

Tutorial: Create a four-stage pipeline

Topics

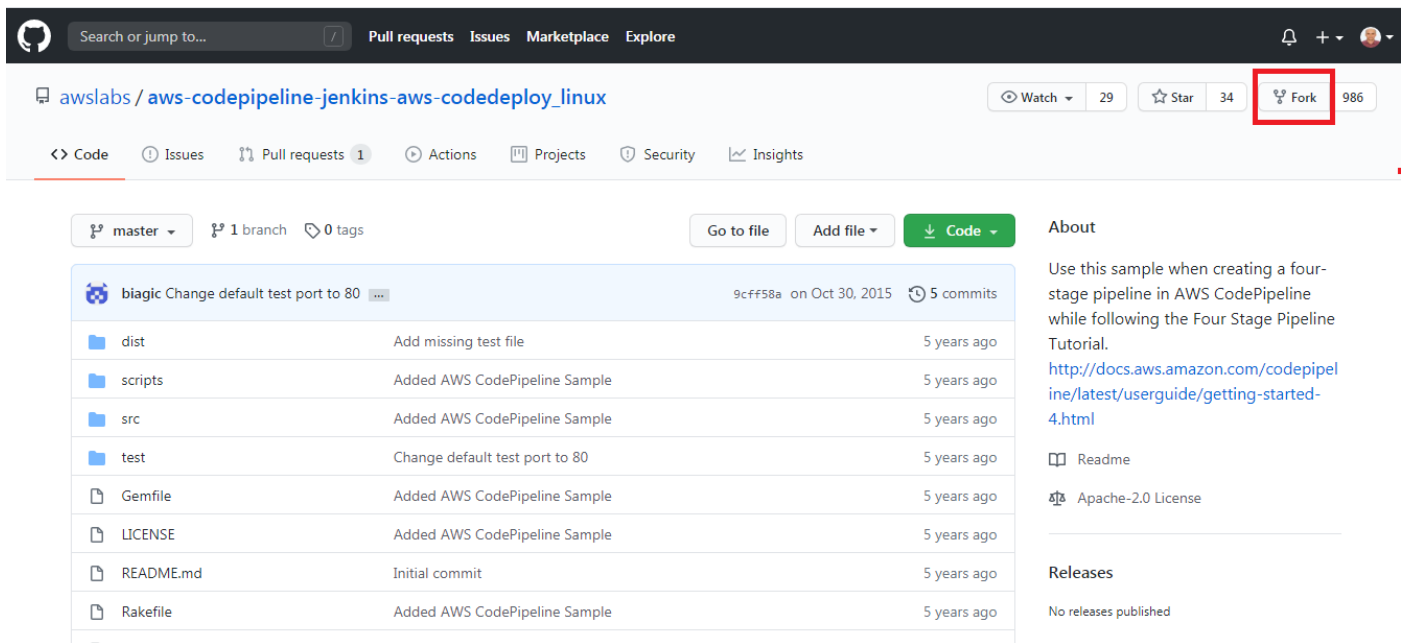
- [Step 1: Complete prerequisites](#)
 - [Step 2: Create a pipeline in CodePipeline](#)
 - [Step 3: Add another stage to your pipeline](#)
 - [Step 4: Clean up resources](#)
-

Step 1: Complete prerequisites

Copy or clone the sample into a GitHub repository

To clone the sample and push to a GitHub repository

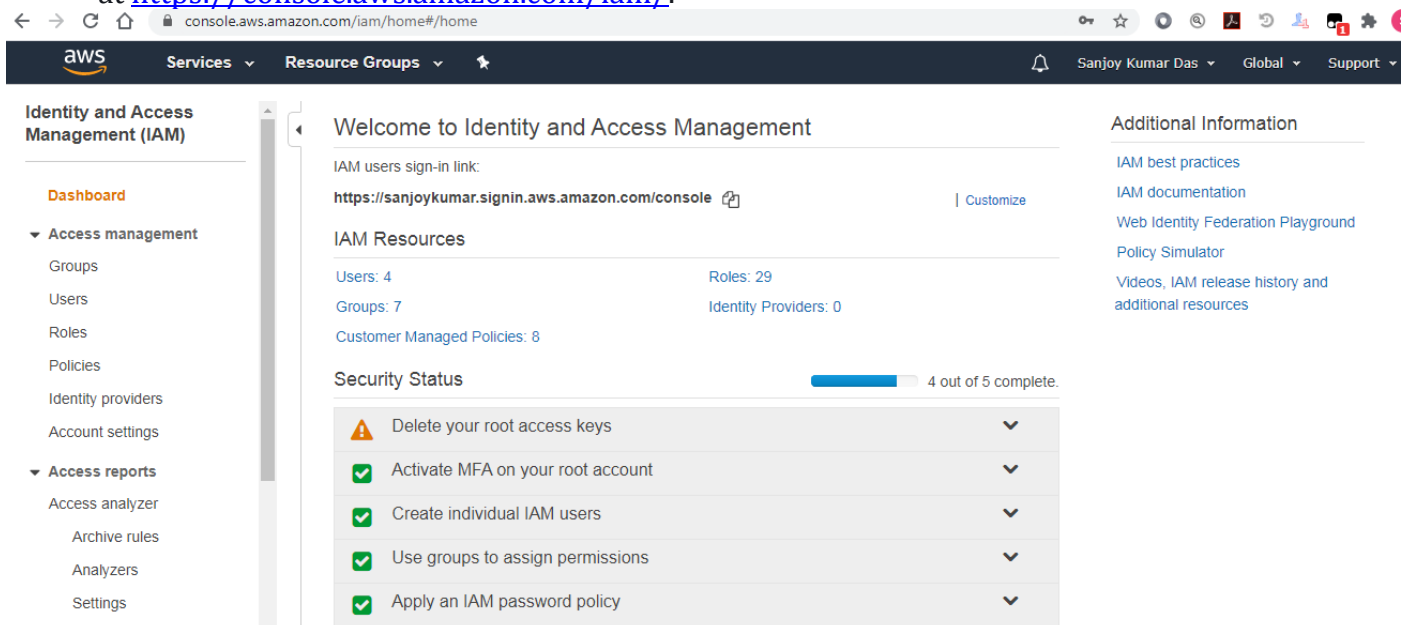
1. Download the sample code from the GitHub repository, or clone the repositories to your local computer. There are two sample packages:
 - If you will be deploying your sample to Amazon Linux, RHEL, or Ubuntu Server instances, choose [aws-codepipeline-jenkins-aws-codedeploy linux.zip](#).
 - If you will be deploying your sample to Windows Server instances, choose [AWSCodePipeline-Jenkins-AWSCodeDeploy Windows.zip](#).
2. From the repository, choose **Fork** to clone the sample repo into a repo in my Github account.



Create an IAM role to use for Jenkins integration

As a best practice, consider launching an EC2 instance to host your Jenkins server and using an IAM role to grant the instance the required permissions for interacting with CodePipeline.

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.



2. In the IAM console, in the navigation pane, choose **Roles**, and then choose **Create role**.

aws Services Resource Groups

Sanjoy Kumar Das Global

Identity and Access Management (IAM)

Dashboard

Access management

Groups

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- [IAM Roles FAQ](#)
- [IAM Roles Documentation](#)
- [Tutorial: Setting Up Cross Account Access](#)
- [Common Scenarios for Roles](#)

Create role Delete role

3. Under **Select type of trusted entity**, choose **AWS service**. Under **Choose the service that will use this role**, choose **EC2**. Under **Select your use case**, choose **EC2**.

aws Services Resource Groups

Sanjoy Kumar Das Global Support

Create role

1 2 3 4

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML
SAML 2.0 federation
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway CodeGuru ElastiCache Kinesis RoboMaker

* Required

Cancel **Next: Permissions**

4. Choose **Next: Permissions**. On the **Attach permissions policies** page, select the **AWSCodePipelineCustomActionAccess** managed policy, and then choose **Next: Tags**. Choose **Next: Review**.

Create role

1
2
3
4

Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies
Showing 1 result

	Policy name	Used as
<input checked="" type="checkbox"/>	AWSCodePipelineCustomActionAccess	None

* Required

Cancel

Previous

Next: Tags

Create role

1
2
3
4

Add tags (optional)

IAM tags are key-value pairs you can add to your role. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this role. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="Name"/>	<input type="text" value="AWSCodePipelineCustomActionAccess"/>	<input checked="" type="checkbox"/>
<input type="text" value="Add new key"/>	<input type="text"/>	

You can add 49 more tags.

Cancel

Previous

Next: Review

- On the **Review** page, in **Role name**, enter the name of the role to create specifically for Jenkins integration (for example, *JenkinsAccess*), and then choose **Create role**.

aws Services Resource Groups

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* JenkinsAccess

Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies AWSCodePipelineCustomActionAccess

Permissions boundary Permissions boundary is not set

* Required

Cancel Previous Create role

When you create the EC2 instance where you will install Jenkins, in **Step 3: Configure Instance Details**, make sure you choose the instance role (for example, *JenkinsAccess*).

Install and configure Jenkins and the CodePipeline Plugin for Jenkins

To install Jenkins and the CodePipeline Plugin for Jenkins

1. Create an EC2 instance where you will install Jenkins, and in **Step 3: Configure Instance Details**, make sure you choose the instance role you created (for example, *JenkinsAccess*).

aws Services Resource Groups

New EC2 Experience

Launch Instance

Connect Actions

EC2 Dashboard

Events

Tags

Limits

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Filter by tags and attributes or search by keyword

You do not have any running instances in this region.

First time using EC2? Check out the [Getting Started Guide](#).

Click the Launch Instance button to start your own server.

Launch Instance

Select an instance above

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Cancel and Exit

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Search by Systems Manager parameter

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Free tier only

Amazon Linux

Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-02354e95b39ca8dec (64-bit x86) / ami-0c5bf07e510b75b11 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebsVirtualization type: hvmENA Enabled: Yes

64-bit (x86)

64-bit (Arm)

Select

Amazon Linux

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-09d8b5222f2b93bf0

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

64-bit (x86)

Select

Feedback

English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of U

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance typesCurrent generationShow/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

CancelPreviousReview and LaunchNext: Configure Instance Details

Feedback

English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances1Launch into Auto Scaling Group

Purchasing optionRequest Spot instances

Networkvpc-a3d09ad9 (default)Create new VPC

Subnetsubnet-fe9fa199 | Default in us-east-1a4091 IP Addresses availableCreate new subnet

Auto-assign Public IPEnable

Placement groupAdd instance to placement group

Capacity ReservationOpen

IAM roleJenkinsAccessCreate new IAM role

CancelPreviousReview and LaunchNext: Add Storage

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0ed3f0f331ab4cbc7	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel

Previous

Review and Launch

Next: Add Tags

Feedback

English (US)

© 2008 – 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
Name	Jenkins	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel

Previous

Review and Launch

Next: Configure Security Group

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name: JenkinsSG

Description: Jenkins Security Group

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP	TCP	8080	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

Cancel

Previous

Review and Launch

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Improve your instances' security. Your security group, JenkinsSG, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-09d8b522f2b93bf0

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

Feedback English (US) © 2008 - 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Improve your instances' security. Your security group, JenkinsSG, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-09d8b522f2b93bf0

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

GSDAKey

☒ I acknowledge that I have access to the selected private key file (GSDAKey.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

Improve your instances' security. Your security group, JenkinsSG, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Free tier eligible

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-09d8b522f2b93bf0

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

New EC2 Experience

EC2 Dashboard

Events

Tags

Limits

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
Jenkins	i-04583aba01c89b363	t2.micro	us-east-1a	running	Initializing	None	ec2-3-235-

Instance: i-04583aba01c89b363 (Jenkins) Public DNS: ec2-3-235-247-31.compute-1.amazonaws.com

Description

Instance ID	Instance state	Instance type	Finding	Private DNS	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Elastic IPs	Availability zone
i-04583aba01c89b363	running	t2.micro	Opt-in to AWS Compute Optimizer for recommendations. Learn more	ip-172-31-11-25.ec2.internal	ec2-3-235-247-31.compute-1.amazonaws.com	3.235.247.31	-	-	us-east-1a

2. Install Jenkins on the EC2 instance.


```
# wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

```
#yum upgrade
```

```
# yum install jenkins java-1.8.0-openjdk-devel
```

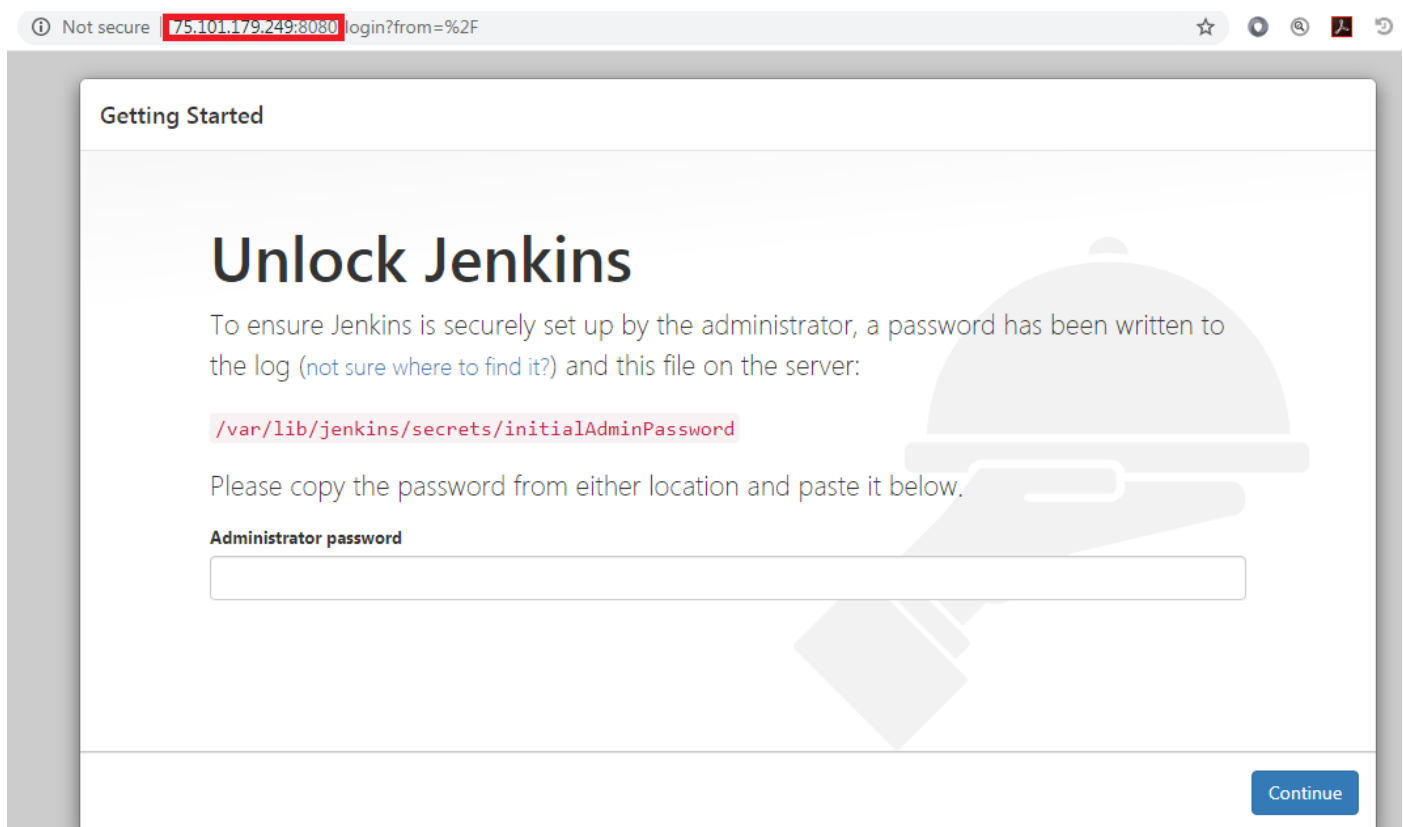
Start Jenkins: You can start the Jenkins service with the command:

```
# sudo service jenkins start
```

You can check the status of the Jenkins service using the command:

```
# sudo service jenkins status
```

3. Launch Jenkins, and on the home page, choose **Manage Jenkins**.



```
[root@ip-172-31-13-175 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
```

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.

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding	<div>** Pipeline: Step API</div> <div>** Token Macro</div> <div>Build Timeout</div> <div>** Credentials</div> <div>** Plain Credentials</div> <div>** SSH Credentials</div> <div>Credentials Binding</div> <div>** SCM API</div> <div>** Pipeline: API</div> <div>Timestamp</div> <div>** Script Security</div> <div>** Pipeline: Supporting APIs</div> <div>** Durable Task</div> <div>** Pipeline: Nodes and Processes</div> <div>** JQuery3 API</div> <div>** Snakeyaml API</div> <div>** Jackson 2 API</div> <div>** Plugin Utilities API</div> <div>** ECharts API</div> <div>** - required dependency</div>
✓ Timestamp	Workspace Cleanup	Ant	Gradle	
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	Subversion	SSH Build Agents	Matrix Authorization Strategy	
PAM Authentication	LDAP	Email Extension	Mailer	

Jenkins 2.235.4

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.235.4

[Skip and continue as admin](#)[Save and Continue](#)

Getting Started

Instance Configuration

Jenkins URL:

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 2.235.4

Not now

[Save and Finish](#)<http://75.101.179.249:8080/>

Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

← → ↻ 🏠 ⓘ Not secure | 75.101.179.249:8080

Jenkins 🔍 search 🗑️ 👤 Sanjoy Kumar Das 🚪 log out

Jenkins >

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

Welcome to Jenkins!

[Create an agent](#) or [configure a cloud](#) to set up distributed builds. [Learn more.](#)

[Create a job](#) to start building your software project.

Build Queue —

No builds in the queue.

Build Executor Status —

1 Idle
2 Idle

4. On the Manage Jenkins page, choose Manage Plugins.

← → ↻ 🏠 ⓘ Not secure | 75.101.179.249:8080/manage




Jenkins 🔍 search 🗑️ 👤 Sanjoy Kumar Das 🚪 log out

Jenkins >

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




Manage Jenkins

System Configuration

-  **Configure System**
Configure global settings and paths.
-  **Global Tool Configuration**
Configure tools, their locations and automatic installers.
-  **Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

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

Security

-  **Configure Global Security**
Secure Jenkins; define who is allowed to access/use the system.
-  **Manage Credentials**
Configure credentials
-  **Configure Credential Providers**
Configure the credential providers and types

Manage Users

5. Choose the Available tab, and in the Filter search box, enter AWS CodePipeline.

Choose CodePipeline Plugin for Jenkins from the list and choose Download now and install after restart.

← → ↻ 🏠 ⚠️ Not secure | 75.101.179.249:8080/pluginManager/available

Jenkins 🔍 search 🗑️ 👤 Sanjoy Kumar Das 🚪 log out

Jenkins > Plugin Manager >

📈 Back to Dashboard
⚙️ Manage Jenkins
🌱 Update Center

🔍 **AWS CodePipeline**

Updates **Available** Installed Advanced

Install ↑	Name	Version	Released
<input checked="" type="checkbox"/>	AWS CodePipeline aws Other Post-Build Actions Source Code Management AWS CodePipeline Integration	0.42	3 mo 1 day ago

Install without restart **Download now and install after restart** Update information obtained: 10 min ago **Check now**

- On the **Installing Plugins/Upgrades** page, select **Restart Jenkins when installation is complete and no jobs are running**.

Not secure | 75.101.179.249:8080/updateCenter/

Please wait while Jenkins is restarting ...

Your browser will reload automatically when Jenkins is ready.

- Choose **Back to Dashboard**.

75.101.179.249:8080/login?from=%2F



Welcome to Jenkins!

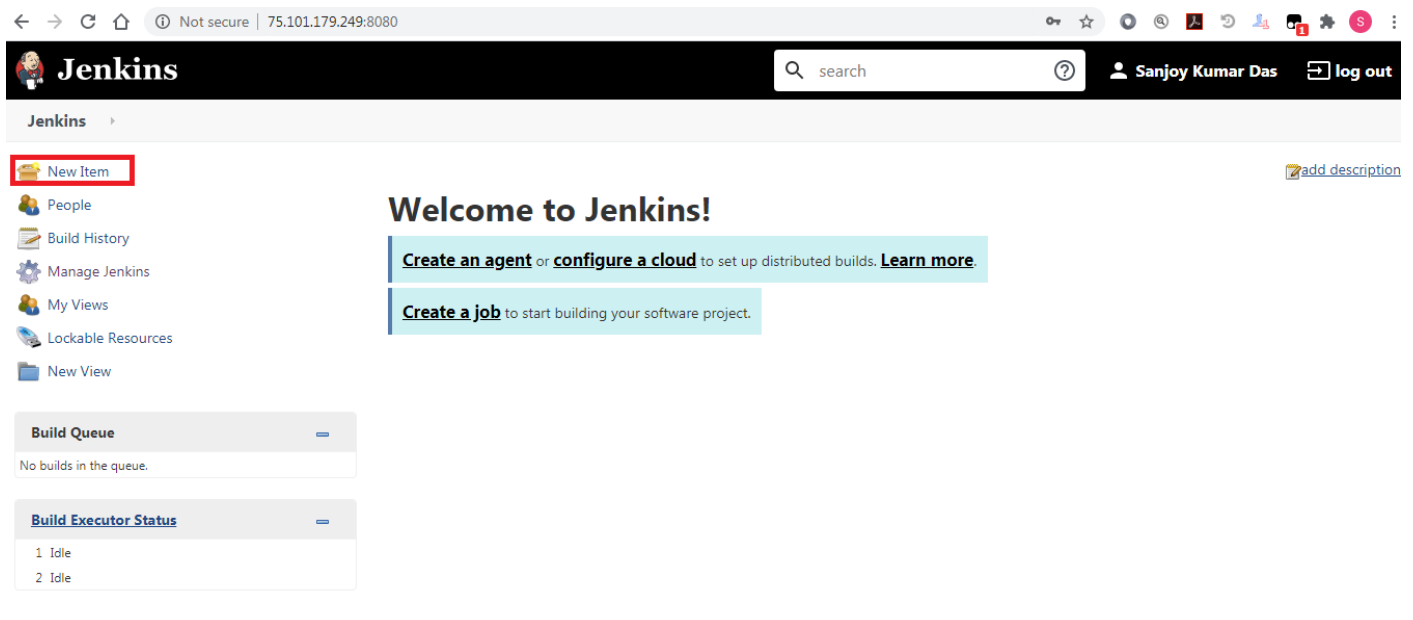
Admin

.....

Sign in

☐ Keep me signed in

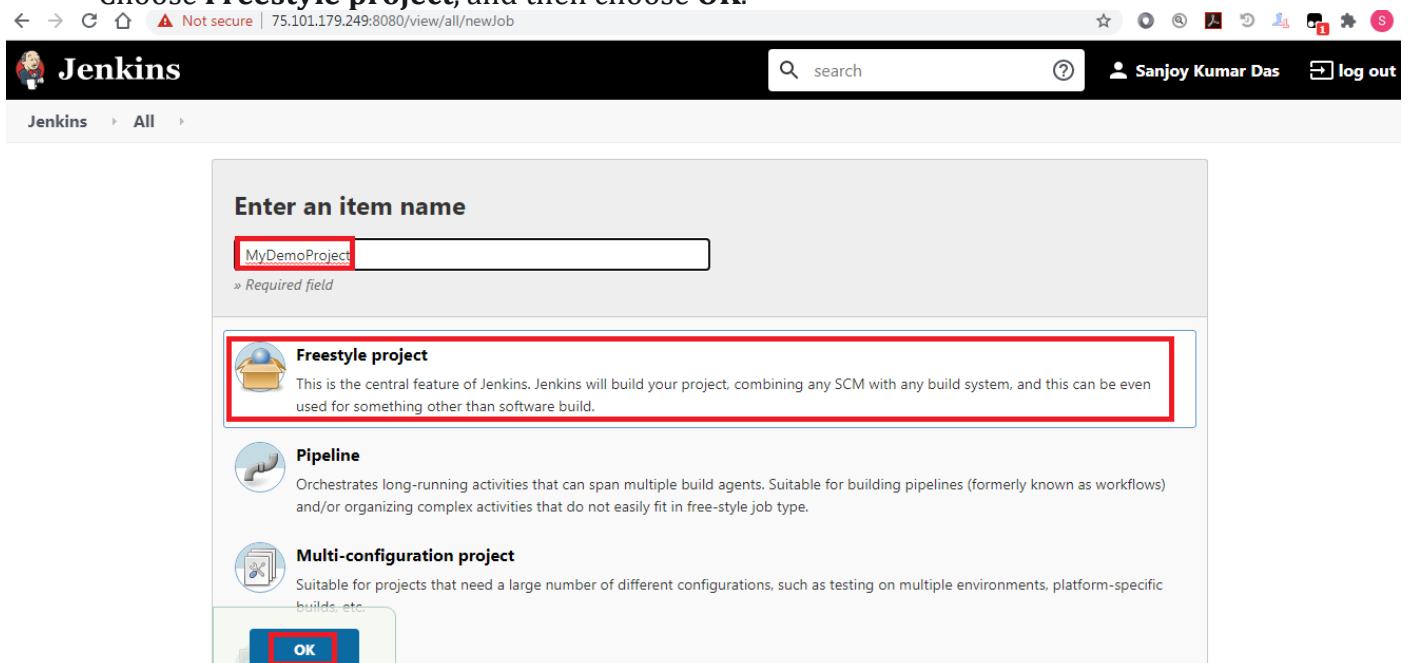
- On the main page, choose **New Item**.



The screenshot shows the Jenkins dashboard. At the top, there's a navigation bar with the Jenkins logo, a search bar, and a user profile for Sanjoy Kumar Das with a 'log out' button. Below the navigation bar, on the left, is a sidebar with a 'New Item' button highlighted by a red rectangle. Other sidebar items include 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. The main content area has a 'Welcome to Jenkins!' heading. Below it, there are two light blue boxes: one with the text 'Create an agent or configure a cloud to set up distributed builds. [Learn more.](#)' and another with 'Create a job to start building your software project.' Below these, there are two sections: 'Build Queue' showing 'No builds in the queue.' and 'Build Executor Status' showing two 'Idle' executors.

9. In **Item Name**, enter a name for the Jenkins project (for example, *MyDemoProject*).

Choose **Freestyle project**, and then choose **OK**.



The screenshot shows the 'New Item' configuration page in Jenkins. The browser address bar shows the URL '75.101.179.249:8080/view/all/newJob'. The page has the same navigation bar as the dashboard. Below the navigation bar, there's a breadcrumb 'Jenkins > All >'. The main content area is titled 'Enter an item name'. It features a text input field containing 'MyDemoProject', which is highlighted with a red rectangle. Below the input field, there's a red note that says '» Required field'. Underneath, there are three project type options: 'Freestyle project' (which is selected and highlighted with a red rectangle), 'Pipeline', and 'Multi-configuration project'. Each option has a brief description. At the bottom of the configuration area, there is a blue 'OK' button, also highlighted with a red rectangle.

10. On the configuration page for the project, select the **Execute concurrent builds if necessary** check box. In **Source Code Management**, choose **AWS CodePipeline**. If you have installed Jenkins on an EC2 instance and configured the AWS CLI with the profile for the IAM user you created for integration between CodePipeline and Jenkins, leave all of the other fields empty.

← → ↻ 🏠 ⓘ Not secure | 75.101.179.249:8080/job/MyDemoProject/configure ☆ 🔍 📄 ↺

Jenkins ▸ MyDemoProject ▸

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

Description

[Plain text] [Preview](#)

- ☐ Discard old builds ?
- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Disable this project ?
- ☒ Execute concurrent builds if necessary ?
- ☐ Quiet period ?

Save **Apply**

← → ↻ 🏠 ⓘ Not secure | 75.101.179.249:8080/job/MyDemoProject/configure ☆ 🔍 📄 ↺

Jenkins ▸ MyDemoProject ▸

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

Source Code Management

☐ None

☒ AWS CodePipeline

AWS Config

AWS Region US East (N. Virginia) us-east-1 ▾

Proxy Host ?

Proxy Port ?

Credentials

If these keys are left blank, the plugin will attempt to use credentials from the default provider chain. That is: Environment Variables, Java System properties, credentials profile file, and finally, EC2 Instance profile.

AWS Access Key ?

AWS Secret Key ?

Save **Apply** ☒

11. Choose **Advanced**, and in **Provider**, enter a name for the provider of the action as it will appear in CodePipeline (for example, *MyJenkinsProviderName*). Make sure that this name is unique and easy to remember. You will use it when you add a build action to your pipeline later in this tutorial, and again when you add a test action.

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

AWS Access Key

AWS Secret Key

Clear workspace before copying ☒

CodePipeline Action Type

Category

Please select a Category Type

Provider

Version

☐ Git

☐ Subversion

12. In **Build Triggers**, clear any check boxes, and then select **Poll SCM**. In **Schedule**, enter five asterisks separated by spaces, as follows:

* * * * *

This polls CodePipeline every minute.

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

Build Triggers

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

☐ Build after other projects are built

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☒ **Poll SCM**

Schedule

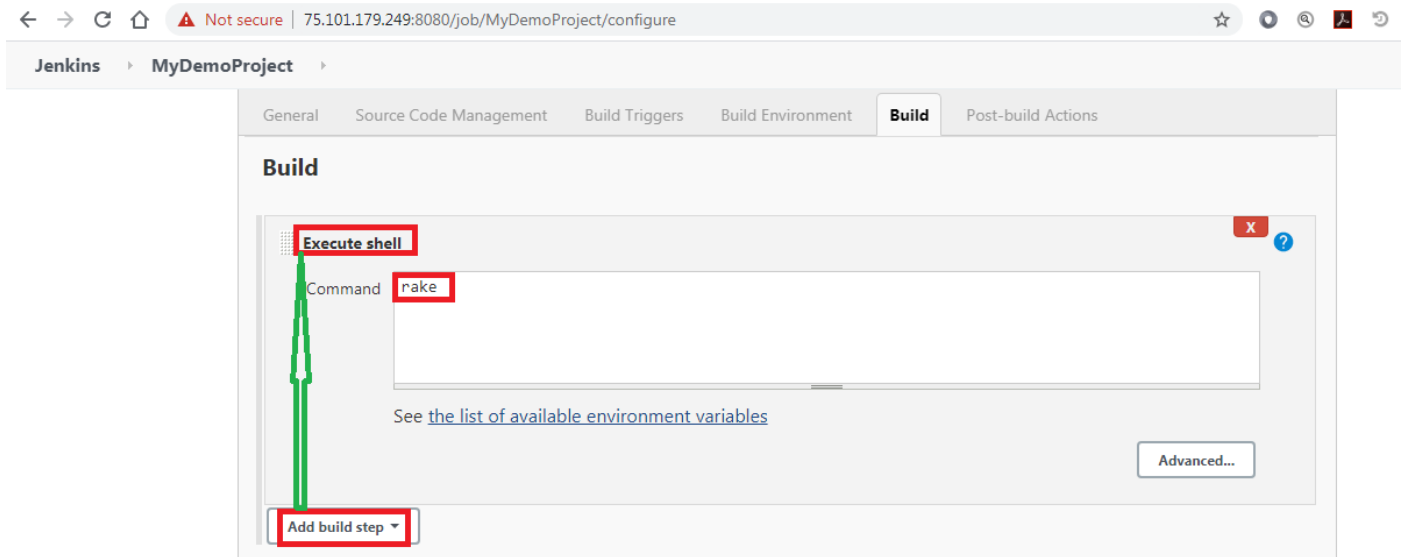
Do you really mean "every minute" when you say "***"? Perhaps you meant "H *****" to poll once per hour**

Would last have run at Thursday, August 13, 2020 3:45:00 PM UTC; would next run at Thursday, August 13, 2020 3:45:00 PM UTC.

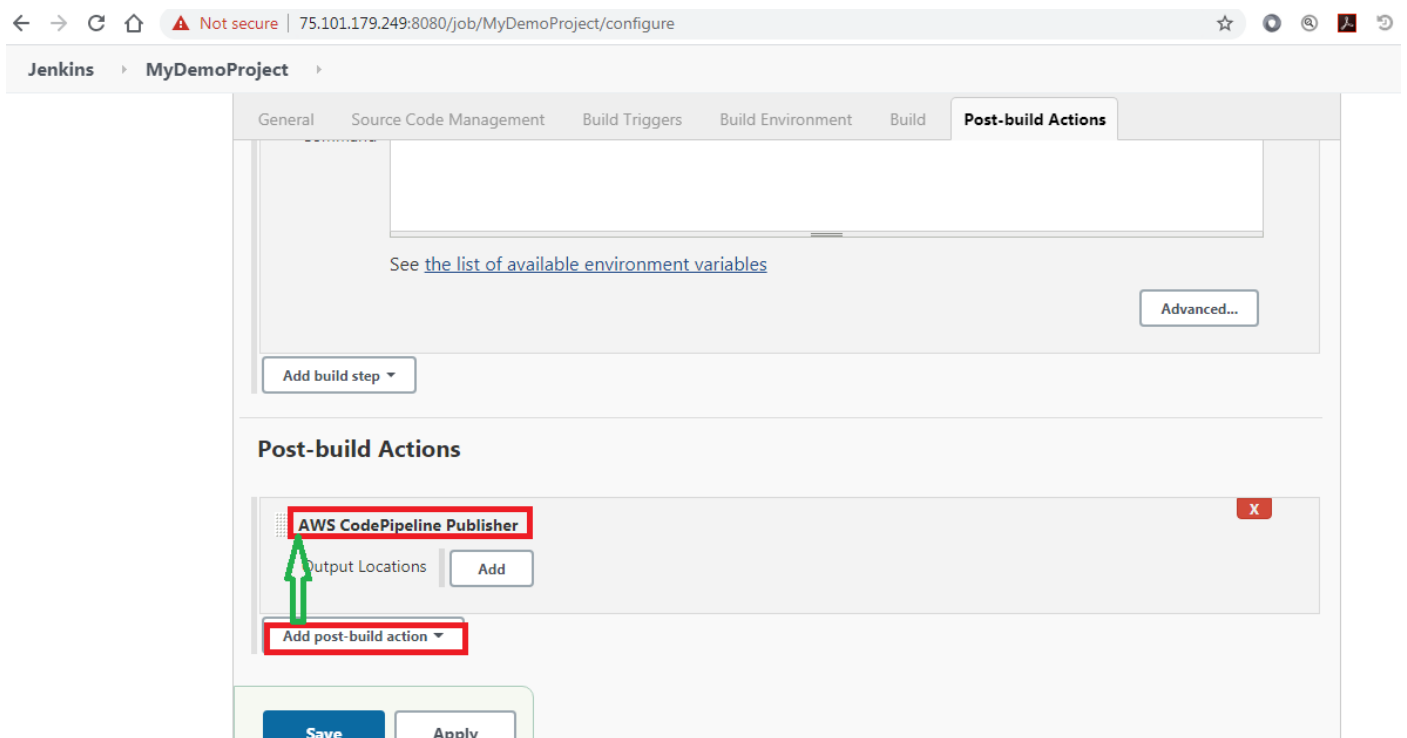
☐ Ignore post-commit hooks

Save **Apply**

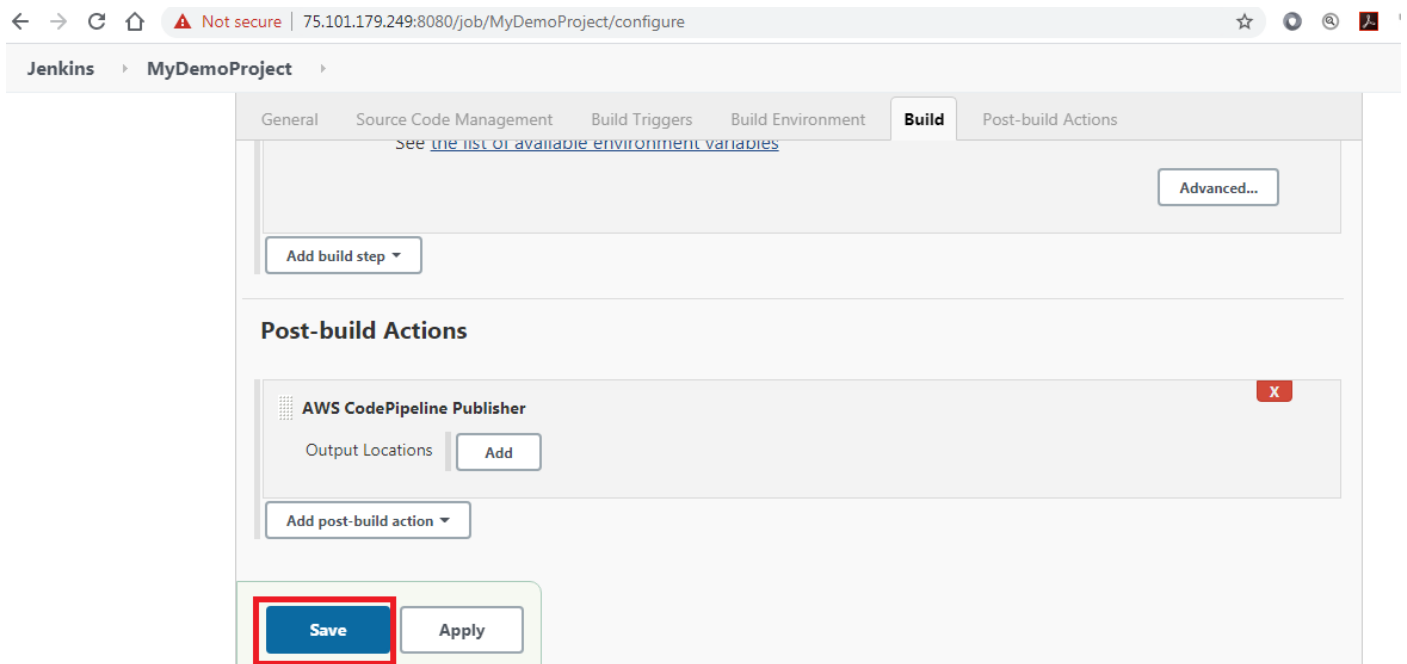
13. In **Build**, choose **Add build step**. Choose **Execute shell** (Amazon Linux, RHEL, or Ubuntu Server) **Execute batch command** (Windows Server), and then enter the following:



14. Choose **Add post-build action**, and then choose **AWS CodePipeline Publisher**.
Choose **Add**, and in **Build Output Locations**, leave the location blank. This configuration is the default. It will create a compressed file at the end of the build process.



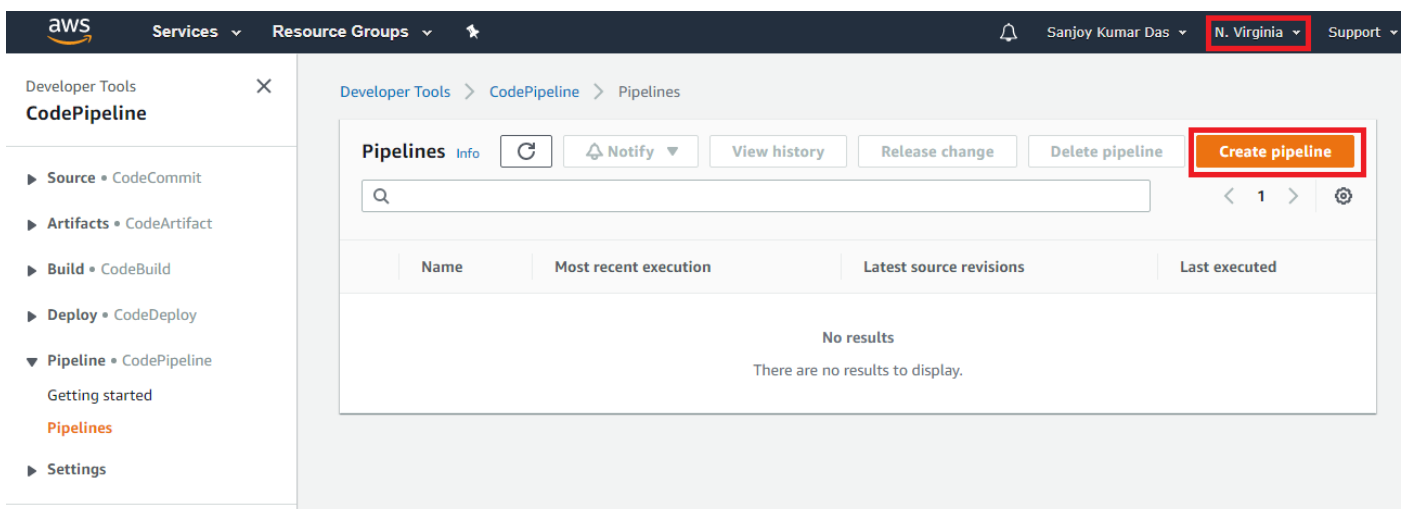
15. Choose **Save** to save your Jenkins project.



Step 2: Create a pipeline in CodePipeline

To create a CodePipeline automated release process

1. Sign in to the AWS Management Console and open the CodePipeline console at <http://console.aws.amazon.com/codesuite/codepipeline/home>.
2. If necessary, use the Region selector to change the Region to the one where your pipeline resources are located. For example, if you created resources for the previous tutorial in `us-east-1`, make sure that the Region selector is set to N. Virginia.
3. On the **Welcome** page, **Getting started** page, or the **Pipelines** page, choose **Create pipeline**.



4. On the **Step 1: Choose pipeline settings** page, in **Pipeline name**, enter the name for your pipeline.
5. In **Service role**, choose **New service role** to allow CodePipeline to create a service role in IAM.
6. Leave the settings under **Advanced settings** at their defaults, and choose **Next**.

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
MyPipeline
No more than 100 characters

Service role

☒ **New service role**
Create a service role in your account

☐ **Existing service role**
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-us-east-1-MyPipeline
Type your service role name

☒ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

► **Advanced settings**

Cancel **Next**

7. In **Step 2: Add source stage**, in **Source provider**, choose **GitHub**, and then choose **Connect to GitHub**. This will open a new browser window that will connect you to GitHub. If prompted to sign in, provide your GitHub credentials.

Add source stage Info

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
GitHub

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.
Connect to GitHub

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ **GitHub webhooks (recommended)**
Use webhooks in GitHub to automatically start my pipeline when a change occurs

☐ **AWS CodePipeline**
Use AWS CodePipeline to check periodically for changes

Cancel Previous **Next**



Authorize AWS CodePipeline (N. Virginia)



AWS CodePipeline (N. Virginia) by [aws-codesuite](#)

wants to access your sanjoy-kumar account



Repository webhooks and services

Admin access



Repositories

Public and private



Cancel

Authorize aws-codesuite

Authorizing will redirect to
<https://console.aws.amazon.com>



Not owned or operated by GitHub




Created 4 years ago



More than 1K GitHub users

github.com/login/oauth/authorize



Confirm password to continue

Password [Forgot password?](#)

Confirm password

Tip: You are entering **sudo mode**. We won't ask for your password again for a few hours.

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

aws Services Resource Groups

Step 4
Add deploy stage
Step 5
Review

GitHub

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Connected

✓ You have successfully configured the action with the provider. ✕

Repository
sanjoy-kumar/aws-codepipeline-jenkins-aws-codedeploy_linux ✕

Branch
master ✕

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☒ GitHub webhooks (recommended)
Use webhooks in GitHub to automatically start my pipeline when a change occurs

☐ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Cancel Previous **Next**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

8. In **Step 3: Add build stage**, choose **Add Jenkins**. In **Provider name**, enter the name of the action you provided in the CodePipeline Plugin for Jenkins (for example *MyJenkinsProviderName*). This name must exactly match the name in the CodePipeline Plugin for Jenkins. In **Server URL**, enter the URL of

the EC2 instance where Jenkins is installed. In **Project name**, enter the name of the project you created in Jenkins, such as *MyDemoProject*, and then choose **Next**.

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add build stage Info

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Add Jenkins

Use the template below to connect a Jenkins instance as a build provider for pipelines in your AWS account. Before you connect your Jenkins instance, you should set up the AWS CodePipeline Plugin for Jenkins and configure it for your project. [Learn more](#)

Provider name
Enter the provider name you configured in the Jenkins plugin

MyJenkinsProviderName

This name must match the name configured in the plugin.

Server URL
Enter the URL for your Jenkins server

http://75.101.179.249:8080/

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add build stage Info

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Add Jenkins

Use the template below to connect a Jenkins instance as a build provider for pipelines in your AWS account. Before you connect your Jenkins instance, you should set up the AWS CodePipeline Plugin for Jenkins and configure it for your project. [Learn more](#)

Provider name
Enter the provider name you configured in the Jenkins plugin

MyJenkinsProviderName

This name must match the name configured in the plugin.

Server URL
Enter the URL for your Jenkins server

http://75.101.179.249:8080/

Project name
Enter the name of the project you created in the Jenkins plugin

MyDemoProject

Cancel Previous Skip build stage **Next**

9. In **Step 4: Add deploy stage**, reuse the CodeDeploy application and deployment group you created in [Tutorial: Create a simple pipeline \(S3 bucket\)](#). In **Deploy provider**, choose **CodeDeploy**. In **Application name**, enter *CodePipelineDemoApplication*, or choose the refresh button, and then choose the application name from the list. In **Deployment group**, enter *CodePipelineDemoFleet*, or choose it from the list, and then choose **Next**.

aws Services ▾ Resource Groups ▾

Step 1: Choose pipeline settings

Step 2: Add source stage

Step 3: Add build stage

Step 4: **Add deploy stage**

Step 5: Review

Add deploy stage Info

Deploy - optional

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region
US East (N. Virginia)

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

CodePipelineDemoApplication

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

CodePipelineDemoFleet

Cancel Previous Skip deployment stage **Next**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy

Note

You can use your own CodeDeploy resources or create new ones, but you might incur additional costs.

10. In **Step 5: Review**, review the information, and then choose **Create pipeline**.

aws Services ▾ Resource Groups ▾

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1: Choose pipeline settings

Step 2: Add source stage

Step 3: Add build stage

Step 4: Add deployment stage

Step 5: **Review**

Review Info

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
MyPipeline

Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name
AWSCodePipelineServiceRole-us-east-1-MyPipeline

Review

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Step 2: Add source stage

Source action provider

Source action provider
ThirdParty GitHub
PollForSourceChanges
false
Repo
aws-codepipeline-jenkins-aws-codedeploy_linux
Owner
sanjoy-kumar
Branch
master

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Step 3: Add build stage

Build action provider

Build action provider
Template NewJenkins
ProviderName
MyJenkinsProviderName
Category
Build
ServerUrl
http://75.101.179.249:8080/
ProjectName
MyDemoProject

aws

Services

Resource Groups

Sanjoy Kumar Das

N. Virginia

Category
Build
ServerUrl
http://75.101.179.249:8080/
ProjectName
MyDemoProject

Step 4: Add deploy stage

Deploy action provider

Deploy action provider
AWS CodeDeploy
ApplicationName
CodePipelineDemoApplication
DeploymentGroupName
CodePipelineDemoFleet

CancelPreviousCreate pipeline

11. The pipeline automatically starts and runs the sample through the pipeline. You can view progress and success and failure messages as the pipeline builds the Haml sample to HTML and deploys it a webpage to each of the Amazon EC2 instances in the CodeDeploy deployment.

aws

Services

Resource Groups

Sanjoy Kumar DasN. VirginiaSupport

Developer ToolsCodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Success

Conratulations! The pipeline MyPipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > MyPipeline

MyPipeline

NotifyEditStop executionClone pipelineRelease change

SourceSucceeded

Pipeline execution ID: 8a55e028-ad48-46b6-a76c-67757bf5d5c7

Source

GitHub

Succeeded - Just now

9cff58a0

9cff58a0 Source: Change default test port to 80

Disable transition

aws

Services

Resource Groups

Sanjoy Kumar DasN. VirginiaSupport

Developer ToolsCodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Success

Conratulations! The pipeline MyPipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > MyPipeline

MyPipeline

NotifyEditStop executionClone pipelineRelease change

BuildIn progress

Pipeline execution ID: 8a55e028-ad48-46b6-a76c-67757bf5d5c7

Build

Custom MyJenkinsProviderName (Version: 1)

In progress - Just now

9cff58a0 Source: Change default test port to 80

Disable transition

DeployDidn't Run

aws

Services

Resource Groups

Sanjoy Kumar DasN. VirginiaSupport

Developer ToolsCodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Success

Stage Build successfully retried

Developer Tools > CodePipeline > Pipelines > MyPipeline

MyPipeline

NotifyEditStop executionClone pipelineRelease change

BuildIn progress

Pipeline execution ID: 8a55e028-ad48-46b6-a76c-67757bf5d5c7

Build

Custom MyJenkinsProviderName (Version: 1)

In progress - Just now

9cff58a0 Source: Change default test port to 80

Disable transition

DeployDidn't Run

Deploy

AWS CodeDeploy

Didn't Run

No executions yet

Step 3: Add another stage to your pipeline

Now you will add a test stage and then a test action to that stage that uses the Jenkins test included in the sample to determine whether the webpage has any content. This test is for demonstration purposes only.

Add a test stage to your pipeline

Topics

- [Look up the IP address of an instance](#)
- [Create a Jenkins project for testing the deployment](#)
- [Create a fourth stage](#)

Look up the IP address of an instance

To verify the IP address of an instance where you deployed your code

1. After **Succeeded** is displayed for the pipeline status, in the status area for the Staging stage, choose **Details**.
2. In the **Deployment Details** section, in **Instance ID**, choose the instance ID of one of the successfully deployed instances.
3. Copy the IP address of the instance (for example, **172.31.13.175**). You will use this IP address in your Jenkins test.

Create a Jenkins project for testing the deployment

To create the Jenkins project

1. On the instance where you installed Jenkins, open Jenkins and from the main page, choose **New Item**.
2. In **Item Name**, enter a name for the Jenkins project (for example, **MyTestProject**). Choose **Freestyle project**, and then choose **OK**.
3. On the configuration page for the project, select the **Execute concurrent builds if necessary** check box. In **Source Code Management**, choose **AWS CodePipeline**. If you have installed Jenkins on an EC2 instance and configured the AWS CLI with the profile for the IAM user you created for integration between CodePipeline and Jenkins, leave all the other fields empty.

← → ↻ 🏠 ⓘ Not secure | 75.101.179.249:8080/job/MyTestProject/configure

Jenkins > MyTestProject >

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

- ☐ GitHub project
- ☐ This build requires lockable resources
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Disable this project ?
- ☒ Execute concurrent builds if necessary ?
- ☐ Quiet period ?
- ☐ Retry Count ?
- ☐ Block build when upstream project is building ?
- ☐ Block build when downstream project is building ?
- ☐ Use custom workspace ?

Jenkins > MyTestProject >

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

Source Code Management

☐ None

☒ AWS CodePipeline

AWS Config

AWS Region

Proxy Host ?

Proxy Port ?

Credentials

If these keys are left blank, the plugin will attempt to use credentials from the default provider chain. That is: Environment Variables, Java System properties, credentials profile file, and finally, EC2 Instance profile.

AWS Access Key ?

AWS Secret Key ?

☒

Important

If you are configuring a Jenkins project and it is not installed on an Amazon EC2 instance, or it is installed on an EC2 instance that is running a Windows operating system, complete the fields as required by your proxy host and port settings, and provide the credentials of the IAM user you configured for integration between Jenkins and CodePipeline.

4. Choose **Advanced**, and in **Category**, choose **Test**.
5. In **Provider**, enter the same name you used for the build project (for example, *MyJenkinsProviderName*). You will use this name when you add the test action to your pipeline later in this tutorial.

← → ↻ 🏠 ⚠ Not secure | 75.101.179.249:8080/job/MyTestProject/configure

Jenkins > MyTestProject >

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

Clear workspace before copying ☒

CodePipeline Action Type

This value must match the Category field that is on the Custom Action in your corresponding Pipeline.

Category Test ?

This value must match the Provider field that is on the Custom Action in your corresponding Pipeline.

Provider MyJenkinsProviderName ?

This value must match the Version field that is on the Custom Action in your corresponding Pipeline.

Version 1 ?

☐ Git ?

☐ Subversion ?

Build Triggers

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

6. In **Build Triggers**, clear any check boxes, and then select **Poll SCM**. In **Schedule**, enter five asterisks separated by spaces, as follows:

```
* * * * *
```

This polls CodePipeline every minute.

← → ↻ 🏠 ⚠ Not secure | 75.101.179.249:8080/job/MyTestProject/configure

Jenkins > MyTestProject >

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

Build Triggers

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

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☒ **Poll SCM** ?

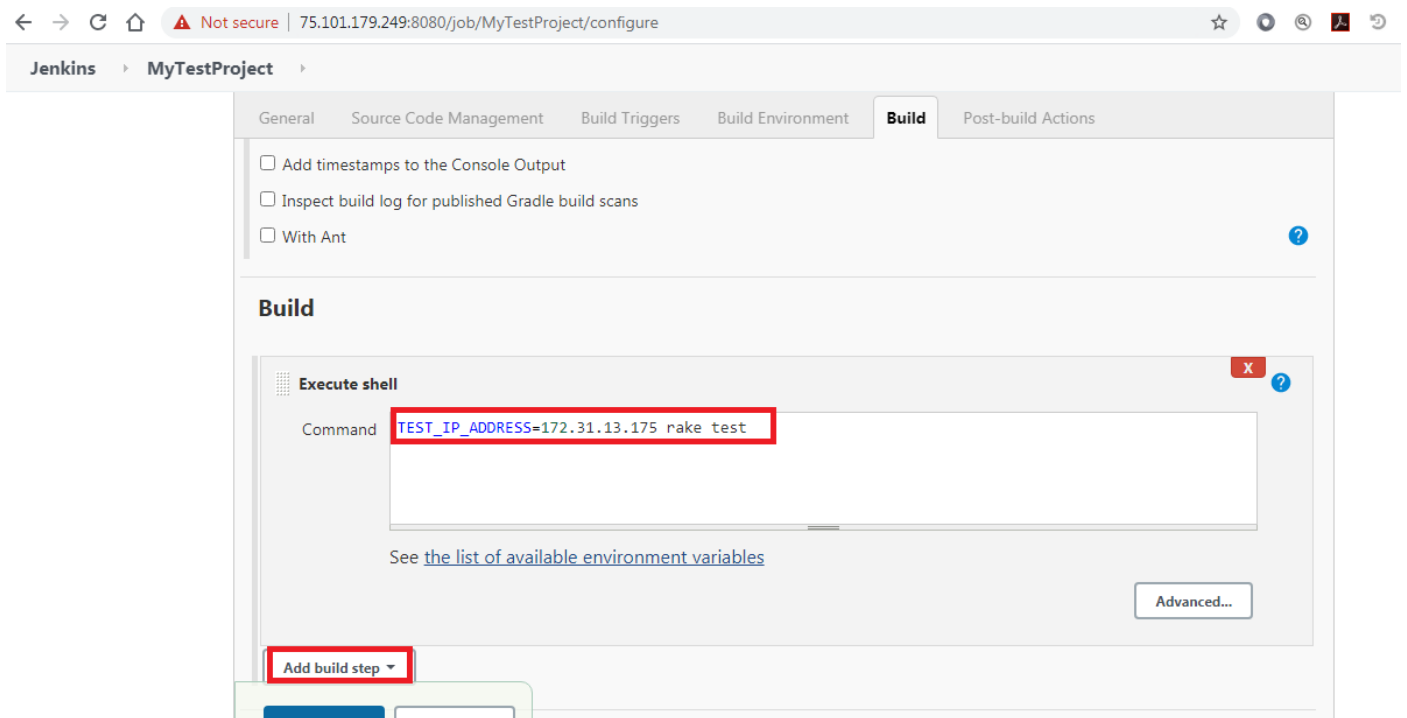
Schedule ***** ?

⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H * * * * *" to poll once per hour
Would last have run at Thursday, August 13, 2020 5:01:16 PM UTC; would next run at Thursday, August 13, 2020 5:01:16 PM UTC.

☐ Ignore post-commit hooks ?

7. In **Build**, choose **Add build step**. If you are deploying to Amazon Linux, RHEL, or Ubuntu Server instances, choose **Execute shell**. Then enter the following, where the IP address is the address of the EC2 instance you copied earlier:

```
TEST_IP_ADDRESS= 172.31.13.175 rake test
```



If you are deploying to Windows Server instances, choose **Execute batch command**, and then enter the following, where the IP address is the address of the EC2 instance you copied earlier:

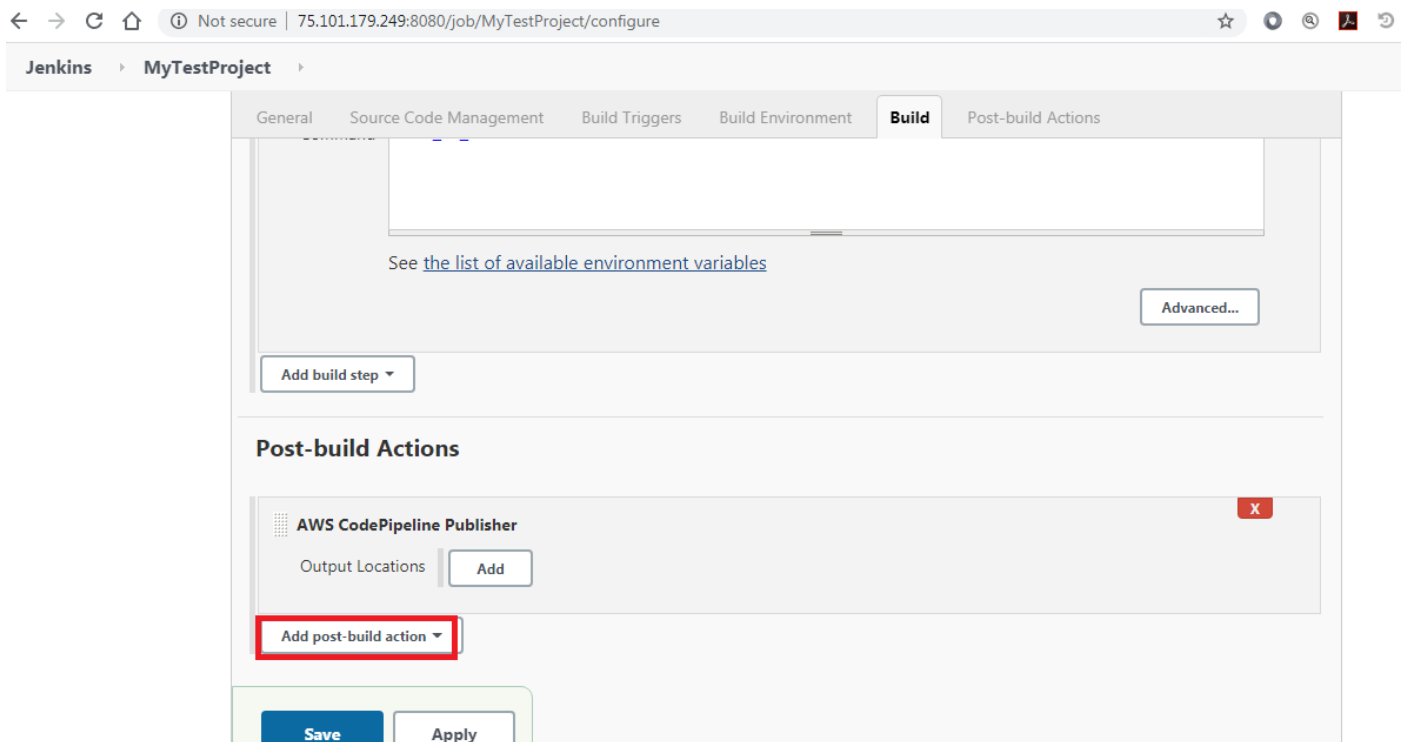
```
set TEST_IP_ADDRESS=172.31.13.175 rake test
```

Note

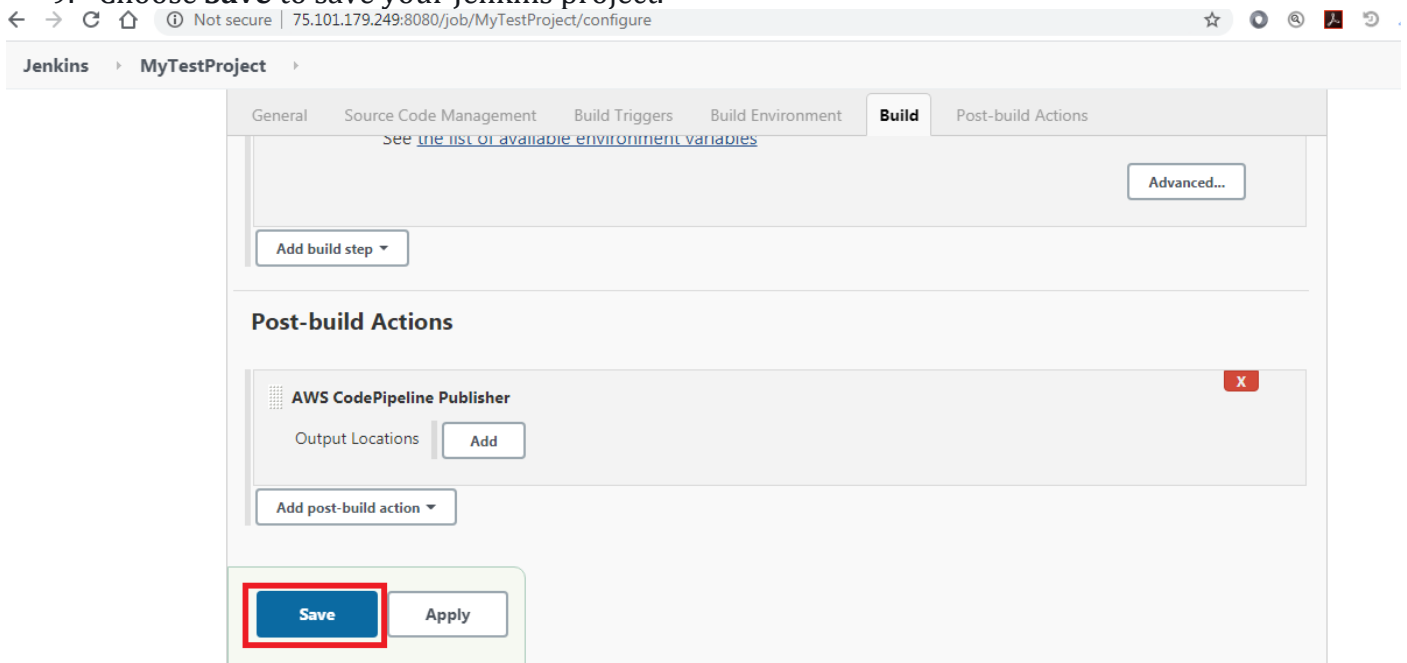
The test assumes a default port of 80. If you want to specify a different port, add a test port statement, as follows:

```
TEST_IP_ADDRESS= 172.31.13.175 TEST_PORT=8000 rake test
```

8. Choose **Add post-build action**, and then choose **AWS CodePipeline Publisher**. Do not choose **Add**.



9. Choose **Save** to save your Jenkins project.



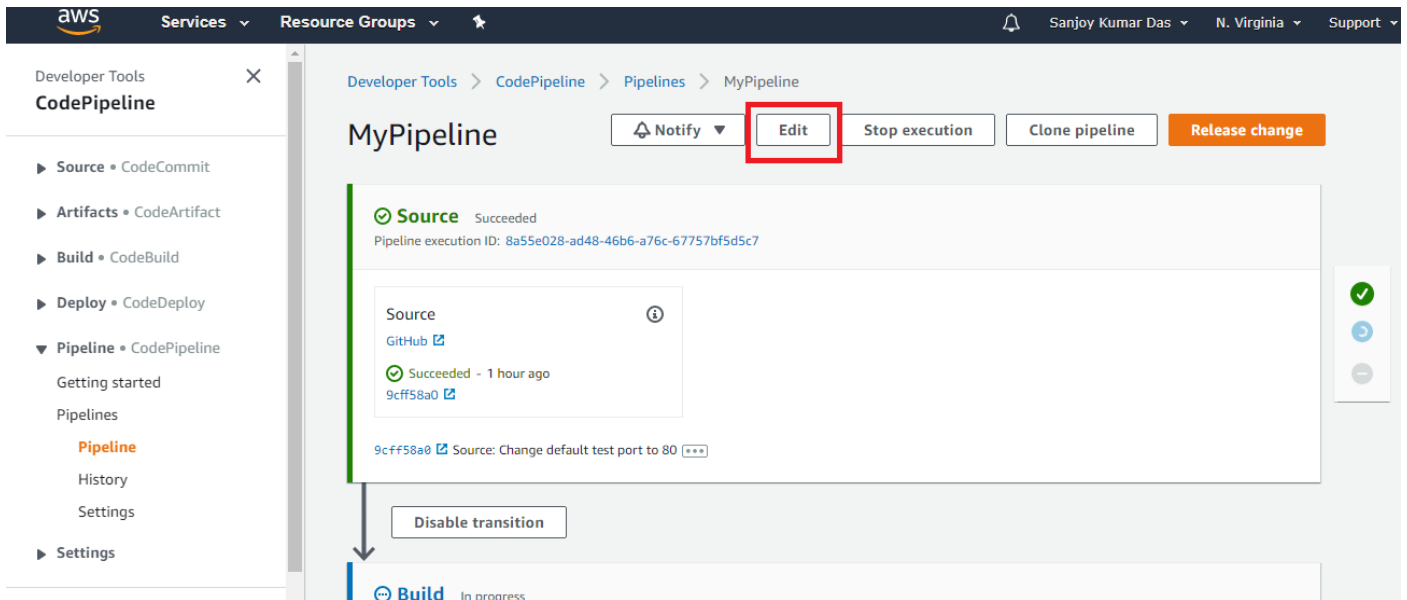
Page generated:

DECT ADI

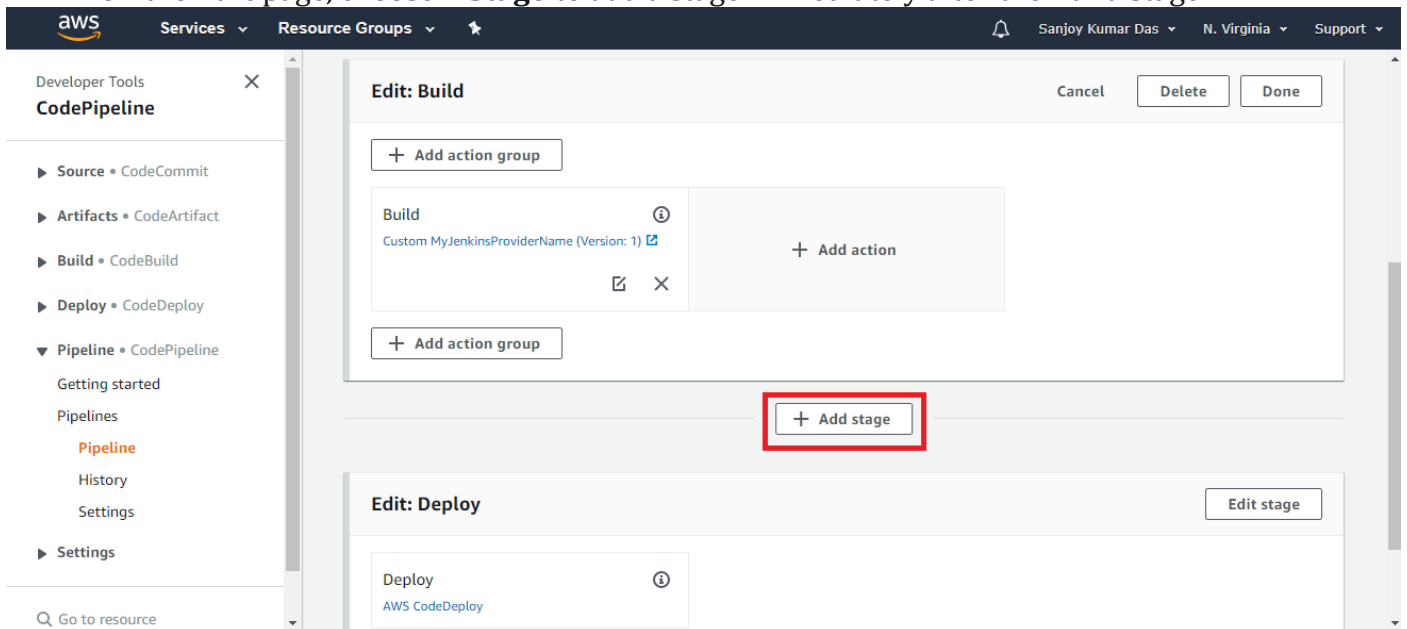
Create a fourth stage

To add a stage to your pipeline that includes the Jenkins test action

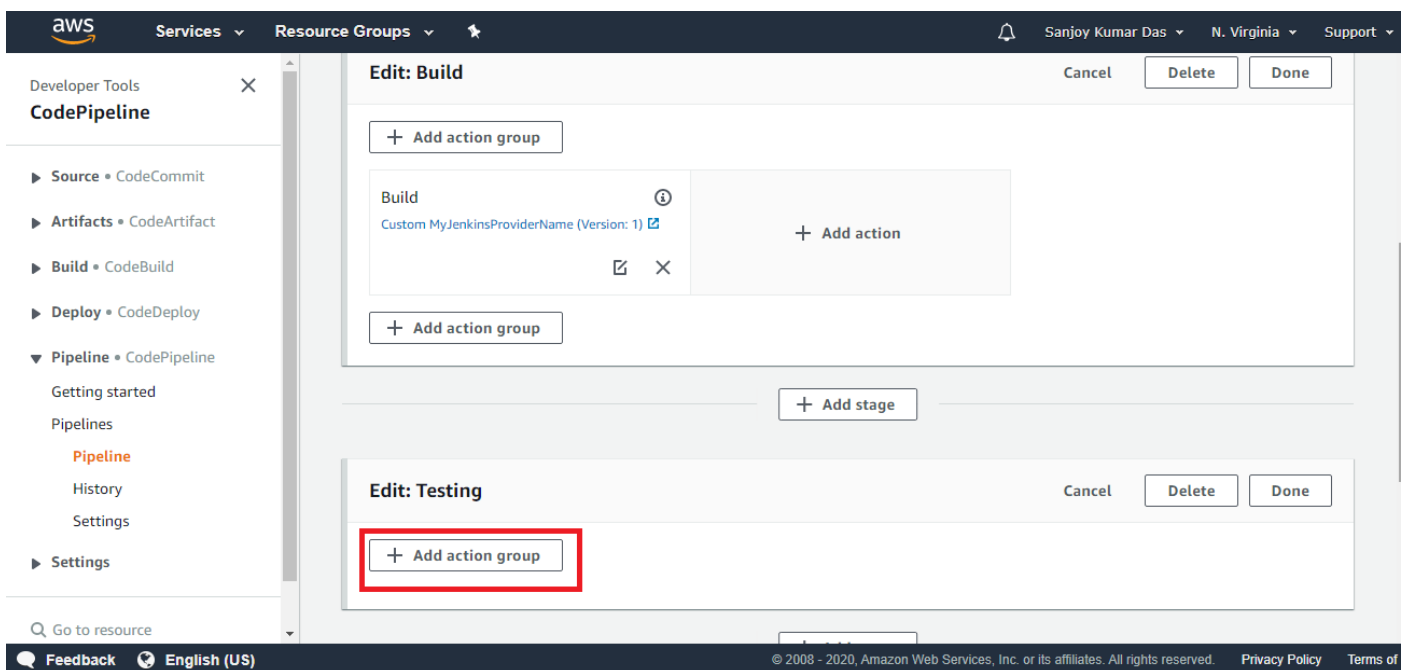
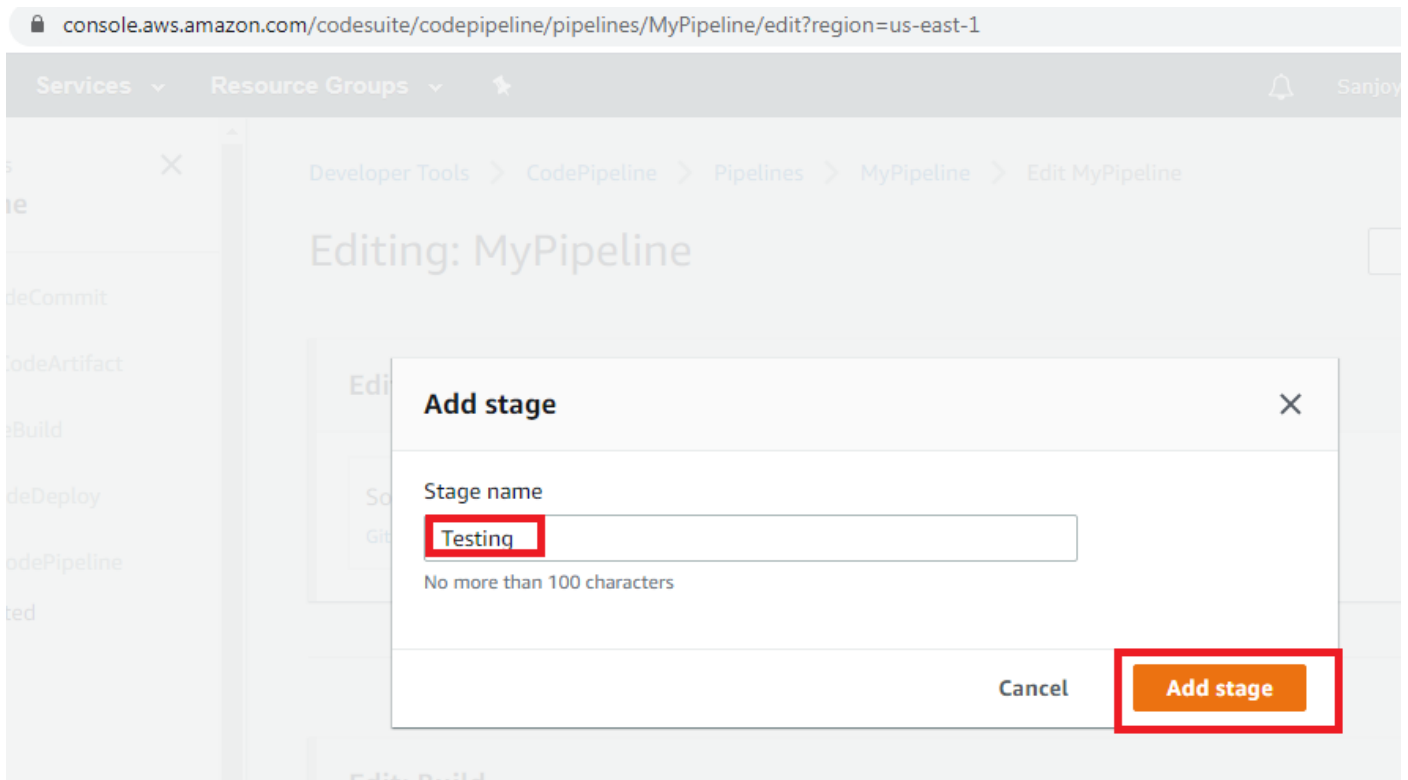
1. Sign in to the AWS Management Console and open the CodePipeline console at <http://console.aws.amazon.com/codesuite/codepipeline/home>.
2. In **Name**, choose the name of the pipeline you created, MyPipeline.
3. On the pipeline details page, choose **Edit**.



4. On the **Edit** page, choose **+ Stage** to add a stage immediately after the Build stage.



5. In the name field for the new stage, enter a name (for example, **Testing**), and then choose **+ Add action group**.



6. In **Action name**, enter *MyJenkinsTest-Action*. In **Test provider**, choose the provider name you specified in Jenkins (for example, *MyJenkinsProviderName*). In **Project name**, enter the name of the project you created in Jenkins (for example, *MyTestProject*). In **Input artifacts**, choose the artifact from the Jenkins build whose default name is *BuildArtifact*, and then choose **Done**.

→ ↻ ↩ ⌨ console.aws.amazon.com/codesuite/codepipeline/pipelines/MyPipeline/edit?region=us-east-1 ☆ ⚙ ⌂ 🔍 📄 📌

aw

Develop

CodeP

Source

Artif

Build

Depl

Pipe

Gett

Pipe

Pr

H

S

Sett

Q

Go to resource

Edit action

Action name
Choose a name for your action

MyJenkinsTest-Action

No more than 100 characters

Action provider

Custom MyJenkinsProviderName (Version: 1)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

BuildArtifact

Add

No more than 100 characters

ProjectName

MyTestProject

Variable namespace - optional
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

Output artifacts
Choose a name for the output of this action.

BuildArtifact

Add

No more than 100 characters

Variable namespace - optional
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

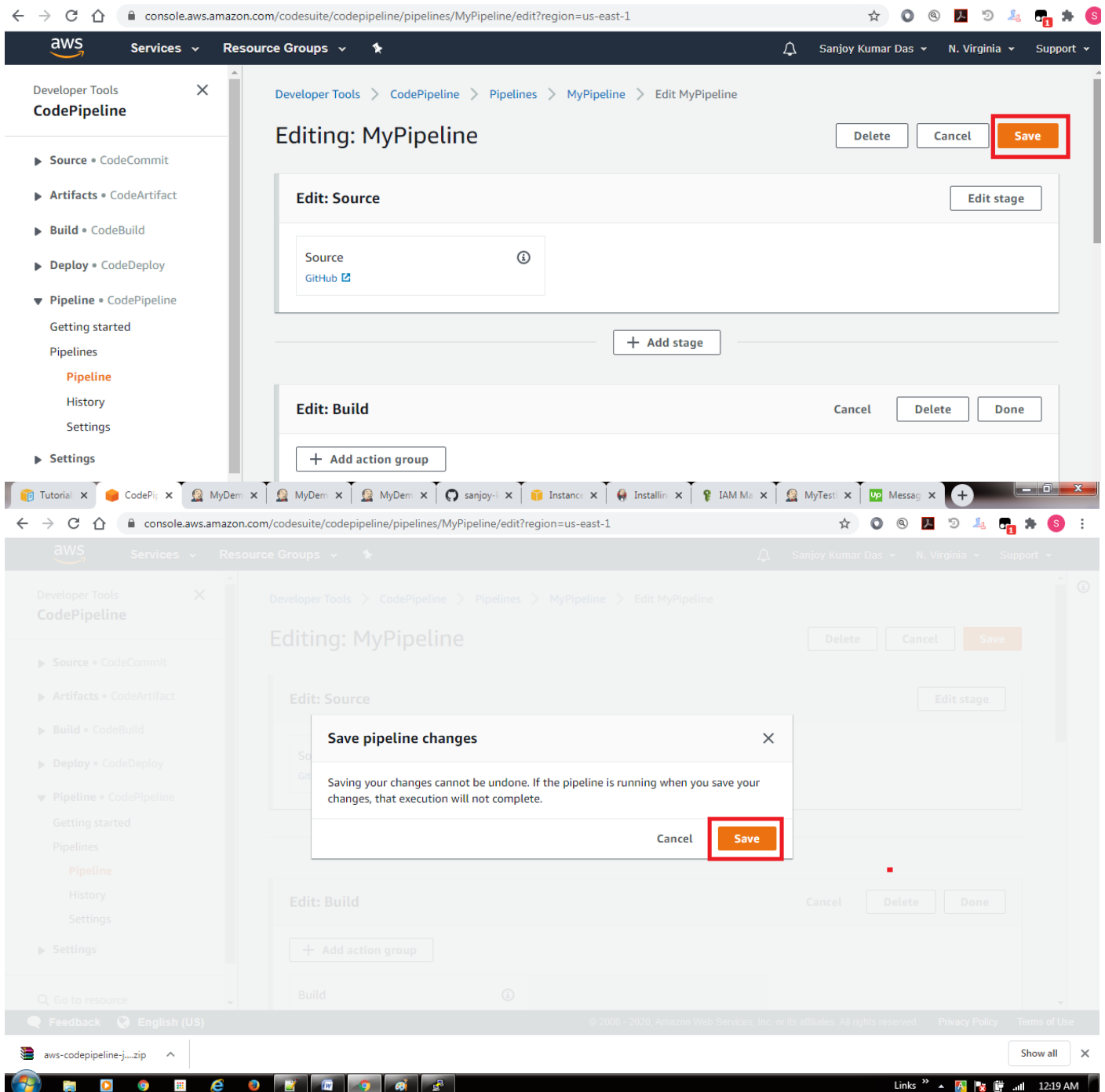
Output artifacts
Choose a name for the output of this action.

Add

No more than 100 characters

Cancel Done

- On the **Edit** page, choose **Save pipeline changes**. In the **Save pipeline changes** dialog box, choose **Save and continue**.



8. Although the new stage has been added to your pipeline, a status of **No executions yet** is displayed for that stage because no changes have triggered another run of the pipeline. To run the sample through the revised pipeline, on the pipeline details page, choose **Release change**.

← → ↻ 🏠 console.aws.amazon.com/codesuite/codepipeline/pipelines/MyPipeline/view?region=us-east-1 ☆ 🔍 📄 🔄 👤 ⚙️

aws Services ▾ Resource Groups ▾ ⭐

Developer Tools **CodePipeline** X

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline**
 - History
 - Settings

Success Pipeline was saved successfully. X

Developer Tools > CodePipeline > Pipelines > MyPipeline

MyPipeline 🔔 Notify ▾ Edit Stop execution Clone pipeline Release change

Source Succeeded Pipeline execution ID: 8a55e028-ad48-46b6-a76c-67757bf5d5c7

Source ⓘ

GitHub [🔗](#)

✔ Succeeded - 1 hour ago [9cff58a0 🔗](#)

[9cff58a0 🔗](#) Source: Change default test port to 80 ***

← → ↻ 🏠 console.aws.amazon.com/codesuite/codepipeline/pipelines/MyPipeline/view?region=us-east-1 ☆ 🔍 📄 🔄 👤 ⚙️

aws Services ▾ Resource Groups ▾ ⭐

Developer Tools **CodePipeline** X

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline**
 - History
 - Settings

Success Pipeline was saved successfully. X

Success The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view. X

Developer Tools > CodePipeline > Pipelines > MyPipeline

MyPipeline 🔔 Notify ▾ Edit Stop execution Clone pipeline Release change

Source Succeeded Pipeline execution ID: 28210dd0-88b9-4f3e-a59c-061db406a87d

Source ⓘ

GitHub [🔗](#)

✔ Succeeded - Just now [9cff58a0 🔗](#)