D D M M Y Y Y Y

Experiment - 11

Creating Release Pipelines : Deploying Applications to Azure App Services, Managing secrets and Configuration with Azure Key Vault, Hands-on : Continous Deployment with Azure Pipelines.

Step 1 : Download and extract the Spring Boot application from the following link :
https://drive.google.com/file/d/1Te3FJFm30A3D8sb
G13QYqKRWMxfPFxd3U/view? usp = sharing.

Step 2 : open the extracted file in intelliJ idea and run the spring Boot application.

Step 3 : open the browser and verify the application by visiting :
http://localhost:8080/api/data
Observe the expected output.

Step 4 : Stop the application after successfull verification.

Step 5 : Create a new Github repository and push the spring Boot Project code into it.

D D M M Y Y Y Y

Step 6: Go to your Github account and create a new repository (make it Public).

- Copy the HTTPS clone URL.
- On your local machine, navigate the Project directory using terminal or Git Bash.
- Run the following GIT commands:

```
git init
git remote add origin <your-repository -URL>
git add
git commit -m "Initial Commit"
git branch -M main
git Push -u origin main
```

Step 7: Deploy to Azure:

- Login to Azure Portal
- Go to App services and create a new Web App.
- Provide a name, select Java 17, choose Linux Hosting Plan
- Enable Continous Deployment and link your Github repository.
- Choose the organisation, repository and the main branch.
- Enable basic Authentication, click Review + Create, then Create.
- After deployment, go to the resource and Copy the default domain link.

- Paste the domain URL in the browser and verify output.

Step 8: Configure Github Actions for CI/CD:

- In your Github repository, go to the Actions tab.
- Create a new workflow file in .github/workflows/main_SJBIT_CSE-test.yml
- Paste the given code in Github Action YAML code.

* Note: Make sure your Azure credentials (client-id, tenant-id, subscription-Id) are added to your Github secrets.