

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

Aim : You have a requirement 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-Engine

Steps and Screenshots :

1. Create nodejs app to host a web application

Code :

```
let express = require('express');
let app = express();

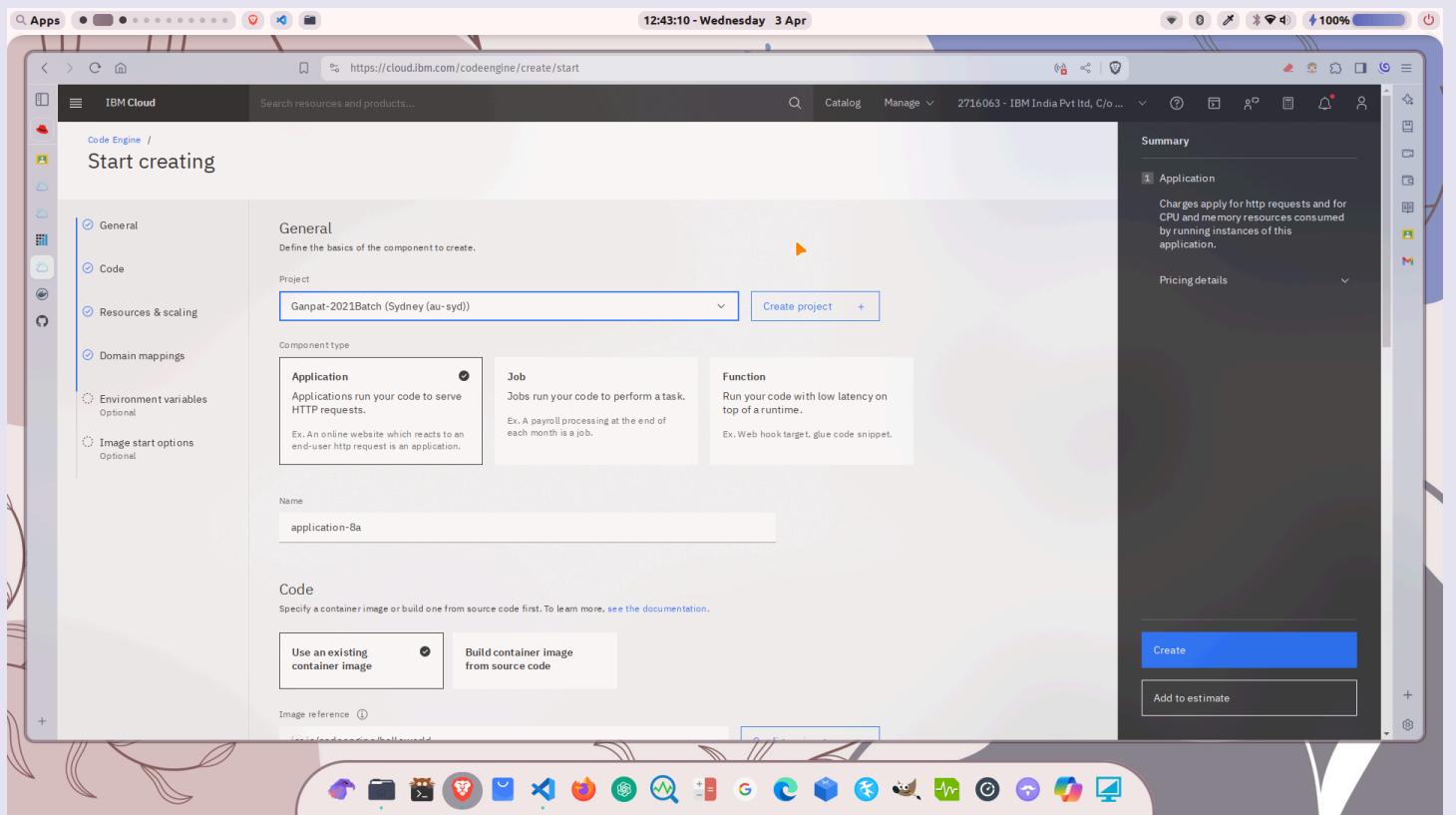
let port = process.env.PORT || 3000;
app.use(express.static('public'));

app.listen(port, function () {
  console.log('Server is listening on port 3000. Ready to accept
requests!');
});
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

Rest code of website available at : <https://github.com/yashslakhtariya/eadc-p9.git>

2. Create new or use existing project to create a new application on IBM Code Engine service



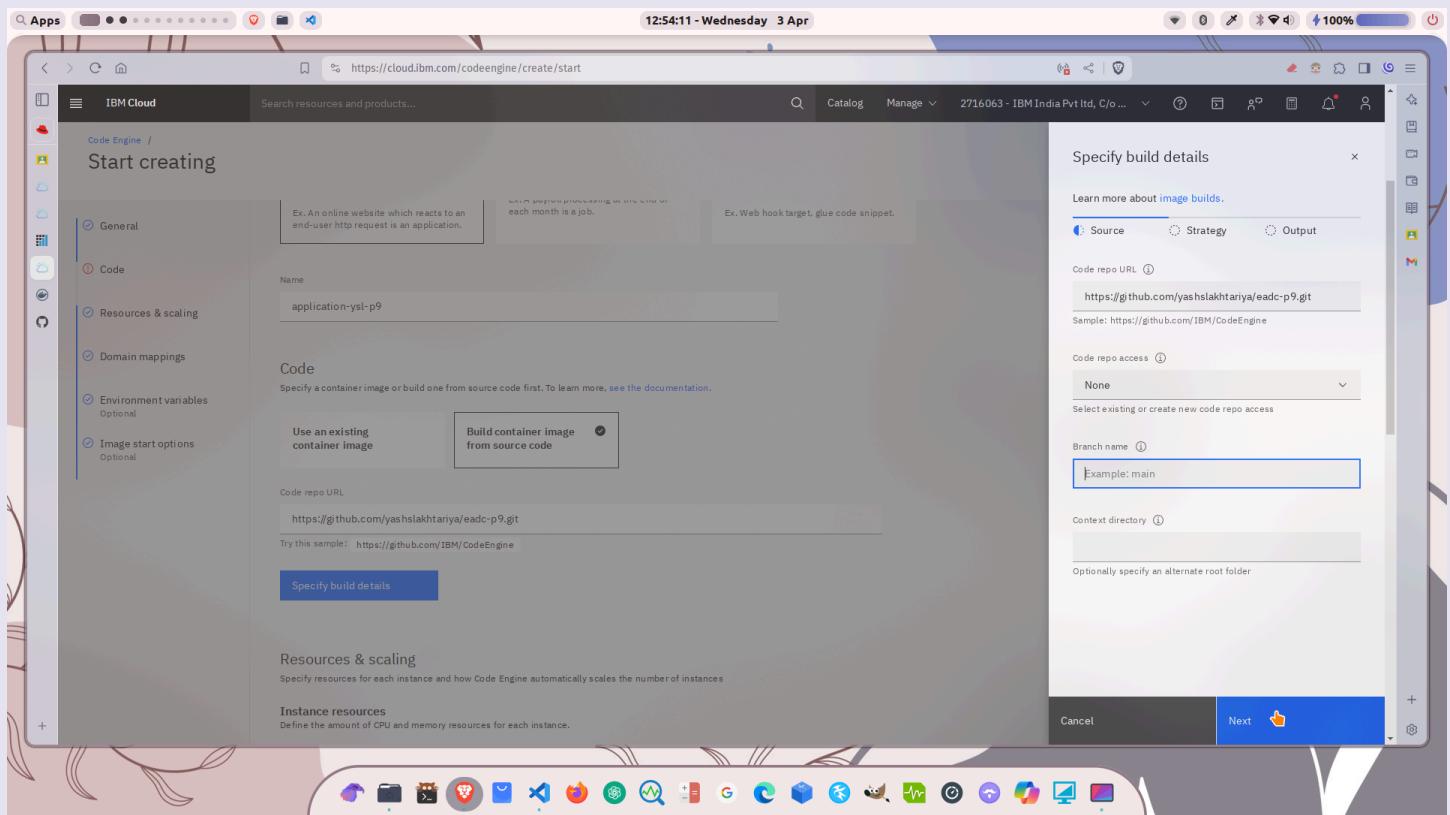
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

3. Specify to build container image from source as we don't have prebuilt image and also Git repo as source

The screenshot shows the 'Start creating' interface for IBM Cloud Code Engine. On the left, a sidebar lists configuration sections: General, Code, Resources & scaling, Domain mappings, Environment variables (Optional), and Image start options (Optional). The 'Image start options' section is currently selected. In the main area, under 'Component type', there are three options: Application (selected), Job, and Function. The 'Application' section describes it as running code to serve HTTP requests. The 'Job' section describes it as running code to perform a task, with an example of a payroll processing at the end of each month. The 'Function' section describes it as running code with low latency on top of a runtime, with an example of a webhook target. Below this, the 'Name' field is populated with 'application-ysh-p9'. Under the 'Code' section, there are two options: 'Use an existing container image' and 'Build container image from source code'. The 'Build container image from source code' option is selected. The 'Code repo URL' field contains the URL 'https://github.com/yashlakhtariya/eadc-p9.git'. At the bottom right, there is a 'Create' button.

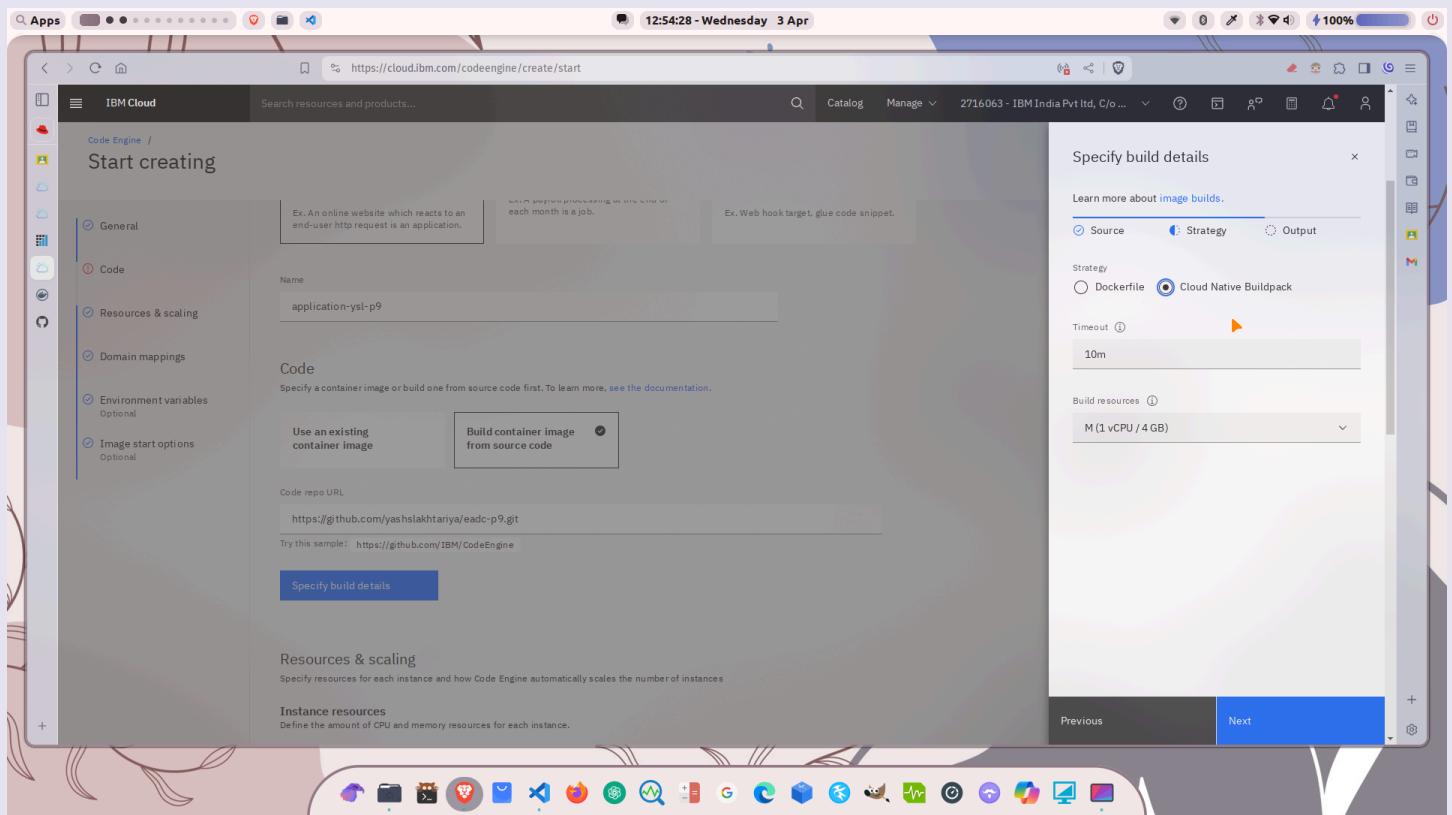
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

4. Specify build details source as Git repository and respective branch



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

5. Strategy, first as Cloud Native Buildpack



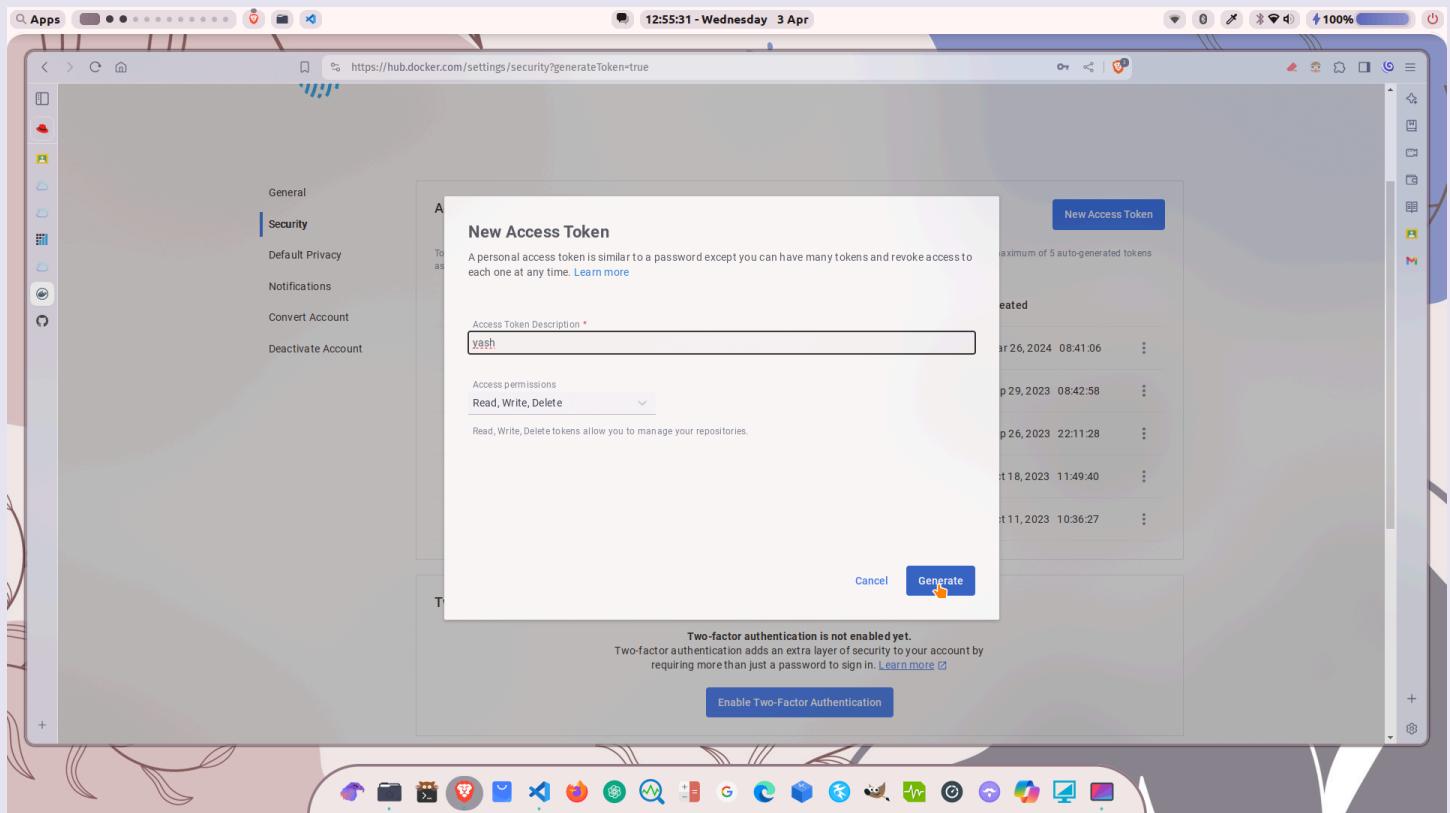
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

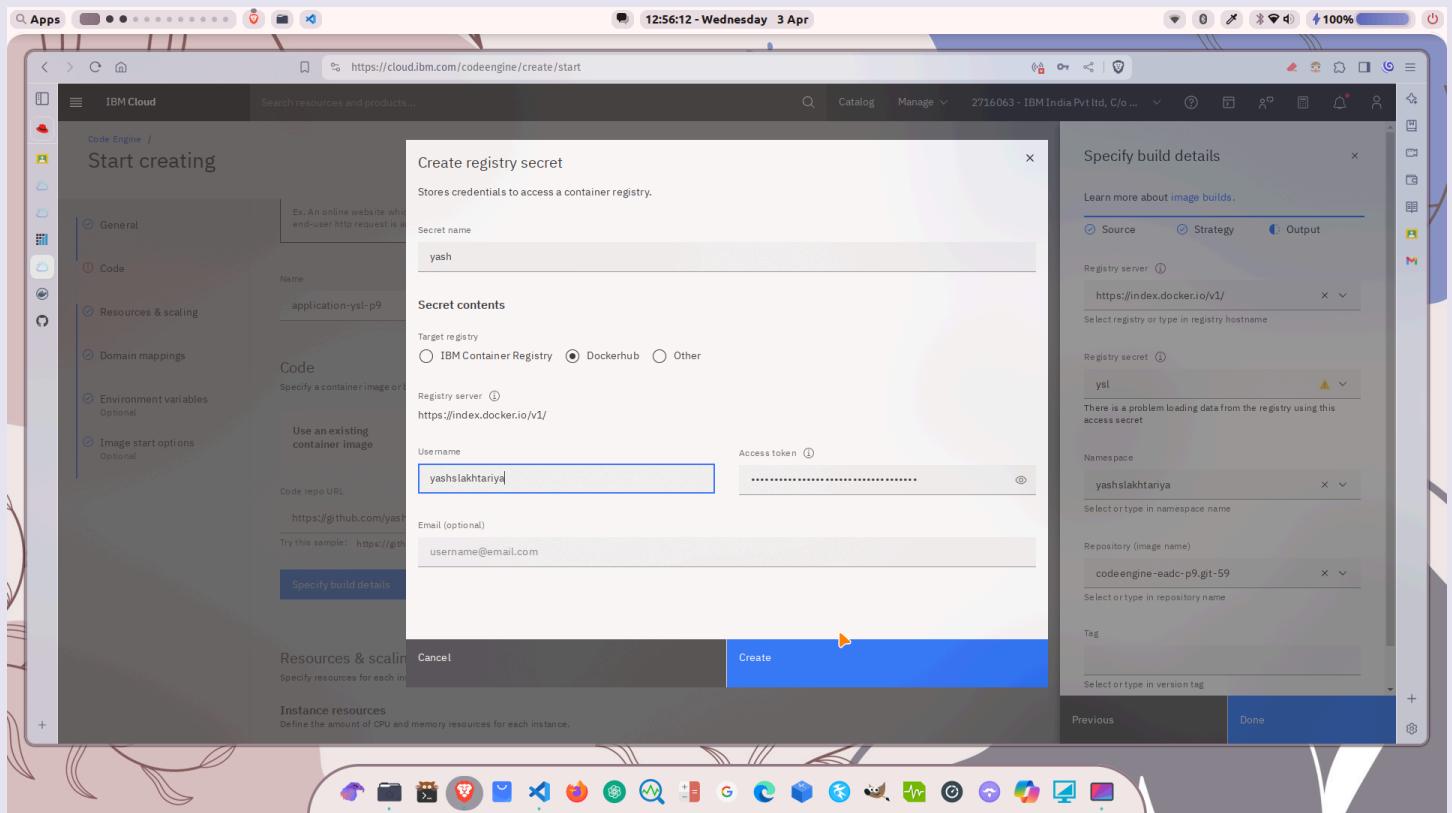
EADC Practical 9

6. For hosting image on Docker hub, create access token from Account Settings on hub.docker.com with all permissions



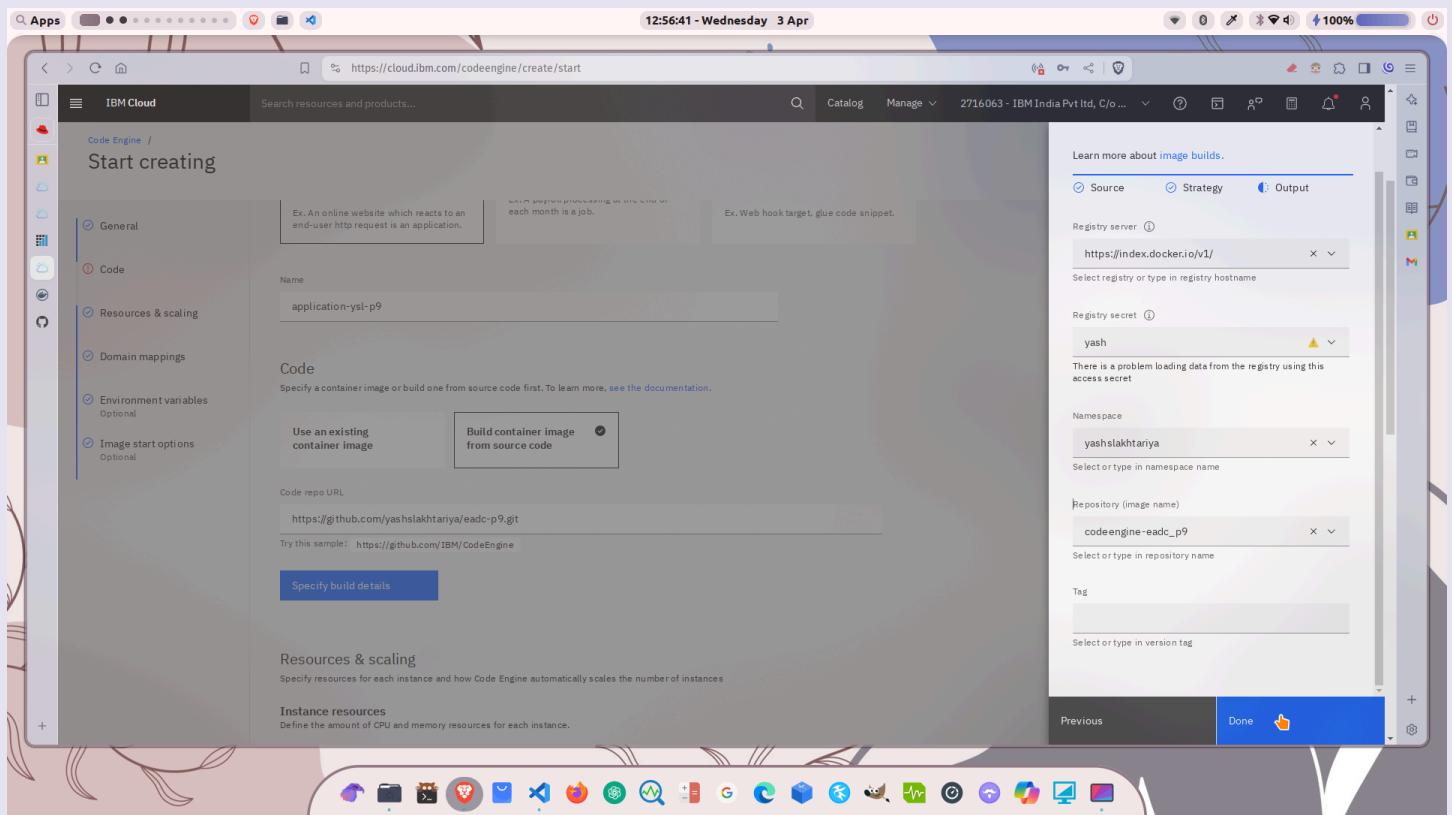
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

7. Create new registry secret in Output section, with dockerhub access token details and username



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

8. Specify image name if required else keep default



The screenshot shows the 'Start creating' wizard for IBM Cloud Code Engine. The left sidebar lists steps: General, Code, Resources & scaling, Domain mappings, Environment variables (Optional), and Image start options (Optional). The main area is titled 'Code' and shows a 'Name' field containing 'application-yasl-p9'. Below it, there are two options: 'Use an existing container image' and 'Build container image from source code', with the latter selected. A 'Code repo URL' field contains 'https://github.com/yashslakhtariya/eadc-p9.git'. The right side of the screen displays configuration details for an image build, including:

- Source: Registry server set to 'https://index.docker.io/v1/'.
- Strategy: Registry secret set to 'yash'.
- Output: Namespace set to 'yashslakhtariya'.
- Repository (image name): Set to 'codeengine-eadc_p9'.
- Tag: Placeholder for selecting or typing a version tag.

At the bottom right, there are 'Previous' and 'Done' buttons, with 'Done' being highlighted.

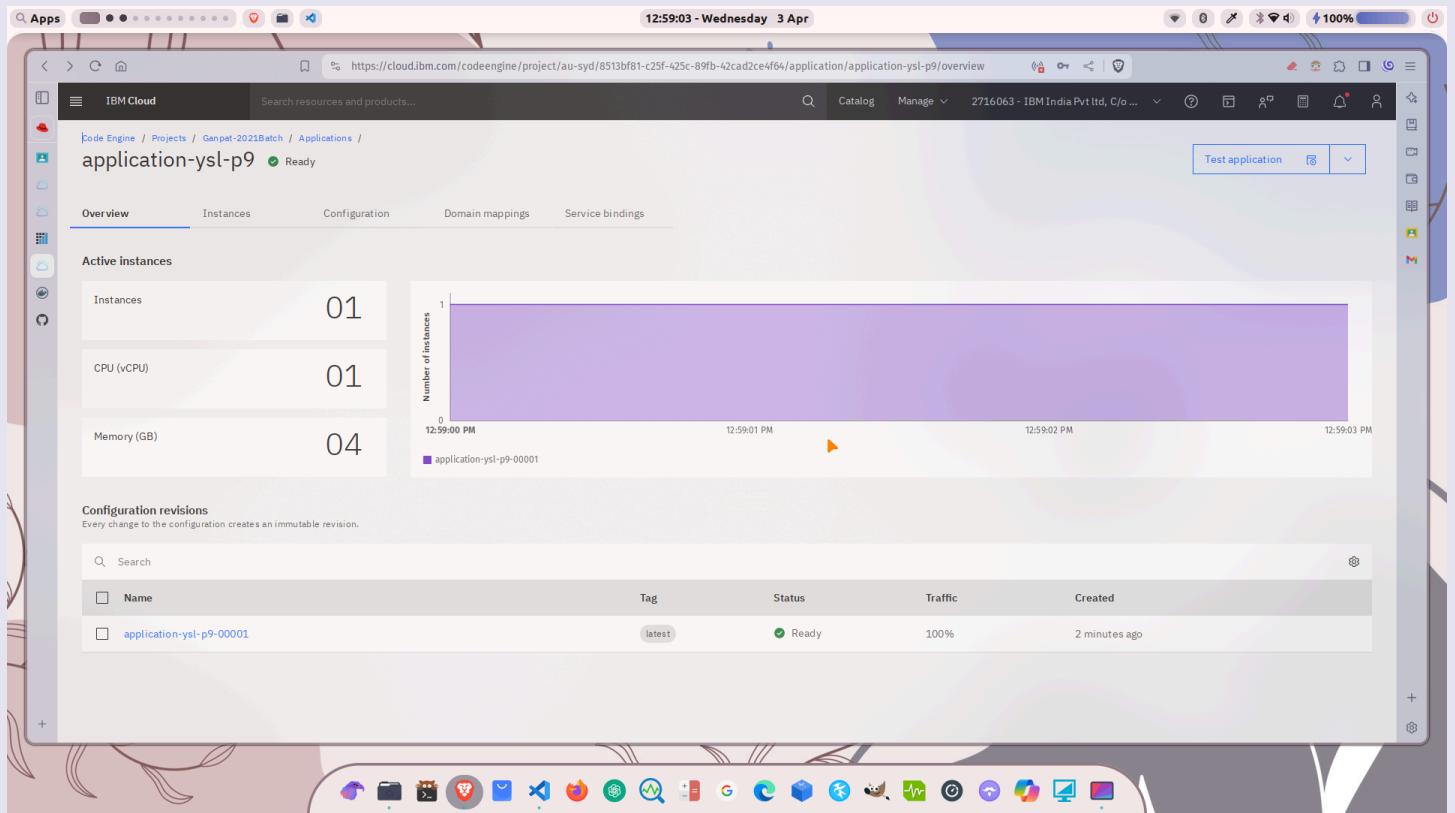
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

9. Now, create the application

The screenshot shows the IBM Cloud Code Engine creation interface. The URL in the browser is <https://cloud.ibm.com/codeengine/create/start>. The page title is "Start creating". On the left, there's a sidebar with navigation links: General, Code, Resources & scaling, Domain mappings, Environment variables (Optional), and Image start options (Optional). The "General" section is selected. It contains fields for "Project" (Ganpat-2021Batch (Sydney (au-syd))) and "Component type" (Application, which is selected). The "Application" section describes applications as running code to serve HTTP requests. The "Job" section describes jobs as performing tasks like payroll processing. The "Function" section describes functions as running code with low latency. Below these sections, there's a "Name" input field with "application-ysl-p9" typed in. In the bottom right corner of the main form area, there's a large blue "Create" button with a hand cursor icon pointing at it. To the right of the main form, there are two summary sections: "Summary" (Application, Pricing details, and Image Build) and "Pricing details" (for Application and Image Build).

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

10. Wait for instances to be ready



The screenshot shows the IBM Cloud Code Engine application overview page for 'application-ysl-p9'. The application is marked as 'Ready'. The 'Overview' tab is selected, displaying resource details and a timeline chart.

Active instances:

Instances	CPU (vCPU)	Memory (GB)
01	01	04

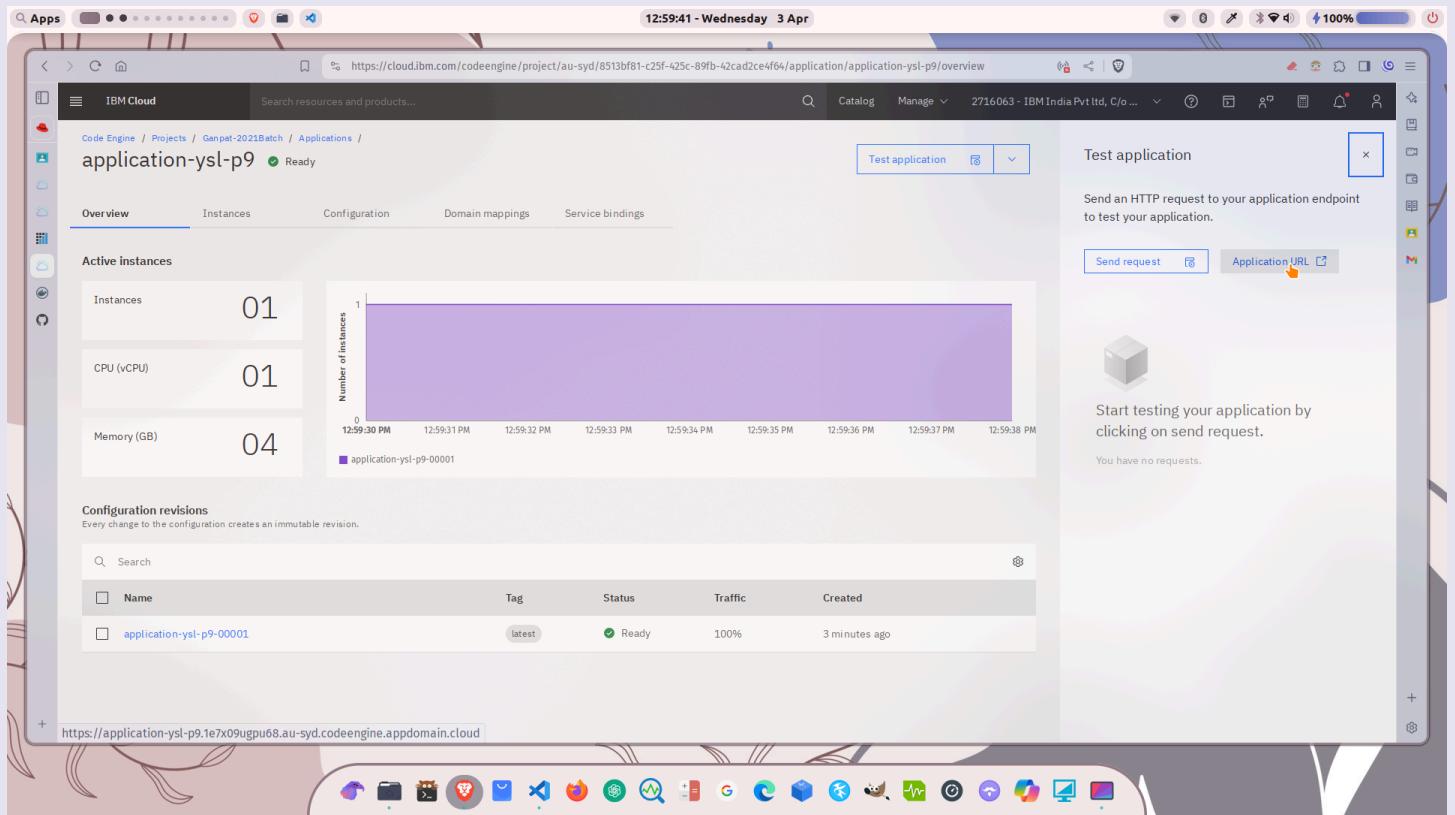
Configuration revisions:

Name	Tag	Status	Traffic	Created
application-ysl-p9-00001	latest	Ready	100%	2 minutes ago

Timeline Chart: A chart showing the number of instances over time. It shows a single instance starting at 12:59:00 PM and remaining at 1 instance until 12:59:03 PM.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

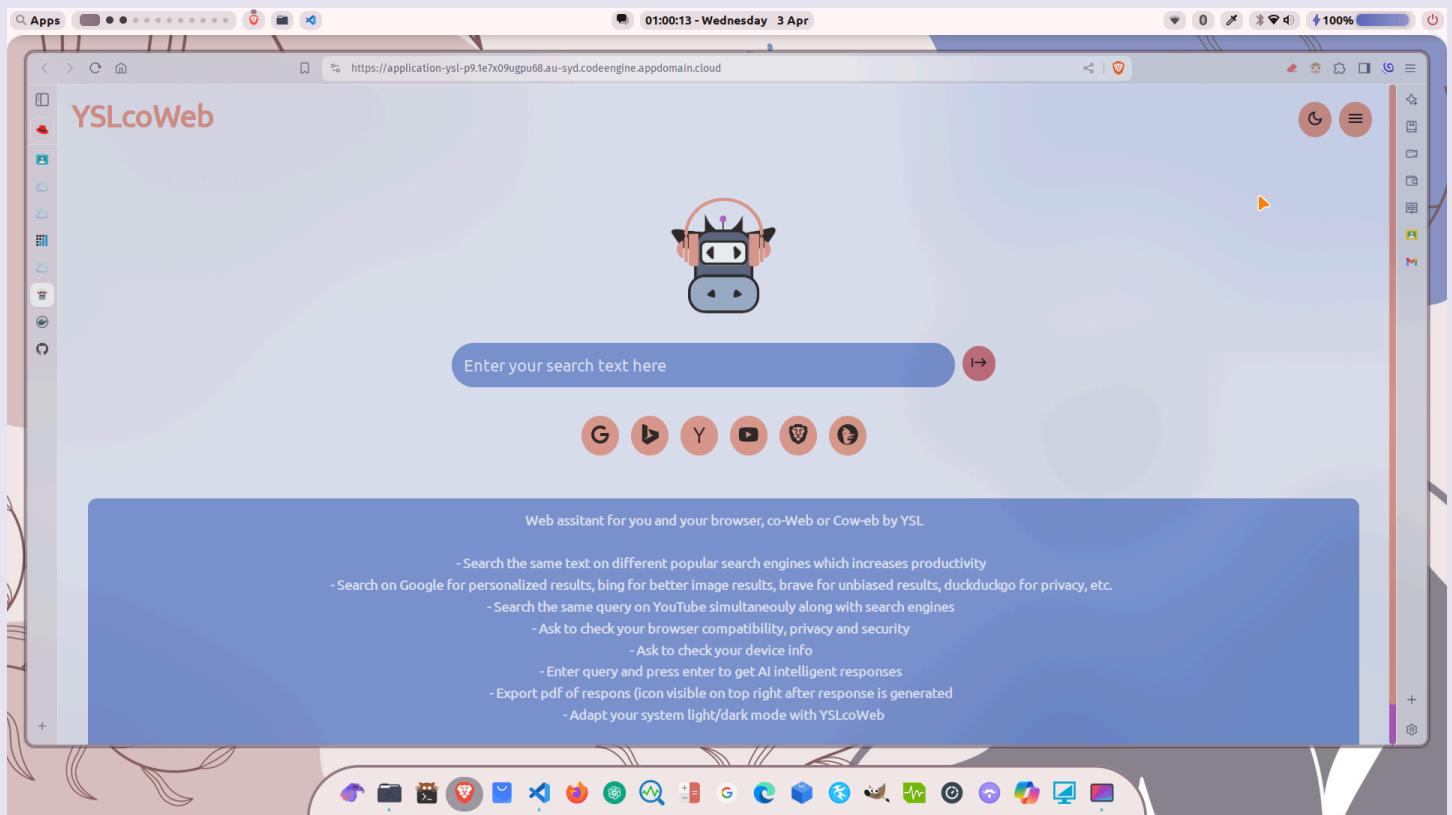
11. Afterwards, select Test Application option and open with URL



The screenshot shows the IBM Cloud Code Engine application overview page for the application `application-ysl-p9`. The application is marked as `Ready`. The `Overview` tab is selected, displaying resource usage: 1 instance, 1 vCPU, and 0.4 GB of memory. A chart shows 1 instance active from 12:59:30 PM to 12:59:38 PM. The `Test application` section contains a button to `Send request` and a link to the `Application URL`, which is highlighted with a yellow arrow. The URL shown in the browser address bar is `https://application-ysl-p9.1e7x09ugpu68.au-syd.codeengine.appdomain.cloud`.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

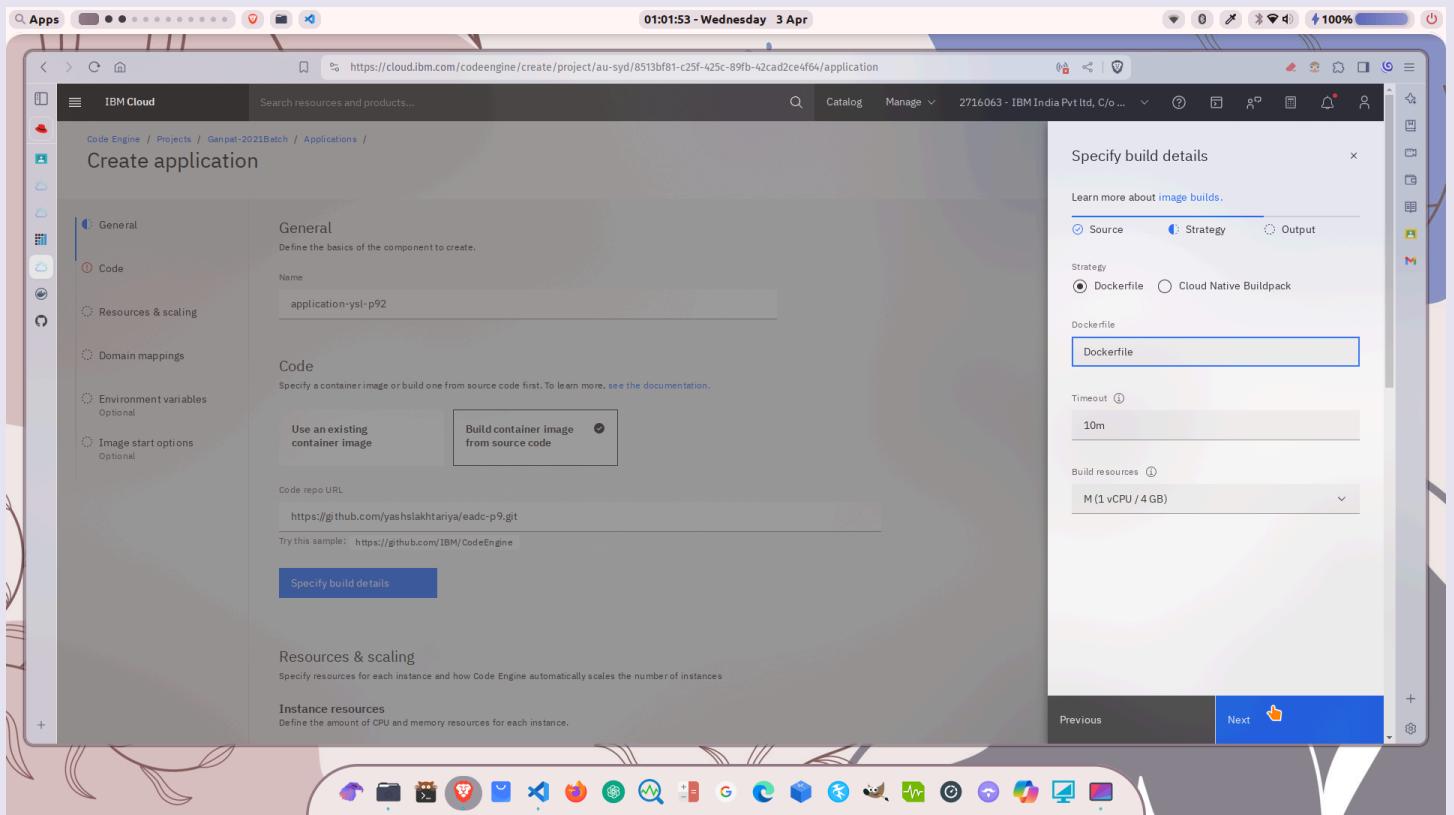
12. Our web application is ready as seen below



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

→ If Dockerfile is used to build image the the steps would be same accept the change in strategy tab :

1. Select Dockerfile in Strategy to build, making sure that Dockerfile exists in the root folder of the repository

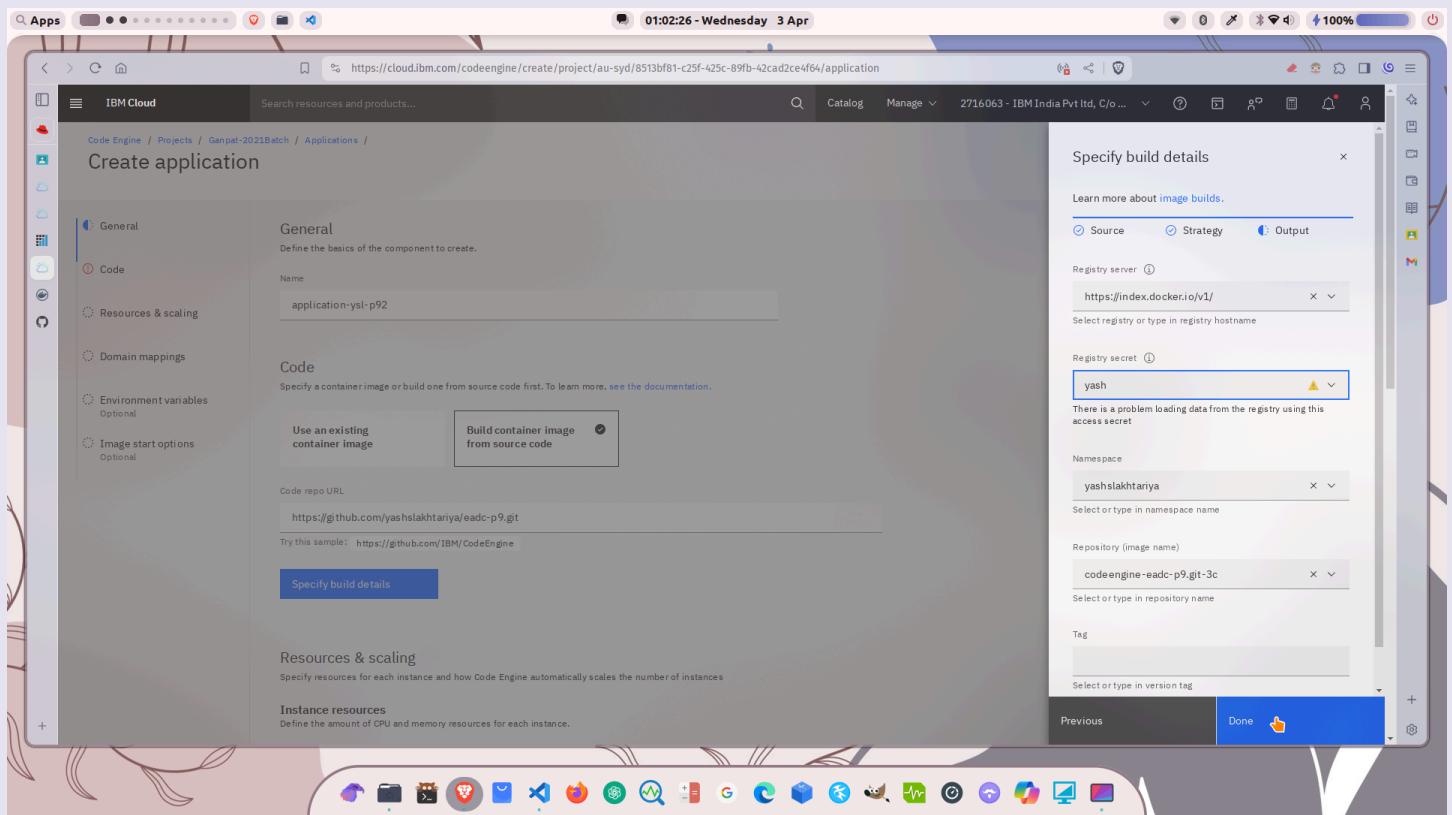


Dockerfile used here :

```
FROM node:14
WORKDIR /app
COPY package*.json .
RUN npm install express
COPY . .
EXPOSE 3000
CMD [ "node", "app.js" ]
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

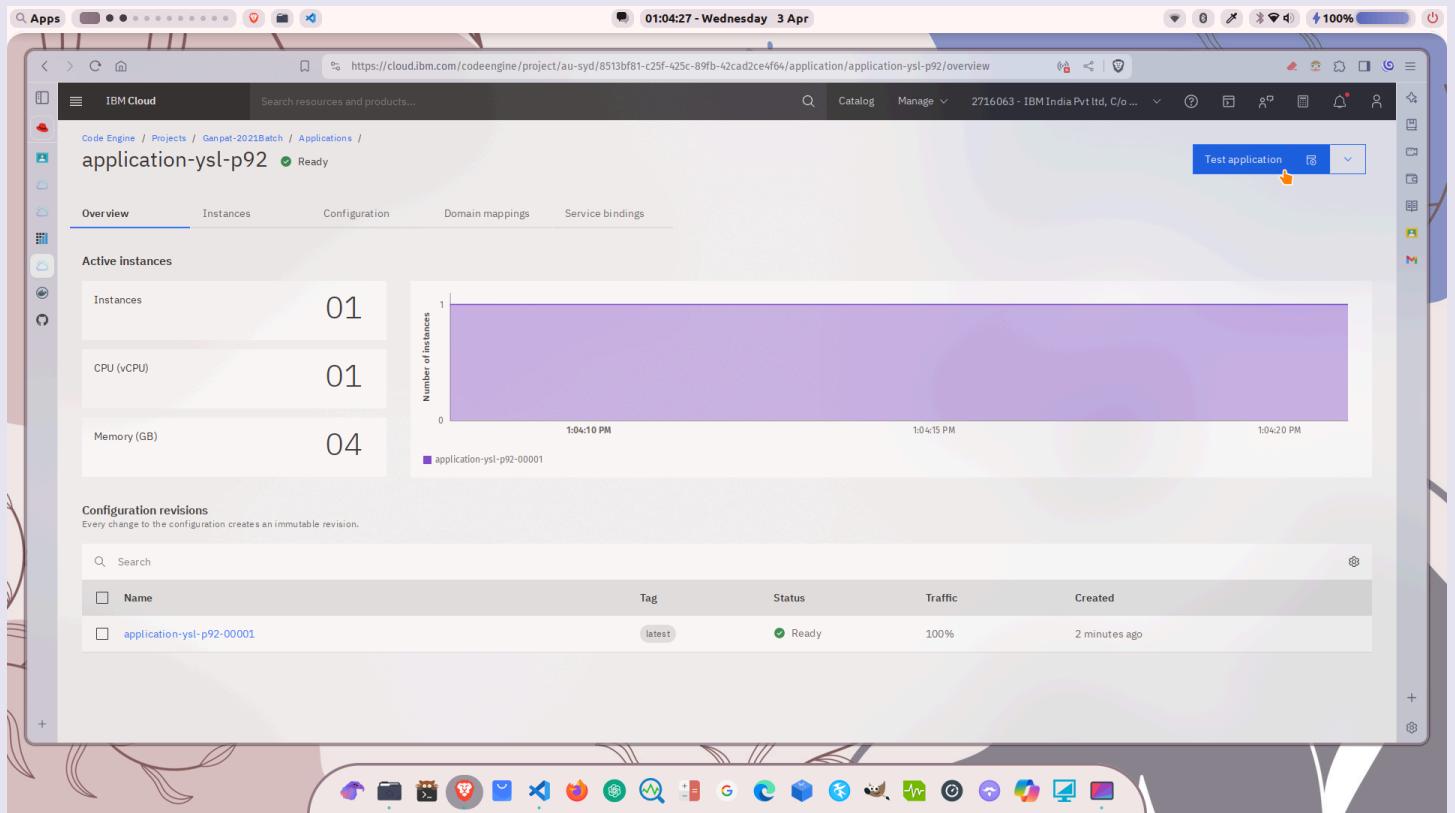
2. Follow same steps to specify output and image name



The screenshot shows the 'Create application' page in the IBM Cloud Code Engine interface. On the left, a sidebar lists options like General, Code, Resources & scaling, Domain mappings, Environment variables, and Image start options. The 'General' section is active, showing a 'Name' input field with 'application-ysh-p92'. The 'Code' section has a radio button for 'Build container image from source code' selected, with a 'Code repo URL' input field containing 'https://github.com/yashslakhtariya/eadc-p9.git'. A 'Specify build details' button is visible. The 'Resources & scaling' section is partially visible at the bottom. On the right, a modal window titled 'Specify build details' is open, showing tabs for Source, Strategy, and Output (Output is selected). It includes fields for Registry server ('https://index.docker.io/v1/'), Registry secret ('yash' - with a warning message about access secret), Namespace ('yashslakhtariya'), Repository ('codeengine-eadc-p9.git-3c'), and Tag. A 'Done' button with a checkmark is at the bottom right of the modal.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 9

3. Wait for instance to be ready



The screenshot shows the IBM Cloud Code Engine application overview for 'application-ysl-p92'. The application is marked as 'Ready'. The 'Overview' tab is selected, displaying metrics for 01 instance, 01 vCPU, and 04 GB Memory. A chart shows 1 instance active from 1:04:10 PM to 1:04:20 PM. The 'Configuration revisions' section lists a single revision named 'application-ysl-p92-00001' with a 'latest' tag, 'Ready' status, 100% traffic, and created 2 minutes ago. A status bar at the top right indicates 'Test application'.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 9

4. Now, test application and visit URL, below is the same application built using dockerfile

