

Week 13 : SOFTWARE DEVELOPMENT TOOLS AND ENVIRONMENTS

LAB 01 : Run Jenkinfile using SCM From Github

In Jenkins, "SCM" stands for Source Code Management. SCM in Jenkins refers to the process of managing and integrating source code from version control systems into Jenkins projects.

Tuchsanai / DevTools

<> Code Issues Pull requests Actions Projects Security Insights Settings

main DevTools / 03_Jenkins / week13 / 01_Run_Jenkinsfile_From_Github_WithSCM /

Tuchsanai d c89be10 · now History

Name	Last commit message	Last commit date
JENKINFILE	d	now
readme.md	ww	1 minute ago
status.py	ww	1 minute ago

readme.md

Tuchsanai / DevTools

<> Code Issues Pull requests Actions Projects Security Insights Settings

dev DevTools / 03_Jenkins / week13 / 01_Run_Jenkinsfile_From_Github / status.py

Tuchsanai s

Code Blame 14 lines (12 loc) · 286 Bytes

```
1 import os
2
3 def main():
4     # Displaying status
5     print("System Status:")
6     print("-----")
7     print("Operating System:", os.name)
8     print("\n")
9     print("Environment Variables:", os.environ)
10    print("\n")
11    print("finished.")
12
13 if __name__ == "__main__":
14    main()
```


main DevTools / 03_Jenkins / week13 / 01_Run_Jenkinsfile_From_Github / JENKINFILE

Tuchsanai aa 65dd21b · 6 minutes ago History

29 lines (27 loc) · 814 Bytes

Code Blame Raw Download Edit

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('Check Python Installation') {
6             steps {
7                 script {
8                     // Check if Python is installed
9                     def pythonInstalled = sh(script: "which python3", returnStatus: true) == 0
10                    if (!pythonInstalled) {
11                        // Install Python if not installed
12                        sh 'sudo apt update'
13                        sh 'sudo apt install -y python3'
14                    }
15                }
16            }
17        }
18
19        stage('Run Python Script') {
20            steps {
21                script {
22                    // Run Python script with only os library
23                    sh 'python3 status.py'
24                    sh 'ls -l'
25                }
26            }
27        }
28    }
29 }
```

**Jenkins**

Search (⌘+K)

Dashboard >

+ New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project


Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

**Jenkins**


Search (⌘+K)

Dashboard > All >


Enter an item name

jenkins_file


» Required field

**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.

**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.

OK

Configure

General

Advanced Project Options

Pipeline

Build Triggers

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

Advanced Project Options

Advanced ▾

Pipeline

Definition

- ✓ Pipeline script
- Pipeline script from SCM

Script ?

1

try sample Pipeline... ▾

4

main ▾ DevTools / 03_Jenkins / week13
/ 01_Run_Jenkinsfile_From_Github_WithSCM
/ JENKINFILE

Tuchsanai d c89be10 · 1 minute ago History

30 lines (27 loc) · 894 Bytes

Code Blame Raw Download Edit

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Check Python Installation') {
6       steps {
7         script {
8           // Check if Python is installed
9           def pythonInstalled = sh(script: "which python3", returnStatus: true) == 0
10          if (!pythonInstalled) {
11            // Install Python if not installed
12            sh 'sudo apt update'
13            sh 'sudo apt install -y python3'
14          }
15        }
16      }
17    }
18
19    stage('Run Python Script') {
20      steps {
21        script {
22          // Run Python script with only os library
23          sh 'ls -l'
24          sh 'python3 ../03_Jenkins/week13/01_Run_Jenkinsfile_From_Github_WithSCM/status.py'
25        }
26      }
27    }
28  }
29 }
30 }
```

Configure

General

Advanced Project Options

Pipeline

☐ Throttle builds ?

Build Triggers

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

Advanced Project Options

Advanced ▾

Pipeline

Definition

Pipeline script from SCM

SCM ?

✓ None

Git

Script Path ?

./03_Jenkins/week13/01_Run_Jenkinsfile_From_Github_WithSCM/JENKINFILE

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Save

Apply

Copy path of
Jenkinsfile

Repositories ?

Repository URL ?

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

Credentials ?

tuchsanai/***** (github account)

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/dev

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

Script Path ?

./03_Jenkins/week13/01_Run_Jenkinsfile_From_Github_WithSCM/JENKINFILE

Save

Apply

If private repo, you need credentials


Selected dev branch




10

- Last build (#3), 2 min 15 sec ago
- Last stable build (#3), 2 min 15 sec ago
- Last successful build (#3), 2 min 15 sec ago
- Last completed build (#3), 2 min 15 sec ago


Declarative: Checkout SCM	Check Python Installation	Run Python Script
1s	394ms	982ms
1s	394ms	982ms


LAB 02 : Run Jenkinfile without SCM From Github


 **Jenkins**


Search (⌘+K)    Tuchsana Ploysuwan


Dashboard >


 New Item


 People


 Build History

 Project Relationship


 Check File Fingerprint

 Manage Jenkins

 My Views

Build Queue 

No builds in the queue.

Build Executor Status 


1 Idle

2 Idle


Welcome to Jenkins!


This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.


Start building your software project

Create a job 

Set up a distributed build

Set up an agent 


Configure a cloud 

Learn more about distributed builds 


Enter an item name

No_SCM


» Required field

 **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.

 **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.

OK

Configure

General

Advanced Project Options

Pipeline

☐ Throttle builds ?

Build Triggers

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

Advanced Project Options

Advanced ▾

Pipeline

Definition

Pipeline script ▾

```
1- pipeline {
2-   agent any
3-
4-   stages {
5-
6-       stage('copy repository') {
7-           steps {
8-               // Copy the repository
9-           }
10-       }
11-
12-       stage('Check Python Installation') {
13-           steps {
14-               script {
15-               }
16-           }
17-       }
18-   }
19- }
```

try sample Pipeline...

☒ Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

Copy code

3

status.py

www

readme.md

Jenkins file with GitHub

- detail in slides

```
pipeline {
    agent any

    stages {
        stage('copy repository') {
            steps {
                // Copy the repository
                checkout scmGit(branches: [[name: '*/dev']], extensions: [], userRemoteConfigs: [[credentialsId: 'g
            ]
        }

        stage('Check Python Installation') {
            steps {
                script {
                    // Check if Python is installed
                    def pythonInstalled = sh(script: "which python3", returnStatus: true) == 0
                    if (!pythonInstalled) {
                        // Install Python if not installed
                        sh 'sudo apt update'
                        sh 'sudo apt install -y python3'
                    }
                }
            }
        }

        stage('Run Python Script') {
            steps {
                script {
                    // Run Python script with only as library
                    sh 'pwd'
                    sh 'python3 ../03_Jenkins/week13/02_Run_Jenkinsfile_From_Github_withoutSCM/status.py'

                    sh 'ls -l'
                }
            }
        }
    }
}
```

Dashboard > No_SCM > Pipeline Syntax

↑ Back

Snippet Generator

Declarative Directive Generator

Declarative Online Documentation

Steps Reference

Global Variables Reference

Online Documentation

Examples Reference

IntelliJ IDEA GDSL

Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step

archiveArtifacts: Archive the artifacts

Files to archive ?

Advanced ▾

Generate Pipeline Script

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

↑ Back

Snippet Generator

Declarative Directive Generator

Declarative Online Documentation

Steps Reference

Global Variables Reference

Online Documentation

Examples Reference

IntelliJ IDEA GDSL

Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

Steps

Sample Step

- ✓ archiveArtifacts: Archive the artifacts
- bat: Windows Batch Script
- build: Build a job
- catchError: Catch error and set build result to failure
- checkout: Check out from version control
- cleanWs: Delete workspace when build is done
- configFileProvider: Provide Configuration files
- deleteDir: Recursively delete the current directory from the workspace
- dir: Change current directory
- echo: Print Message
- emailxext: Extended Email
- emailxextrecipients: Extended Email Recipients
- error: Error signal
- fileExists: Verify if file exists in workspace
- findBuildScans: Find published build scans
- fingerprint: Record fingerprints of files to track usage
- git: Git
- input: Wait for interactive input
- isUnix: Checks if running on a Unix-like node
- junit: Archive JUnit-formatted test results
- library: Load a library on the fly

checkout ?

SCM

Git

Repositories ?

Repository URL ?

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

Credentials ?

tuchsanai/*****

+ Add -

Advanced -

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/dev

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add -

☒ Include in polling? ?

☒ Include in changelog? ?

Generate Pipeline Script

Push button

checkout ?

SCM

Git

Repositories ?

Repository URL ?

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

Credentials ?

tuchsanai/*****

+ Add -

Advanced -

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/dev

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add -

☒ Include in polling? ?

☒ Include in changelog? ?

Generate Pipeline Script

```
checkout scmGit[branches: [[name: '**/dev']], extensions: [], userRemoteConfigs: [[credentialsId: 'github', url: 'https://github.com/Tuchsanai/DevTools.git']]]
```

Configure

General

Advanced Project Options

Pipeline

☐ Preserve stashes from completed builds ?

☐ This project is parameterized ?

☐ Throttle builds ?

Build Triggers

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

☐ Quiet period ?

☐ Trigger builds remotely (e.g., from scripts) ?

Advanced Project Options

Advanced ▾

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5
6     stage('copy repository') {
7       steps {
8
9         // Copy the repository
10        checkout scmGit(branches: [[name: '*/dev']], extensions: [], userRemoteConfigs: [])
11      }
12    }
13  }
14 }
15
```

Paste

☒ Use Groovy Sandbox ?

Pipeline Syntax

Save

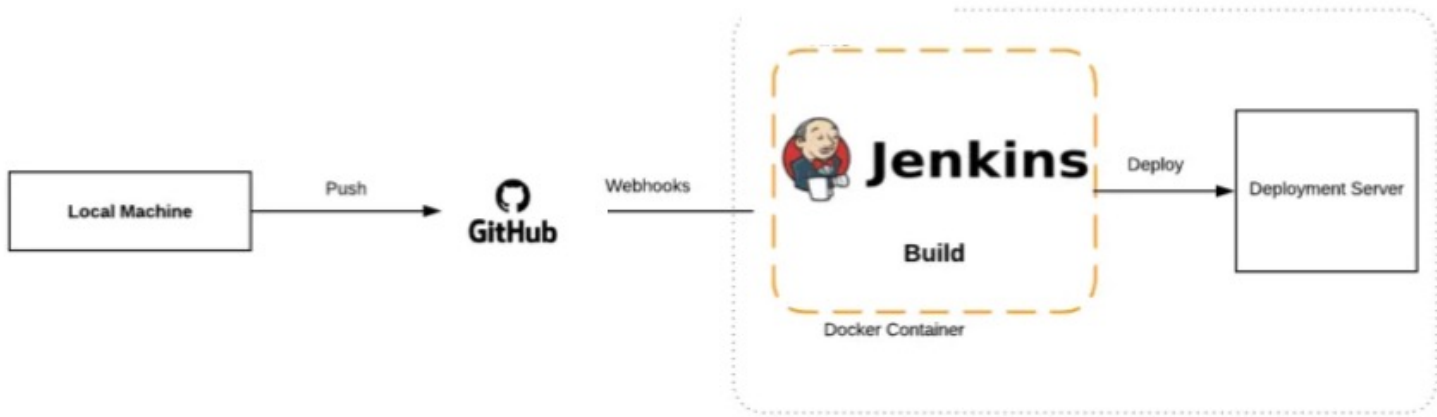
Apply

Console Output

```
Started by user tp
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/No_SCM
[Pipeline] {
[Pipeline] stage
[Pipeline] { (copy repository)
[Pipeline] checkout
The recommended git tool is: NONE
using credential github
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/No_SCM/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Tuchansai/DevTools.git # timeout=10
Fetching upstream changes from https://github.com/Tuchansai/DevTools.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/Tuchansai/DevTools.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/dev^{commit} # timeout=10
Checking out Revision 773695d843e3e2f982ebecde9a88461af605 (refs/remotes/origin/dev)
> git config core.sparsecheckout # timeout=10
Commit message: "delete files"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Check Python Installation)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ which python3
/var/lib/python3
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run Python Script)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ pwd
/var/lib/jenkins/workspace/No_SCM
[Pipeline] }
+ python3 ./03_Jenkins/week13/02_Run_Jenkinsfile_From_Github_withoutSCM/status.py
System Status:
Operating System: posix

Environment Variables: environ({'JENKINS_HOME': '/var/lib/jenkins', 'USER': 'jenkins', 'CI': 'true', 'RUN_CHANGES_DISPLAY_URL': 'http://175.41.181.239:8080/', 'MODE_LABELS': 'built-in', 'HUDSON_URL': 'http://175.41.181.239:8080/', 'HOME': '/var/lib/jenkins', 'BUILD_URL': 'http://175.41.181.239:8080/job/No_SCM/2/', 'HUDSON_COOKIE': 'a7658642-e8f8-4763-9815-314baf6139f', 'JENKINS_SERVER_COOKIE': 'durable-586d74ee4c761eccf6b9c3ed728f70e911ca0476f3899a9c02a53e39619', 'NOTIFY_SOCKET': '/run/systemd/notify', 'SYSTEMD_EXEC_PID': '7568', 'WORKSPACE': '/var/lib/jenkins/workspace/No_SCM', 'LOGNAME': 'jenkins', 'NODE_NAME': 'built-in', 'JOURNAL_STREAM': '8:44869', 'RUN_ARTIFACTS_DISPLAY_URL': 'http://175.41.181.239:8080/job/No_SCM/2/display/redirect?page=artifacts', 'STAGE_NAME': 'Run Python Script', 'EXECUTOR_NUMBER': '1', 'RUN_TESTS_DISPLAY_URL': 'http://175.41.181.239:8080/job/No_SCM/2/display/redirect?page=tests', 'BUILD_DISPLAY_NAME': '92', 'HUDSON_HOME': '/var/lib/jenkins', 'JOB_BASE_NAME': 'No_SCM', 'PATH': '/usr/local/sbin:/usr/local/bin:/usr/sbin:/bin:/sbin:/snap/bin', 'INVOCATION_ID': 'aca2f3592ce43aaa80d8f676ec0bae', 'BUILD_ID': '2', 'BUILD_TAG': 'jenkins-No_SCM-2', 'LANG': 'C.UTF-8', 'JENKINS_URL': 'http://175.41.181.239:8080/', 'JOB_URL': 'http://175.41.181.239:8080/job/No_SCM/', 'BUILD_NUMBER': '2', 'JENKINS_NODE_COOKIE': '36180eds-4913-4788-990d-25497ee17a95', 'SHELL': '/bin/bash', 'RUN_DISPLAY_URL': 'http://175.41.181.239:8080/job/No_SCM/2/display/redirect', 'HUDSON_SERVER_COOKIE': '4d0b2b048492119', 'JOB_DISPLAY_URL': 'http://175.41.181.239:8080/job/No_SCM/display/redirect', 'JOB_NAME': 'No_SCM', 'PWD': '/var/lib/jenkins/workspace/No_SCM', 'WORKSPACE_TMP': '/var/lib/jenkins/workspace/No_SCM/tmp'})

Finished.
[Pipeline] sh
+ ls -l
total 28
drwxr-xr-x 6 jenkins jenkins 4096 Feb 28 08:53 00_GIT
drwxr-xr-x 3 jenkins jenkins 4096 Feb 28 08:53 01_Google Cloud
drwxr-xr-x 6 jenkins jenkins 4096 Feb 28 08:53 02_Docker
drwxr-xr-x 4 jenkins jenkins 4096 Feb 28 08:53 03_Jenkins
drwxr-xr-x 4 jenkins jenkins 4096 Feb 28 08:53 04_kubernetes
drwxr-xr-x 2 jenkins jenkins 4096 Feb 28 08:53 Mini_Project
-rw-r--r-- 1 jenkins jenkins 1695 Feb 28 08:53 README.md
```

Delivery (CD).

Tuchsanai / DevTools

<> Code Issues Pull requests Actions Projects Security Insights Settings

Tuchsanai Public

Unpin Unwatch 1 Fork 4 Star 20

main

Go to file + <> Code

Tuchsanai q36a75ee · 6 hours ago🕒 629 Commits

00_GIT	ss	2 weeks ago
01_Google Cloud	ss	2 weeks ago
02_Docker	n	5 days ago
03_Jenkins	q	6 hours ago
04_kubernetes	y	5 days ago
Mini_Project	nn	yesterday
.gitignore	Update .gitignore	6 months ago
README.md	d	3 months ago

Readme

Activity

20 stars

1 watching

4 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Readme

Tuchsanai / DevTools

<> Code Issues Pull requests Actions Projects Security Insights Settings

General

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Repository name

DevToolsRename

Template repository

Require contributors to sign off on web-based commits

Default branch

main

Social preview

Tuchsanai / DevTools

🔍

+

🔄

🔧

📧

<> Code

🕒 Issues

🔗 Pull requests

🕒 Actions

📁 Projects

🛡 Security

📊 Insights

⚙ Settings

⚙ General

Webhooks

Add webhook

Access

👤 Collaborators

🗨 Moderation options

Code and automation

🔑 Branches

📁 Tags

📄 Rules

🕒 Actions

🔗 Webhooks

📁 Environments

📄 Codespaces

📁 Pages

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](#).

