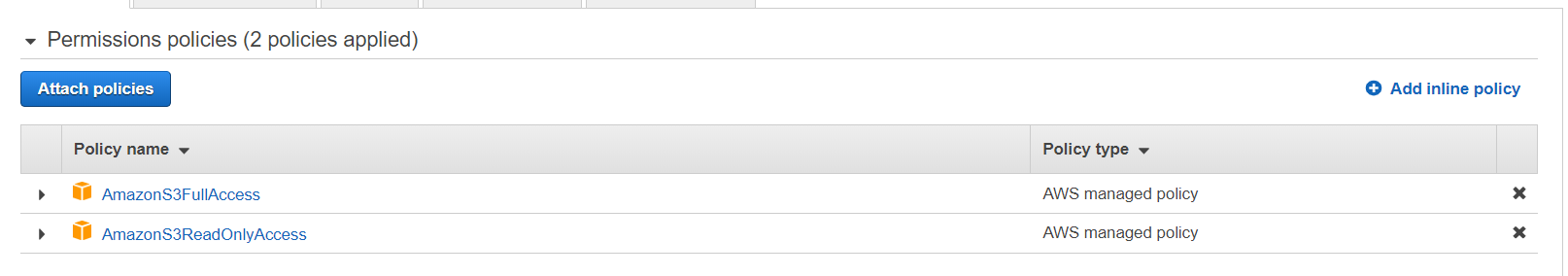
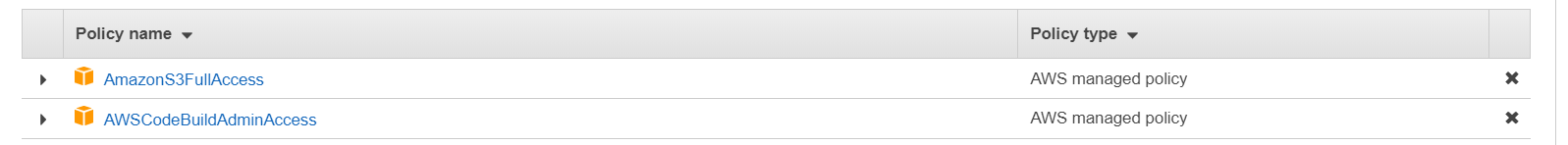
1. Create Roles for Access
2. Creating Repository in Code Commit and Cloning Repository to local folder
3. More with Code Build
4. Launch Instance
5. Code Deploy in Action
6. One click Code Pipe Line
7. **Create Roles for Access:**
8. **Role for Ec2 Instance –** Create a role with “AmazonS3FullAccess” and “AmazonS3ReadOnlyAccess” to attach Ec2 Instance at the time of launch.



1. **Role for Code Build –** Create a role with “AmazonS3FullAccess” and “AWSCodeBuildAdminAccess” to attach to Code Build tool.



1. **Role for Code Deploy –** Create a role with “AWSCodeDeployRole” to attach to Code Deploy tool.



1. **Creating Repository in Code Commit and Cloning Repository to local folder**
2. Login to AWS console and select service Code Commit and create a repository with user defined name.
3. Create an empty folder and clone the repository using SSH or HTTP link.
4. Initialize Git for the folder and copy all the source files to the folder.
5. Using Git commands commit and push the files using git commands to the Code Commit repository.
6. **More with Code Build**
7. In Code Build tool, click on create new Build project.
8. Provide desired Name to the build, Description.
9. In the source section, select Code commit as source and select repository created in code commit.
10. In the Environment Section, select Operating System as Ubuntu, Runtime as Standard, Image as aws/codebuild/standard:2.0, Image version as Latest version.
11. Select Role which we created for code build tool.
12. Select Use a Buildspec file option.
13. Select S3 Object if created as Artifact.
14. Enable Cloudwatch Logs to check the log files.
15. Click on create Build project.
16. **Launch Instance:**
17. Select EC2 Service and Launch the Linux Instance.
18. Select Amazon Linux AMI.
19. Select Role which we created for EC2 Instance.
20. In Advanced Details, type the below commands to run during Instance launch

#!/bin/bash

yum -y update

yum install -y ruby

yum install -y aws-cli

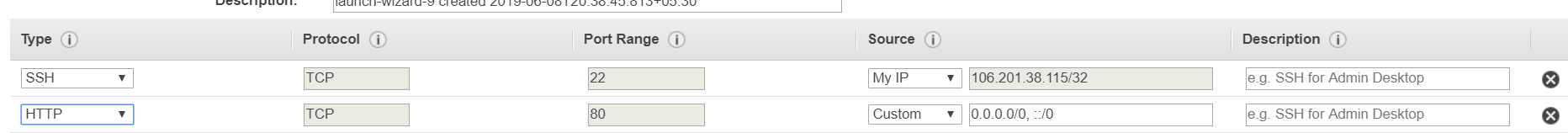
cd /home/ec2-user

aws s3 cp s3://aws-codedeploy-us-east-2/latest/install . --region us-east-2

chmod +x ./install

./install auto

1. Add tags to notice Instance during Code Deploy stage.
2. In Configuring Security Group, Select SSH and select Source as My IP and add HTTP rule to allow all the traffic from outside also.



1. Select EC2 Key pair or create new one and launch the instance.
2. **Code Deploy in Action**
3. In Code Deploy tool, click on Create Application.
4. Provide Application Name and select compute platform as EC2/Onpremises.
5. Create Deployment Group and provide Deployment Group name.
6. Select Service Role created for Code Deploy.
7. Select Deployment type as In-Place Deployment.
8. In Environment Configuration, select Amazon EC2 Instance and enter Key value Tag name of instance launched.
9. Deploy Settings select Deploy All at Once.
10. Finally click on Create Deployment Group.
11. **One click Code Pipe Line**
12. In Code Pipe line, select create new pipeline
13. Provide name for the pipeline and select new service role so that it will create the service role with desired permissions.
14. In Advanced Settings, select Artifact as Default location hence the files will store in default S3 bucket created by the pipeline.
15. In Source Stage, select source as AWS Code Commit and select the repository created and select branch as Master.
16. Leave Change detector option with Amazon cloud watch events.
17. In Build Stage option, select AWS CodeBuild for build provider, select the region where the build is located and select project name.
18. In Add Deploy stage, select DeployProvider as AWS CodeDeploy, select the region where the code deploy project is located and provide Application name and Deployment Group.
19. Then click on create pipeline.