Yesterday, we have create a java file Munish.java and push that to git repository.

Then that git url is provided to the Jenkins.

In Jenkins, in Build configuration, we provided the command to run that java program.

We run a spring boot project using Jenkins via git hub url.

1. Create a folder and sync with git repository
2. Create a Jenkins new item as git project and provide url, branch
3. Build phase

mvn spring-boot:run

1. Build now

AWS CLI

We need to install aws cli in our system.

# **Getting started with Amazon SNS**

## Step 1: Create a topic

1. Sign in to the [Amazon SNS console](https://console.aws.amazon.com/sns/home).
2. In the left navigation pane, choose **Topics**.
3. On the **Topics** page, choose **Create topic**.
4. By default, the console creates a FIFO topic. Choose **Standard**.
5. In the **Details** section, enter a **Name** for the topic, such as *MyTopic*.
6. Scroll to the end of the form and choose **Create topic**.

The console opens the new topic's **Details** page.

## Step 2: Create a subscription to the topic

1. In the left navigation pane, choose **Subscriptions**.
2. On the **Subscriptions** page, choose **Create subscription**.
3. On the **Create subscription** page, choose the **Topic ARN** field to see a list of the topics in your AWS account.
4. Choose the topic that you created in the previous step.
5. For **Protocol**, choose **Email**.
6. For **Endpoint**, enter an email address that can receive notifications.
7. Choose **Create subscription**.

The console opens the new subscription's **Details** page.

1. Check your email inbox and choose **Confirm subscription** in the email from AWS Notifications. The sender ID is usually "no-reply@sns.amazonaws.com".
2. Amazon SNS opens your web browser and displays a subscription confirmation with your subscription ID.

## Step 3: Publish a message to the topic

1. In the left navigation pane, choose **Topics**.
2. On the **Topics** page, choose the topic that you created earlier, and then choose **Publish message**.

The console opens the **Publish message to topic** page.

1. (Optional) In the **Message details** section, enter a **Subject**, such as:

Hello from Amazon SNS!

1. In the **Message body** section, choose **Identical payload for all delivery protocols**, and then enter a message body, such as:

Publishing a message to an SNS topic.

1. Choose **Publish message**.

The message is published to the topic, and the console opens the topic's **Details** page.

1. Check your email inbox and verify that you received an email from Amazon SNS with the published message.

## Step 4: Delete the subscription and topic

1. On the navigation panel, choose **Subscriptions**.
2. On the **Subscriptions** page, choose a confirmed subscription and then choose **Delete**.

###### Note

You can't delete a pending confirmation. After 3 days, Amazon SNS deletes it automatically.

1. In the **Delete subscription** dialog box, choose **Delete**.

The subscription is deleted.

1. On the navigation panel, choose **Topics**.
2. On the **Topics** page, choose a topic and then choose **Delete**.

###### Important

When you delete a topic, you also delete all subscriptions to the topic.

1. On the **Delete topic *MyTopic*** dialog box, enter delete me and then choose **Delete**.

The topic is deleted.

<https://codezup.com/aws-simple-notification-servicesns-using-spring-boot/>

note: Application.properties exclude 3 auto configurations

check out project in

<https://github.com/jagasia/ust1/tree/jag/advanced/26-jul-sns>