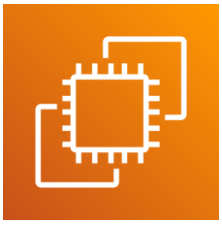
Ajay Wankhade

Create A Simple Pipeline

We Are used this aws services to create pipeline.

S3



EC2

Code Deploy

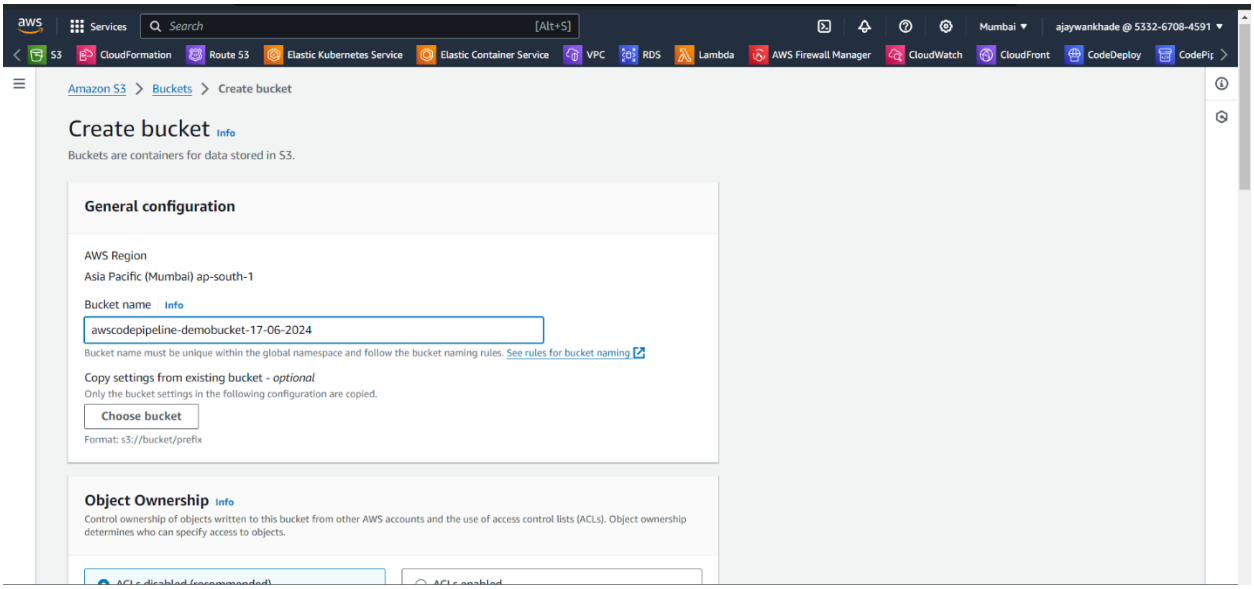
Code Pipeline

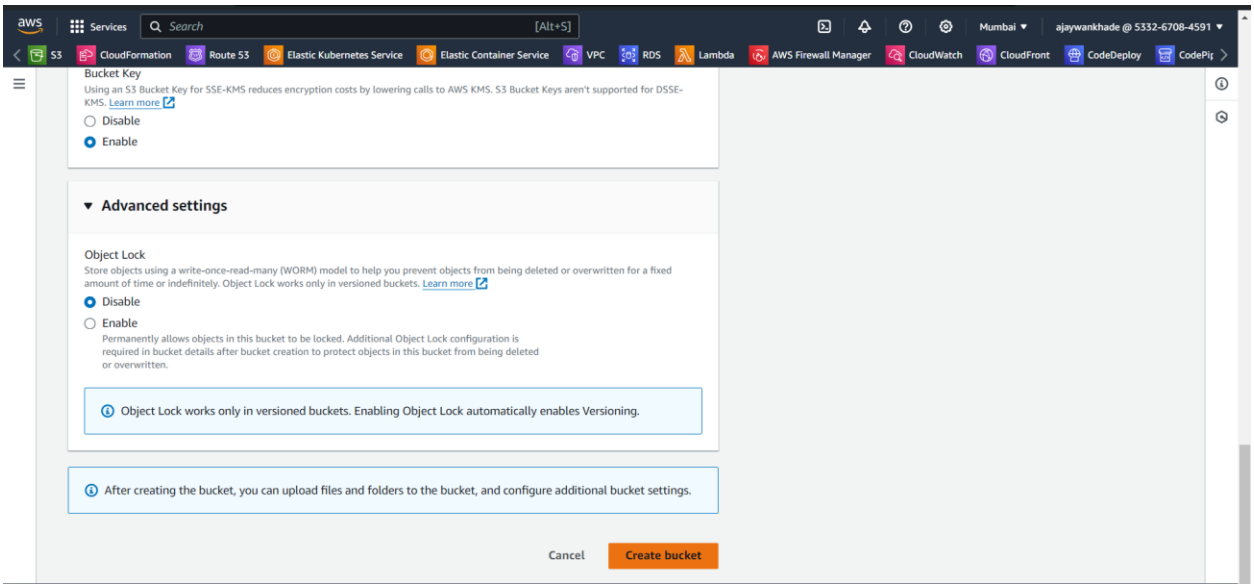
Step 1: Create an S3 bucket for your application.

You can store your source files or applications in any versioned location. In this tutorial, you create an S3 bucket for the sample application files and enable versioning on that bucket. After you have enabled versioning, you copy the sample applications to that bucket.

To create an S3 bucket:

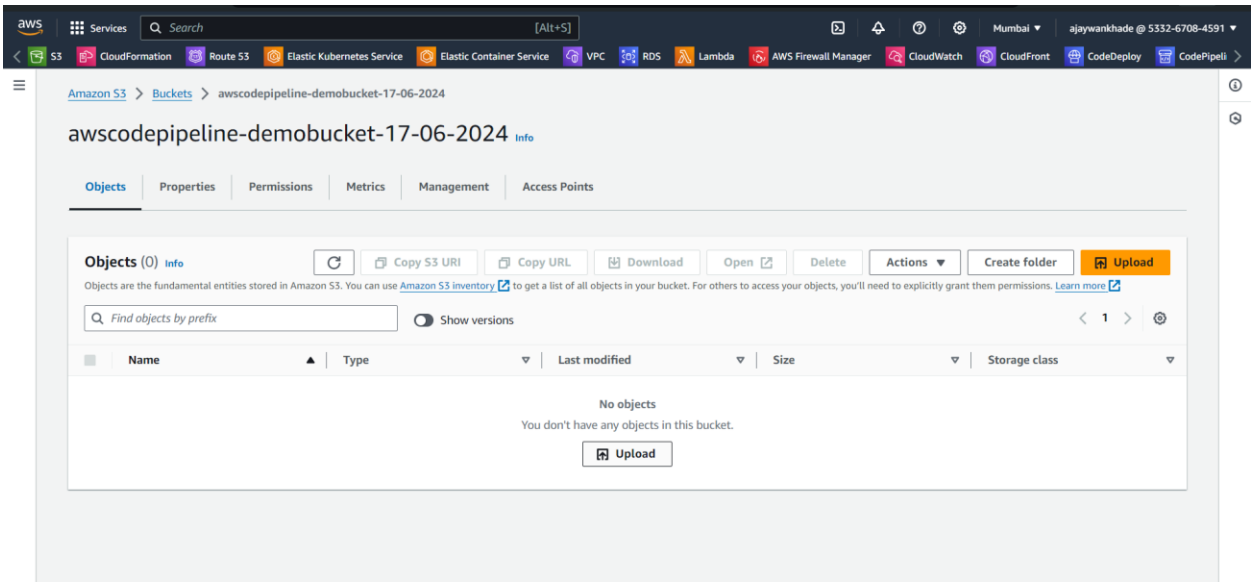
1. Choose **Create bucket**.
2. In **Bucket name**, enter a name for your bucket (for example, `awscodepipeline-demobucket.`)

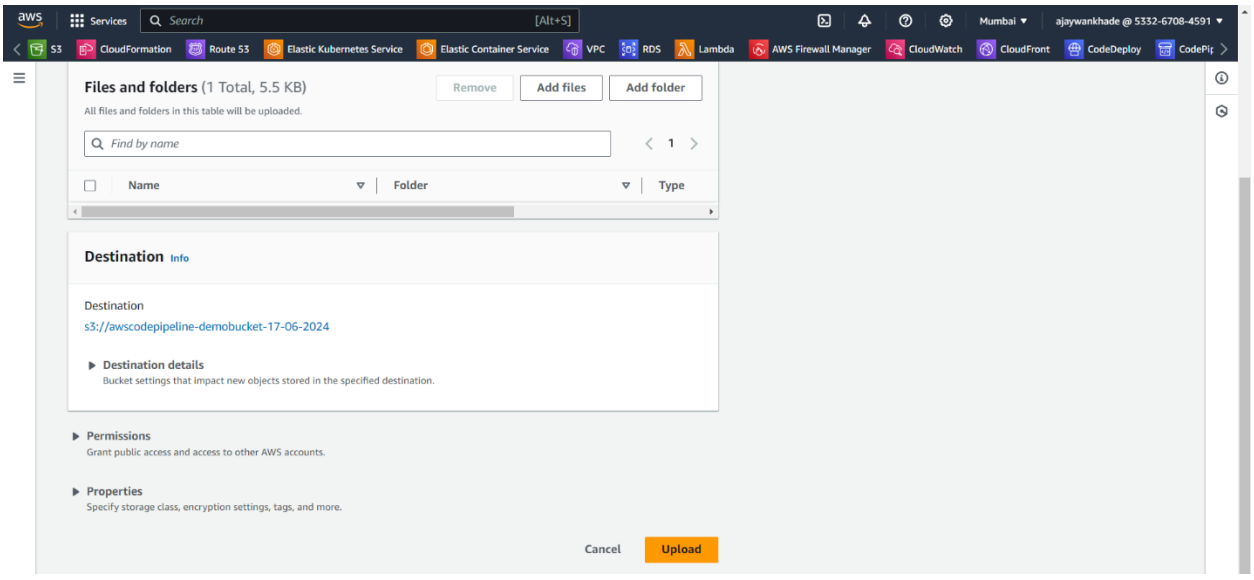




3. Next, download a sample and save it into a folder or directory on your local computer.
4. https://docs.aws.amazon.com/codepipeline/latest/userguide/samples/SampleApp_Windows.zip

5. Download the compressed (zipped) file. Do not unzip the file.
6. In the Amazon S3 console, for your bucket, upload the file:
7. Choose Upload.
8. Drag and drop the file or choose Add files and browse for the file.
9. Choose Upload.





Step 2: Create Amazon EC2 Windows instances and install the Code Deploy agent.

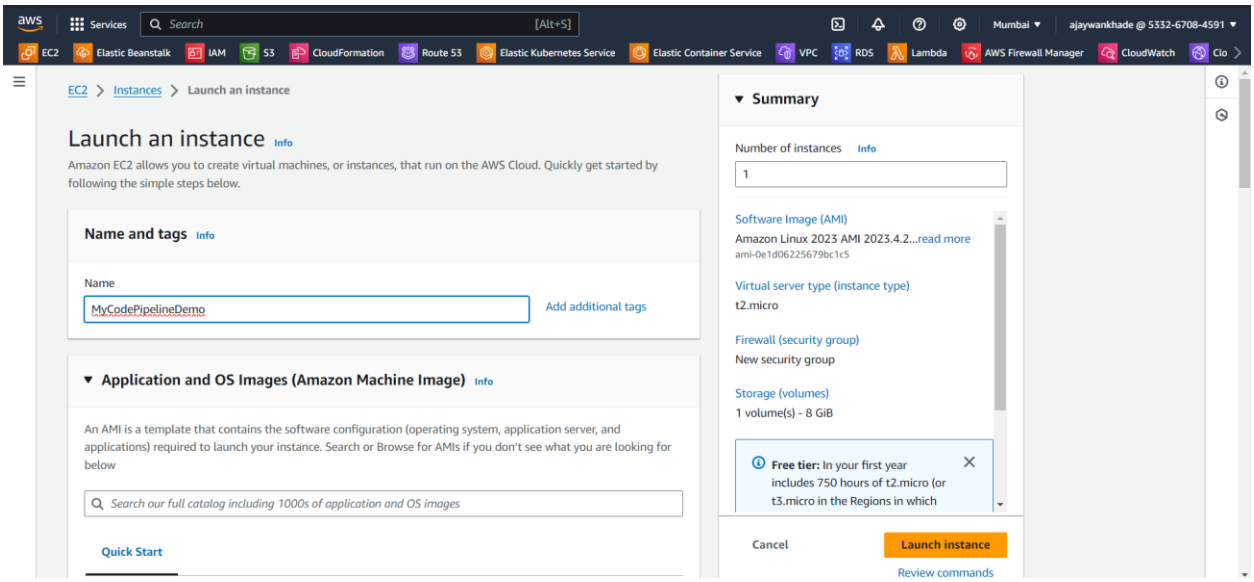
In this step, you create the Windows Server Amazon EC2 instances to which you will deploy a sample application. As part of this process, you create an instance role with policies that allow install and management of the Code Deploy agent on the instances. The Code Deploy agent is a software package that enables an instance to be used in Code Deploy deployments. You also attach policies that allow the instance to fetch files that the Code Deploy agent uses to deploy your application and to allow the instance to be managed by SSM.

To create an instance role:

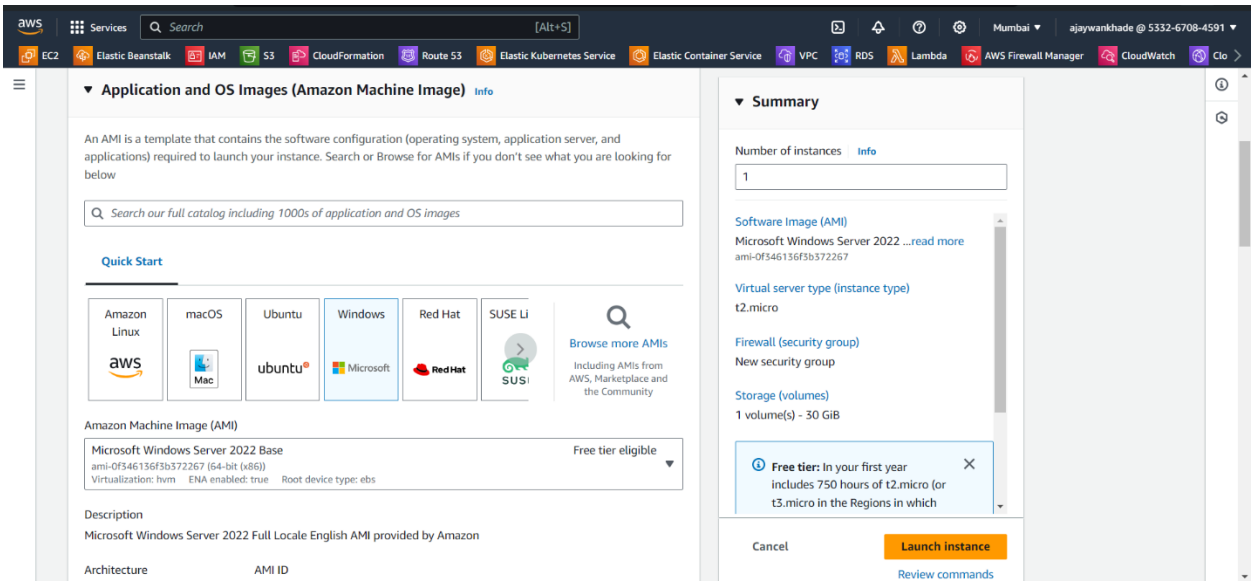
1. Open the IAM console at https://console.aws.amazon.com/iam/).
2. From the console dashboard, choose **Roles**.
3. Choose **Create role**.
4. Under **Select type of trusted entity**, select **AWS service**. Under **Choose a use case**, select **EC2**, and then choose **Next: Permissions**.
5. Search for and select the policy named `AmazonEC2RoleforAWSCodeDeploy`.
6. Search for and select the policy named `AmazonSSMManagedInstanceCore`. Choose **Next: Tags**.
7. Choose **Next: Review**. Enter a name for the role (for example, `EC2InstanceRole`).
8. Choose **Create role**.

To launch instances:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Under **Name and tags**, in **Name**, enter `MyCodePipelineDemo`. This assigns the instances a tag **Key** of `Name` and a tag **Value** of `MyCodePipelineDemo`. Later, you create a Code Deploy application that deploys the sample application to the instances. Code Deploy selects instances to deploy based on the tags

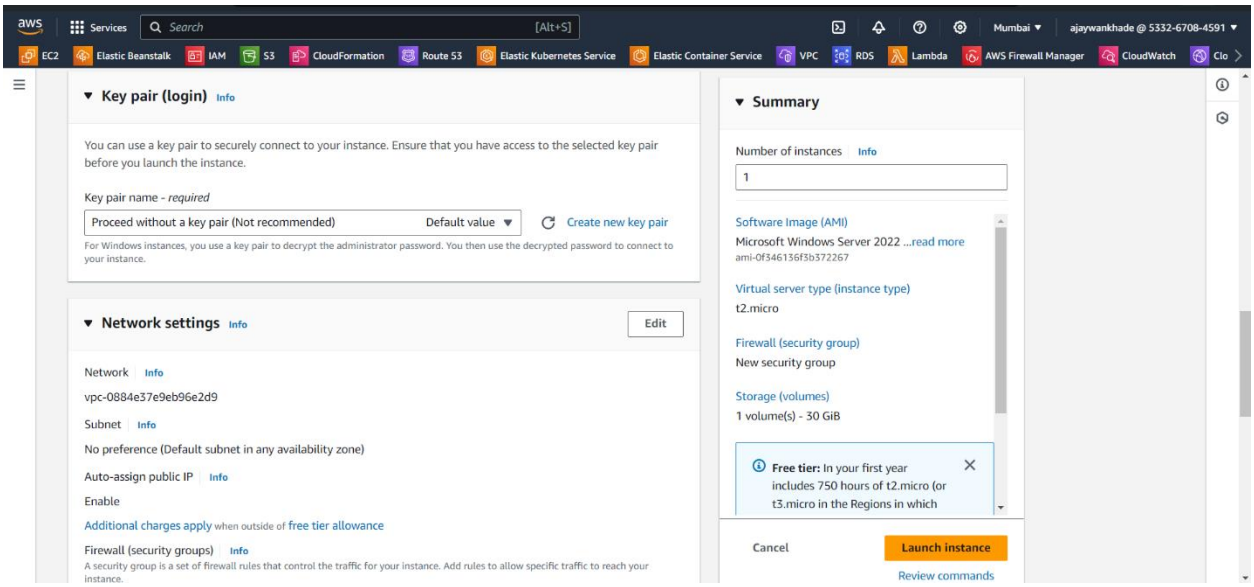


3. Under **Application and OS Images (Amazon Machine Image)**, choose the **Windows** option. (This AMI is described as the **Microsoft Windows Server 2019 Base** and is labeled "Free tier eligible" and can be found under **Quick Start**.)
4. Under **Instance type**, choose the free tier eligible `t2. micro` type as the hardware configuration for your instance.



5. Under **Key pair (login)**, choose a key pair or create one.

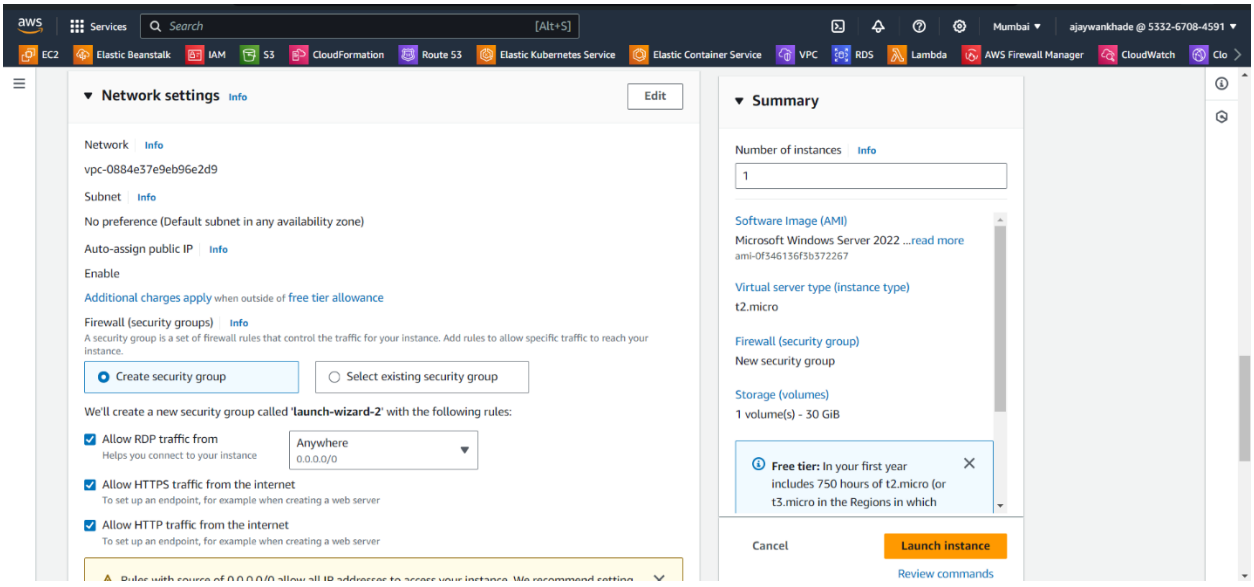
   You can also choose **Proceed without a key pair**.



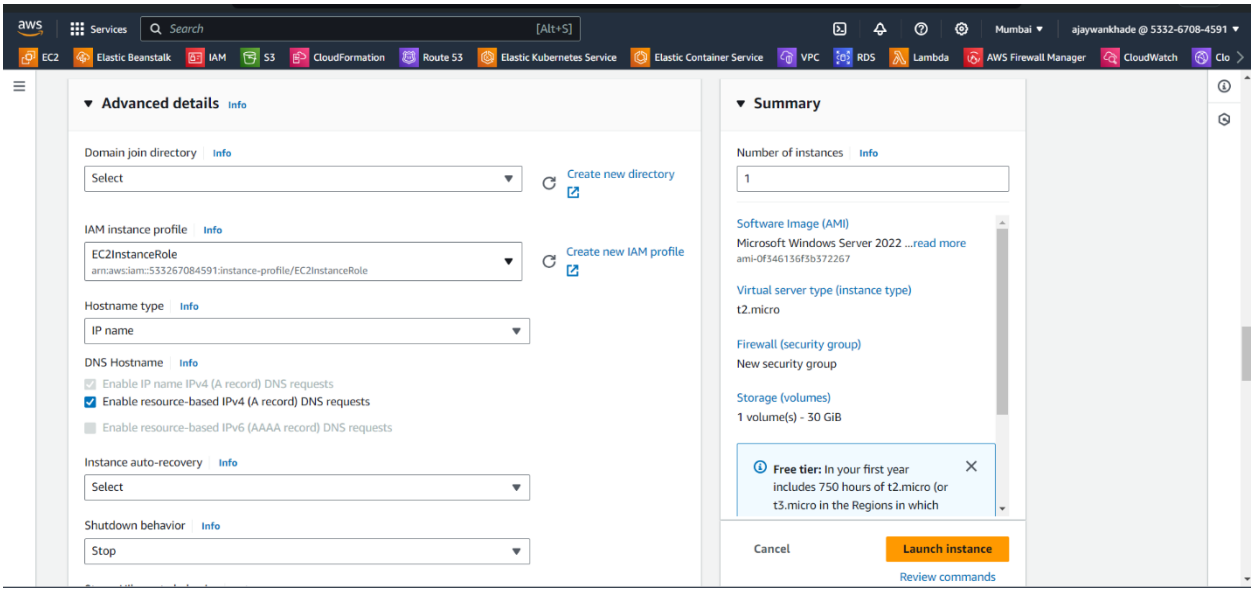
6. Under **Network settings**, do the following:

   In **Auto-assign Public IP**, make sure the status is **Enable**.

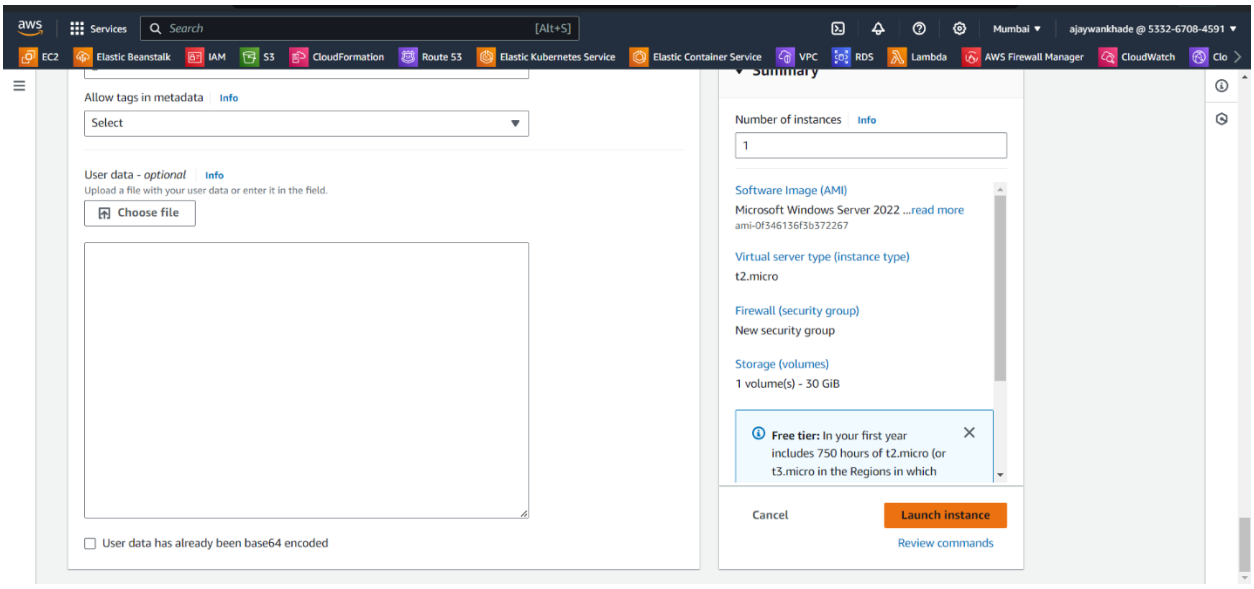
   1. Next to **Assign a security group**, choose **Create a new security group**.
   2. In the row for **SSH**, under **Source type**, choose **My IP**.
   3. Choose **Add security group**, choose **HTTP**, and then under **Source type**, choose **My IP**.



7. Expand **Advanced details**. In **IAM instance profile**, choose the IAM role you created in the previous procedure (for example, `EC2InstanceRole`).



Choose **Launch instance**.



Step 3: Create an application in Code Deploy.

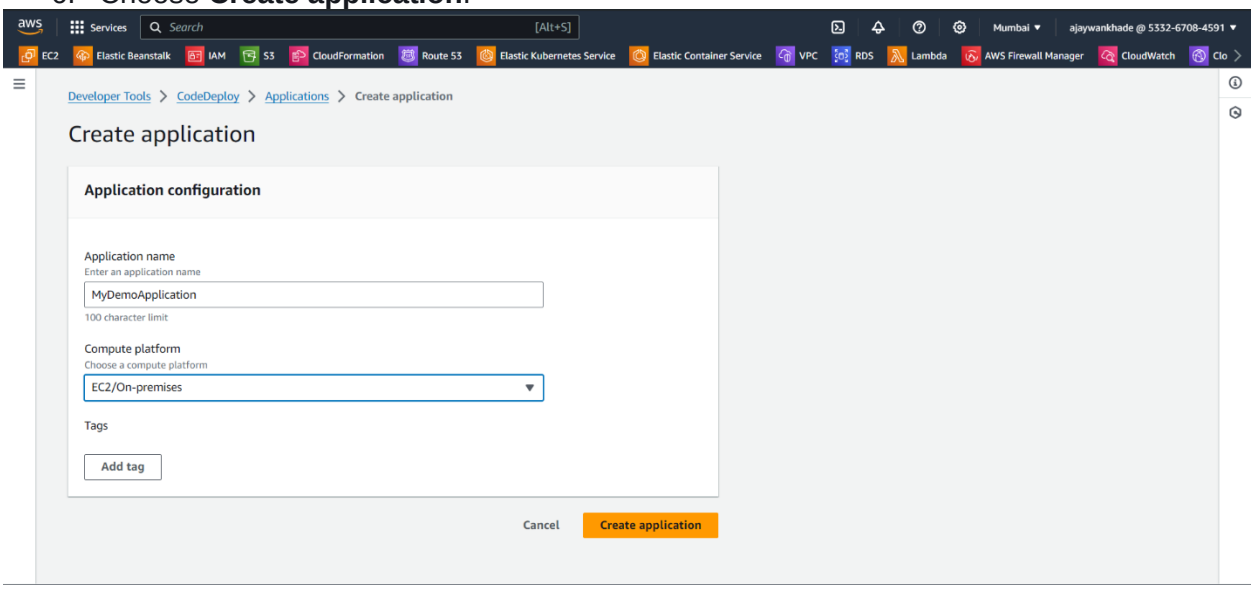
In Code Deploy, an application is an identifier, in the form of a name, for the code you want to deploy. Code Deploy uses this name to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment. You select the name of the Code Deploy application you create in this step when you create your pipeline later in this tutorial.

To create a Code, Deploy service role:

1. Choose **Create role**.
2. Under **Select trusted entity**, choose **AWS service**. Under **Use case**, choose **Code Deploy**. Choose **Code Deploy** from the options listed. Choose **Next**.
   The `AWSCodeDeployRole` managed policy is already attached to the role.
3. Choose **Next**.
4. Enter a name for the role (for example, `CodeDeployRole`), and then choose **Create role**.

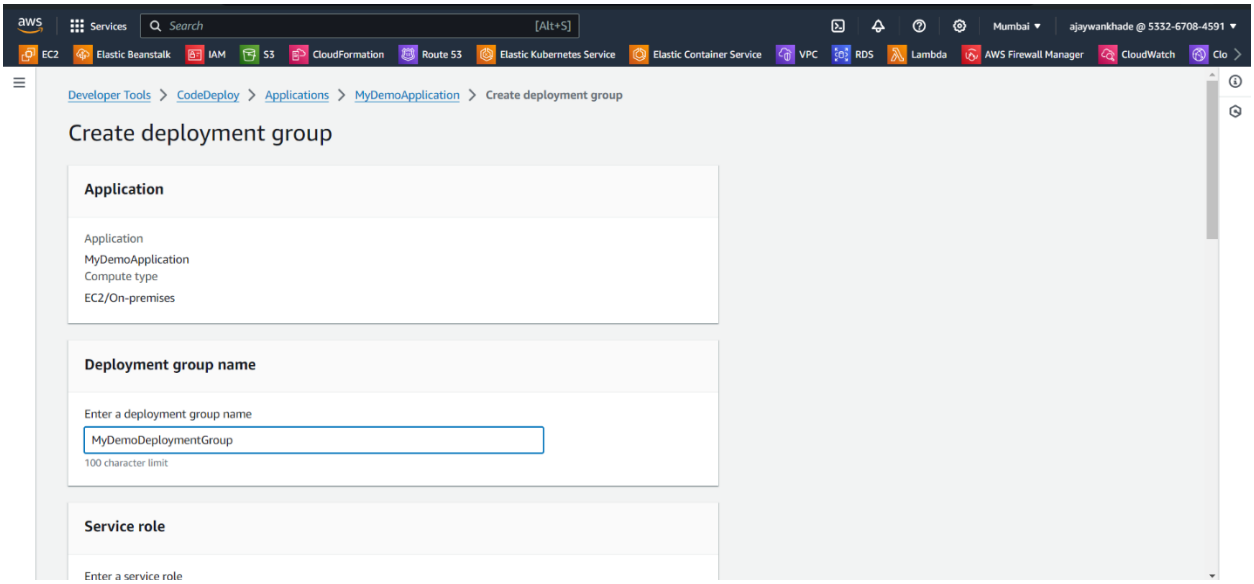
To create an application in Code Deploy

1. Open the Code Deploy console at https://console.aws.amazon.com/codedeploy.
2. If the **Applications** page does not appear, on the AWS Code Deploy menu, choose **Applications**.
3. Choose **Create application**.
4. In **Application name**, enter `MyDemoApplication`.
5. In **Compute Platform**, choose **EC2/On-premises**.
6. Choose **Create application**.

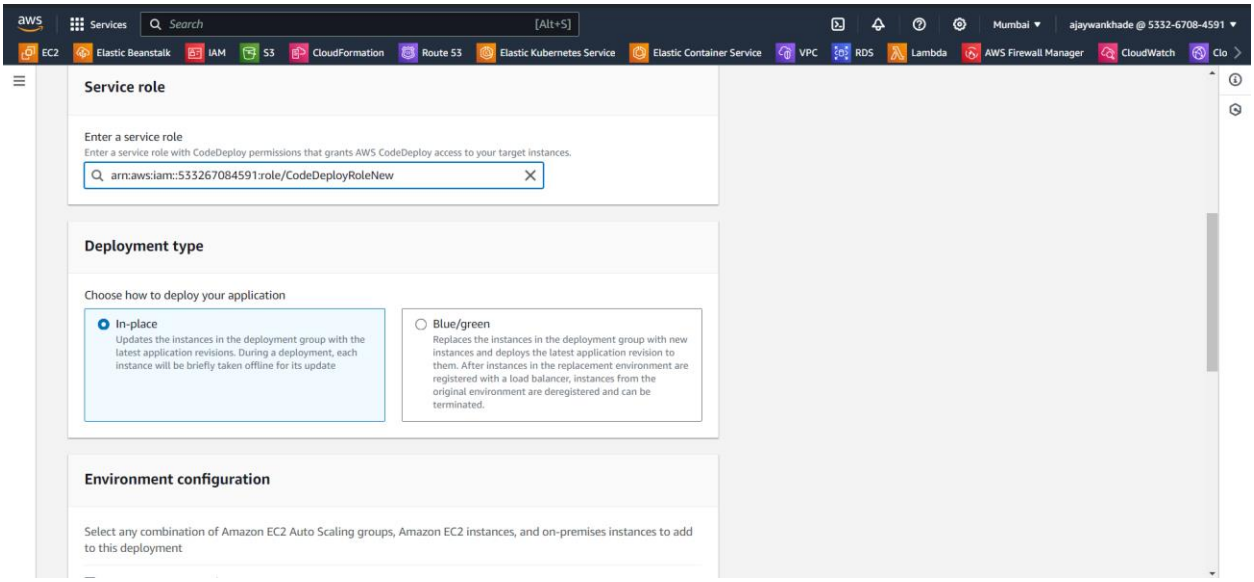


To create a deployment group in Code Deploy:

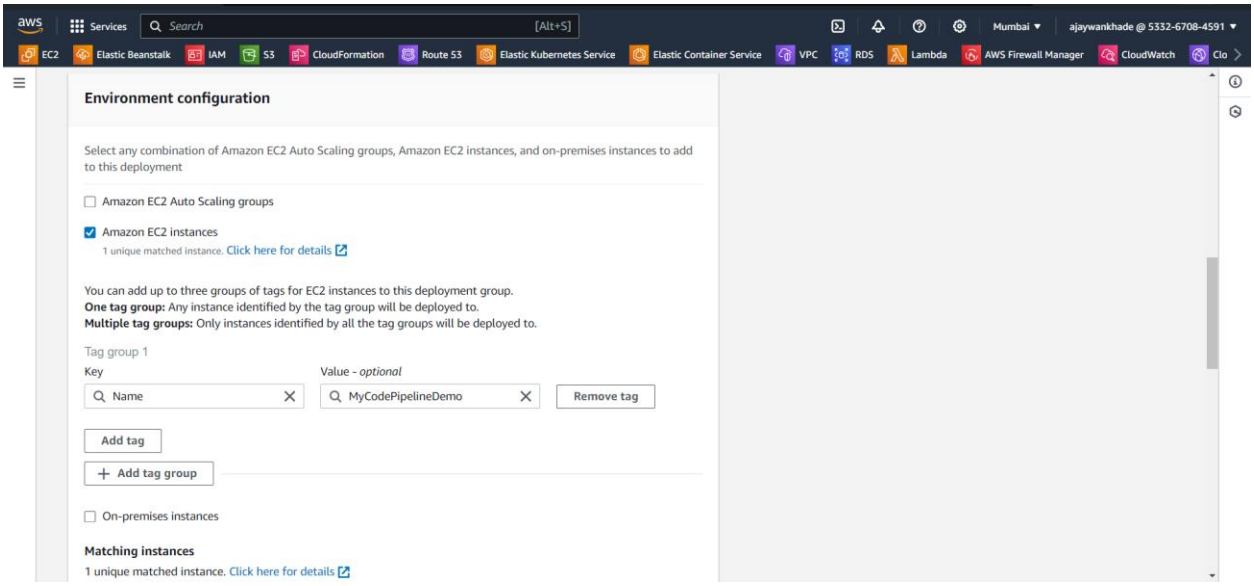
1. On the page that displays your application, choose **Create deployment group**.
2. In **Deployment group name**, enter `MyDemoDeploymentGroup.`



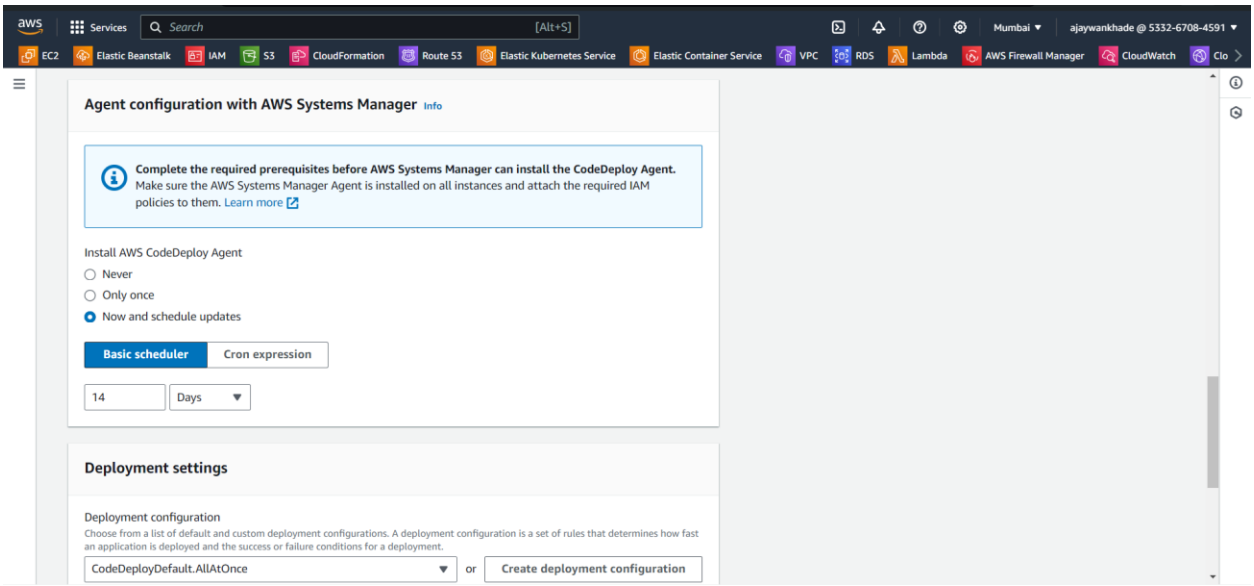
3. In **Service role**, choose the service role you created earlier.
4. Under **Deployment type**, choose **In-place**.



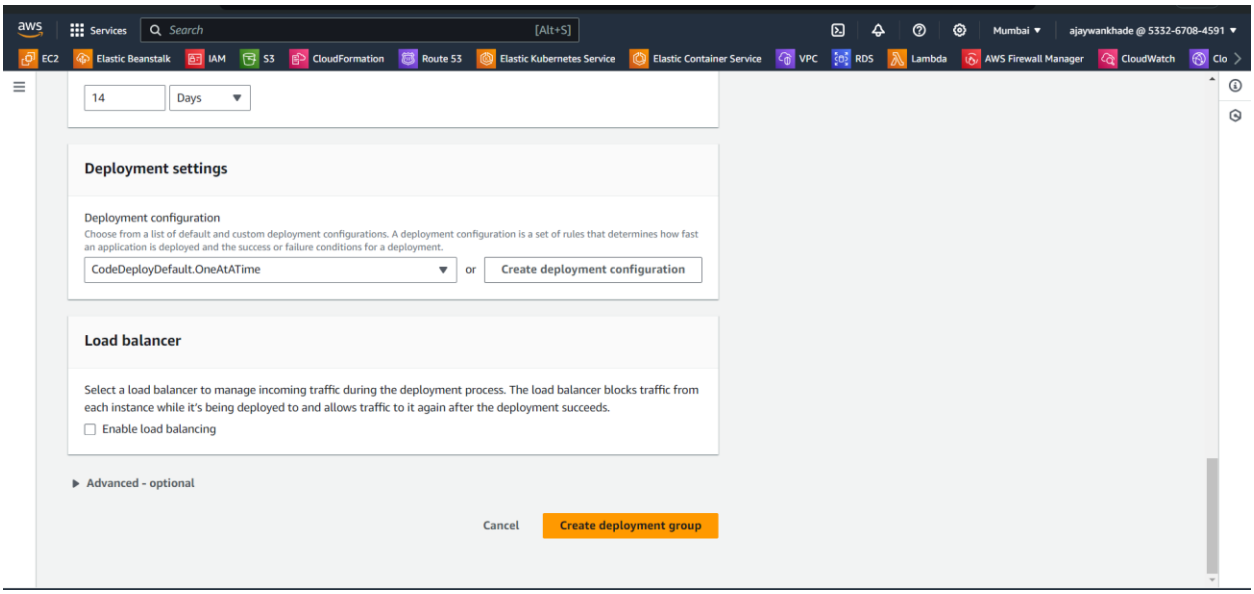
5. Under **Environment configuration**, choose **Amazon EC2 Instances**. Choose **Name** in the **Key** field, and in the **Value** field, enter `MyCodePipelineDemo`.



6. Under **Agent configuration with AWS Systems Manager**, choose **Now and schedule updates**. This installs the agent on the instance.



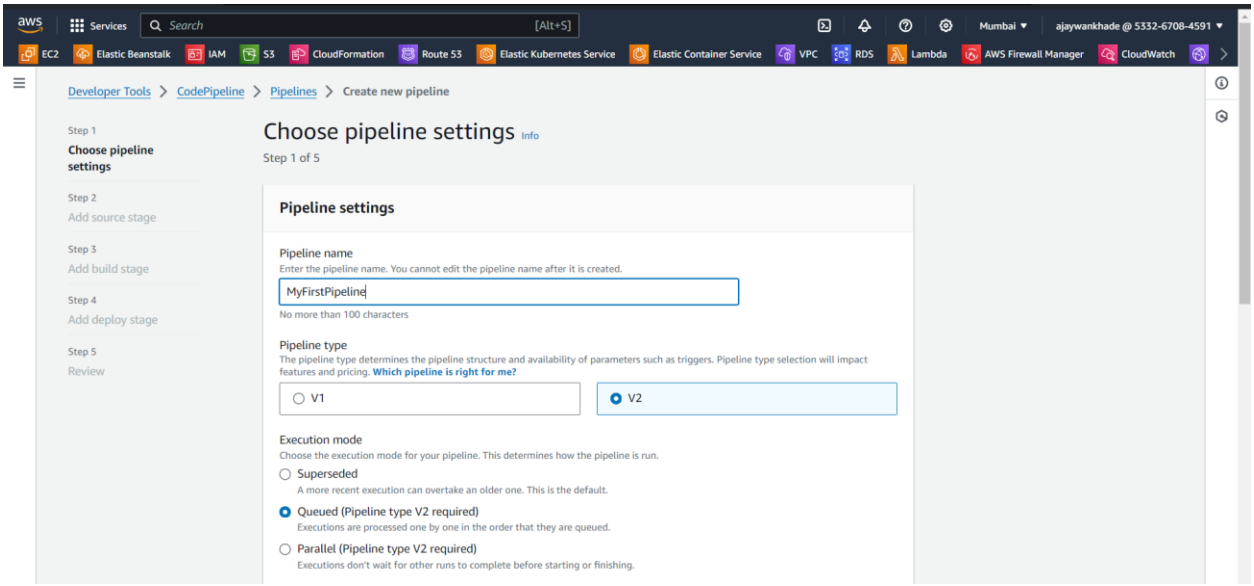
7. Under **Deployment settings**, choose `CodeDeployDefault.OneAtaTime`.
8. Under **Load Balancer**, make sure the **Enable load balancing** box is not selected.
9. In the **Advanced** section, leave the defaults.
10. Choose **Create deployment group**.



Step 4: Create your first pipeline in Code Pipeline.

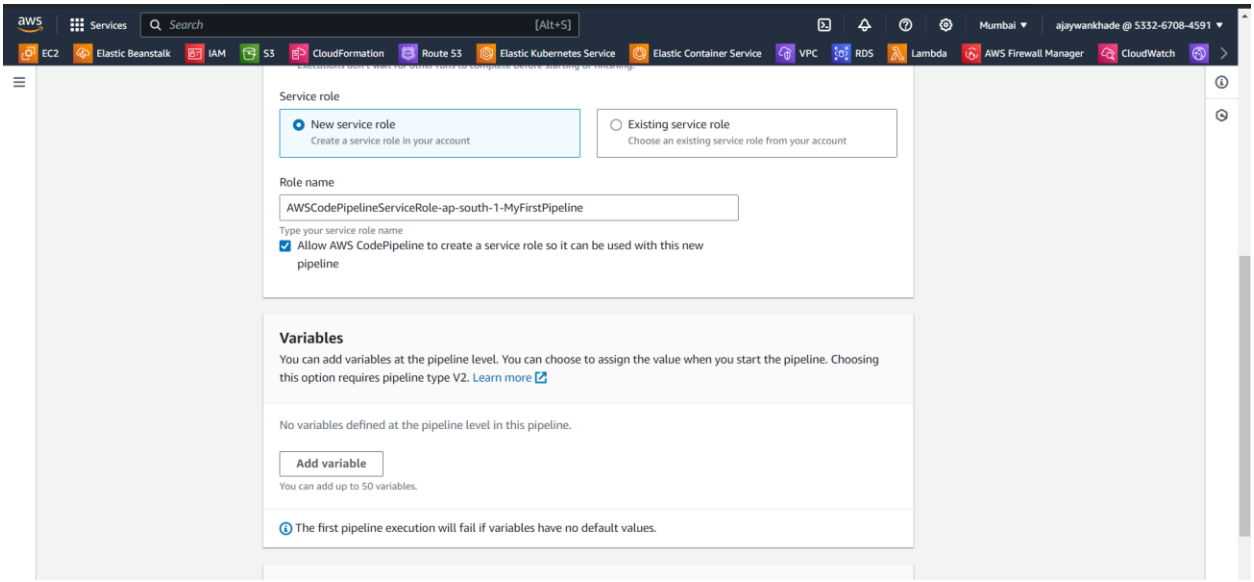
In this part of the tutorial, you create the pipeline. The sample runs automatically through the pipeline.

1. On the **Welcome** page, **Getting started** page, or the **Pipelines** page, choose **Create pipeline**.
2. In **Step 1: Choose pipeline settings**, in **Pipeline name**, enter `MyFirstPipeline`.

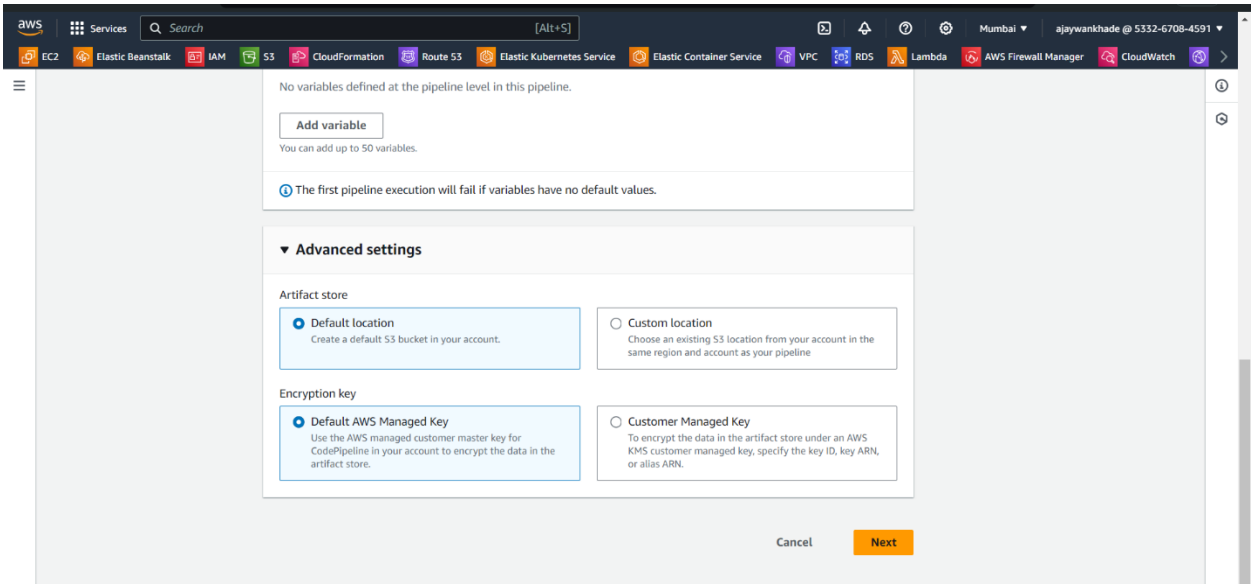


3. In **Service role**, do one of the following:

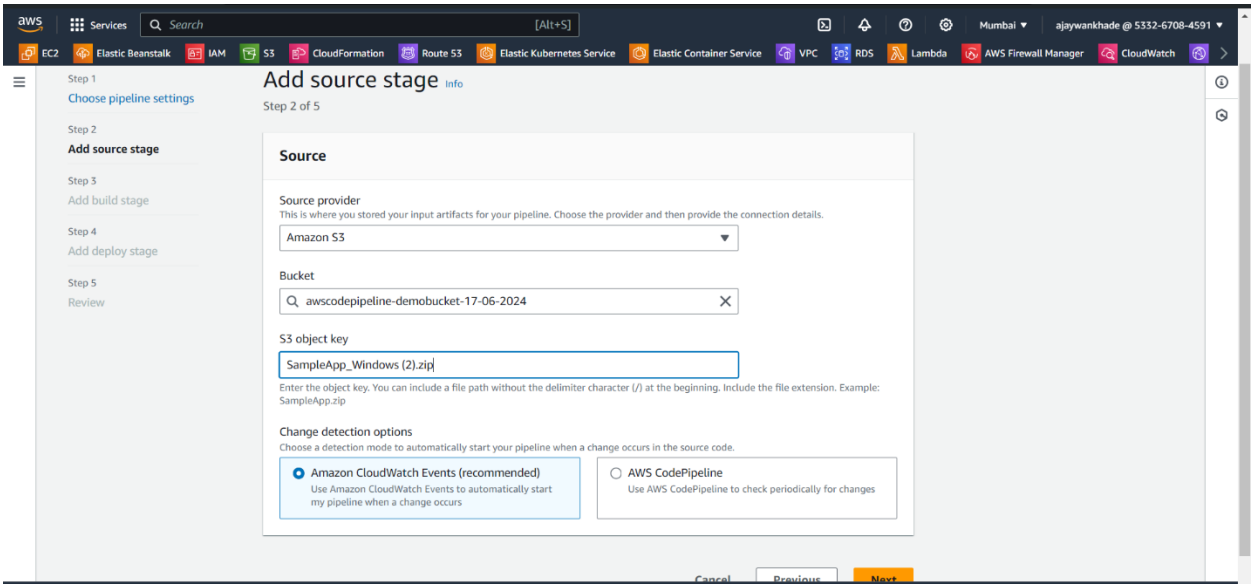
- Choose **New service role** to allow Code Pipeline to create a new service role in IAM.
- Choose **Existing service role** to use a service role already created in IAM. In **Role name**, choose your service role from the list.



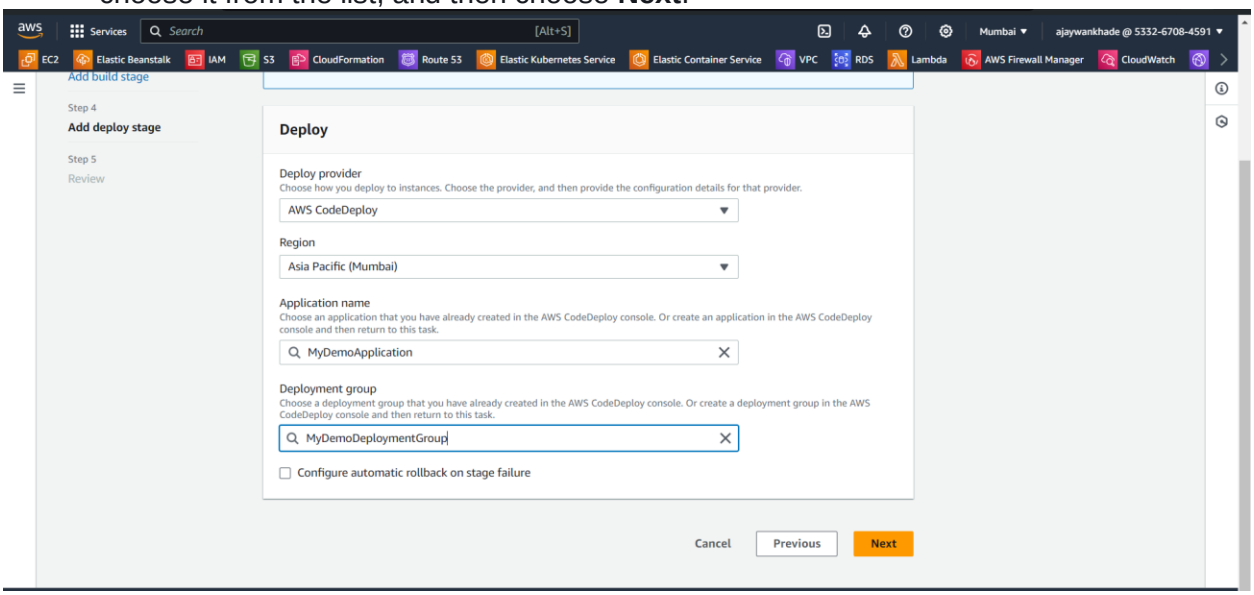
4. Leave the settings under **Advanced settings** at their defaults, and then choose **Next**.



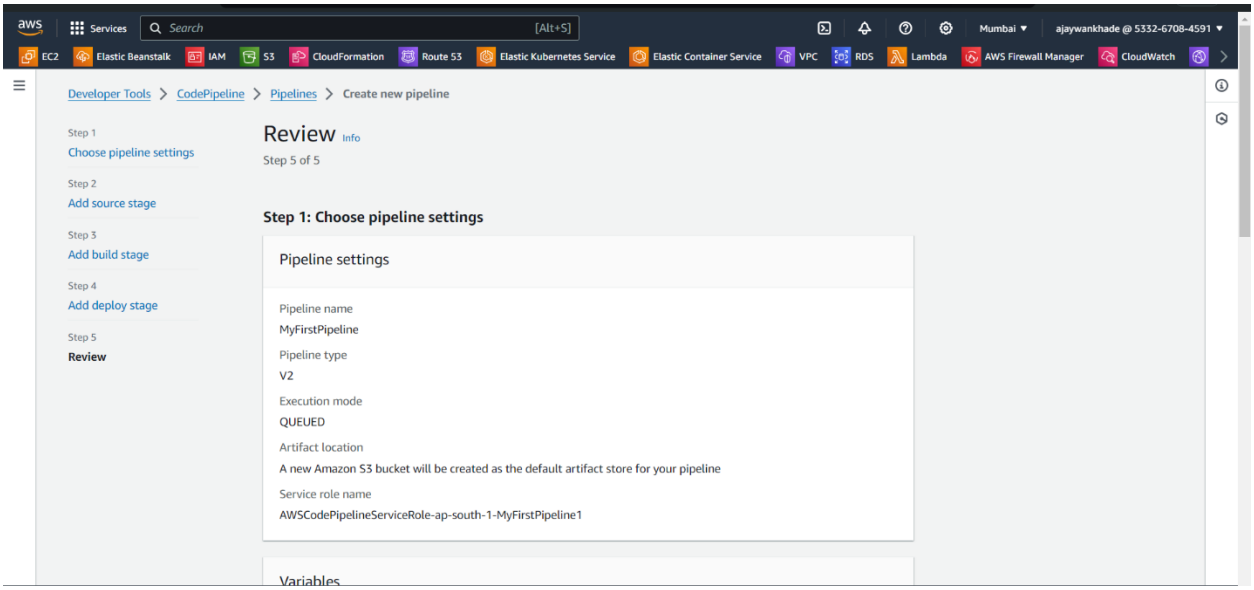
5. In **Step 2: Add source stage**, in **Source provider**, choose **Amazon S3**. In **Bucket**, enter the name of the S3 bucket you created in [Step 1: Create an S3 bucket for your application](). In **S3 object key**, enter the object key with or without a file path, and remember to include the file extension. For example, for `SampleApp_Windows.zip`, enter the sample file name as shown in this example:



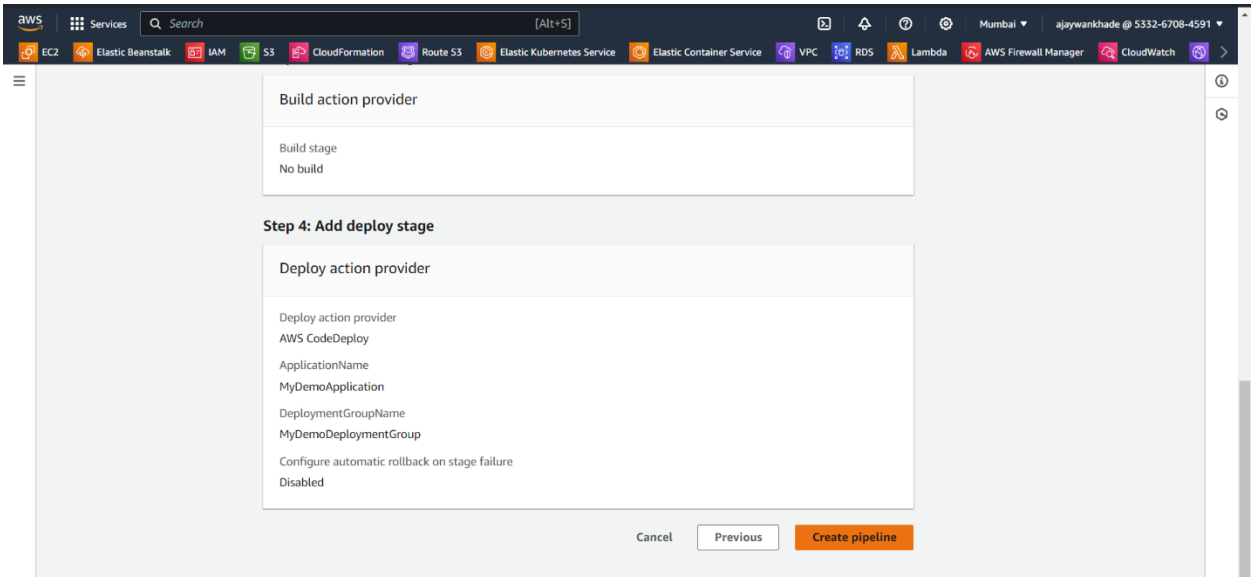
6. In **Step 3: Add build stage**, choose **Skip build stage**, and then accept the warning message by choosing **Skip** again. Choose **Next**.
7. In **Step 4: Add deploy stage**, in **Deploy provider**, choose **Code Deploy.**
   The **Region** field defaults to the same AWS Region as your pipeline. In **Application name**, enter `MyDemoApplication`, or choose the **Refresh** button, and then choose the application name from the list. In **Deployment group**, enter `MyDemoDeploymentGroup`, or choose it from the list, and then choose **Next**.



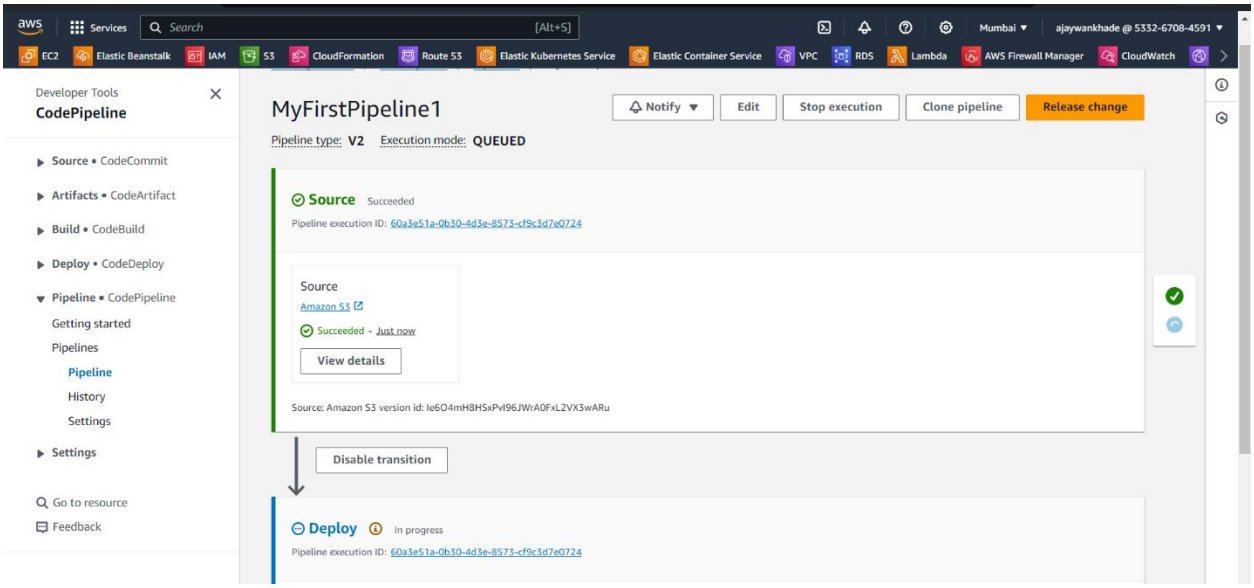
8. In **Step 5: Review**, review the information, and then choose **Create pipeline**.



9. Choose **Create pipeline**.

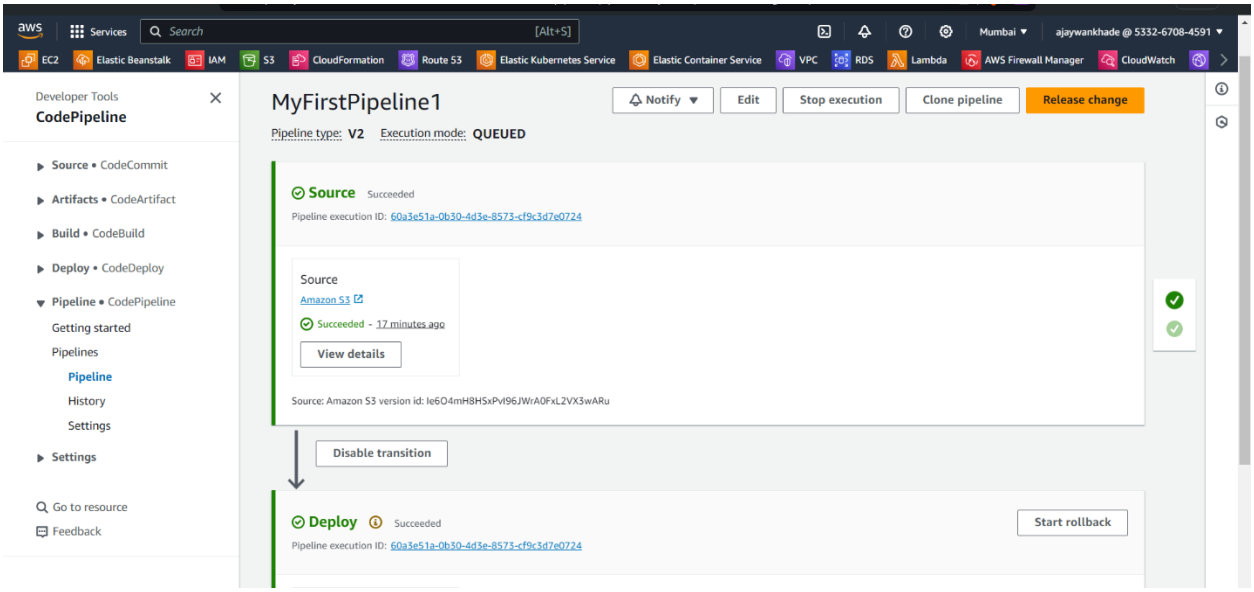


10. The pipeline starts to run. You can view progress and success and failure messages as the Code Pipeline sample deploys a webpage to each of the Amazon EC2 instances in the Code Deploy deployment.



11. View the initial progress of the pipeline. The status of each stage changes from **No executions yet** to **In Progress**, and then to either **Succeeded** or **Failed**. The pipeline should complete the first run within a few minutes.

12. After **Succeeded** is displayed for the action status, in the status area for the **Deploy** stage, choose **Details**. This opens the Code Deploy console.



Step 6: To verify your pipeline ran successfully.

On the **Description** tab, in **Public DNS**, copy the address, and then paste it into the address bar of your web browser. View the index page for the sample application you uploaded to your S3 bucket.