

School: CLOUDBOOSTA	Course: DEVOPS
Name: AKOSA SAMUEL ONYEJEKWE	Module: Jenkins
	Class 1: Lab-2_ Jenkins-1
Student No: CBALC0168	Issue Date: May 6, 2026

Tasks:

1. Generate GitHub Personal Access Token (PAT)
2. Configure GitHub token permissions
3. Add GitHub credentials into Jenkins
4. Configure Jenkins system settings
5. Configure GitHub server integration in Jenkins
6. Generate SSH key pair on Jenkins server
7. Add SSH public key to GitHub account
8. Verify GitHub SSH authentication
9. Configure Jenkins security settings
10. Create Jenkins freestyle project for GitHub repository
11. Configure Git repository connection in Jenkins
12. Configure Jenkins credentials for repository access
13. Configure GitHub webhook trigger
14. Configure post-build GitHub status reporting
15. Validate Jenkins and GitHub integration

Goal:

To integrate Jenkins with GitHub using Personal Access Tokens, SSH authentication, GitHub webhooks, and Jenkins credentials management in order to automate CI/CD workflows and enable secure repository integration in a DevOps environment.

Result:

- GitHub Personal Access Token successfully generated
- GitHub token permissions configured successfully
- Jenkins credentials configured successfully
- GitHub server integration configured successfully in Jenkins
- SSH key pair generated successfully on Jenkins server
- GitHub SSH authentication configured successfully
- Jenkins security settings configured successfully
- Jenkins freestyle project created successfully
- GitHub repository connected successfully to Jenkins
- GitHub webhook trigger configured successfully
- Post-build GitHub status integration configured successfully
- Jenkins and GitHub integration validated successfully

Action:

Step 1: Generated GitHub Personal Access Token

- Opened GitHub Developer Settings
- Navigated to:
 - Personal access tokens → Tokens (classic)
- Selected:
 - Generate new token
- Created token for Jenkins integration

Step 2: Configured Token Permissions

- Configured repository access permissions
- Enabled workflow permissions
- Enabled webhook access permissions
- Enabled read and write repository access
- Generated GitHub PAT successfully

Step 3: Verified Generated Token

- Copied generated GitHub token
- Stored token securely for Jenkins authentication
- Verified token creation success message displayed

Step 4: Opened Jenkins System Configuration

- Navigated to:
 Manage Jenkins → System
- Verified Jenkins system configuration settings
- Reviewed Jenkins URL and global properties

Step 5: Configured GitHub Server Integration

- Navigated to GitHub section in Jenkins system configuration
- Added GitHub server configuration
- Configured API URL:

`https://api.github.com`

- Added GitHub credentials for authentication
- Enabled:
 Manage hooks
- Tested GitHub connection successfully

Step 6: Reviewed Jenkins Global Configuration

- Verified Git plugin configuration
- Reviewed pipeline and build settings
- Verified SMTP and email notification sections
- Reviewed Jenkins administrative settings

Step 7: Generated SSH Key Pair

- Opened Jenkins EC2 terminal
- Generated SSH key pair using:

`ssh-keygen`

- Verified generated files:
 - `id_rsa`
 - `id_rsa.pub`
- Verified successful SSH fingerprint generation

Step 8: Added SSH Public Key to GitHub

- Opened GitHub:
Settings → SSH and GPG keys
- Selected:
New SSH key
- Added Jenkins public SSH key
- Saved SSH key successfully

Step 9: Verified SSH Key Registration

- Confirmed Jenkins public key added successfully in GitHub
- Verified SSH authentication key listed under GitHub account

Step 10: Configured Jenkins Credentials

- Opened:
Manage Jenkins → Credentials
- Added GitHub PAT credential
- Added GitHub private SSH key credential
- Verified credentials successfully stored in Jenkins system scope

Step 11: Configured Jenkins Security Settings

- Opened:
Manage Jenkins → Security
- Reviewed authentication and authorization settings
- Verified CSRF protection enabled
- Reviewed Git host verification settings
- Saved Jenkins security configuration

Step 12: Created Jenkins Freestyle Project

- Navigated to:
New Item
- Created freestyle project named:
test-job
- Selected:
Freestyle project
- Clicked OK

Step 13: Configured GitHub Repository Connection

- Opened freestyle project configuration
- Selected:
 - Git under Source Code Management
- Added GitHub repository URL:
 - `git@github.com:samuelakosaonyekwe/three-tier-aws-terraform.git`
- Selected Jenkins SSH credentials for authentication
- Configured branch specifier:
 - `*/main`

Step 14: Configured GitHub Trigger

- Opened Triggers section in Jenkins job
- Enabled:
 - GitHub hook trigger for GITScm polling
- Configured automated trigger integration with GitHub webhooks

Step 15: Configured Post-Build GitHub Actions

- Added post-build action:
 - Set GitHub commit status (universal)
- Configured repository and commit status reporting
- Enabled Jenkins build result reporting back to GitHub

Step 16: Verified Jenkins and GitHub Integration

- Verified successful repository connectivity
- Verified Jenkins authentication with GitHub
- Verified webhook integration configured successfully
- Confirmed Jenkins automation ready for CI/CD workflows

Summary

- GitHub Personal Access Token authentication was successfully configured
- Jenkins and GitHub integration completed successfully
- SSH authentication between Jenkins and GitHub configured successfully
- Jenkins credentials management implemented successfully

- GitHub webhook automation configured successfully
- Jenkins freestyle project connected successfully to GitHub repository
- GitHub commit status reporting configured successfully
- Secure CI/CD integration workflow successfully implemented using Jenkins and GitHub

Lab objective achieved:

Implemented secure Jenkins and GitHub integration using Personal Access Tokens, SSH authentication, Jenkins credentials management, and GitHub webhook automation for CI/CD workflow execution in a DevOps environment.

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

- GitHub Apps
- OAuth Apps
- Personal access tokens
 - Fine-grained tokens
 - Tokens (classic)

Personal access tokens (classic) Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

terraform-git-access — repo, workflow

Last used within the last 1 weeks

Delete

Expires on Tue, Jun 2 2026.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

×

- GitHub Apps

OAuth Apps

Personal access tokens

Time-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Access token for Jenkins

What's this token for?

Expiration

No expiration


 GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- repo

Full control of private repositories

repo:status

Access commit status

repo_deployment

Access deployment status

public_repo

Access public repositories

repo:invite

Access repository invitations

security_events

Read and write security events
- workflow

Update GitHub Action workflows
- write:packages

Upload packages to GitHub Package Registry

read:packages

Download packages from GitHub Package Registry
- delete:packages

Delete packages from GitHub Package Registry
- admin:org

Full control of orgs and teams, read and write org projects

write:org

Read and write org and team membership, read and write org projects

read:org

Read org and team membership, read org projects

manage_runners:org

Manage org runners and runner groups
- admin:public_key

Full control of user public keys

write:public_key

Write user public keys

read:public_key

Read user public keys
- admin:repo_hook

Full control of repository hooks

write:repo_hook

Write repository hooks

read:repo_hook

Read repository hooks
- admin:org_hook

Full control of organization hooks
- gist

Create gists
- notifications

Access notifications
- user

Update ALL user data

read:user

Read ALL user profile data

user:email

Access user email addresses (read-only)

user:follow

Follow and unfollow users
- delete_repo

Delete repositories
- write:discussion

Read and write team discussions

read:discussion

Read team discussions
- admin:enterprise

Full control of enterprises

manage_runners:enterprise

Manage enterprise runners and runner groups

manage_billing:enterprise

Read and write enterprise billing data

read:enterprise

Read enterprise profile data

scim:enterprise

Provisioning of users and groups via SCIM
- audit_log

Full control of audit log

read:audit_log

Read access of audit log

☐ codespace	Full control of codespaces
☐ codespace:secrets	Ability to create, read, update, and delete codespace secrets
☐ copilot	Full control of GitHub Copilot settings and seat assignments
☐ manage_billing:copilot	View and edit Copilot Business seat assignments
☐ write:network_configurations	Write org hosted compute network configurations
☐ read:network_configurations	Read org hosted compute network configurations
☐ project	Full control of projects
☐ read:project	Read access of projects
☐ admin:gpg_key	Full control of public user GPG keys
☐ write:gpg_key	Write public user GPG keys
☐ read:gpg_key	Read public user GPG keys
☐ admin:ssh_signing_key	Full control of public user SSH signing keys
☐ write:ssh_signing_key	Write public user SSH signing keys
☐ read:ssh_signing_key	Read public user SSH signing keys

Generate token

Cancel



On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

- GitHub Apps
- OAuth Apps
- Personal access tokens
 - Time-grained tokens
 - Tokens (classic)

Personal access tokens (classic) Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_mdKcVlFbUjMugIIIsR1G1GykRDmJ8HM3wgjyB

Delete

terraform-git-access — repo, workflow

Last used within the last 2 weeks

Delete

Expires on **Tue, Jun 2 2026**.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

System

Configure global settings and paths.

General

Home directory ?

By default, Jenkins stores all of its data in this directory on the file system

/var/lib/jenkins

System Message

This message will be displayed at the top of the Jenkins main page. This can be useful for posting notifications to your users

Plain text Preview

☒ Built-In Node is now configured in [Nodes](#).

Computer Retention Check Interval ?

60

Quiet period ?

5

SCM checkout retry count

0

☐ Restrict project naming

Jenkins Location

Jenkins URL ?

http://35.177.85.22:8080/

System Admin e-mail address ?

address not configured yet <nobody@nowhere>

Serve resource files from another domain

Resource Root URL ?

Without a resource root URL, resources will be served from the Jenkins URL with Content-Security-Policy set.

Global properties

☐ Disable deferred wipeout on this node ?

☐ Disk Space Monitoring Thresholds

☐ Environment variables

☐ Tool Locations

Metrics

Access keys [?](#)

+ Add new access key

Pipeline Speed / Durability

Default Speed / Durability Level [?](#)

None: use pipeline default (Maximum survivability/durability but slowest)

Writes data with every step, using atomic writes for integrity. Provides maximum ability to retain running pipeline data and resume in the event of a Jenkins failure.

Usage Statistics

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project [?](#)

Folder Health Metrics

List of default health metrics for Folders

+ Add

HTTP Proxy Configuration

Setup [v](#)

Global Build Time Out

☐ Enable Global Time Out

Develocity Build Scans

☐ Detect Build Scan links in all build logs [?](#)

☐ Enable enriched summary [?](#)

Develocity integration

Develocity Access Key credential ID [?](#)

[?](#) The access key must be in the `<server host name>=<access key>` format. For more details please refer to the [documentation](#).

- none -

+ Add

Develocity short-lived access tokens expiry [?](#)

☐ Enable auto-injection

Timestamp

System clock time format [?](#)

'HH:mm:ss'

Elapsed time format ?

''HH:mm:ss.S''

☐ Enabled for all Pipeline builds ?

Fingerprints

☐ Disable Fingerprint Cleanup

Administrative monitors

Administrative monitors are warnings shown to Jenkins administrators about the state of the Jenkins instance. It is generally strongly recommended to keep all administrative monitors enabled, but if you are not interested in specific warnings, uncheck them here to permanently hide them.

Administrative monitors ▾

Global Build Discarders

⌵ Project Build Discarder ?

Build discarders configured for a job are only run after a build finishes. This option runs jobs' configured build discarders periodically, applying configuration changes even when no new builds are run. This option has no effect if there is no build discarder configured for a job.

+ Add

Default notification URL

Notification URL ?

Default ▾

GitHub

GitHub Servers ?

⌵ GitHub Server ?

Name ?

cba-github

API URL ?

https://api.github.com

Credentials ?

Login details to access my GitHub Account ▾

+ Add

Credentials verified for user samuelakosaonyejekwe, rate limit: 5000

Test connection

☒ Manage hooks

Advanced ▾

+ Add GitHub Server

Advanced ▾

✎ Edited

GitHub API usage

GitHub API usage rate limiting strategy ?

Normalize API requests



GitHub Enterprise Servers

+ Add

Global Trusted Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

+ Add

Global Untrusted Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be untrusted, meaning they run with "sandbox" restrictions and cannot use @Grab.

+ Add

Build-timeout Plugin > BuildStep Action

☐ Enable BuildStep Action ?

Git plugin

Global Config user.name Value ?

Global Config user.email Value ?

☐ Create new accounts based on author/committer's email ?

☐ Use existing account with same email if found ?

☐ Show the entire commit summary in changes ?

☐ Hide credential usage in job output ?

☐ Disable performance enhancements ?

☐ Preserve second fetch during checkout ?

☐ Add git tag action to jobs ?

Shell

Shell executable ?

Extended E-mail Notification

SMTP server

SMTP Port

25

Advanced ▾

Default user e-mail suffix ?

Advanced ▾

 Edited

Default Content Type ?

Plain Text (text/plain)



List ID ?

☐

Add 'Precedence: bulk' E-mail Header ?

Default Recipients ?

Reply To List ?

Emergency reroute ?

Allowed Domains ?

Excluded Recipients ?

Default Subject ?

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS!

Maximum Attachment Size ?

-1

Default Content ?

\$PROJECT_NAME - Build # \$BUILD_NUMBER - \$BUILD_STATUS:

Check console output at \$BUILD_URL to view the results.



Default Pre-send Script ?

Default Post-send Script ?

Additional groovy classpath ?

+ Add

- ☐ Enable Throttling ?
- ☐ Enable Debug Mode ?
- ☐ Require Administrator for Template Testing ?
- ☐ Enable watching for jobs ?
- ☐ Allow sending to unregistered users ?

Default Triggers ▾

Content Token Reference ?

E-mail Notification

SMTP server

Default user e-mail suffix ?

Advanced ▾

 Filled

- ☐ Test configuration by sending test e-mail

Save

Apply

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:E7EZRRjP+cwd00wARZIbcVWHlgVUvRGyo5//DaBjwjs ubuntu@ip-172-31-30-37
The key's randomart image is:
+---[RSA 3072]---+
|           +=o +OBX@|
|           .B .oo*=o|
|           + +  *o =|
|           . +o..+ |
|           S  . = . |
|           . . . . . |
|           o +  o. |
|           E+ .   ...|
|           ..      .+|
+-----[SHA256]-----+
ubuntu@ip-172-31-30-37:~$
```

i-0d1acbb6a56eba636 (jenkins_lab_1)

PublicIPs: 35.177.85.22 PrivateIPs: 172.31.30.37

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

samuelakosaonyekwe (samuelakosaonyekwe)

Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

- Access
- Billing and licensing
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys
- Organizations
- Enterprises
- Moderation
- Code, planning, and automation
- Repositories
- Codespaces
- Models
- Packages
- Copilot
- Pages
- Saved replies
- Security
- Code security
- Integrations
- Applications
- Scheduled reminders
- Archives
- Security log
- Sponsorship log
- Developer settings

Add new SSH Key

Title

Jenkins public key

Key type

Authentication Key

Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGCiQAYT6trMHf6C0xZWO+/3d6okw8coTgDVK+5SkLBAnQ7e3/albeyYeB1D6ks5ix8MSxNXtVl3DYsOJF/X64e+OCgpvsMzlb7o1NkCclQm6qOwxEK6WCwjflynzfpv8D3mUKPQL8xZ+AjjLzWrtsu+Du3Oa8Dj+CpJsVgGN5rPIFoQeAixN46xy26q85qn7AD0EPptBS3jkEXhrXH+M8Z9K8ym1N6rT/5vz0Vif5haDcxEr8nHNv2eeVK+TayuDiYYT/hr2Qv2IjdWJefB5pRH+RyzhTk8dxOvoM0Bf9aH2XNPVl5I2UauL+MMtDddj71JXvXR6jdp7dDN8wpXupd2xUOnydAllbOBZNZji/rSGDOqGcK+BE9dF1RhPKJHovOxcpMwtZCwk2jrQh5CqRfVywTf+Sdrt0rBVK0qI3znkg6vfOzncjjiWIH3AZtqEupRR6pYewWi8Kq1AklnGgwmz1Q23uxYmBRmqOpAmHVCrXwUWbtZq82bW2IveJ5qOvc= ubuntu@ip-172-31-30-37

Add SSH key

Settings

Type to search

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

You have successfully added the key 'Jenkins public key'.

samuelakosaonyejekwe (samuelakosaonyejekwe)

Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Models

Packages

Copilot

Pages

Saved replies

Security

Code security

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

Jenkins public key

SHA256:EFEXRRjP1cxd00wARZ1bcVvH1gVUvR6yuc//DuBjwjs

Added on May 2, 2026

Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or [troubleshoot common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

[Learn how to generate a GPG key and add it to your account.](#)

Vigilant mode

Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key. Note that this will include your existing unsigned commits. [Learn about vigilant mode.](#)

© 2025 GitHub, Inc.

Terms

Privacy

Security

Status

Community

Docs

Contact

Manage cookies

Do not share my personal information

Credentials

+ Add Credentials



	github-PAT	System - Global - Login details to access my GitHub Account	
	github-private-key jenkins	System - Global - my-jenkins-private-key	

Stores scoped to Jenkins

	System	Domains: Global
--	--------	-----------------



Security

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

Authentication

☐ Disable "Keep me signed in" ?

Security Realm

Jenkins' own user database

☐ Allow users to sign up ?

Authorization

Logged in users can do anything

☐ Allow anonymous read access ?

Markup Formatter

Markup Formatter ?

Plain text

Shows descriptions mostly as written. HTML unsafe characters like < and & are escaped to their respective character entities, and line breaks are converted to their HTML equivalent.

Agents

TCP port for inbound agents ?

- ☐ Fixed
☐ Random
☒ Disable

CSRF Protection

Crumb Issuer

Default Crumb Issuer

Git plugin notifyCommit access tokens

Current access tokens ?

There are no access tokens yet.

+ Add new access token

Prism syntax highlighting

Permitted Source Code Directories ?

Valid and permitted source code locations on agents (outside the workspace).

+ Add

Git Hooks

☐ Allow on Controller ?

Allow git hooks to run on the Jenkins Controller

☐ Allow on Agents ?

Allow git hooks to run on Jenkins Agents

Hidden security warnings

Security warnings ▾

API Token

☐ Generate a legacy API token for each newly created user (Not recommended) ?

☐ Allow users to manually create a legacy API token (Not recommended) ?

☒ Enable API Token usage statistics ?

Content Security Policy

Content Security Policy enforcement is currently disabled. [Learn more.](#)

Git Host Key Verification Configuration

Host Key Verification Strategy ?

No verification ▾

⚠ This option is generally insecure. Host key will not be verified during SSH connection

Sandbox Configuration

☐ Force the use of the sandbox globally in the system ?

Save

Apply

New Item

Enter an item name

test-job

Select an item type



Pipeline

Build, test, and deploy using pipelines. Supports stages, parallel work, and running on multiple agents.



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.



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.



Multibranch Pipeline

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



Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.



Duplicate an existing item

Create a new item by copying the configuration of an existing one.

OK

main 1 Branch 0 Tags

Go to file

Add file

Code

- Three-tier AWS Terraform: ALBs, ASGs, Aurora,
- infra Three-tier AWS Ter
- user-data Three-tier AWS Ter
- .gitignore Three-tier AWS Ter
- README.md Three-tier AWS Ter

README

Local

Codespaces

Clone

HTTPS SSH GitHub CLI

git@github.com:samuelakosaonyekwe/three-tier

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

About

Infrastructure as Code (Terraform) for provisioning a secure AWS three-tier web application architecture.

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

[Create a new release](#)

Three-Tier AWS Terraform (eu-west-2)

Configure

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

General

Enabled

Description

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

Advanced

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

- ☐ None
- ☒ Git ?

Repositories ?

Repository URL ?

git@github.com:samuelakosaonyekwe/three-tier-aws-terraform.git

Credentials ?

jenkins (my-jenkins-private-key)

+ Add

Advanced

+ Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

+ Add Branch

Repository browser ?

(Auto)

Additional Behaviours

+ Add

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

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

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

+ Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

⋮ Set GitHub commit status (universal) ?

Where:

Commit SHA:

Latest build revision

?

Repositories:

Any defined in job repository

?

What:

Commit context:

From GitHub property with fallback to job name

?

Status result:

One of default messages and statuses

?

Status backref:

Backref to the build

?

?

Advanced ▾

+ Add post-build action

Save

Apply

REST API

Jenkins 2.555.1

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

Edit Delete

Pages

- ❖ Secrets and variables

 Email notifications

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages

Security and quality

Advanced Security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

http://35.177.85.22:8080/github-webhook/

Content type *

application/json

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

☒ Enable SSL verification

☐ Disable (not recommended)

Which events would you like to trigger this webhook?

- ☒ Just the push event.
- ☐ Send me everything.
- ☐ Let me select individual events.

☒ Active

We will deliver event details when this hook is triggered.

Add webhook

On April 24 we'll start using GitHub Copilot interaction data for AI model training unless you opt out. [Review this update](#) and manage your preferences in your [GitHub account settings](#).

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages

Security and quality

Advanced Security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Webhooks / Manage webhook

SettingsRecent Deliveries

✓

afccaa6fc462b11f195a4b783b51ba08c

ping

2026-05-02 16:34:32

Status

Changes

Console Output

Edit Build Information

Delete build '1'

Polling Log

Timings

Git Build Data

Console

Download

Copy

View as plain text

```
Started by GitHub push by samuelakosaonyejekwe
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/test-job
The recommended git tool is: NONE
using credential github-private-key
Cloning the remote Git repository
Cloning repository git@github.com:samuelakosaonyejekwe/three-tier-aws-terraform.git
> git init /var/lib/jenkins/workspace/test-job # timeout=10
Fetching upstream changes from git@github.com:samuelakosaonyejekwe/three-tier-aws-terraform.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
using GIT_SSH to set credentials my-jenkins-private-key
> git fetch --tags --force --progress -- git@github.com:samuelakosaonyejekwe/three-tier-aws-terraform.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url git@github.com:samuelakosaonyejekwe/three-tier-aws-terraform.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 407223445b6ed0dab5b9423f05780c844d6107c0 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 407223445b6ed0dab5b9423f05780c844d6107c0 # timeout=10
Commit message: "Update README header for Three-Tier AWS Terraform"
First time build. Skipping changelog.
[Set GitHub commit status (universal)] SUCCESS on repos [GHRepository@7a1e7cb5[archived=false,commits=
(),compareUsePaginatedCommits=false,description=Infrastructure as Code (Terraform) for provisioning a secure
AWS three-tier web application architecture.,disabled=false,fork=false,homepage=
<null>,isTemplate=false,language=HCL,license=<null>,milestones={},name=three-tier-aws-
terraform,nodeId=R_kgDOReoJBA,parent=<null>,size=7,source=<null>,visibility=public,createdAt=2026-03-
04T22:07:21Z,id=1172972292,nodeId=<null>,updatedAt=2026-05-
02T13:48:04Z,url=https://api.github.com/repos/samuelakosaonyejekwe/three-tier-aws-terraform]] (sha:4072234)
with context:test-job
Setting commit status on GitHub for https://github.com/samuelakosaonyejekwe/three-tier-aws-
terraform/commit/407223445b6ed0dab5b9423f05780c844d6107c0
Finished: SUCCESS
```

- + New Item
- Build History
- Project Relationship
- Check File Fingerprint

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	my-first-freestyle-job	3 hr 24 min #1	N/A	0.14 sec ▶
✓	☀	my-first-pipeline-job	3 hr 5 min #1	N/A	4 sec ▶
✓	☀	my-second-freestyle-job	3 hr 13 min #1	N/A	21 ms ▶
✓	☀	my-second-pipeline-job	2 hr 59 min #2	N/A	1.6 sec ▶
✓	☀	test-job	48 sec #2	N/A	1.7 sec ▶



Icon: S M L

256