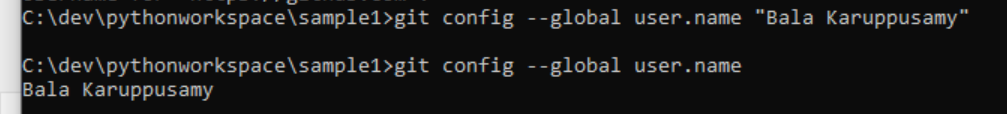
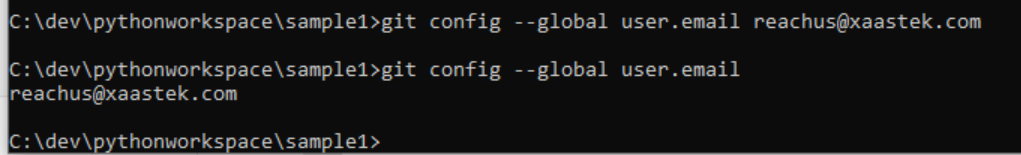
Refer to the pythonlab1 code in the github: <https://github.com/bkaruppu/pythonlab1>

Set up deployment environment:

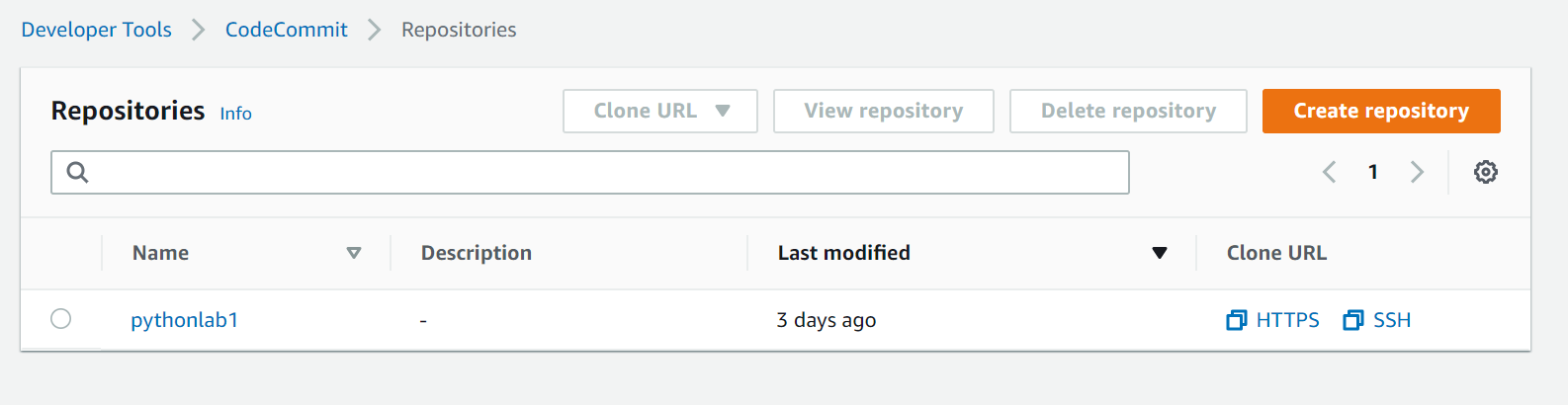
1. Install Anaconda and python(3.7x)
   1. <https://www.anaconda.com/distribution/#download-section>
2. Setup anaconda navigator: Launch anaconda prompt and run:
   1. Update to the latest conda version: *conda update anaconda*
   2. Setup navigator: *conda install -c anaconda anaconda-navigator*
3. Install Git :
   1. <https://git-scm.com/download/win>
   2. Set your username and email address in Git.
      1. Open Git Bash
      2. Set a Git username and confirm:
         1. $ git config --global user.name “Bala Karuppusamy“
         2. $ git config --global user.name
      3. Set a Git email address and confirm:
         1. $ git config --global user.email reachus@xaastek.com
         2. $ git config --global user.email

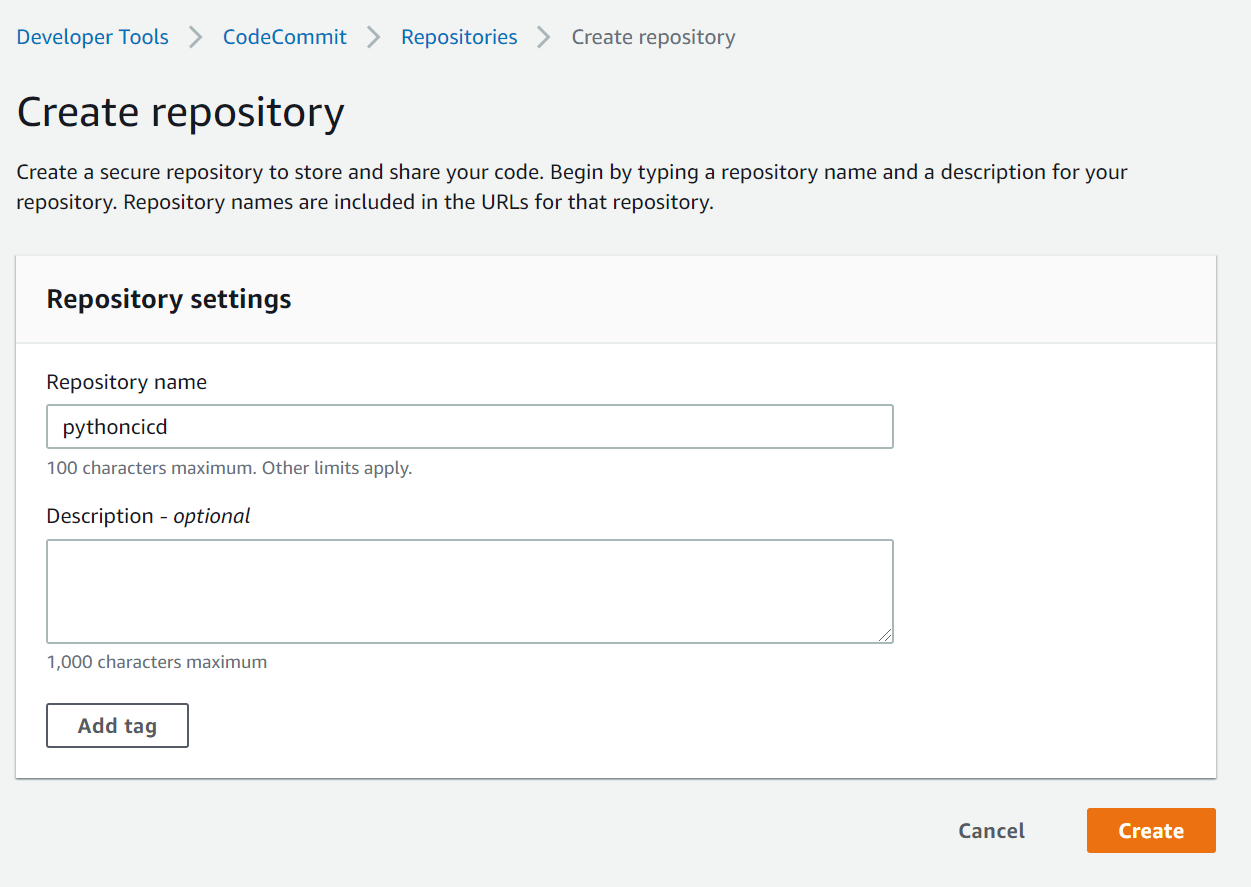


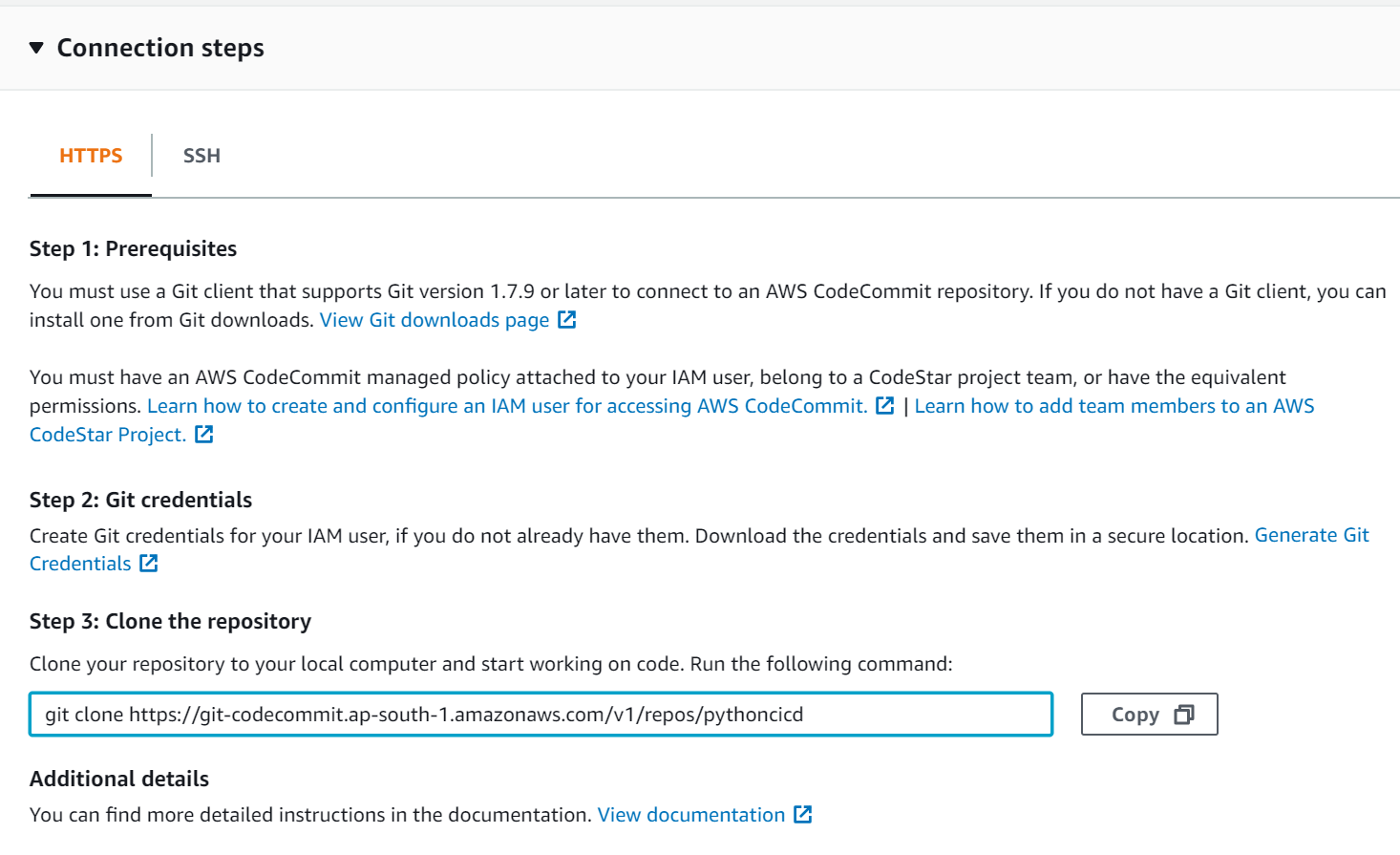


* 1. Learning git command:
     1. git clone
     2. git add
     3. git commit
     4. git push

1. Setup IDE in Anaconda Navigator:
   1. Launch Anaconda Navigator app and install spyder
2. Setup AWS account
3. Setup Putty and PuttyGen to validate AWS Environment:
   1. <https://www.putty.org/>
4. Setup S3 bucket for build process ( example <https://s3.console.aws.amazon.com/s3/buckets/btechlab4/?region=ap-south-1&tab=overview>)
5. Setup Source code management:
   1. Create a Repository in CodeCommit
   2. Copy URL ( example : https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythonlab1)







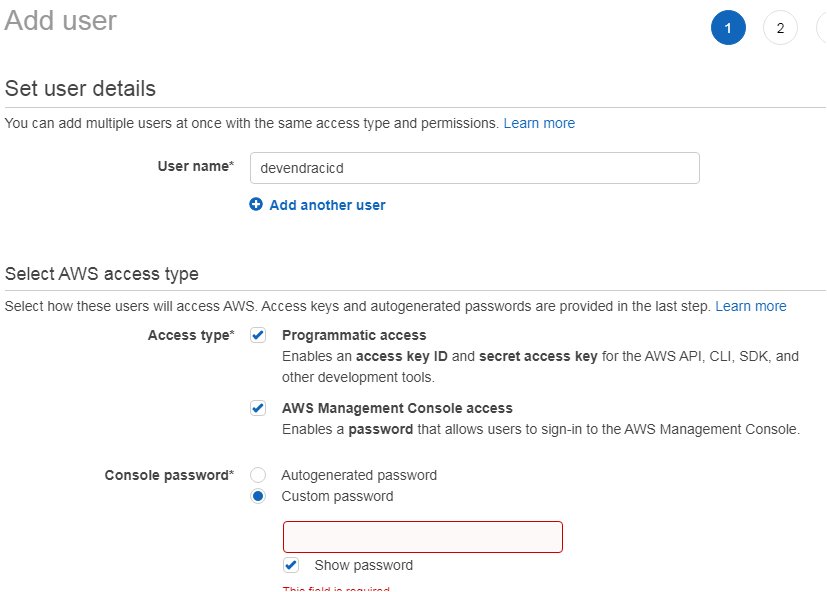
Repository URL :

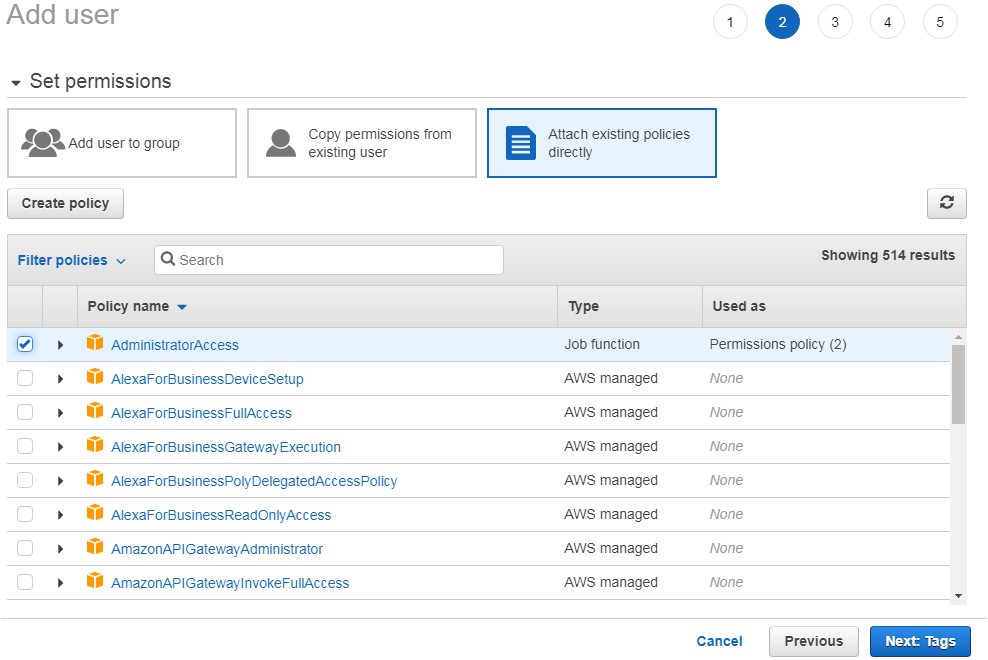
[**https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythoncicd**](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythoncicd)

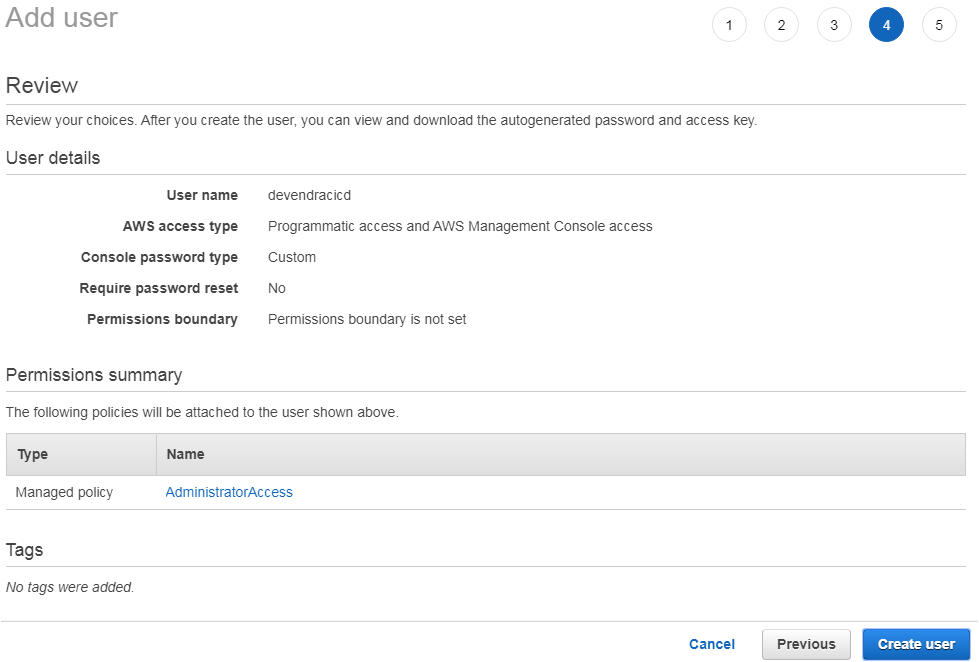
**Commad to clone:**

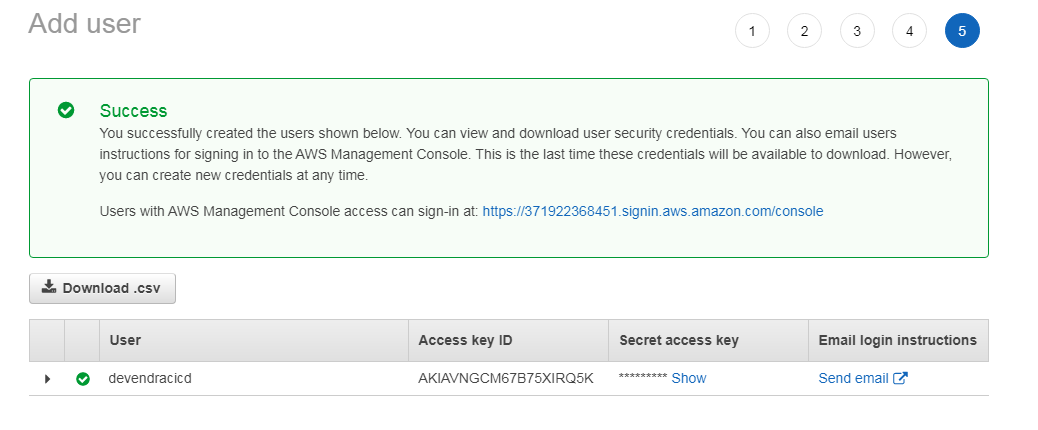
**git clone** [**https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythoncicd**](https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythoncicd)

* 1. Create GIT credentials for accessing the repository: (<https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-gc.html>)
     1. Add user:

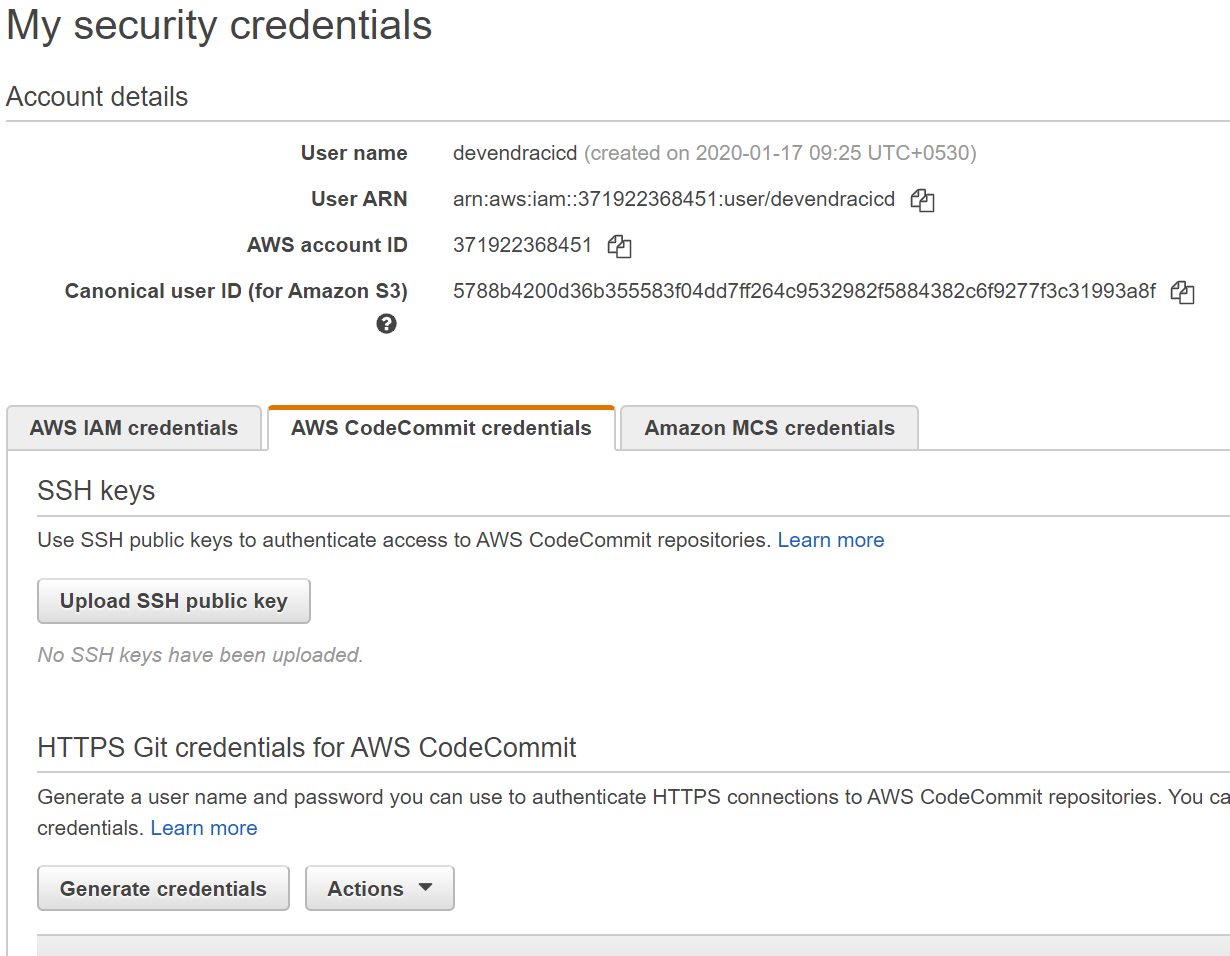


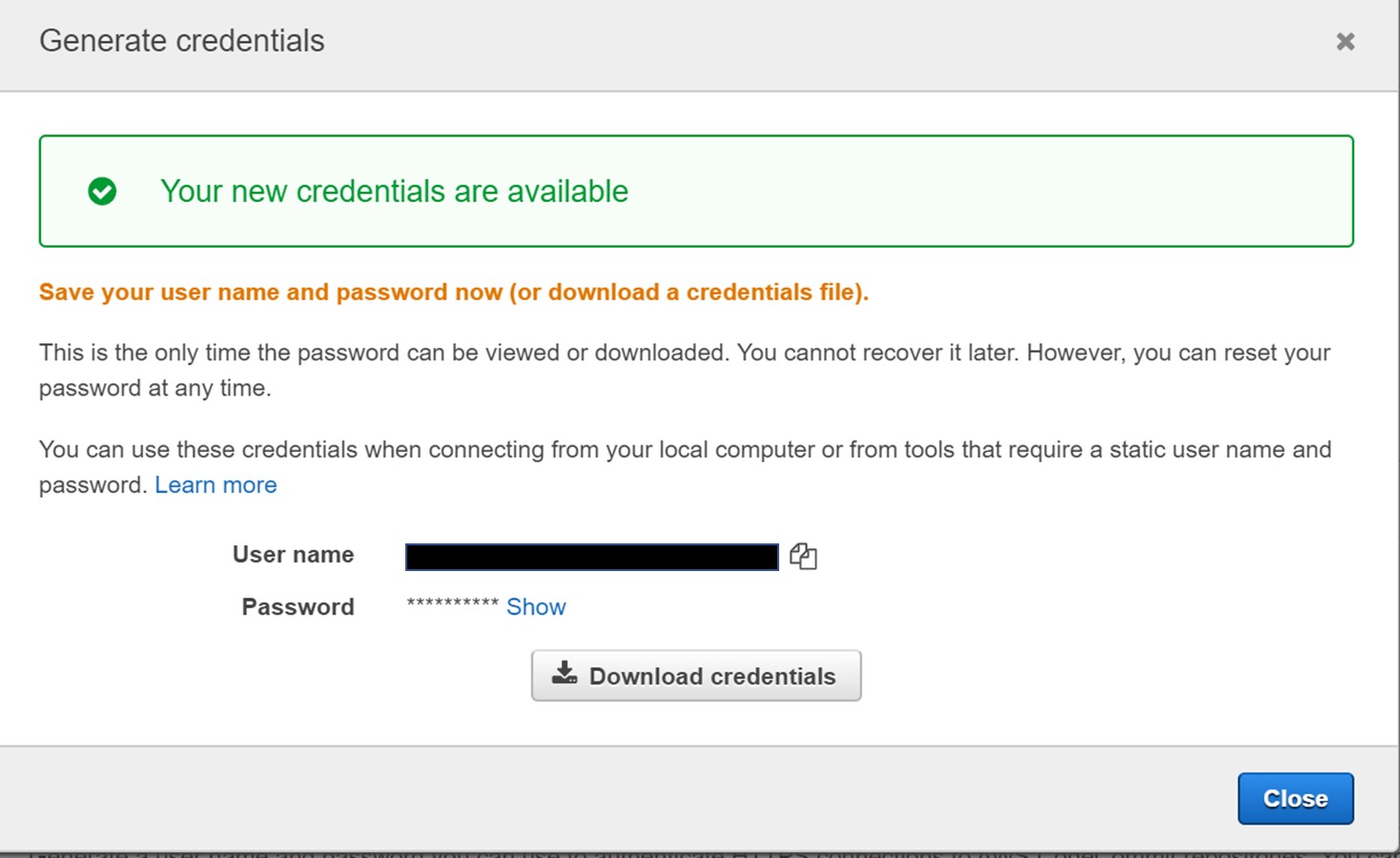




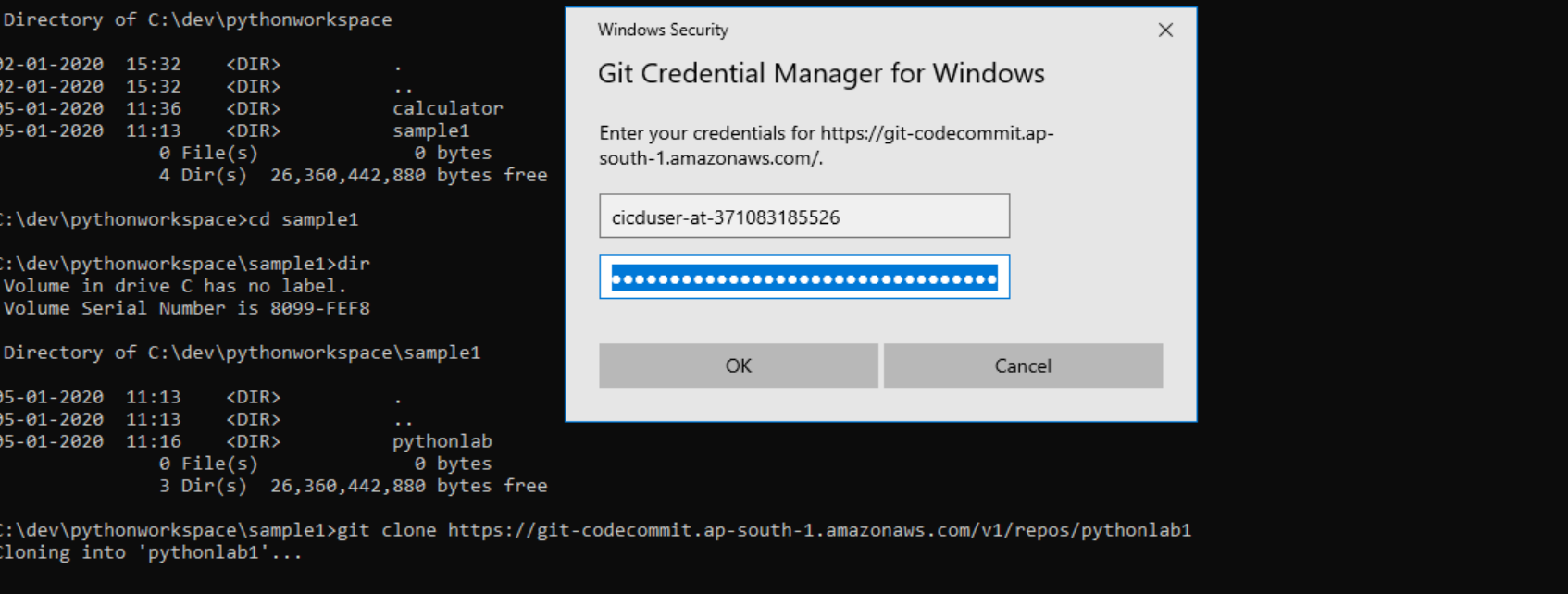


* + 1. Generate and download Git credentials:

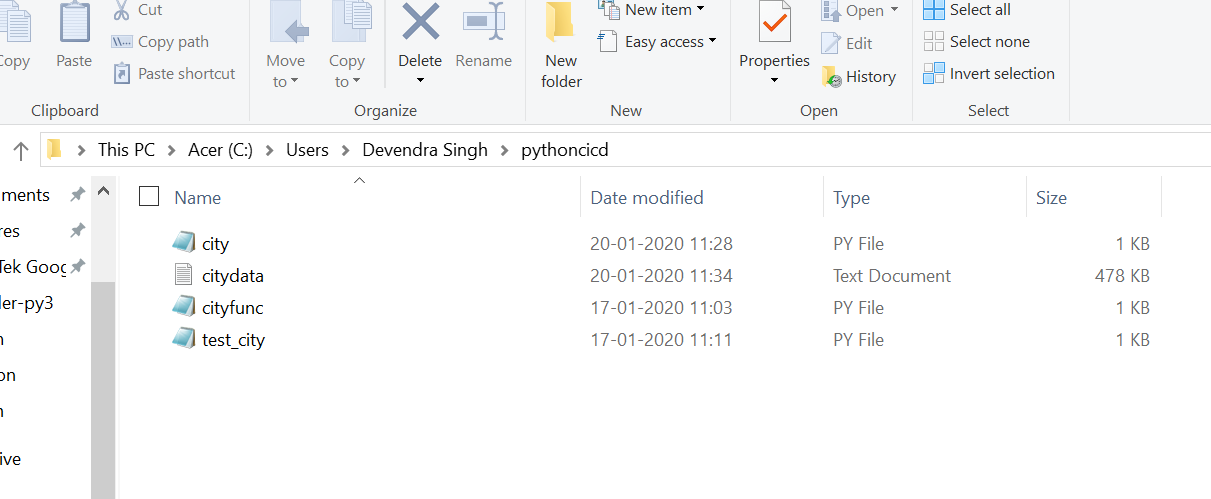




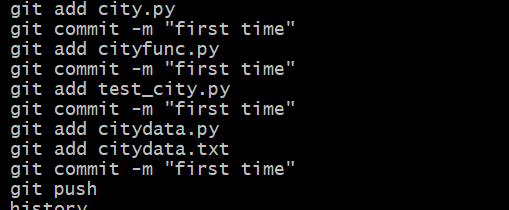
1. Create clone of the repository in the local desktop:
   1. git clone <https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/pythonlab1>



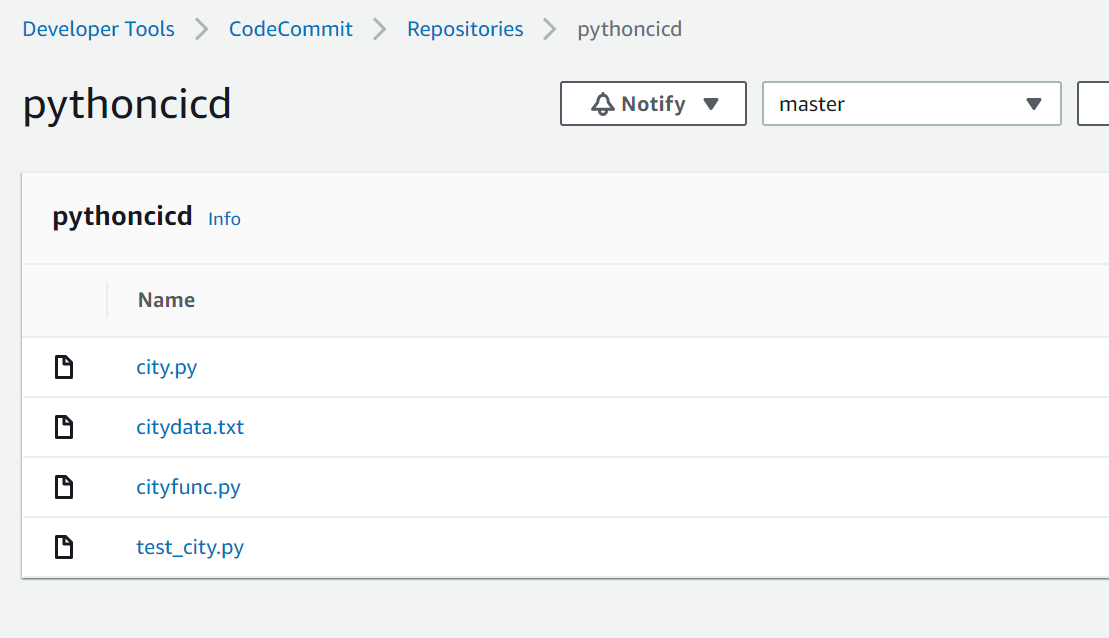
1. Copy the following files under the new repository and check-in those files to CodeCommit:
   1. Copy the files:



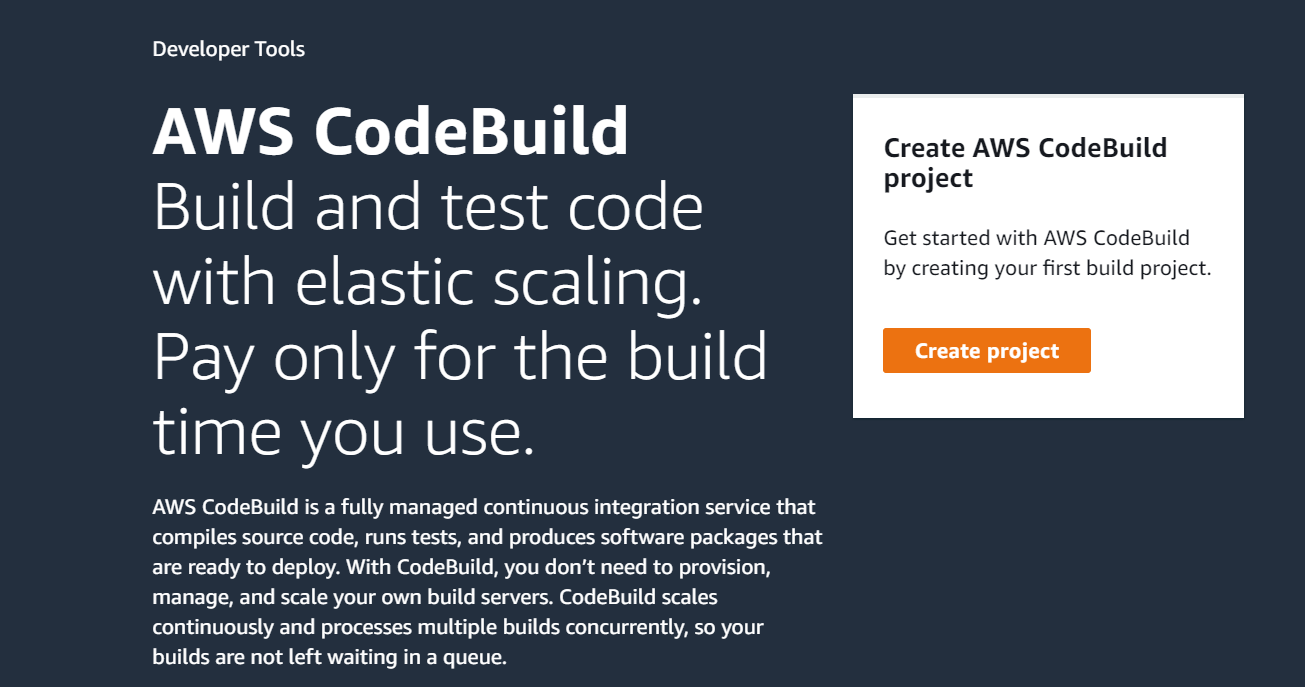
* 1. Add the files to the repository using git commands add, commit and push;



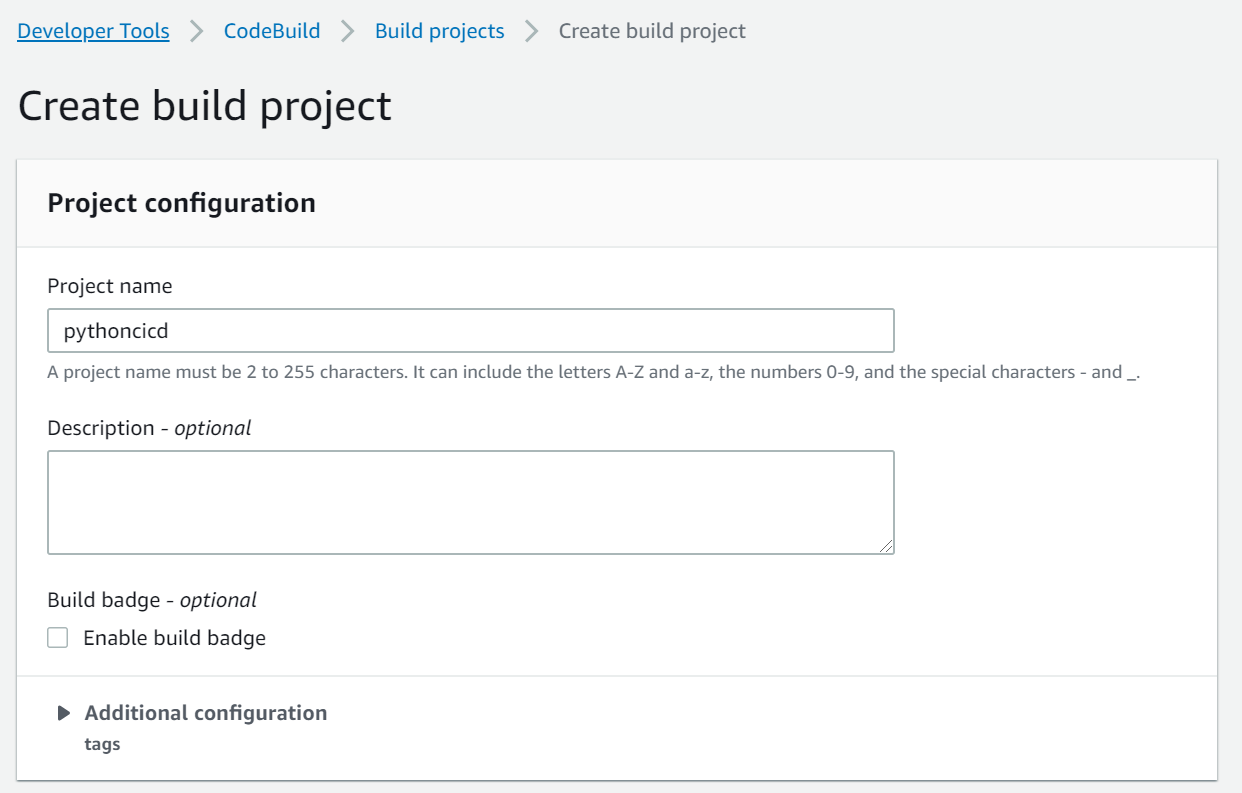
* 1. Verify AWS CodeCommit repository



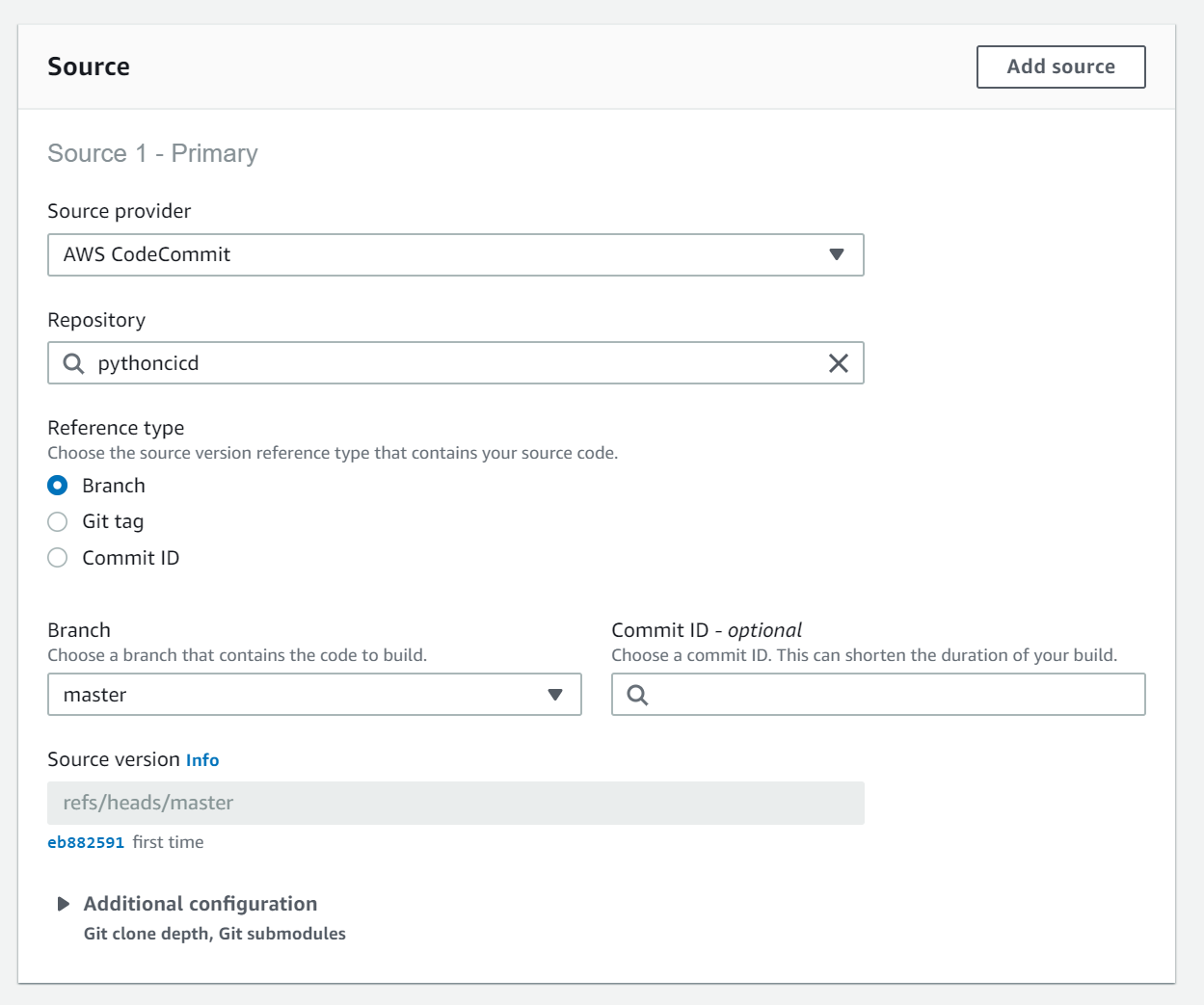
1. Setup CodeBuild for the pythonlab1 repository:
   1. Create a CodeBuild Project



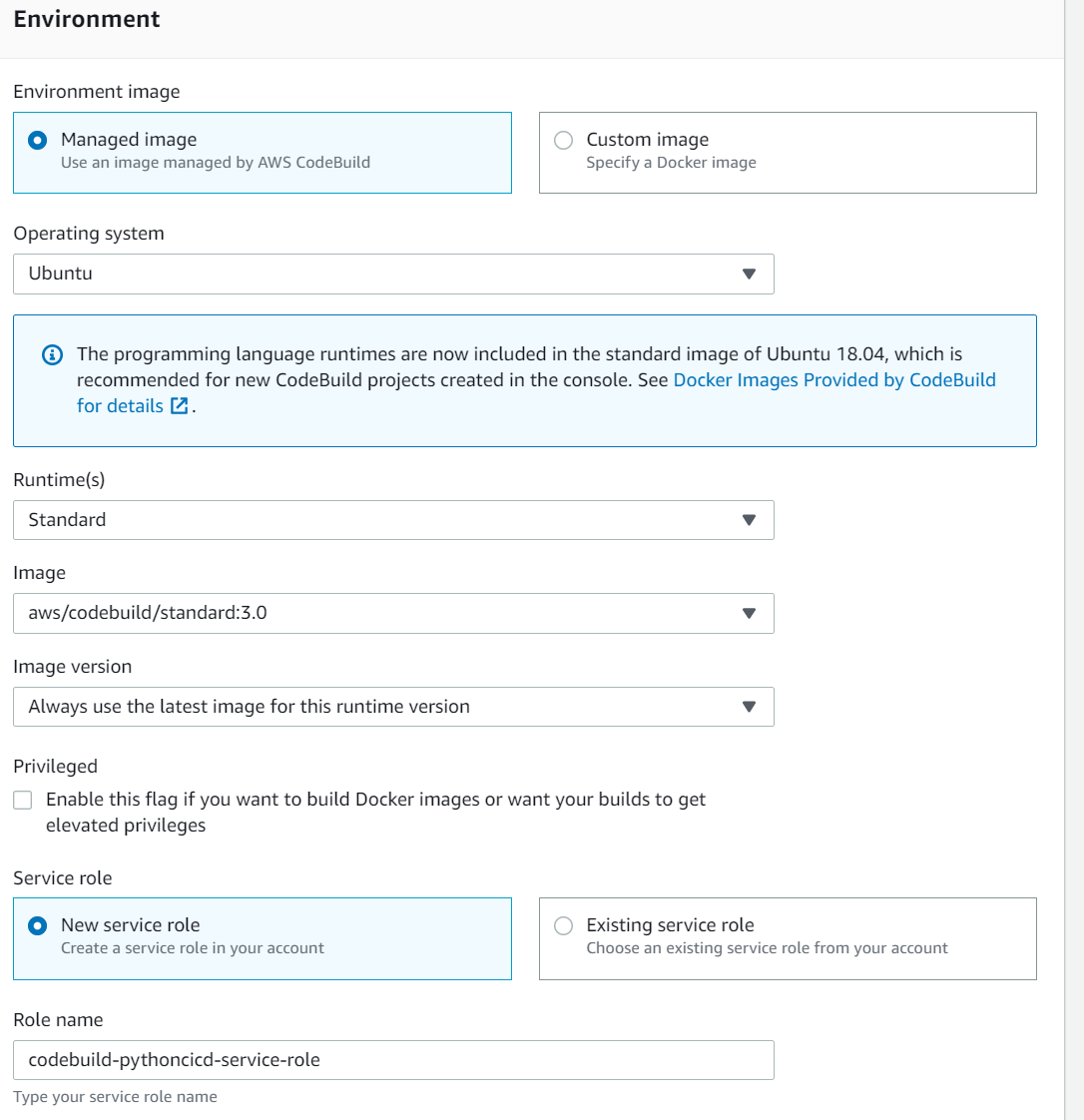
* 1. Define the project configuration



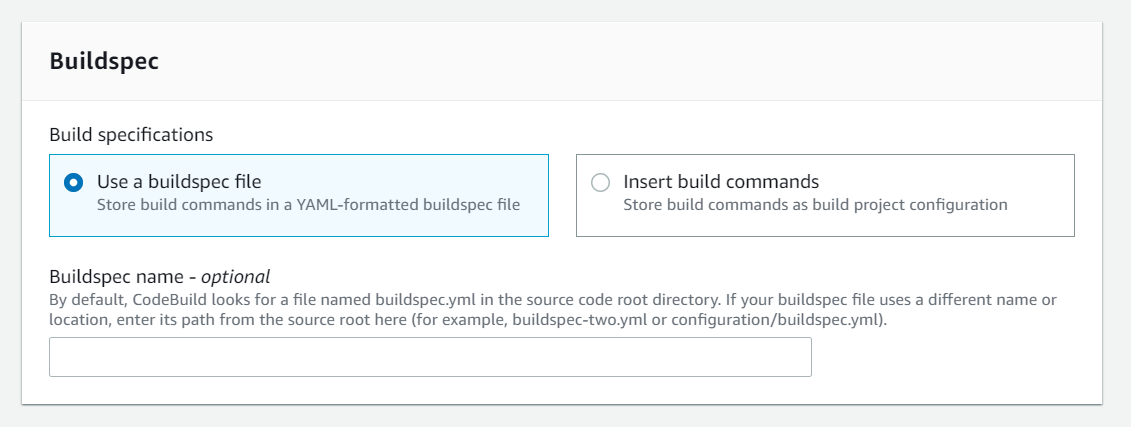
* 1. Define the source ( here use the repository created in CodeCommit)



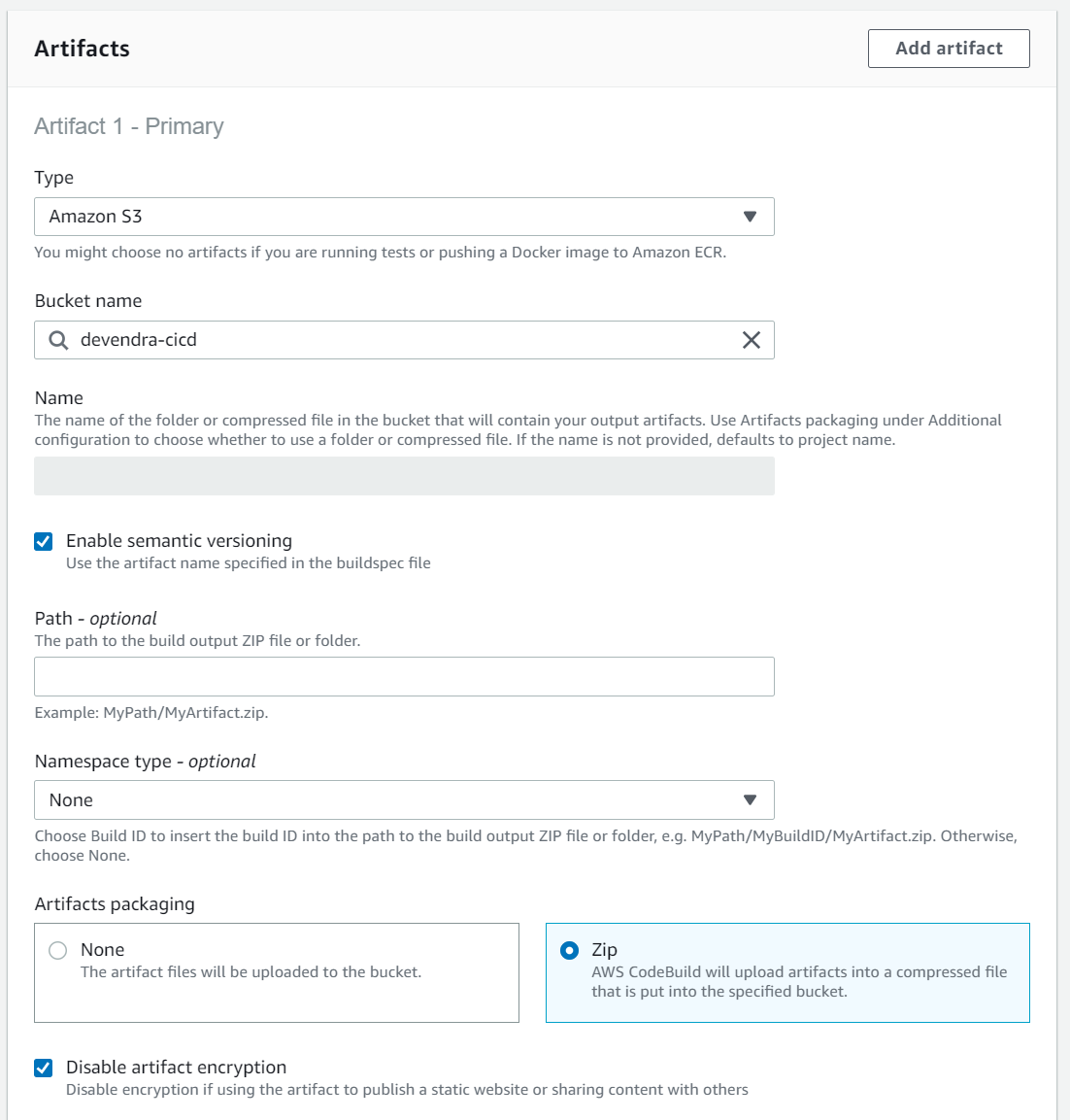
* 1. Environment



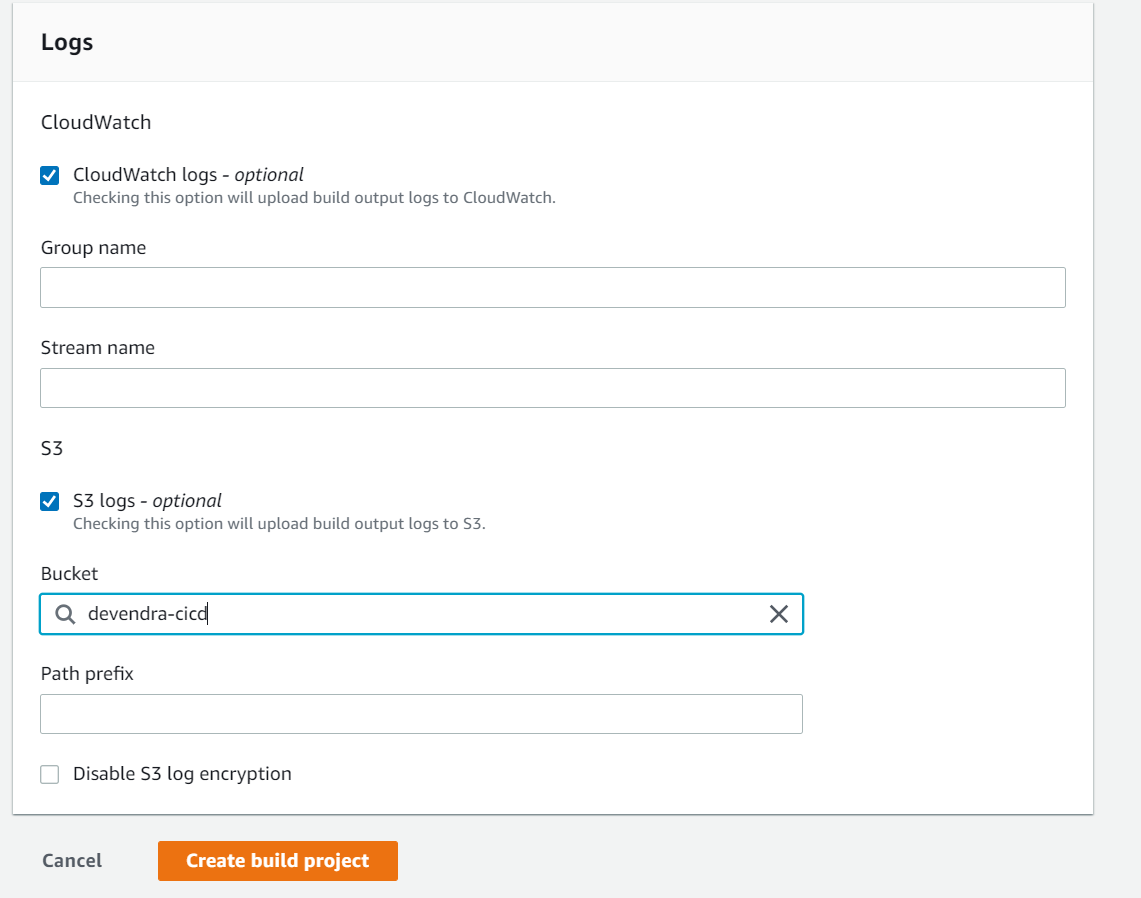
* 1. Build spec:

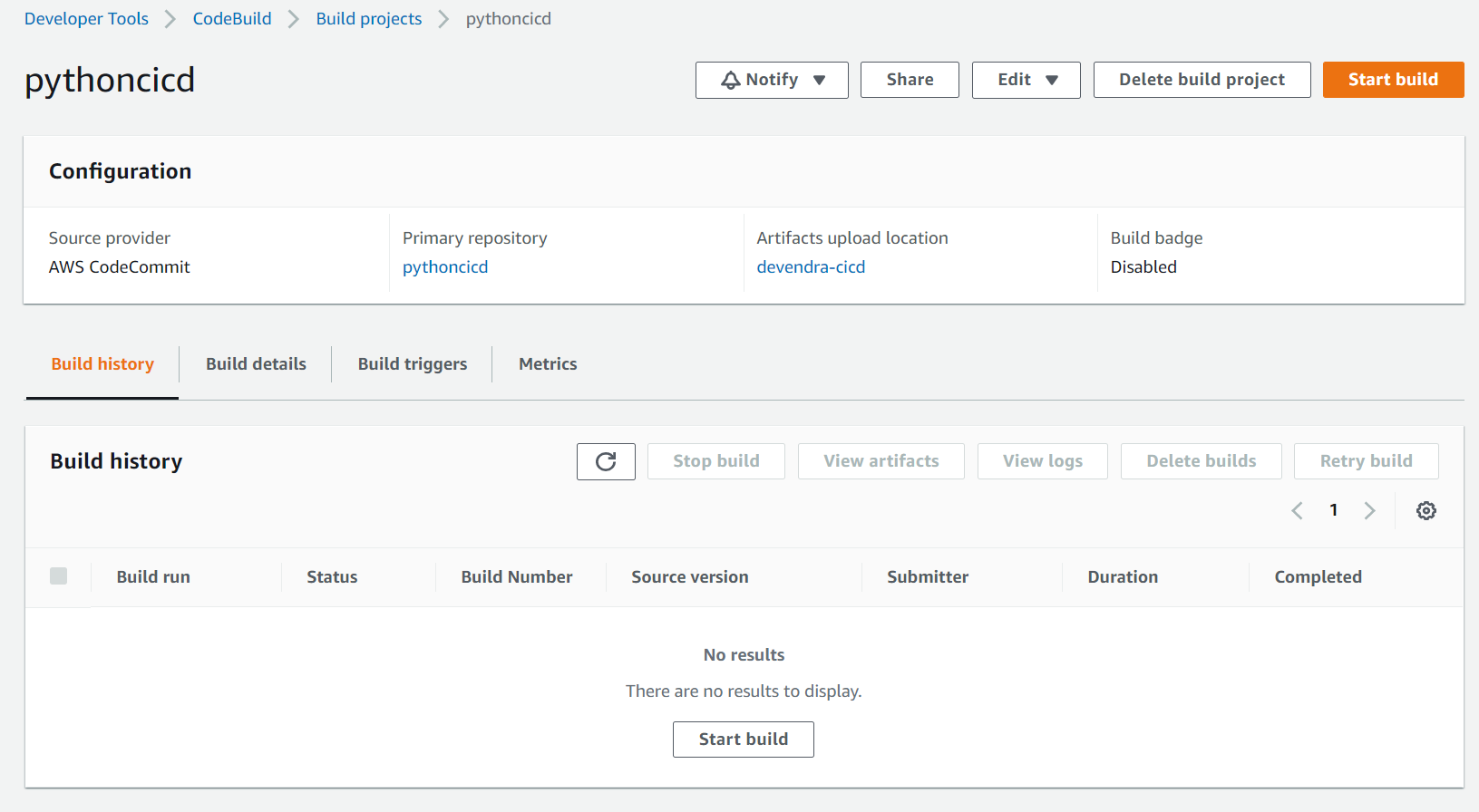


* 1. Artifacts



* 1. Logs





* 1. Buildspec for python app:

version: 0.2

phases:

install:

runtime-versions:

python: 3.7

pre\_build:

commands:

- echo "Before running tests"

- python -m pip install --upgrade pip

- pip install flake8

- pip install pytest

build:

commands:

- flake8 . --count --select=E9,F63,F7,F82 --show-source --statistics

- flake8 . --count --exit-zero --max-complexity=10 --max-line-length=127 --statistics

- echo "Running all unit tests"

- pytest

- tar -cf pythonlab1-1\_0.tar \*

post\_build:

commands:

- echo Build completed on `date`

artifacts:

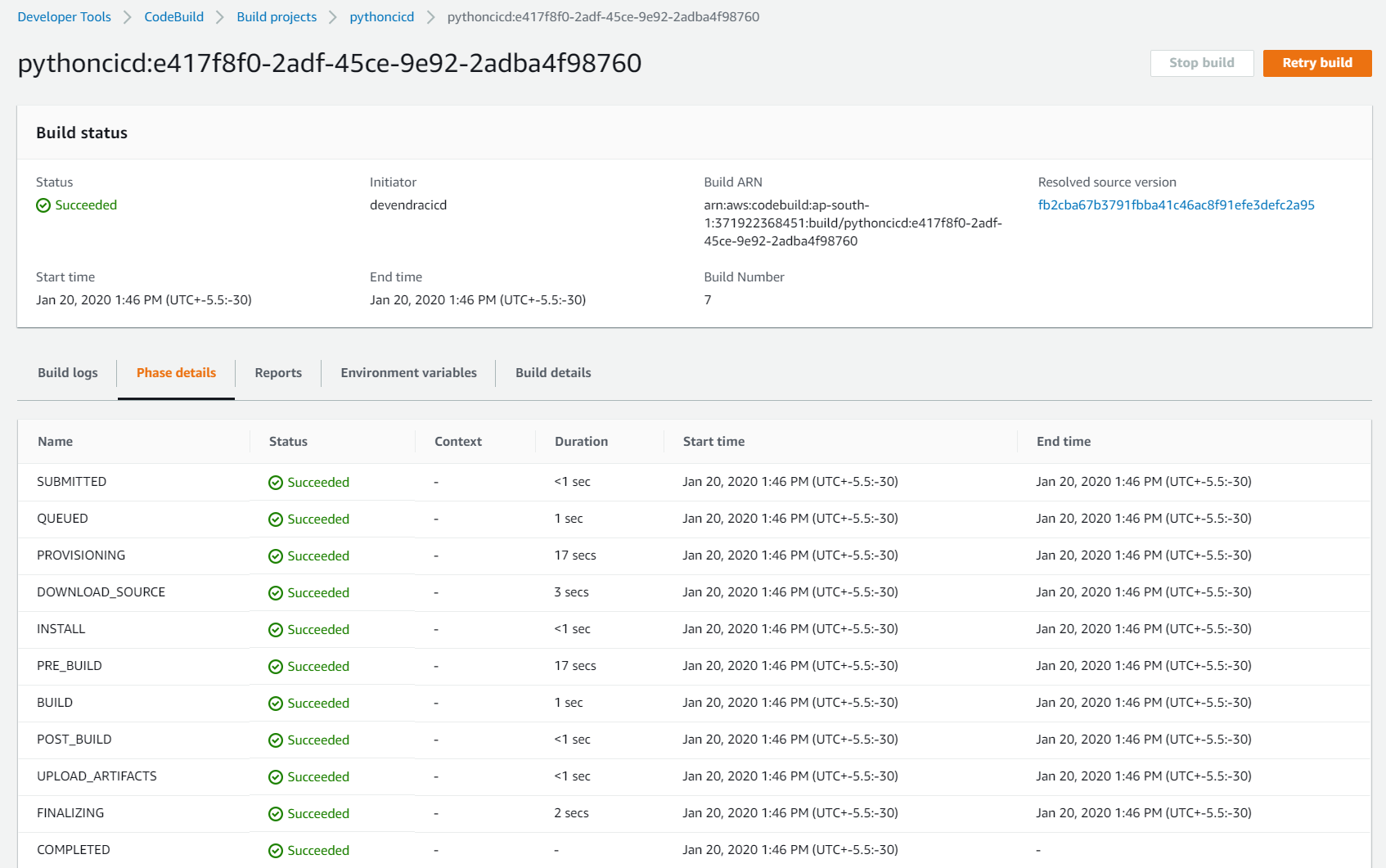
files:

- pythonlab1-1\_0.tar

- appspec.yml

discard-paths: yes

* 1. Run codebuild



1. Set Up Putty , PuttyGen and other dependencies to validate AWS EC2 Environment for production
   1. Install Putty and PuttyGen ( from <https://www.putty.org/>)
   2. Create a EC2 instance with Ubuntu image deployment tag (Name = pythonlab)
   3. Install Code deploy agent( run as sudo -s)
      1. apt-get update
      2. apt-get install ruby
      3. apt-get install wget
      4. cd /home/ubuntu
      5. wget https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
      6. chmod +x ./install
      7. ./install auto
      8. : sudo /opt/codedeploy-agent/bin/install auto
      9. service codedeploy-agent status
      10. service codedeploy-agent start
   4. Install Python and Pytest
      1. $ apt install python3.8
      2. $ apt install python-pip
      3. $ pip install flake8
      4. $ pip install pytest
      5. $ pip install pandas
      6. sudo apt update
      7. sudo apt install python3-pandas
2. appspec.yml for the deploy

version: 0.0

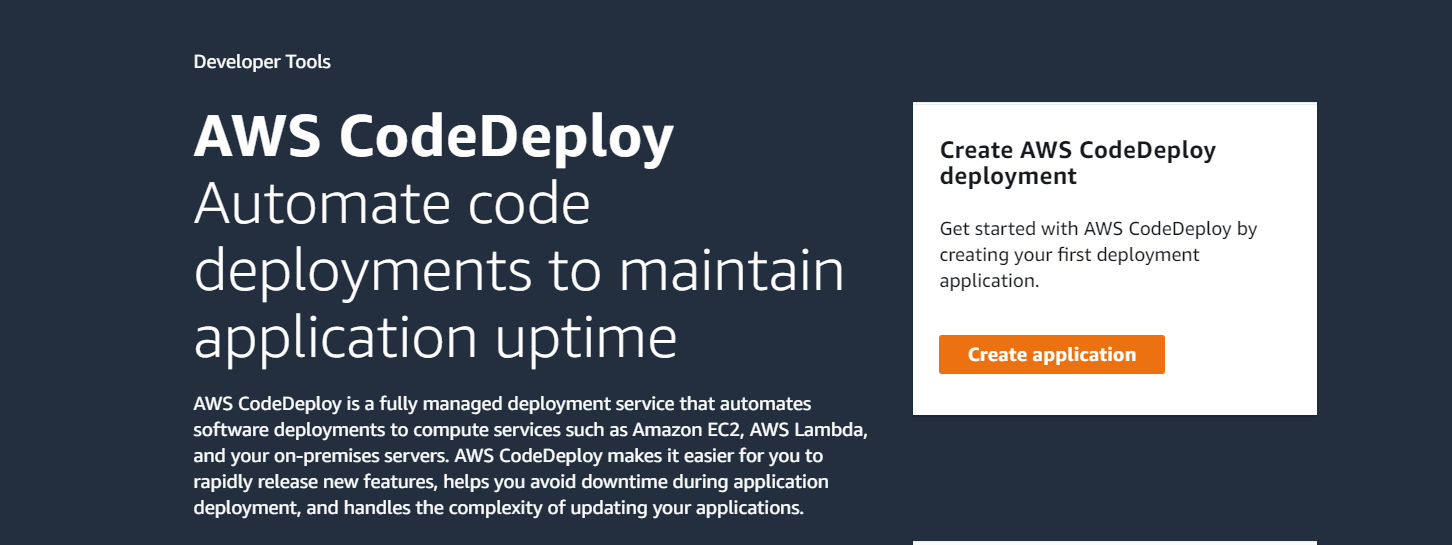
os: linux

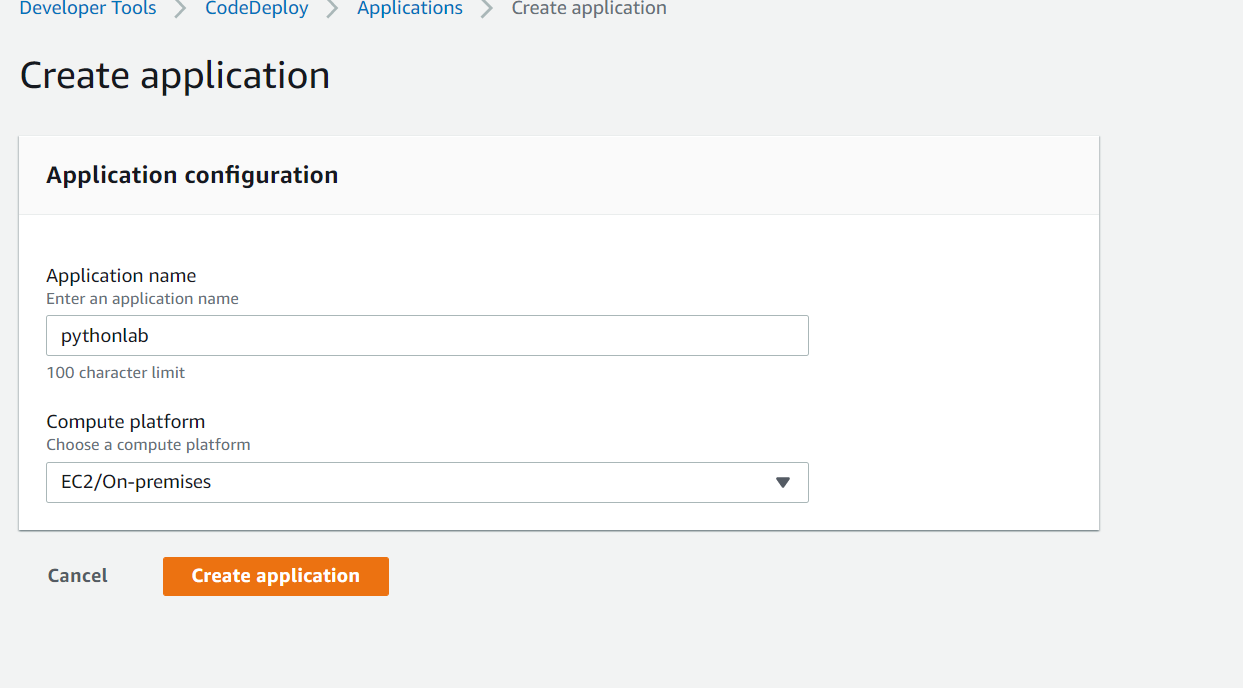
files:

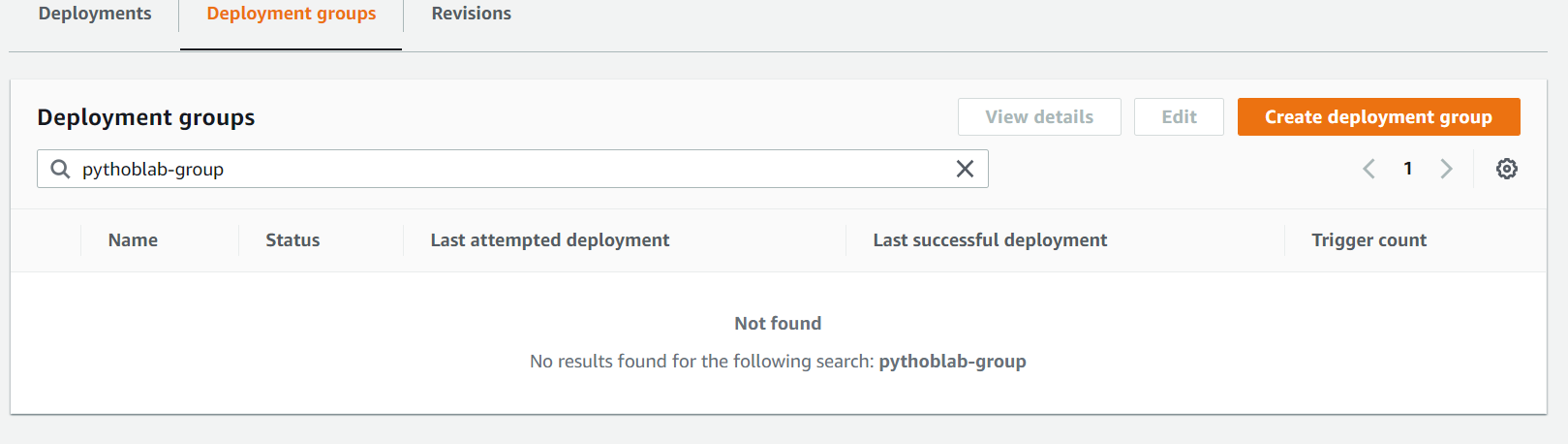
- source: pythonlab1-1\_0.tar

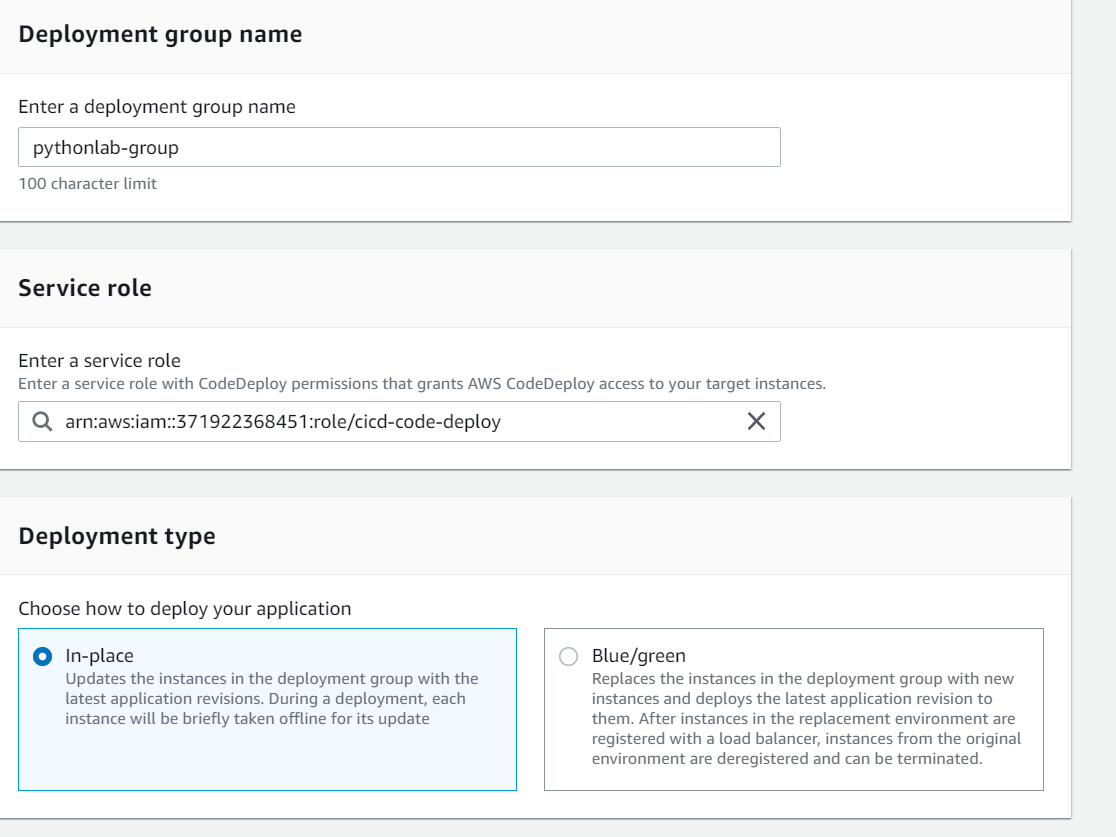
destination: /tmp

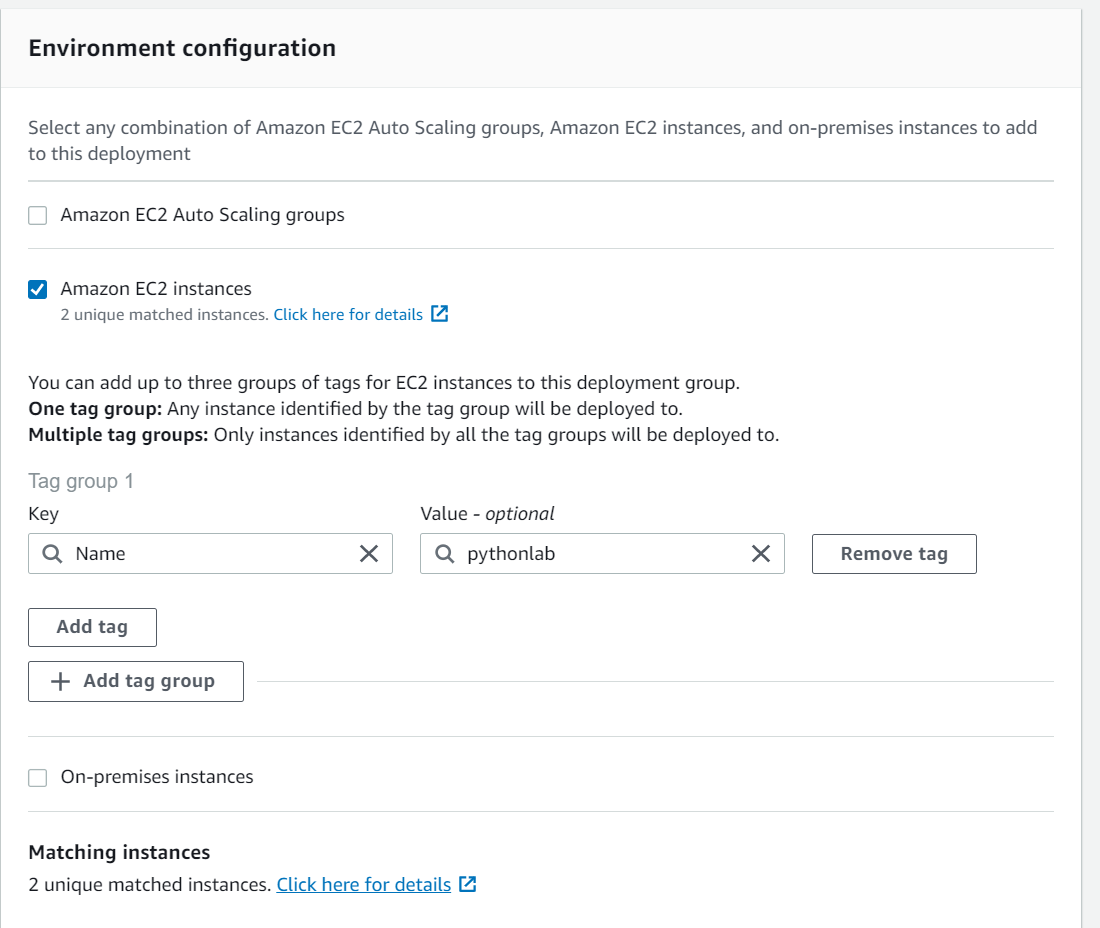
1. Setup Code Deploy
   1. Create a Service role cicds3access ( attach the policy AmazonS3FullAccess).
   2. Attach the cicds3access to the EC2 instance
   3. Create a service role cicdsdeploy( attach the policy AWSCodeDeployRole)
   4. Create Application
   5. Create Deployment group (attach the service role pythonlab1codedeploy)

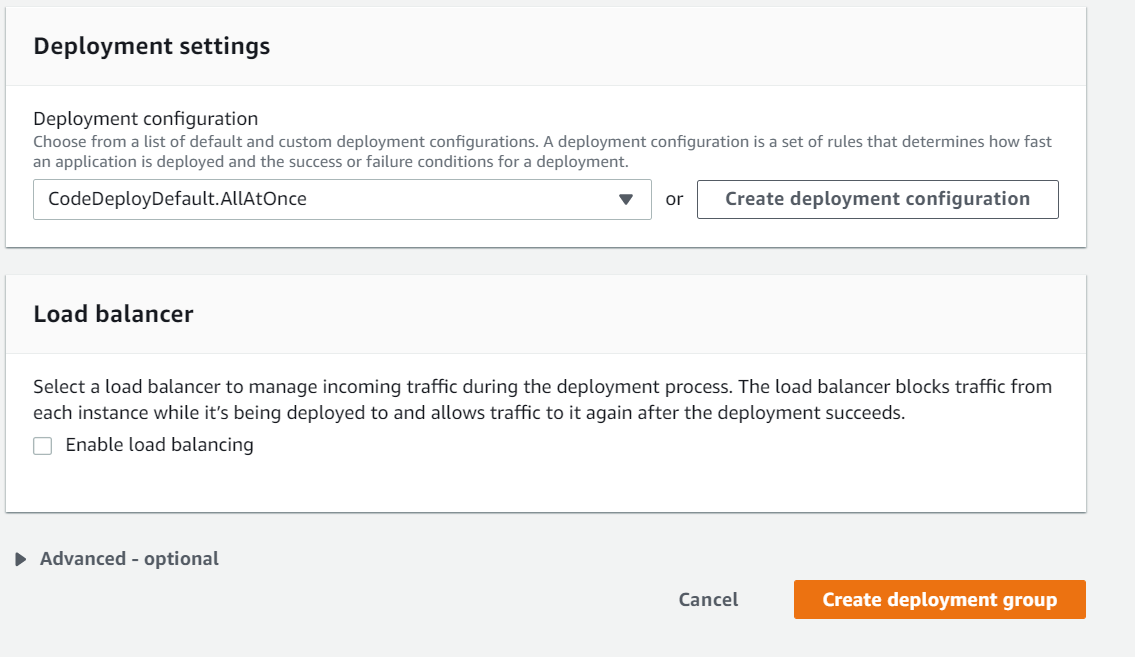




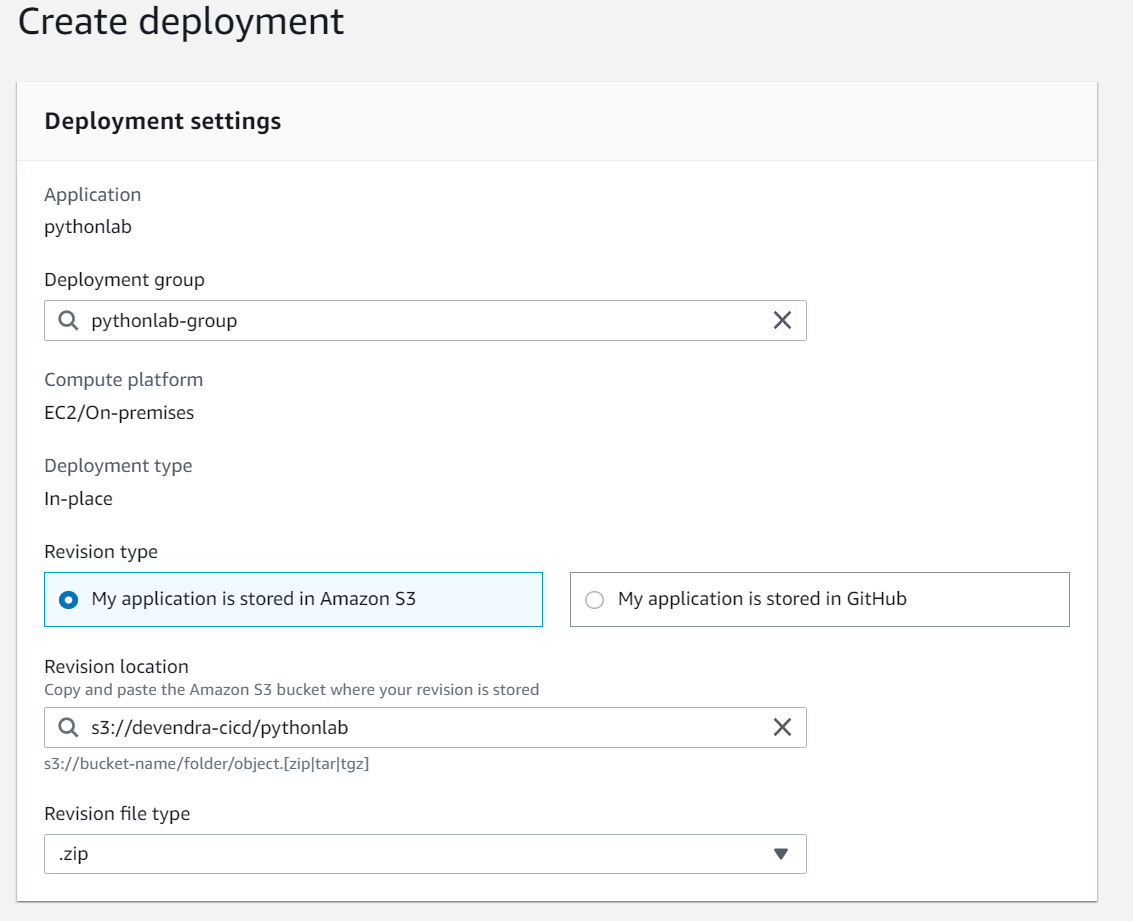


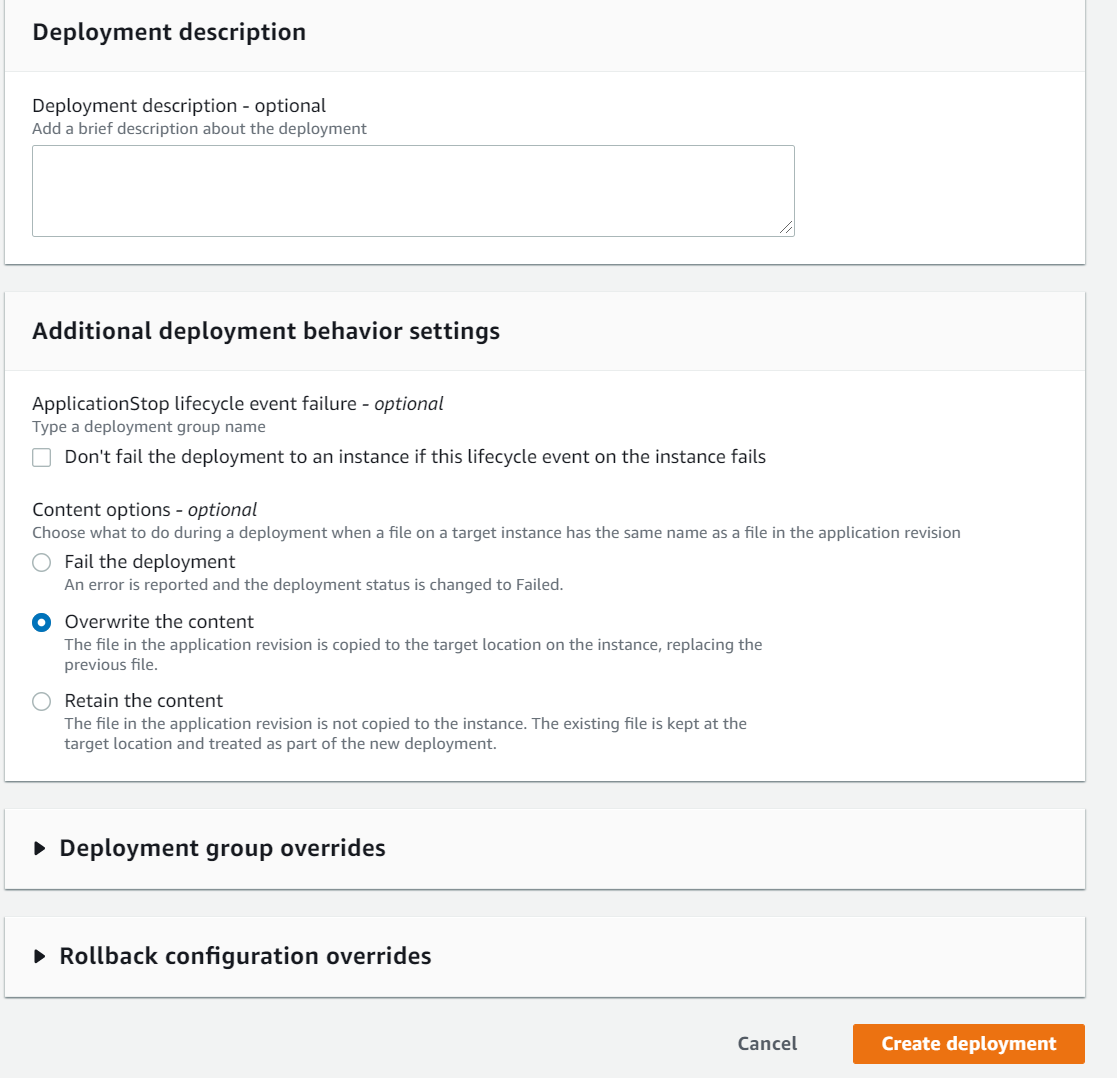






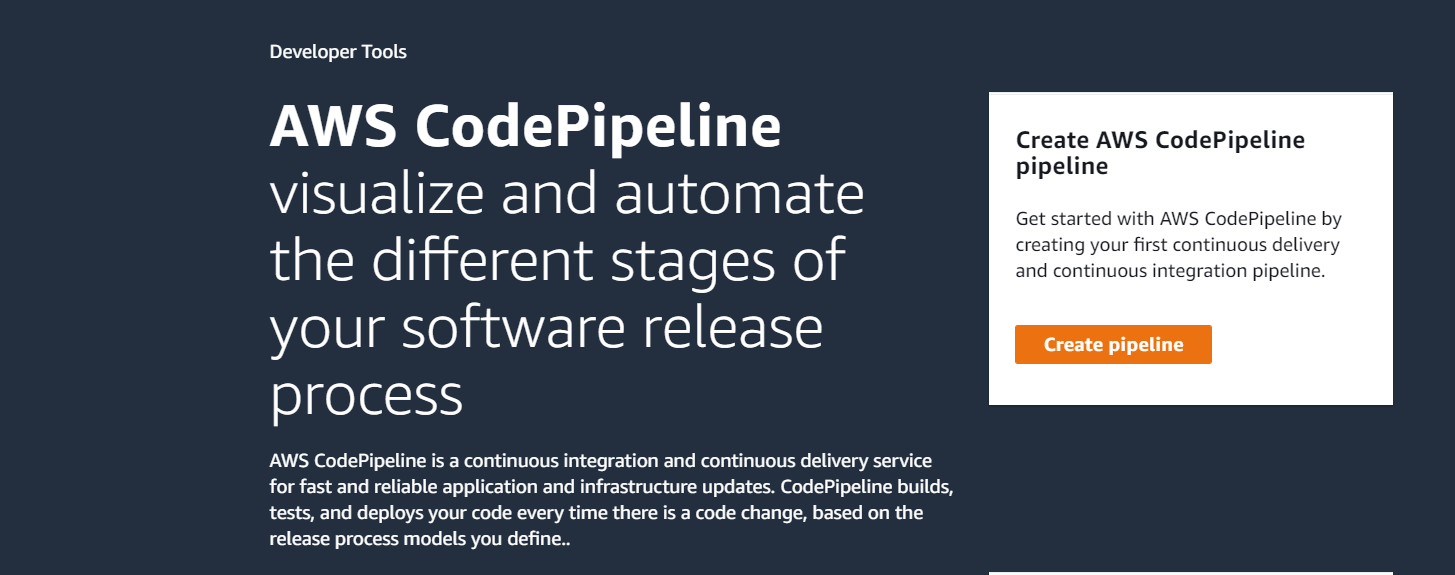
* 1. Create Deployment

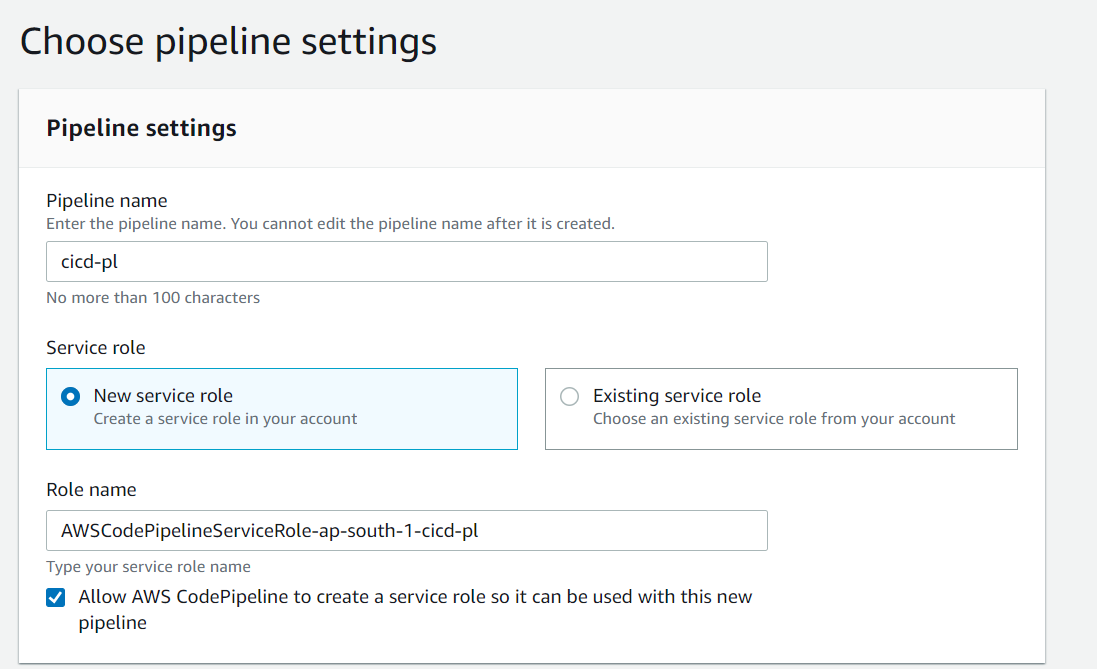


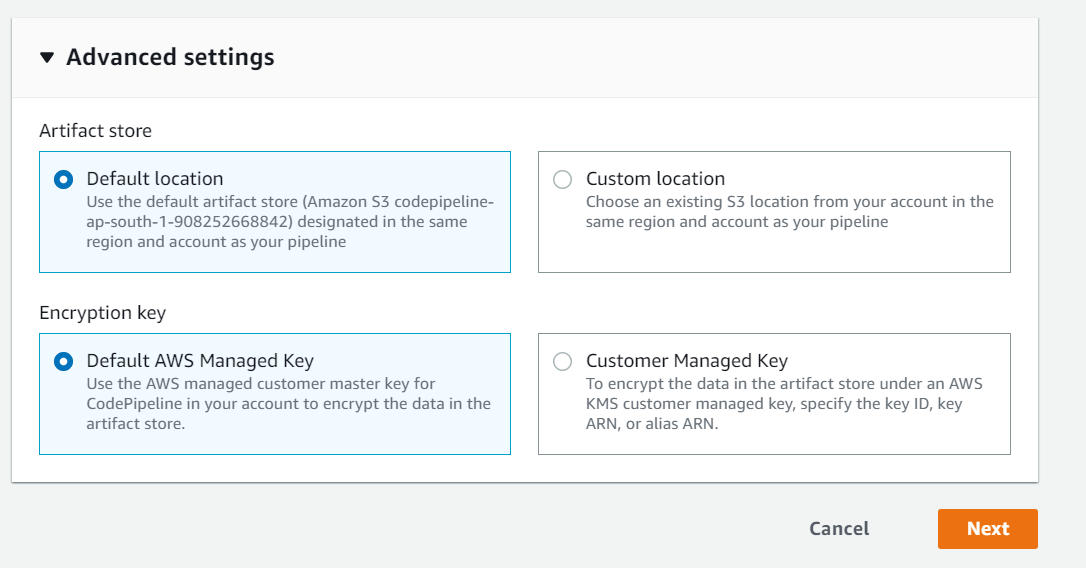


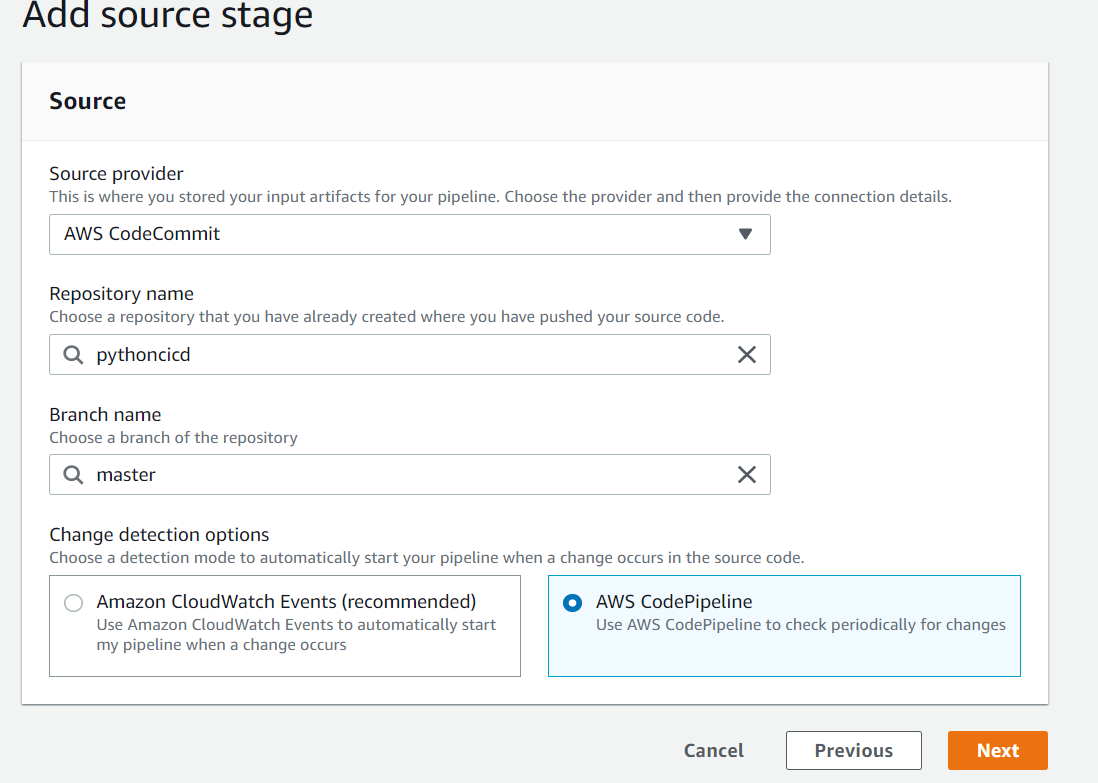
* 1. Debugging:
     1. If any code deploy agent issue run the command: sudo /opt/codedeploy-agent/bin/install auto
        1. Refer to <https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-update.html>
     2. Delete the app and recreate the deployment, since it carries forwards some of the old artifacts even though it doesn’t exist in the build
     3. Also check the archive folder /opt/codedeploy-agent/deployment-root/ for any issue
     4. Also use the file name without zip or other archive extension
     5. Code Build enable zip archive for the artifacts

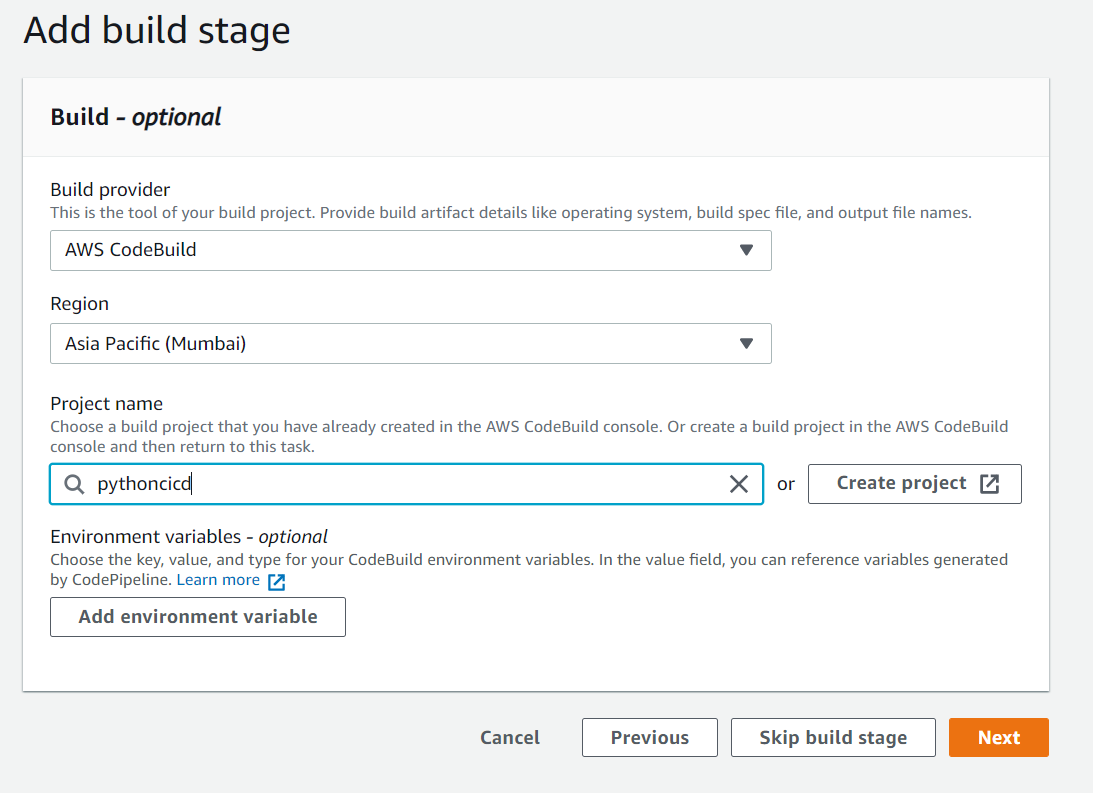
1. Setup Code Pipeline

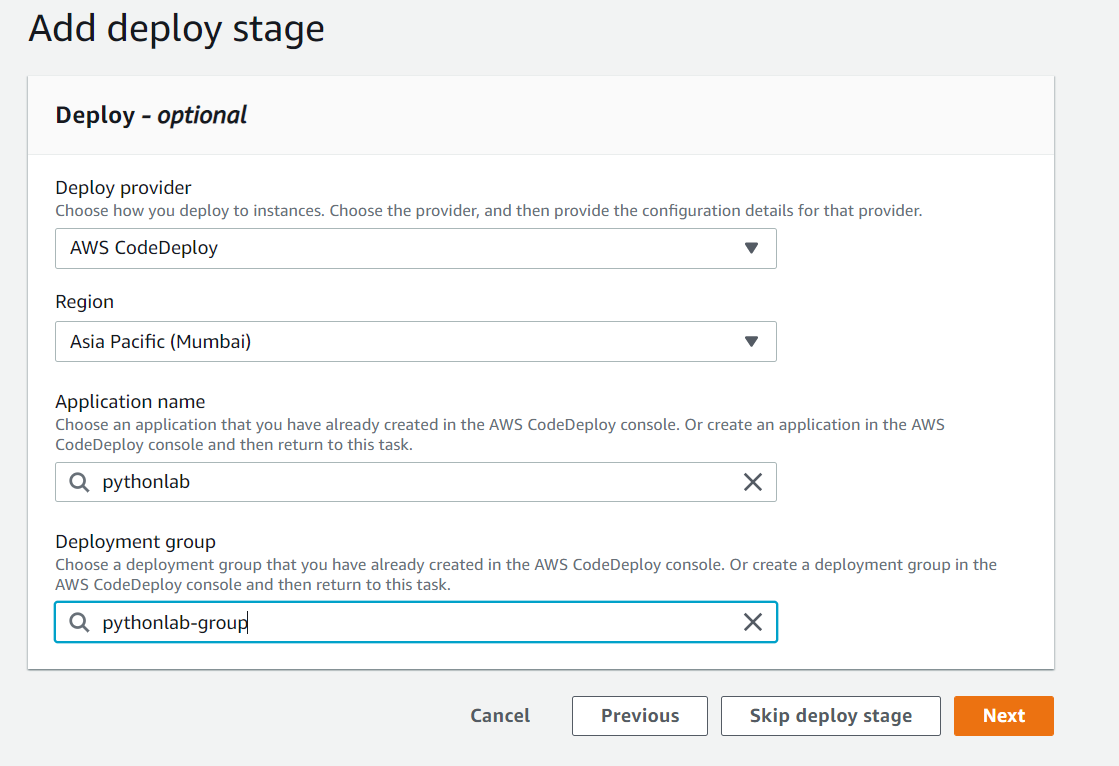


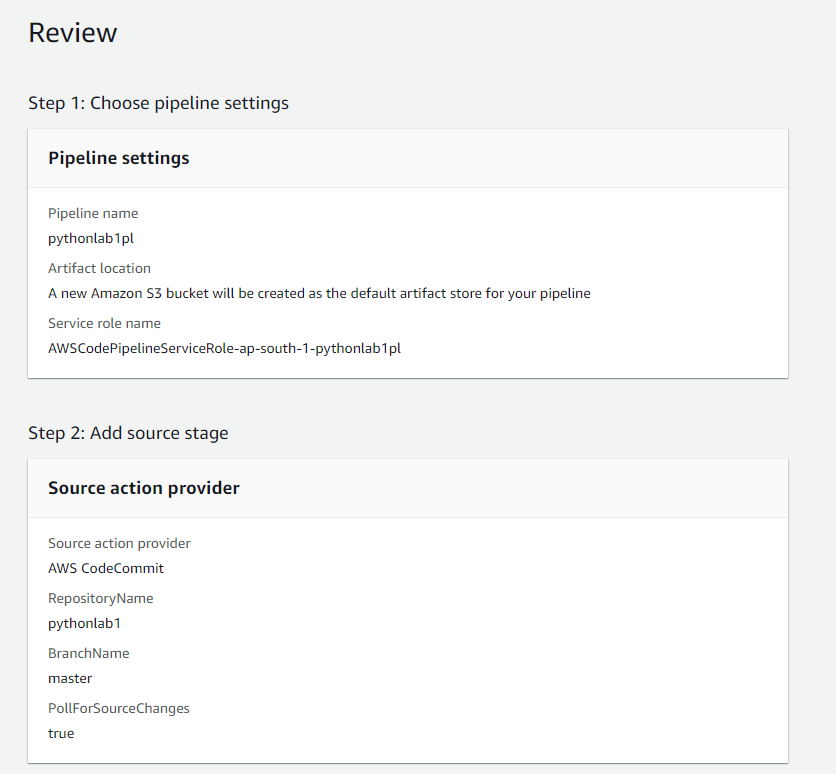


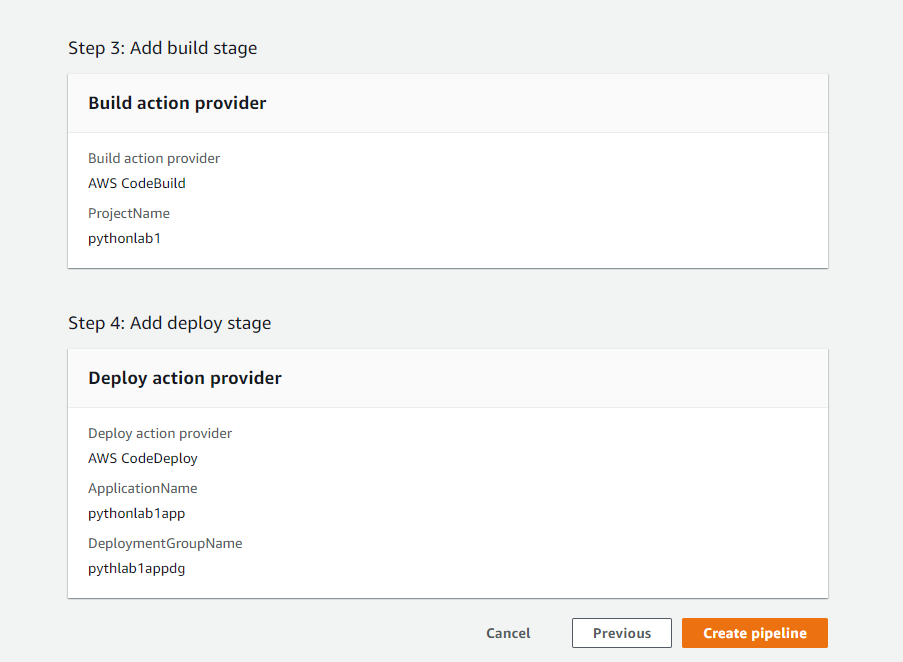


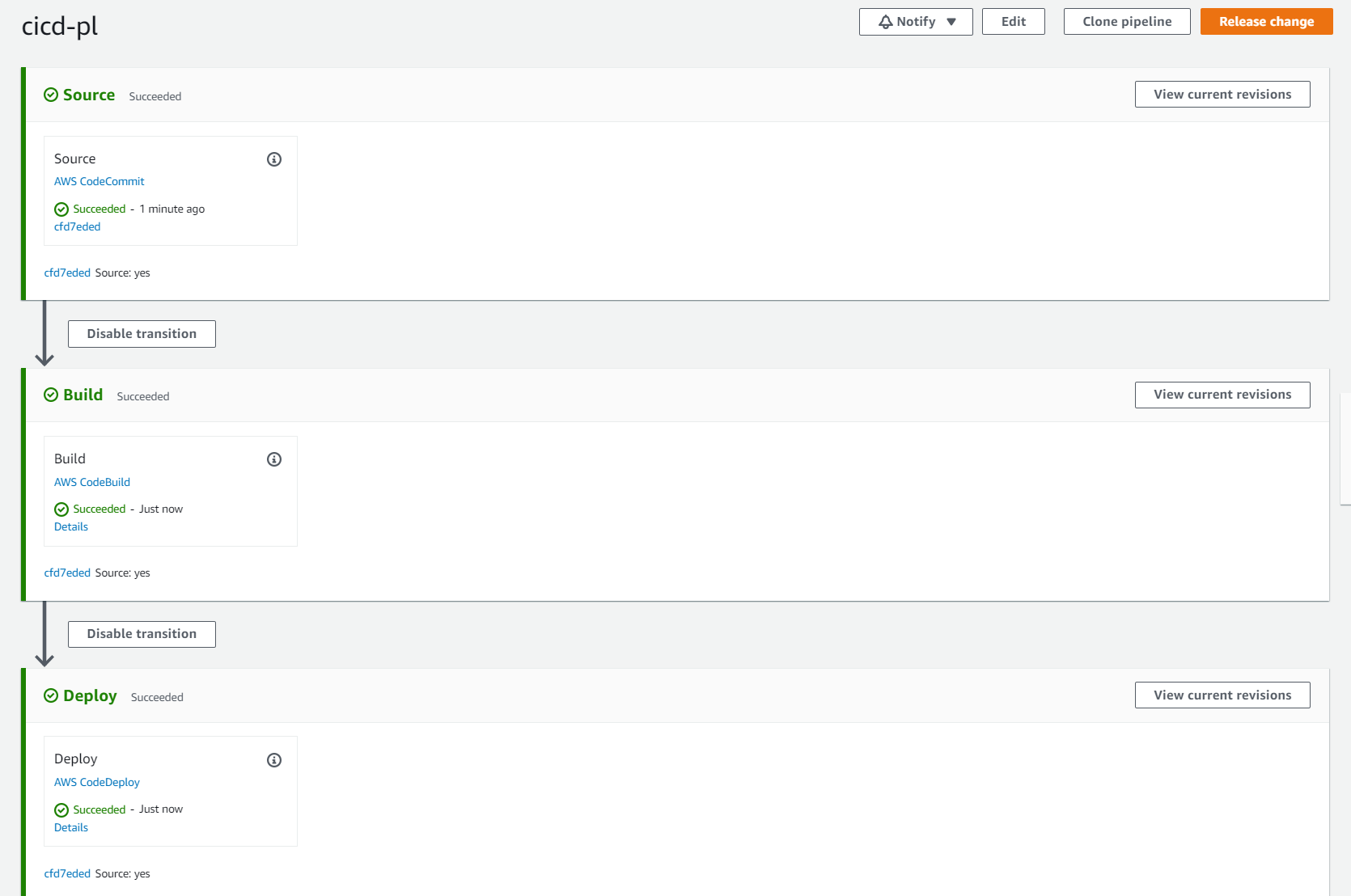




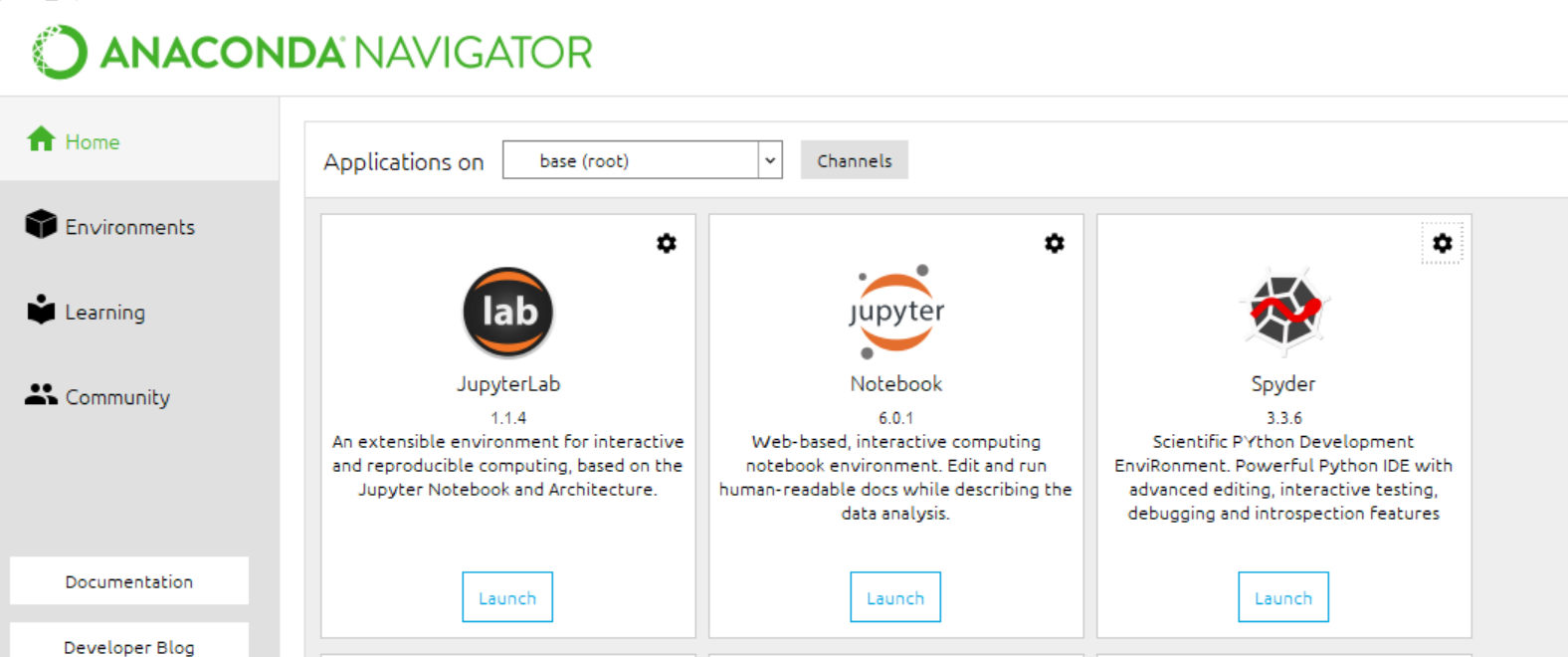






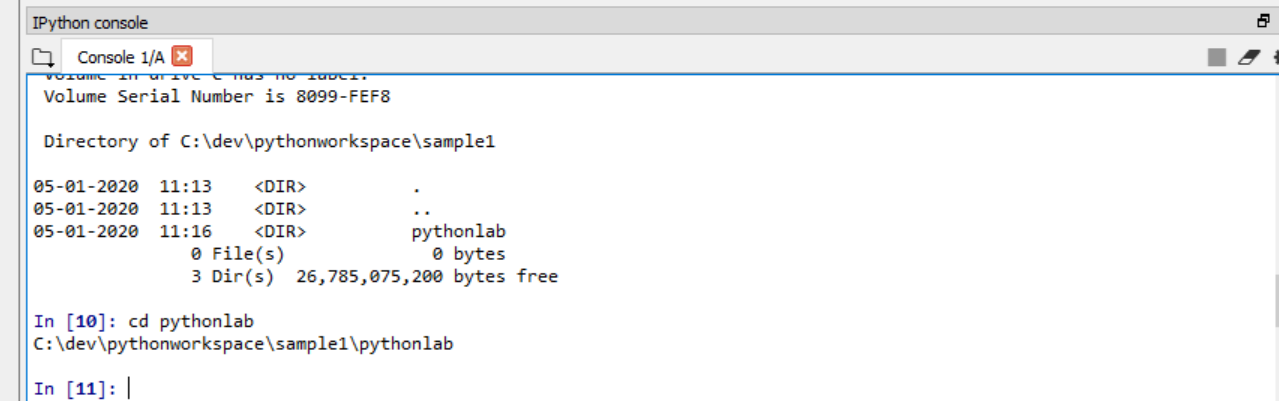


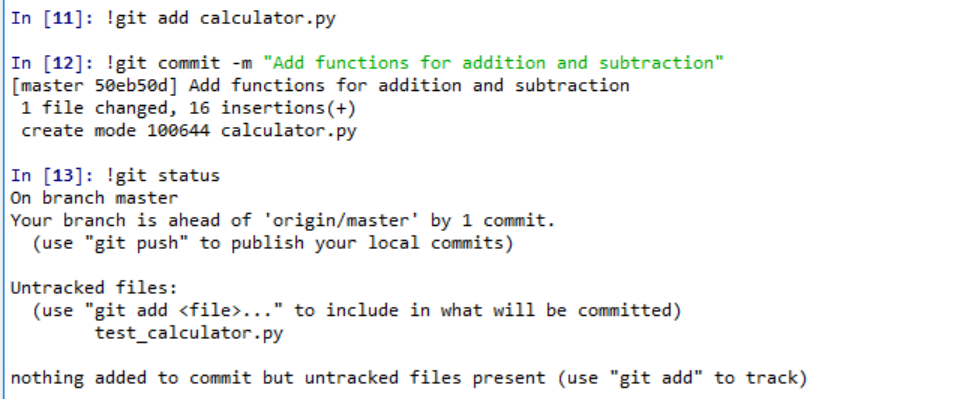
Using Spyder for managing the python code and check-in

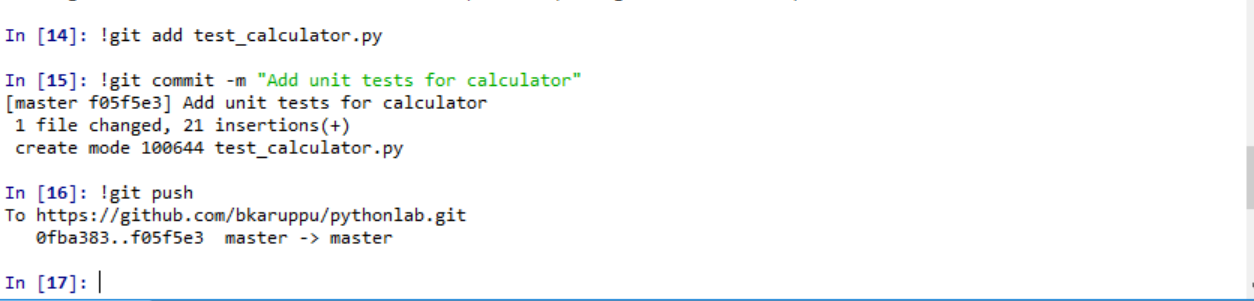


Copy or create .gitignore, calculator.py and test\_calcualtor.py, push the changes to git repository:

Refer to <https://realpython.com/python-continuous-integration/#write-a-simple-python-example>







Modify to test case so that the assert fails:

