

# **Advanced DevOps Lab**

## **Experiment 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:

Continuous deployment allows you to deploy revisions to a production environment automatically

without explicit approval from a developer, making the entire software release process automated.

You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your

code every time there is a code change. You will use your GitHub account, an Amazon Simple

Storage Service (S3) bucket, or an AWS CodeCommit repository as the source location for the

sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the

sample app. Your completed pipeline will be able to detect changes made to the source repository

containing the sample app and then automatically update your live sample app.

Steps:

- 1.Create a role in an IAM.
- 2.Add EC2 for a service or use case.
- 3.Give name to the role.

The screenshot shows the AWS IAM console interface for creating a new role. The breadcrumb navigation indicates the path: IAM > Roles > Create role. The left sidebar shows the progress: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area is titled 'Name, review, and create' and contains a 'Role details' section. In this section, the 'Role name' field is filled with 'Saachi\_Iam' and has a description: 'Allows EC2 instances to call AWS services on your behalf.' Below this, the 'Trust policy' section shows a JSON snippet for the role's trust policy, which allows EC2 instances to assume the role. The bottom of the page shows 'Step 1: Select trusted entities' with an 'Edit' button.

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
  
Maximum 64 characters. Use alphanumeric and '+, @, \_' characters.

**Description**  
Add a short explanation for this role.  
  
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+@-/\[\]!#\$%^&\*~'

**Step 1: Select trusted entities** Edit

**Trust policy**

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Effect": "Allow",  
6       "Action": "sts:AssumeRole",  
7       "Principal": {  
8         "Service": "ec2.amazonaws.com"  
9       }  
10    ]  
11 }
```

4.Required policies (permissions) to be added while creating IAM user.

The screenshot shows the 'Step 2: Add permissions' section of the AWS IAM console. It features a 'Permissions policy summary' table with three rows, each representing an AWS managed policy attached to the role. Below the table, the 'Step 3: Add tags' section is visible, showing a message that no tags are currently associated with the resource and a button to 'Add new tag'. The bottom of the page includes 'Cancel', 'Previous', and 'Create role' buttons.

**Step 2: Add permissions** Edit

**Permissions policy summary**

Policy name	Type	Attached as
<a href="#">AWSElasticBeanstalkMulticontainerDocker</a>	AWS managed	Permissions policy
<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Permissions policy
<a href="#">AWSElasticBeanstalkWorkerTier</a>	AWS managed	Permissions policy

**Step 3: Add tags**

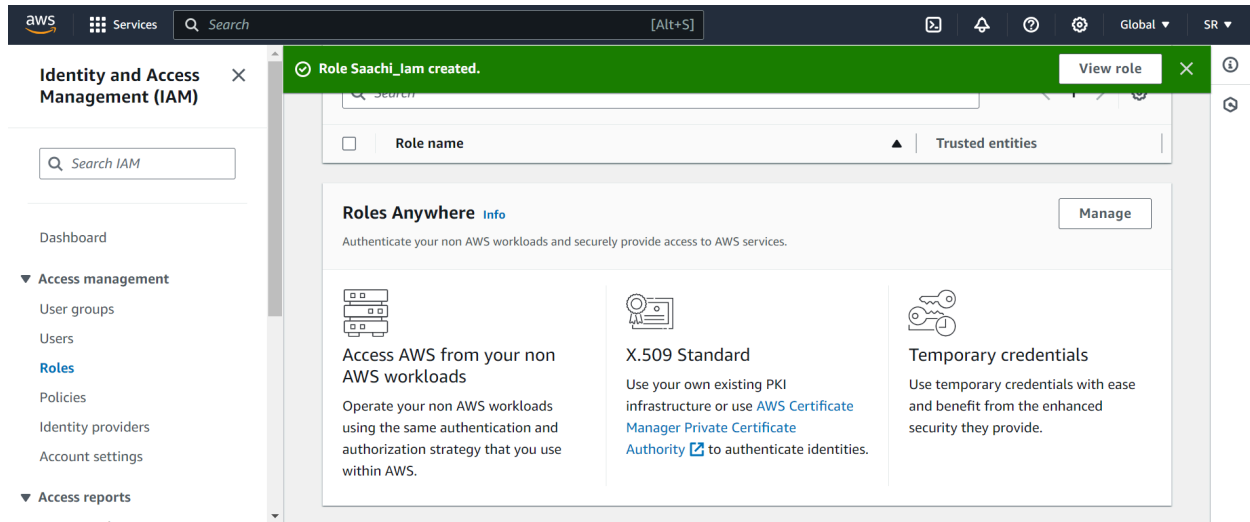
**Add tags - optional** [Info](#)  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

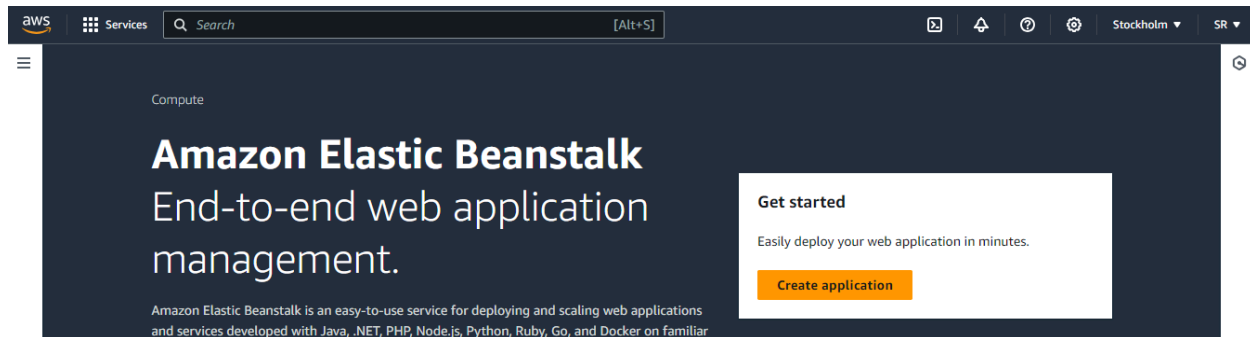
Add new tag  
You can add up to 50 more tags.

Cancel Previous Create role

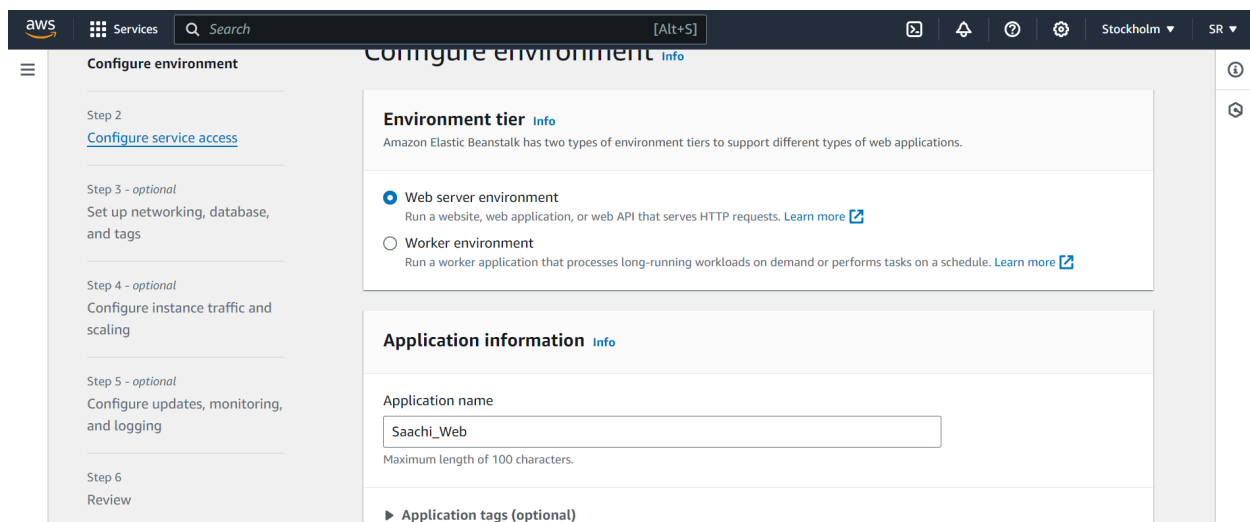
5.IAM Role Created



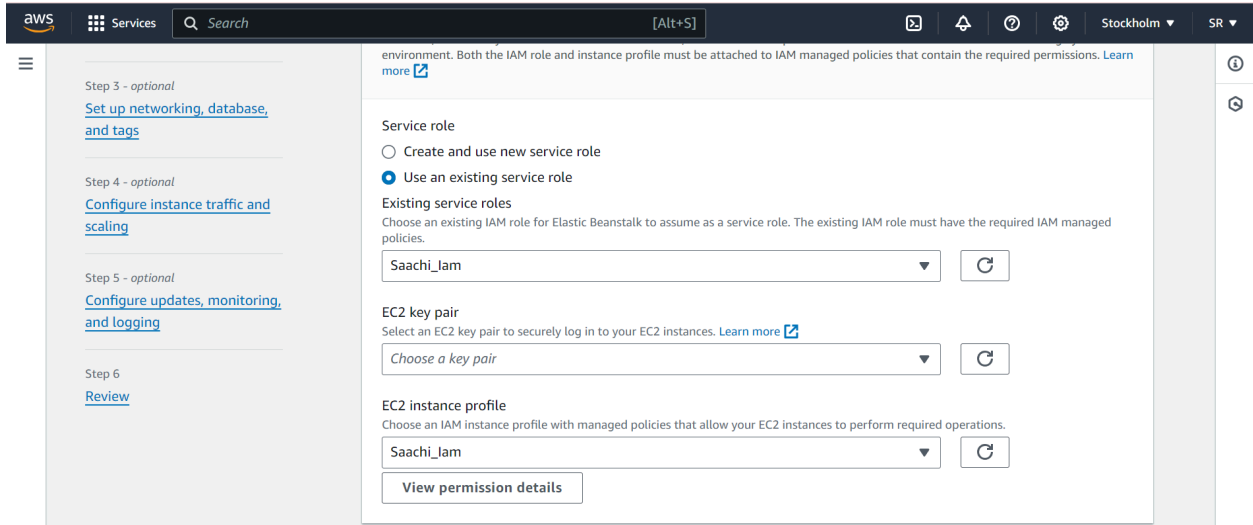
6. Go to the Elastic beanstalk and create an application. Give the appropriate name for the application.



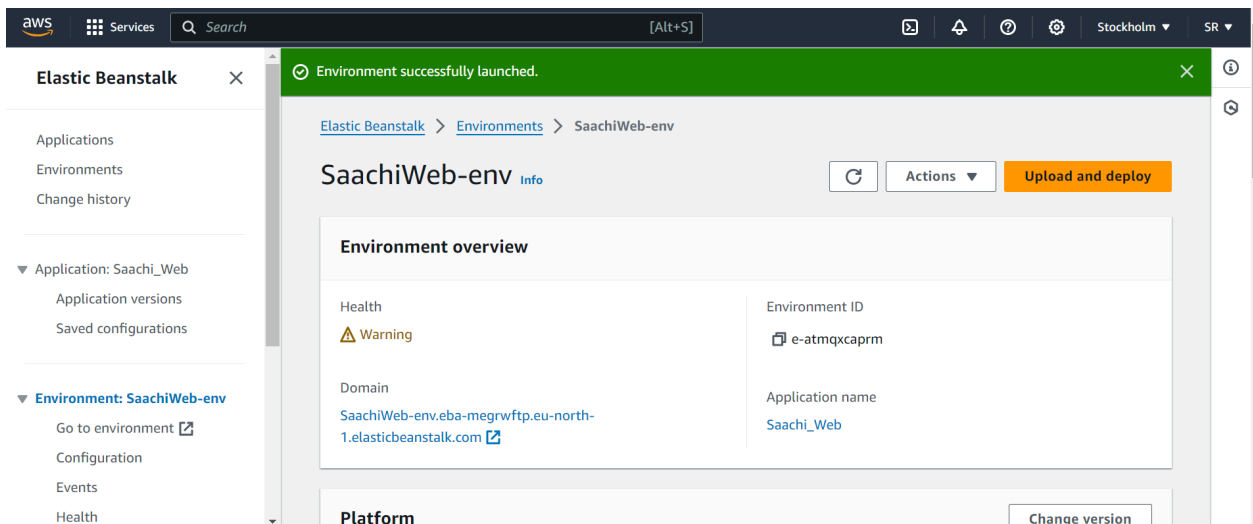
7. Select the platform as PHP.



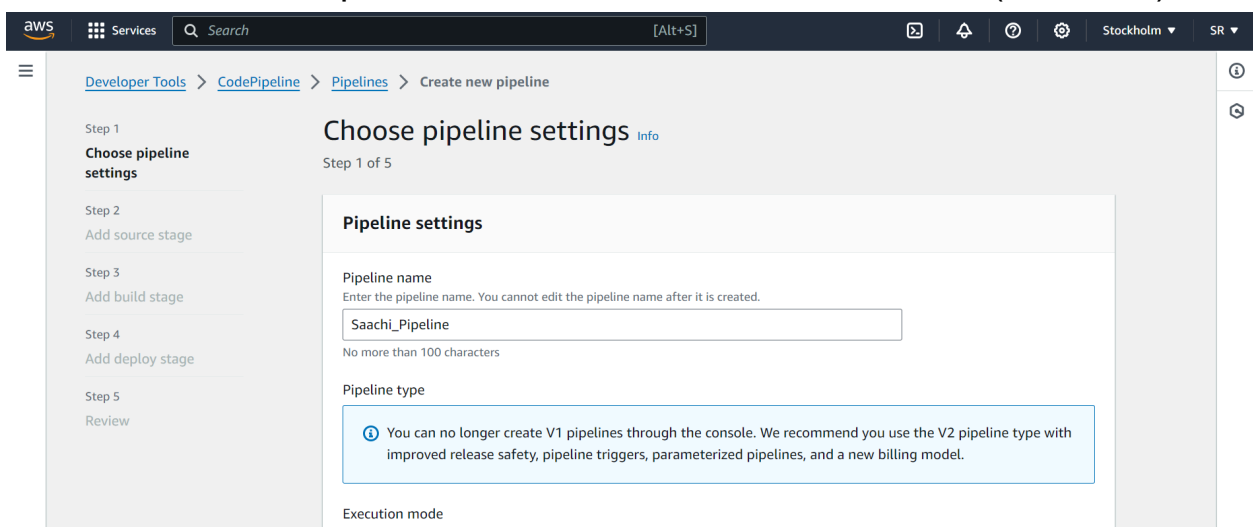
8. In an ec2 instance profile, select the created IAM role



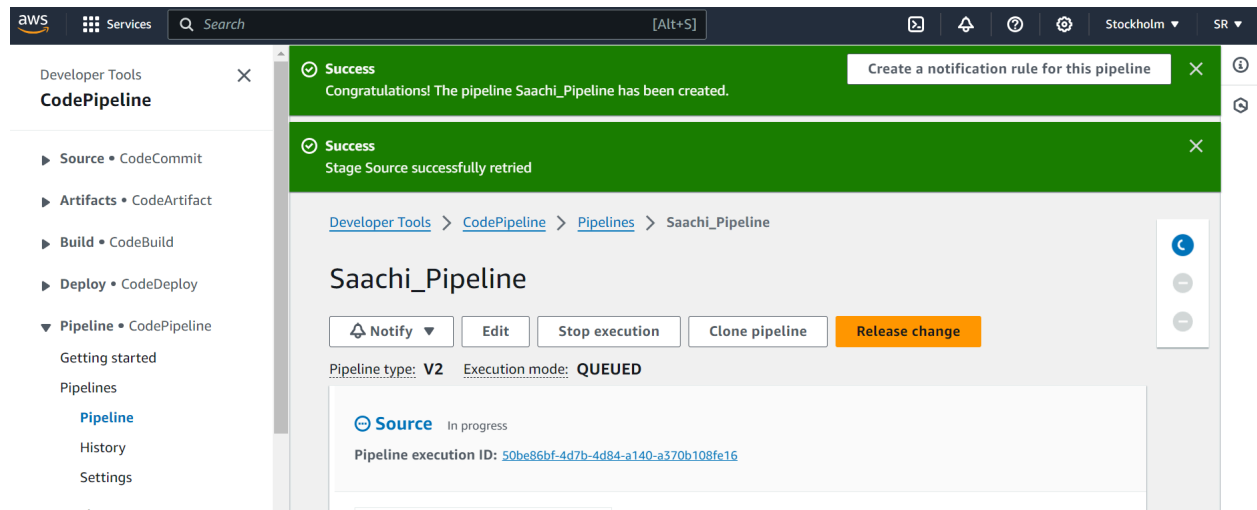
## 9.Environment is launched successfully



## 10.Go to the CodePipeline and select the source as GitHub (version 1).



11. After skipping the build stage, AWS Elastic beanstalk is to be selected in the Deploy Provider. Select your recently created application name and environment name.



12. Go to the elastic beanstalk environment and click on domain.

