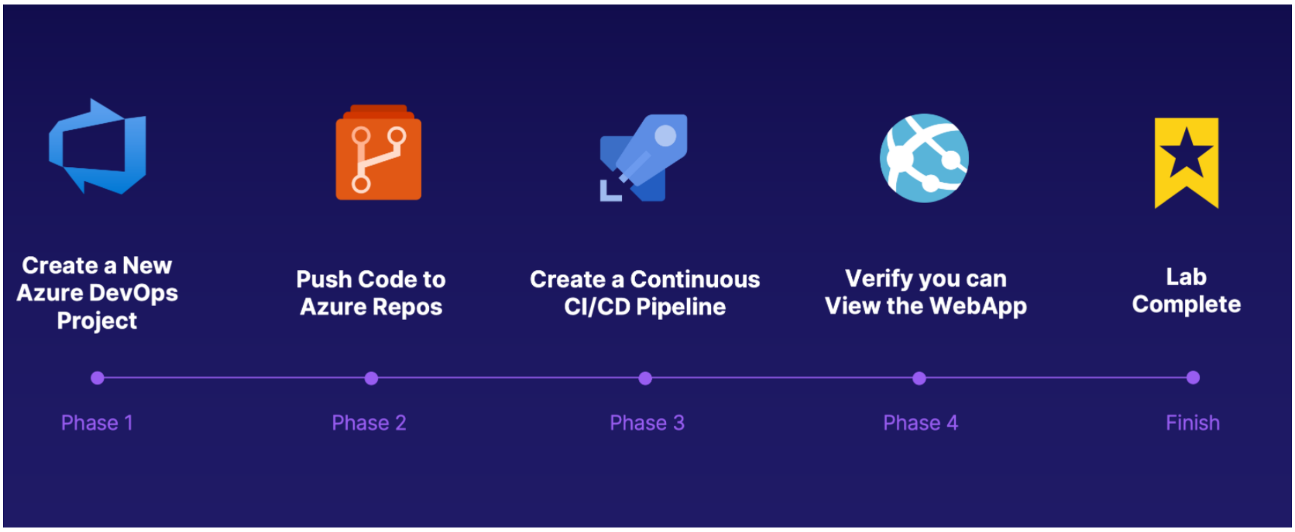
Create a CI/CD Pipeline for Azure WebApp for Containers

Introduction

You've been given an application, and you must setup a continuous delivery pipeline for it in Azure DevOps. Import the code into Azure Repos and create the YAML pipeline that will continuously build and deploy this application.


Create a New Azure DevOps Project
Push Code to Azure Repos
Create a Continuous CI/CD Pipeline
Verify you can View the WebApp
Lab Complete
Phase 1
Phase 2
Phase 3
Phase 4
Finish

Create an Azure DevOps Organization

1. In the Azure Portal, click the Cloud Shell icon (>_) in the upper right.
2. Select PowerShell.
3. Click Show advanced settings.
4. For Storage account, select Create new and give it a globally unique name (e.g., "cloudshell" with a series of numbers at the end).
5. For File share, select Create new and give it a name of "fileshare1".
6. Click Create storage.
7. Retrieve the subscription information:

   ```
   Get-AzSubscription
   ```
8. Copy the Name, Id, and TenantId for later use.
9. In the left-hand menu, click All services.

10. In the search box, type in "Azure DevOps", and select Azure DevOps organizations.
11. Click My Azure DevOps Organizations.
12. For the region, choose United States.
13. Click Continue.
14. Click Create new organization.
15. Click Continue.

Leave the organization, and host region as the default settings, and click Continue.

Import code to Azure Repos

1. For the project name, enter "MyFirstProject".
2. Click Create project.
3. In the left-hand menu, click Repos.
4. Click Import.
5. In the Clone URL field, paste the following:

https://github.com/kascade-training/content-az400-lab-resources

6. Click Import.
7. Once imported, switch to the docker branch.

Create the CI/CD Pipeline

1. In the left-hand menu, click Pipelines.
2. Click Create Pipeline.
3. Click Azure Repos Git.
4. Click the project from the list.
5. Click the Existing Azure Pipelines YAML file.
6. For the Branch, select docker.
7. For the Path, select /azure-pipelines.yml, and click Continue.
8. At the bottom of the page, open the Project Settings page in a new tab.
9. Under Pipelines, click Service connections.
10. Click Create service connection.
11. Select Azure Resource Manager, and click Next.
12. Select Service principal (manual), and click Next.

13. Using the information copied at the beginning of the lab, and the service principal provided in the Lab Credentials, enter the following:

- Subscription Id: <SUBSCRIPION_ID>
- Subscription Name: <SUBSCRIPTION_NAME>
- Service Principal Id: <APPLICATION_CLIENT_ID>
- Service Principal Key: <SECRET>

14. Click Verify
15. For the Service connection name, enter "AzureSC"
16. Click Verify and save.
17. Click New service connection.
18. Select Docker Registry, and click Next.
19. For the Docker ID and Docker Password fields, use your Docker Hub credentials.
20. For the Service connection name, enter "Docker", and click Save.
21. Back on the pipeline review page, click Save and run.
22. Leave the settings as default, and click Save and run again.
23. Click on the Build stage to monitor the build process.
24. Once built, go back to the Azure Portal, and navigate to All resources.
25. Click the web app from the list.
26. Click on the web app URL to verify connectivity to the app.

Conclusion

Congratulations — you've completed this hands-on lab!