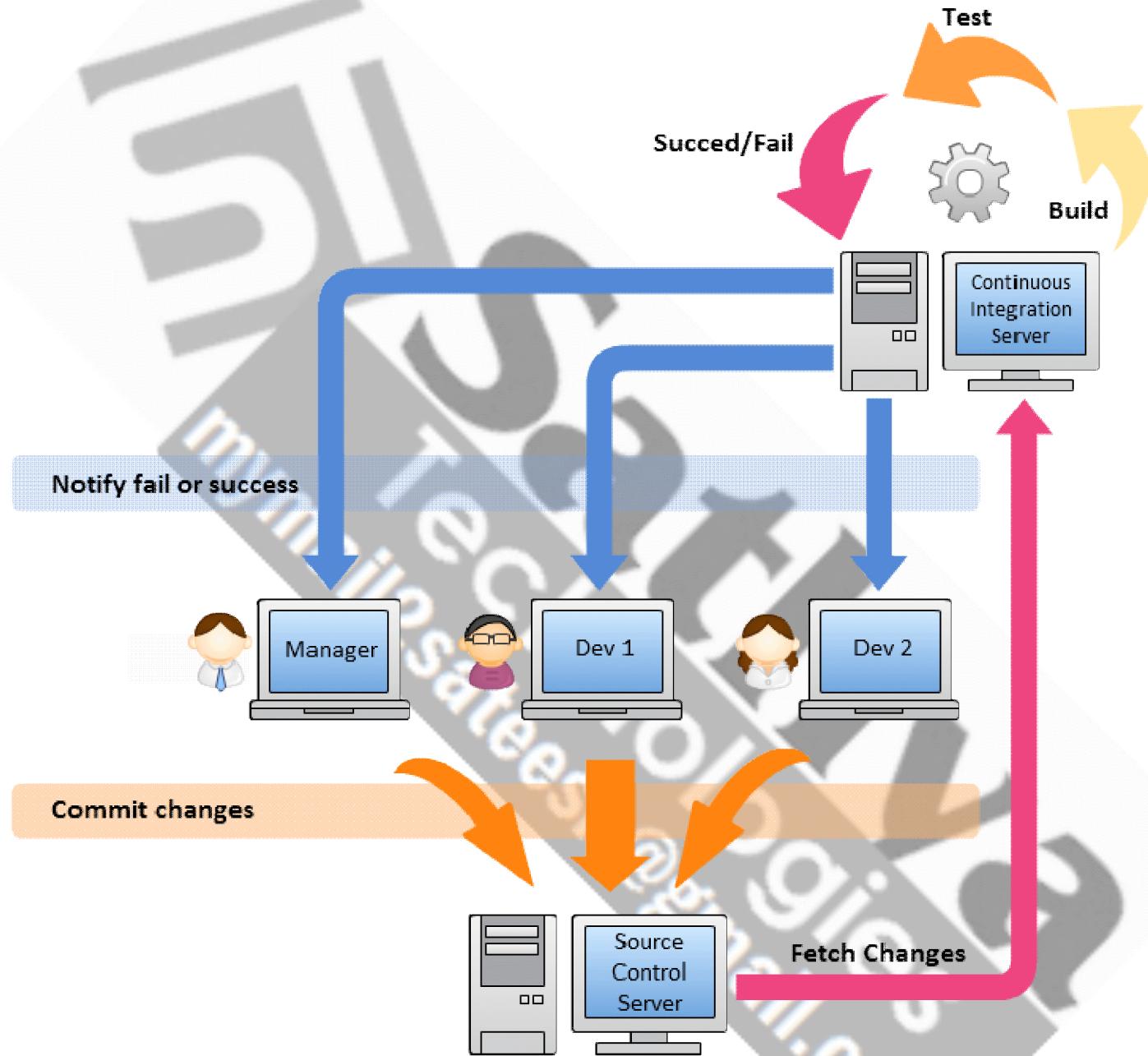




Jenkins



Jenkins Installation

Installation Methods

- 1. apt-get
- 2. Jenkins war file

Jenkins Installation using “apt”

Install Java , Tomcat, Maven

Step-1: This is the Debian/Ubuntu package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system:

- # wget -q -O - <https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add –

Jenkins Installation

Step-2 : Then add the following entry in your /etc/apt/sources.list:

deb <https://pkg.jenkins.io/debian-stable> binary/

Step-3 : Update your local package index, then finally install Jenkins:

```
#sudo apt-get update
```

```
#sudo apt-get install jenkins
```

Start Service

#service jenkins start

Open Browser → <http://localhost:8080>

Jenkins Installation using “war”

- i) Install tomcat Server
- ii) Download jenkins.war file
- iii) deploy jenkins.war in tomcat/webapps
- iv) start tomcat service
- v) browser --->
<http://localhost:8080/jenkins>

Java path:

```
#export JAVA_HOME=/opt/jdk1.8.0_121  
#export PATH=$JAVA_HOME/bin:$PATH
```

Tomcat path

```
#export CATALINA_HOME=/opt/apache-tomcat-8.5.12  
#export PATH=$CATALINA_HOME/bin:$PATH
```

Maven Path

```
#export M2_HOME=/opt/apache-maven-3.3.9  
#export PATH=$M2_HOME/bin:$PATH
```

ANT path

```
#export ANT_HOME=/opt/apache-ant-1.10.1  
#export PATH=$ANT_HOME/bin:$PATH
```

Git Project

1) install git in machine (Server)

```
#apt-get update
```

```
#apt-get install git-core -y
```

2) add git path

jenkins --> manage jenkins

--> Global tool Configuration

Name : git

path : /usr/bin/git ---> save

3) New item --> Name : git_demo --> ok

select "Source code management"

select "Git" ---> provide git Hub URL for a Repo.

--> save and build

ANT Proj

- 1) select new item (ant-proj) --> select "free style"**
- 2) provide a Git path in Source Code Management**
- 3) select 'Manage Jenkins' --> 'Global tool config'**
add Home_dir for JDK , ANT
- 4) 'Build' --> select 'invoke ant' --> select Ant Version
--> save & build**

MAVEN Proj

- 1) select new item (mvn-proj) --> select "free style"**
- 2) provide a Git path in Source Code Management**
- 3) select 'Manage Jenkins' --> 'Global tool config'
add Home_dir for MAVEN**
- 4) 'Build' --> select 'invoke top level maven target'
--> provide Maven version : mvn 3.3
goal : compile
--> save & build**

to Add Plugins:

Manage Jenkins --> Manage plugins --> Available
search for Maven integration plugin
--> select "install without restart"

- 1) select new item (mvn-pack) --> select "maven proj"
- 2) provide a Git path in Source Code Management
 - 3) 'Build' --> goal : package
 - 4) select "post steps" --> select "run only if build success"
 - Execute shell -->
 - cp /root/.jenkins/workspace/mvn_pack/target
 - /javaee7-simple-sample.war
 - /opt/apache-tomcat-9.0.0.M15/webapps/
 - echo "Application Deployed"

Build Pipe Line Plugin

- 1) select "Manage Jenkins" --> "Manage Plugins"
select "Available" --> select "Build Pipe Line Plugin"
- 2) select the Project --> "Configure"
Post Build Actions --> select "Build other Projects" -> save
- 3) select (+) in all Projects
View Name : pipe_view
select Build pipe line view --> ok
"select Initial Job" : slave_init --> ok

Jenkins Master – Slave

Step-1: Create Jenkins Master

- * install jdk , tomcat
- * deploy jenkins.war

Step-2: Create Jenkins Slave

- * install jdk (java)
- * create jenkins home dir
(/tmp/jenkins)

```
slave# mkdir /tmp/jenkins
```

Step-3: config slave machine

```
slave# vi /etc/ssh/sshd_config
```

```
    PasswordAuthentication yes
```

```
    PermitrootLogin yes
```

```
:wq ---> save and quit
```

```
slave# service ssh restart (for Ubuntu)
```

```
slave# systemctl restart sshd (for RedHat)
```

Step-4: change the root password

```
slave#passwd root
```

```
Enter new passwd : root
```

```
master# ssh <slave ip>
```

Step-5: to connect to slave using ssh keys

master# ssh-keygen

master# ssh-copy-id <slave ip>

master# ssh <slave ip>

Step-6: to Add Jenkins Slave

**Jenkins --> Manage Jenkins --> Manage
Nodes**

New Node ---> Name : node1

(.) Permanent Agent --> ok

No of executors: 5

Remote root directory: /tmp/jenkins

Jenkins

search admin | log

Jenkins > Nodes > sahya

[Back to List](#)

[Status](#)

[Delete Agent](#)

[Configure](#)

[Build History](#)

[Load Statistics](#)

[Log](#)

[Build Executor Status](#)

Name: sahya

Description:

of executors: 1

Remote root directory: /tmp/jenkins

Labels: slave1

Usage: Use this node as much as possible

Launch method: Launch slave agents via SSH

Host:

Credentials: - none - Add Jenkins
The selected Jenkins could not be found

Host Key Verification Strategy: Known hosts file verification Jenkins Credentials Provider

[Back to List](#)[Status](#)[Delete Agent](#)[Configure](#)[Build History](#)[Load Statistics](#)[Log](#)[Build Executor](#)

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

root

Password

....

ID

Description

[Add](#)[Cancel](#)[Save](#)

[Back to List](#)

 Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: **SSH Username with private key** (highlighted with a red box)

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: root

Private Key: Enter directly

Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowlBAAKCAQEAp/kKWyQhXtp2Sley3JUPredf0oHdB8HYdmG/zYSFRIEh5
w8z
gg/mh9+nOhHiO7QR5rYQeo3by8CPe+8KB71Z6EEd3pW83bAePoOOyuvSxL
cRbVg
```

[Back to List](#)[status](#)[Delete Agent](#)[Configure](#)[Build History](#)[Load Statistics](#)[Log](#)[Build Executor Status](#)

Agent sahya

This agent is offline because Jenkins failed to launch the agent process on it. [See log for more details](#)

[Launch agent](#)

Labels

slave1

Projects tied to sahya

None

[New Item](#) [People](#) [Build History](#) [Manage Jenkins](#) [My Views](#) [Credentials](#)[Build Queue](#)

No builds in the queue.

[Build Executor Status](#)

master

1 Idle

2 Idle

node1

1 Idle

sahya

1 Idle

[All](#) [demo](#) [sathya](#) [+](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		ant_demo	1 day 23 hr - #4	1 day 23 hr - #1	1.2 sec	
		demo	12 hr - #1	N/A	1 sec	
		mvn_compile	23 hr - #8	N/A	4.1 sec	
		mvn_deploy	23 hr - #8	N/A	37 ms	
		mvn_package	23 hr - #8	N/A	5.2 sec	
		test	1 day 23 hr - #1	1 day 23 hr - #2	0.14 sec	

Icon: [SML](#)[Legend](#)[RSS for all](#)[RSS for failures](#)[RSS for just latest builds](#)

[New Item](#)[People](#)[Build History](#)[Manage Jenkins](#)[My Views](#)[Credentials](#)[System](#)

Build Queue

No builds in the queue.

Build Executor Status

master

- 1 Idle
- 2 Idle

node1

- 1 Idle

sahya

- 1 Idle

Credentials

T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	57dfcc9b-e81b-4985-93f1-7444255d7a1e	root/*****
		Jenkins	(global)	6e13b171-0073-43bc-8efb-dfc8584b5d41	root/*****
		Jenkins	(global)	b3c92af5-52d9-4a76-9856-65a89d309ff0	root

Icon: S M L

Stores scoped to Jenkins

P	Store ↓	Domains
	Jenkins	(global)

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

[Plain text] [Preview](#)

Discard old builds [?](#)

GitHub project [?](#)

This project is parameterized [?](#)

Throttle builds [?](#)

Disable this project [?](#)

Execute concurrent builds if necessary [?](#)

Restrict where this project can be run [?](#)

Label Expression [?](#)

[Label sahya is serviced by 1 node](#)

[Advanced...](#)

Source Code Management

None

Git

Subversion [?](#)

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

Sonatype Nexus Installation

download jdk1.8 (from www.oracle.com/downloads....)

download nexus-3.x (from www.sonatype.com/download..)

```
$ tar -xzf jdk1.8....
```

```
$ tar -xzf nexus-3.x...
```

```
$ vi .profile
```

```
    export JAVA_HOME="/opt/jdk1.8.0_121"
```

```
    export PATH=$JAVA_HOME/bin:$PATH
```

```
$ source .profile
```

```
$ cd nexus-3.x../bin/
```

```
$ ./nexus run
```

Browser ---> <http://192.168.x : 8081>