Deployment and Operations Guide (Runbook)

Form Scriber DevSecOps

Version 3.0

Ivy Pham

Vincent Leung

SWEN 670 Software Engineering Project

University of Maryland Global Campus

March 30, 2021

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 02/02/2021 | 1.0 | Initial draft |
| 02/16/2021 | 1.0 | Added section 5 |
| 02/17/2021 | 1.1 | Added sections 1 and 2 |
| 02/18/2021 | 1.2 | Added section 3 |
| 02/19/2021 | 1.3 | Added section 4 |
| 02/23/2021 | 1.4 | Added/Edited various sections |
| 03/15/2021 | 2.0 | Modified for Milestone 3 |
| 03/30/2021 | 3.0 | Final Review |

Contents

[1. Introduction 4](#_Toc65142505)

[1.1 Overview 4](#_Toc65142506)

[1.2 System Components 4](#_Toc65142507)

[2. Infrastructure 4](#_Toc65142508)

[2.1 Git 4](#_Toc65142509)

[Client Installation 4](#_Toc65142510)

[Initial Configuration 5](#_Toc65142511)

[Common Git Tasks 5](#_Toc65142512)

[2.2 GitHub 6](#_Toc65142513)

[Branch Overview 6](#_Toc65142514)

[GitHub Initial Configuration 7](#_Toc65142515)

[Create development teams 7](#_Toc65142516)

[Create repositories 8](#_Toc65142517)

[Assign repository to development teams 8](#_Toc65142518)

[Create development branch and set the branching strategy 10](#_Toc65142519)

[Add development team member to repository 11](#_Toc65142520)

[Create pull request 11](#_Toc65142521)

[Merge pull request 11](#_Toc65142522)

[2.3 Microsoft Teams 12](#_Toc65142523)

[Create signup sheet for developer access to repositories 12](#_Toc65142524)

[Create DevSecOps guidelines for development teams 13](#_Toc65142525)

[3. Build 14](#_Toc65142526)

[3.1 Microsoft Azure DevOps 14](#_Toc65142527)

[Add Users 14](#_Toc65142528)

[Access the Capstone Azure DevOps Project 15](#_Toc65142529)

[Create a continuous integration pipeline 16](#_Toc65142530)

[3.2 Docker 17](#_Toc65142531)

[Getting Docker images and running them 17](#_Toc65142532)

[Interacting with Docker containers 18](#_Toc65142533)

[Docker network 18](#_Toc65142534)

[Docker Compose file (running multiple containers with different configurations) 19](#_Toc65142535)

[Dockerfile (creating a Docker image) 19](#_Toc65142536)

[4. Deploy 19](#_Toc65142537)

[4.1 Docker and Azure Kubernettes Service (AKS) 19](#_Toc65142538)

[Deploy the AKS cluster 19](#_Toc65142539)

[Configure AKS environment on Azure DevOps 21](#_Toc65142540)

[4.2 Microsoft Azure DevOps 22](#_Toc65142541)

[Create a continuous deployment pipeline 22](#_Toc65142542)

[4.3 Useful Kubernetes Tasks 23](#_Toc65142543)

[Deploying an application: 23](#_Toc65142544)

[Exposing the deployment for services: 23](#_Toc65142545)

[Seeing all the happenings in log: 23](#_Toc65142546)

[Editing configuration of deployment: 24](#_Toc65142547)

[Getting terminal of a pod: 24](#_Toc65142548)

[Deleting deployments: 24](#_Toc65142549)

[Run a yaml config file to do deployments: 24](#_Toc65142550)

[Check components that can't be put in a namespace: 24](#_Toc65142551)

[5. License 24](#_Toc65142552)

# 1. Introduction

## 1.1 Overview

This deployment and operations guide shall detail the setup and configuration of the tools utilized for development teams, DSO team, and the CI/CD pipeline for Form Scriber, as listed in section 1.3. The tools represent widely-accepted implementations of various technologies involved in the creation of a DevSecOps development environment and are designed to provide services and compatibility with Form Scriber development teams.

## 1.2 System Components

The following lists the tools utilized by the DSO team for Form Scriber.

* GitHub
* Microsoft Teams
* Microsoft Azure
* Microsoft Azure DevOps
* Docker
* Kubernetes

# 2. Infrastructure

This section involves the tools required of development and DSO teams to align with expectations of Agile development collaboration, DevSecOps practices, and the CI/CD pipeline.

## 2.1 Git

### Client Installation

You can install any Git client as you prefer. The instruction below is for the client that is commonly used.

* Windows

Option A: Install/enable Windows Subsystem for Linux (Recommended).

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Since WSL is Ubuntu based, please run the following command in the shell to install Git client:

**sudo apt-get install git**

Option B: Install Git for Windows.

<https://gitforwindows.org/>

Download the EXE file and execute it.

* Linux

Install on Ubuntu/Mint.

**sudo apt-get install git**

Install on CentOS/RHEL.

**sudo yum install git**

* MacOS

Please follow the "Install on macOS" section.

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

### Initial Configuration

* Verify your Git client installation.

**git --version**

* Set your Git global configuration.

**git config --global user.name "<name>"**

**git config --global user.email "<email>"**

### Common Git Tasks

Please see this link **(**[**Git Pro**](https://teams.microsoft.com/l/file/3837923D-FEED-4B8C-B595-49660B141022?tenantId=d7a68d3a-81dd-434d-bc2b-786d33959b55&fileType=pdf&objectUrl=https%3A%2F%2Fumgcdev361.sharepoint.com%2Fsites%2FSWEN670SPRING2021%2FShared%20Documents%2FGeneral%2FPRO%20GIT.pdf&baseUrl=https%3A%2F%2Fumgcdev361.sharepoint.com%2Fsites%2FSWEN670SPRING2021&serviceName=teams&threadId=19:22a02f1b55724d98ab7cf0122b7a1559@thread.tacv2&groupId=a3582484-62a9-4f2b-baa0-0d7253188601)**)** to get familiar with Git.

* Clone the remote repository to local desktop/repository.

**git clone <repository url>**

* Check out the development branch.

**git checkout <branch name>**

**git pull origin develop**

* Create a new feature branch.

**git checkout -b <feature name>**

* Update, add, commit and push changes.

**git status**

**git add -A**

**git commit -m "<description of the change>"**

**git push -u origin <feature name>**

## 2.2 GitHub

### Branch Overview

Diagram

Description automatically generated

Figure 1. Branch Overview

* The master branch is protected, which requires the approval of the DSO team for the merge from the development branch in order to check for code quality and security. Direct merge to master branch from the feature branch is not prohibited.
* The development branch is restricted to the lead developer from each team to force a code review before merging to the development branch.
* The feature branch allows developers to create a new branch for each feature being worked on.

### GitHub Initial Configuration

Account can be created at <https://github.com>. If you would like to use GitHub with SSH keys, please follow [Connecting to GitHub with SSH](https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh). To get the respective HTTPS or SSH clone methods, go to the repository and click on **Code.**

### Create development teams

1. Login to UMGC SWEN Capstone Course admin GitHub account using provided credentials.
2. Go to **Profile →** **Your organizations →** **umgc** → **Teams** → <team name>.
3. Enter **Team name** → **Description** →select semester under **Parent team** → **Create team.**
4. Teams (4) are created for this project.

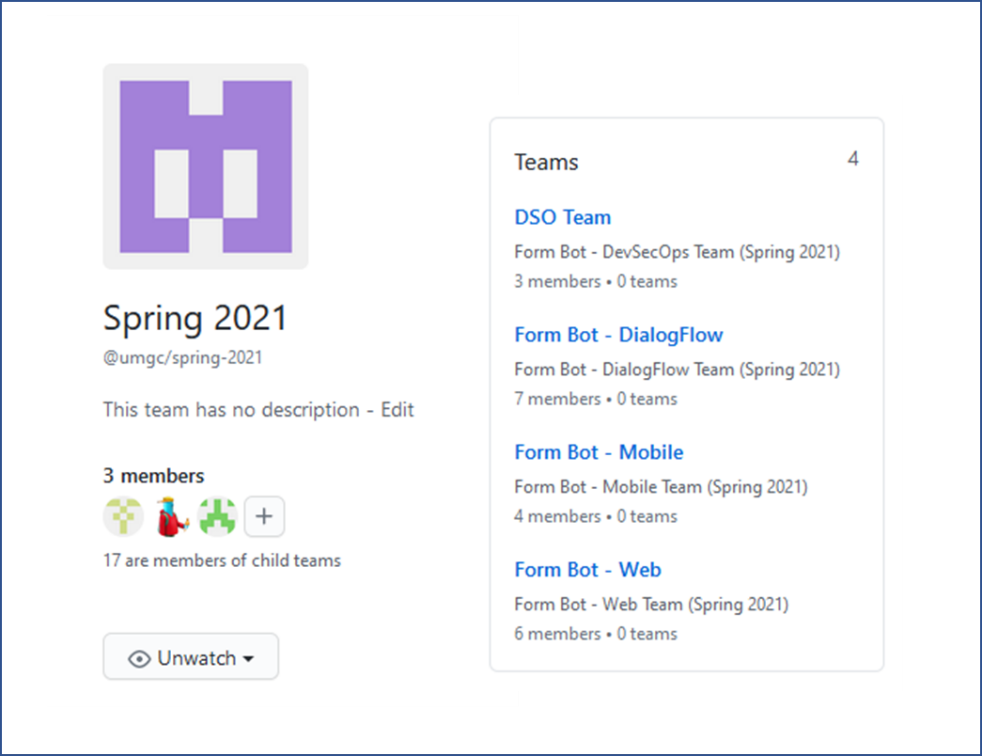


Figure 2. GitHub Teams

### Create repositories

1. Login to UMGC SWEN Capstone Course admin GitHub account using provided credentials
2. Go to **Profile** → **Your organizations** → **umgc** → **New**
3. Enter **Repository name** using the naming convention **umgc.<projectname>.<teamname>** → choose **Public** → **Create repository**
4. Development repositories (4) are created for this project.

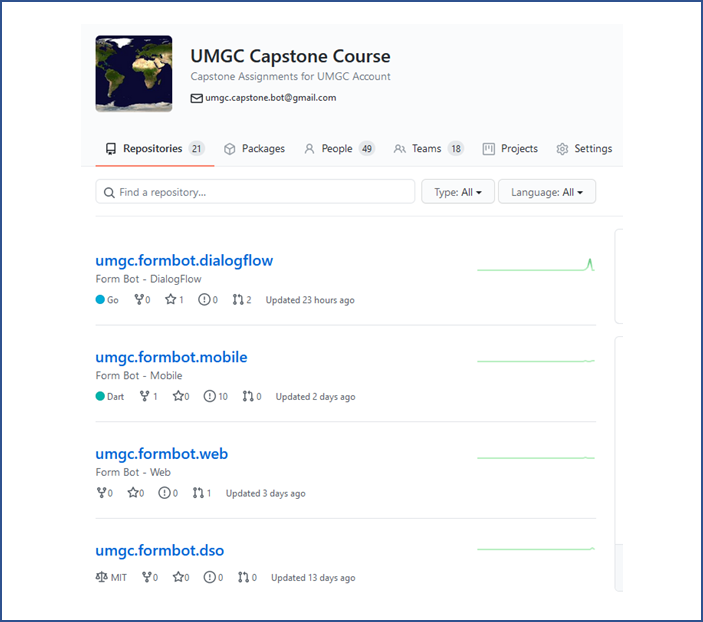
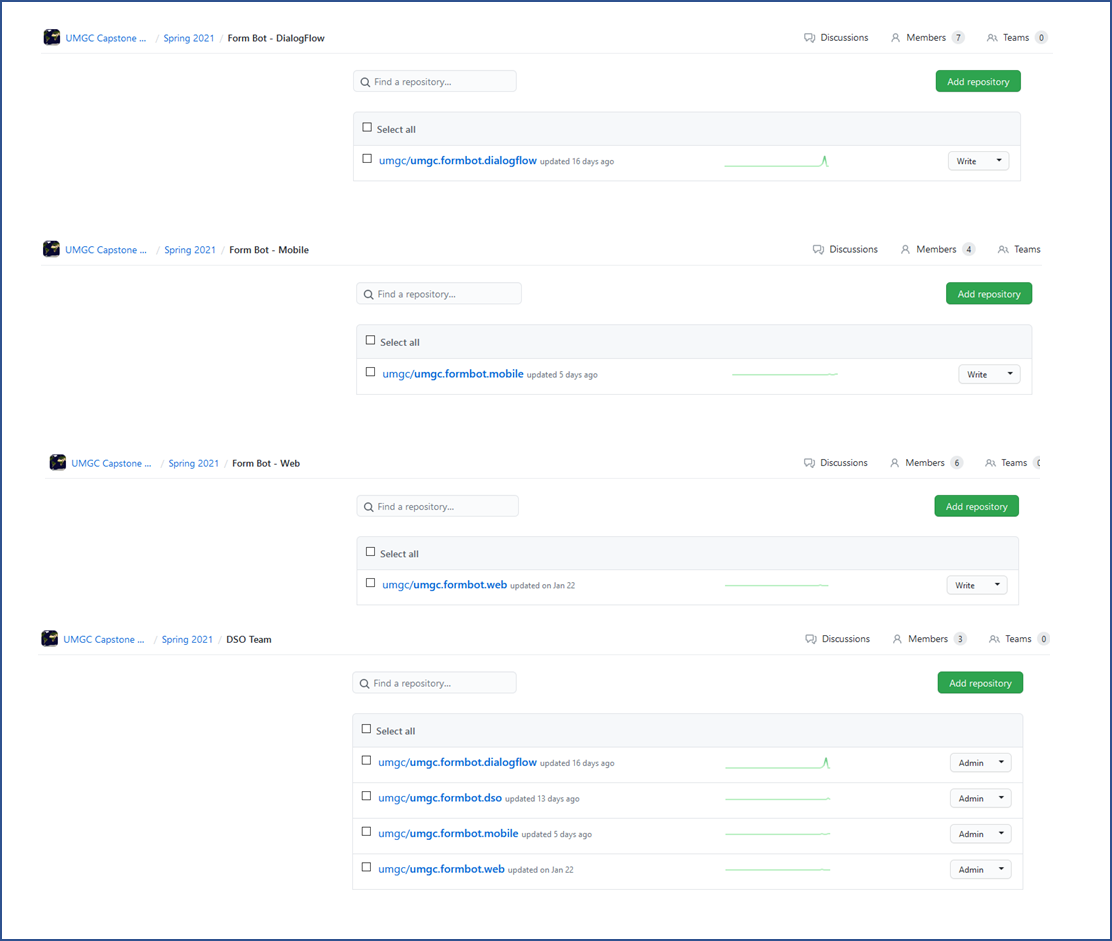


Figure 3. GitHub Repository

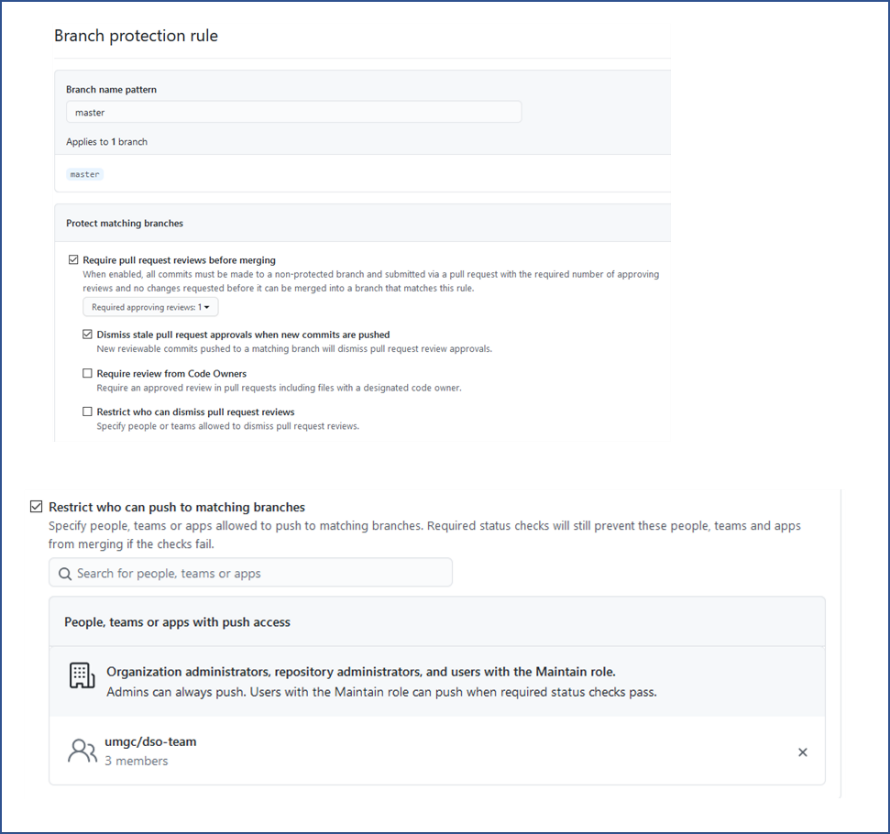
### Assign repository to development teams

1. Login to UMGC SWEN Capstone Course admin GitHub account using provided credentials.
2. Go to **Profile** → **Your organizations** → **umgc** → **Teams →** select semester → select team.
3. Click **Repositories** → **Add** **repository** → enter repository name → **Add repository to team.**
4. Change the access level for the repository via the associated dropdown menu
   1. If for DSO team, set the access level to **Admin**
   2. If for development team, set the access level to **Write**

  
  
Figure 4. Repository Access Level

### Create development branch and set the branching strategy

1. Login to UMGC SWEN Capstone Course admin GitHub account using provided credentials.
2. Browse to repository.
3. Click **master** branch → enter development branch name → click **Create branch: <**name**> from ‘master’.**
4. Click **Settings** → **Branches**
5. Click **Edit** for the master branch → enable the following:  
   **Require pull request reviews before merging**   
   **Dismiss stale pull request approvals when new commits are pushed**  
   **Restrict who can push to matching branches →** add DSO team → **Save** **changes**
6. Go back to **Branches** option → click **Edit** for the development branch → enable the following:  
   **Restrict who can push to matching branches** → add DSO team and the development team member designated as the DSO point of contact → **Save** **changes**

  
Figure 5. Branch Restriction

### Add development team member to repository

1. Login to UMGC SWEN Capstone Course admin GitHub account using provided credentials
2. Go to **Profile** > **Your organizations** > **umgc** > **Teams >** select semester > select team
3. Click the **+** icon > search by username or email > **Invite** > **Add <name> to <team name>**

### Create pull request

1. Login to GitHub website and navigate to the repository.
2. Select **Pull request → New pull request →** Select <branch> → **Create pull request**

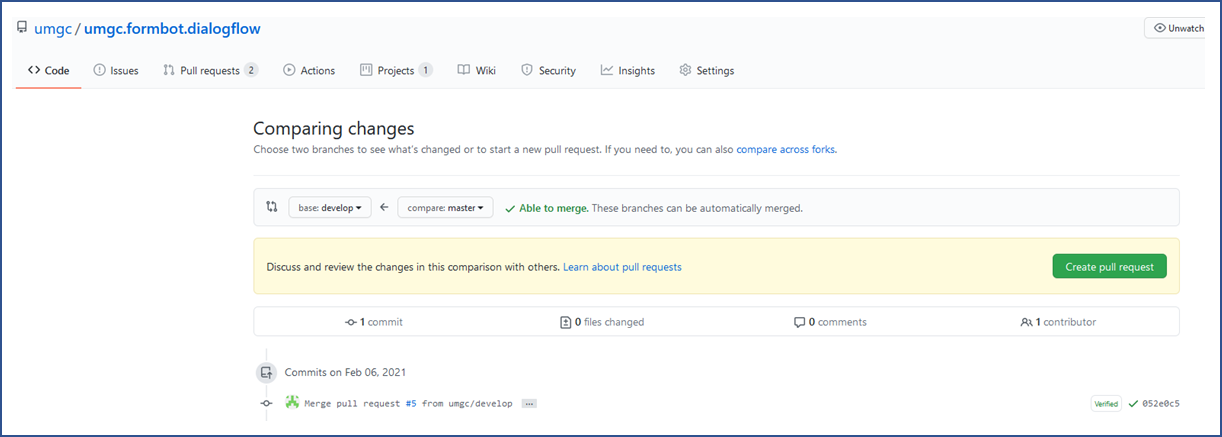


Figure 6. Pull Request

### Merge pull request

1. Login to GitHub website and navigate to the repository.
2. Select **Pull request → New pull request →** Select the pull request.
3. Review the changes and **Merge pull request.**

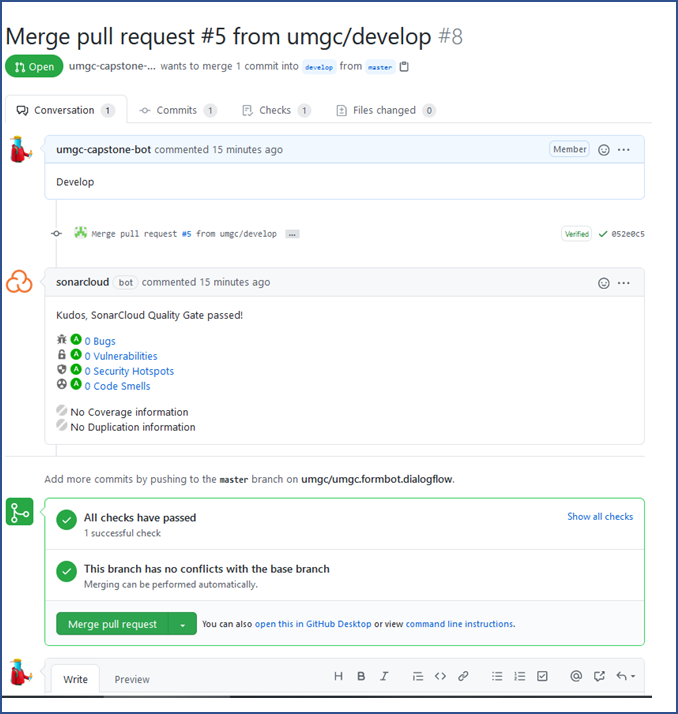


Figure 7. Approval and Merge Pull Request

## 2.3 Microsoft Teams

### Create signup sheet for developer access to repositories

1. Utilize the DevSecOps channel Wiki to create a table within a page

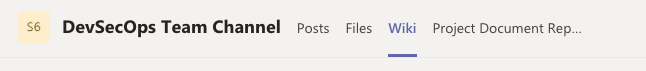
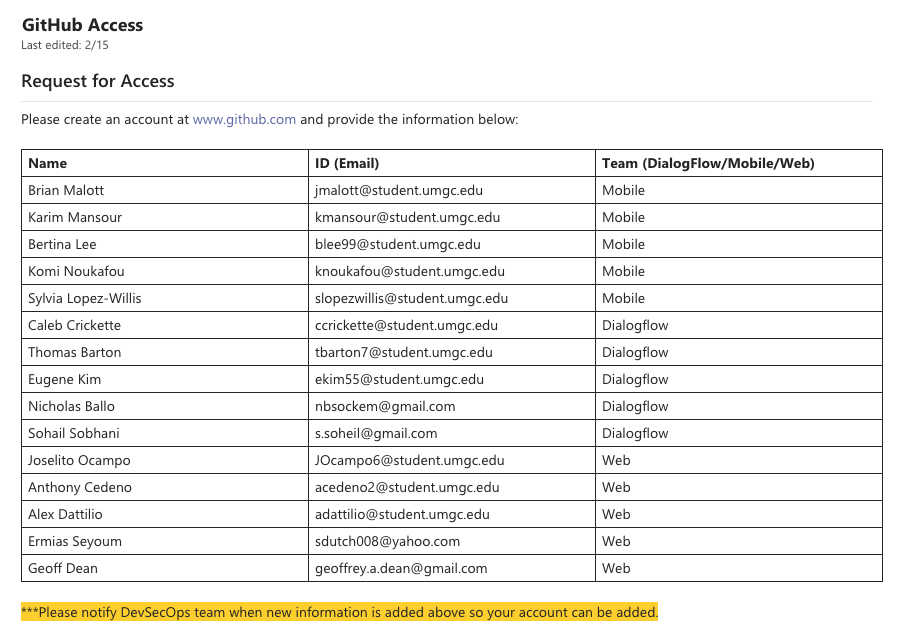
Figure 8. DevSecOps Wiki   
   


Figure 9. DevSecOps Wiki – GitHub Access



1. Notify development teams to provide their information to be added to their respective team/repository

### Create DevSecOps guidelines for development teams

1. Utilize the DevSecOps channel Wiki to create a page and sections for each programming language and add best practices

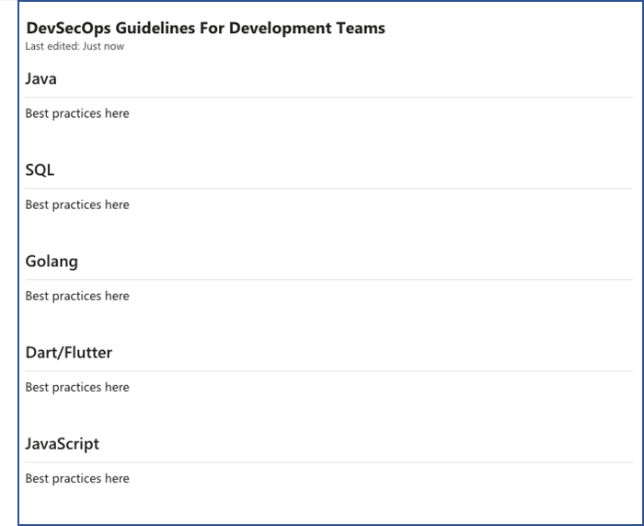


Figure 10. DevSecOps Wiki – Guidelines

1. Notify development teams to reference the guidelines during development activities.

# 3. Build

This section involves the setup and configuration of the tools required of the DSO team to create and manage the build components of the CI/CD pipeline.

## 3.1 Microsoft Azure DevOps

### Add Users

1. Go to <https://azure.microsoft.com/en-us/services/devops/> and log in with the admin UMGC SWEN Capstone account
2. Under Azure DevOps Organizations, click **dev.azure.com/umgccapstone**
3. Click **Organization settings**
4. Click **Users → Add users →** enter email of DSO team member **→** choose **Basic** Accesslevel **→** under **Add to projects** choose **Capstone → Add**

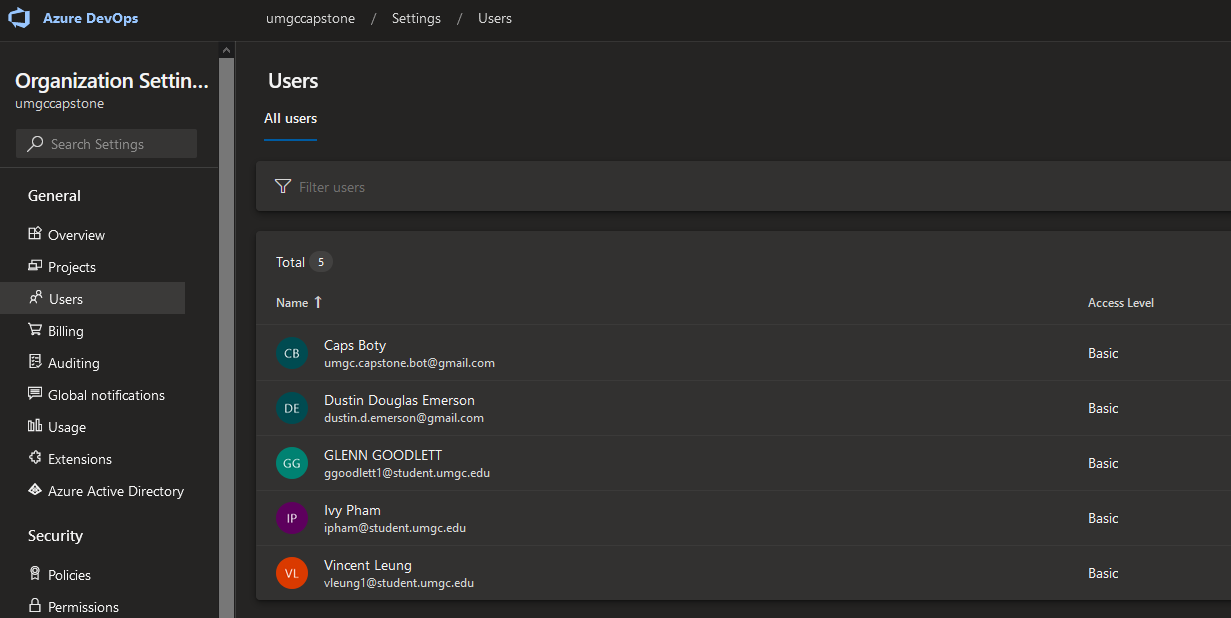
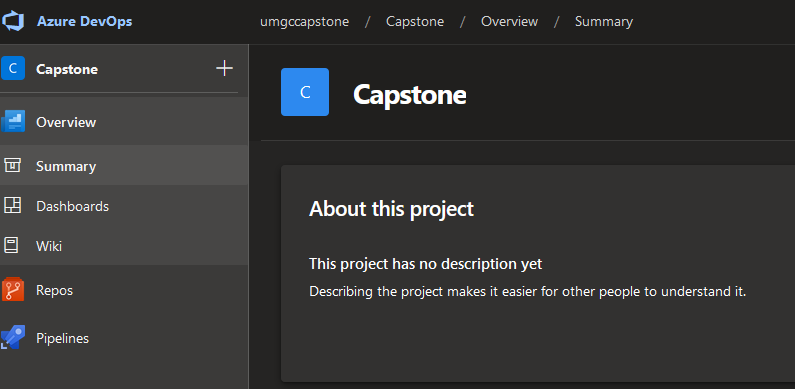


Figure 11. Azure DevOps – Add Users

### Access the Capstone Azure DevOps Project

1. Go to <https://azure.microsoft.com/en-us/services/devops/> and log in with your DSO team member account
2. Under Azure DevOps Organizations, expand **dev.azure.com/umgccapstone**
3. Under Projects, Click **Capstone**

Figure 12. Azure DevOps – Capstone Project



### Create a continuous integration pipeline

1. In the right-side menu, click **Pipelines → Pipelines → New pipeline**
2. Click **GitHub**
3. Select **All repositories** from the search criteria dropdown
4. Search for the repository for which to create a pipeline and select it (enter admin account password if prompted)
5. Approval the access to GitHub repository for Azure DevOps

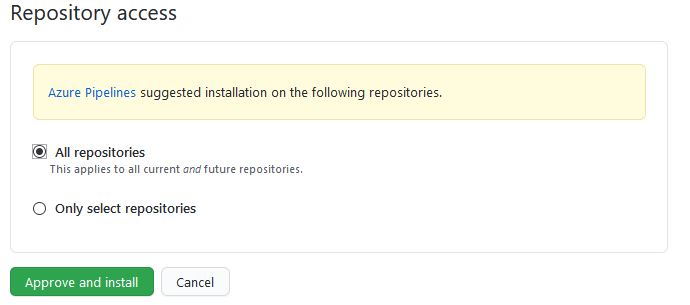


Figure 13. GitHub – Approve Azure DevOps Access

1. Create the pipeline based on the programming language or services being used

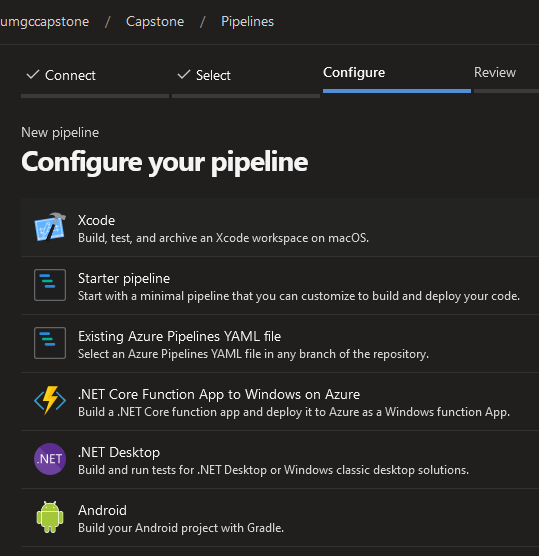


Figure 14. Azure DevOps – Create Pipeline

1. Add or edit the initial pipeline YAML template
2. **Save and run**

## 3.2 Docker

### Getting Docker images and running them

* Check installed docker images

**docker images**

* Pull a docker image to your machine (from Docker Hub, if another repo, use full URL)

**docker pull <image>**

* Download AND run docker image

**docker run <image>**

* Get a specific version of docker

**docker pull <image>:<version>**

* If running more than one version of the same image and they use the same port, need to bind your host port to the container's port:

**docker run -p<hostport>:<imageport> <image>**

* Run docker image in detached mode, so you can continue using terminal

**docker run -d**

* Delete an image

**docker rmi <image id>**

### Interacting with Docker containers

* Check running docker containers

**docker ps**

* Check both running and stopped images

**docker ps -a**

* Stop docker image

**docker stop <container id>**

* Start docker image again

**docker start <container id>**

* Delete container

**docker rm <container id>**

### Docker network

* List networks

**docker network ls**

* Create docker network (for images to communicate)

**docker network create <network name>**

* Tell container to run on a specific docker network

**docker run <any other options> --net <network name>**

### Docker Compose file (running multiple containers with different configurations)

* Reference the following for examples: <https://docs.docker.com/compose/>
* Start images using Compose configurations

**docker-compose -f <yaml file name> up**

* Shut down images using Compose

**docker-compose -f <yaml file name> down**

### Dockerfile (creating a Docker image)

* Reference the following for examples: <https://docs.docker.com/engine/reference/builder/>
* Create docker image

**docker build -t <name of your app>:<tag name, such as 1.0 or version-1, etc> <location of Dockerfile, such as root directory: .>**

* If you modify the Dockerfile, must rebuild image

# 4. Deploy

This section involves the setup and configuration of the tools required of the DSO team to create and manage the deployment components of the CI/CD pipeline.

## 4.1 Docker and Azure Kubernettes Service (AKS)

### Deploy the AKS cluster

1. Login to Azure Portal (<https://portal.azure.com>)
2. Select **Create a resource**
3. Enter **Kubernetes Service** in the search box and select the result
4. Select **Create**

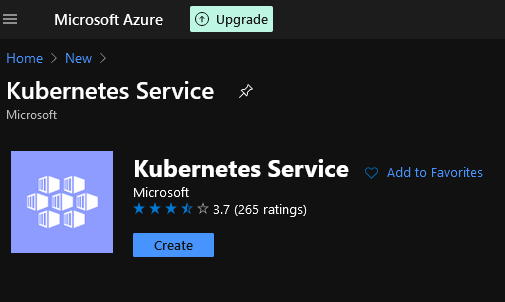


Figure 15. MS Azure – Add AKS

1. Insert appropriate information and select **Review + Create**

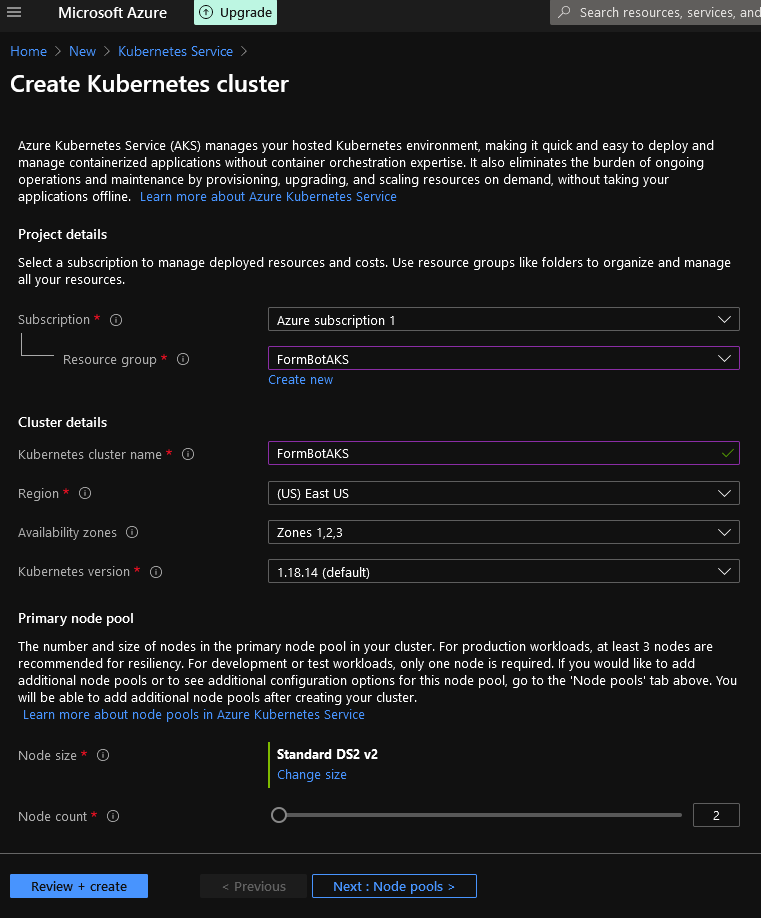
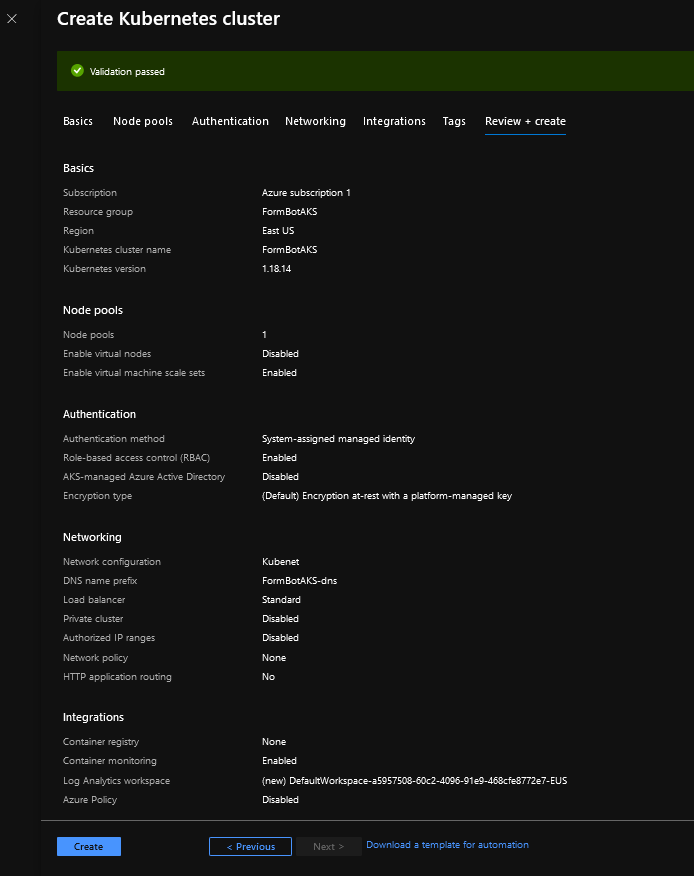


Figure 16. MS Azure – Create AKS Cluster

1. Accept all defaults and select **Create**

  
Figure 17. MS Azure – AKS Default Settings

### Configure AKS environment on Azure DevOps

1. Select **Environment** in the Azure DevOps Capstone project
2. Select **Create environment**
3. Insert the environment name and select **Kubernetes**

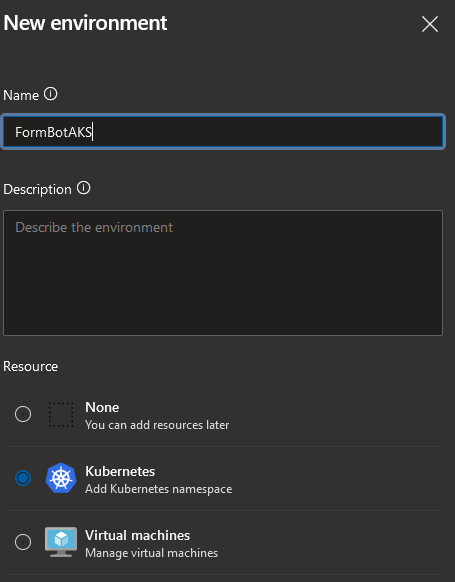


Figure 18. Azure DevOps – AKS Integration

1. Connect to the existing AKS Cluster and create

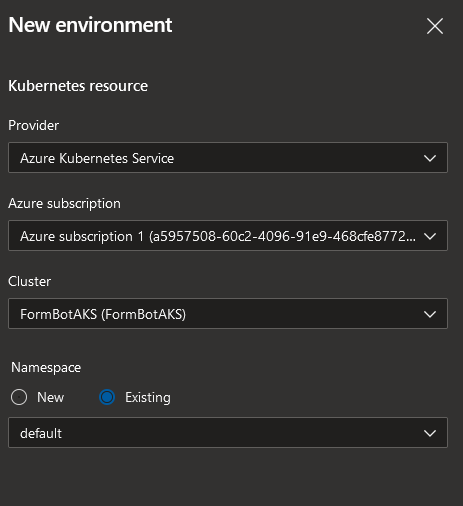


Figure 19. Azure DevOps – AKS Cluster

## 4.2 Microsoft Azure DevOps

### Create a continuous deployment pipeline

1. In the right-side menu, click **Pipelines → Releases → New pipeline**
2. Select **Deploy to a Kubernetes cluster → Apply**
3. Addan **Artifact →** select **Build** from Source type **→** select build pipeline **→ Add**
4. Click the lightning bolt icon → select **Enable** for Continuous deployment trigger
5. Edit and/or add Stages as needed
6. Add **Pre-deployment conditions** and **Post-deployment** **conditions** for stagesif needed
7. Click **Tasks** and edit kubectl commands
8. To save changes, click **Save**
9. To create a release, click **Create** **release** → **Create**

## 4.3 Useful Kubernetes Tasks

### Deploying an application:

**kubectl create deployment <deployment name> --image=<image location url>:<version>**

### Exposing the deployment for services:

**kubectl expose deployment <deployment name> --type=LoadBalancer --port=<port number>**

### Seeing all the happenings in log:

**kubectl get events**

**kubectl get pods**

**kubectl get deployment**

**kubectl get replicaset**

**kubectl get service**

**kubectl describe pod <pod name>**

* + - Additional logging info

### Editing configuration of deployment:

**kubectl edit deployment <deployment name>**

* Scroll down to configuration and file image and edit the image version for example, then save

### Getting terminal of a pod:

**kubectl exec -it <pod name> -- bin/bash**

### Deleting deployments:

**kubectl delete deployment <deployment name>**

### Run a yaml config file to do deployments:

**kubectl apply -f <file name>**

* Can make changes to this yaml and run again to apply changes

### Check components that can't be put in a namespace:

**kubectl api-resources --namespaced=false**

# 5. License

This software is free to use by anyone. It comes with no warranties and is provided solely "AS-IS". It may contain significant bugs, or may not even perform the intended tasks, or fail to be fit for any purpose. University of Maryland is not responsible for any shortcomings and the user is solely responsible for the use.