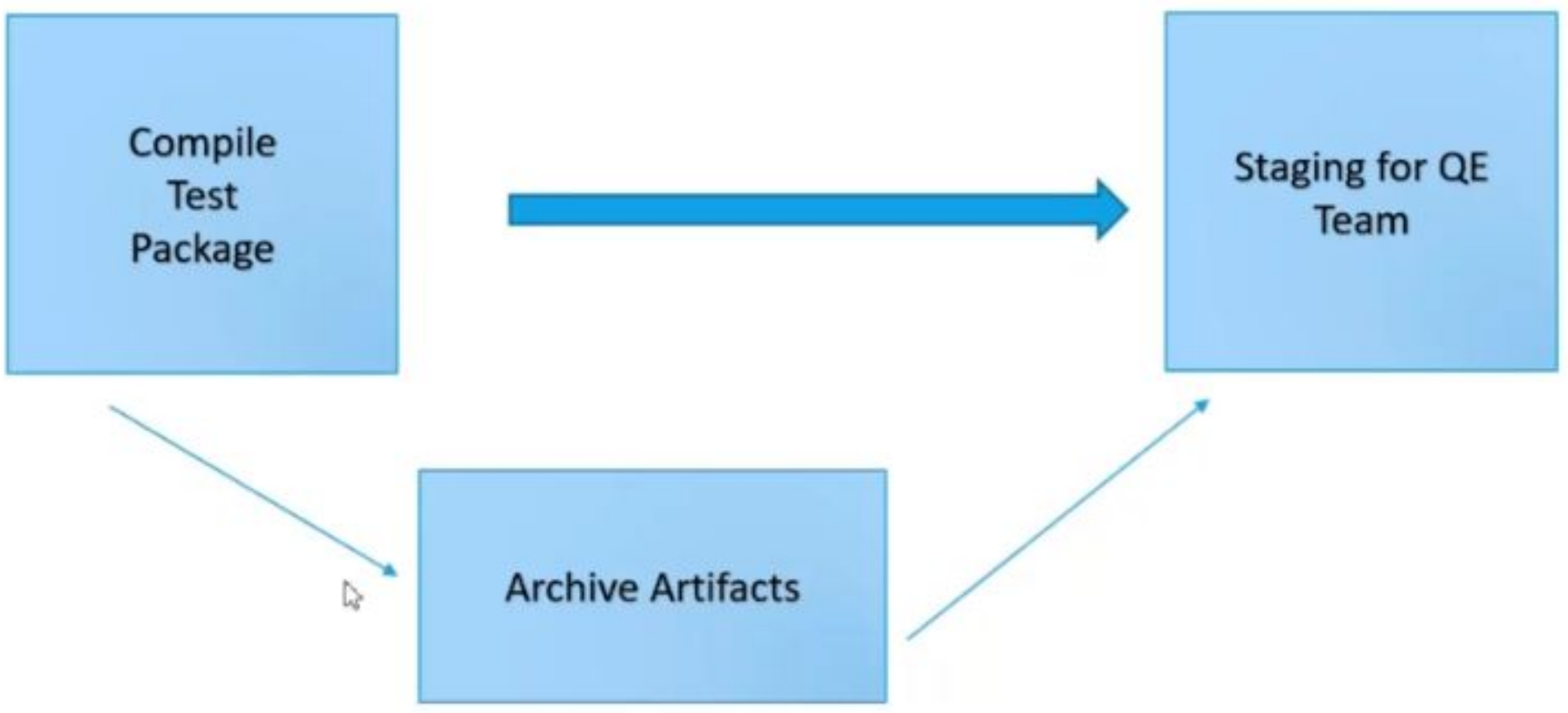
0 0 * * *   : Every data at midnight
0 2-4 * * * : 2 a.m , 3 a.m ,4 a.m Every day

GIT HUB Hook

https://wiki.jenkins-ci.org/display/JENKINS/GitHub+Plugin

Jenkins: Artifacts

.



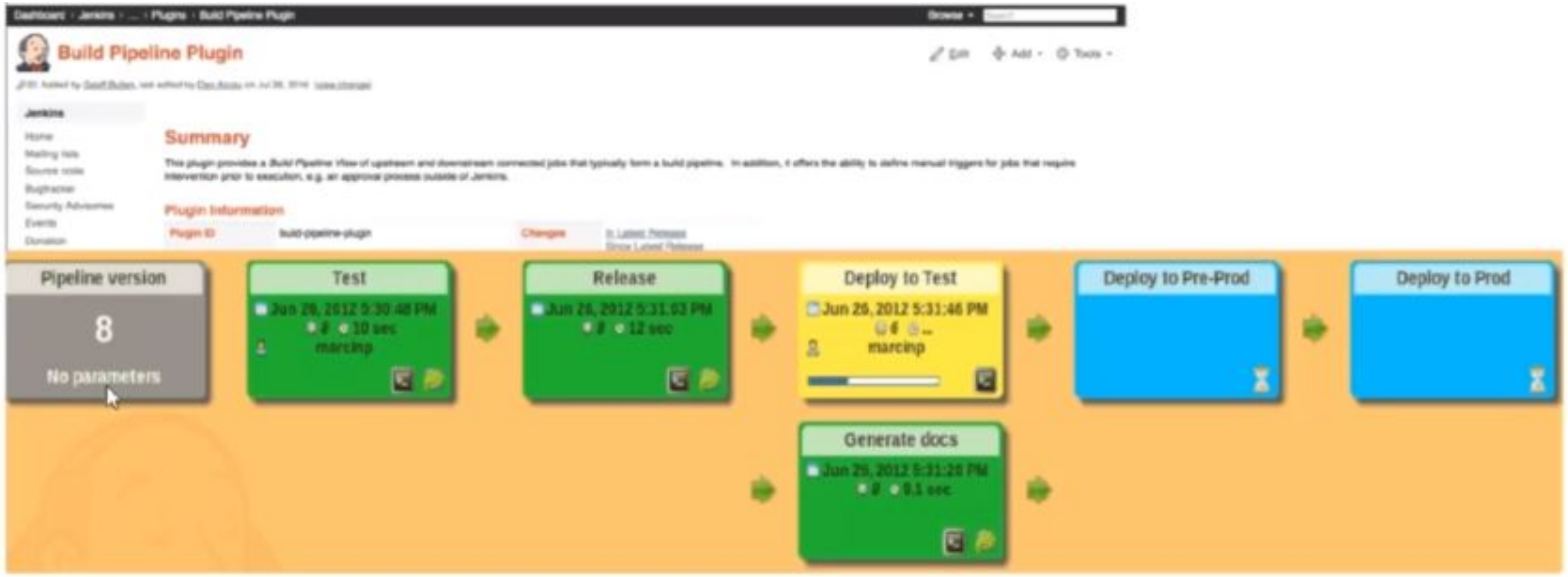
Jenkins : Tomcat

- Tomcat is an Open source HTTP webserver that will deploy and run applications inside the tomcat container
- Default port no of tomcat was :8080
- If we want to change the port for tomcat ,you can go through conf/server.xml and search for connector port and change your port no
- IF you want to run the server , you need to go bin directory and running startup from command prompt
- we need to keep WAR file inside the WebApps folder
- Once we kept WAR file inside then need to restart /start the server

Jenkins : Deploy Artifacts

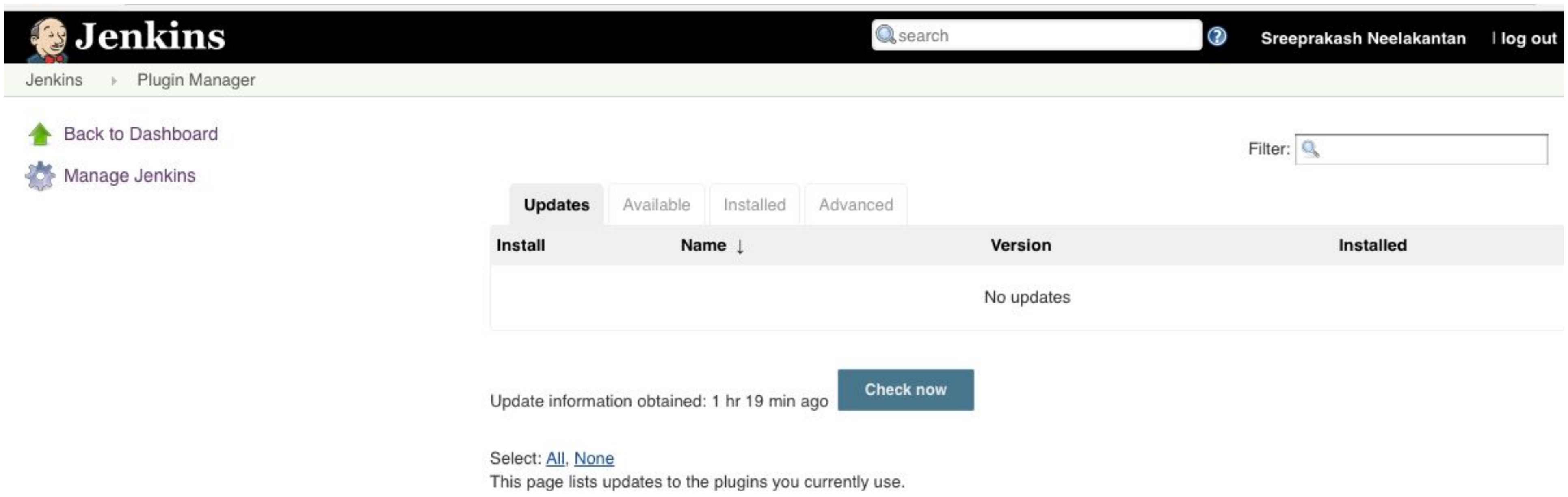
- Install copy artifact and deploy to container plugin
- Deploy our application to staging environments

Jenkins: Build Pipeline



- Configure and install build pipe line
- Create own custom view dashboard

Jenkins: Jobs - Adding and updating Plugins

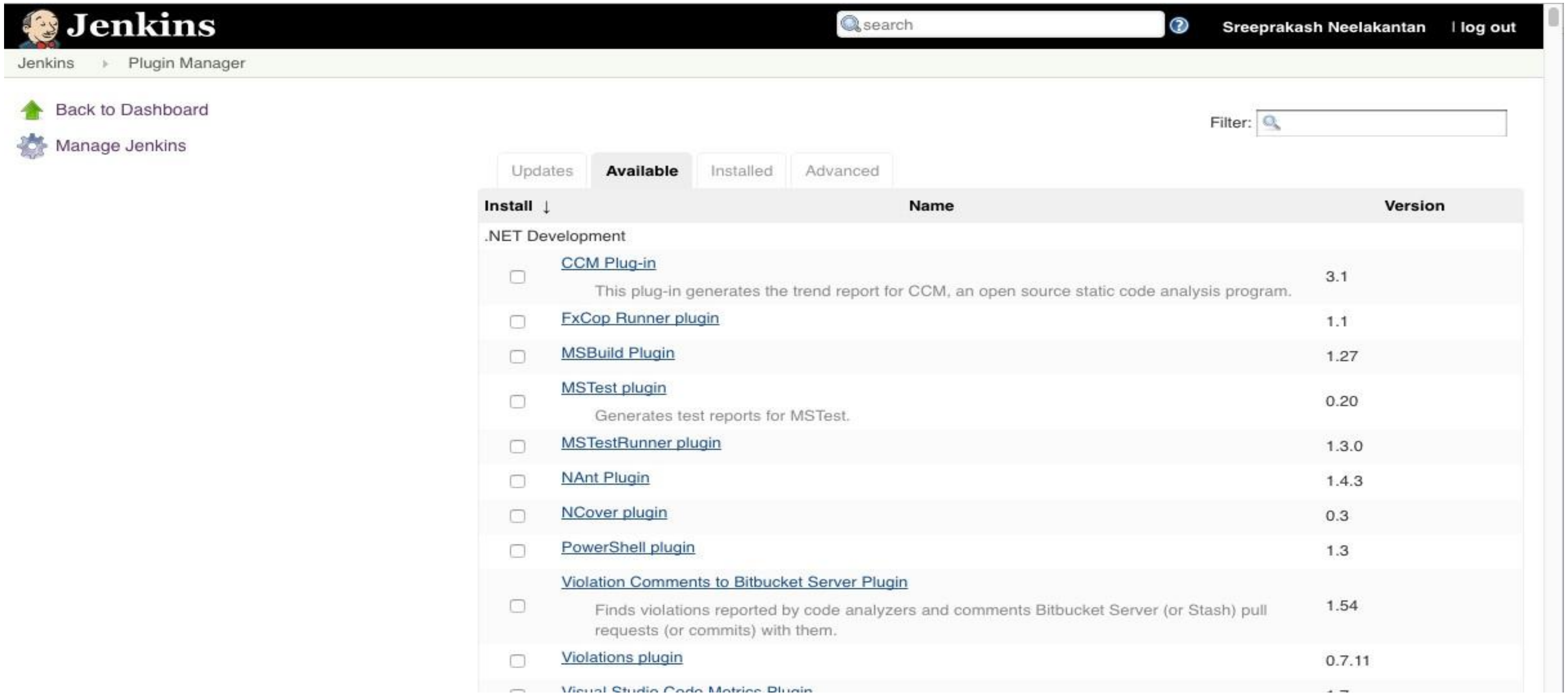


Install	Name ↓	Version	Installed
	No updates		

Update information obtained: 1 hr 19 min ago

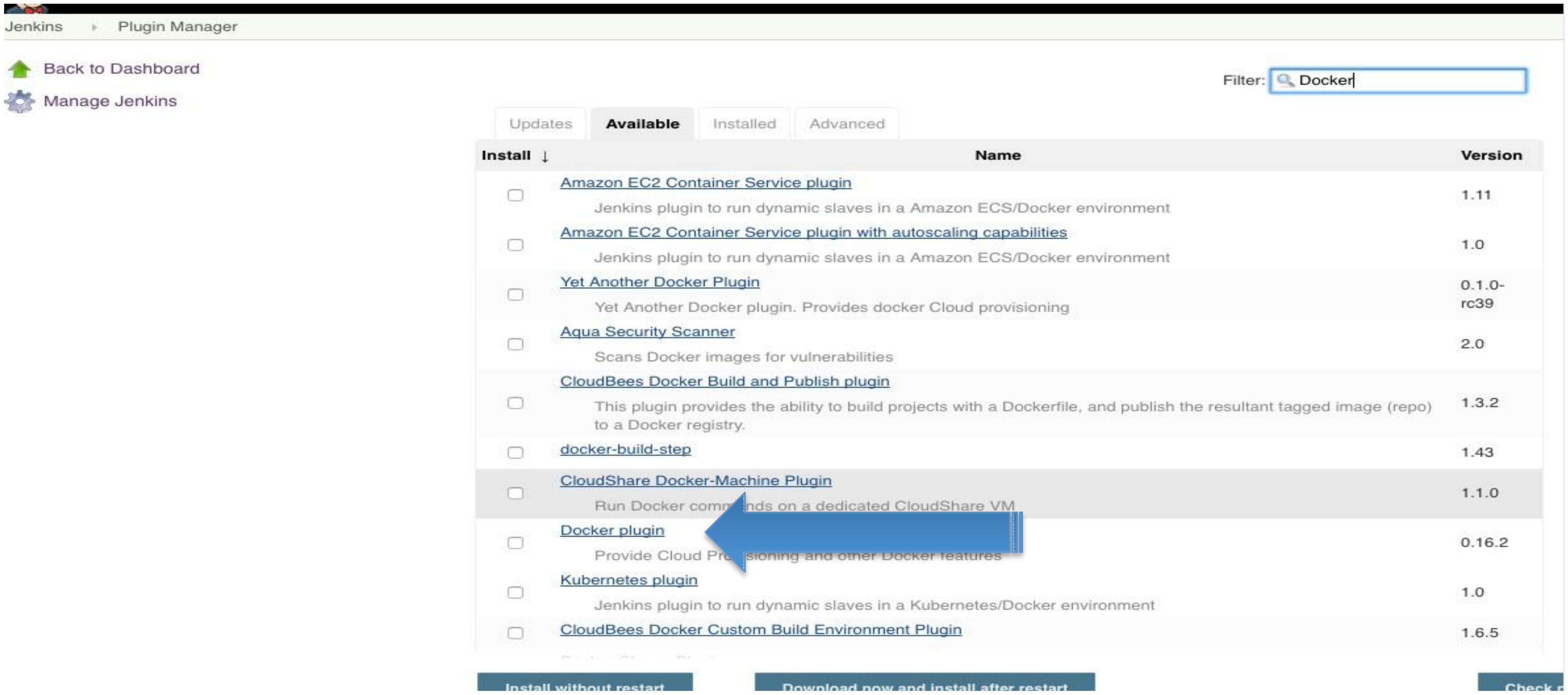
Select: All, None
This page lists updates to the plugins you currently use.

Jenkins: Jobs - Adding and updating Plugins



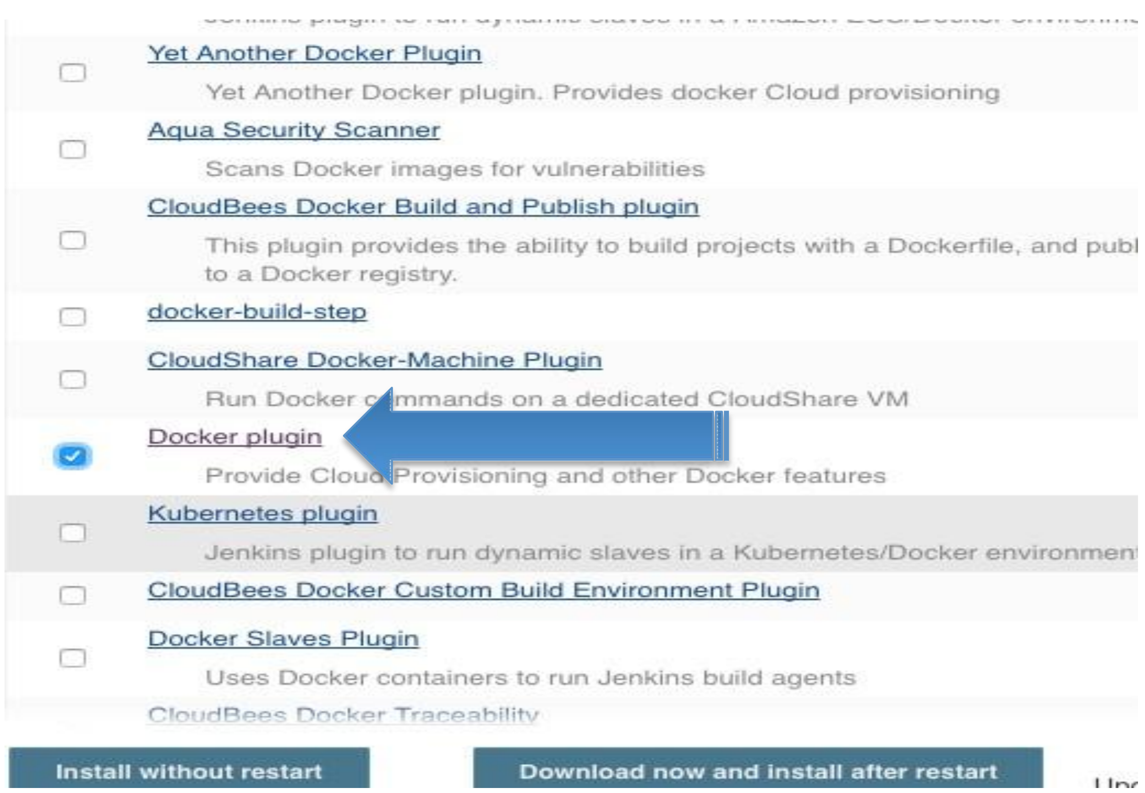
Install ↓	Name	Version
.NET Development		
☐	CCM Plug-in This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
☐	FxCop Runner plugin	1.1
☐	MSBuild Plugin	1.27
☐	MSTest plugin Generates test reports for MSTest.	0.20
☐	MSTestRunner plugin	1.3.0
☐	NAnt Plugin	1.4.3
☐	NCover plugin	0.3
☐	PowerShell plugin	1.3
☐	Violation Comments to Bitbucket Server Plugin Finds violations reported by code analyzers and comments Bitbucket Server (or Stash) pull requests (or commits) with them.	1.54
☐	Violations plugin	0.7.11

Jenkins: Jobs - Adding and updating Plugins



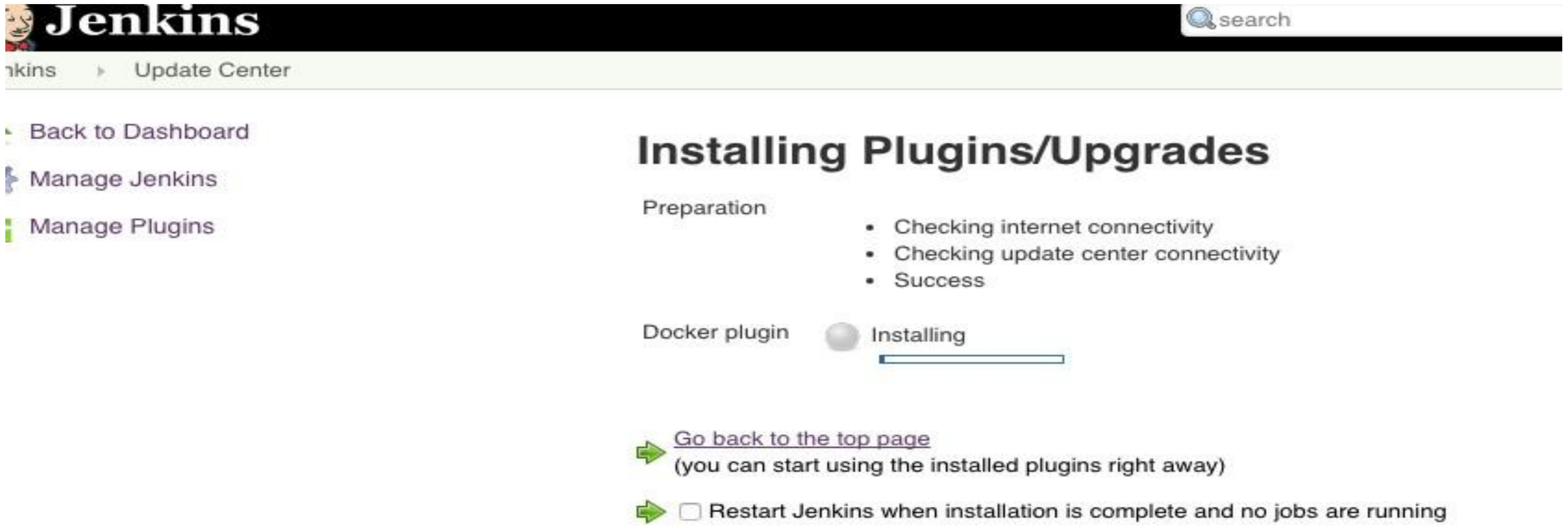
Install ↓	Name	Version
☐	Amazon EC2 Container Service plugin Jenkins plugin to run dynamic slaves in a Amazon ECS/Docker environment	1.11
☐	Amazon EC2 Container Service plugin with autoscaling capabilities Jenkins plugin to run dynamic slaves in a Amazon ECS/Docker environment	1.0
☐	Yet Another Docker Plugin Yet Another Docker plugin. Provides docker Cloud provisioning	0.1.0-rc39
☐	Aqua Security Scanner Scans Docker images for vulnerabilities	2.0
☐	CloudBees Docker Build and Publish plugin This plugin provides the ability to build projects with a Dockerfile, and publish the resultant tagged image (repo) to a Docker registry.	1.3.2
☐	docker-build-step	1.43
☐	CloudShare Docker-Machine Plugin Run Docker commands on a dedicated CloudShare VM	1.1.0
☐	Docker plugin Provide Cloud Provisioning and other Docker features	0.16.2
☐	Kubernetes plugin Jenkins plugin to run dynamic slaves in a Kubernetes/Docker environment	1.0
☐	CloudBees Docker Custom Build Environment Plugin	1.6.5

Jenkins: Jobs - Adding and updating Plugins



- [ ] Yet Another Docker Plugin
  Yet Another Docker plugin. Provides docker Cloud provisioning
- [ ] Aqua Security Scanner
  Scans Docker images for vulnerabilities
- [ ] CloudBees Docker Build and Publish plugin
  This plugin provides the ability to build projects with a Dockerfile, and publ
  to a Docker registry.
- [ ] docker-build-step
- [ ] CloudShare Docker-Machine Plugin
  Run Docker commands on a dedicated CloudShare VM
- [✓] Docker plugin
  Provide Cloud Provisioning and other Docker features
- [ ] Kubernetes plugin
  Jenkins plugin to run dynamic slaves in a Kubernetes/Docker environmen
- [ ] CloudBees Docker Custom Build Environment Plugin
- [ ] Docker Slaves Plugin
  Uses Docker containers to run Jenkins build agents
- CloudBees Docker Traceability

Jenkins: Jobs - Adding and updating Plugins


Jenkins

search

nkins ▸ Update Center

Back to Dashboard

Manage Jenkins

Manage Plugins

Installing Plugins/Upgrades

Preparation

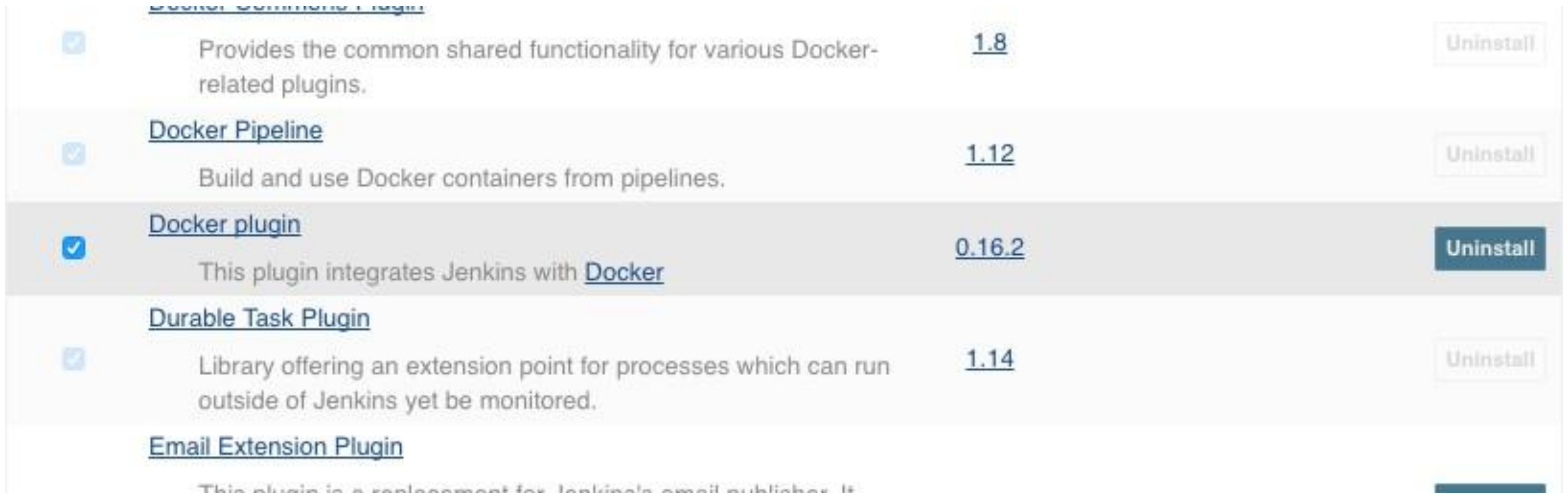
- Checking internet connectivity
- Checking update center connectivity
- Success

Docker plugin Installing

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: Jobs - Adding and updating Plugins



✓	Provides the common shared functionality for various Docker-related plugins.	1.8	Uninstall
✓	Docker Pipeline Build and use Docker containers from pipelines.	1.12	Uninstall
✓	Docker plugin This plugin integrates Jenkins with Docker	0.16.2	Uninstall
✓	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.14	Uninstall
	Email Extension Plugin		

Jenkins: Build Deployments - Java with Tomcat

TASK #1: Build the Java Application:

```
docker run -rm -v $PWD/.m2:/root/.m2 -v $PWD/my_host_folder/SreeJavaExample:/project -w /project maven mvn clean package
```

TASK #2: Deploy the application to Tomcat

```
docker rm -f my-tcc
docker run -d \
-p 8123:8080 \
--name my-tcc \
-v $PWD/my_host_folder/SreeJavaExample/target/SreeJavaExample.war:/usr/local/tomcat/webapps/sree-example.war \
tomcat
```

NOTES:

- When using docker commands in Jenkins NEVER use the -ti option.
- Initially, while testing, preferably use the full path instead of $PWD because $PWD points to your task's workspace which may not have your files.
- To browse the application: http://localhost:8123/gsa-example/

Jenkins: Build Deployments - Java with Tomcat

In case you are not a Java developer and you want to quickly create a sample Java Web Application to test the pipeline flow, use the ‘scaffolding’ option of Maven

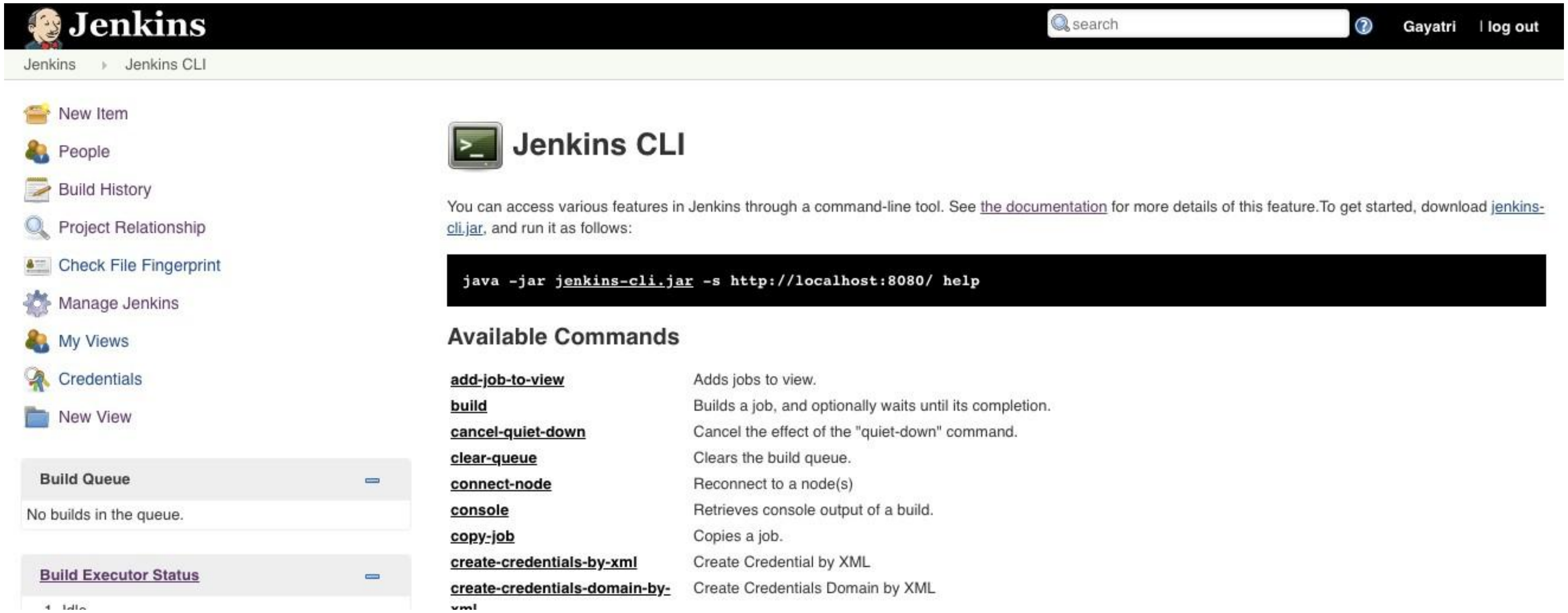
```
docker run --rm -it -v $PWD/my_host_folder:/external \
-v $PWD/.m2:/root/.m2 \
-w /external maven mvn archetype:generate \
-DgroupId=com.schogini.dockermvn.example \
-DartifactId=GsaJavaExample \
-DarchetypeArtifactId=maven-archetype-webapp \
-DinteractiveMode=false
```

Remember, the .m2 folder that you are mapping should be a valid maven repository. If you are not sure what that is then, remove it from the command before you execute it.

Jenkins: CLI

Ensure that you have the latest version of Java installed else, you may get an error like this:

Exception in thread "main" java.lang.UnsupportedClassVersionError: hudson/cli/CLI : Unsupported major.minor version 52.0



Jenkins CLI

You can access various features in Jenkins through a command-line tool. See the documentation for more details of this feature.To get started, download jenkins-cli.jar, and run it as follows:

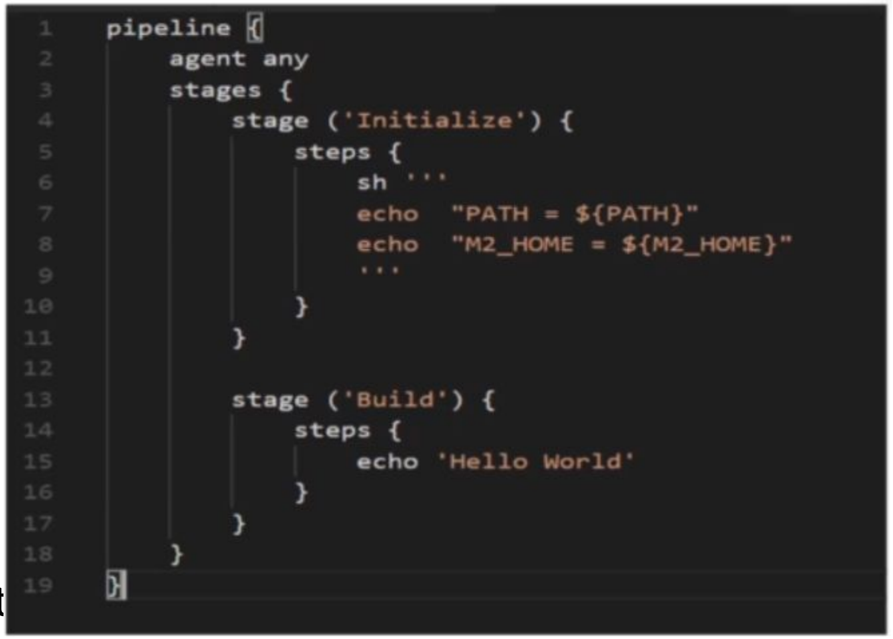
```
java -jar jenkins-cli.jar -s http://localhost:8080/ help
```

Available Commands

add-job-to-view	Adds jobs to view.
build	Builds a job, and optionally waits until its completion.
cancel-quiet-down	Cancel the effect of the "quiet-down" command.
clear-queue	Clears the build queue.
connect-node	Reconnect to a node(s)
console	Retrieves console output of a build.
copy-job	Copies a job.
create-credentials-by-xml	Create Credential by XML
create-credentials-domain-by-xml	Create Credentials Domain by XML

Jenkins: Pipeline As Code

- Pipeline code uses a DSL
- DSL allows you to perform the tests
- Jenkins text file defied in a txt file, called a Jenkinsfil
- Can defined version controls of Jenkinsfile
- less error –prone execution of jobs
- logic based execution of steps



```
pipeline {
    agent any
    stages {
        stage ('Initialize') {
            steps {
                sh '''
                echo  "PATH = ${PATH}"
                echo  "M2_HOME = ${M2_HOME}"
                '''
            }
        }

        stage ('Build') {
            steps {
                echo 'Hello World'
            }
        }
    }
}
```

pipeline : This pipe line is a set of instructions given in t
delivery of entail build process

- Node : The machine on which Jenkins is running
- Agent :An agent is an directive multiple builds will runs in single Jenkins instance
- Stage :A stage block containers serous of steps in the pipeline , That is build ,test , and deploy process are in one stage
- Step : A step is a single task that executes a specific process
-

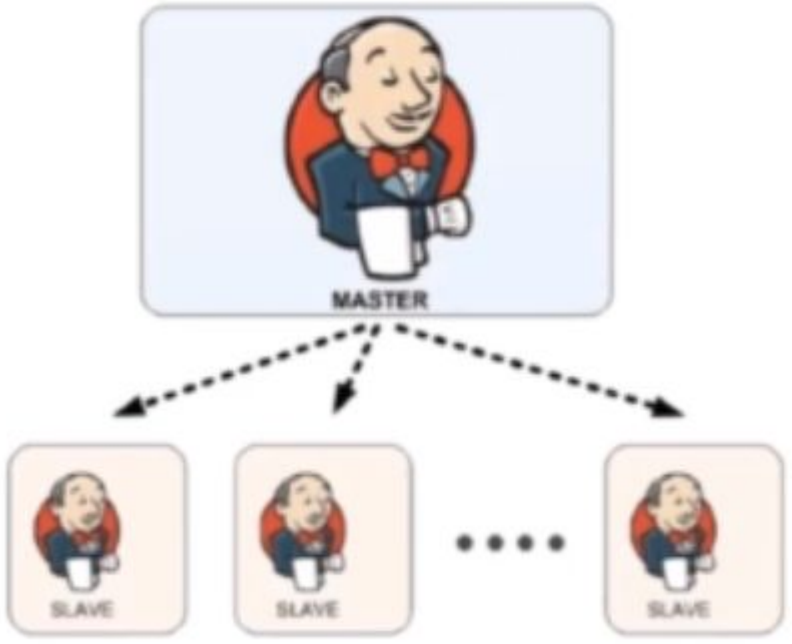
Jenkins – Distributed Builds

Master:

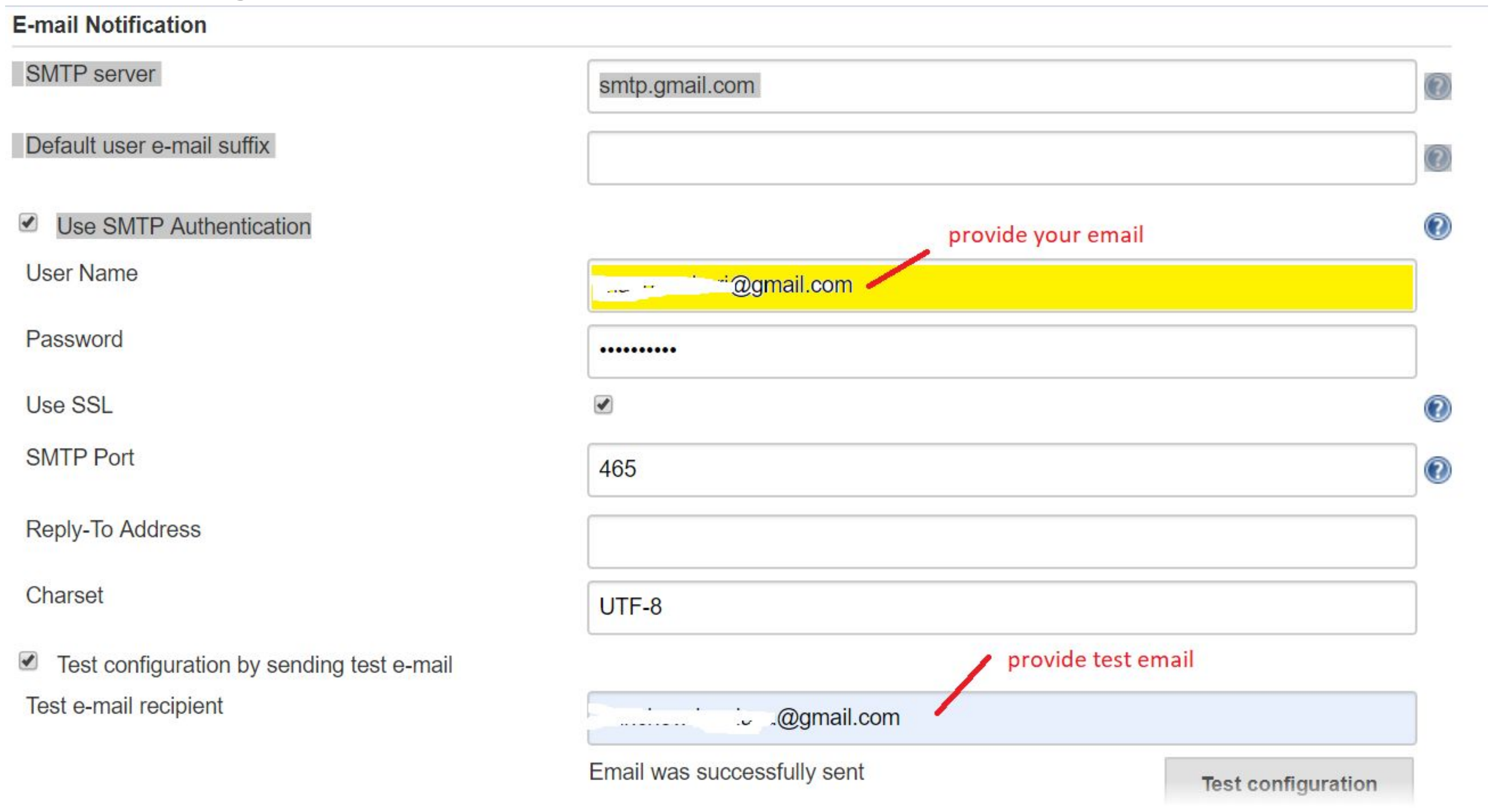
- Schedule build job
- Dispatches Builds to the slave for actual j
- Monitoring the slaves and recording the

Slave :

Executes builds jobs dispatched by master



Jenkins – Email configuration



E-mail Notification

SMTP server	smtp.gmail.com
Default user e-mail suffix	
✓ Use SMTP Authentication	
User Name	[redacted]@gmail.com
Password	••••••••••
Use SSL	✓
SMTP Port	465
Reply-To Address	
Charset	UTF-8
✓ Test configuration by sending test e-mail	
Test e-mail recipient	[redacted].@gmail.com

provide your email

provide test email

Email was successfully sent

Jenkins - Best Practices

- Always secure Jenkins.
- In larger systems, don't build on the master.
- Always build from Source Control – Clean Builds
- Connect Issue Management or Help Desk System with Jenkins
- Backup Jenkins Home regularly.
- Limit project names to a sane (e.g. alphanumeric) character set
- The most reliable builds will be clean builds, which are built fully from Source Code Control.
- Integrate tightly with your issue tracking system, like JIRA or bugzilla, to reduce the need for maintaining a Change Log
- Always configure your job to generate trend reports and automated testing when running a Java build
- Set up Jenkins on the partition that has the most free disk-space
- Archive unused jobs before removing them.
- Setup a different job/project for each maintenance or development branch you create
- Prevent resource collisions in jobs that are running in parallel.
- Avoid scheduling all jobs to start at the same time
- Set up email notifications mapping to ALL developers in the project, so that everyone on the team has his pulse on the project's current status.
- Take steps to ensure failures are reported as soon as possible.
- Write jobs for your maintenance tasks, such as cleanup operations to avoid full disk problems.
- Tag, label, or baseline the codebase after the successful build.