



Jenkins

DevOps by SATISH @ SathyaTech

Git Configuration

1) install git in Jenkins Machine (Server)

```
#apt-get update
```

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

2) set git path

jenkins --> Manage Jenkins

--> Global tool Configuration

Name : git

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

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

Jenkins EMAIL Notifications

Manage Jenkins → Configure System

Configure which of these warnings are shown

-  [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.
-  [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 CLI](#)
Access/manage Jenkins from your shell, or from your script.

Jenkins EMAIL Notifications

E-mail Notification

SMTP server

smtp.gmail.com



Default user e-mail suffix

@gmail.com



Use SMTP Authentication



User Name

satish@gmail.com



Password

.....



Use SSL



SMTP Port

465



Reply-To Address

satsbitcoin@gmail.com



Charset

UTF-8



Test configuration by sending test e-mail

Test e-mail recipient

mymail2sateesh@gmail.com

Email was successfully sent

Test configuration



Save

Apply

Jenkins EMAIL Notifications

New Item → Free Style project → Post Build Action

The screenshot shows the Jenkins configuration interface for a 'Free Style project'. The top navigation bar includes tabs for General, Source Code Management, Build Triggers, Build Environment, Build (selected), and Post-build Actions. The 'Build' tab contains several build-related options like 'Archive the artifacts' and 'Publish JUnit test result report'. The 'Post-build Actions' section is expanded, showing a list of actions. One action, 'E-mail Notification', is highlighted with a blue selection bar. Other listed actions include 'Editable Email Notification', 'Set GitHub commit status (universal)', 'Set build status on GitHub commit [deprecated]', 'Trigger parameterized build on other projects', and 'Delete workspace when build is done'. At the bottom of the configuration page, there are 'Save' and 'Apply' buttons.

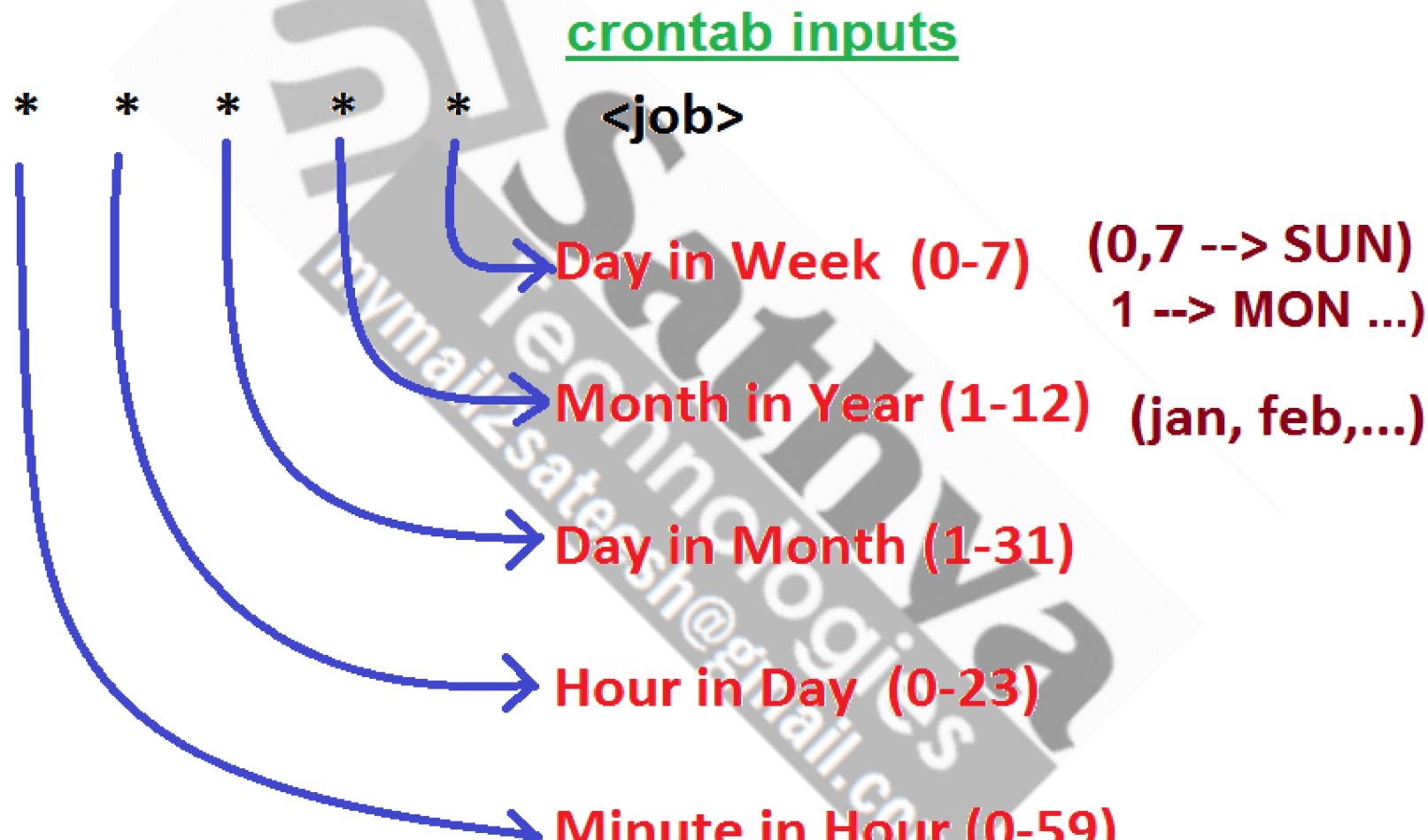
Jenkins EMAIL Notifications

New Item → Free Style project → Post Build Action

The screenshot shows the Jenkins configuration interface for a 'Free Style project'. The top navigation bar includes tabs for General, Source Code Management, Build Triggers, Build Environment, Build (selected), and Post-build Actions. The 'Build' tab contains several build-related options like 'Archive the artifacts' and 'Publish JUnit test result report'. The 'Post-build Actions' section is expanded, showing a list of actions. One action, 'E-mail Notification', is highlighted with a blue selection bar. Other listed actions include 'Editable Email Notification', 'Set GitHub commit status (universal)', 'Set build status on GitHub commit [deprecated]', 'Trigger parameterized build on other projects', and 'Delete workspace when build is done'. At the bottom of the configuration page, there are 'Save' and 'Apply' buttons.

How to Schedule Jobs in Jenkins

Jenkins uses "crontab" functionality



How to Schedule Jobs in Jenkins

- Build Periodically
- Poll SCM

| | |
|---------------|------------------------------------|
| 0 5 * * * | ----> for every day @ 5 am |
| 30 6,18 * * * | ----> for every day @ 6:30 am & pm |
| 0 10 * * 0 | ----> for every sunday @ 10 am |
| 0 0 31 3 * | ----> for every 31st March |
| 0 */5 * * * | ----> for every 5 hrs |
| */10 * * * * | ----> for every 10 minutes |
| * * * * * | ----> for every minute |

Build Periodically

the Jenkins Jobs will execute for specific time interval

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

Build Triggers

- Trigger builds remotely (e.g., from scripts) (?)
- Build after other projects are built (?)
- Build periodically (?)

Schedule  * * * * *

⚠ Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *"

Would last have run at Thursday, January 11, 2018 2:50:29 PM UTC; would next run at Thursday, January 11, 2018 2:50:29 PM UTC.

- GitHub hook trigger for GITScm polling (?)
- Poll SCM (?)

Build Environment

- Delete workspace before build starts
- Abort the build if it's stuck

Save **Apply** [Console Output](#)

Poll SCM

- the Jenkins server checks the repository at regular intervals for changes.
- After a commit occurs, the Jenkins server detects the changes that have occurred in the source code repository.
- Jenkins will pull those changes and will start preparing a new build.

Poll SCM

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

Source Code Management

None
 Git

Repositories

| | | |
|---|---|------------------------------------|
| Repository URL | <input type="text" value="https://github.com/sathyadevops/ant_proj.git"/> | (?) |
| Credentials | <input type="button" value="- none -"/> | <input type="button" value="Add"/> |
| <input type="button" value="Advanced..."/> | | |
| <input type="button" value="Add Repository"/> | | |

Branches to build

| | | | |
|---|---------------------------------------|---|-----|
| Branch Specifier (blank for 'any') | <input type="text" value="*/master"/> | X | (?) |
| <input type="button" value="Add Branch"/> | | | |

Repository browser (Auto)

Additional Behaviours

or

Poll SCM

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 [?](#)

Schedule



⚠ Do you really mean "every minute" when you say "* * * *"? Perhaps you meant "H * * * *" to poll once per hour

Would last have run at Thursday, January 11, 2018 2:56:37 PM UTC; would next run at Thursday, January 11, 2018 2:56:37 PM UTC.

Ignore post-commit hooks [?](#)

Build Environment

Delete workspace before build starts
 Abort the build if it's stuck

Save **Apply** **Console Output**

Use secret text(s) or file(s)

Jenkins User Management



[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 CLI](#)

Access/manage Jenkins from your shell, or from your script.



[Script Console](#)

Executes arbitrary script for administration/trouble-shooting/diagnostics.



[Manage Nodes](#)

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



[About Jenkins](#)

See the version and license information.



[Manage Old Data](#)

Scrub configuration files to remove remnants from old plugins and earlier versions.



[Manage Users](#)

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



[Prepare for Shutdown](#)

Stops executing new builds, so that the system can be eventually shut down safely.

Jenkins User Management

The screenshot shows the Jenkins User Management interface. At the top left is the Jenkins logo. To its right is a red box containing the number '2'. Next to it is a search bar with a magnifying glass icon and a question mark icon. Below the header, the breadcrumb navigation shows 'Jenkins > Jenkins' own user database'. On the left, there's a sidebar with links: 'Back to Dashboard', 'Manage Jenkins', and 'Create User', which is highlighted with a red box and a red arrow pointing towards it. The main content area has a title 'Users' and a descriptive text: 'These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who have made some commits on some projects and have no direct Jenkins access.' A table lists the users:

| User Id | Name |
|-----------------------|-----------------------|
| admin | admin |

Jenkins User Management

The image shows the Jenkins User Management interface. At the top, there is a navigation bar with the Jenkins logo, a '2' indicating two users, and a search bar. Below the navigation bar, the page title is 'Jenkins' own user database'. On the left side, there is a sidebar with links: 'Back to Dashboard', 'Manage Jenkins', and 'Create User'. The main content area is titled 'Create User'. It contains five input fields: 'Username' (satish), 'Password' (redacted), 'Confirm password' (redacted), 'Full name' (satish kumar), and 'E-mail address' (mymail2sateesh@gmail.com). A large red arrow points from the bottom-left towards the 'Create User' button, which is highlighted with a red border. The background features a watermark with the text 'SathyaTech' and 'mymail2sateesh@gmail.com'.

2

search

Jenkins Jenkins' own user database

Back to Dashboard

Manage Jenkins

Create User

Create User

Username: satish

Password:

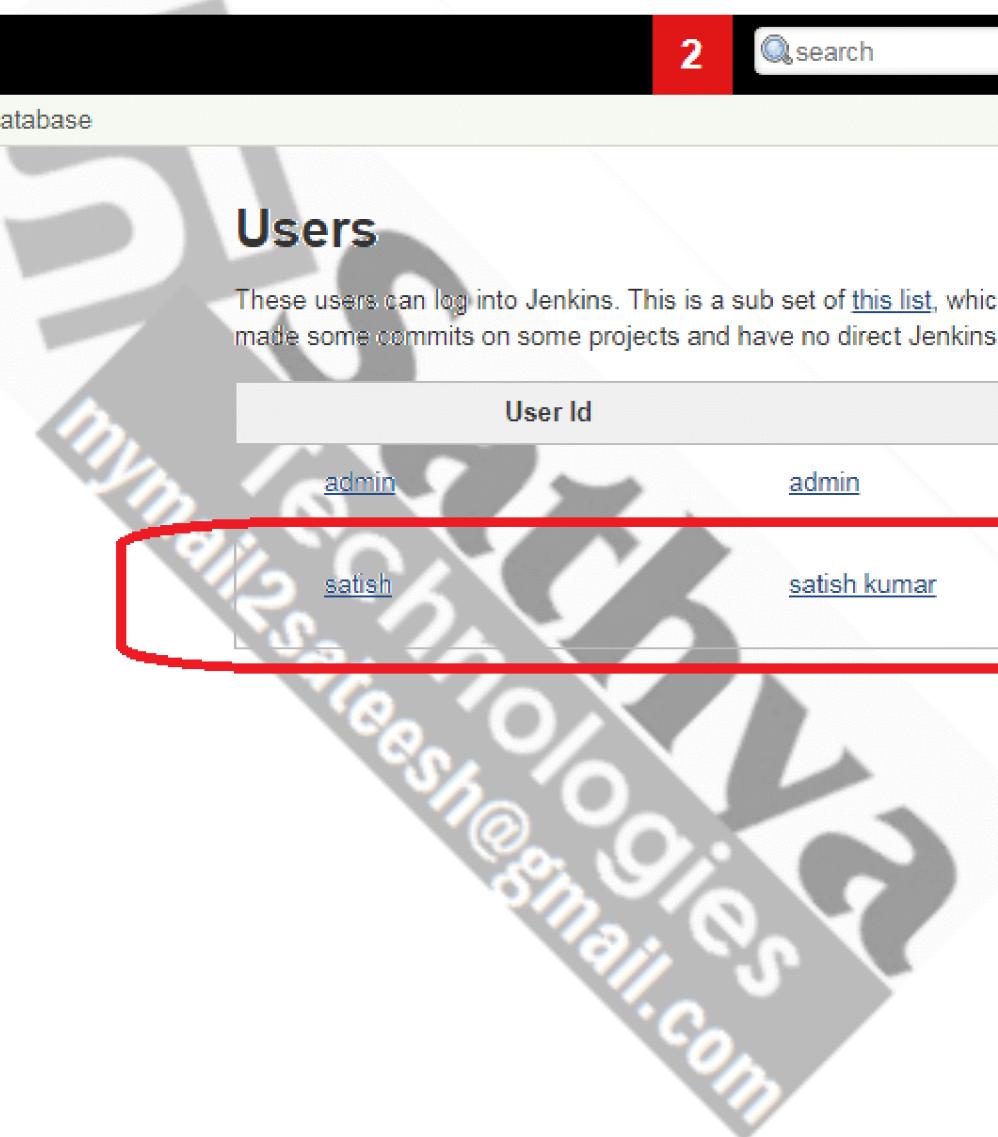
Confirm password:

Full name: satish kumar

E-mail address: mymail2sateesh@gmail.com

Create User

Jenkins User Management



The screenshot shows the Jenkins User Management interface. At the top, there is a navigation bar with the Jenkins logo, a user count of 2, a search bar, and a link for "admin". Below the navigation bar, the URL "Jenkins' own user database" is displayed. On the left, there is a sidebar with links: "Back to Dashboard", "Manage Jenkins", and "Create User". The main content area is titled "Users" and contains a table with two rows. The table has columns for "User Id" and "Name". The first row contains "admin" under both columns. The second row contains "satish" under "User Id" and "satish kumar" under "Name". A red oval highlights the second row.

| User Id | Name |
|---------|--------------|
| admin | admin |
| satish | satish kumar |

Jenkins Plugin Installation

select "Manage Jenkins" --> "Manage Plugins"

- 1) select "Available" --> select “Green Ball”
select “Install without restart”**
- 2) select "Available" --> select "Build Pipe Line Plugin"
select “Install without restart”**
- 3) select "Available" --> select “Maven integration
plugin” → select "install without restart"**

Jenkins Plug-in Installation

Configure which of these warnings are shown



[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.



[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 CLI](#)

Access/manage Jenkins from your shell, or from your script.

Jenkins Plug-in Installation

Jenkins 2 search ?

Back to Dashboard Manage Jenkins Update Center

Updates Available Installed Advanced

Filter: build pi

Name

[Build Pipeline Plugin](#) This plugin provides build pipeline functionality to Hudson and Jenkins. This a chain of jobs to be visualised in a new view. Manual jobs in the pipeline can b by a user with the appropriate permissions manually confirming.

[CodeScene Plugin](#) CodeScene detects potential maintenance problems and early warnings in yo codebase. The earlier you can react to those findings, the better. That's why C offers integration points that let you incorporate the analysis results into your l pipeline. This plugin lets you use CodeScene's Delta Analysis to catch potent problems before they are delivered to your main branch.

Install without restart Download now and install after restart Update

Page generated: Jan 11, 2018 3:04:37 PM UTC [RES](#)

DevOps by SATISH @ SathyaTech

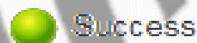
Jenkins Plug-in Installation

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

External Monitor Job Type Plugin



Success

Green Balls



Success

Javadoc Plugin



Success

Maven Integration plugin



Success

Run Condition Plugin



Success

Conditional BuildStep



Success

Parameterized Trigger plugin



Pending

jQuery plugin



Pending

Build Pipeline Plugin



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