

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

Tasks:

1. Launch AWS EC2 instance for Jenkins server
2. Configure EC2 networking and security groups
3. Configure Jenkins installation user data script
4. Deploy Ubuntu EC2 instance
5. Connect to Jenkins server instance
6. Access Jenkins web interface
7. Unlock Jenkins using administrator password
8. Install Jenkins suggested plugins
9. Create Jenkins administrator user
10. Configure Jenkins instance URL
11. Verify Jenkins installation completion
12. Create Jenkins freestyle job
13. Configure build parameters
14. Configure Jenkins build steps
15. Execute Jenkins job with parameters
16. Validate Jenkins console output
17. Duplicate Jenkins job configuration

Goal:

To deploy and configure a Jenkins automation server on AWS EC2 infrastructure, create freestyle Jenkins jobs with parameterized builds, execute automated build processes, and validate successful CI/CD workflow execution in a DevOps environment.

Result:

- AWS EC2 instance successfully launched for Jenkins server
- Jenkins server successfully installed on Ubuntu EC2 instance
- Security groups configured successfully for SSH, HTTP, and Jenkins access
- Jenkins web interface accessed successfully
- Jenkins unlocked using administrator password
- Suggested Jenkins plugins installed successfully
- Jenkins administrator account created successfully
- Jenkins instance configured successfully

- Jenkins freestyle job created successfully
- Parameterized build configuration implemented successfully
- Build steps configured successfully
- Jenkins build executed successfully
- Console output validated successfully
- Jenkins job duplication implemented successfully

Action:

Step 1: Launched AWS EC2 Instance

- Opened AWS EC2 Console
- Selected Ubuntu Server AMI
- Chose instance type: t3.small
- Configured key pair for SSH access
- Verified launch configuration summary

Step 2: Configured Security Groups

- Configured inbound security group rules:
 - SSH → Port 22
 - HTTP → Port 80
 - Custom TCP → Port 8080 (Jenkins)
 - Allowed traffic from Anywhere (0.0.0.0/0)
 - Verified VPC and subnet configuration

Step 3: Configured Jenkins User Data Script

- Added Jenkins installation commands in EC2 user data section
- Configured package installation and Jenkins startup
- Example commands included:

```
sudo apt update
sudo apt install fontconfig openjdk-21-jre -y
sudo apt install jenkins -y
sudo systemctl enable jenkins
sudo systemctl start jenkins
```

Step 4: Verified EC2 Instance Deployment

- Confirmed EC2 instance launched successfully
- Verified instance state: Running
- Verified public IPv4 address assigned
- Verified instance name: jenkins_lab_1

Step 5: Connected to Jenkins Server

- Opened EC2 instance details
- Selected Connect option
- Verified public IP connectivity
- Confirmed successful EC2 instance access

Step 6: Accessed Jenkins Web Interface

- Opened browser using: http://:8080
- Accessed Jenkins initial setup page
- Verified Unlock Jenkins screen displayed successfully

Step 7: Unlocked Jenkins

- Retrieved administrator password from:
`/var/lib/jenkins/secrets/initialAdminPassword`
- Entered administrator password into Jenkins setup page
- Clicked Continue
- Verified successful authentication

Step 8: Installed Jenkins Plugins

- Selected: Install suggested plugins
- Jenkins automatically downloaded and installed plugins
- Verified plugin installation completed successfully

Step 9: Created Jenkins Administrator User

- Configured Jenkins administrator account
- Entered: · Username
- Password

- Full name
- Email address
- Saved administrator configuration successfully

Step 10: Configured Jenkins Instance URL

- Verified Jenkins URL configuration:

`http://35.177.85.22:8080/`

- Saved Jenkins instance configuration
- Completed Jenkins setup process

Step 11: Verified Jenkins Installation

- Jenkins setup completed successfully
- Verified message: “Jenkins is ready!”
- Clicked: Start using Jenkins

Step 12: Created Jenkins Freestyle Job

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

Step 13: Configured Build Parameters

- Enabled: This project is parameterized
- Added parameters:
 - FirstName
 - LastName
 - Gender
 - Configured descriptions for each parameter

Step 14: Configured Build Steps

- Added Execute Shell build step
- Added shell commands using environment variables
- Example:

```
echo "My name is $FirstName $LastName"  
echo "I am a $Gender"
```

Step 15: Saved Jenkins Job Configuration

- Reviewed freestyle job configuration
- Verified parameters and build steps
- Clicked Save
- Confirmed successful job creation

Step 16: Executed Parameterized Build

- Selected: Build with Parameters
- Entered: · FirstName → Akosa
- LastName → Onyejekwe
- Gender → Male
- Triggered Jenkins build execution

Step 17: Verified Console Output

- Opened Console Output section
- Verified successful build execution
- Verified output:

```
My name is Akosa Onyejekwe  
I am a Male  
Finished: SUCCESS
```

- Confirmed successful freestyle job execution

Step 18: Duplicated Jenkins Job

- Navigated to: New Item
- Selected: Copy existing item
- Copied: my-first-freestyle-job
- Created duplicated job: my-second-freestyle-job

Summary

- AWS EC2 infrastructure was successfully used to host Jenkins server
- Jenkins was installed and configured successfully on Ubuntu server

- Jenkins administrator setup completed successfully
- Jenkins freestyle project was created successfully
- Parameterized builds were configured and executed successfully
- Jenkins build automation workflow validated successfully
- Console output verified successful execution of shell commands
- Jenkins job duplication functionality validated successfully
- CI/CD automation concepts were successfully implemented using Jenkins

Lab objective achieved:

Implemented Jenkins server deployment and freestyle job automation on AWS EC2 infrastructure with successful parameterized build execution and CI/CD workflow validation in a DevOps environment.

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Take a walkthrough

Do not show me this message again

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

jenkins_lab_1

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Search our full catalog including 1000s of application and OS images

Recents

My AMIs

Quick Start



Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0951a43515d1f167b (64-bit (x86)) / ami-0c1de60a540962231 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (...)

AMI ID

ami-0951a43515d1f167b

Publish Date

2026-04-24

Username Info

ubuntu

Verified provider

Instance type Info | Get advice

Instance type

t3.small

Family: t3 2 vCPU 2 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.042 USD per Hour
On-Demand RHEL base pricing: 0.0524 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0271 USD per Hour
On-Demand Linux base pricing: 0.0236 USD per Hour
On-Demand SUSE base pricing: 0.0546 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required

cba_keypair

Create new key pair

Network settings Info

Summary

Number of Instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd64...read more
ami-0951a43515d1f167b

Virtual server type (instance type)

t3.small

Cancel

Launch instance

Preview code

VPC - required [Info](#)

vpc-0dd7c25d7f1fb2785 (default) [↻](#)

Subnet [Info](#)

No preference [↕](#)

[↻](#) Create new subnet [↗](#)

Availability Zone [Info](#)

No preference [↕](#)

[↻](#) Enable additional zones [↗](#)

Auto-assign public IP [Info](#)

Enable [↕](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&:[]!\$*

Description - required [Info](#)

launch-wizard-1 created 2026-05-02T09:35:55.490Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

ssh [↕](#)

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Anywhere [↕](#)

Source [Info](#)

[Q](#) Add CIDR, prefix list or security

0.0.0.0/0 [✕](#)

Description - optional [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

HTTP [↕](#)

Protocol [Info](#)

TCP

Port range [Info](#)

80

Source type [Info](#)

Anywhere [↕](#)

Source [Info](#)

[Q](#) Add CIDR, prefix list or security

0.0.0.0/0 [✕](#)

Description - optional [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 8080, 0.0.0.0/0)

[Remove](#)

Type [Info](#)

Custom TCP [↕](#)

Protocol [Info](#)

TCP

Port range [Info](#)

8080

Source type [Info](#)

Anywhere [↕](#)

Source [Info](#)

[Q](#) Add CIDR, prefix list or security

0.0.0.0/0 [✕](#)

Description - optional [Info](#)

e.g. SSH for admin desktop

[⚠](#) Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[✕](#)

[Add security group rule](#)

▼ Configure storage [Info](#)

[Advanced](#)

1x GiB Root volume, 3000 IOPS, Not encrypted

[Add new volume](#)

The selected AMI contains instance store volumes, however the instance does not allow any instance store volumes. None of the instance store volumes from the AMI will be accessible from the instance.

🔄 Click refresh to view backup information



The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

File systems

☐ S3 Files - new

☐ EFS

☐ FSx

▼ Advanced details [Info](#)

Domain join directory [Info](#)

Select ▼

🔄 [Create new directory](#)

IAM instance profile [Info](#)

Select ▼

🔄 [Create new IAM profile](#)

Hostname type [Info](#)

IP name ▼

DNS Hostname [Info](#)

☒ Enable IP name IPv4 (A record) DNS requests

☒ Enable resource-based IPv4 (A record) DNS requests

☐ Enable resource-based IPv6 (AAAA record) DNS requests

Instance auto-recovery [Info](#)

Select ▼

Shutdown behavior [Info](#)

Stop ▼

Stop - Hibernate behavior [Info](#)

Select ▼

Termination protection [Info](#)

Select ▼

Stop protection [Info](#)

Select ▼

Detailed CloudWatch monitoring [Info](#)

Select ▼

Credit specification [Info](#)

Unlimited ▼

[Additional charges apply](#)

Placement group [Info](#)

Select ▼

🔄 [Create new placement group](#)

EBS-optimized instance [Info](#)

Enable ▼

Instance bandwidth configuration [Info](#)

Select ▼

Purchasing option [Info](#)

☒ None

☐ Capacity Blocks

Launch instances for your active capacity blocks

☐ Interruptible Capacity Reservations

Launch instances for your Interruptible Capacity Reservations

☐ Spot instances

Request Spot Instances at the Spot price, capped at the On-Demand price

Capacity reservation [Info](#)

Select ▼

Tenancy [Info](#)

Select ▼

RAM disk ID [Info](#)

Select ▼

Kernel ID [Info](#)

Select ▼

Nitro Enclave [Info](#)

Select ▼

License configurations [Info](#)

Select ▼

**CPU options - update** [Info](#)

Configure CPUs for your instance to optimize performance and save on licensing costs.

- ☒ Use default CPU options
- ☐ Specify CPU options

Default active vCPUs

2

Total vCPUs

2

Nested virtualization [Info](#)

Select ▼

Nested virtualization is not supported with the selected instance type.

Metadata accessible [Info](#)

Enabled ▼

Metadata IPv6 endpoint [Info](#)

Select ▼

Metadata version [Info](#)

V2 only (token required) ▼

⚠ For V2 requests, you must include a session token in all instance metadata requests. Applications or agents that use V1 for instance metadata access will break.

Metadata response hop limit [Info](#)

2

Allow tags in metadata [Info](#)

Select ▼

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

⬆ Choose file

```
# Install Java 21 (required for Jenkins)
sudo apt install fontconfig openjdk-21-jre -y

# Install Jenkins
sudo apt-get install jenkins -y

# Enable and start Jenkins service
sudo systemctl enable jenkins
sudo systemctl start jenkins

# Check status
sudo systemctl status jenkins

# Get initial admin password
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

☐ User data has already been base64 encoded

- EC2 > Instances
- EC2
- Dashboard
- AWS Global View 
- Events
- Instances
- Instances
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations
- Capacity Manager [New](#)
- Images
- AMIs
- AMI Catalog
- Elastic Block Store
- Volumes
- Snapshots
- Lifecycle Manager
- Network & Security
- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces
- Load Balancing
- Load Balancers
- Target Groups
- Trust Stores
- Auto Scaling
- Auto Scaling Groups
- Settings

Instances (1/1) [Info](#)

Last updated 11 minutes ago 

[Connect](#)

[Instance state](#) 

[Actions](#) 

[Launch instances](#) 

Saved filter sets

Choose filter set 

< 1 > 

☒	Name 	Instance ID	Instance state 	Instance type 	Status check
☒	jenkins_lab_1	i-0d1acbb6a56eba636	 Running  	t3.small	 3/3 checks

i-0d1acbb6a56eba636 (jenkins_lab_1)

Instance ID

 i-0d1acbb6a56eba636

Public IPv4 address

 35.177.85.22 | [open address](#) 

Private IPv4 addresses

 172.31.30.37

IPv6 address

-

Instance state

 Running

Public DNS



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Create First Admin User

Username

phoenixsamuel

Password

Confirm password

Full name

Akosa Samuel Onyejekwe

E-mail address

samuelsonyejekwe852@gmail.com

Instance Configuration

Jenkins URL:

http://35.177.85.22:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

New Item

Enter an item name

my-first-freestyle-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.

OK

Configure

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

General

Enabled ☒

Description

This job prints out my name

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☒ This project is parameterized ?

String Parameter ?

Name ?

FirstName

Default Value ?

Description ?

Please enter your name

Plain text [Preview](#)

☐ Trim the string ?

String Parameter ?

Name ?

LastName

Default Value ?

Description ?

Please enter your last name

Plain text [Preview](#)

☐ Trim the string ?

Choice Parameter ?

Name ?

Gender

Choices ?

None
Male
Female
Prefer Not to Say

Description ?

Please select your gender

Plain text [Preview](#)

+ Add Parameter

☐ 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 ?

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.

⌵ Execute shell ?

Command

See [the list of available environment variables](#)

```
#!/bin/bash
echo "My name is ${FirstName} ${LastName}"
echo "I am a ${Gender}"
```

Advanced ▾

+ Add build step

Post-build Actions

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

+ Add post-build action

Save

Apply

 Status

 Changes

 Workspace

 Build with Parameters

 Configure

 Delete Project

 Rename

my-first-freestyle-job

This job prints out my name

Permalinks

 Edit description

Builds >

...

No builds

- Status
- Changes
- Workspace
- Build with Parameters
- Configure
- Delete Project
- Rename

Builds >

No builds

Project my-first-freestyle-job

This build requires parameters:

FirstName

Please enter your name:

Akosa

LastName

Please enter your last name:

Onyejekwe

Gender

Please select your gender:

Male

Build

Cancel



Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Parameters

Timings

Console

Download

Copy

View as plain text

```
Started by user Akosa Samuel Onyejekwe
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-first-freestyle-job
[my-first-freestyle-job] $ /bin/bash /tmp/jenkins4381917822790791735.sh
My name is Akosa Onyejekwe
I am a Male
Finished: SUCCESS
```

New Item

Enter an item name

my-second-freestyle-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.

my-first-freestyle-job

OK

Configure

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

General

Enabled ☒

Description

This `jenkins` job gives you the option to copy some details from the job into a file.

Plain text [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☒ This project is parameterized ?

String Parameter ?

Name ?

FirstName

Default Value ?

Description ?

Please enter your name

Plain text [Preview](#)

☐ Trim the string ?

String Parameter ?

Name ?

LastName

Default Value ?

Description ?

Please enter your last name

Plain text [Preview](#)

☐ Trim the string ?

Choice Parameter ?

Name ?

Gender

Choices ?

None
Male
Female
Prefer Not to Say

Description ?

Please select your gender

Plain text [Preview](#)

Boolean Parameter ?

Name ?

ResultsInFile

☐ Set by Default ?

Description ?

Would you like this job to copy its content to a file?

Plain text [Preview](#)

+ Add Parameter

☐ 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 ?

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

Build steps

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

Execute shell ?

Command

See the list of available environment variables

```
then
  echo " My name is ${FirstName} ${LastName} and I am a ${Gender} " >> name.txt
else
  echo " No information found " >> name01.txt
fi
```

Advanced ▾

+ Add build step

Post-build Actions

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

+ Add post-build action

Save

Apply

 Status

 Changes

 Workspace

 Build with Parameters

 Configure

 Delete Project

 Rename

my-second-freestyle-job

 Edit description

This Jenkins job gives you the option to copy some details from the job into a file.

Permalinks

Builds >

...

No builds

- Status
- Changes
- Workspace
- Build with Parameters
- Configure
- Delete Project
- Rename

Builds >

No builds

Project my-second-freestyle-job

This build requires parameters:

FirstName

Please enter your name:

Akosa

LastName

Please enter your last name:

Onyejekwe

Gender

Please select your gender:

Male

☒ ResultsInFile

Would you like this job to copy its content to a file?

Build

Cancel



Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Parameters

Timings

Console

Download

Copy

View as plain text

```
Started by user Akosa Samuel Onyejekwe
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/my-second-freestyle-job
[my-second-freestyle-job] $ /bin/bash /tmp/jenkins301200811305977053.sh
Finished: SUCCESS
```

Status

Changes

Workspace

Wipe Out Current Workspace

Build with Parameters

Configure

Delete Project

Rename

Workspace of my-second-freestyle-job on Built-In Node

my-second-freestyle-job /

 name.txt May 2, 2026, 10:43:04 AM 45 B 

 (all files in zip)

Builds > ***

/

Today

 #1 10:43 AM ▾

← → ↻ ⚠ Not secure 35.177.85.22:8080/job/my-second-freestyle-job/ws/name.txt



My name is Akosa Onyejekwe and I am a Male

New Item

Enter an item name

my-first-pipeline-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

Configure

⚙️ General

🕒 Triggers

🔧 Pipeline

🔑 Advanced

General

Enabled 

Description

My first jenkins job to output "Hello World."

Plain text [Preview](#)

- ☐ Discard old builds ?
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☐ GitHub project
- ☐ Pipeline speed/durability override ?
- ☐ Preserve stashes from completed builds ?
- ☐ This project is parameterized ?
- ☐ Throttle builds ?

Triggers

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

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

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script 

Script ?

```
1 ~ pipeline {  
2   agent any  
3 ~   stages {  
4 ~     stage('Hello Jenkins') {  
5 ~       steps {  
6         sh 'echo "hello world" '  
7       }  
8     }  
9   }  
10 }
```

[try sample Pipeline...](#) ☒ Use Groovy Sandbox ?[Pipeline Syntax](#)

Advanced

Advanced 

Save

Apply

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax

my-first-pipeline-job

Edit description

My first jenkins job to output "Hello World."

Permalinks

Builds	>	...
today		
	#1	10:50 AM

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Console

Download

Copy

View as plain text

```
Started by user Akosa Samuel Onyejekwe
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/my-first-pipeline-job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello Jenkins)
[Pipeline] sh
+ echo hello world
hello world
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

New Item

Enter an item name

my-second-pipeline-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

Configure

General

Triggers

Pipeline

Advanced

General

Enabled 

Description

Building a pipeline version of a freestyle job

Plain text [Preview](#)

- ☐ Discard old builds ?
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☐ GitHub project
- ☐ Pipeline speed/durability override ?
- ☐ Preserve stashes from completed builds ?
- ☐ This project is parameterized ?
- ☐ Throttle builds ?

Triggers

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

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

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script 

Script ?

```
3 ~ parameters {
4     string(name: 'FirstName', defaultValue: '', description: 'Please enter your first name')
5     choice(name: 'Gender', choices: ['male', 'female'], description: 'Please select your gender')
6     string(name: 'Age', defaultValue: '', description: 'Please enter your age')
7 }
8 ~ stages {
9     stage('Print Info') {
10         steps {
11             sh 'echo "My name is ${FirstName}"'
12             sh 'echo "Gender: ${Gender}"'
13             sh 'echo "Age: ${Age}"'
14         }
15     }
16 }
17 ~
```

[try sample Pipeline...](#)☒ Use Groovy Sandbox ?[Pipeline Syntax](#)

Advanced

Advanced 

Save

Apply

Status

Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

my-second-pipeline-job

Building a pipeline version of a freestyle job

Permalinks

Edit description

Builds >

...

today



#1 10:55 AM



Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Timings

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Console

Download

Copy

View as plain text

```
Started by user Akosa Samuel Onyejekwe
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/my-second-pipeline-job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Print Info)
[Pipeline] sh
+ echo My name is
My name is
[Pipeline] sh
+ echo Gender:
Gender:
[Pipeline] sh
+ echo Age:
Age:
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- Status
- Changes
- Build with Parameters
- Configure
- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax

Builds

Filter

Today

✓ #1 10:55 AM

Pipeline my-second-pipeline-job

This build requires parameters:

FirstName

Please enter your first name

Akosa

Gender

Please select your gender

male

Age

Please enter your age

42

Build Cancel

Status

Changes

Console Output

Edit Build Information

Delete build '#2'

Parameters

Timings

Pipeline Overview

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

Console

Download

Copy

View as plain text

```
Started by user Akosa Samuel Onyejekwe
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/my-second-pipeline-job
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Print Info)
[Pipeline] sh
+ echo My name is Akosa
My name is Akosa
[Pipeline] sh
+ echo Gender: male
Gender: male
[Pipeline] sh
+ echo Age: 42
Age: 42
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```