

Experiment no. 2

Aim :- To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

Theory :-

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

Benefits

Fully managed build service

AWS CodeBuild eliminates the need to set up, patch, update, and manage your own build servers and software. There is no software to install or manage.

Continuous scaling

AWS CodeBuild scales up and down automatically to meet your build volume. It immediately processes each build you submit and can run separate builds concurrently, which means your builds are not left waiting in a queue.

Pay as you go

With AWS CodeBuild, you are charged based on the number of minutes it takes to complete your build. This means you no longer have to worry about paying for idle build server capacity.

Extensible

You can bring your own build tools and programming runtimes to use with AWS CodeBuild by creating customized build environments in addition to the prepackaged build tools and runtimes supported by CodeBuild.

Enables continuous integration and delivery

AWS CodeBuild belongs to a family of AWS Code Services, which you can use to create complete, automated software release workflows for continuous integration and delivery (CI/CD). You can also integrate CodeBuild into your existing CI/CD workflow. For example, you can use CodeBuild as a worker node for your existing Jenkins server setup for distributed builds.

Secure

With AWS CodeBuild, your build artifacts are encrypted with customer-specific keys that are managed by the AWS Key Management Service (KMS). CodeBuild is integrated with AWS Identity and Access Management (IAM), so you can assign user-specific permissions to your build projects.

AWS CodePipeline

AWS CodePipeline is a continuous delivery service you can use to model, visualize, and automate the steps required to release your software. You can quickly model and configure the different stages of a software release process. CodePipeline automates the steps required to release your software changes continuously.

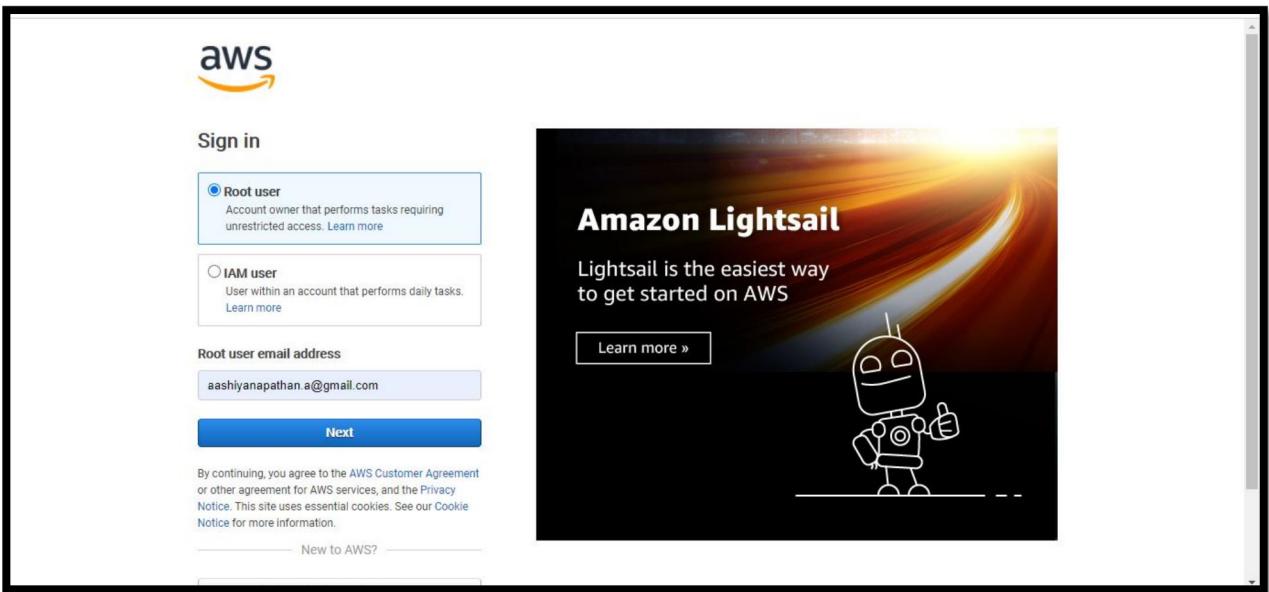
When developers commit changes to a source repository, CodePipeline automatically detects the changes. Those changes are built, and if any tests are configured, those tests are run. After the tests are complete, the built code is deployed to staging servers for testing. From the staging server, CodePipeline runs more tests, such as integration or load tests. Upon the successful completion of those tests, and after a manual approval action that was added to the pipeline is approved, CodePipeline deploys the tested and approved code to production instances.

CodePipeline can deploy applications to EC2 instances by using CodeDeploy, AWS Elastic Beanstalk, or AWS OpsWorks Stacks. CodePipeline can also deploy container-based applications to services by using Amazon ECS. Developers can also use the integration points provided with CodePipeline to plug in other tools or services, including build services, test providers, or other deployment targets or systems.

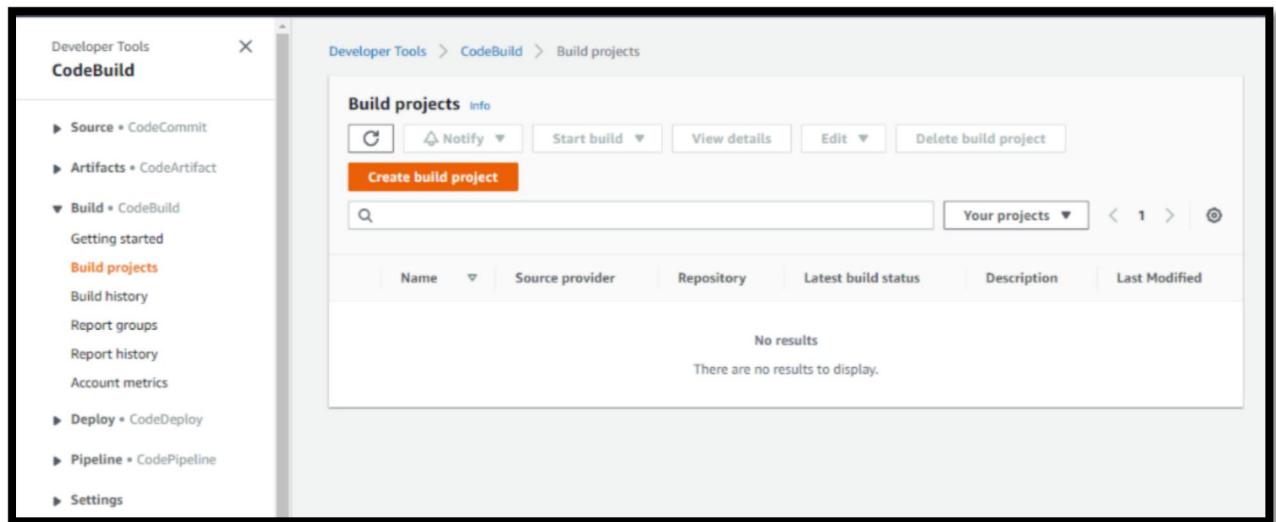
Steps:

1. Login with your AWS account.
2. Navigate to CodeBuild service from Developer tools section as below:

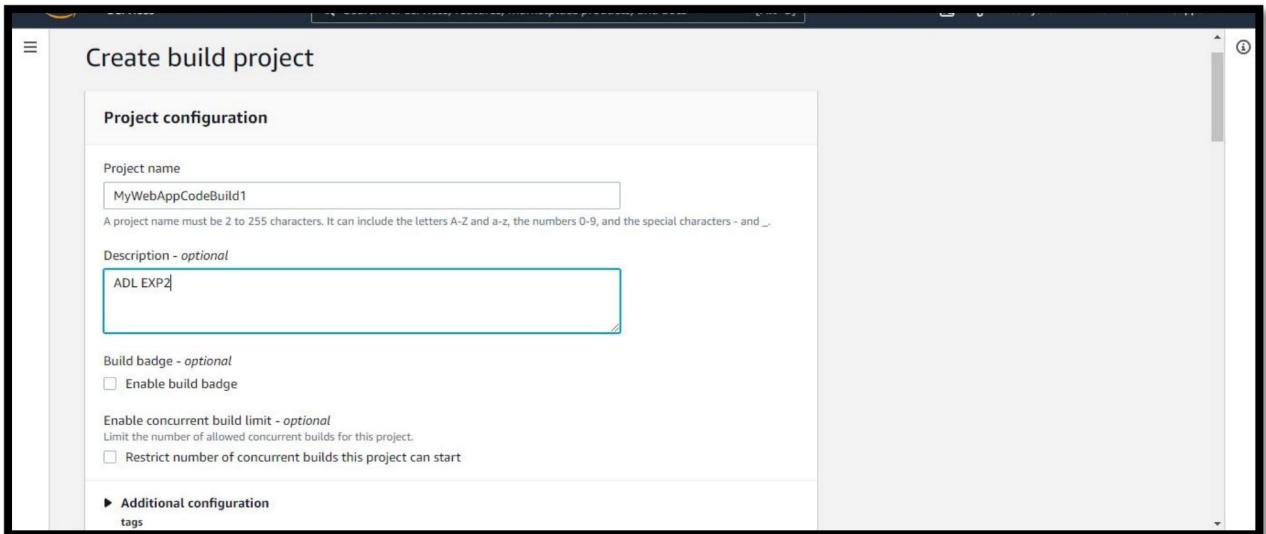




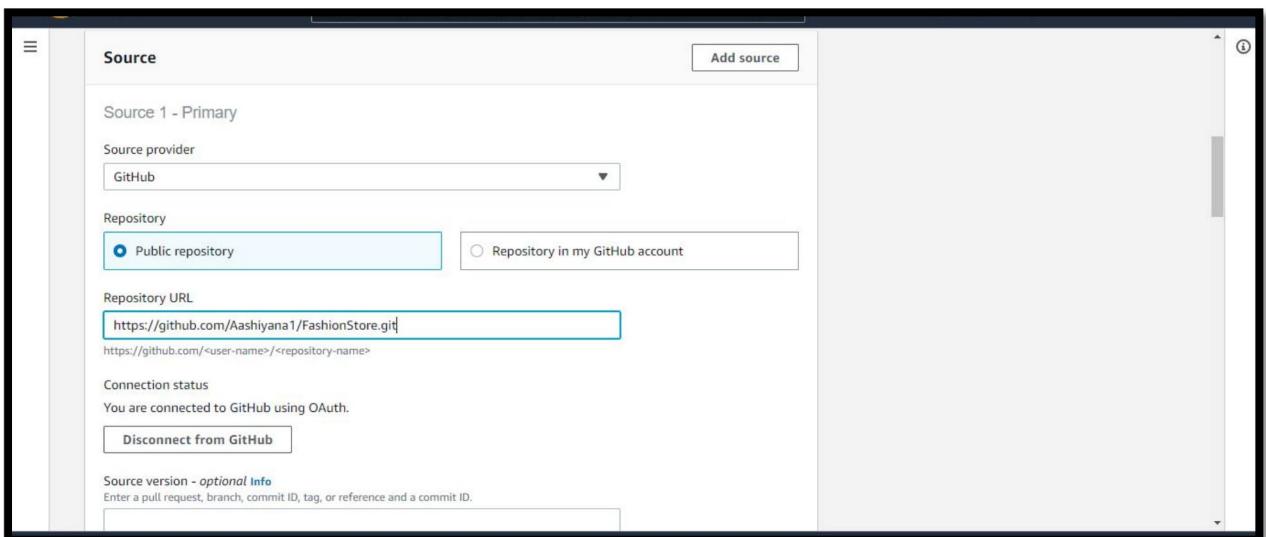
3. Click on Create build project :



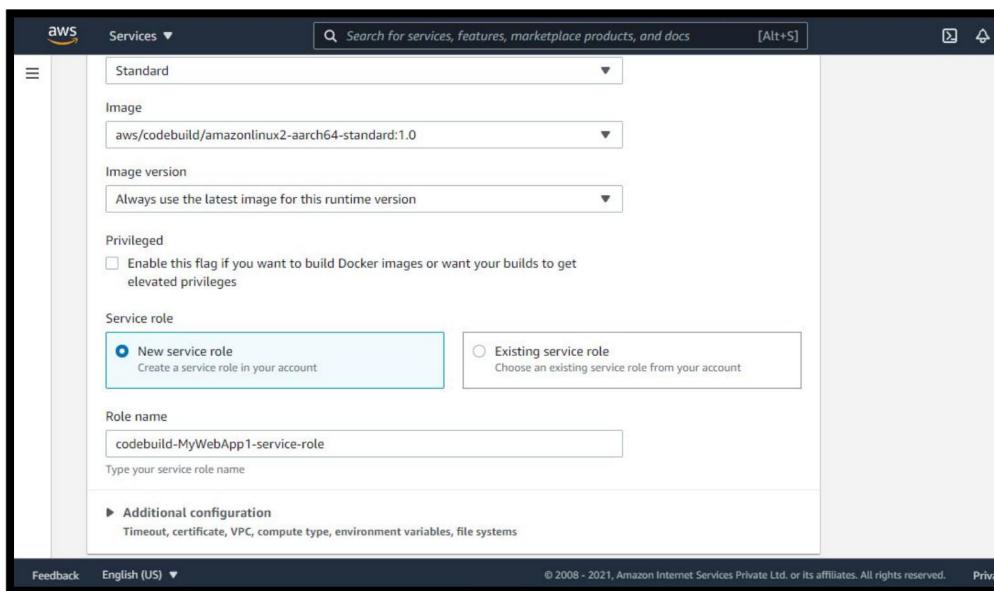
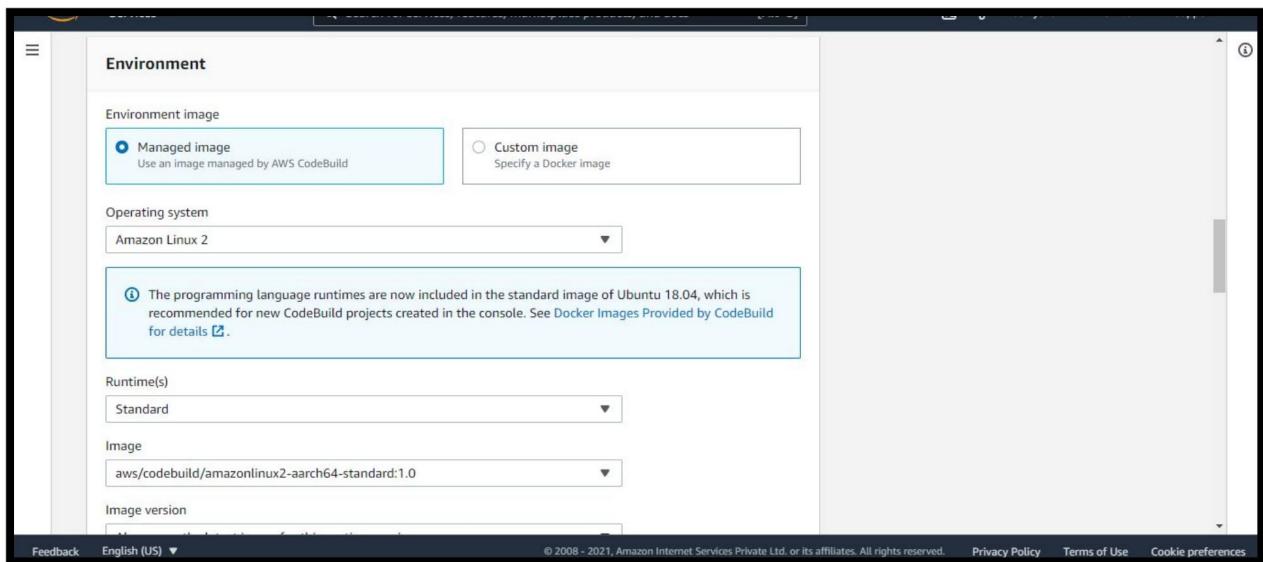
4. Provide the name for the build project(MyWebAppCodeBuild) and click on Next.

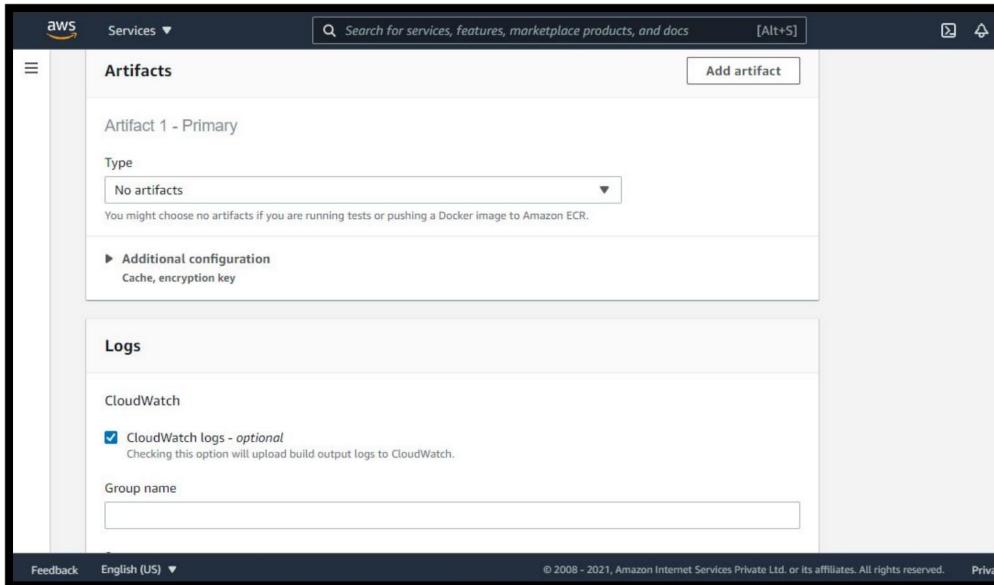
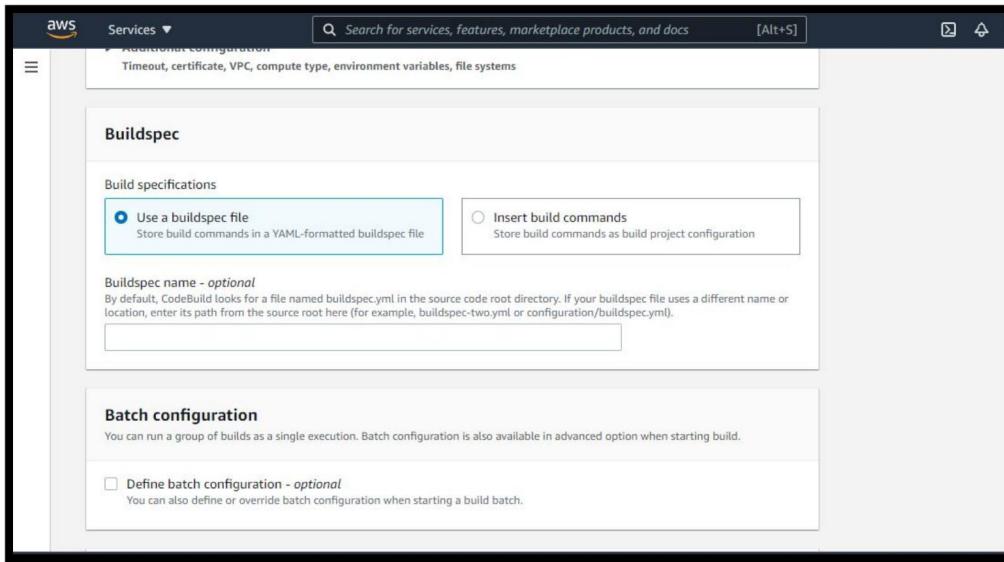


5. Add Source

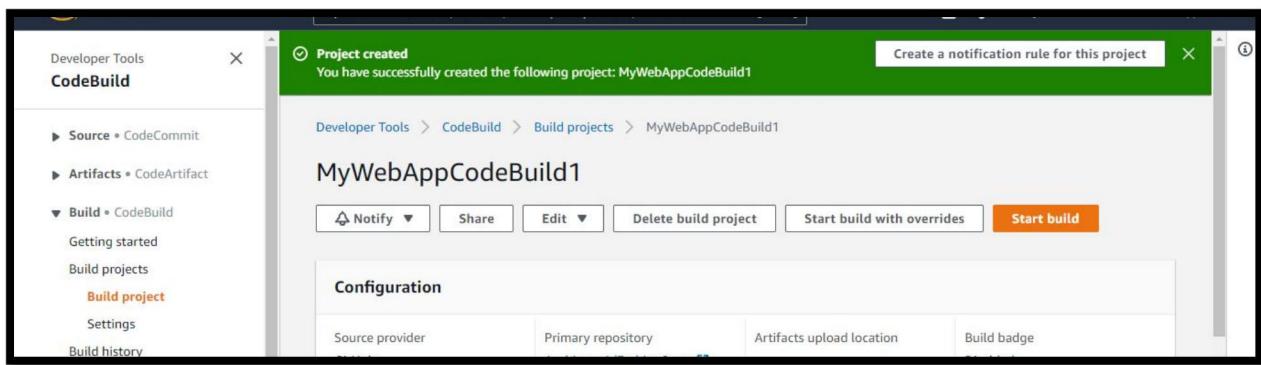
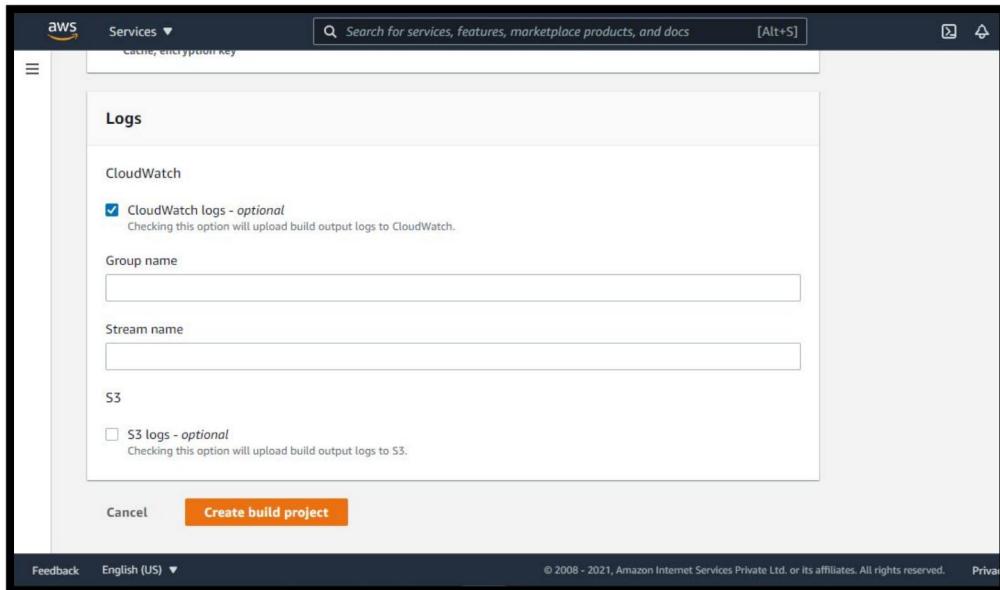


5. Add Environment image

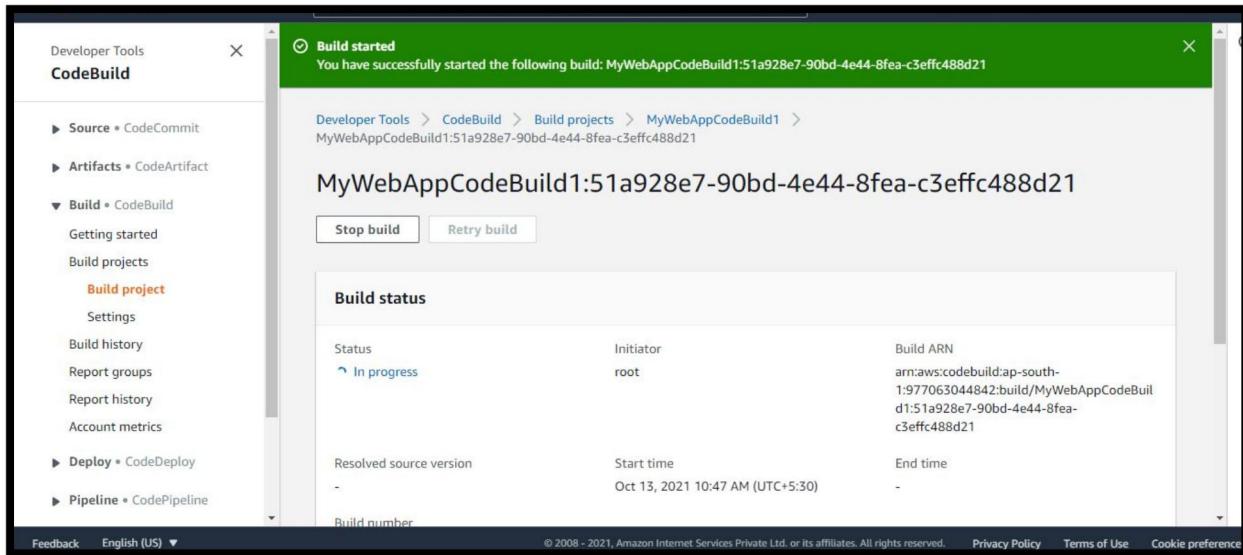




7. Click on Create build project



8. Click on Start build



Deploy on S3 / SEBS using AWS CodePipeline

Create an S3 bucket for your application

To create an S3 bucket

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.

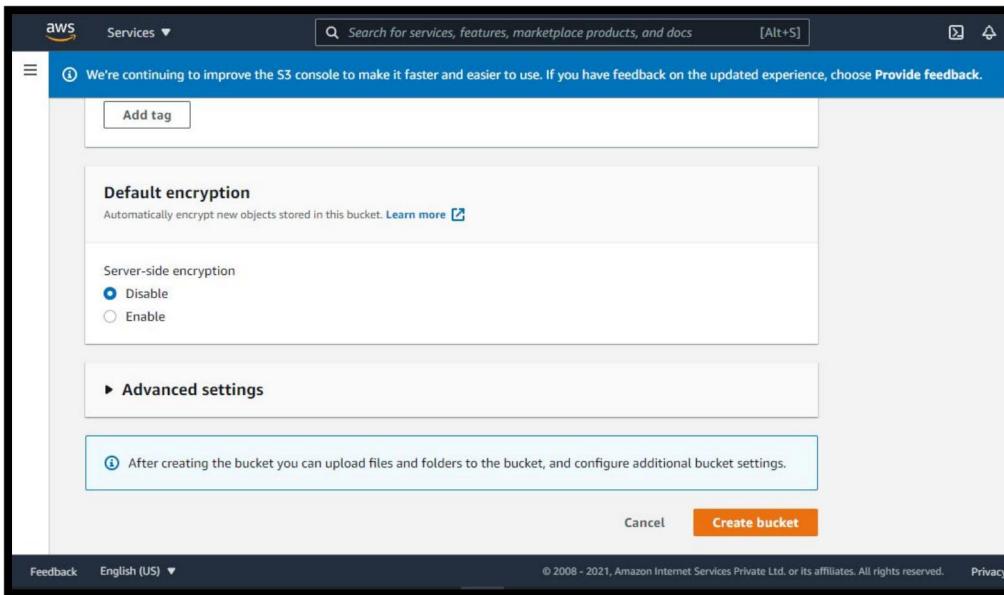
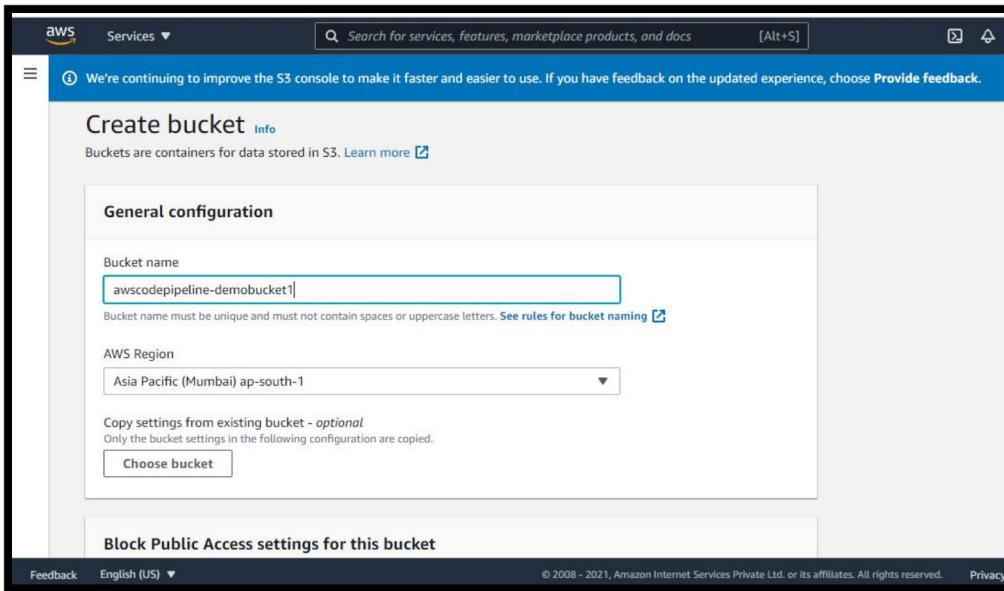
The screenshot shows the Amazon S3 console interface. On the left, there's a navigation sidebar with options like 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', and 'Feature spotlight'. The main content area has a blue header bar with a feedback message: 'We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.' Below this is an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. The main focus is a table titled 'Buckets (2) Info' which lists two buckets: 'codepipeline-ap-south-1-209041297449' and 'elasticbeanstalk-ap-south-1-977063044842'. The table includes columns for Name, AWS Region, Access, and Creation date.

2. Choose Create bucket.

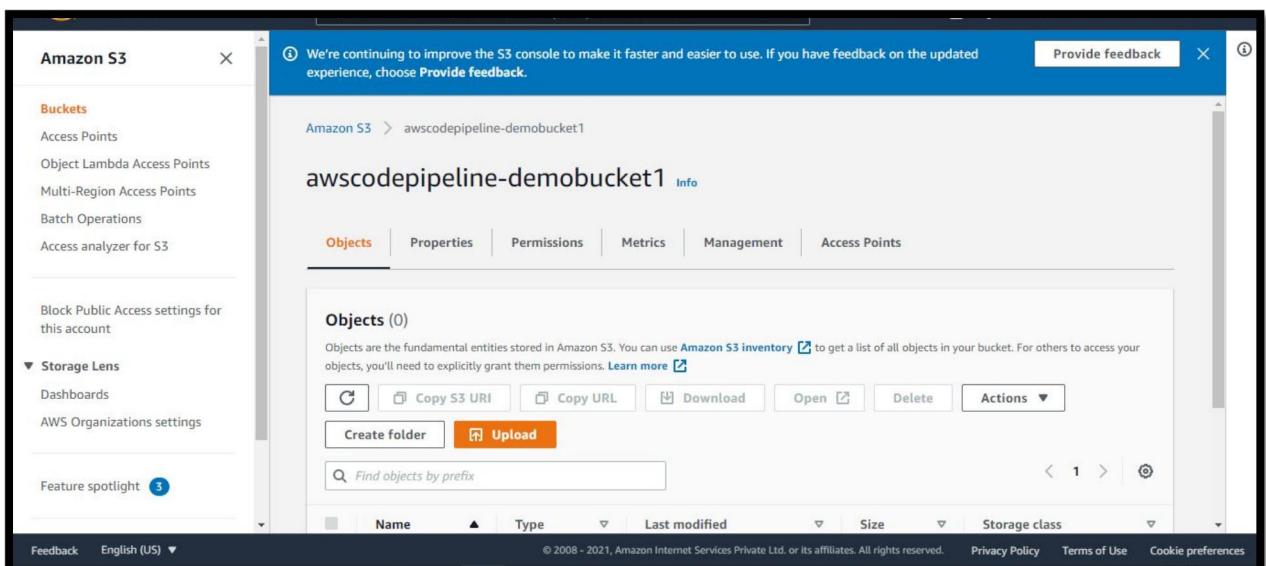
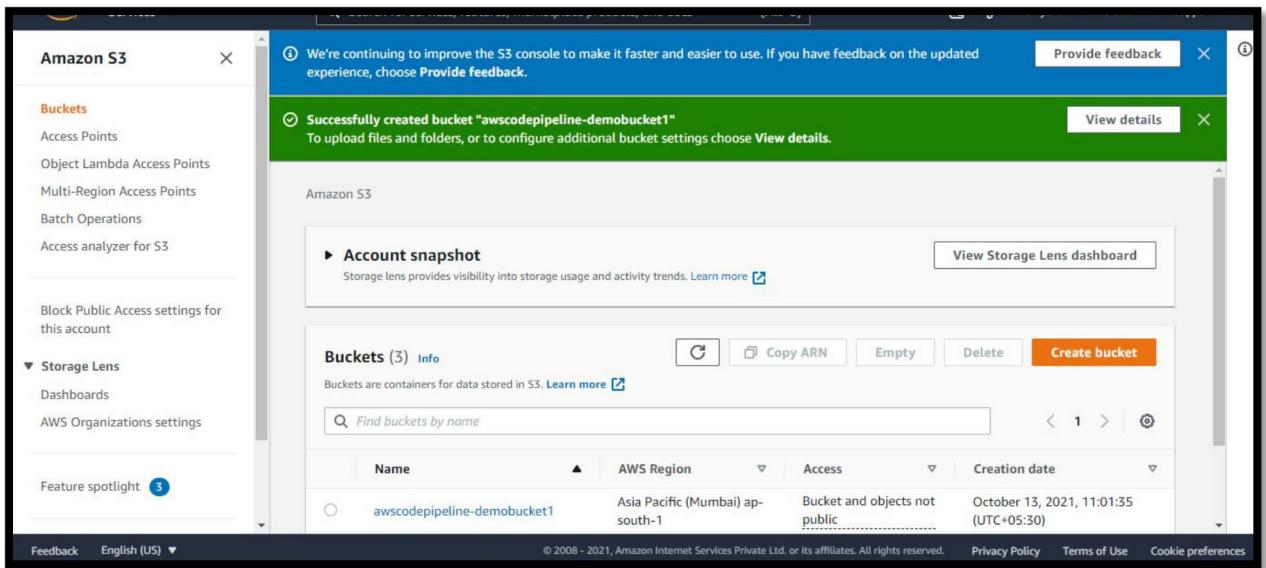
The screenshot shows the 'Create bucket' configuration page. At the top, there's a message: 'We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.' Below this is a section titled 'General configuration' with fields for 'Bucket name' (containing 'myawsbucket') and 'AWS Region' (set to 'Asia Pacific (Mumbai) ap-south-1'). There's also a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. At the bottom, there's a 'Block Public Access settings for this bucket' section.

3. In Bucket name, enter a name for your bucket (for example, awscodepipeline-demobucket-example-date).

In Region, choose the Region where you intend to create your pipeline, such as Asia Pacific (Mumbai), and then choose Create bucket.

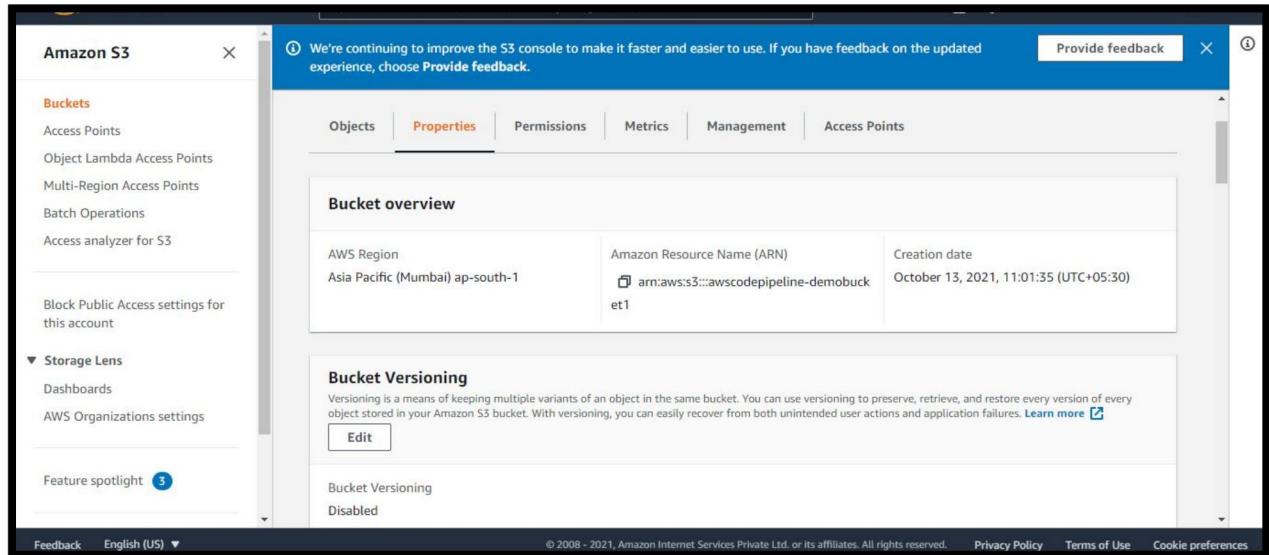


4. After the bucket is created, a success banner displays. Choose Go to bucket details.

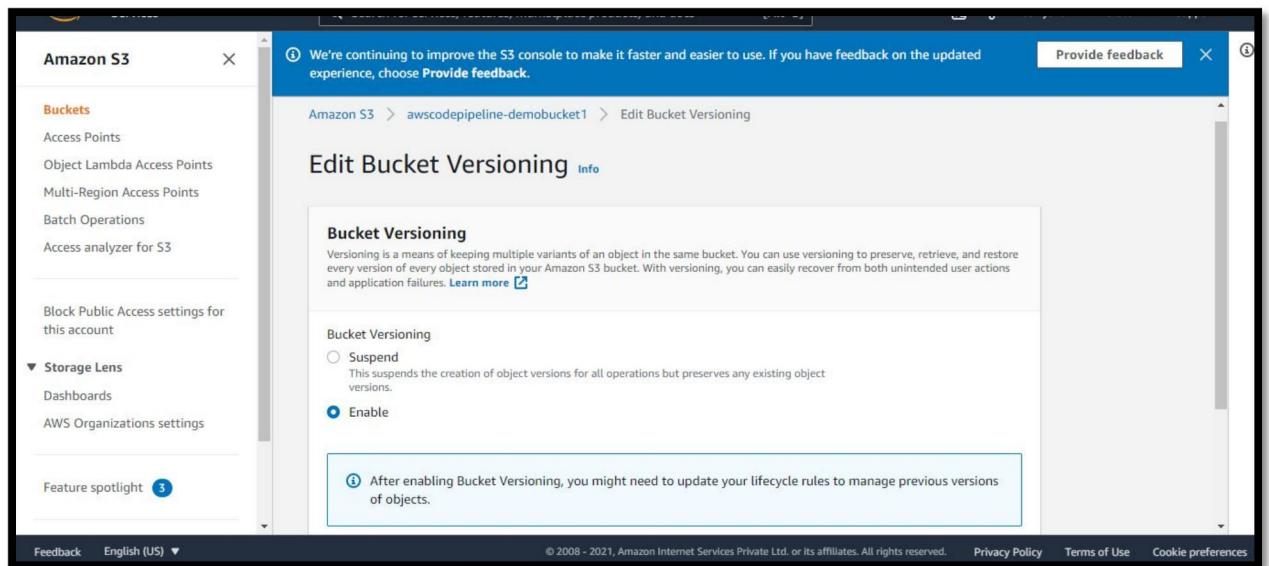


5. On the Properties tab, choose Versioning. Choose Enable versioning, and

then choose Save.
When versioning is enabled, Amazon S3 saves every version of every object in the bucket.

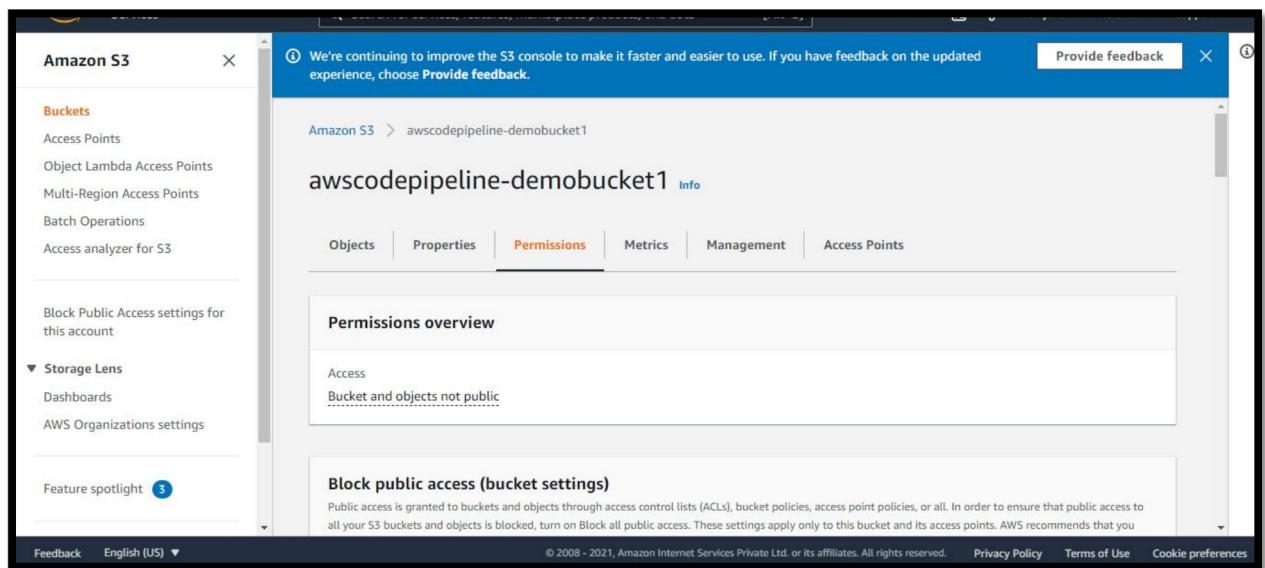


The screenshot shows the 'Bucket overview' section of the Amazon S3 console. At the top, there are tabs for Objects, Properties (which is selected), Permissions, Metrics, Management, and Access Points. Below the tabs, the 'Bucket overview' card displays basic information: AWS Region (Asia Pacific (Mumbai) ap-south-1), ARN (arn:aws:s3:::awscodepipeline-demobucket1), and Creation date (October 13, 2021, 11:01:35 (UTC+05:30)). Under the 'Bucket Versioning' section, it says 'Disabled'. There is an 'Edit' button next to the versioning status. A note at the bottom of the page states: 'Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more'.



The screenshot shows the 'Edit Bucket Versioning' page. The left sidebar has the same navigation as the previous screenshot. The main content area is titled 'Edit Bucket Versioning'. It contains a 'Bucket Versioning' section with a note: 'Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more'. Below this, there is a 'Bucket Versioning' dropdown with two options: 'Suspend' (radio button not selected) and 'Enable' (radio button selected). A callout box with an info icon says: 'After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.' The footer of the page includes standard copyright and legal links.

6. On the Permissions tab, leave the defaults. For more information about S3 bucket and object permissions, see Specifying Permissions in a Policy.



7. Next, download a sample and save it into a folder or directory on your local computer.
- Choose one of the following. Choose SampleApp_Windows.zip if you want to follow the steps in this tutorial for Windows Server instances.
 - If you want to deploy to Amazon Linux instances using CodeDeploy, download the sample application here: SampleApp_Linux.zip.
 - If you want to deploy to Windows Server instances using CodeDeploy, download the sample application here: SampleApp_Windows.zip.
 - Download the compressed (zipped) file. Do not unzip the file.

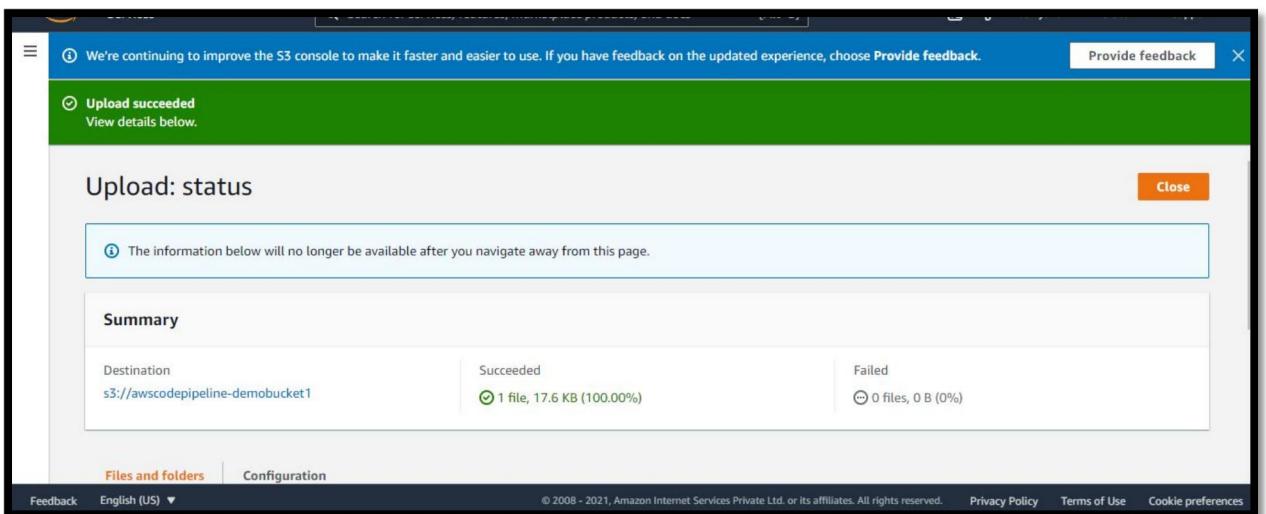
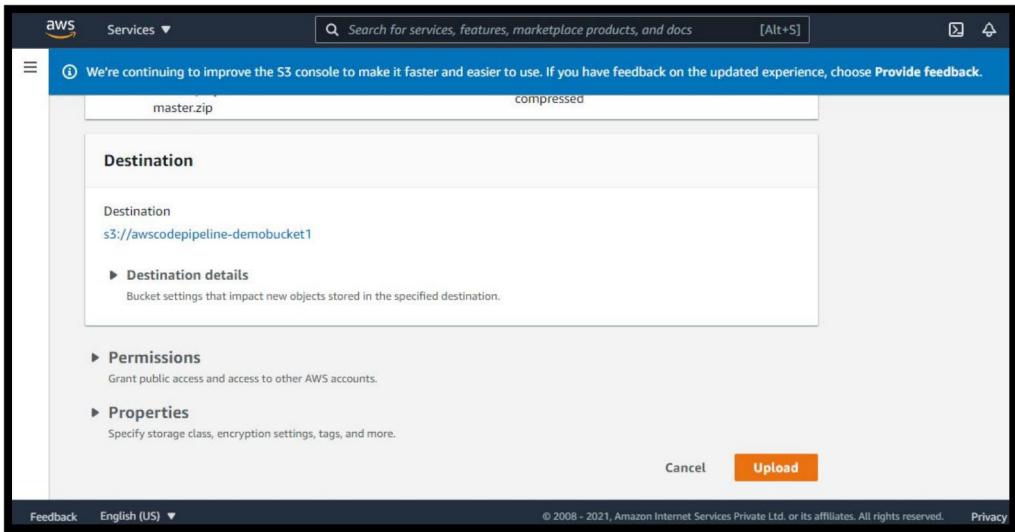
SampleApp_Windows.zip

8. In the Amazon S3 console, for your bucket, upload the file:
- Choose **Upload**.

2. Drag and drop the file or choose **Add files** and browse for the file.
3. Choose **Upload**.

The screenshot shows the AWS S3 console's upload interface. At the top, there is a message: "We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose [Provide feedback](#)". Below this, the path is shown as "Amazon S3 > awscodipeline-demobucket1 > Upload". The main title is "Upload [Info](#)". A descriptive text block says: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more [\[?\]](#)". Below this is a dashed box containing the instruction: "Drag and drop files and folders you want to upload here, or choose [Add files](#), or [Add folders](#)". A table titled "Files and folders (0)" shows a single entry: "All files and folders in this table will be uploaded.". The table has columns for "Name", "Folder", "Type", and "Size". At the bottom of the page, there are links for "Feedback", "English (US) ▾", and "Privacy".

This screenshot shows the same AWS S3 upload interface after a file has been added. The "Files and folders" table now contains one item: "aws-codepipeline-s3-codedeploy-linux-2.0-master.zip" (1 Total, 17.6 KB). The file details are shown in the table: Name (aws-codepipeline-s3-codedeploy-linux-2.0-master.zip), Type (application/x-zip-compressed), and Size (17.6 KB). The rest of the interface remains the same, including the message bar, path, and footer links.



Deploy Sample Application on EC2 instance using AWS CodeDeploy

To create an instance role

- 1. Open the IAM console at (<https://console.aws.amazon.com/iam/>)**
•

Identity and Access Management (IAM)

Introducing the new IAM dashboard experience
We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.

IAM dashboard

Security recommendations 1

Add MFA for root user

Enable multi-factor authentication (MFA) for the root user to improve security for this account.

Root user has no active access keys

Using access keys attached to an IAM user instead of the root user improves security.

AWS Account

Account ID: 977063044842

Account Alias: 977063044842 Create

Sign-in URL: https://977063044842.signin.amazonaws.com/console

IAM resources

User groups	Users	Roles	Policies	Identity providers
2	2	12	5	0

Quick Links

Feedback English (US) ▾

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

2. From the console dashboard, choose Roles.

Identity and Access Management (IAM)

Introducing the new Roles list experience
We've redesigned the Roles list experience to make it easier to use. Let us know what you think.

IAM > Roles

Roles (12) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

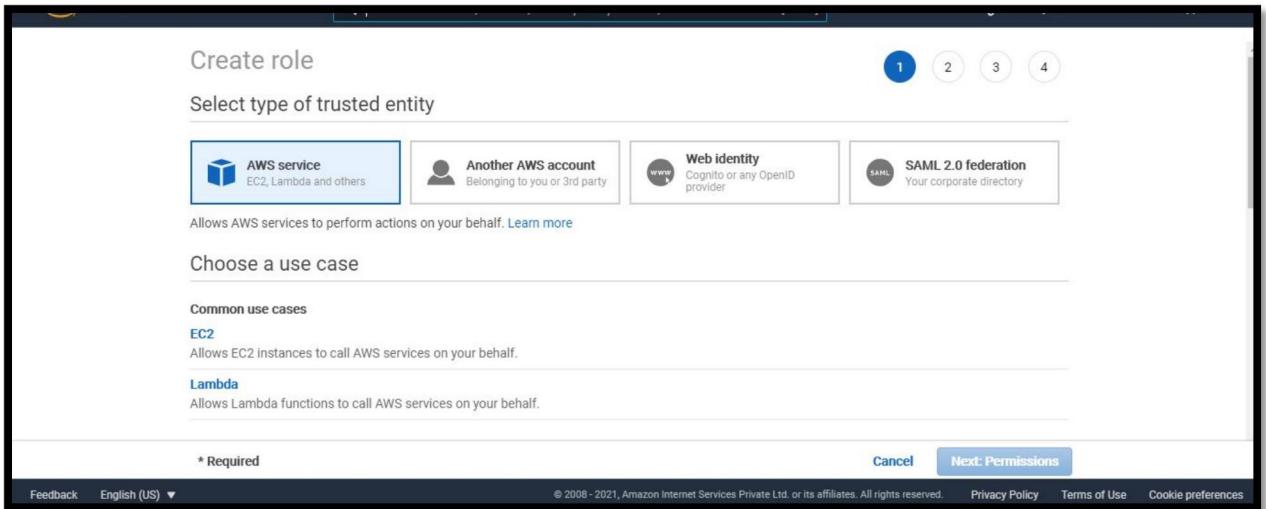
Role name	Trusted entities	Last act...
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	14 minutes ago
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	23 minutes ago
AWSCodePipelineServiceRole-ap-south-1-awspipeline	AWS Service: codepipeline	13 hours ago
AWSCodePipelineServiceRole-ap-south-1-Demopipeline	AWS Service: codepipeline	13 hours ago

Create role

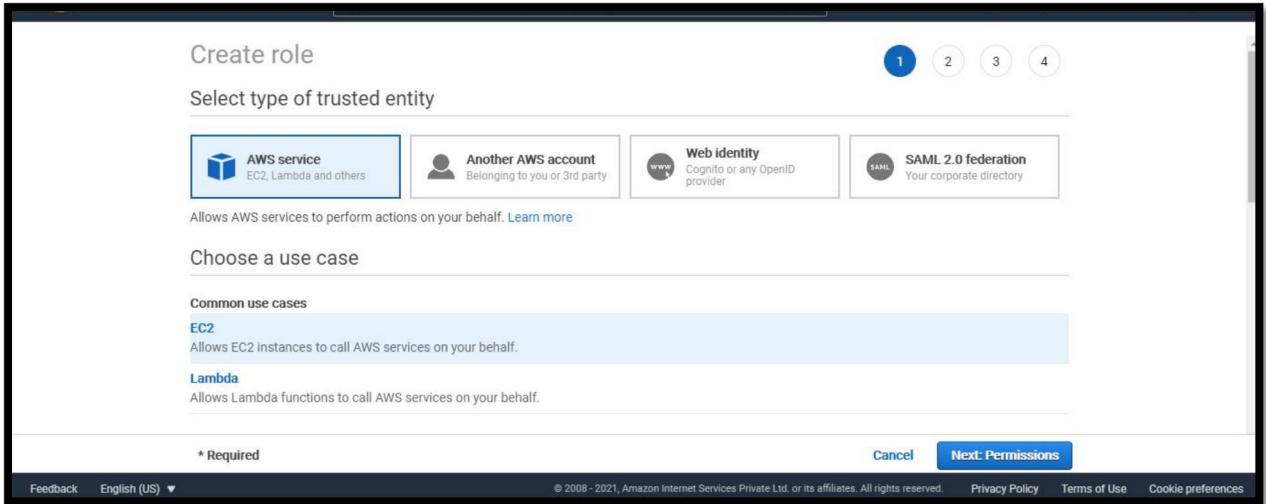
Feedback English (US) ▾

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

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, and then choose Next: Tags.

The screenshot shows the 'Create role' wizard, Step 2: Attach permissions policies. A search bar filters results for 'AmazonEC2R'. A policy named 'AmazonEC2RoleforAWSCodeDeploy' is selected (indicated by a checked checkbox). Other policies listed include 'AmazonEC2ReadOnlyAccess', 'AmazonEC2RoleforAWSCodeDeployLimited', 'AmazonEC2RoleforDataPipelineRole', and 'AmazonEC2RoleforSSM'. The interface includes tabs for 'Create policy', 'Previous', 'Next: Tags', and 'Cancel'.

6. Choose Next: Review. Enter a name for the role (for example, EC2InstanceRole).

Choose Create role.

The screenshot shows the 'Create role' wizard, Step 4: Review. The role name is 'EC2InstanceRole'. The role description is 'Allows EC2 instances to call AWS services on your behalf.' The trusted entity is 'AWS service: ec2.amazonaws.com'. Policies attached are 'AmazonEC2RoleforAWSCodeDeploy'. The interface includes tabs for 'Cancel', 'Previous', and 'Create role'.

The screenshot shows the AWS Identity and Access Management (IAM) console. On the left, there's a navigation sidebar with options like Dashboard, Access management, Roles (which is selected), Policies, Identity providers, and Account settings. Under Access management, there are sub-options for User groups, Roles, Policies, and Account settings. Below that is a section for Access reports with options for Access analyzer, Archive rules, Analyzers, and Settings. At the bottom of the sidebar, there's a Credential report. The main content area has a blue header bar with the text "Introducing the new Roles list experience" and "We've redesigned the Roles list experience to make it easier to use. Let us know what you think." Below this, a green success message box says "The role EC2InstanceRole has been created." The main table title is "Roles (13) Info". It describes an IAM role as an identity that can be created with specific permissions. The table lists three roles: "aws-elasticbeanstalk-ec2-role" (AWS Service: ec2, Last act... 13 minutes ago), "aws-elasticbeanstalk-service-role" (AWS Service: elasticbeanstalk, Last act... 22 minutes ago), and "AWSCodePipelineServiceRole-ap-south-1-awspipeline" (AWS Service: codepipeline, Last act... 13 hours ago). There are buttons for "Edit", "Delete", and "Create role". A search bar is at the top of the table. The footer includes links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

To launch instances

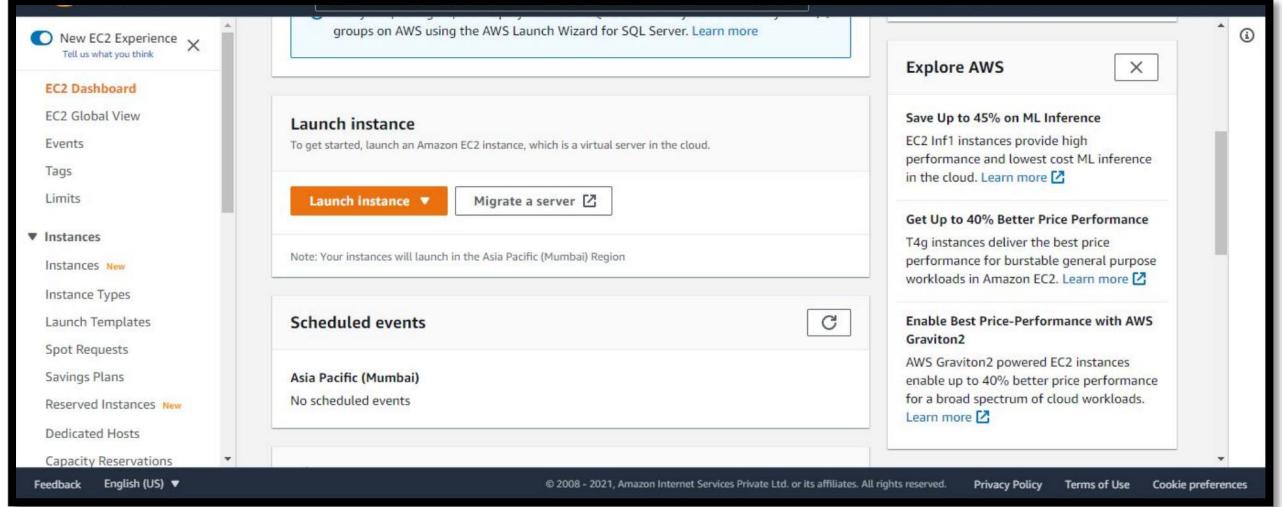
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with "New EC2 Experience" (Tell us what you think), "EC2 Dashboard", "Events", "Tags", "Limits", and a "Instances" section containing "Instances", "Instance Types", "Launch Templates", "Spot Requests", "Savings Plans", "Reserved Instances", "Dedicated Hosts", and "Capacity Reservations". The main content area has a "Resources" section with a table showing the following data:

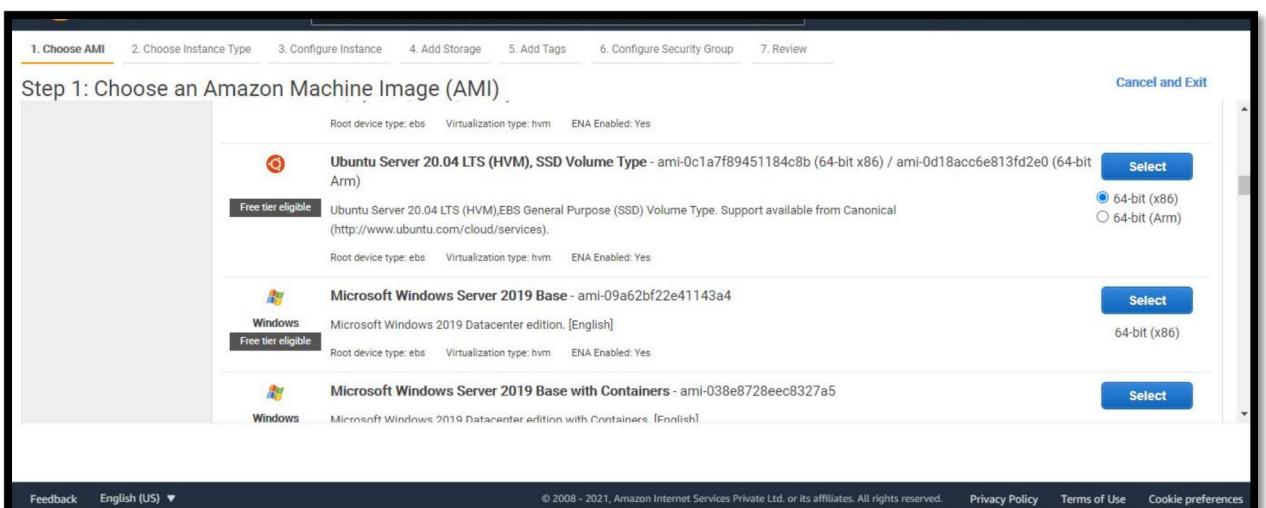
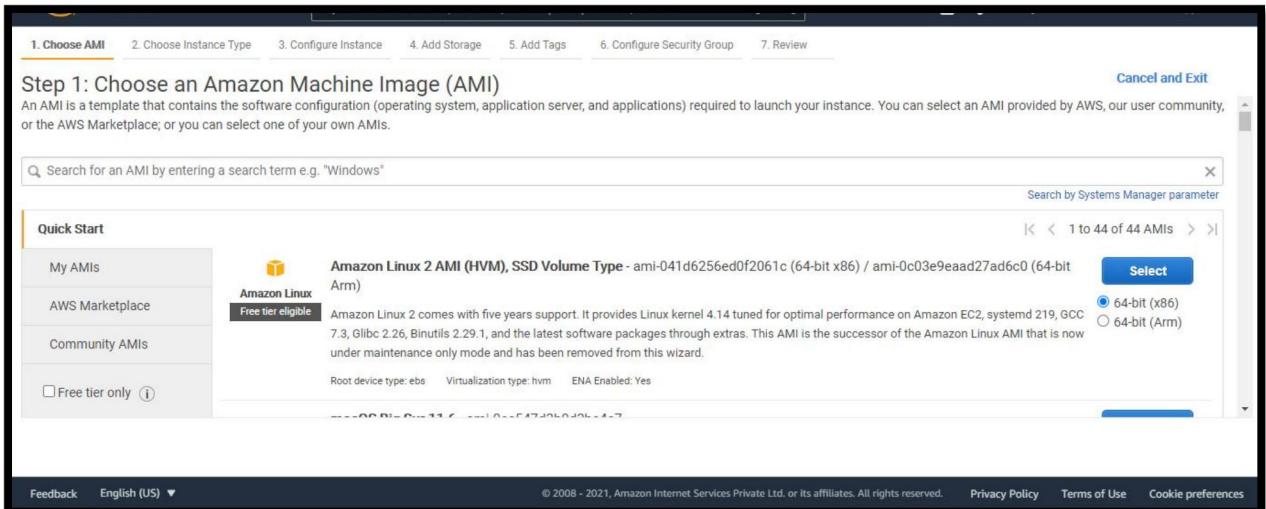
Instances (running)	3	Dedicated Hosts	0
Elastic IPs	0	Instances	4
Key pairs	1	Load balancers	1
Placement groups	0	Security groups	5
Snapshots	0	Volumes	4

A callout box in the center of the dashboard says "Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more". To the right, there's an "Account attributes" section with "Supported platforms" (VPC, Default VPC vpc-1610cd7d), "Settings" (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and an "Explore AWS" section. The footer includes links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

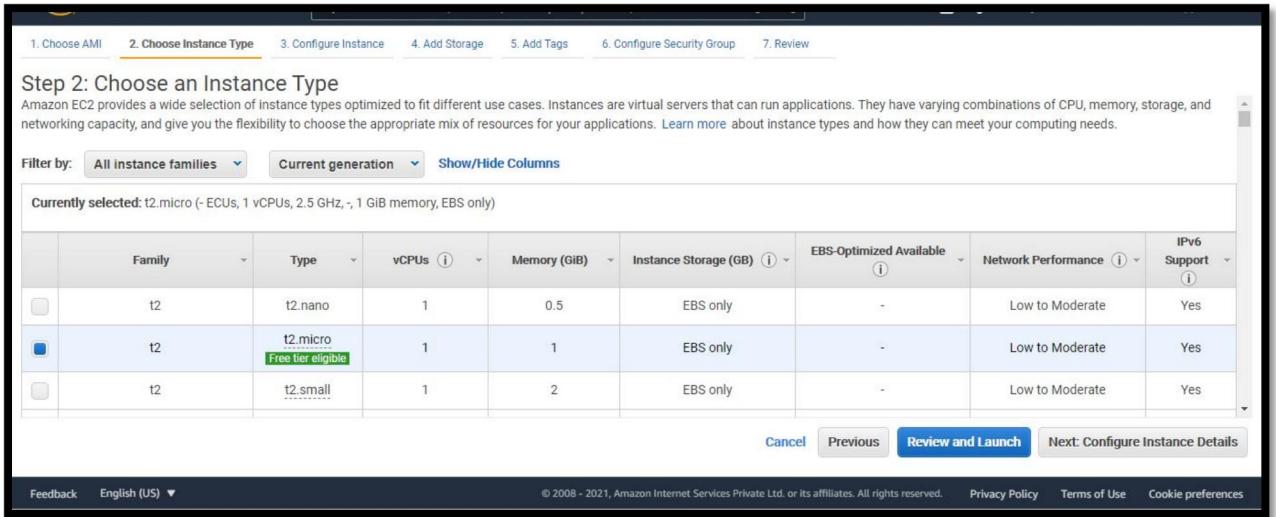
2. From the console dashboard, choose Launch instance, and select Launch instance from the options that pop up.



3. On the Step 1: Choose an Amazon Machine Image (AMI) page, locate the Microsoft Windows Server 2019 Base option, and then choose Select. (This AMI is labeled "Free tier eligible" and can be found at the top of the list.)



- 4.** On the Step 2: Choose an Instance Type page, choose the free tier eligible t2.micro type as the hardware configuration for your instance, and then choose Next: Configure Instance Details.



5. On the Step 3: Configure Instance Details page, do the following:

- In Number of instances, enter 2.
- In Auto-assign Public IP, choose Enable.
- In IAM role, choose the IAM role you created in the previous procedure (for example, EC2InstanceRole).
- Expand Advanced Details, and in User data, with As text selected, enter the following:

```
<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>
```

- *bucket-name* is the name of the S3 bucket that contains the CodeDeploy Resource Kit files for your Region. For example, for the US West (Oregon) Region, replace *bucket-name* with aws-codedeploy-us-west-2. For a list of bucket names, see [Resource Kit Bucket Names by Region](#).

This code installs the CodeDeploy agent on your instance as it is created. This script is written for Windows instances only.

- Leave the rest of the items on the Step 3: Configure Instance Details page unchanged. Choose Next: Add Storage.

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 instances Launch into Auto Scaling Group [\(i\)](#)

You may want to consider launching these instances into an Auto Scaling Group to help you maintain application availability and for easy scaling in the future. Learn how Auto Scaling can help your application stay healthy and cost effective.

Purchasing option [\(i\)](#) Request Spot instances

Network [\(i\)](#) vpc-1610cd7d (default) [Create new VPC](#)

Subnet [\(i\)](#) No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP [\(i\)](#) Enable

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

Step 3: Configure Instance Details

Placement group [\(i\)](#) Add instance to placement group

Capacity Reservation [\(i\)](#) Open

Domain join directory [\(i\)](#) No directory [Create new directory](#)

IAM role [\(i\)](#) EC2InstanceRole [Create new IAM role](#)

Shutdown behavior [\(i\)](#) Stop

Stop - Hibernate behavior [\(i\)](#) Enable hibernation as an additional stop behavior

Enable termination protection [\(i\)](#) Protect against accidental termination

Monitoring [\(i\)](#) Enable CloudWatch detailed monitoring
Additional charges apply.

Buttons: Cancel, Previous, **Review and Launch**, Next: Add Storage

Step 3: Configure Instance Details

Advanced Details

Enclave Enable

Metadata accessible Enabled

Metadata version V1 and V2 (token optional)

Metadata token response hop limit 1

User data As text

```
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>
```

Cancel Previous Review and Launch Next: Add Storage

- Leave the Step 4: Add Storage page unchanged, and then choose Next: Add Tags.

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/sda1	snap-0e176080baecbd8f8	30	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

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

- 7.** On the Add Tags page, choose Add Tag. Enter Name in the Key field, enter MyCodePipelineDemo in the Value field, and then choose Next: Configure Security Group.

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	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name		MyCodePipelineDemo		<input checked="" type="checkbox"/>	<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

- 8.** On the Configure Security Group page, allow port 80 communication so you can access the public instance endpoint.

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:

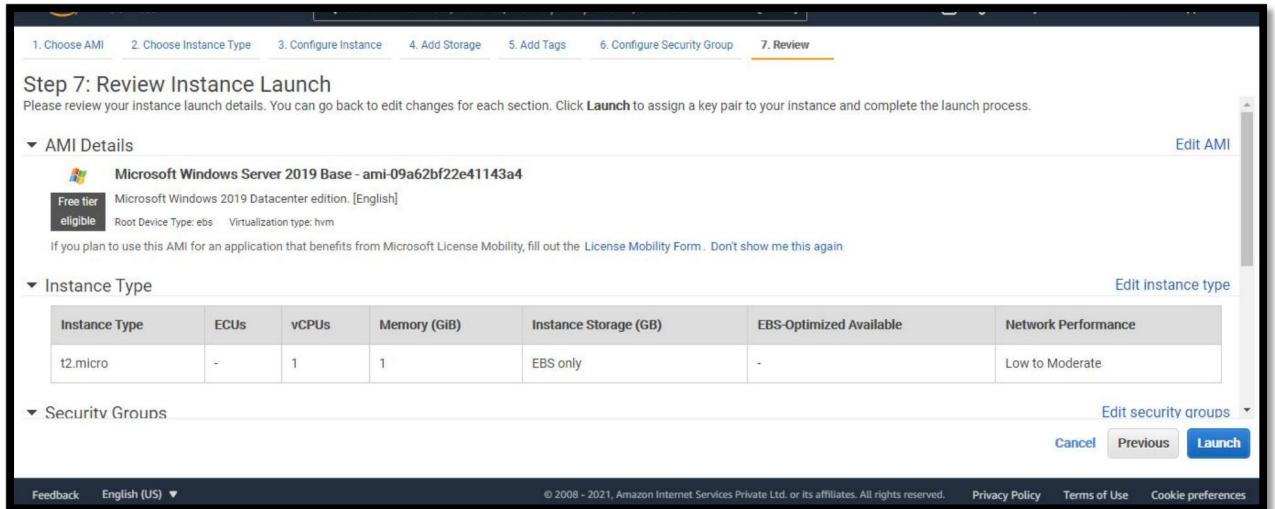
Description:

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	My IP	106.209.184.197/32 e.g. SSH for Admin Desktop

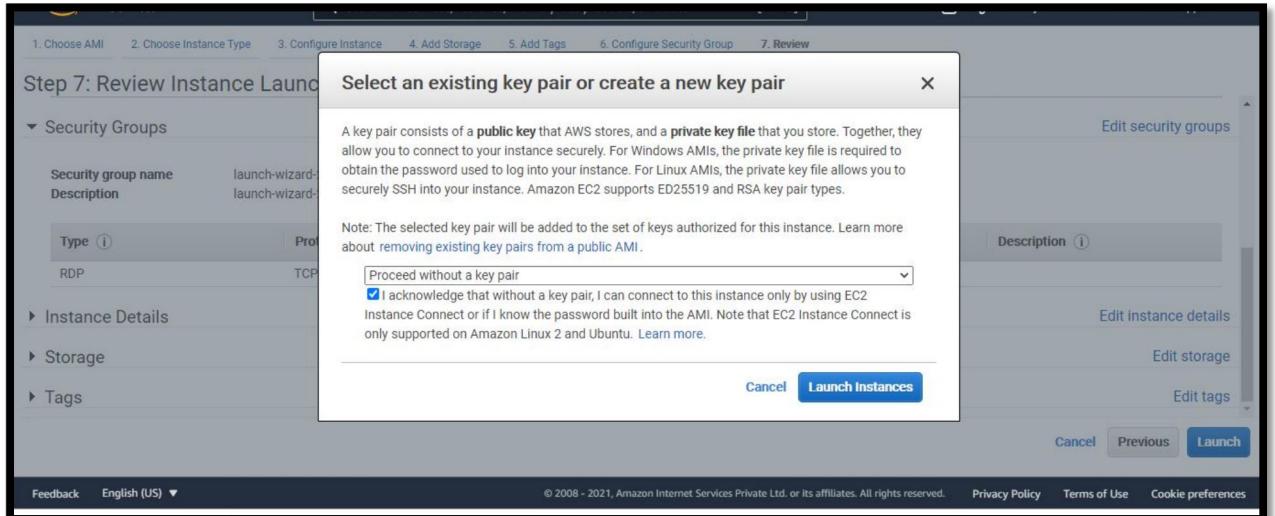
Add Rule

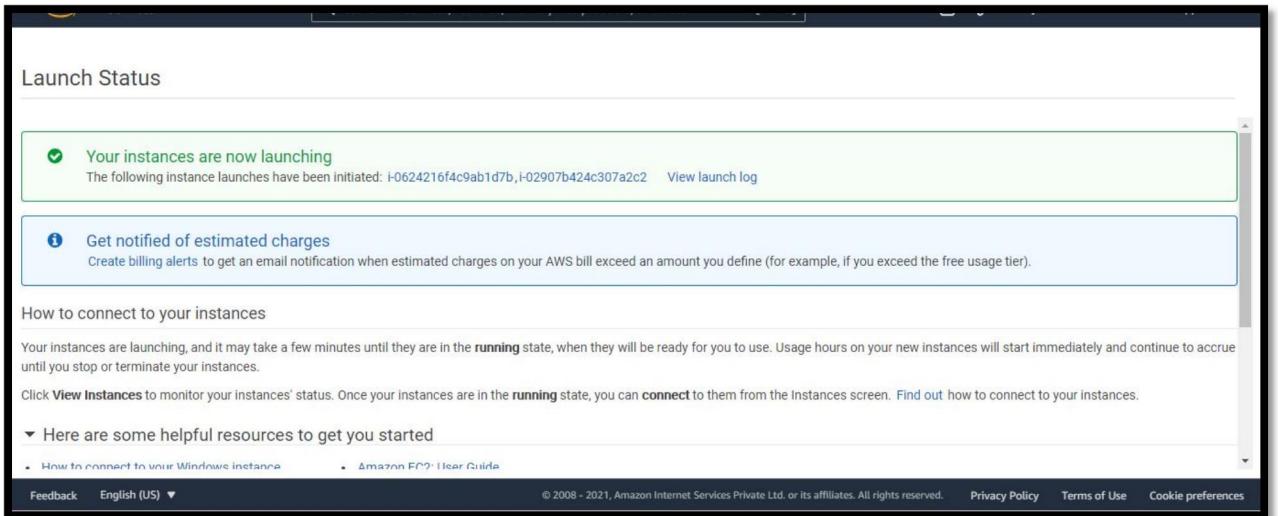
Cancel Previous Review and Launch Next: Review

9. Choose Review and Launch.



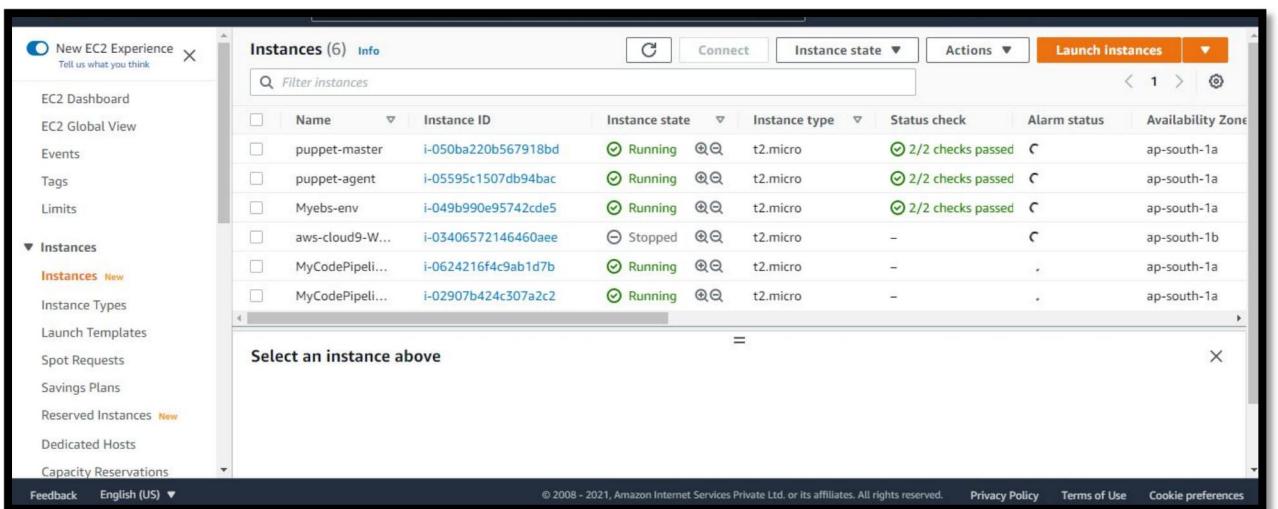
10. On the Review Instance Launch page, choose Launch. When prompted for a key pair, choose Proceed without a key pair. When you are ready, select the acknowledgment check box, and then choose Launch Instances.





11. Choose View Instances to close the confirmation page and return to the console.

12. You can view the status of the launch on the Instances page. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running, and it receives a public DNS name. (If the Public DNS column is not displayed, choose the Show/Hide icon, and then select Public DNS.)



13. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks. You can view this information in the Status Checks column.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Feedback, English (US), and a 'Tell us what you think' link. The main table lists six instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
puppet-master	i-050ba220b567918bd	Running	t2.micro	2/2 checks passed	C	ap-south-1a
puppet-agent	i-05595c1507db94bac	Running	t2.micro	2/2 checks passed	C	ap-south-1a
Myeks-env	i-049b990e95742cde5	Running	t2.micro	2/2 checks passed	C	ap-south-1a
aws-cloud9-W...	i-03406572146460aee	Stopped	t2.micro	-	C	ap-south-1b
MyCodePipelineDemo	i-0624216f4c9ab1d7b	Running	t2.micro	-	-	ap-south-1a

A modal window titled "Instance: i-0624216f4c9ab1d7b (MyCodePipelineDemo)" displays detailed information for the selected instance. It includes fields for Instance ID, Public IPv4 address (13.234.240.58), Private IPv4 addresses (172.31.35.9), IPv6 address (-), Instance state (Running), Public IPv4 DNS (ec2-13-234-240-58.ap-south-1.compute.amazonaws.com), and a "Log on address" link.

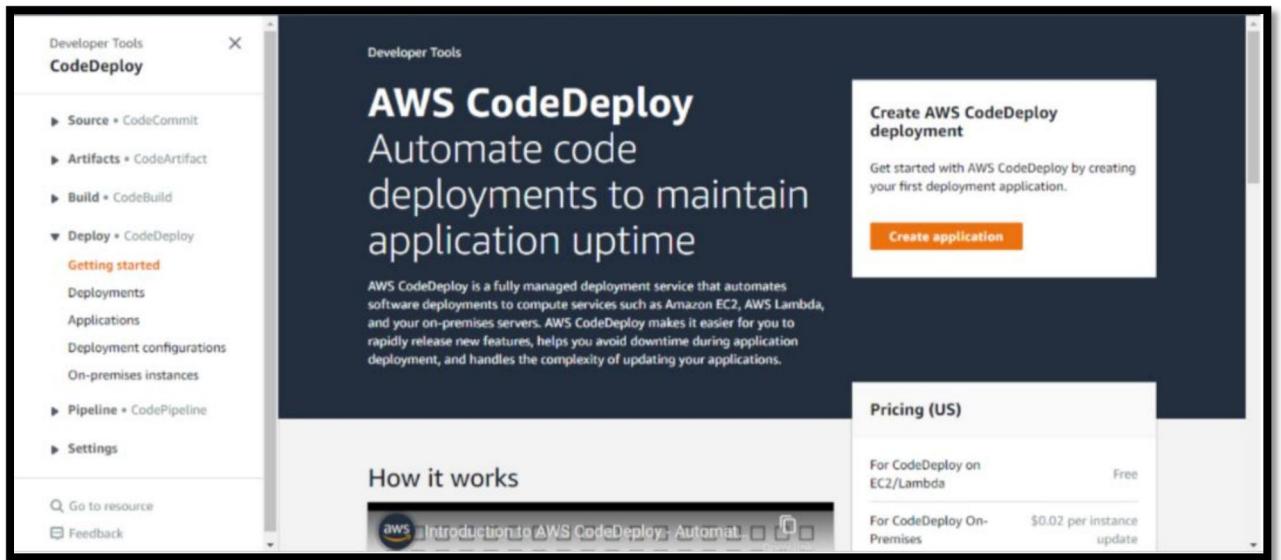
The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (selected), and AMIs. The main table lists three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
aws-cloud9-W...	i-0571401829c430e6c	Stopped	t2.micro	-	-	ap-south-1a
MyCodePipelineDemo	i-01dead92e3ef572c5	Running	t2.micro	2/2 checks passed	-	ap-south-1a
MyCodePipelineDemo	i-0c5b6fc63bd374e18	Running	t2.micro	2/2 checks passed	-	ap-south-1a

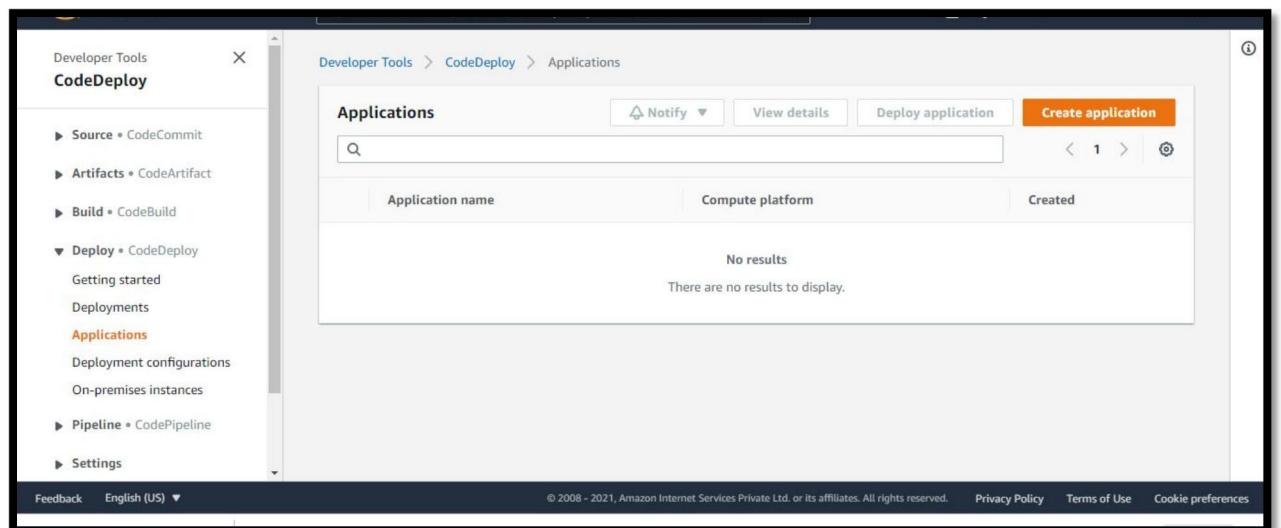
A modal window titled "Instance: i-01dead92e3ef572c5 (MyCodePipelineDemo)" displays the "Status checks" section. It includes a table comparing System status checks (System reachability check passed) and Instance status checks (Instance reachability check passed). A "Report instance status" button is at the bottom.

To create an application in CodeDeploy

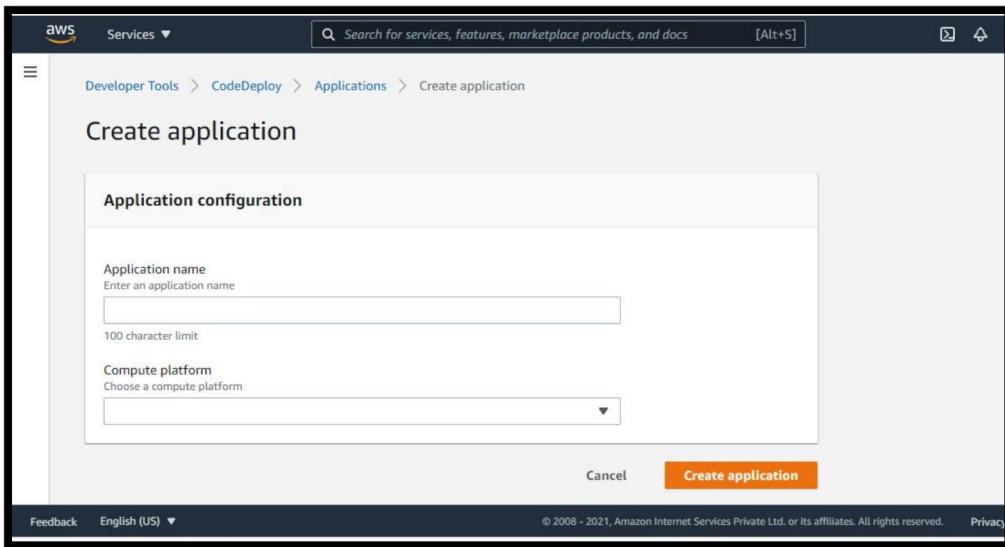
1. Open the CodeDeploy console at [https://console.aws.amazon.com/codedeploy.](https://console.aws.amazon.com/codedeploy)



2. If the Applications page does not appear, on the AWS CodeDeploy menu, choose Applications.

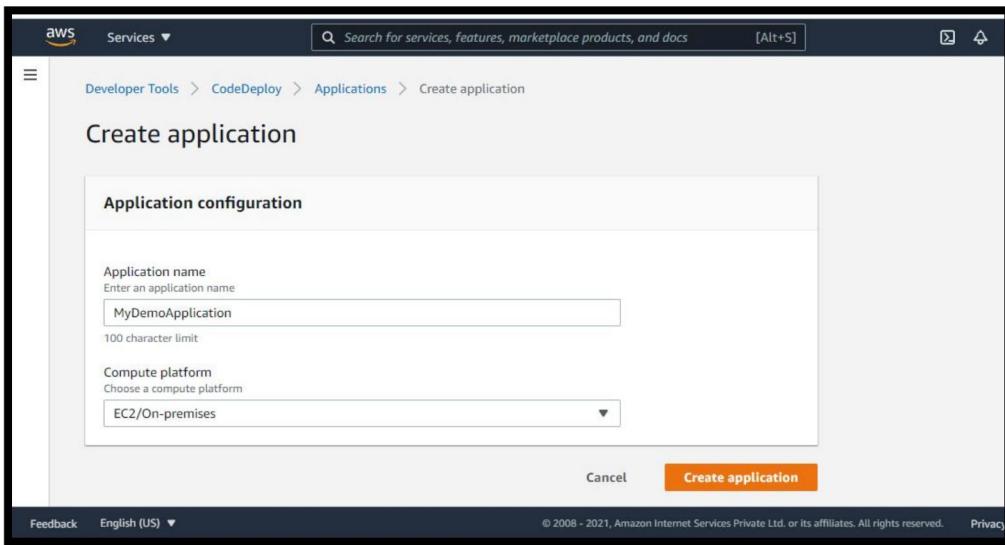


3. Choose Create application.

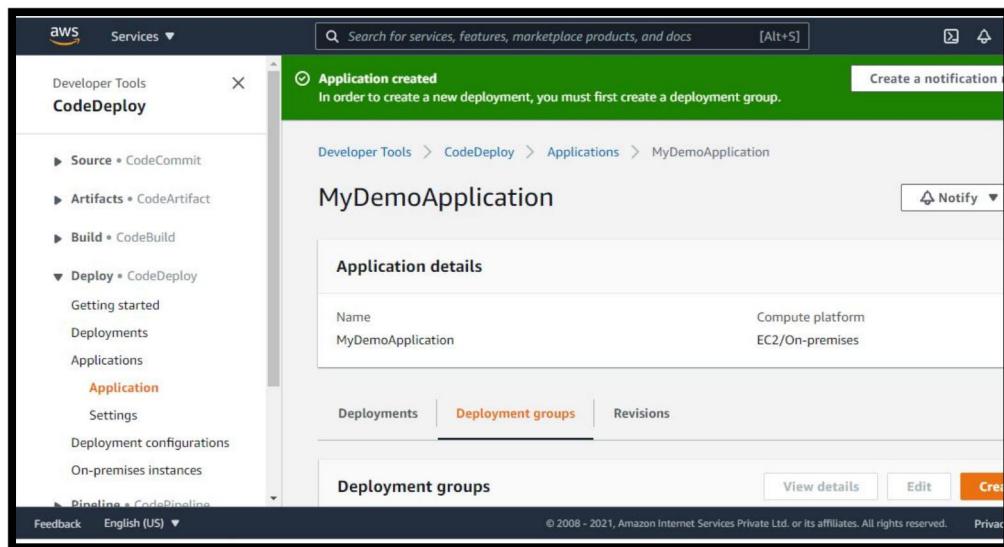


4. In Application name, enter MyDemoApplication.

5. In Compute Platform, choose EC2/On-premises.

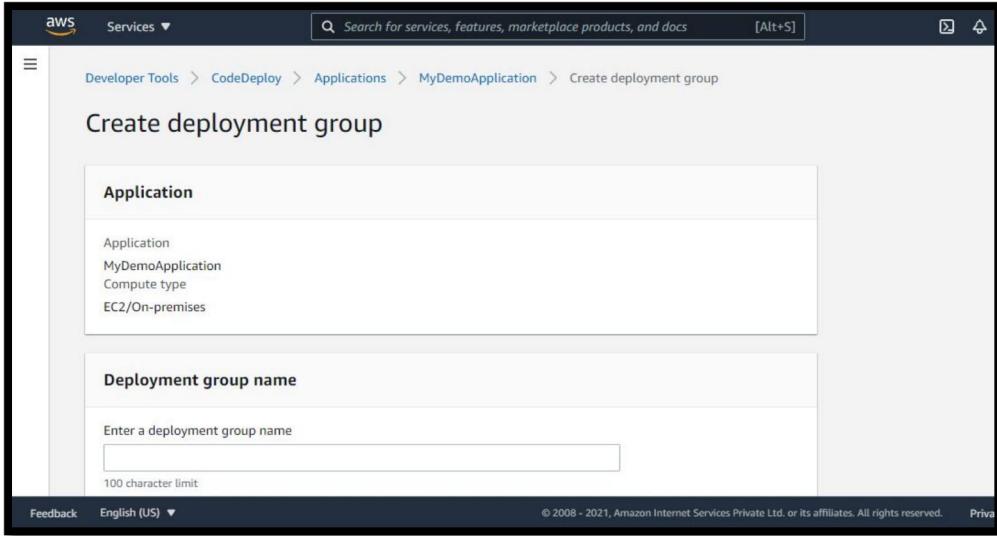


6. Choose Create application.

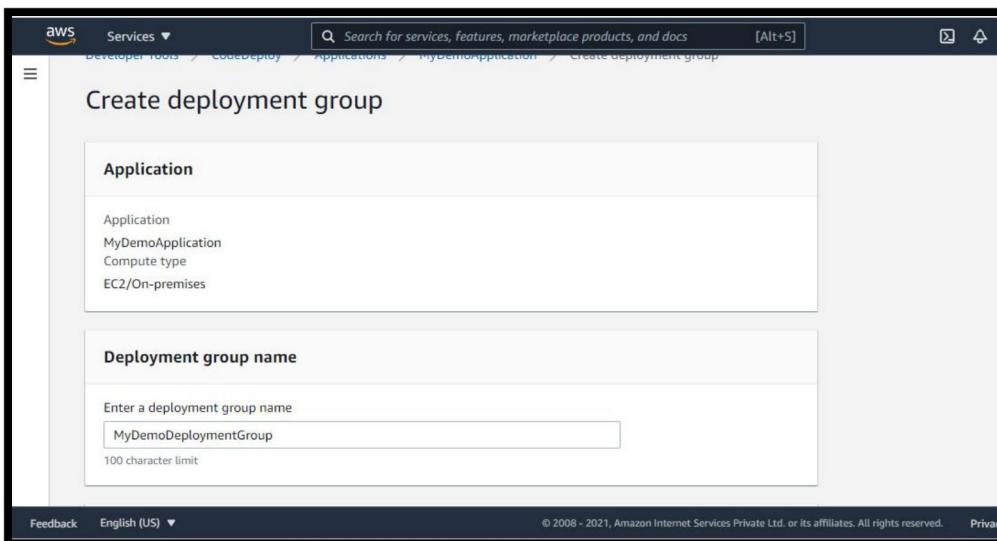


To create a deployment group in CodeDeploy

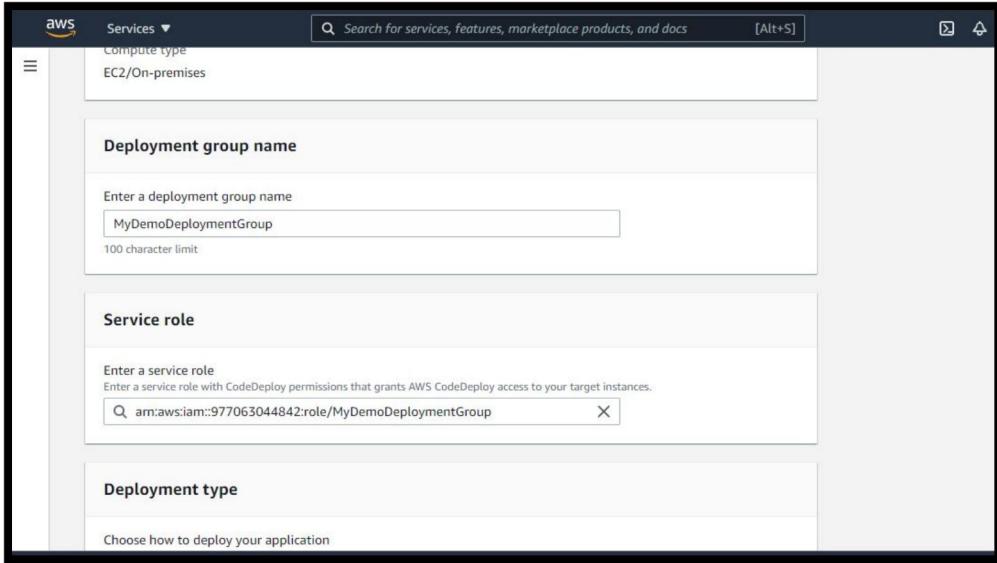
1. On the page that displays your application, choose Create deployment group.



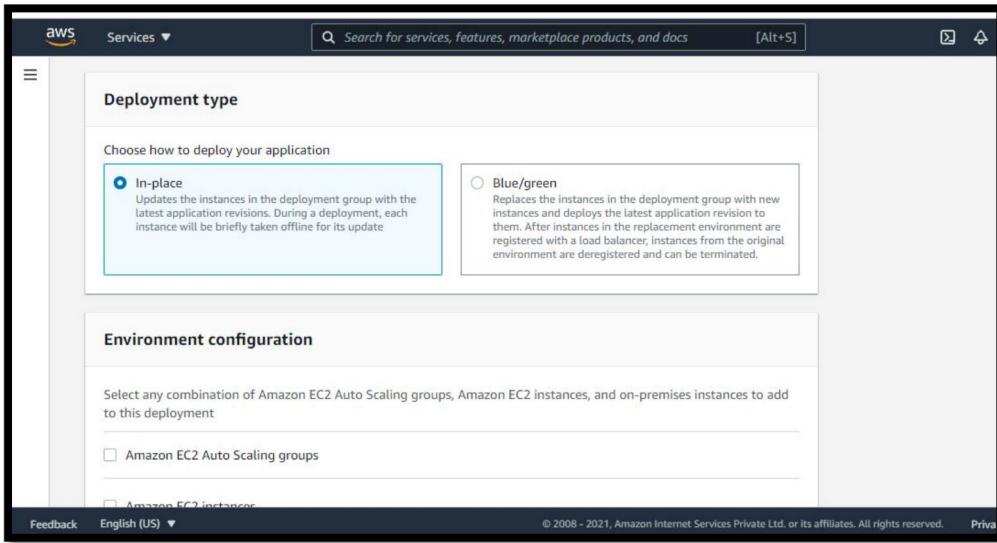
2. In Deployment group name, enter MyDemoDeploymentGroup.



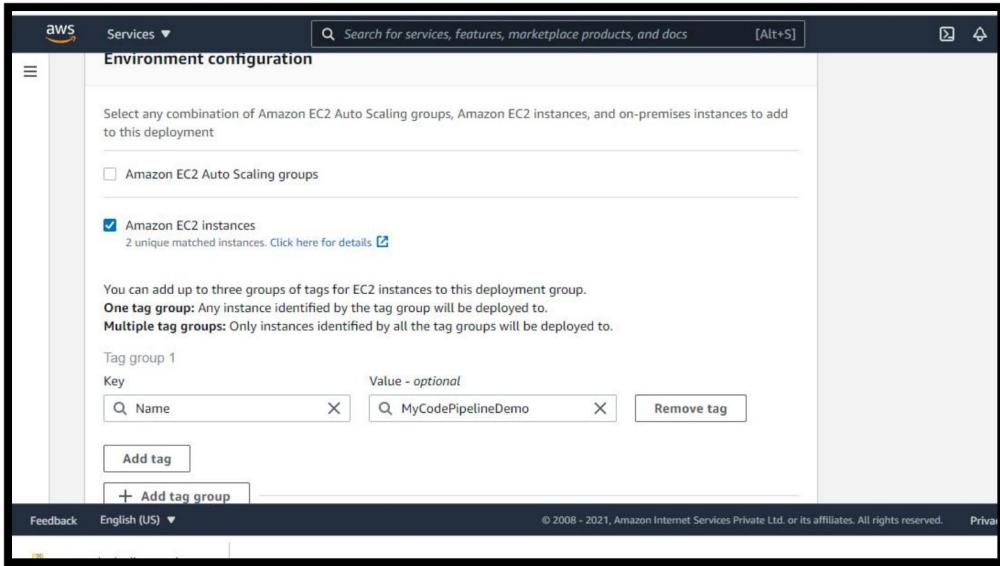
3. In Service Role, choose a service role that trusts AWS CodeDeploy with, at minimum, the trust and permissions described in Create a Service Role for CodeDeploy. To get the service role ARN, see Get the Service Role ARN (Console).



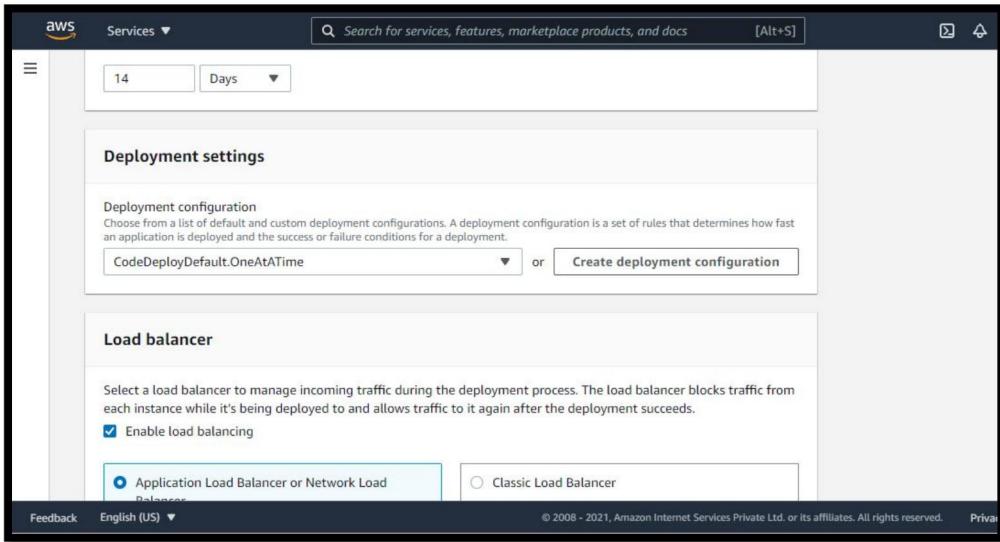
4. Under Deployment type, choose In-place.



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

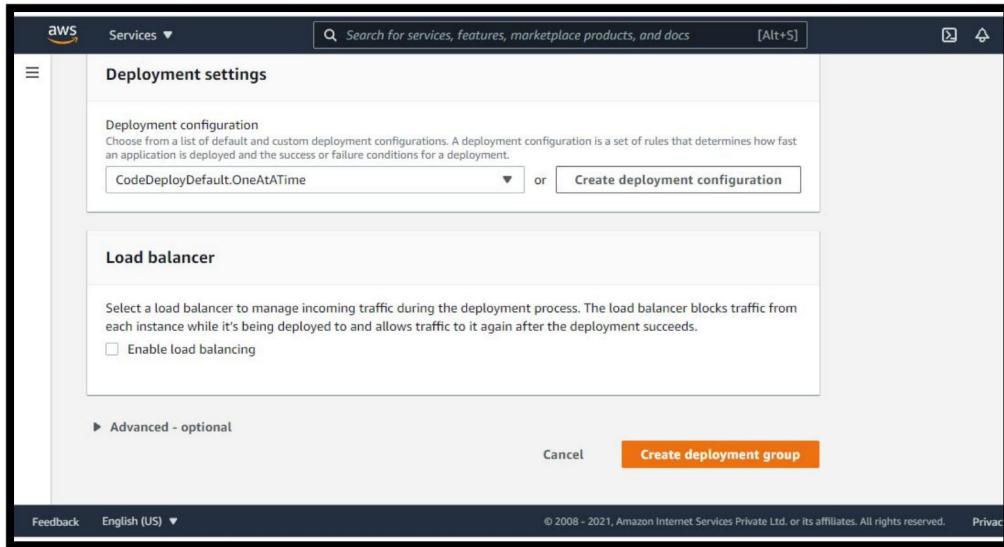


6. Under Deployment settings, choose CodeDeployDefault.OneAtATime.

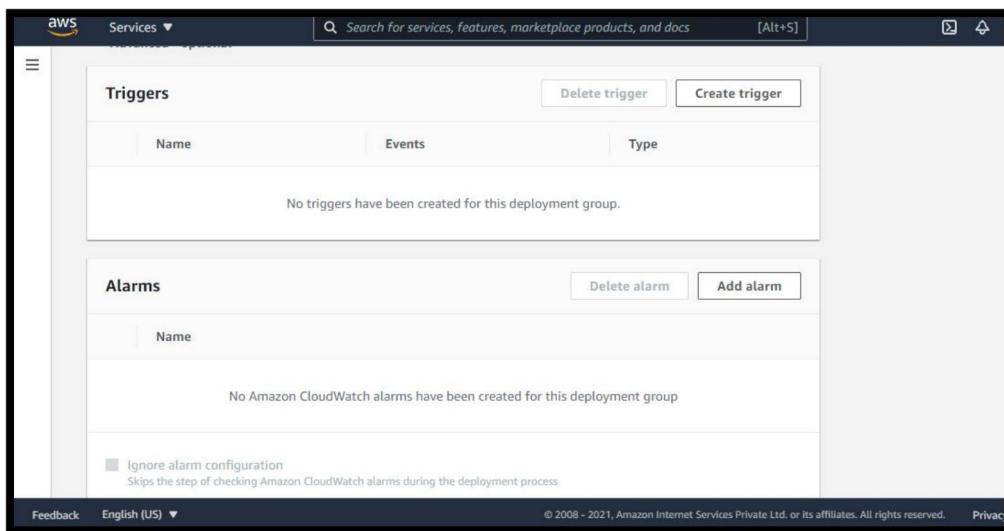


7. Under Load Balancer, make sure the Enable load balancing box is not selected. You do not need to set up a load balancer or choose a target group

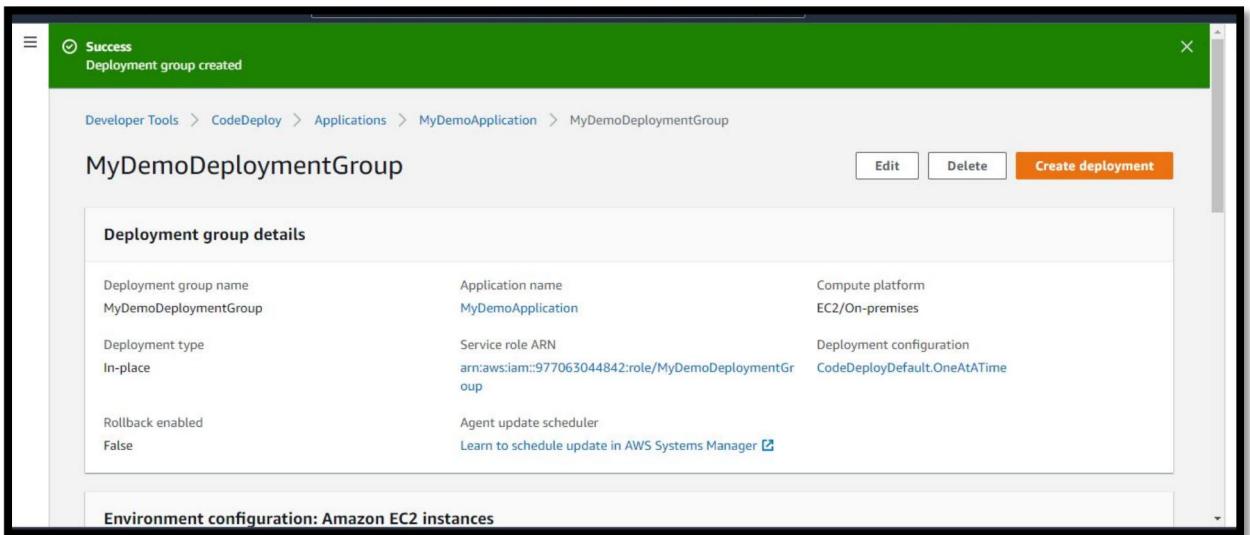
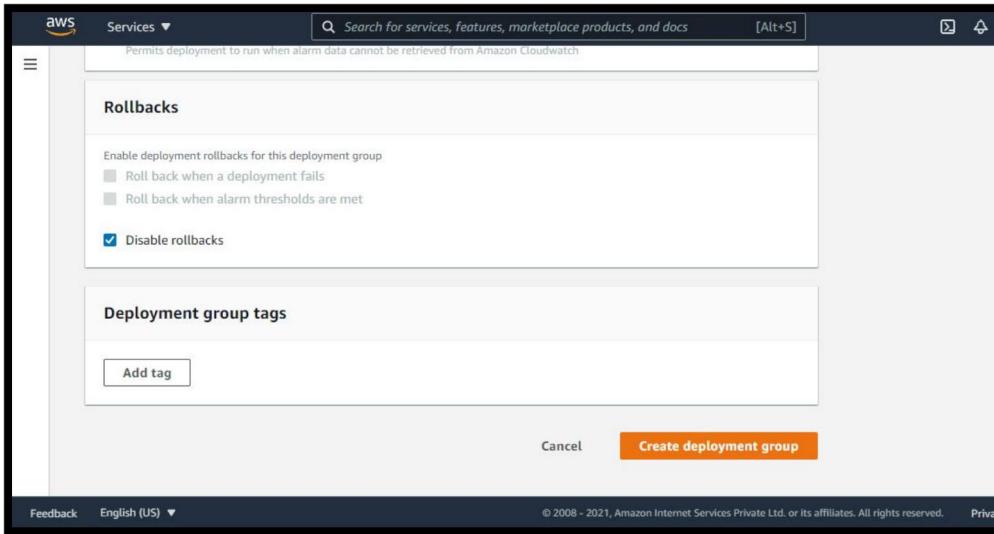
for this example. After you deselect the checkbox, the load balancer options do not display.



8. In the Advanced section, leave the defaults.



9. Choose Create deployment group.



Conclusion :-

We have successfully Build an Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.