



DevOps

Docker: Jenkins

Trainer: Abhijith V G – AWS, eCommerce, Mobile & DevOps Architect



Certified

Developer - Associate

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

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



Jenkins

Can automate most of the repetitive tasks! This facilitates **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

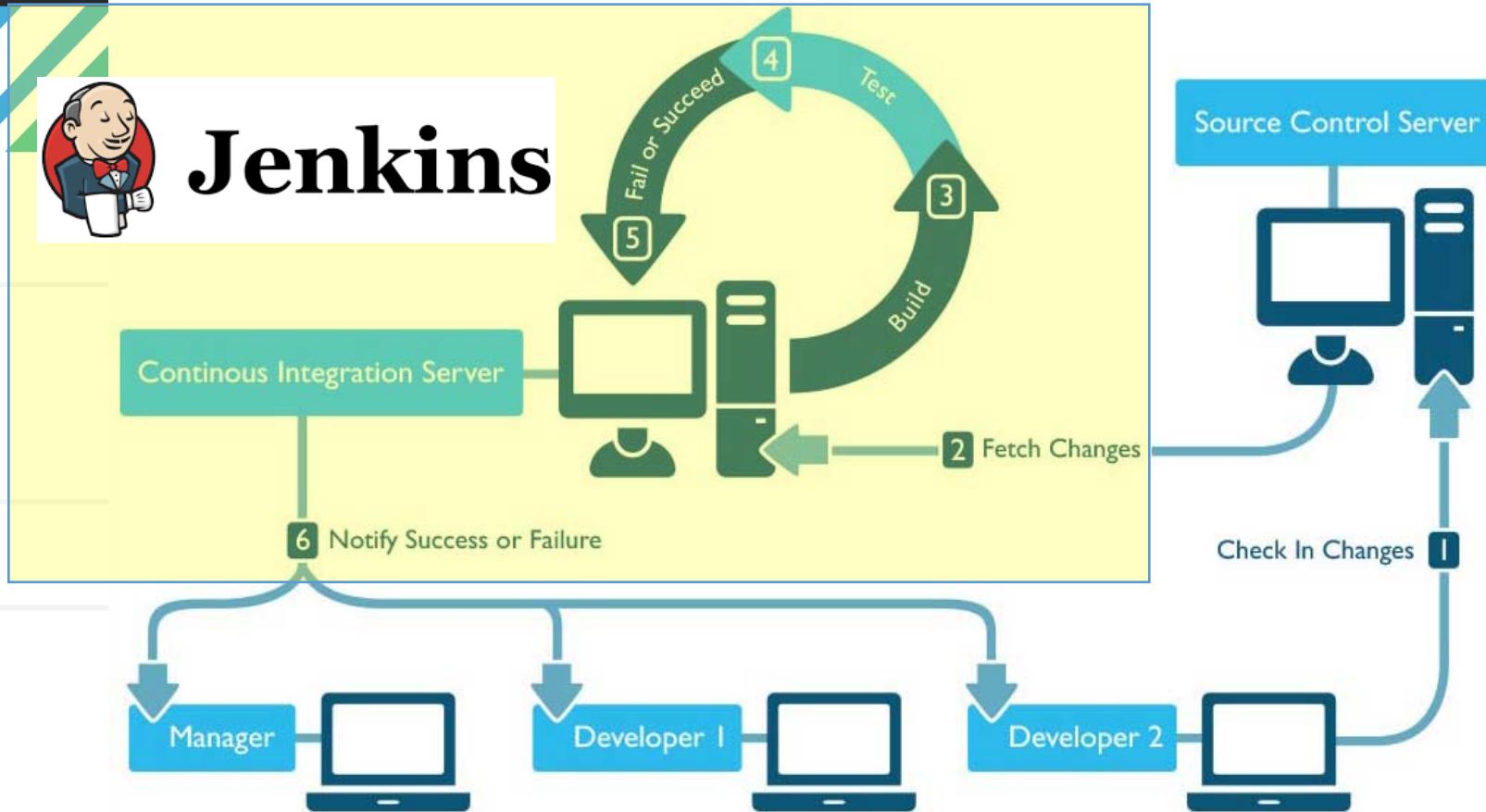


Travis CI



Jenkins

Jenkins: Introduction - Understanding continuous integration



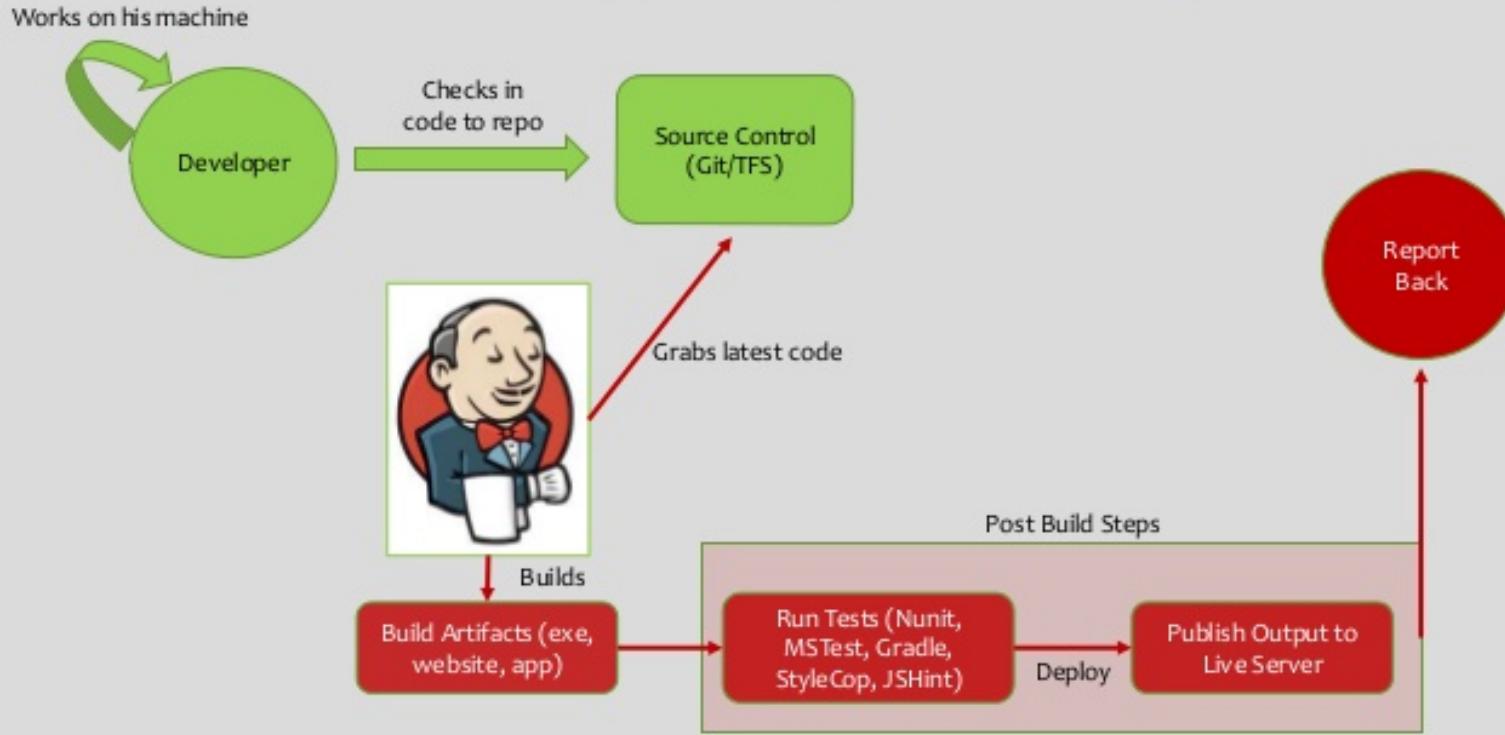
Jenkins: Introduction - Introduction about Jenkins

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: Introduction - Introduction about Jenkins

How does Jenkins typically fit into my work?

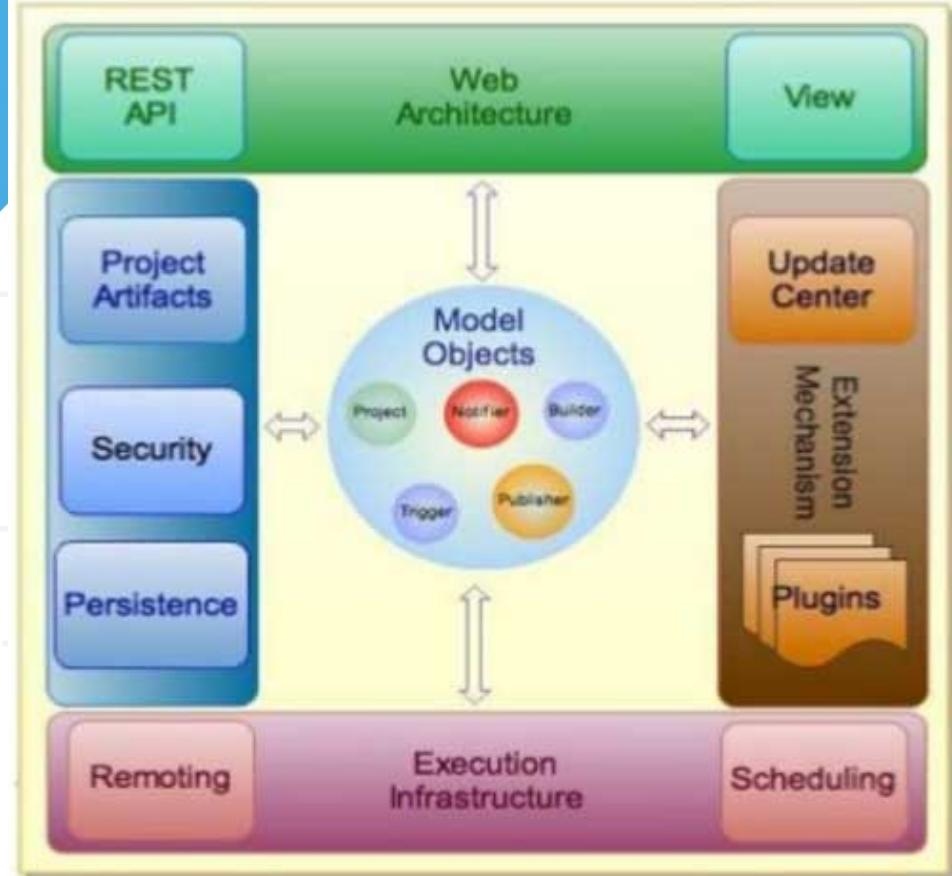


A build lifecycle is the process of building and distributing a particular artifact (project).

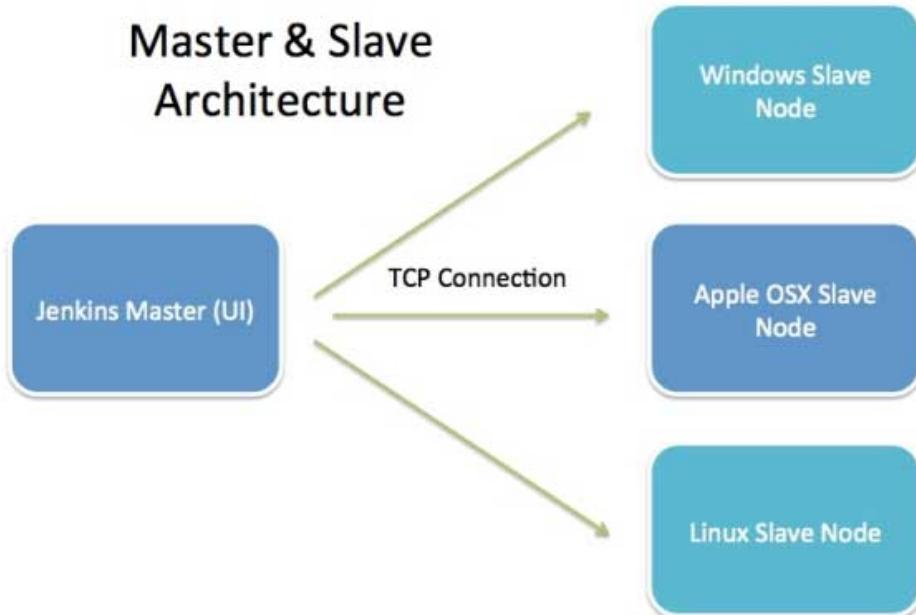
Build Lifecycle phases:

- **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



Master & Slave Architecture



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: Installation – Docker + Jenkins

Remember to map a host folder for persistence.

Check Docker logs to ensure that the container has finished its launching before browsing.

```
mkdir -p jenkins_home  
docker run -dit -e JENKINS_USER=$(id -u) --rm -p 8125:8080 -p 50000:50000 -v  
$PWD/jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -  
-name my-jenkins trion/jenkins-docker-client
```

OFFICIAL Docker Jenkins is [jenkins/jenkins](#)

Jenkins: Installation - Docker + Jenkins: Docker Logs

```
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

352e147cc8304f1bbf18edac9b8b540a

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
*****
Aug 25, 2017 5:09:14 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Aug 25, 2017 5:09:15 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
--> setting agent port for jnlp
--> setting agent port for jnlp... done
Aug 25, 2017 5:09:19 PM hudson.model.DownloadService$Downloadable load
INFO: Obtained the updated data file for hudson.tools.JDKInstaller
Aug 25, 2017 5:09:19 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Download metadata. 14,898 ms
Aug 25, 2017 5:09:21 PM hudson.model.UpdateSite updateData
INFO: Obtained the latest update center data file for UpdateSource default
Aug 25, 2017 5:09:21 PM hudson.WebAppMain$3 run
INFO: Jenkins is fully up and running
Aug 25, 2017 5:10:09 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Started Workspace clean-up
Aug 25, 2017 5:10:09 PM hudson.model.AsyncPeriodicWork$1 run
INFO: Finished Workspace clean-up. 18 ms
Sree=>
```

Jenkins: Installation - Docker + Jenkins: Jenkins Home

```
Sree=>ls -l jenkins_home/
total 64
-rw-r--r--  1 sree  staff  1716 Aug 25 22:39 config.xml
-rw-r--r--  1 sree  staff   102 Aug 25 22:38 copy_reference_file.log
-rw-r--r--  1 sree  staff  159 Aug 25 22:38 hudson.model.UpdateCenter.xml
-rw-----  1 sree  staff  1712 Aug 25 22:39 identity.key.enc
drwxr-xr-x  3 sree  staff   102 Aug 25 22:38 init.groovy.d
-rw-r--r--  1 sree  staff    94 Aug 25 22:39 jenkins.CLI.xml
-rw-r--r--  1 sree  staff     4 Aug 25 22:39 jenkins.install.UpgradeWizard.state
drwxr-xr-x  2 sree  staff    68 Aug 25 22:38 jobs
drwxr-xr-x  3 sree  staff   102 Aug 25 22:39 logs
-rw-r--r--  1 sree  staff   907 Aug 25 22:39 nodeMonitors.xml
drwxr-xr-x  2 sree  staff    68 Aug 25 22:38 nodes
drwxr-xr-x  2 sree  staff    68 Aug 25 22:38 plugins
-rw-r--r--  1 sree  staff    64 Aug 25 22:38 secret.key
-rw-r--r--  1 sree  staff     0 Aug 25 22:38 secret.key.not-so-secret
drwx----- 11 sree  staff   374 Aug 25 22:39 secrets
drwxr-xr-x  5 sree  staff   170 Aug 25 22:39 updates
drwxr-xr-x  3 sree  staff   102 Aug 25 22:39 userContent
drwxr-xr-x  3 sree  staff   102 Aug 25 22:39 users
drwxr-xr-x 25 sree  staff   850 Aug 25 22:38 war
```

Jenkins: Installation - Docker + Jenkins: Initial Password

```
[Sree=>cat jenkins_home/secrets/initialAdminPassword  
352e147cc8304f1bbf18edac9b8b540a
```

Jenkins: Installation - Docker + Jenkins: Browsing

Check host port

Sree=>docker ps	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
CONTAINER ID 6f94cb6170d2	jenkins/jenkins	"bin/tini -- /usr..."	About an hour ago	Up About an hour	0.0.0.0:32775->8080/tcp, 0.0.0.0:32774->50000/tcp	my-jenkins
Sree=>						

The screenshot shows a web browser window with the URL `localhost:32775/login?from=%2F`. The page is titled "Getting Started" and features a large heading "Unlock Jenkins". Below it, a message states: "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`". A placeholder text box is provided for pasting the password. At the bottom right is a "Continue" button.

Jenkins: Installation - Docker + Jenkins: Browsing

The screenshot shows a web browser window titled "SetupWizard [Jenkins]" with the URL "localhost:32775". The main content is titled "Getting Started" and "Customize Jenkins". It explains that plugins extend Jenkins with additional features to support many different needs. Two options are presented: "Install suggested plugins" (which installs plugins the Jenkins community finds most useful) and "Select plugins to install" (which allows selecting and installing plugins most suitable for your needs). A Jenkins logo watermark is visible in the background.

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

The screenshot shows the Jenkins Setup Wizard interface. At the top, it says "Setupwizard [Jenkins]" and the URL is "localhost:32775". The main title is "Getting Started". Below it, there's a large "Getting Started" heading. A table lists suggested Jenkins plugins:

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	build timeout plugin	Credentials Binding 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
⌚ 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	

At the bottom left, it says "Jenkins 2.75". On the right side, there's a note: "** - required dependency".

Jenkins: Installation - Install Suggested Plugins

```
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
Aug 25, 2017 5:32:06 PM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Aug 25, 2017 5:32:10 PM hudson.PluginManager dynamicLoad
INFO: Plugin docker-workflow:1.12 dynamically installed
Aug 25, 2017 5:32:10 PM hudson.model.UpdateCenter$DownloadJob run
INFO: Installation successful: Docker Pipeline
Aug 25, 2017 5:32:10 PM hudson.model.UpdateCenter$DownloadJob run
INFO: Starting the installation of Pipeline: Stage Tags Metadata on behalf of admin
Aug 25, 2017 5:32:12 PM hudson.model.UpdateCenter$UpdateCenterConfiguration download
INFO: Downloading Pipeline: Stage Tags Metadata
Aug 25, 2017 5:32:12 PM hudson.PluginManager dynamicLoad
INFO: Attempting to dynamic load /var/jenkins_home/plugins/pipeline-stage-tags-metadata.jpi
Aug 25, 2017 5:32:14 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Aug 25, 2017 5:32:14 PM jenkins.InitReactorRunner$1 onAttained
```

Jenkins: Installation – Ready to View Jenkins Dashboard

```
INFO: Downloading Email Extension Plugin
Aug 25, 2017 5:37:55 PM hudson.PluginManager dynamicLoad
INFO: Attempting to dynamic load /var/jenkins_home/plugins/email-ext.jpi
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
Aug 25, 2017 5:37:59 PM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
Aug 25, 2017 5:38:07 PM hudson.PluginManager dynamicLoad
INFO: Plugin email-ext:2.58 dynamically installed
Aug 25, 2017 5:38:07 PM hudson.model.UpdateCenter$DownloadJob run
INFO: Installation successful: Email Extension Plugin
Aug 25, 2017 5:38:07 PM hudson.model.UpdateCenter$DownloadJob run
INFO: Starting the installation of Mailer Plugin on behalf of admin
Aug 25, 2017 5:38:07 PM hudson.model.UpdateCenter$InstallationJob _run
INFO: Skipping duplicate install of: Mailer Plugin@1.20
Aug 25, 2017 5:38:07 PM hudson.model.UpdateCenter$DownloadJob run
INFO: Installation successful: Mailer Plugin
```

Jenkins: Installation – Create First Admin User

The screenshot shows a web browser window titled "SetupWizard [Jenkins]" with the URL "localhost:32775". The main content is a "Getting Started" section with a large heading "Create First Admin User". Below the heading are five input fields: "Username" (containing a placeholder " "), "Password", "Confirm password", "Full name", and "E-mail address". At the bottom left is the text "Jenkins 2.75". At the bottom right are two buttons: "Continue as admin" and a blue "Save and Finish" button.

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.75

Continue as admin

Save and Finish

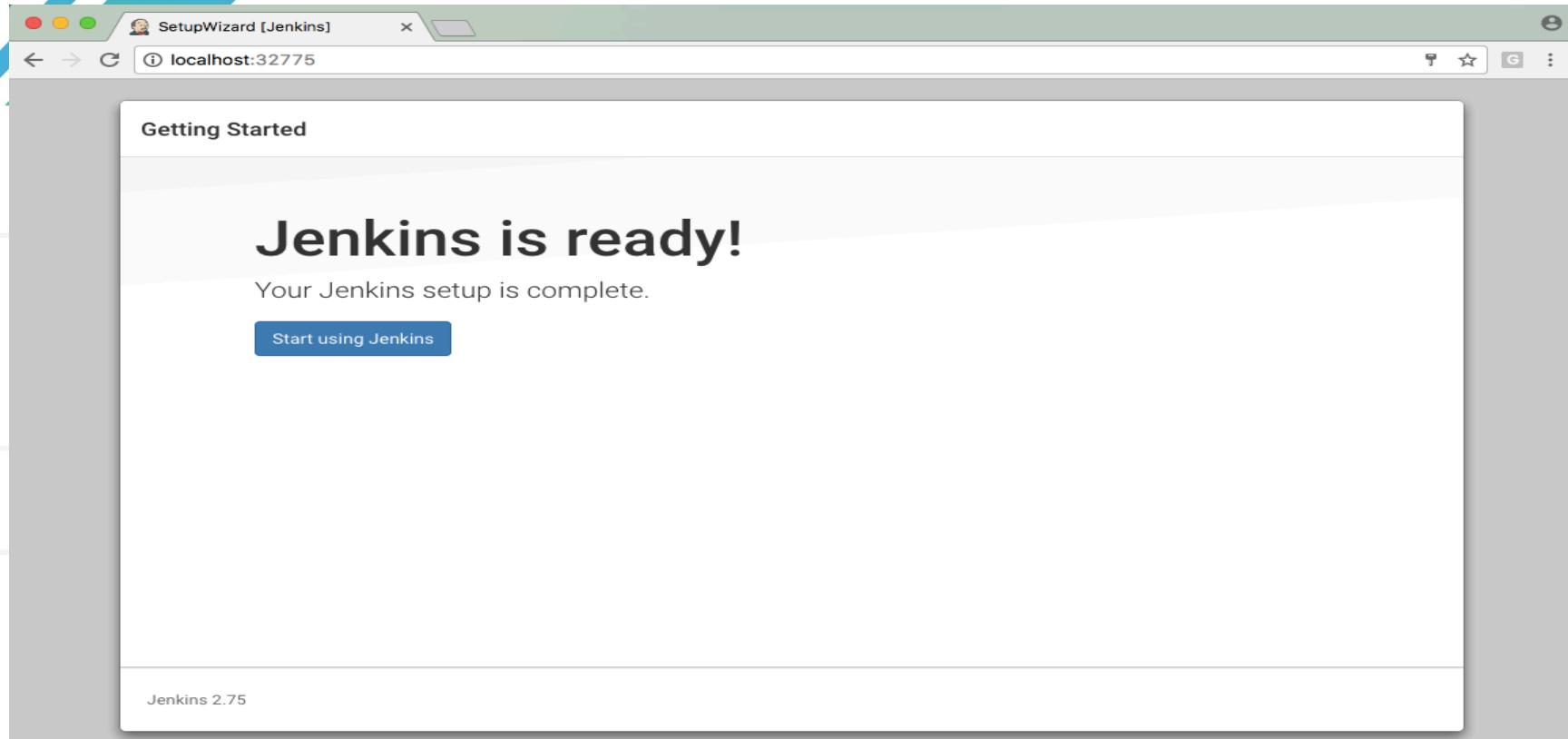
Jenkins: Installation – Create First Admin User

The screenshot shows a web browser window titled "SetupWizard [Jenkins]" with the URL "localhost:32775". The main content is a "Getting Started" section with a large heading "Create First Admin User". Below it are five input fields:

- Username: sree
- Password: (redacted)
- Confirm password: (redacted)
- Full name: Sreepakash Neelakanta
- E-mail address: sree@localhost

At the bottom left is the text "Jenkins 2.75". At the bottom right are two buttons: "Continue as admin" and "Save and Finish".

Jenkins: Installation – Ready to View Jenkins Dashboard



Jenkins: Installation – Ready to View Jenkins Dashboard

The screenshot shows the Jenkins dashboard at localhost:32775. The title bar says "Dashboard [Jenkins]". The main header features the Jenkins logo, a search bar, and user information for "Sreeprakash Neelakantan" with a "log out" link. A "ENABLE AUTO REFRESH" button is also present. On the left, a sidebar lists navigation options: "New Item", "People", "Build History", "Manage Jenkins", "My Views", and "Credentials". Below these are two collapsed sections: "Build Queue" (showing "No builds in the queue.") and "Build Executor Status" (showing "1 Idle" and "2 Idle"). The central area displays the "Welcome to Jenkins!" message and a call to action: "Please [create new jobs](#) to get started."

New Item

People

Build History

Manage Jenkins

My Views

Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

Please [create new jobs](#) to get started.

Jenkins: Installation – Pause Jenkins

```
|Sree=>docker pause my-jenkins  
my-jenkins  
Sree=>docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS  
S 6f94cb6170d2      jenkins/jenkins   "/bin/tini -- /usr... "   About an hour ago   Up About an hour (Paused)   0.0.0.0:32775->8080/tcp, 0.0.0.0:32774->50000/tcp   my-jenkins  
Sree=>
```

Jenkins: Installation – Mapped jenkins_home host folder

This folder should be preserved and backed up!

```
[Sree=>du -sh jenkins_home  
 200M   jenkins_home]
```

Jenkins: Securing Jenkins



Jenkins



Sreeprakash Neelakantan

| log out

[ENABLE AUTO REFRESH](#)

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

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` related to Jenkins.



[Load Statistics](#)

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



[Jenkins CLI](#)

18008338228

+65 31586636

+1(973) 598-3969

44 203-808-4216

www.cognixia.com • bdg@cognixia.com

Jenkins: Securing Jenkins - Authentication

1. Configure Global Security

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links like New Item, People, Build History, Manage Jenkins (which is selected and highlighted in purple), My Views, and Credentials. Below that are sections for Build Queue (empty) and Build Executor Status (2 Idle). The main content area is titled "Manage Jenkins". It features a yellow warning icon and text about a new Jenkins version (2.76) available for download. A list of management options is provided, each with an icon and a brief description:

- 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` related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Looking Glass**: (partially visible)

Jenkins: Securing Jenkins - Authentication

The screenshot shows the Jenkins 'Configure Global Security' configuration page. A large blue arrow points from the left towards the 'Enable security' checkbox. Another blue arrow points from the right towards the 'Matrix-based security' option under Authorization. A red arrow points from the bottom right towards the 'Matrix-based security' option.

Configure Global Security

1 search Sreeprakash Neelakantan | log out

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

Save Apply

After you create admin users, choose Matrix-based security

Jenkins: Securing Jenkins - Creating users

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

The screenshot shows the Jenkins 'Create User' interface. On the left sidebar, there are three options: 'Back to Dashboard', 'Manage Jenkins', and 'Create User'. The 'Create User' option is selected and highlighted with a blue background. The main content area is titled 'Create User'. It contains five input fields with the following values:

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

At the bottom of the form is a blue 'Create User' button.

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

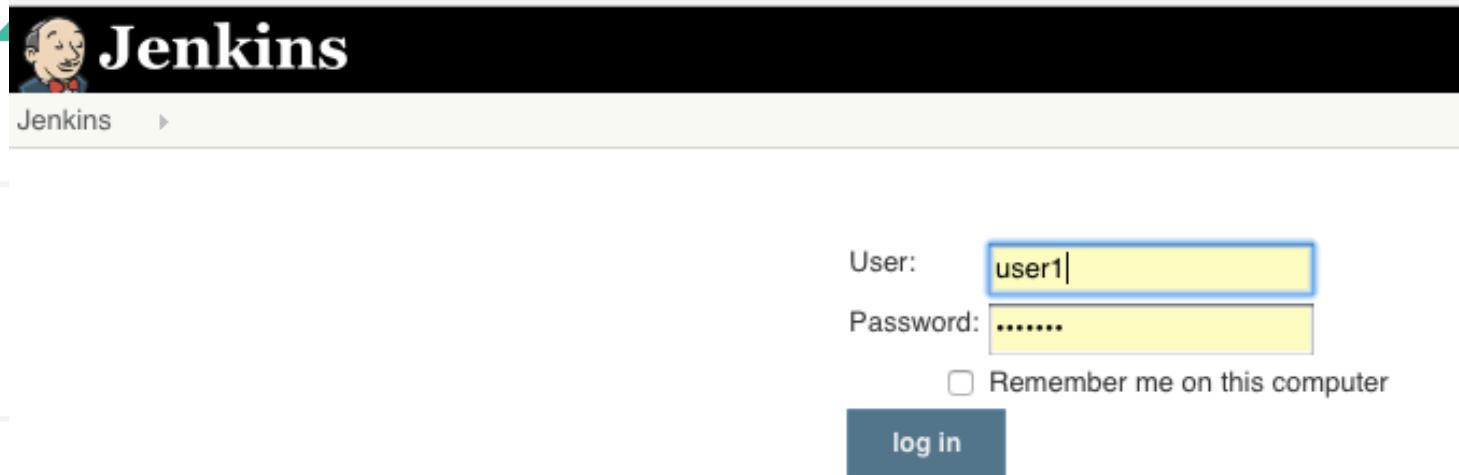
- Anyone can do anything
- Legacy mode
- Logged-in users can do anything
- Matrix-based security

User/group	Overall	Credentials					Agent					Job					Run			V...				
	Administer	Read	Create	Delete	Manage Domains	Update View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read Workspace	Delete Replay	Update	Configure Create
sree	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
user1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																					
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Be careful when allowing users to delete jobs

Jenkins: Securing Jenkins - Creating users

1. Logout and Login as User1



The image shows the Jenkins login interface. At the top left is the Jenkins logo. Below it, the word "Jenkins" is displayed in a large, bold, white font on a black background. Underneath the logo, there is a navigation bar with the word "Jenkins" and a right-pointing arrow. The main area is a white form with the following fields:
User: (The input field is highlighted with a blue border.)
Password: (The input field is highlighted with a yellow border.)
 Remember me on this computer

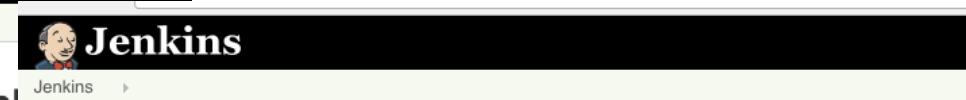
Jenkins: Securing Jenkins - Creating users

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

User1



Admin



Welcome to Jenkins!

Welcome to Jenkins!

Please [create new jobs](#) to get started.

Please [create new jobs](#) to get started.

Jenkins: Installation - Exploring Jenkins Dashboard

The screenshot shows the Jenkins dashboard at localhost:32775. The top navigation bar includes a search field, user profile for Sreeprakash Neelakantan, and log out link. On the left, there's a sidebar with links for New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main content area features a large "Welcome to Jenkins!" message and a call to action to "create new jobs". Below this are sections for "Build Queue" (empty) and "Build Executor Status" (1 Idle, 2 Idle).

Dashboard [Jenkins] ×

localhost:32775

Sreeprakash Neelakantan | log out

ENABLE AUTO REFRESH

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

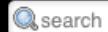
1 Idle

2 Idle

Jenkins: Jobs - Creating Jobs



Jenkins



search



Sreeprakash Neelakantan

| log out

[ENABLE AUTO REFRESH](#)

[add description](#)

[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: Jobs - Creating Jobs

 Jenkins

Jenkins ➤

search ? Sreeprakash Neelakantan | log out

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.

OK

 **Multibranch Pipeline**

Jenkins: Jobs - Creating Jobs

Jenkins > Sree1 > Click to go back, hold to see history

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

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 

[Advanced...](#)

Source Code Management

None 

Git 

Subversion 

Build Triggers

[Save](#) [Apply](#)

Jenkins: Jobs - Creating Jobs

The screenshot shows the Jenkins job configuration interface for a job named "Sree1". The top navigation bar includes links for Jenkins, Sree1, General, Source Code Management, Build Triggers (which is selected), Build Environment, Build, and Post-build Actions.

Build Triggers:

- 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** (selected)
- 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

The screenshot shows the Jenkins job configuration interface for a job named "Sree1". The top navigation bar includes links for Jenkins, Sree1, General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions.

Build Triggers:

- 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](#)

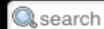
Add build step ▾

Save **Apply**

Jenkins: Jobs - Running the Jobs – Jobs Details



Jenkins



search



Sreeprakash Neelakantan

I log out

Jenkins > Sree1 >

[ENABLE AUTO REFRESH](#)

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Project Sree1

Freestyle job

[edit description](#)

[Disable Project](#)



[Workspace](#)



[Recent Changes](#)

Permalinks



[Build History](#)

[trend](#)

find

[RSS for all](#) [RSS for failures](#)

18008338228

+65 31586636

+1(973) 598-3969

44 203-808-4216

www.cognixia.com • bdg@cognixia.com

Jenkins: Jobs - Running the Jobs – Build Now

Jenkins Sree1 [ENABLE AUTO REFRESH](#)

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Delete Project](#)
[Configure](#)

Project Sree1
Freestyle job

[edit description](#)
[Disable Project](#)

[Workspace](#)
[Recent Changes](#)

Permalinks

Build History [trend](#) —
find
#1 Sep 9, 2017 2:08 AM
[RSS for all](#) [RSS for failures](#)

Jenkins: Jobs - Running the Jobs – Build Details



Jenkins



search



Sreeprakash Neelakantan

| log out

[Jenkins](#) > [Sree1](#) > [#1](#)[ENABLE AUTO REFRESH](#)[Back to Project](#)[Status](#)[Changes](#)[Console Output](#)[Edit Build Information](#)[Delete Build](#)

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

Started 10 sec ago

Took [0.13 sec](#)[Add description](#)

No changes.

Started by user [Sreeprakash Neelakantan](#)

Project Sree1

Freestyle job

[edit description](#)[Disable Project](#)[Workspace](#)[Recent Changes](#)[trend](#)

S for failures

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

Sree1

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

Jenkins

Jenkins > Sree1 >

[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](#)

localhost:32769 says:

Are you sure about deleting the Project 'Sree1'?

Cancel OK

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: Build Deployments - Deployment Plugins

There are numerous plugins to assist deployments, some of them are

AWS Elastic Beanstalk Deployment Plugin

AWS CodeDeploy Plugin for Jenkins

Deploy to container Plugin

Ansible plugin

Etc.

Jenkins: Jobs - Adding and updating Plugins



Sreeprakash Neelakantan

| log out

Jenkins > Plugin Manager

Back to Dashboard

Manage Jenkins

Updates

Available

Installed

Advanced

Filter:

**Install****Name ↓****Version****Installed**

No updates

Update information obtained: 1 hr 19 min ago

Check nowSelect: All, None

This page lists updates to the plugins you currently use.

Jenkins: Jobs - Adding and updating Plugins

 Jenkins

search  Sreeprakash Neelakantan | log out

Jenkins ▶ Plugin Manager

 Back to Dashboard  Manage Jenkins

Filter:

Updates	Available	Installed	Advanced
Install ↓			
.NET Development			
<input type="checkbox"/>	CCM Plug-in	3.1	
This plug-in generates the trend report for CCM, an open source static code analysis program.			
<input type="checkbox"/>	FxCop Runner plugin	1.1	
<input type="checkbox"/>	MSBuild Plugin	1.27	
<input type="checkbox"/>	MSTest plugin	0.20	
Generates test reports for MSTest.			
<input type="checkbox"/>	MSTestRunner plugin	1.3.0	
<input type="checkbox"/>	NAnt Plugin	1.4.3	
<input type="checkbox"/>	NCover plugin	0.3	
<input type="checkbox"/>	PowerShell plugin	1.3	
Violation Comments to Bitbucket Server Plugin			
<input type="checkbox"/>	Finds violations reported by code analyzers and comments Bitbucket Server (or Stash) pull requests (or commits) with them.	1.54	
<input type="checkbox"/>	Violations plugin	0.7.11	
<input type="checkbox"/>	Visual Studio Code Metrics Plugin	1.7	

Jenkins: Jobs - Adding and updating Plugins

Jenkins ▶ Plugin Manager

Back to Dashboard Manage Jenkins Filter: Docker

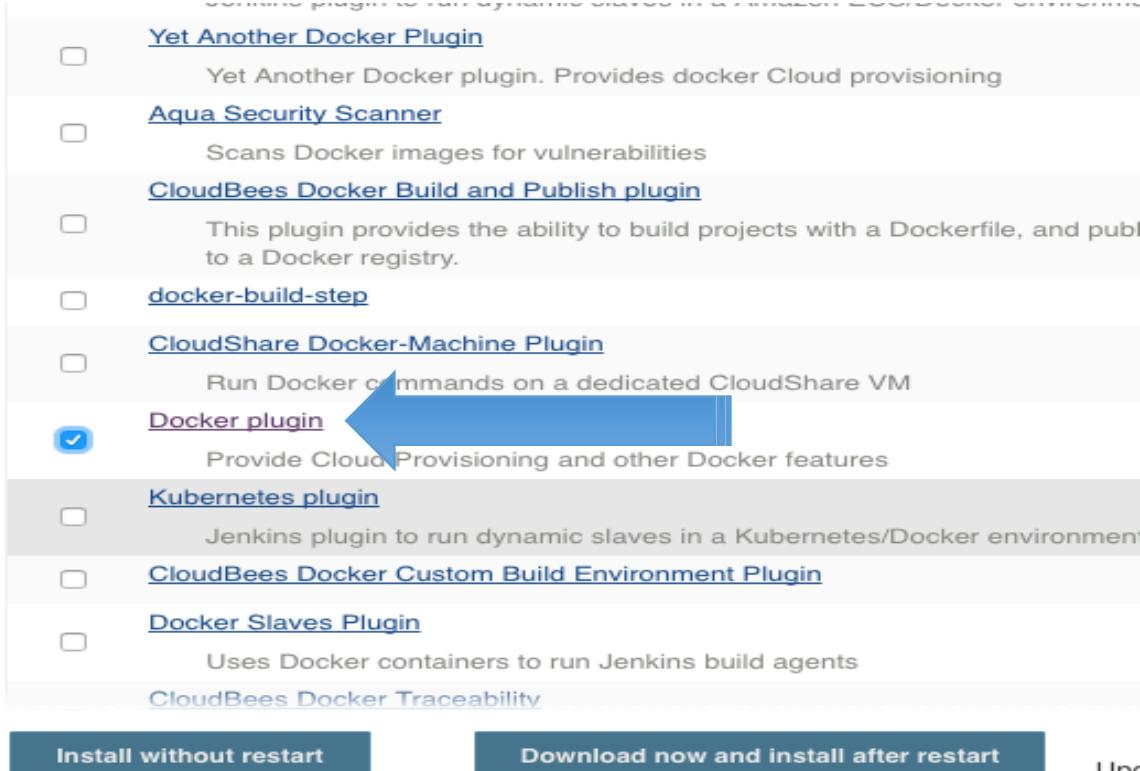
Install ↓	Name	Version
<input type="checkbox"/>	Amazon EC2 Container Service plugin Jenkins plugin to run dynamic slaves in a Amazon ECS/Docker environment	1.11
<input type="checkbox"/>	Amazon EC2 Container Service plugin with autoscaling capabilities Jenkins plugin to run dynamic slaves in a Amazon ECS/Docker environment	1.0
<input type="checkbox"/>	Yet Another Docker Plugin Yet Another Docker plugin. Provides docker Cloud provisioning	0.1.0-rc39
<input type="checkbox"/>	Aqua Security Scanner Scans Docker images for vulnerabilities	2.0
<input type="checkbox"/>	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
<input type="checkbox"/>	docker-build-step	1.43
<input type="checkbox"/>	CloudShare Docker-Machine Plugin Run Docker commands on a dedicated CloudShare VM	1.1.0
<input type="checkbox"/>	Docker plugin Provide Cloud Provisioning and other Docker features	0.16.2
<input type="checkbox"/>	Kubernetes plugin Jenkins plugin to run dynamic slaves in a Kubernetes/Docker environment	1.0
<input type="checkbox"/>	CloudBees Docker Custom Build Environment Plugin	1.6.5

Install without restart Download now and install after restart Check for updates



Docker plugin
Provides Cloud Provisioning and other Docker features

Jenkins: Jobs - Adding and updating Plugins



The screenshot shows a list of Jenkins plugins for Docker. The 'Docker plugin' is highlighted with a blue arrow pointing to it. The other plugins listed are:

- [Yet Another Docker Plugin](#)
- [Aqua Security Scanner](#)
- [CloudBees Docker Build and Publish plugin](#)
- [docker-build-step](#)
- [CloudShare Docker-Machine Plugin](#)
- [Kubernetes plugin](#)
- [CloudBees Docker Custom Build Environment Plugin](#)
- [Docker Slaves Plugin](#)
- [CloudBees Docker Traceability](#)

At the bottom, there are two buttons: 'Install without restart' and 'Download now and install after restart'.

Jenkins: Jobs - Adding and updating Plugins



Jenkins

Search

Back to Dashboard Manage Jenkins Manage Plugins

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

[Back to Dashboard](#)[Manage Jenkins](#)[Manage Plugins](#)

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Docker plugin

 Installing

Restarting Jenkins

 Pending

➔ [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

Selected Plugins			
<input checked="" type="checkbox"/>	Provides the common shared functionality for various Docker-related plugins.	1.8	<button>Uninstall</button>
<input checked="" type="checkbox"/>	Docker Pipeline Build and use Docker containers from pipelines.	1.12	<button>Uninstall</button>
<input checked="" type="checkbox"/>	Docker plugin This plugin integrates Jenkins with Docker	0.16.2	<button>Uninstall</button>
<input checked="" type="checkbox"/>	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be monitored.	1.14	<button>Uninstall</button>
<input checked="" type="checkbox"/>	Email Extension Plugin <small>This plugin is a replacement for Jenkins' own mailer. It</small>		

Jenkins: Build Deployments - Deploying a war file from Jenkins to Tomcat

Demonstration

- Run this to generate a sample Java application in ‘\$PWD/my_host_folder’ on your host machine.
This process of creating a sample application is called ‘scaffolding’

```
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=SreeJavaExample \
-DarchetypeArtifactId=maven-archetype-webapp \
-DinteractiveMode=false
```

- Run this to package the sample Java application you created above

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

Jenkins: Build Deployments - Deploying a war file from Jenkins to Tomcat

Demonstration

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

Jenkins - Best Practices

- Always secure Jenkins.
- In larger systems, don't build on the master.
- Always build from Source Control – Clean Builds
- The most reliable builds will be clean builds, which are built fully from Source Code Control.
- Connect Issue Management or Help Desk System with Jenkins
- Integrate tightly with your issue tracking system, like JIRA or bugzilla, to reduce the need for maintaining a Change Log
- Backup Jenkins Home regularly.
- Limit project names to a sane (e.g. alphanumeric) character set
- 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.

1. Start a Jenkins Docker container using Docker Image [trion/jenkins-docker-client](#)
2. Ensure that the a host folder called jenkins_home is created and mapped to the above container as per the pdf
3. Browse the exposed/mapped port and proceed with the setup/plugins
4. Secure the installation and create a user “user1” with access to jobs
5. Login as the main(admin) user and the “user1” and see the differences in the dashboard
6. Stop and Start the container and confirm that the user1 is accessible
7. Pause and Unpause the container and confirm that the user1 is accessible
8. **Note:** the host folder, jenkins_folder should be backedup/preserved using any OS tool of your choice.
9. BUILD AND DEPLOY A JAVA WEB APPLICATION ON TO A TOMCAT SERVER!
10. CHAIN THE JOBS... (edit the jsp to confirm that the changes are seen)



Cognixia

THANK YOU

