| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

☐ **Mark Complete**

Implement phased rollout with rollback

Resist the initial temptation to rush on the keyboard! This is the time to pause and think as a team about your end-to-end deployment process that you would implement in this company.

Reminders:

- Use an InPrivate/Incognito window in your browser to avoid any confusion with any other credentials that you may use to access Azure resources.

The executive committee is hesitant to deploy new features and would like to be able to validate any changes on a small subset of customers before deployment into production.

Challenge

Your challenge is to implement a solution that enables the phased rollout (or percentage based, also sometimes referred to as canary) of a new version of the API of your choice.

If required, revise your deployment strategy relative to how you will perform the phased update of each API to its respective web application.

> **Note:** It may be a good idea to step back and evaluate your overall strategy at this point.

Update your existing pipeline to implement a mechanism to facilitate the gradual switching of your user base to the new version of the application while simultaneously monitoring the health of the new version of the API. The updated pipeline flow should include at least one manual approval and one fully automated step.

Your updated pipeline must include the ability to roll back to the previous version of an API if an issue is detected during phased rollout. If an issue is detected at ny point during your release automation, an issue relating to the cause must be created in your backlog.

Success Criteria

- Explain your updated strategy and the associated technical details thereof to your coach.
- Demonstrate that a change in the code of your chosen API is deployed with a phased rollout to your coach. Walk through each phase of your process and explain the state of your deployment.
- Demonstrate your ability to automatically rollback to the previous stable version and the creation of the associated issue in your backlog.

Stretch goal

If you have implemented challenge 6, automate your rollback based on monitoring the environment.

Reference

- Topical References
  - Route traffic with Azure App Service and Deployment slots (https://docs.microsoft.com/azure/app-service/deploy-staging-slots#route-traffic)
  - Explore how to progressively expose your Azure DevOps extension releases in production to validate, before impacting all users

(https://docs.microsoft.com/azure/devops/migrate/phase-rollout-with-rings)

- Deployment considerations for DevOps (https://docs.microsoft.com/azure/architecture/framework/devops/deployment)
- Canary Deployments Using Azure Web App Deployment Slots (https://www.c-sharpcorner.com/blogs/doing-canary-deployments-using-azure-web-app-deployment-slots)

- GitHub

  - GitHub Action Rollback Release (https://github.com/marketplace/actions/rollback-release)
  - Context and expression syntax for GitHub Actions (https://help.github.com/actions/reference/context-and-expression-syntax-for-github-actions)

- Azure Pipelines

  - How to implement rollback strategies in Azure Pipelines (https://mattvsts.github.io/2019/07/07/how-to-implement-rollback-strategies-in-azure-pipelines/)