

School: CLOUDBOOSTA	Course: DEVOPS
Name: AKOSA SAMUEL ONYEJEKWE	Module: Jenkins
	Class 1: Lab-3_ Jenkins-1
Student No: CBALC0168	Issue Date: May 6, 2026

Tasks:

1. Launch AWS EC2 instance for Jenkins server
2. Configure EC2 security groups and networking
3. Install Jenkins using EC2 user data script
4. Access Jenkins web interface
5. Unlock Jenkins server
6. Install Jenkins suggested plugins
7. Create Jenkins administrator account
8. Configure Jenkins instance settings
9. Install AWS-related Jenkins plugins
10. Verify plugin installation status
11. Restart Jenkins after plugin installation
12. Configure AWS credentials in Jenkins
13. Configure AWS region settings in Jenkins
14. Create Jenkins AWS pipeline project
15. Validate Jenkins AWS integration

Goal:

To deploy Jenkins on AWS EC2 infrastructure, install AWS integration plugins, configure AWS credentials within Jenkins, and prepare Jenkins for AWS-based CI/CD automation workflows in a DevOps environment.

Result:

- AWS EC2 instance successfully launched for Jenkins server
- Jenkins installed successfully on Ubuntu EC2 instance
- Jenkins web interface accessed successfully
- Jenkins administrator setup completed successfully
- Jenkins plugins installed successfully
- AWS-related Jenkins plugins configured successfully
- Jenkins restarted successfully after plugin installation
- AWS credentials configured successfully in Jenkins
- AWS region configuration completed successfully
- Jenkins AWS pipeline project created successfully
- Jenkins AWS integration validated 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 secure SSH access
- Reviewed launch instance summary before deployment

Step 2: Configured Security Groups

- Configured inbound security rules:
- SSH → Port 22
- HTTP → Port 80
- Custom TCP → Port 8080 for Jenkins
- Allowed access from:
0.0.0.0/0
- Verified VPC and subnet configuration

Step 3: Added Jenkins Installation User Data Script

- Configured EC2 user data section
- Added automated Jenkins installation commands
- 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 Jenkins EC2 instance launched successfully
- Verified instance state:
Running
- Verified public IP assignment
- Confirmed instance name:
jenkins_sam

Step 5: Connected to Jenkins Server

- Opened EC2 Connect page
- Connected using public IPv4 address
- Verified username:
ubuntu
- Successfully accessed Jenkins server terminal

Step 6: Retrieved Jenkins Administrator Password

- Ran command:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```
- Copied Jenkins administrator password
- Verified successful password retrieval

Step 7: Accessed Jenkins Web Interface

- Opened browser using:

```
http://18.175.54.158:8080
```
- Accessed Unlock Jenkins setup page
- Entered administrator password successfully

Step 8: Installed Suggested Jenkins Plugins

- Selected:
 - Install suggested plugins
- Jenkins automatically installed required plugins
- Verified plugin installation progress successfully

Step 9: Created Jenkins Administrator User

- Configured Jenkins administrator account
- Added:
 - Username
 - Password
 - Full name
 - Email address
- Saved Jenkins administrator configuration

Step 10: Configured Jenkins Instance URL

- Verified Jenkins instance configuration:
 - `http://18.175.54.158:8080/`
- Saved instance configuration successfully

Step 11: Verified Jenkins Installation Completion

- Verified Jenkins setup message:
 - “Jenkins is ready!”
- Selected:
 - Start using Jenkins
- Successfully accessed Jenkins dashboard

Step 12: Opened Jenkins Plugin Manager

- Navigated to:
 - Manage Jenkins → Plugins
- Opened:

Available plugins section

- Searched for AWS-related plugins

Step 13: Selected AWS Plugins for Installation

- Selected plugins:
 - AWS Global Configuration
 - Pipeline AWS Steps
 - S3 Publisher
- Additional AWS-related plugins and dependencies
- Initiated plugin installation process

Step 14: Verified Plugin Installation Progress

- Opened:
 - Download progress section
- Verified successful installation of plugins:
 - AWS SDK plugins
 - CloudFormation plugins
 - EC2 plugins
 - S3 plugins
 - CodeDeploy plugins
 - Lambda plugins
 - CloudFront plugins
 - API Gateway plugins
 - ECR plugins
 - SNS plugins
- Verified all plugin installation statuses showed:
 - Success

Step 15: Restarted Jenkins

- Jenkins automatically initiated restart process
- Verified restart message:
 - “Jenkins is restarting”
- Waited for Jenkins services to reload successfully

Step 16: Configured AWS Settings in Jenkins

- Navigated to:
Manage Jenkins → AWS
- Configured:
 - AWS Region → Europe (London)
 - Added AWS account credentials
 - Saved AWS configuration successfully

Step 17: Created Jenkins AWS Pipeline Project

- Navigated to:
New Item
- Created project named:
my-first-upload-to-aws
- Selected:
Pipeline project
- Clicked OK

Summary

- Jenkins server was successfully deployed on AWS EC2 infrastructure
- Jenkins installation and setup completed successfully
- AWS integration plugins installed successfully in Jenkins
- Jenkins restart completed successfully after plugin installation
- AWS credentials and region configuration completed successfully
- Jenkins environment prepared for AWS CI/CD automation workflows
- AWS service integration capabilities enabled successfully within Jenkins
- Jenkins AWS pipeline project created successfully
- AWS-based DevOps automation environment configured successfully

Lab objective achieved:

Implemented Jenkins deployment and AWS integration using Jenkins AWS plugins, AWS credentials configuration, and pipeline project preparation for cloud-based CI/CD automation workflows in a DevOps environment.

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-2

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-2 created 2026-05-08T04:45:24.367Z

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 8 GiB gp3 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
- 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) Info Last updated 2 minutes ago  [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Saved filter sets Choose filter set

< 1 > 

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	jenkins_sam	i-046e6ece801188238	Running  	t3.small	3/3 checks passed	View alarms

Select an instance  

aws

Search

[Alt+S]

1

Europe (London)

Samuel A. O. (009850210027)

cba_admin

EC2

Instances

i-046e6ece801188238

Connect to instance

Connect using a Public IP

Connect using a public IPv4 or IPv6 address.

Public IPv4 address

18.175.54.158

IPv6 address

Connect using a Private IP

Connect using a private IP address and a VPC endpoint.

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel

Connect

CloudShell

Feedback

Console Mobile App

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

Privacy

Terms

Cookie preferences

```
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-14-41:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword  
061fc55b084741adbae3f66f93ac9dc3  
ubuntu@ip-172-31-14-41:~$
```

i-046e6ece801188238 (jenkins_sam)

PublicIPs: 18.175.54.158 PrivateIPs: 172.31.14.41

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

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

✓ Folders	Workspace Cleanup	Ant	Gradle	Jenkins API
☐ Timestamp	☐ Workspace Cleanup	☐ Ant	☐ Gradle	Folders
☐ Pipeline	☐ GitHub Branch Source	☐ Pipeline: GitHub Groovy Libraries	☐ Pipeline Graph View	OWASP Markup Formatter
☐ Git	☐ SSH Build Agents	☐ Matrix Authorization Strategy	☐ LDAP	** ASM API
☐ Email Extension	☐ Mailer	☐ Dark Theme		** JSON Path API
				** Struts
				** Pipeline: Step API
				** Commons-Text API
				** Token Macro
				Build Timeout
				** hudsoncastile API
				** - required dependency

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://18.175.54.158:8080/`

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 p

Install [Dropdown] Refresh

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

Install	Name	Released	Health
☒	AWS Global Configuration 159.v082a_ecd03fe6 Library plugins (for use by other plugins) A Jenkins plugin to configure AWS related settings	1 mo 21 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 27 days ago	94
☐	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

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

→ [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 Cloudboosta AWS Account)

+ Add

Save

Apply

New Item

Enter an item name

my-first-upload-to-aws

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
- Triggers
- Pipeline
- Advanced

General

Enabled

Description

my-first-upload-to-aws

Plain text [Preview](#)

- ☐ Discard old builds ?
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☐ GitHub project
- ☐ Permission to Copy Artifact
- ☐ 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 ?

try sample Pipeline...

```

1
2
3
4
5
6
7
8
9
10
11 n: 'eu-west-2' {
12   to.txt
13   signingEnabled: true, file: 'hello.txt', bucket: 'materialsfromlocalmachinesam', path: 'hello.txt'
14
15

```

- ☒ Use Groovy Sandbox ?

Advanced

Advanced ▾

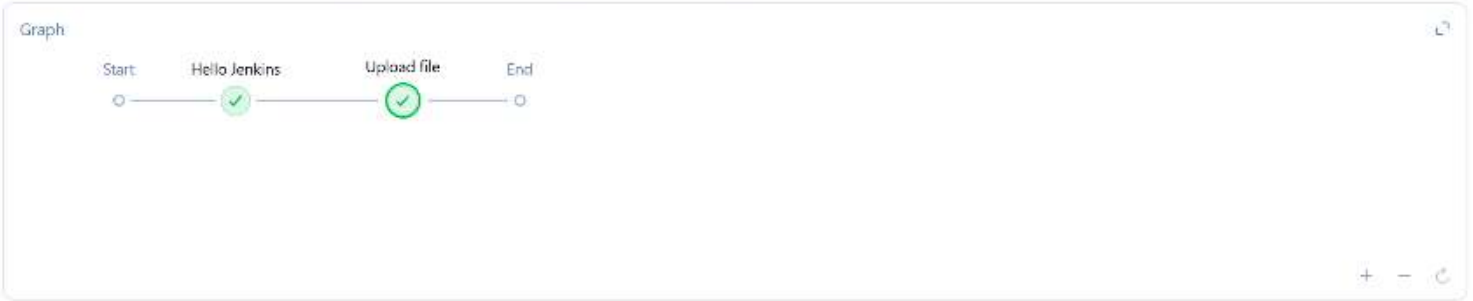
Save

Apply

✓ #1

🔄 Rerun ⚙️

👤 Started by Akosa Samuel Onyejekwe ⌚ Started 1 min 48 sec ago ⌚ Queued 29 ms ⌚ Took 6.6 sec



🔍 Search

- ✓ Hello Jenkins 0.46s
- ✓ Upload file 1s

✓ Upload file 1s ⌚ Started 2m ago 🖨️ Jenkins ⋮

✓ echo "hello again Cloudboosta Students"> hello.txt 0.3s ↗

✓ Copy file to S3 0.88s ↗

⬆️ ⬆️ 355.1

