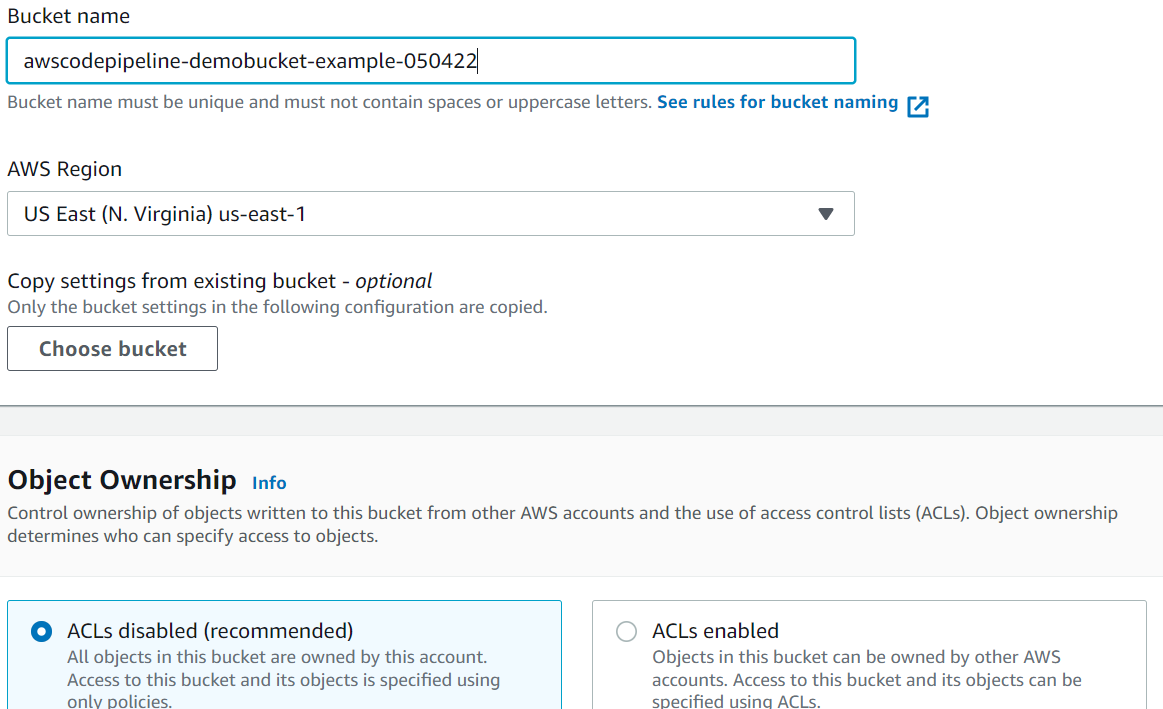
**Working with AWS CodePipeline**

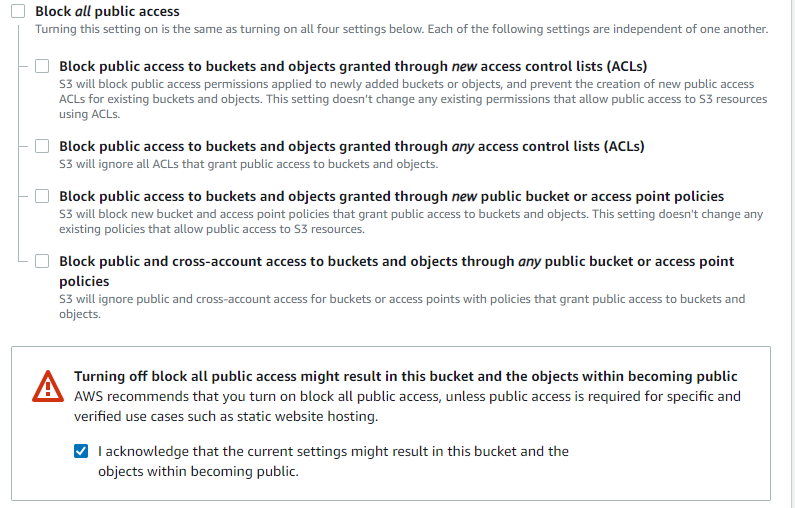
In this lab, you would push a file named index.html which has content from your local machine to Windows EC2 instance using S3, Code Deploy and CodePipleine .

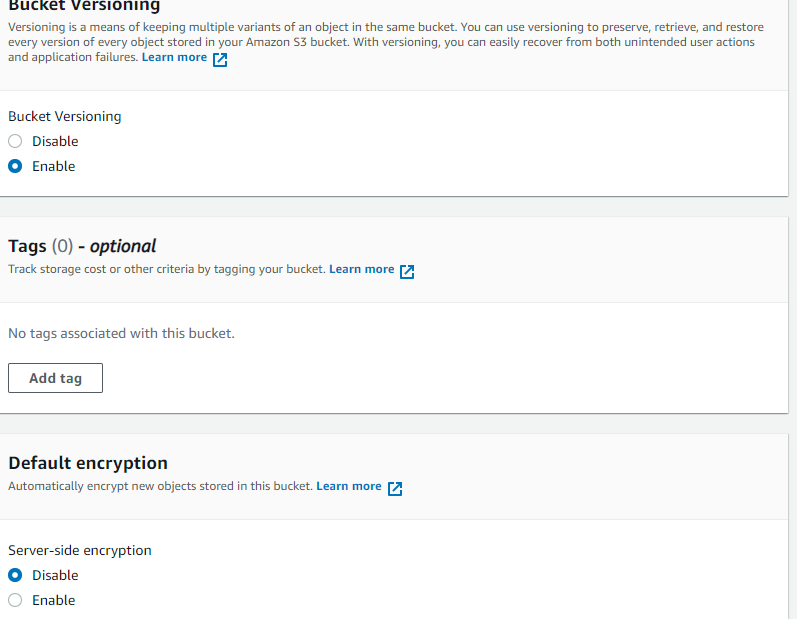
**Step 1: Create an S3 bucket for your application**

**Create S3 bucket**

* Sign in to the console at AWS Management Console. Open the S3 console.
* Choose Create bucket.(Under Object Ownership choose ACLs enable)
* In Bucket name, enter a name for your bucket (for example, awscodepipeline-demobucket-example-date).
* Make the bucket public and enable versioning on the bucket



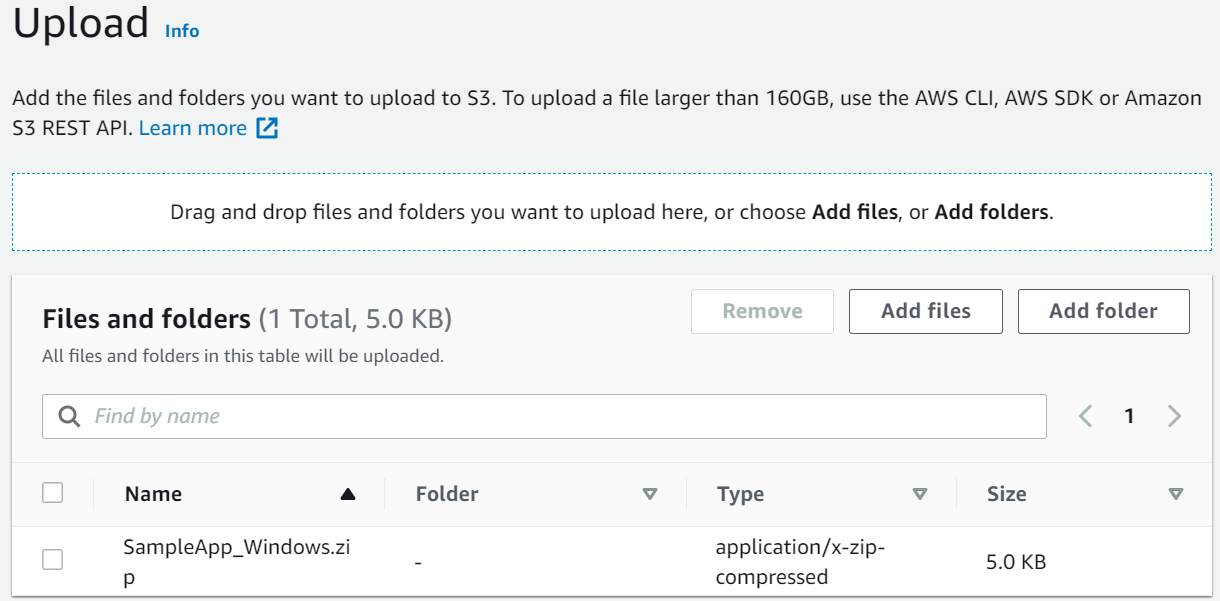


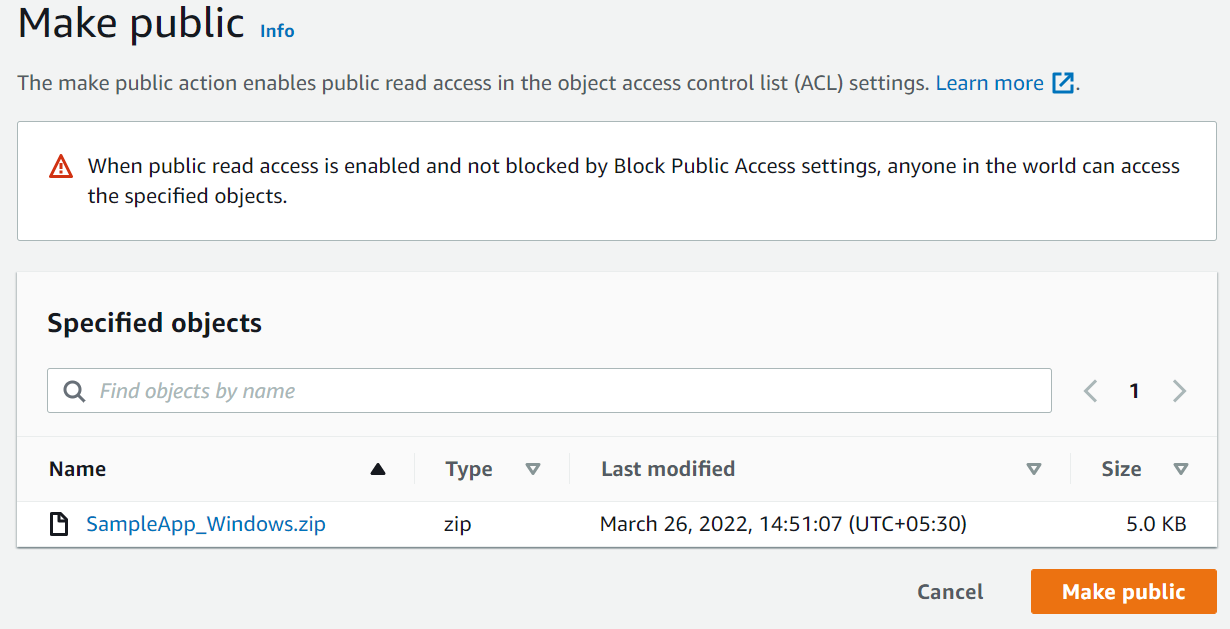


**Step 2: Download the code file which has index.html on your syste,**

* Download SampleApp\_Windows.zip from https://docs.aws.amazon.com/codepipeline/latest/userguide/samples/SampleApp\_Windows.zip
* Observe the contents of the zip file , there is an appspec yaml file which has the details of the location in windows ec2 instance where index.html would be deployed ( C:\inetpub\wwwroot)

**Step 3: Upload the zip file in the bucket and make the object /file public**

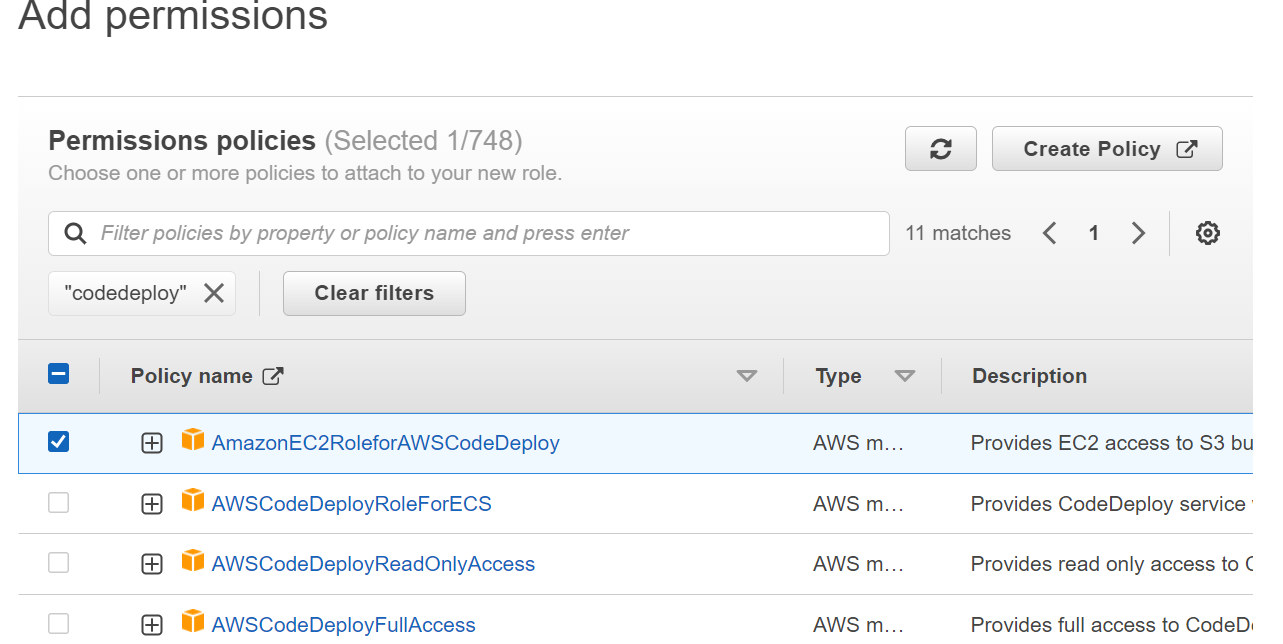


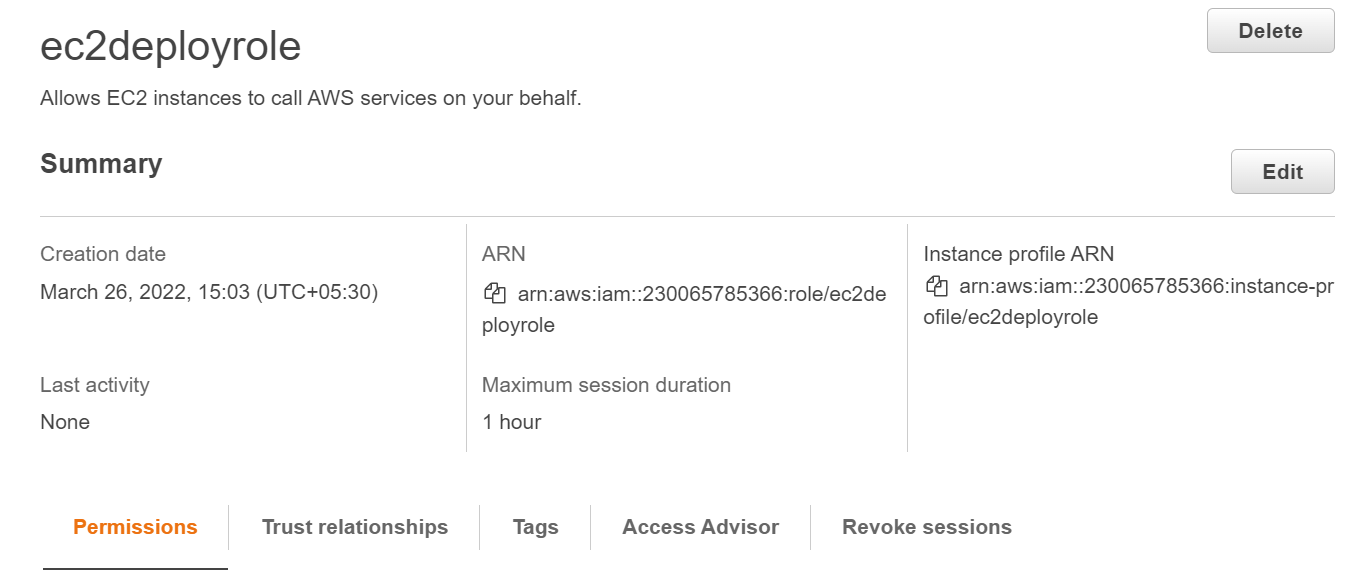


**4. Create roles in AWS IAM To create an instance role**

**EC2 instance role:**

* Open the IAM console at https://console.aws.amazon.com/iam/).
* From the console dashboard, choose Roles.
* Choose Create role.
* Under Select type of trusted entity, select AWS service. Under Choose a use case, select EC2, and then choose Next: Permissions.
* Search for and select the policy named AmazonEC2RoleforAWSCodeDeploy, and then choose Next: Tags.
* Choose Next: Review. Enter a name for the role named as ec2deployrole

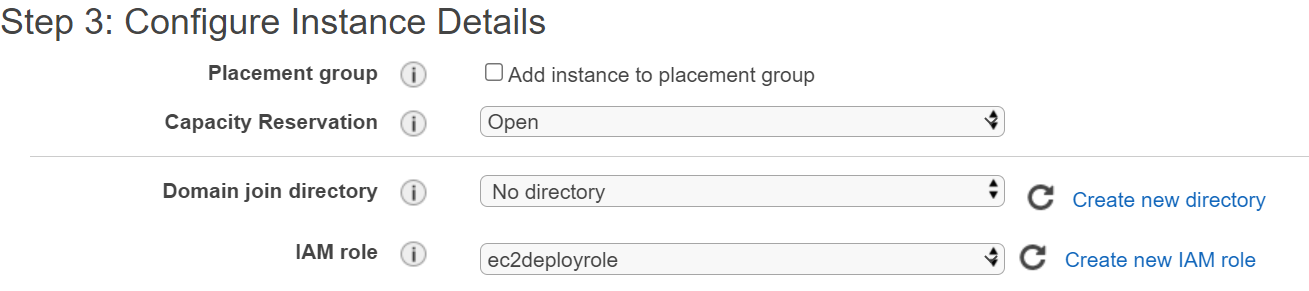




* Also create one more service role for Codedeploy with permission/policy - [AWSCodeDeployRole](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole)

**Step 5: Create EC2 instances with appropriate roles and install code deploy agent**

* Under EC2 Dashboard create a new Windows 2019 EC2 instance in default VPC and choose the role which was created in the previous step



* For installing codeployagent pass the below script under User Data section.

<powershell>

New-Item -Path c:\temp -ItemType "directory" -Force

powershell.exe -Command Read-S3Object -BucketName bucket-name/latest -Key codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi

Start-Process -Wait -FilePath c:\temp\codedeploy-agent.msi -WindowStyle Hidden

</powershell>

In the bucket-name replace with the region in which you are creating Windows instance (For N.Virginia it would be aws-codedeploy-us-east-1)

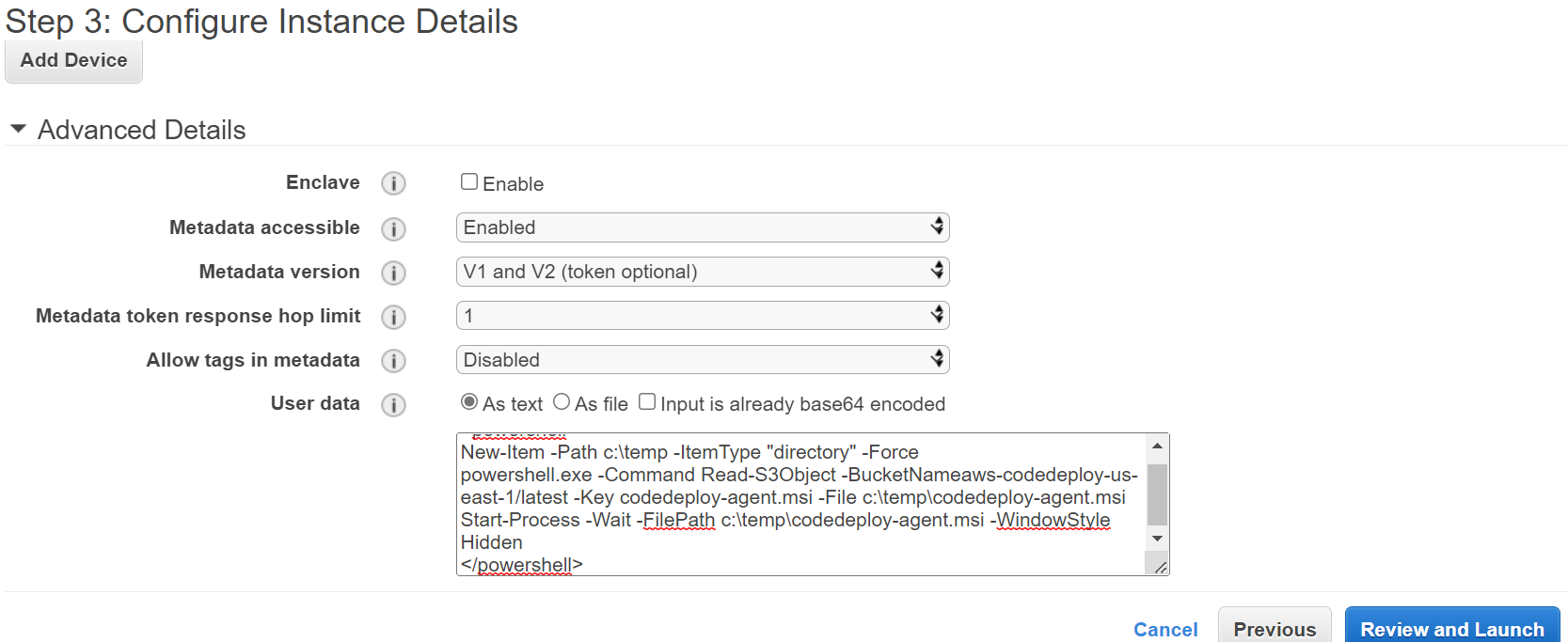
Eg: <powershell>

New-Item -Path c:\temp -ItemType "directory" -Force

powershell.exe -Command Read-S3Object -BucketName aws-codedeploy-us-east-1/latest -Key codedeploy-agent.msi -File c:\temp\codedeploy-agent.msi

Start-Process -Wait -FilePath c:\temp\codedeploy-agent.msi -WindowStyle Hidden

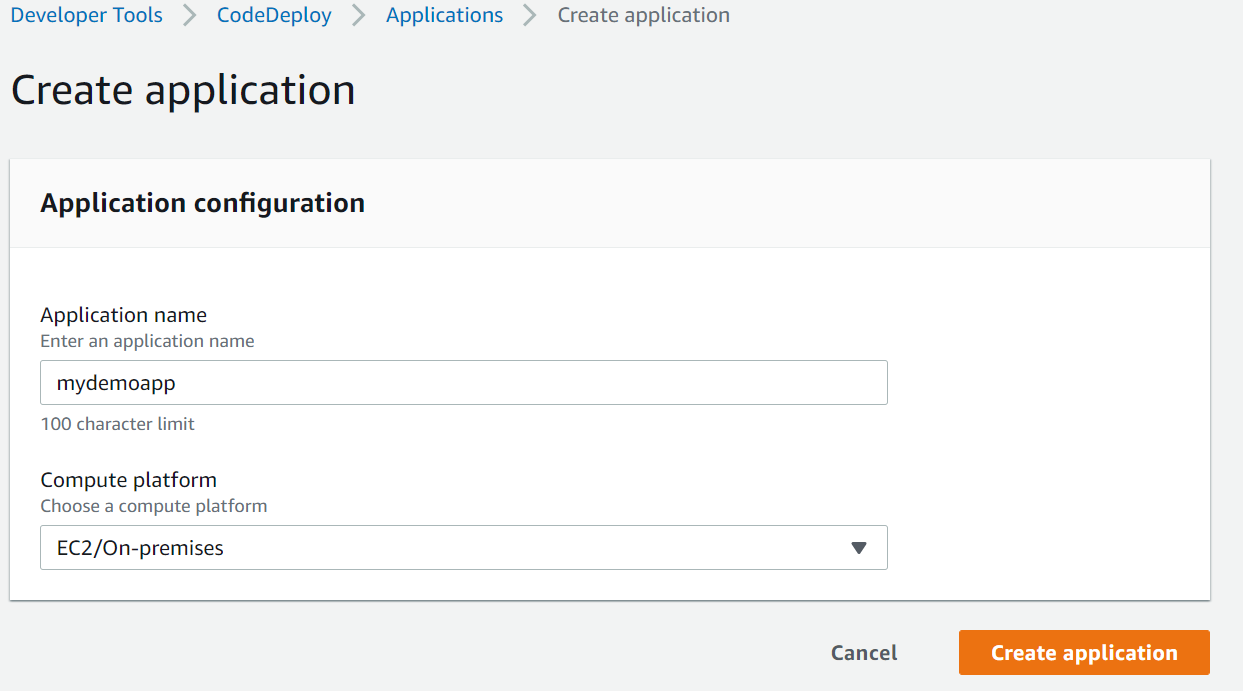
</powershell>



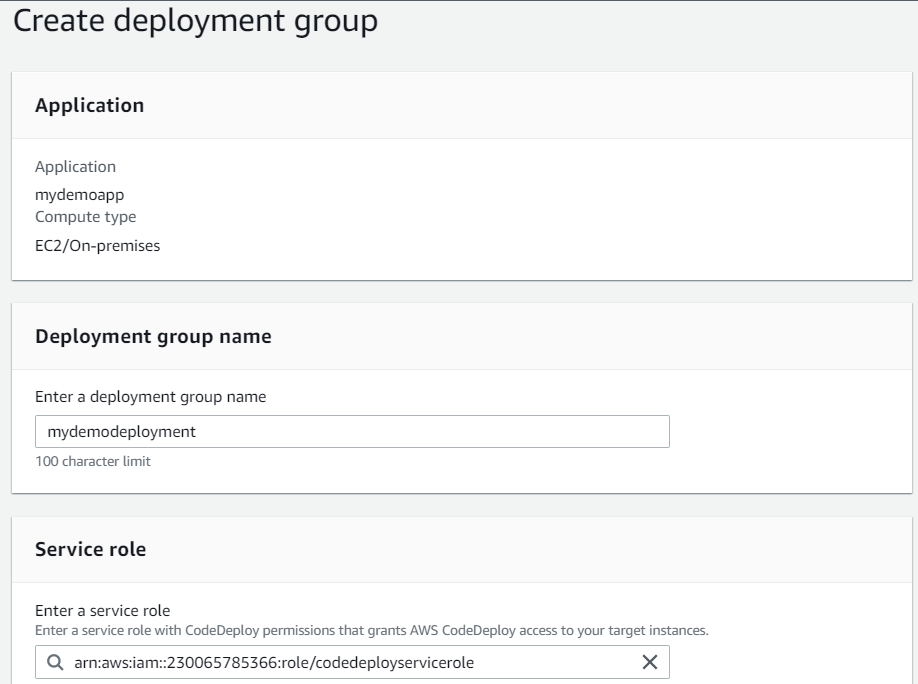
* Click on Next and provide tag name as CodePipelinedemo. Choose All traffic under Security Group and from anywhere and create EC2 instance
* During the boot up of EC2 instance , codedeploy agent script would run and install codedeploy agent in the instance

**Step 6: Create application and deployment group**

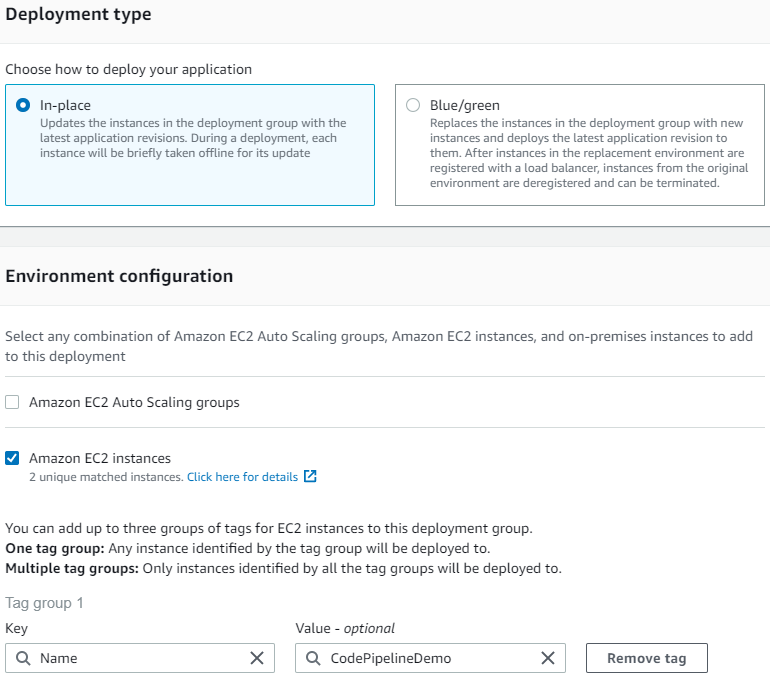
* Under CodeDeploy service create an application named as mydemoapp and choose compute platform as EC2.On-Premises



* Create deployment group named as mydemodeployment and choose the codedeploy service role

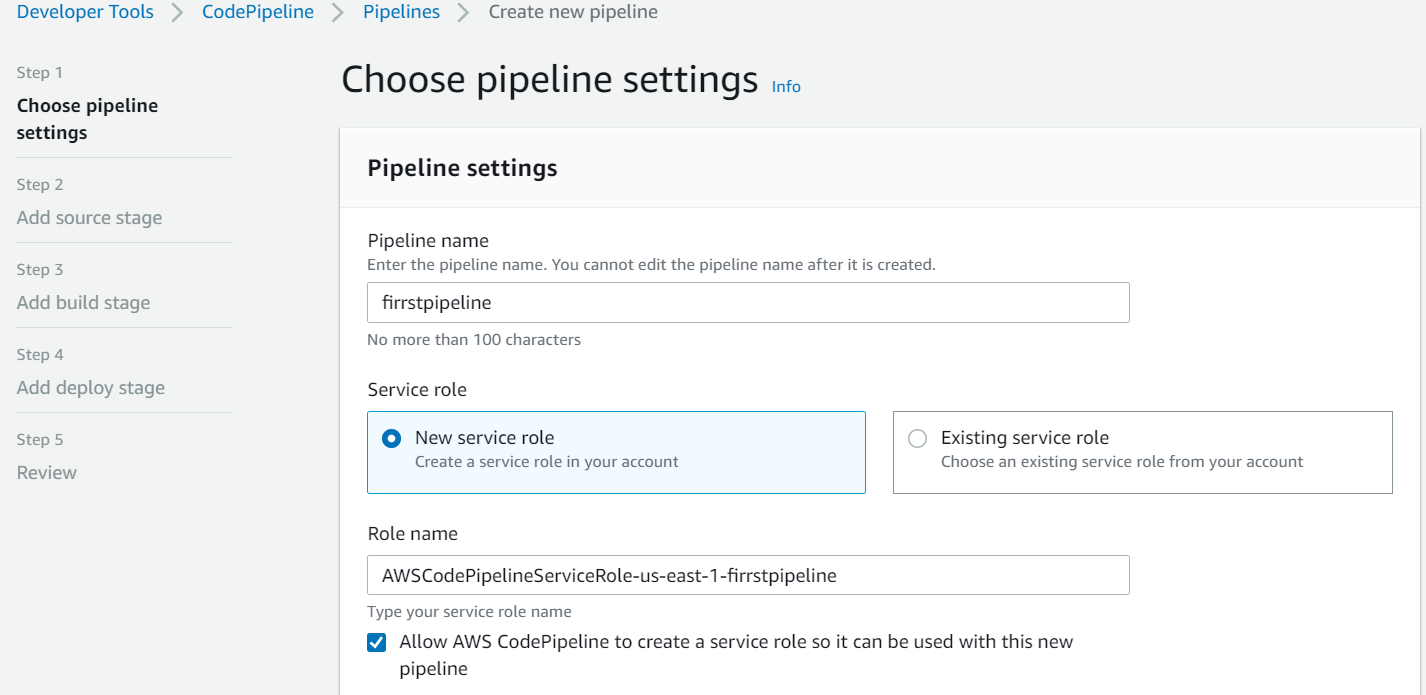


* Choose deployment type as In-place and choose the Windows instance using the tag names in which Codedeploy agent was installed

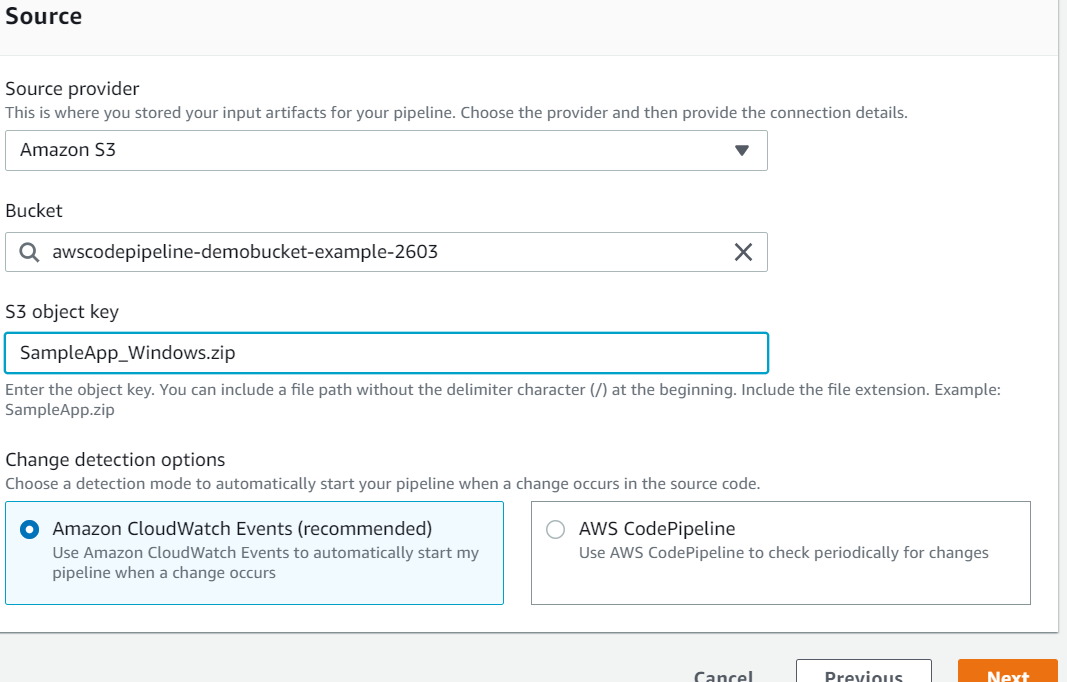


**Step 6: Working with CodePipeline**

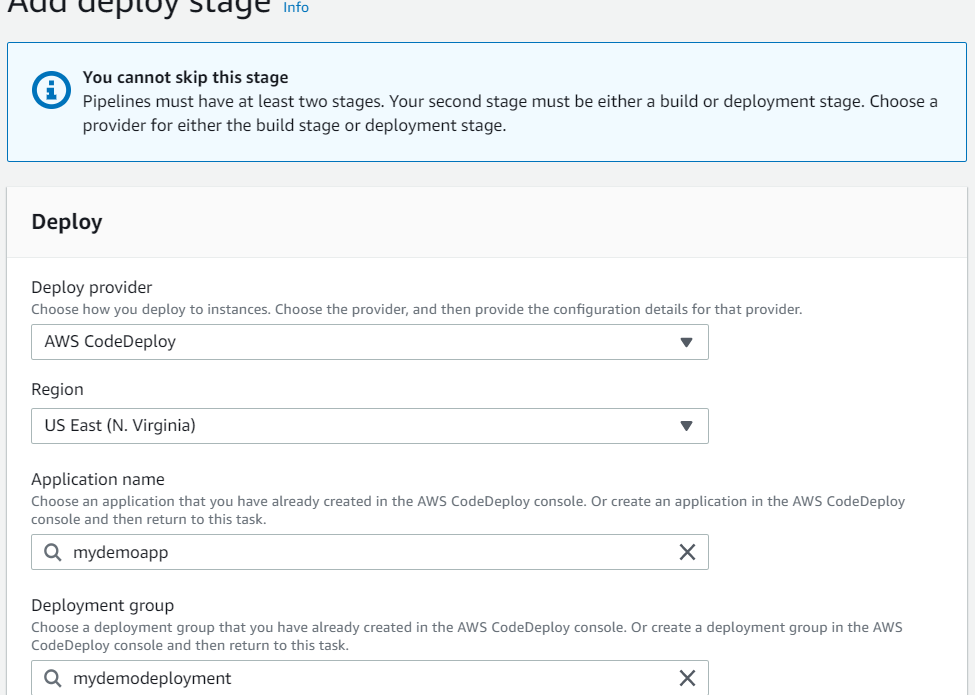
* Create a new pipeline named as firstpipeline application and choose service role and click Next



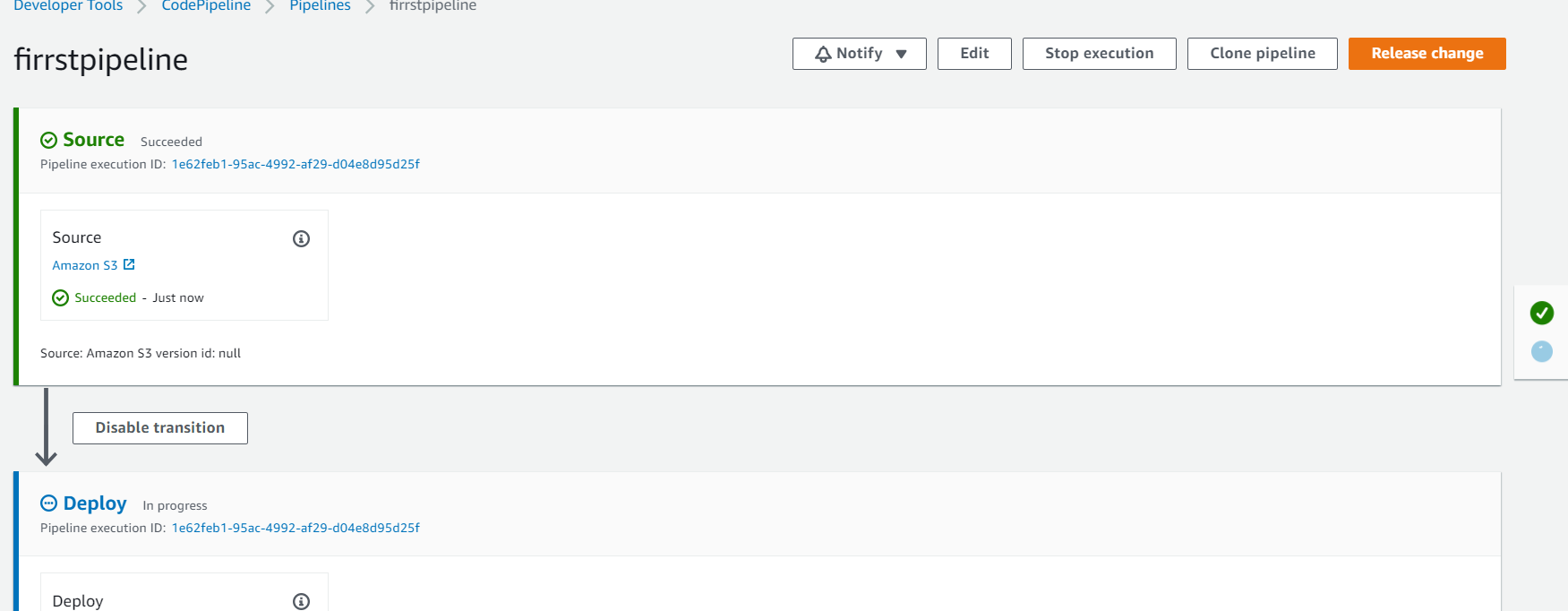
* Under Source Provider Choose Amazon S3 , update the bucket name and Object name as shown below .Also update Amazon Cloudwatch under change detection option.



* Under Deploy Stage, choose AWS CodeDeploy under Deploy Provider , Choose the region , Deployment Group name and application name and finish.



* Observe the pipeline runs and each stage gets completed.



**Step 7:** To verify your pipeline ran successfully

* 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.
* After Succeeded is displayed for the action status, in the status area for the Deploy stage, choose Details. This opens the CodeDeploy console.
* In the Deployment group tab, under Deployment lifecycle events, choose an instance ID. This opens the EC2 console.
* 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.
* The web page displays for the sample application you uploaded to your S3 bucket.