

Jenkins 2

Create a simple project with Jenkins connected to your GitHub repository. When a commit is made to your repo, automatically build must get triggered from Jenkins and the output must be shared via email."


Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465


Advanced ▾  Edited


adding the email configuration.

E-mail Notification

SMTP server

smtp.gmail.com

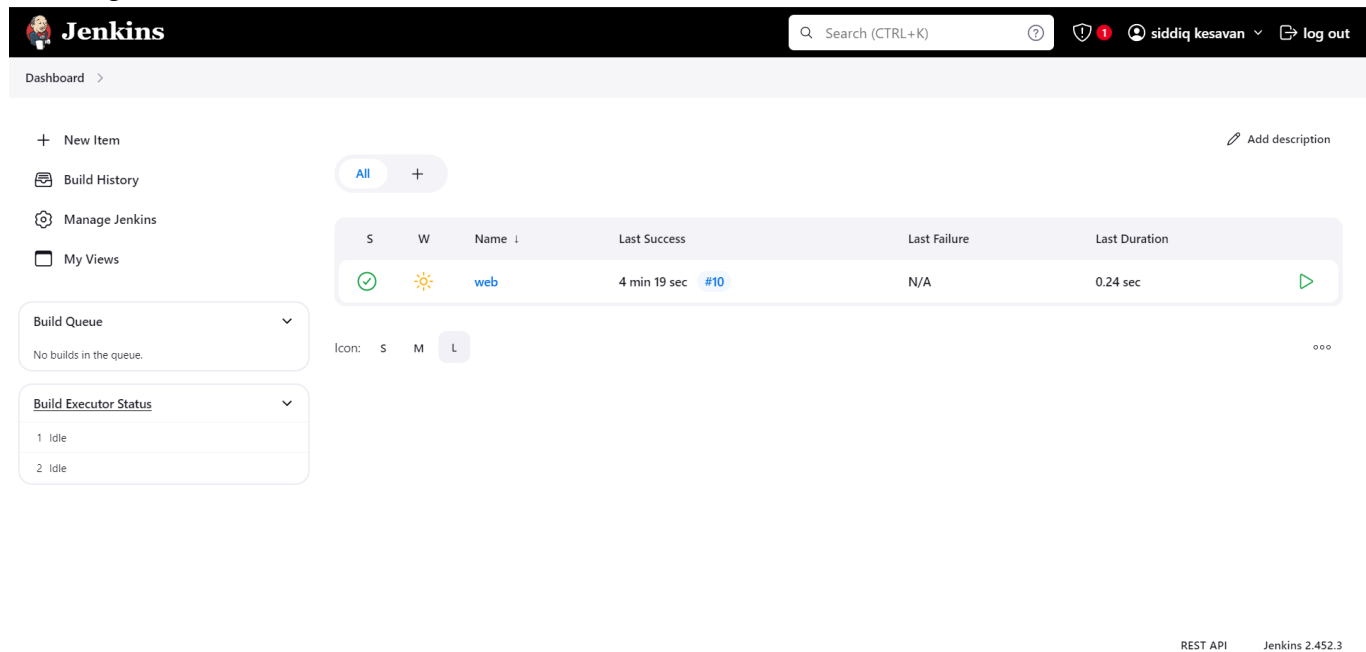
Default user e-mail suffix 

Advanced ▾  Edited

☐ Test configuration by sending test e-mail

test the data.

creating the new build :



The screenshot shows the Jenkins Dashboard. At the top, there's a header with the Jenkins logo, a search bar, and user information (siddiq kesavan). Below the header, the left sidebar contains navigation links: New Item, Build History, Manage Jenkins, and My Views. The main area displays a table of build history for a project named 'web'. The table has columns for status (S), warning (W), name, last success, last failure, and last duration. The first row shows a successful build with a green checkmark, a sun icon, the name 'web', a duration of 4 min 19 sec, and build number #10. Below the table, there's a section for 'Build Queue' showing 'No builds in the queue.' and a 'Build Executor Status' section showing two idle executors. At the bottom right, there's a REST API link and the version number Jenkins 2.452.3.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	web	4 min 19 sec #10	N/A	0.24 sec

Icon: S M L

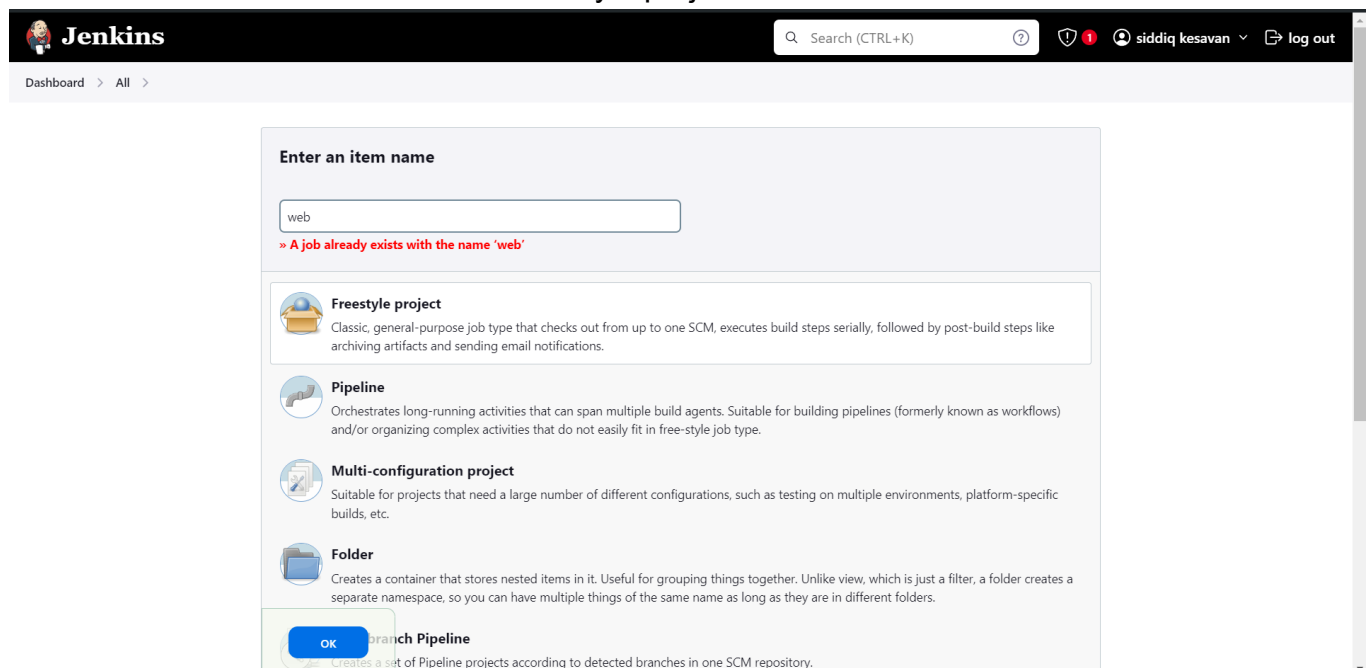
Build Queue: No builds in the queue.

Build Executor Status:

- 1 Idle
- 2 Idle

REST API Jenkins 2.452.3

Enter the item name and select the freestyle project.



The screenshot shows the 'Enter an item name' form in Jenkins. The form has a text input field with the value 'web'. Below the input field, there's a red error message: '» A job already exists with the name 'web''. Below the error message, there's a list of project types: Freestyle project, Pipeline, Multi-configuration project, and Folder. Each project type has a description. At the bottom, there's a 'branch Pipeline' option with an 'OK' button.

Enter an item name

web

» A job already exists with the name 'web'

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- 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.

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

OK

enter the description.

The screenshot shows the Jenkins Configuration page for a job named 'web'. The 'General' tab is selected. The 'Description' field contains the text 'git data pulling and email data sending'. Below the description, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', and 'Execute concurrent builds if necessary'. An 'Advanced' dropdown menu is visible at the bottom of the configuration area. The 'Save' button is highlighted in blue. The top navigation bar shows the user 'siddiq kesavan' and a 'log out' button. The bottom status bar shows the system time as 04:10 PM.

Jenkins

Search (CTRL+K)

Dashboard > web > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

General

Description

git data pulling and email data sending

Plain text: [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterised ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced

Save Apply

☐ None

☒ Git ?

Repositories ?

Repository URL ?

<https://github.com/SIDDIQKESAVANSK/GUVI-Task.git>

Credentials ?

- none -

+ Add

Advanced

Add Repository

enter the github link

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 ?

Schedule ?

* * * * *

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

Would last have run at Tuesday, July 16, 2024 at 10:39:53 AM Coordinated Universal Time; would next run at Tuesday, July 16, 2024 at 10:39:53 AM Coordinated Universal Time.

☐ Ignore post-commit hooks ?

Giving the Poll SCM.

Post-build Actions

≡ Editable Email Notification ?

Allows the user to disable the publisher, while maintaining the settings

☐ Disable Extended Email Publisher ?

Project From

web

Project Recipient List ?

Comma-separated list of email address that should receive notifications for this project.

siddiqesavan@gmail.com

Project Reply-To List ?

Comma-separated list of email address that should be in the Reply-To header for this project.

\$DEFAULT_REPLYTO

Entering the project recipient List.

Enter the email ID

☰ Success ?

Send To

☰ Developers ?

Add ▾

Advanced ▾

Add Trigger ▾

Success Trigger.

GUVI-Task / python1.py

SIDDIQKESAVANSK Update python1.py

8b79844 · now History

Code

Blame

1 lines (1 loc) · 21 Bytes

Code 55% faster with GitHub Copilot

Raw

1 new script file1

Update the data in git.

```

Started by an SCM change
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/web
The recommended git tool is: NONE
No credentials specified
> /usr/bin/git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/web/.git # timeout=10
Fetching changes from the remote Git repository
> /usr/bin/git config remote.origin.url https://github.com/SIDDIQESAVANSK/GUVI-Task.git # timeout=10
Fetching upstream changes from https://github.com/SIDDIQESAVANSK/GUVI-Task.git
> /usr/bin/git --version # timeout=10
> git --version # 'git version 2.40.1'
> /usr/bin/git fetch --tags --force --progress -- https://github.com/SIDDIQESAVANSK/GUVI-Task.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> /usr/bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c0d46ebd59569a26d47596362f4d4de7b79bd739 (refs/remotes/origin/master)
> /usr/bin/git config core.sparsecheckout # timeout=10
> /usr/bin/git checkout -f c0d46ebd59569a26d47596362f4d4de7b79bd739 # timeout=10
Commit message: "Update python1.py"
> /usr/bin/git rev-list --no-walk 97a1dc6bf17aafccad41f7fa5f4bd4677b2af35a # timeout=10
No emails were triggered.
[web] $ /bin/sh -xe /tmp/jenkins7018981115692197574.sh
+ java --version
openjdk 17.0.11 2024-04-16 LTS
OpenJDK Runtime Environment Corretto-17.0.11.9.1 (build 17.0.11+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.11.9.1 (build 17.0.11+9-LTS, mixed mode, sharing)
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: siddiqesavan@gmail.com
Finished: SUCCESS

```

message send to the email :

