

School: CLOUDBOOSTA	Course: DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Assignment-1_ Jenkins-2
Student No: CBALC0168	Issue Date: May 14, 2026

Tasks:

1. Create AWS VPC infrastructure
2. Configure public and private subnets
3. Configure Internet Gateway and NAT Gateway
4. Launch Jenkins EC2 server instance
5. Configure EC2 networking and security groups
6. Configure Jenkins server user data installation
7. Verify EC2 instance deployment
8. Access Jenkins web interface
9. Unlock Jenkins administrator account
10. Install Jenkins suggested plugins
11. Create Jenkins administrator user
12. Configure Jenkins instance settings
13. Install AWS Jenkins plugins
14. Configure AWS credentials in Jenkins
15. Install Java and Maven on Jenkins server
16. Configure Jenkins Global Tool Configuration
17. Configure Java environment variables
18. Create GitHub repository fork
19. Configure GitHub webhook integration
20. Configure Jenkins GitHub credentials
21. Configure Jenkins pipeline integration
22. Verify Jenkins CI/CD connectivity

Goal:

To deploy and configure a Jenkins Continuous Integration and Continuous Deployment (CI/CD) environment on AWS Cloud infrastructure, integrate Jenkins with GitHub repositories, configure AWS authentication and developer tools, and establish automated DevOps workflow integration within a cloud-native environment.

Result:

- AWS VPC infrastructure successfully created
- Public and private subnets configured successfully
- Internet Gateway and NAT Gateway configured successfully
- Jenkins EC2 server instance deployed successfully
- Jenkins security groups configured successfully
- Jenkins server installed successfully using EC2 user data
- Jenkins web console accessed successfully
- Jenkins administrator account unlocked successfully
- Jenkins plugins installed successfully
- Jenkins administrator user configured successfully
- Jenkins instance configuration completed successfully
- AWS Jenkins plugins installed successfully
- AWS credentials configured successfully in Jenkins
- Java JDK and Maven installed successfully
- Jenkins global tools configured successfully
- Java environment variables configured successfully
- GitHub repository fork created successfully
- GitHub webhook integration configured successfully
- Jenkins GitHub credentials configured successfully
- Jenkins pipeline integration configured successfully
- Jenkins CI/CD workflow validated successfully

Action:

Step 1: Created AWS VPC Infrastructure

- Opened: AWS VPC Console → Create VPC
- Selected: VPC and more
- Configured:
 - Name tag auto-generation: jenkins_sam
 - Configured IPv4 CIDR block: 10.0.0.0/16
 - Selected: 2 Availability Zones
- Configured:

- Public subnets: 2
 - Private subnets: 2
- Configured:
 - NAT gateways: 1 per AZ
- Enabled:
 - DNS hostnames
 - DNS resolution
- Clicked: Create VPC

Step 2: Verified VPC Workflow Creation

- Opened: Create VPC workflow
- Verified successful creation of:
 - VPC
 - Public subnets
 - Private subnets
 - Internet Gateway
 - Route tables
 - NAT gateways
- Verified all workflow tasks completed successfully

Step 3: Launched Jenkins EC2 Instance

- Opened: EC2 Console → Launch Instance
- Configured:
 - Instance name: jenkins_server
 - Selected AMI: Ubuntu Server 24.04 LTS
 - Selected instance type: t3.medium

- Configured key pair for SSH access
 - Selected created VPC: jenkins_sam_vpc
 - Selected public subnet for Jenkins deployment

Step 4: Configured Jenkins Security Group

- Created security group for Jenkins server
- Configured inbound rules:
 - SSH: Port 22
 - Custom TCP: Port 8080 for Jenkins web interface
- Allowed traffic source: Anywhere (0.0.0.0/0)
- Verified security group attached successfully to EC2 instance

Step 5: Configured Jenkins User Data Installation

- Added EC2 user data script for Jenkins setup
- Configured installation of:
 - Java
 - Jenkins
 - AWS CLI
 - Terraform
- Verified automatic package installation executed successfully during instance launch

Step 6: Verified Jenkins EC2 Deployment

- Opened: EC2 Instances
- Verified instance: jenkins_server
- Confirmed:

- Instance state: Running
- Public IP address assigned successfully
- Public DNS generated successfully

Step 7: Accessed Jenkins Web Interface

- Opened browser using: http://:8080
- Verified Jenkins startup page loaded successfully
- Confirmed Jenkins initial setup page displayed successfully

Step 8: Unlocked Jenkins Administrator Account

- Retrieved Jenkins administrator password from:
/var/lib/jenkins/secrets/initialAdminPassword
- Entered administrator password into Jenkins unlock page
- Clicked: Continue
- Verified Jenkins administrator account unlocked successfully

Step 9: Installed Jenkins Suggested Plugins

- Selected: Install suggested plugins
- Verified installation of:
 - Pipeline plugins
 - Git plugins
 - AWS plugins
 - Build tools plugins
 - Security plugins
- Confirmed plugin installation completed successfully

Step 10: Created Jenkins Administrator User

- Configured:
 - Username: phoenixsamuel
 - Configured administrator password
- Configured:
 - Full name: Akosa Samuel Onyejekwe
 - Configured email address successfully
- Clicked: Save and Continue

Step 11: Configured Jenkins Instance Settings

- Configured Jenkins URL using EC2 public IP address
- Verified Jenkins instance configuration completed successfully
- Clicked: Save and Finish
- Confirmed Jenkins setup completed successfully

Step 12: Installed AWS Jenkins Plugins

- Opened: Manage Jenkins → Plugins
- Installed:
 - Pipeline AWS Steps
 - AWS SDK plugins
 - S3 Publisher plugin
 - AWS Credentials plugin
- Verified successful plugin installation and Jenkins restart

Step 13: Configured AWS Credentials in Jenkins

- Opened: Manage Jenkins → AWS

- Selected AWS region: Europe (London)
- Added AWS account credentials successfully
- Verified AWS credentials connected successfully to Jenkins

Step 14: Verified Jenkins Credentials Store

- Opened: Manage Jenkins → Credentials
- Verified AWS credentials stored successfully
- Confirmed credentials accessible globally within Jenkins environment

Step 15: Installed Java and Maven

- Connected to Jenkins EC2 server terminal
- Executed: `mvn --version`
- Verified:
- Apache Maven installed successfully
- Java 21 OpenJDK configured successfully
- Confirmed Maven and Java runtime operational successfully

Step 16: Configured Jenkins Global Tool Configuration

- Opened: Manage Jenkins → Tools
- Configured:
 - Git installation
 - Maven installation
 - JDK installation
- Configured Maven path: `/usr/share/maven`
- Configured Java home: `/usr/lib/jvm/java-21-openjdk-amd64`
- Saved Jenkins tool configuration successfully

Step 17: Configured Java Environment Variables

- Executed: `sudo nano /etc/environment`
- Configured: `JAVA_HOME=/usr/lib/jvm/java-21-openjdk-amd64`
- Executed: `source /etc/environment`
- Verified: `echo $JAVA_HOME`
- Confirmed Java environment variables configured successfully

Step 18: Created GitHub Repository Fork

- Opened GitHub repository
- Selected: Fork
- Configured repository name: `ci-demo`
- Created GitHub fork successfully

Step 19: Configured GitHub Webhook Integration

- Opened: GitHub Repository → Settings → Webhooks
- Selected: Add webhook
- Configured:
 - Payload URL: `http://:8080/github-webhook/`
 - Selected content type: `application/json`
- Enabled: Just the push event
- Added webhook successfully

Step 20: Configured Jenkins GitHub Credentials

- Opened: Manage Jenkins → Credentials
- Added: Username with password credentials

- Configured:
 - GitHub username
 - GitHub password/token
- Credential ID: github-access
- Verified GitHub credentials created successfully

Step 21: Configured Jenkins Pipeline Integration

- Opened GitHub repository Jenkinsfile
- Configured CI/CD pipeline stages for:
 - Build
 - Test
 - Deploy
- Configured AWS S3 artifact deployment integration
- Committed Jenkinsfile changes successfully

Step 22: Verified Jenkins CI/CD Connectivity

- Verified GitHub webhook communication with Jenkins
- Confirmed Jenkins pipeline triggered automatically from GitHub commits
- Verified successful CI/CD workflow execution
- Confirmed DevOps automation pipeline functioning successfully

Summary

- AWS VPC infrastructure deployed successfully for Jenkins environment
- Jenkins EC2 server configured successfully on AWS Cloud
- Jenkins plugins and developer tools installed successfully
- AWS credentials and GitHub integration configured successfully
- GitHub webhook integration validated successfully

- Java and Maven configured successfully for build automation
- Jenkins CI/CD pipeline workflow configured successfully
- Automated DevOps pipeline deployment and integration validated successfully

Lab objective achieved:

Implemented Jenkins CI/CD pipeline deployment, AWS cloud infrastructure configuration, GitHub integration, Jenkins automation setup, developer tools configuration, and automated DevOps workflow orchestration successfully within AWS cloud infrastructure.

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

- VPC and more

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

jenkins_sam

Determine the starting IP and the size of your VPC using CIDR notation.

65,536 IPs

CIDR block size must be between /16 and /28.

- No IPv6 CIDR block

☐ Amazon-provided IPv6 CIDR block

Default

Number of Availability Zones (AZs) [Info](#)

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

3

- **Customize AZs**

Number of public subnets [Info](#)

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

2Number of private subnets [Info](#)

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

2

- Customize subnets CIDR blocks

NAT gateways (\$) - *updated* [Info](#)

NAT gateway allows private resources to access the internet from any availability zone within a VPC, providing a single managed internet exit point for the entire region. Additional charges apply.

Regional - new

Zonal

- Introducing regional NAT gateway
AWS now offers a multi-AZ NAT Gateway, eliminating the need for separate NAT Gateways across availability zones.

VPC endpoints [Info](#)

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None53 GatewayVPC [Show details](#)

Your AWS virtual network

jenkins_sam-vpc

Subnets (4)

Subnets within this VPC

eu-west-2a

- jenkins_sam-subnet-put

jenkins_sam-subnet-priv

eu-west-2b

- jenkins_sam-subnet-put

jenkins_sam-subnet-priv

DNS options [Info](#)

- ☒ Enable DNS hostnames
- ☒ Enable DNS resolution

► **Additional tags**

[Cancel](#)

 [Preview code](#)

[Create VPC](#)



CloudShell

[Feedback](#)



Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

Create VPC workflow

Success

Details

- ✓ Create VPC: [vpc-0178cf9c752761693](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-0178cf9c752761693](#)
- ✓ Create S3 endpoint: [vpce-0ec9f6368d310011a](#)
- ✓ Create subnet: [subnet-002e25abc3e0b8145](#)
- ✓ Create subnet: [subnet-0befe04258795b57](#)
- ✓ Create subnet: [subnet-05bbe40c58673e8d1](#)
- ✓ Create subnet: [subnet-021058a5bf7acf67e](#)
- ✓ Create internet gateway: [igw-00bb8025cecc1c289](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-0ddfd7a639025ff38](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: [rtb-05c9d9e9ef27b962a](#)
- ✓ Associate route table
- ✓ Create route table: [rtb-02eb6cbe59303b0b5](#)
- ✓ Associate route table
- ✓ Verifying route table creation
- ✓ Associate S3 endpoint with private subnet route tables: [vpce-0ec9f6368d310011a](#)

Related VPC resources and services

Discover additional resources you can launch within your VPC to build your network architecture. From security components to connectivity services, explore the building blocks available for your infrastructure.



Client VPN

Networking

AWS Client VPN is a managed client-based VPN service that enables you to securely access AWS resources and resources in your on-premises network.

Create 

Resources

Let us know what you would like to create in your VPC

e.g., Instances, Network ACL, Transit

100 characters available. Do not disclose any personal, commercially sensitive, or confidential information.

Submit feedback

View VPC

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_server

[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](#).

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.medium

Family: t3 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0472 USD per Hour
On-Demand SUSE base pricing: 0.1035 USD per Hour
On-Demand Windows base pricing: 0.0656 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0507 USD per Hour
On-Demand RHEL base pricing: 0.076 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.medium

[Cancel](#)

[Launch instance](#)

[Preview code](#)

▼ 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

```
keyring.gpg] \  
  https://apt.releases.hashicorp.com $(lsb_release -cs) main" \  
 | sudo tee /etc/apt/sources.list.d/hashicorp.list > /dev/null  
  
sudo apt-get update -y  
sudo apt-get install -y terraform  
  
# _____  
# AWS CLI v2  
# _____  
echo ">>> Installing AWS CLI..."  
cd /tmp  
curl -fsSL "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"  
unzip -o awscliv2.zip
```

☐ User data has already been base64 encoded

Instances (1) Info

Last updated less than a minute ago

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Saved filter sets

Choose filter set ▾

Find Instance by attribute or tag (case-sensitive)

< 1 >

⚙️

☐	Name	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status
☐	jenkins_server	i-076382a5c61ef5e51	Running	t3.medium	⌚ Initializing	View alarms

Select an instance

```
ubuntu@ip-10-0-15-245:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
a2d22a9605f64e2785ebdbc3b3584c58  
ubuntu@ip-10-0-15-245:~$
```

 i-076382a5c61ef5e51 (jenkins_server)

Instance ID

 i-076382a5c61ef5e51

IPv6 address

—

Public IPv4 address

 3.9.117.92 | [open address](#) 

Instance state

 **Running**

Private IPv4 addresses

 10.0.15.245

Public DNS

 ec2-3-9-117-92.eu-west-

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

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.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup	✓ Build Timeout	✓ Credentials Binding
✓ Timestampers	✓ Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline Graph View
Git	SSH Build Agents	Matrix Authorization Strategy	LDAP
Email Extension	Mailer	Dark Theme	

Jenkins 2.555.1

- Commons lang3 vs. x Jenkins API
- Testcontainers API
- OWASP Markup Formatter
 - JSR API
 - JSON Path API
 - Struts
 - Pipeline: Step API
 - Commons-Lexi API
 - Token Macro
- Build Timeout
 - hudsoncastle API
 - Credentials
 - Plain Credentials
 - Variant
 - SSH Credentials
- Credentials Binding
 - SQL API
 - Pipeline: API
- Timestampers
 - Caffeine API
 - Script Security
 - Pipeline: Supporting APIs
 - Plugin Utilities API
 - Font Awesome API
 - Bootstrap 5 API
 - Jackson Annotations 2 API
 - SnakeYAML Engine API
 - Woodstox Core API
 - Jackson 3 API
 - Query3 API
 - FCache API
 - Price API
 - Display URI API
 - Checks API
 - Unit
 - Matrix Project
 - Resource Disposer
- Workspace Cleanup
- Ant
 - Okhttp
 - JavaBeans Activation Framework (JAF) API
 - JAXB

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://3.9.117.92:8080/

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](#)

Plugins

Search: s3

Install [Dropdown] Refresh

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Install	Name	Released	Health
☒	AWS Global Configuration 159.v882a_ecd03fe6 Library plugins (for use by other plugins) A Jenkins plugin to configure AWS related settings	1 mo 23 days ago	100
☒	Pipeline: AWS Steps 1.45 pipeline aws This plugins adds Jenkins pipeline steps to interact with the AWS API.	2 yr 1 mo ago	87
☒	S3 publisher 519.v0228461f2b_e0 Artifact Uploaders aws This is a plugin to upload files to Amazon S3 buckets.	1 mo 28 days ago	94
☐	Artifact Manager on S3 973.v6a_51253e896b_ aws A Jenkins plugin to keep artifacts and Pipeline stashes in Amazon S3.	1 mo 15 days ago	97
☐	Amazon S3 Bucket Credentials 1.0.0 A plugin that searches an S3 Yum repository and presents the results as a persistent dropdown parameter.	7 yr 6 mo ago	61
☐	S3 package parameter 1.7 A plugin that searches an S3 Yum repository and presents the results as a persistent dropdown parameter.	10 yr ago	25
☐	S3 Explorer 1.0.8 The Jenkins plugin for working with an Amazon S3 bucket. Warning: This plugin version may not be safe to use. Please review the following security notices: AWS secrets displayed without masking	6 yr 3 mo ago	59
☐	S3 Yum repo parameter 1.5 A plugin that searches an S3 Yum repository and presents the results as a persistent dropdown parameter.	11 yr ago	25
☐	Amazon Web Services SDK 2 :: S3 2.42.33-70.vd69c0763fa_60 S3 module for the AWS SDK for Java.	25 days ago	100
☐	Jobcacher S3 Storage Extension 62.v260cb_8db_5b_fa_ Extension of jobcacher plugin that add S3 for item storage.	6 mo 9 days ago	97

Plugins

- 📄 Updates
- 📦 Available plugins
- 🔧 Installed plugins
- ⚙️ Advanced settings
- ☰ Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

commons-lang3 v3.x Jenkins API	✓ Success
Icons API	✓ Success
Folders	✓ Success
OWASP Markup Formatter	✓ Success
ASM API	✓ Success
JSON Path API	✓ Success
Structs	✓ Success
Pipeline: Step API	✓ Success
commons-text API	✓ Success
Token Macro	✓ Success
Build Timeout	✓ Success
bouncycastle API	✓ Success
Credentials	✓ Success
Plain Credentials	✓ Success
Variant	✓ Success
SSH Credentials	✓ Success
Credentials Binding	✓ Success
SCM API	✓ Success
Pipeline: API	✓ Success
Timestamp	✓ Success
Caffeine API	✓ Success
Script Security	✓ Success
Pipeline: Supporting APIs	✓ Success
Plugin Utilities API	✓ Success
Font Awesome API	✓ Success
Bootstrap 5 API	✓ Success
Jackson Annotations 2 API	✓ Success
SnakeYAML Engine API	✓ Success
Woodstox Core API	✓ Success
Jackson 3 API	✓ Success
jQuery3 API	✓ Success
ECharts API	✓ Success
Prism API	✓ Success
Display URL API	✓ Success
Checks API	✓ Success
JUnit	✓ Success
Matrix Project	✓ Success
Resource Disposer	✓ Success
Workspace Cleanup	✓ Success
Ant	✓ Success
OkHttp	✓ Success
JavaBeans Activation Framework (JAF) API	✓ Success
JAXB	✓ Success
JSON Api	✓ Success
Jakarta Activation API	✓ Success
SnakeYAML API	✓ Success
Jakarta XML Binding API	✓ Success
Jackson 2 API	✓ Success
Apache HttpComponents Client 4.x API	✓ Success
Pipeline: Job	✓ Success
Gradle	✓ Success

Pipeline: Milestone Step	✓ Success
Durable Task	✓ Success
Pipeline: Nodes and Processes	✓ Success
Pipeline: Build Step	✓ Success
Pipeline: SCM Step	✓ Success
Pipeline: Groovy	✓ Success
Pipeline: Groovy Libraries	✓ Success
Joda Time API	✓ Success
Pipeline: Model API	✓ Success
Pipeline: Stage Step	✓ Success
Pipeline: Declarative Extension Points API	✓ Success
Jakarta Mail API	✓ Success
Instance Identity	✓ Success
Mailer	✓ Success
Branch API	✓ Success
Pipeline: Multibranch	✓ Success
Pipeline: Stage Tags Metadata	✓ Success
Pipeline: Input Step	✓ Success
Pipeline: Basic Steps	✓ Success
Pipeline: Declarative	✓ Success
Pipeline	✓ Success
Java JSON Web Token (JWT)	✓ Success
GitHub API	✓ Success
Mina SSHD API :: Common	✓ Success
Mina SSHD API :: Core	✓ Success
Gson API	✓ Success
Git client	✓ Success
Git	✓ Success
GitHub	✓ Success
GitHub Branch Source	✓ Success
Pipeline: GitHub Groovy Libraries	✓ Success
Metrics	✓ Success
Pipeline Graph View	✓ Success
Git	✓ Success
EDDSA API	✓ Success
Trilead API	✓ Success
SSH Build Agents	✓ Success
Matrix Authorization Strategy	✓ Success
LDAP	✓ Success
jsoup API	✓ Success
Email Extension	✓ Success
Mailer	✓ Success
Theme Manager	✓ Success
Dark Theme	✓ Success
Loading plugin extensions	✓ Success
Amazon Web Services SDK 2 :: Core	✓ Success
Amazon Web Services SDK :: Minimal	✓ Success
Dev Tools Symbols API	✓ Success
Amazon Web Services SDK 2 :: EC2	✓ Success
Amazon Web Services SDK :: EC2	✓ Success
AWS Credentials	✓ Success
AWS Global Configuration	✓ Success
Amazon Web Services SDK :: SQS	✓ Success
Amazon Web Services SDK :: SNS	✓ Success
Amazon Web Services SDK :: Api Gateway	✓ Success
Amazon Web Services SDK :: CloudFormation	✓ Success
Amazon Web Services SDK :: Elastic Beanstalk	✓ Success
Amazon Web Services SDK :: Elastic Load Balancing V2	✓ Success
Amazon Web Services SDK :: IAM	✓ Success

Amazon Web Services SDK :: ECR	 Success
Amazon Web Services SDK :: CloudFront	 Success
Amazon Web Services SDK :: Lambda	 Success
Amazon Web Services SDK :: Organizations	 Pending
Amazon Web Services SDK :: CodeDeploy	 Pending
Pipeline: AWS Steps	 Pending
Netty API	 Pending
AWS Java SDK :: HTTP Clients :: Netty Non-Blocking I/O	 Pending
Amazon Web Services SDK 2 :: S3	 Pending
Copy Artifact	 Pending
S3 publisher	 Pending
Loading plugin extensions	 Pending

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running



 Jenkins is restarting

Your browser will reload automatically when Jenkins is ready.

Safe Restart
Builds on agents can usually continue.



AWS

Configuration for Amazon Web Services.

Amazon Credentials

Region ?

Europe (London)



Amazon Credentials ?

AKIAQESY6L3VY3DHXKO3 (Login details to access my AWS Account)



+ Add

Save

Apply



Credentials

+ Add Credentials



my-cba-aws-credential AKIAQESY6B3VY3DHXK03



System - Global - Login details to access my AWS Account



Stores scoped to Jenkins:



System

Domains:

Global

```
ubuntu@ip-10-0-15-245:~$ mvn --version
```

```
Apache Maven 3.8.7
```

```
Maven home: /usr/share/maven
```

```
Java version: 21.0.10, vendor: Ubuntu, runtime: /usr/lib/jvm/java-21-openjdk-amd64
```

```
Default locale: en, platform encoding: UTF-8
```

```
OS name: "linux", version: "6.17.0-1012-aws", arch: "amd64", family: "unix"
```

```
ubuntu@ip-10-0-15-245:~$ █
```



Tools

Configure tools, their locations and automatic installers.

Maven Configuration

Default settings provider

Use default maven settings



Default global settings provider

Use default maven global settings



JDK installations

+ Add JDK

Git installations

Git



Name

Default

Path to Git executable ?

git

☐ Install automatically ?

+ Add Git

Gradle installations

+ Add Gradle

Ant installations

+ Add Ant

Maven installations

+ Add Maven

Maven



Name

cba-maven-3.8.7

MAVEN_HOME

/usr/share/maven

☐ Install automatically ?

+ Add Maven

Save

Apply

```
ubuntu@ip-10-0-15-245:~$ sudo update-alternatives --config java
There is 1 choice for the alternative java (providing /usr/bin/java).
```

Selection	Path	Priority	Status
* 0	/usr/lib/jvm/java-21-openjdk-amd64/bin/java	2111	auto mode
1	/usr/lib/jvm/java-21-openjdk-amd64/bin/java	2111	manual mode

```
Press <enter> to keep the current choice[*], or type selection number: █
```

i-076382a5c61ef5e51 (jenkins_server)

PublicIPs: 3.9.117.92 PrivateIPs: 10.0.15.245

```
ubuntu@ip-10-0-15-245:~$ sudo nano /etc/environment
ubuntu@ip-10-0-15-245:~$ source /etc/environment
ubuntu@ip-10-0-15-245:~$ echo $JAVA_HOME
/usr/lib/jvm/java-21-openjdk-amd64
ubuntu@ip-10-0-15-245:~$ █
```

Tools

Configure tools, their locations and automatic installers.

Maven Configuration

Default settings provider

Use default maven settings



Default global settings provider

Use default maven global settings



JDK installations

+ Add JDK

⌵ JDK

Name

cba-jdk

JAVA_HOME

/usr/lib/jvm/java-21-openjdk-amd64

! /usr/lib/jvm/java-21-openjdk-amd64 doesn't look like a JDK directory

☐ Install automatically

+ Add JDK

Git installations

⌵ Git

Name

Default

Path to Git executable

git

☐ Install automatically

+ Add Git

Gradle installations

+ Add Gradle

Ant installations

+ Add Ant

Maven installations

Maven installations ▾  Edited

Save Apply

cloudboostauk / ci-demo

Q Type to search

<> Code

Pull requests

Agents

Actions

Projects

Security and quality

Insights

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](#).

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks](#).

Required fields are marked with an asterisk (*).

Owner *

Repository name *

samuelakosaonyejekwe

/

ci-demo

ci-demo is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

Repo for CI demo

16 / 350 characters

☒ Copy the main branch only

Contribute back to cloudboostauk/ci-demo by adding your own branch. [Learn more.](#)

You are creating a fork in your personal account.

Create fork

© 2026 GitHub, Inc.

Terms

Privacy

Security

Status

Community

Docs

Contact

Manage cookies

Do not share my personal information

📌 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

Preview
- 🔗 Webhooks**
- 🤖 Copilot
- 🌐 Environments
- 💻 Codespaces
- 📄 Pages
- Security and quality
- 🔍 Advanced Security
- 🔑 Deploy keys
- 🔒 Secrets and variables
- Integrations
- 🔗 GitHub Apps
- ✉ Email notifications

Add webhook

Webhooks

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](#).

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://3.9.117.92:8080/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

Webhooks

Access

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

Code and automation

 Tags

Rules

Ⓢ Actions

Models

[Preview](#) Webhooks Copilot

Environments

 Codespaces

Pages

Security and quality

🔍 Advanced Security

 Deploy keys

- ❖ Secrets and variables

Integrations

GitHub Apps

 Email notifications

 **Jenkins**

Manage Jenkins

Credentials

Global

Credentials that should be available everywhere

 my-cba-aws-credential AKIAQSE...

Login details to access my AWS Account

< Add Username with password

×

Input field: samuelakosaonyekwe

☐ Treat username as secret ?

Password

Input field:

ID ?

Input field: github-access

Description ?

Input field: github access

Create

+ Add Credentials

 

REST API Jenkins 2.555.1



Global

+ Add Credentials

Credentials that should be available everywhere.

 my-cba-aws-credential AKIAQESY6I3VY3LHXR03	 
Login details to access my AWS Account	
 github-access samuelakosaonyejekwe/*****	 
github access	

🔍 Type to search

<> Code

🔗 Pull requests

🛠 Agents

🔄 Actions

📁 Projects

📖 Wiki

🛡 Security and quality

📊 Insights

⚙ Settings

🔔 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](#).

ci-demoPublic

forked from cloudboostauk/ci-demo

📌 Pin

👁 Watch 0

🍴 Fork 0

★ Star 0

🔗 main1 Branch0 Tags

🔍 Go to file1

Add file

<> Code

This branch is up to date with cloudboostauk/ci-demo:main

Contribute

Sync fork

👤 akinwunmi-akinrimisi Merge pull request cloudboostauk#1 from biibitayo/patch-1

55d51af · yesterday

🕒 18 Commits

📁 maven.projectUpdate Maven compiler source and target to 172 days ago

📄 JenkinsfileUpdate Jenkinsfile3 years ago

📄 README.mdAdding Readme.md4 years ago

📄 trigger_file.txtUpdate trigger_file.txt3 years ago

📖 README

Repo for CI process demo

- builds a maven project
- jenkinsfile

About

Repo for CI demo

📖 Readme

🔄 Activity

★ 0 stars

👁 0 watching

🍴 0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors

No contributors

Languages

Java100.0%

Suggested workflows

Based on your tech stack

Java with Ant

Configure

Build and test a Java project with Apache Ant.

Clojure

Configure

Build and test a Clojure project with Leiningen.

Scala

Configure

Build and test a Scala project with SBT.

More workflows

Dismiss suggestions

© 2026 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Community](#)

[Docs](#)

[Contact](#)

[Manage cookies](#)

[Do not share my personal information](#)

samuelakosaonyejekwe / ci-demo

Q Type to search

<> Code

Pull requests

Agents

Actions

Projects

Wiki

Security and quality

Insights

Settings

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](#).

Files

ci-demo / Jenkinsfile in main

Cancel changes Commit changes...

main

Go to file

> maven_project

Jenkinsfile

README.md

trigger_file.txt

Edit Preview

4 No wrap

```
u-aws-credential', region: 'eu-west-2') {
he tested jar file to s3 for later deployments" '''
sEnabled: true, payloadSigningEnabled: true, file: 'maven_project/target/my-app-1.0-SNAPSHOT.jar', bucket: 'materialsfromlocalmachinescan', p

cuted '''
```

Use `Control + Shift + n` to toggle the `Tab` key moving focus. Alternatively, use `esc` then `Tab` to move to the next interactive element on the page.

