OPEN HACK ENVIRONMENT ≣   **OVERVIEW**   **OPEN HACK GUIDE**   **PROVIDE FEEDBACK**   **MESSAGES**

TEAM CHAT

| 1 | 2 | 3 | 4 | 5 |

☐ **Mark Complete**

Implement a Blue/Green deployment strategy

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.

Classic deployment mechanisms require downtime. And if something goes wrong, you need to rollback or otherwise remediate the broken deployment.

A blue/green deployment strategy involves having two environments and shifting traffic between them. Blue or green can be your current version with the other being the proposed next version. This allows you to 'instantaneously' shift between versions without downtime. In addition, if there is an issue with the new version, rolling back simply requires reverting traffic to the previous deployment.

In the previous challenges, your team implemented workflow(s) for continuous integration, testing, and deployment to a container registry. You also

verified that updated container image was deployed and running in each API's respective web application. In this challenge you'll extend that capability to perform a Blue/Green deployment into your existing Azure App Service.

> **Note:** Blue/Green deployment is different from flighting, canary testing and progressive exposure; those are covered in in another challenge.

Before swapping the Blue and Green environments, ensure that the destination environment is online and available. Two scripts (one in bash (https://github.com/Azure-Samples/openhack-devops-proctor/blob/master/resources/polling.sh) and the other in PowerShell (https://github.com/Azure-Samples/openhack-devops-proctor/blob/master/resources/polling.ps1)) can monitor the response code of your APIs during the deployment process.

> **Note:** Additional gates (e.g., manual approval) which may be desired before switching are excluded in this challenge but may be implemented in another challenge.

Challenge

Leveraging your selected orchestration tool, update your existing workflow(s) to automatically perform Blue/Green deployments into Azure.

Think about your strategy as a team before you start developing your solution.

Success Criteria

To successfully complete the challenge, implement Blue/Green deployments for each of the APIs. Your coach will give you a change to push through your pipeline to test your strategy.

- Demonstrate that a code change in an API is deployed initially to a staging environment and automatically to production with "zero downtime."
- Explain to your coach the logic that you are using for blue/green deployment and address the following points:
  - Describe the mechanism that validates that the container and web app is ready
  - Demonstrate that you are able to perform a rollback to the previous version
  - Demonstrate what triggers or constraints are used to determine when/whether it is safe to swap to the new version
  - Demonstrate how you manage the version of each container image that is deployed
  - Demonstrate the mechanism for determining which version of your code should be in production (blue) and which should be in staging (green)

- Demonstrate which resource optimizations are performed once the deployment is complete > **Note:** this could be any optimization, including but not limited to restricting traffic flow to non-production slots or shutting the slot down altogether.
- Demonstrate how configuration secrets are being managed

References

- Blue/Green deployment (https://martinfowler.com/bliki/BlueGreenDeployment.html)
- Staging environments in Azure App Service (https://docs.microsoft.com/azure/app-service/deploy-staging-slots)
- Azure Slot Deployment with Blue-Green Deployment Model (https://medium.com/@sangeetavsunchu/azure-slot-deployment-with-blue-green-deployment-model-b52cd5ffaf07)

OpenHack © Microsoft 2019. All Rights Reserved - Powered By Opsgility Cloud Sandbox.

Privacy.