Week 6 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

Set Up Jenkins Credentials

Panel 1 — Jenkins Dashboard

Jenkins

Search | Raj Saha | log out

Dashboard

- New Item
- People
- Build History
- **Manage Jenkins**
- My Views
- New View

I will Learn Jenkins and get better paying job

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☼	VeryFirstJenkinsJob	19 days #3	N/A	22 ms	▷

Icon: S M L

Icon legend | Atom feed for all | Atom feed for failures | Atom feed for just latest builds

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

REST API   Jenkins 2.332.3

Panel 2

Dashboard

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Configure global settings and paths.

Configure tools, their locations and automatic installers.

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
⚠ There are updates available

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

Security

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

Manage Credentials
Configure credentials

Configure Credential Providers
Configure the credential providers and types

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

In-process Script Approval
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions. 1 scripts pending approval.

Status Information

System Information
Displays various environmental

System Log
System log captures output from

Load Statistics
Check your resource utilization and see if

Panel 3 — Credentials

Jenkins

Search (⌘+K)

Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name

Stores scoped to Jenkins

P	Store ↓	Domains
System	(global)

Icon: S M L

Panel 4 — Global credentials (unrestricted)

Jenkins

Search (⌘+K)

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)
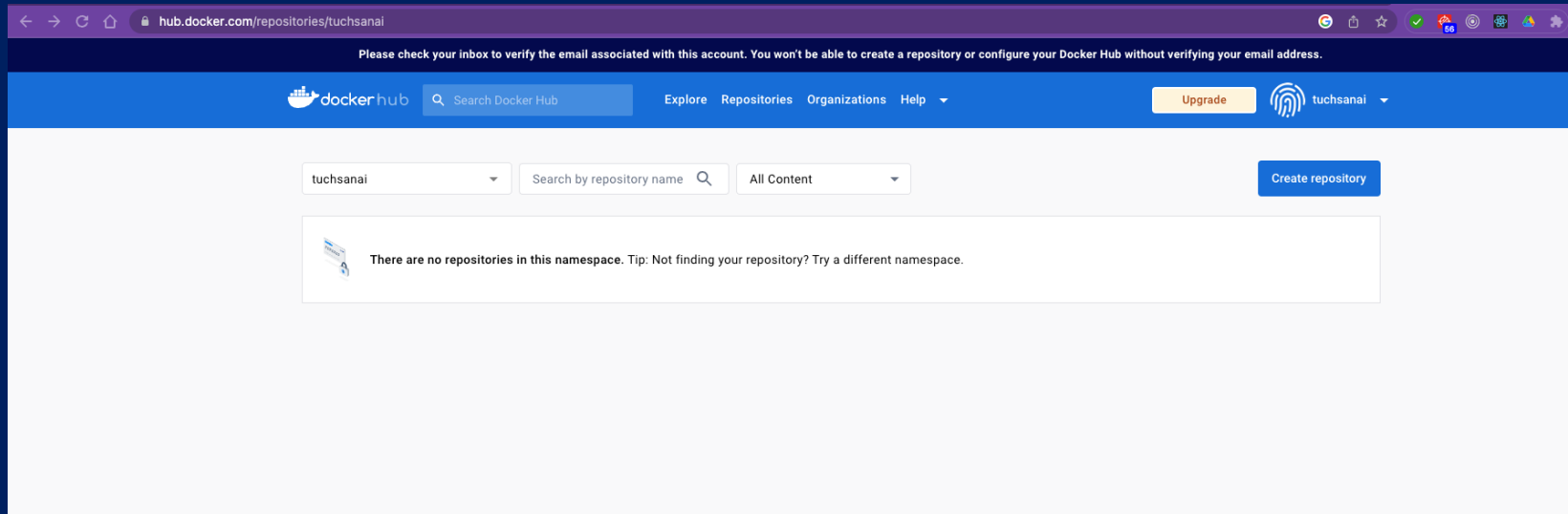
Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
This credential domain is empty. How about adding some credentials?			

Icon: S M L

For Github

For Github

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
github	tuchsanai/****** (my github for test jenkins)	Username with password	my github for test jenkins	🔧

Icon: S M L

For docker Hub

Search (⌘+K)

2 2 Tuchsanai Ploysuwan

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description	
github	tuchsanai/****** (my github for test jenkins)	Username with password	my github for test jenkins	🔧
dockerhub	tuchsanai/****** (my docker hub)	Username with password	my docker hub	🔧

Icon: S   M   L

Intro to Jenkins Plugins

Jenkins cd

Plugins Index

Discover the 1800+ community contributed Jenkins plugins to support building, deploying and automating any project.

| Browse | Find plugins... | 🔍 |

Browse categories

Platforms

User interface

Administration

Build management

Source Code Management

New Plugins

Add Changes to Build Changelog

CodeThreat

DevOps Portal

CatLight Notification

Qualys IaC Security

Pipeline Input Step Notification

BeVigil CI

Matomo analytics

Deploy to webMethods Integration Server

Casdoor Authentication

Recently updated

Venafi CodeSign Protect

Build Failure Analyzer

Add Changes to Build Changelog

Pipeline Aggregator View

Parameterized Scheduler

Nomad

Kubernetes

SnakeYAML API

Pipeline: Basic Steps

Jakarta Mail API

Trending

Configuration as Code

AnsiColor

Job DSL

Pipeline Utility Steps

CloudBees Disk Usage Simple

SSH Agent

Kubernetes

Active Directory

Blue Ocean

Slack Notification

🐙 Improve this page   ⚠ Report a problem

Resources   **Project**   **Community**   **Other**

Plugins Index

Discover the 1800+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse | docker

---

plugins.jenkins.io/ui/search?query=docker

Jenkins | cd

Blog | Documentation | Plugins | Community | Subprojects | Security | About | English | Do

Browse ▾ | docker

Search by 🔍 algolia

x docker

Sort Relevance
- ● Relevance
- ○ Most installed
- ○ Trending
- ○ Title
- ○ Release date

Categories [Show all]

Platforms
- ☐ iOS Development
- ☐ .NET Development
- ☐ Android Development
- ☐ Ruby Development

User interface
- ☐ User Interface
- ☐ List view columns

Administration
- ☐ Agent Management
- ☐ Page Decorators
- ☐ Authentication and User Management
- ☐ Cluster Management
- ☐ Command Line Interface

Build management
- ☐ Build Triggers
- ☐ Build Wrappers
- ☐ Build Notifiers
- ☐ Deployment
- ☐ Build Parameters
- ☐ Other Post-Build Actions
- ☐ Build Tools
- ☐ Build Reports
- ☐ Artifact Uploaders

Source Code Management

1 to 26 of 26

Previous | 1 | Next

Docker
Installs: 40105
Released: 4 weeks ago

adopt-this-plugin, Cloud Provide

This plugin integrates Jenkins with Docker

Kanstantsin Shautsou
magnayn
(4 other contributors) | Do

Docker Commons
Installs: 140949
Released: 6 months ago

Library plugins (for use by other

Provides the common shared functionality for various Docker-related plugins.

Jesse Glick
Kohsuke Kawaguchi
(3 other contributors) | DC

Docker Pipeline
Installs: 128051
Released: 3 months ago

adopt-this-plugin, Deployment,

Build and use Docker containers from pipelines.

Jesse Glick
Oleg Nenashev
(5 other contributors) | DP

Docker API
Installs: 43867
Released: 4 days ago

adopt-this-plugin, Library plugin

This plugin provides docker-java API for other plugins.

Yoann Dubreuil
Eric Citaire | DA

docker-build-step
Installs: 15524
Released: 4 months ago

Build Tools, docker

This plugin allows to add various docker commands to your job as build steps.

vjuranek | Do

CloudBees Docker Build and Publish
Installs: 9934
Released: 6 months ago

Build Tools, docker

This plugin enables building Dockerfile based projects, as well as publishing of the built images/repos to the docker
Carlos Sanchez
Michael Neale
(2 other contributors) | CD

Docker Compose Build Step
Installs: 5378
Released: 5 years ago

This plugin has no labels

Docker Compose plugin for Jenkins

João Galego | DC

Docker Slaves
Installs: 5330
Released: 6 years ago

docker

Uses Docker containers to run Jenkins build agents.

Yoann Dubreuil | DS

CloudBees Docker Custom Build
Installs: 3103
Released: 5 years ago

docker

Run builds inside a docker container, defined by a Docker image or Dockerfile stored in project SCM.

Vincent Latombe
Jon Hermansen | CD

CloudBees Docker Hub/Registry Notification
Installs: 1767
Released: a week ago

docker

Integrates Jenkins with DockerHub and Docker Registry

rsandell
Adrien Lecharpentier
(11 other contributors) | CD

Docker Swarm
Installs: 1398
Released: 2 years ago

docker

Docker swarm cloud plugin.

surya gaddipati
Roman Bäriswyl | DS

CloudBees Docker Traceability
Installs: 1062
Released: 7 years ago

docker

Provides an ability to trace server deployments via fingerprints

Oleg Nenashev
Alex Taylor | CD

CloudShare Docker-Machine
Installs: 162
Released: 6 years ago

Build Wrappers, Cloud Provider

Yet Another Docker
Installs: 1396
Released: 3 years ago

Cloud Providers, Cluster Manag

Panoptica Vulnerability Scanner
Installs: 6
Released: 4 months ago

Amazon ECR
Installs: 8908
Released: 3 weeks ago

aws

Aqua MicroScanner
Installs: 189
Released: 3 years ago

This plugin has no labels

CrowdStrike Security
Installs: 13
Released: 4 months ago

This plugin has no labels

First pipeline using Pipeline script

First pipeline

Pipeline

Definition

Pipeline script

Script ?

```
1  pipeline {
2      agent any
3      stages {
4          stage('first pipeline') {
5              steps {
6
7                  echo 'welcome to first pipeline'
8                  sh 'python3 --version'
9
10             }
11         }
12     }
13  }
```

☑ Use Groovy Sandbox ?

✓ **Console Output**

```
Started by user Tuchsanai Ploysuwan
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/e
[Pipeline] {
[Pipeline] stage
[Pipeline] { (first pipeline)
[Pipeline] echo
welcome to first pipeline
[Pipeline] sh
+ python3 --version
Python 3.7.16
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

https://github.com/Tuchsanai/devtool_week6_jenkins/tree/main/First_pipeline_using_Pipeline_script

```
pipeline {

    agent any      I

    stages {

        stage("build") {

            steps {

            }
        }
    }
}

node {
    // groovy script
}
```

- "pipeline" must be top-level

- "agent" - where to execute

- "stages" - where the "work" happens

    - "stage" and "steps"

https://www.youtube.com/watch?v=7KCS70sCoK0&t=1452s

Declarative Pipeline fundamentals

In Declarative Pipeline syntax, the `pipeline` block defines all the work done throughout your entire Pipeline.

```
Jenkinsfile (Declarative Pipeline)
pipeline {
    agent any  1
    stages {
        stage('Build') {  2
            steps {
                //  3
            }
        }
        stage('Test') {  4
            steps {
                //  5
            }
        }
        stage('Deploy') {  6
            steps {
                //  7
            }
        }
    }
}
```

1. Execute this Pipeline or any of its stages, on any available agent.

2. Defines the "Build" stage.

3. Perform some steps related to the "Build" stage.

4. Defines the "Test" stage.

5. Perform some steps related to the "Test" stage.

6. Defines the "Deploy" stage.

7. Perform some steps related to the "Deploy" stage.

https://www.jenkins.io/doc/book/pipeline/

How To Update the Linux PATH Environment Variable

which python3

```
Last login: Mon Feb 20 06:06:03 2023 from ec2-3-0-5-37.ap-southeast-1.compute.amazonaws.com

      __|  __|_  )
      _|  (     /   Amazon Linux 2 AMI
     ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
30 package(s) needed for security, out of 34 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-17-81 ~]$ which python3
/usr/bin/python3
[ec2-user@ip-172-31-17-81 ~]$ []
```

You can view the PATH variable with the following command:

```
$ echo $PATH                                          Copy
```

An unchanged PATH may look something like this (file paths may differ slightly depending on your system):

```
[ec2-user@ip-172-31-17-81 ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/ec2-user/.local/bin:/home/ec2-user/bin
[ec2-user@ip-172-31-17-81 ~]$ []
```

https://www.digitalocean.com/community/tutorials/how-to-view-and-update-the-linux-path-environment-variable

How To Update the Linux PATH Environment Variable

1. Verify that Python 3 is installed on the system: You can check if Python 3 is installed by running the following command in a terminal:

```css
python3 --version
```

If the output shows the version number of Python 3, it means Python 3 is installed.

2. Add the path to Python 3 to the system path: If Python 3 is installed, but Jenkins is still unable to find it, you can add the path to Python 3 to the system path. To do this, you can add the following line to the end of the `.bashrc` or `.bash_profile` file in your home directory:

```ruby
export PATH=$PATH:/path/to/python3/bin
```

Replace `/path/to/python3/bin` with the actual path to the directory containing the `python3` binary.

3. Restart Jenkins: After adding the path to Python 3 to the system path, you may need to restart Jenkins for the changes to take effect. You can do this by running the following command in a terminal:

```
sudo systemctl restart jenkins
```

This command may vary depending on your system and how Jenkins is installed.

JENKINSFILE using (Github)

https://github.com/Tuchsanai/devtool_week6_jenkins

Configure

- General
- Advanced Project Options
- **Pipeline**

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Tuchsanai/devtool_week6_jenkins.git

1

Credentials ?

tuchsanai/****** (my github)

2

+ Add

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

3

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

jenkinsfile_Pipeline_using_SCM

4

☑ Lightweight checkout ?

Pipeline Syntax

devtool_week6_jenkins / jenkinsfile_Pipeline_using_SCM

main

Tuchsanai Update jenkinsfile_Pipeline_using_SCM

1 contributor

12 lines (11 sloc) | 220 Bytes

```
 1  pipeline {
 2      agent any
 3      stages {
 4          stage('build') {
 5              steps {
 6
 7                  echo ' welcome to github pipeline'
 8                  sh 'python3 --version'
 9              }
10          }
11      }
12  }
```

Give feedback

```
Started by user Tuchsanai Ploysuwan
Obtained jenkinsfile_Pipeline_using_SCM from git https://github.com/Tuchsanai/devtool_week6_jenkins.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/s
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/s/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Tuchsanai/devtool_week6_jenkins.git # timeout=10
Fetching upstream changes from https://github.com/Tuchsanai/devtool_week6_jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.1'
using GIT_ASKPASS to set credentials my github
 > git fetch --tags --force --progress -- https://github.com/Tuchsanai/devtool_week6_jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 1d977a9cc984a899946be4a83188d65aebffb790 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 1d977a9cc984a899946be4a83188d65aebffb790 # timeout=10
Commit message: "Update .gitattributes"
 > git rev-list --no-walk 1d977a9cc984a899946be4a83188d65aebffb790 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] echo
 welcome to github pipeline
[Pipeline] sh
+ python3 --version
Python 3.7.16
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Pipeline with post actions

Go to file | Add file ▾ | ⋯

Tuchsanai ygg   5d283f6 · 1 minute ago   ⏱ History

..

📄 README.md   ygg   1 minute ago
📄 jenkinsfile_Post_Actions   edit1   8 minutes ago

```
pipeline {
    agent any
    stages {
        stage('build') {
            steps {
                sh 'python --version'
            }
        }
    }
    post {
        always {
            echo 'Always'
        }
        success {
            echo 'Only on SUCCESS'
        }
        failure {
            echo 'Only on Failure'
        }
        unstable {
            echo 'Only if run is unstable'
        }
        changed {
            echo 'Only if status changed from Success to Failure or vice versa w.r.t. last run.'
        }
    }
}
```
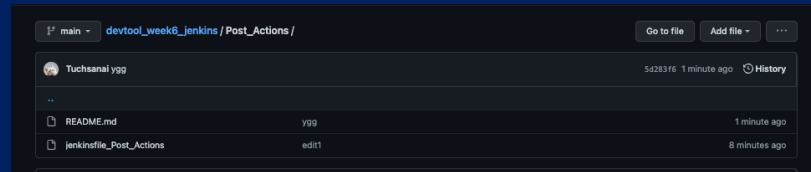
Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Tuchsanai/devtool_week6_jenkins.git

Credentials ?

tuchsanai/****** (my github)

＋ Add

Advanced...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

./Post_Actions/jenkinsfile_Post_Actions

☑ Lightweight checkout ?

Pipeline Syntax

Save | Apply

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Tuchsanai/devtool_week6_jenkins.git

Credentials ?

tuchsanai/****** (my github)

[+ Add]

[Advanced...]

[Add Repository]

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

[Add Branch]

Repository browser ?

(Auto)

Additional Behaviours

[Add ▾]

Script Path ?

./Post_Actions/Jenkinsfile_Post_Actions

☑ Lightweight checkout ?

Pipeline Syntax

[Save] [Apply]

Console Output

Started by user Tuchsanai Ploysuwan
Obtained ./Post_Actions/jenkinsfile_Post_Actions from git https://github.com/Tuchsanai/devtool_week6_jenkins.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/fff
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github
Cloning the remote Git repository
Cloning repository https://github.com/Tuchsanai/devtool_week6_jenkins.git
 > git init /var/lib/jenkins/workspace/fff # timeout=10
Fetching upstream changes from https://github.com/Tuchsanai/devtool_week6_jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.1'
using GIT_ASKPASS to set credentials my github
 > git fetch --tags --force --progress -- https://github.com/Tuchsanai/devtool_week6_jenkins.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Tuchsanai/devtool_week6_jenkins.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 456b89843707bf7e7e773c992b97977f8bec6b59 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 456b89843707bf7e7e773c992b97977f8bec6b59 # timeout=10
Commit message: "Merge branch 'main' of https://github.com/Tuchsanai/devtool_week6_jenkins"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (build)
[Pipeline] sh
+ python --version
Python 2.7.18
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Always
[Pipeline] echo
Only if status changed from Success to Failure or vice versa w.r.t. last run.
[Pipeline] echo
Only on SUCCESS
[Pipeline] }
[Pipeline] // stage
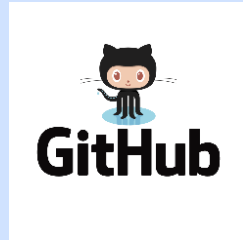[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
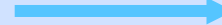Finished: SUCCESS
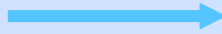
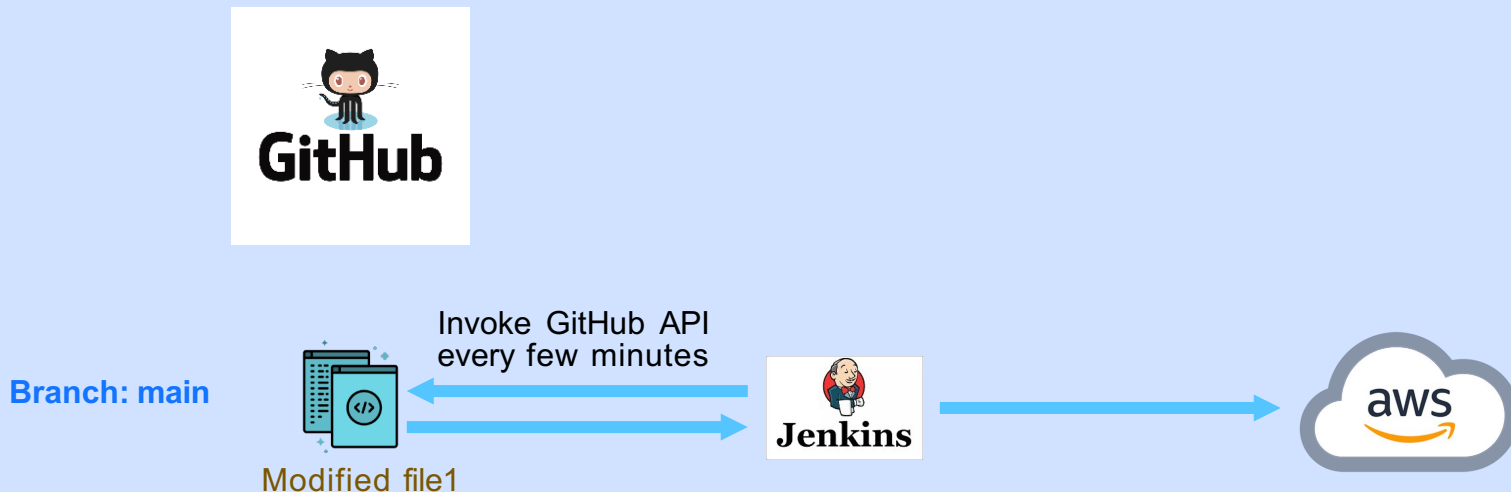# Pipeline with Multiple Stages

Pipeline with Github Webhook

Branch: main

Modified file1
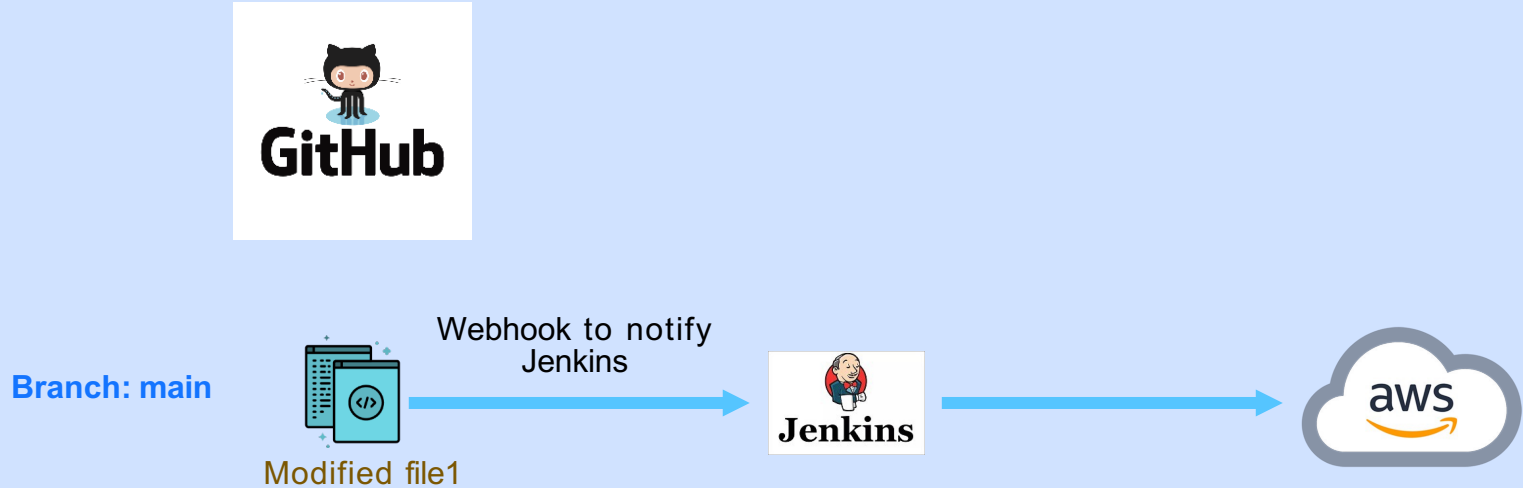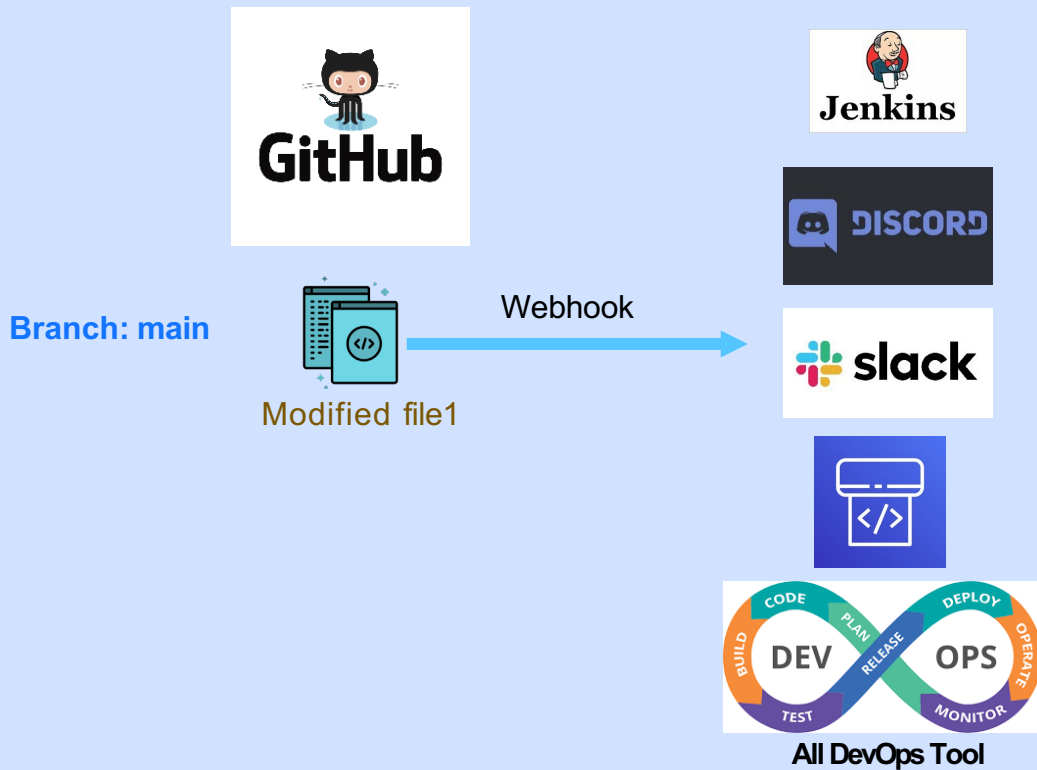
- Most of the times API will return stagnant data
- GitHub server will be bombarded
- Apps will exceed API limits

Webhook

Branch: main

Modified file1

Webhook to notify Jenkins

- GitHub will do a POST call to your app if repo changes
- Lightweight
- Realtime

Implementation

Branch: main

Modified file1

Webhook

All DevOps Tool

Python

Jenkinsfile (Declarative Pipeline)

```
/* Requires the Docker Pipeline plugin */
pipeline {
    agent { docker { image 'python:3.10.7-alpine' } }
    stages {
        stage('build') {
            steps {
                sh 'python --version'
            }
        }
    }
}
```

Node.js / JavaScript

Jenkinsfile (Declarative Pipeline)

```
/* Requires the Docker Pipeline plugin */
pipeline {
    agent { docker { image 'node:16.17.1-alpine' } }
    stages {
        stage('build') {
            steps {
                sh 'node --version'
            }
        }
    }
}
```

Go

Jenkinsfile (Declarative Pipeline)

```
/* Requires the Docker Pipeline plugin */
pipeline {
    agent { docker { image 'golang:1.19.1-alpine' } }
    stages {
        stage('build') {
            steps {
                sh 'go version'
            }
        }
    }
}
```

https://www.jenkins.io/doc/pipeline/tour/hello-world/

https://medium.com/@knoldus/setup-jenkins-in-docker-container-eb420f3a994

https://medium.com/tektutor/ci-cd-with-maven-github-docker-jenkins-aca28c252fec

Jenkins & GitHub Integration with Selenium Python

https://www.youtube.com/watch?v=h2Abne2jIjM&t=248s