



**Ganpat  
University**

॥ विद्यया समाजोत्कर्षः ॥

**Institute of  
Computer  
Technology**

**Name: Tushar Panchal**

**En.No: 21162101014**

**Sub: EADC (Enterprise Application Development for Cloud)**

**Branch: CBA**

**Batch:61**

## **PRACTICAL 09**

### **❖ Question :**

**You have arequirement for deploying an existing Python based application to IBM Cloud. There is a need for automatic scaling for the underlying environment. Implement the IBM Cloud service and resource used to deploy this environment in the quickest way possible.**

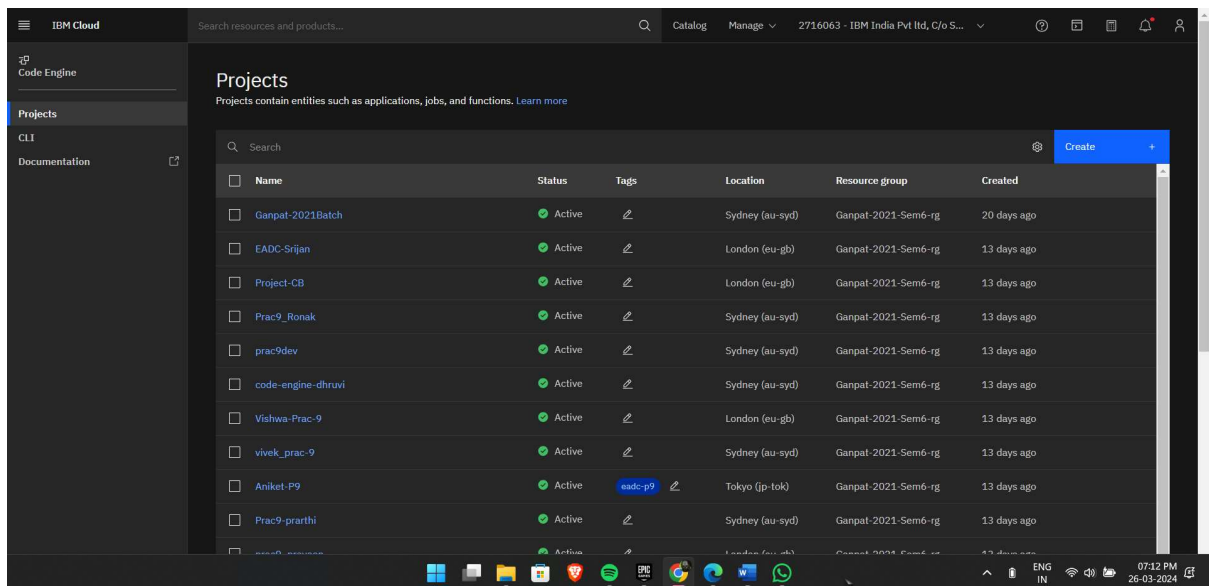
Go through the scenario of development and deployment and perform the following tasks:

1. Developing Python flask application and upload it on github for version management
2. Deploying the application through the IBM Cloud Code engine dashboard.
3. Explore the Autoscaling policies for your application and set the required policies
4. Deploy Node js application on Code engine using Cloud Native Builpacks.

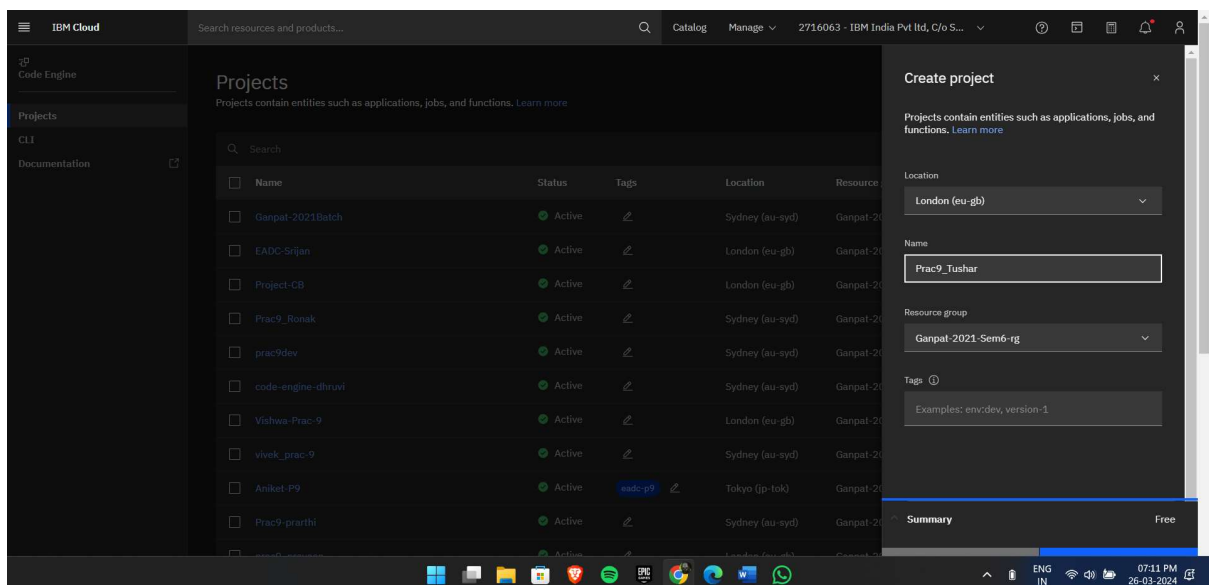
### **TASK:**

**Deploy your portfolio Project or Practical-1 of EADC on Code-Engin**

- » I Logged into the IBM cloud and searched for the codeEngine service, and after that went into the ganpat batch61.



- » Now creating a new application.

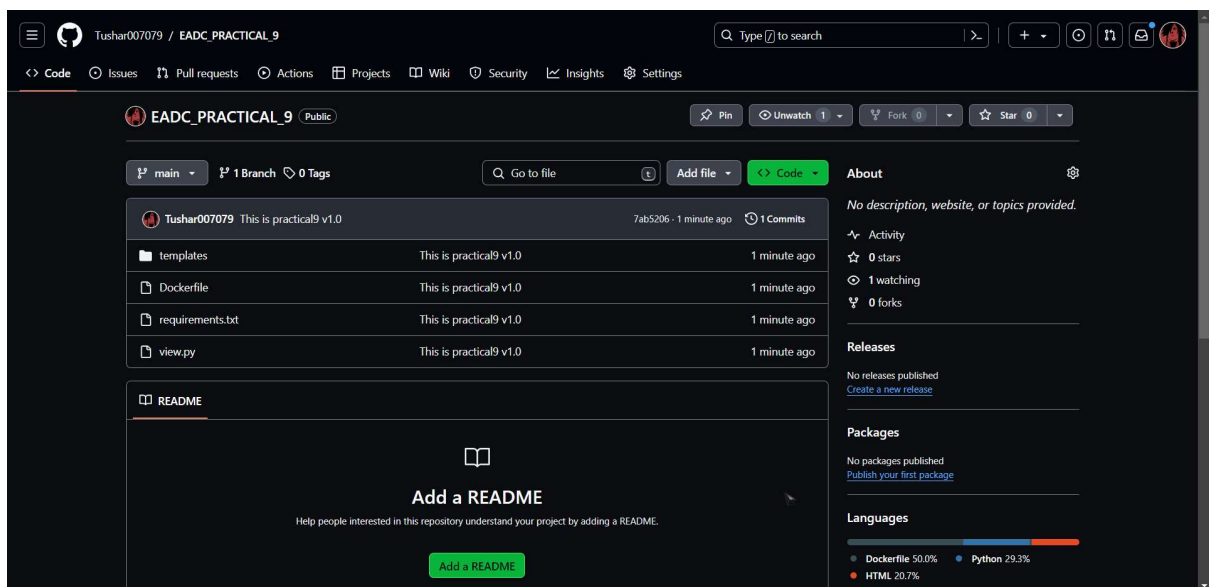


- » Then I downloaded the flask files uploaded by the faculty.
- » Created the repository from the terminal and initialized and pushed all the files from the terminal.

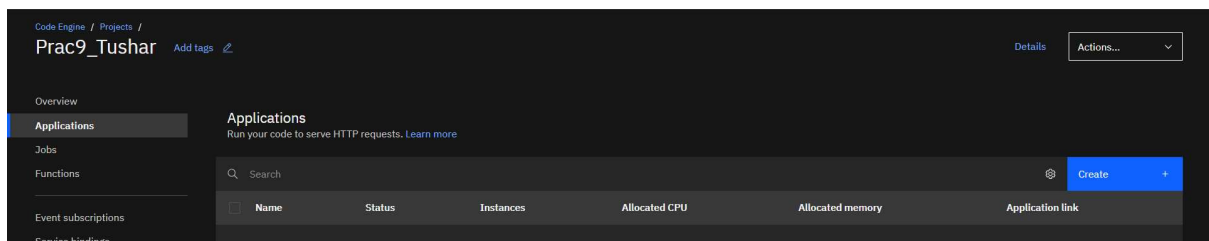
```

> pwsh Practical-9 master # 74 62ms
> git init
Reinitialized existing Git repository in C:/Users/Tushar/Documents/SEM 6/EADC/Practical-9/.git/
> pwsh Practical-9 master # 74 65ms
> git add .
> pwsh Practical-9 master # 74 654ms
> git commit -m "This is practical9 v1.0"
[master (root-commit) 7ab5206] This is practical9 v1.0
4 files changed, 44 insertions(+)
create mode 100644 Dockerfile
create mode 100644 requirements.txt
create mode 100644 templates/index.html
create mode 100644 view.py
> pwsh Practical-9 master # 74 123ms
> git branch -M main
> pwsh Practical-9 main # 74 69ms
> git remote add origin https://github.com/Tushar00709/EADC_PRACTICAL_9.git
> pwsh Practical-9 main # 74 66ms
> git push -u origin main --force
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.19 KiB | 1.19 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Tushar00709/EADC_PRACTICAL_9.git
 * [new branch] main -> main
branch 'main' set up to track 'origin/main'.
> pwsh Practical-9 main # 74 426ms

```



➤ Now in that project we have create a new application.



- » Name it and select build container image from source and don't do any changes in other things

Now hit specify build details in that provide your github repo url

IBM Cloud

Search resources and products...

Code Engine / Projects / Prac9 Tushar / Applications /

Create application

General

Define the basics of the component to create.

Name

tushar-prac-9

Code

Specify a container image or build one from source code first. To learn more, see the documentation.

Use an existing container image

Build container image from source code

Code repo URL

https://github.com/IBM/CodeEngine

Try this sample: https://github.com/IBM/CodeEngine

Specify build details

Resources & scaling

Specify resources for each instance and how Code Engine automatically scales the number of instances

Specify build details

Learn more about image builds.

Source Strategy Output

Code repo URL

https://github.com/Tushar007079/EADC\_PRACTICAL\_

Sample: https://github.com/IBM/CodeEngine

Code repo access

None

Select existing or create new code repo access

Branch name

Example: main

Context directory

Optionally specify an alternate root folder

Cancel Next

In strategy select dockerfile

IBM Cloud

Search resources and products...

Code Engine / Projects / Prac9 Tushar / Applications /

Create application

General

Define the basics of the component to create.

Name

tushar-prac-9

Code

Specify a container image or build one from source code first. To learn more, see the documentation.

Use an existing container image

Build container image from source code

Code repo URL

https://github.com/IBM/CodeEngine

Try this sample: https://github.com/IBM/CodeEngine

Specify build details

Resources & scaling

Specify resources for each instance and how Code Engine automatically scales the number of instances

Specify build details

Learn more about image builds.

Source Strategy Output

Strategy

Dockerfile Cloud Native Buildpack

Dockerfile

Timeout

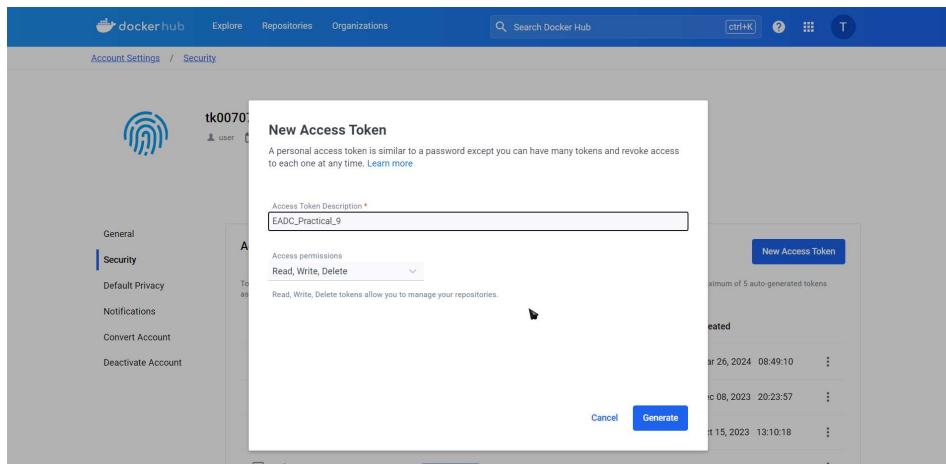
10m

Build resources

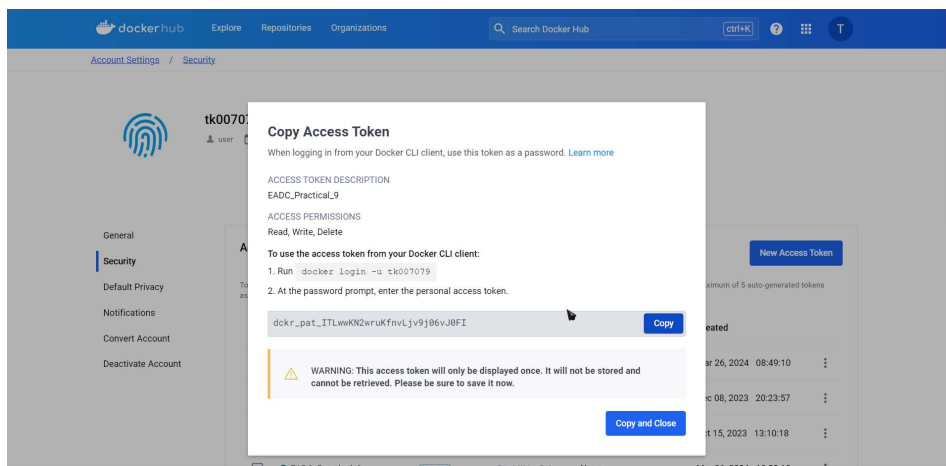
M (1 vCPU / 4 GB)

Previous Next

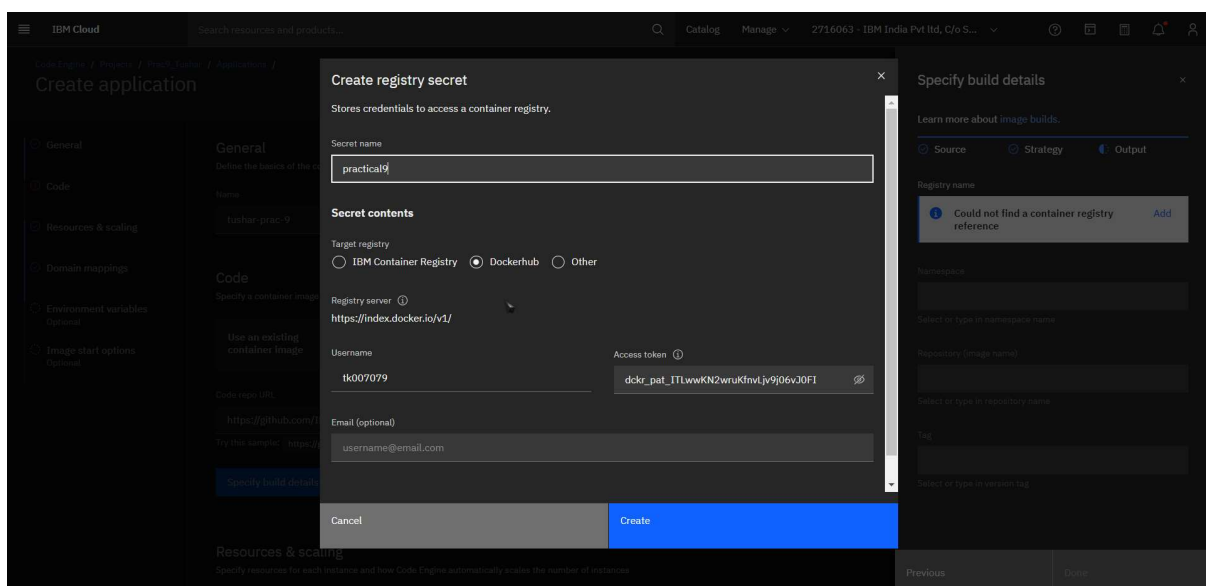
- » Now log in to dockehub and get access token to create registry container for output
- » In dockerhub click on your profile and enter into my account then in security section create new access token:



➤ Then copy that access token



➤ Then in target registry select dockehub and provide that access token and your username of dockerhub



» That's it now our registry secret has been added :

The screenshot shows the 'Specify build details' step of the 'Create application' wizard in IBM Cloud. The left sidebar lists the steps: General, Code, Resources & scaling, Domain mappings, Environment variables (Optional), and Image start options (Optional). The main area is divided into three sections: General, Code, and Resources & scaling. The 'Code' section is active, showing options to 'Use an existing container image' or 'Build container image from source code'. The 'Build container image from source code' option is selected. Below this, the 'Code repo URL' is set to 'https://github.com/IBM/CodeEngine'. A 'Specify build details' button is visible. On the right, a 'Specify build details' panel is open, showing fields for 'Registry server' (https://index.docker.io/v1/), 'Registry secret' (practical9), 'Namespace' (tk007079), 'Repository (image name)' (codeengine-eadc\_practical\_9\_git-46), and 'Tag'. The 'Done' button is highlighted in blue.

» Now hit create to create application:

The screenshot shows the 'Summary' step of the 'Create application' wizard in IBM Cloud. The left sidebar lists the steps: General, Code, Resources & scaling, Domain mappings, Environment variables (Optional), and Image start options (Optional). The main area is divided into three sections: General, Code, and Resources & scaling. The 'Code' section is active, showing options to 'Use an existing container image' or 'Build container image from source code'. The 'Build container image from source code' option is selected. Below this, the 'Image build details' section is expanded, showing fields for 'Image reference' (docker.io/tk007079/codeengine-eadc\_practical\_9\_git-46), 'Registry access secret' (practical9), 'Source code URL' (https://github.com/Tushar007079/EADC\_PRACTICAL\_9.git), 'Branch name' (/), and 'Context directory' (/). An 'Edit build details' link is visible. On the right, a 'Summary' panel is open, showing details for the 'Application' and 'Image Build' components, including pricing details. The 'Create' button is highlighted in blue, and an 'Add to estimate' button is also visible.

» Now test application & send request then access application url:

The screenshot shows the IBM Cloud Code Engine dashboard for an application named 'tushar-prac-9'. The application is in a 'Ready' state. The 'Overview' tab is selected, showing 'Active instances' with 01 instance, 01 CPU (vCPU), and 04 Memory (GB). A graph shows the 'Number of instances' over time. The 'Configuration revisions' table shows one revision, 'tushar-prac-9-00001', with a status of 'Ready' and 100% traffic.

On the right, the 'Test application' panel is open, showing a 'Send request' button and the 'Application URL'. Below, the response of the application is displayed as HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Flask Docker</title>
</head>
<body>
  <h1>This is a Flask App containerised with Docker</h1>
</body>
</html>
```

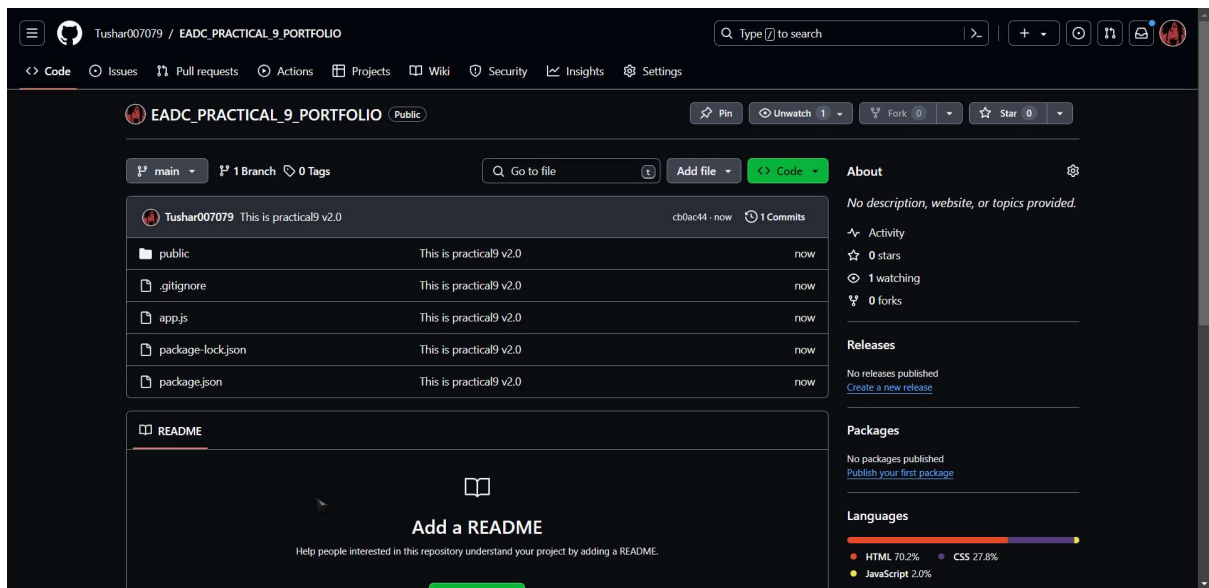
» As you can see our application url is working :

The screenshot shows a web browser window with the URL 'tushar-prac-9.1f0ktlbnos7.eu-gb.codeengine.appdomain.cloud'. The browser displays the application's output, which is a black background with the text 'This is a Flask App containerised with Docker' in white.

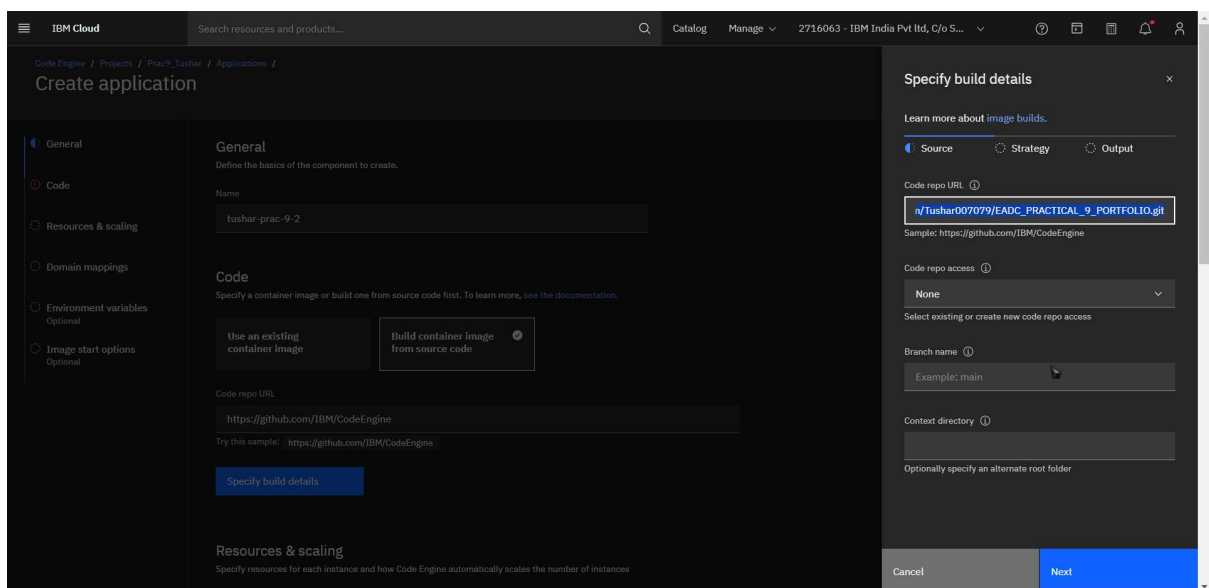
» That's it this is application deployed using dockefile.  
 » Now we have to deploy an application using cloud native



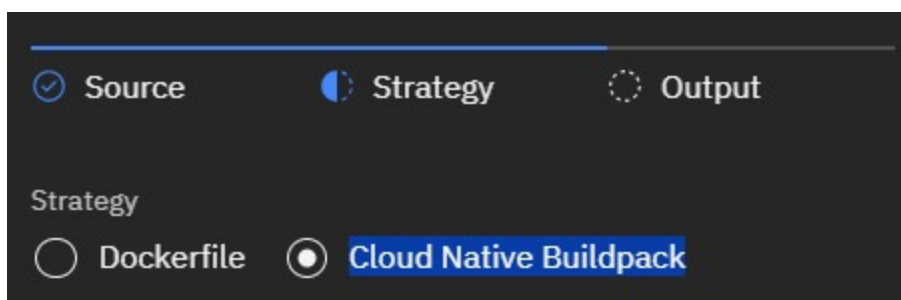
» This is my github repository that I'm using for this deployment:



» now create a new application for this provide your github url :



» Then in strategy select Cloud Native Buildpack:





- » In output don't do any changes we will use previously created:

IBM Cloud

Search resources and products...

Code Engine / Projects / Prac9\_Tushar / Applications /

## Create application

General

Code

Resources & scaling

Domain mappings

Environment variables

Image start options

General

Define the basics of the component to create.

Name

tushar-prac-9-2

Code

Specify a container image or build one from source code first. To learn more, see the documentation.

Use an existing container image

Build container image from source code

Code repo URL

<https://github.com/IBM/CodeEngine>

Try this sample: <https://github.com/IBM/CodeEngine>

Specify build details

Resources & scaling

Specify resources for each instance and how Code Engine automatically scales the number of instances

Specify build details

Learn more about image builds.

Source Strategy Output

Registry server

<https://index.docker.io/v1/>

Select registry or type in registry hostname

Registry secret

practical9

Select existing or create new registry secret

Namespace

tk007079

Select or type in namespace, >ame

Repository (image name)

codeengine-eadc\_practical\_9\_portfolio...

Select or type in repository name

Tag

Previous Done

- » Then simply create application and test it and access url of it.
- » Here this is my portfolio deployed using cloud native buildpack:

