School of Information Technology & Business
Bachelor of Information Technology
CS7504 – Security for DevOps
Assignment 2

Due Date:	**3rd June 2022 (11:55pm)**
Total marks:	**100**
Weighting:	**30% of final mark**

Assessment Guidelines

- You are required to submit an electronic copy of your ***short video, source code and assignment documentation.***
- Create a project folder called **CS7504_Assignment2_FirstName_LastName**, where you will place all your files and code. Remember to zip this folder before submitting it to the Assignment 1 submission box in the CS7504 Moodle course page.
- Submit your work on or before the due date. The submission box will be closed and will no longer be accessible after the deadline.

Late Work & Extension

- Handing in work after the due date without extension is considered **late**. Students will be advised of any penalties to be imposed for a late submission.
- Lecturers reserve the right not to mark work that is handed in late.
- Students are encouraged to manage their schedules to meet deadlines. Under exceptional circumstances, (e.g. bereavement, illness, accident) students may be granted an extension of the due date of Assignment 2.
- Extensions are **not** granted automatically. A request for an extension must be made at least 24 hours before the deadline of Assignment 2. If a student applies for an extension, relevant evidence (e.g. medical certificate) must be presented.

Pass Requirements

- All Assignments must be attempted.
- Achieve an average of 50% or above over all assessments.

Assignment 2 Objectives:

This assignment aims to allow the learners to:

- Evaluate the appropriate automation strategy for a selected scenario
- Implement a properly designed and configured Continuous Integration and Continuous Deployment (CI/CD) pipeline.
- Develop a simple software that utilises the DevOps methodology and practices
- Implement a security feature in the DevOps process

Task 1: Developing a simple software, implementing, and configuring the CI/CD Pipeline with an appropriate automation strategy for the Docker and Kubernetes scenario.

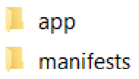
1. Using Azure DevOps, Source Code Management System and Azure Services, you should be able to deploy the .NET application on the Azure Kubernetes Services (AKS) through Azure CI/CD.

   a. You may use the same C# (or any .NET) software application from Assignment 1 or create a new .NET Web application.

   b. Ensure that you have already configured the required tools with Git/GitHub and Azure Repos before starting the software development.

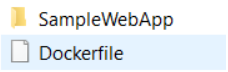
   c. ***You may refer to the Lab Activities related to the Docker and Kubernetes topics.***

   d. Your main folder should have 2 subfolders – app and manifests.



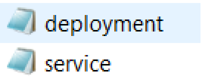
app
manifests

   e. Your app subfolder should have the main application which you have developed and Dockerfile.



SampleWebApp
Dockerfile

   f. The manifests subfolder should have the appropriate .yaml files.

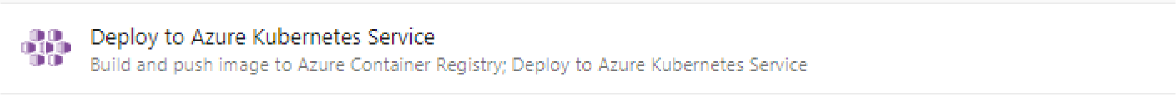


deployment
service

   g. You should be able to Push the changes to the Azure Repos or any remote repository of your choice.

   h. Ensure that you have correct services created in a Resource Group in Azure Portal to build and push the Image to the private registry (ACR) and to deploy the application to Kubernetes cluster.

   i. While configuring the pipeline in Azure DevOps, you may use the existing template in Azure DevOps. *(Hint: Deploy to Azure Kubernetes Service)*



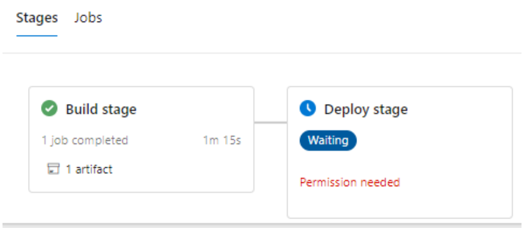
Deploy to Azure Kubernetes Service
Build and push image to Azure Container Registry; Deploy to Azure Kubernetes Service

   j. If any changes need to be made, you may use the Show Assistant to make changes to the YAML file.

   k. After the build, you should see the deploy stage getting triggered automatically. The development stage should be green indicating that Build is succeeded.

   l. You should be able to publish an artifact.

   m. The deploy stage should be blue and show that it's pending for approval.



Build stage
1 job completed 1m 15s
1 artifact

Deploy stage
Waiting
Permission needed

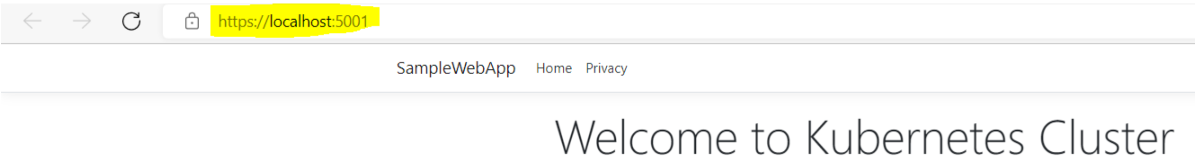
n. Using Azure Portal, you should integrate the kubectl using the CLI commands.

o. Provide evidence of the commands which were executed. *(Hint: use Azure PowerShell Command Shell instead of Bash shell for all the commands)*

p. You should be able to watch the status of the pods and services in Kubernetes cluster using the kubectl commands.

q. You should be able to view your Web Application using the External IP address that was generated by the Deploy stage and the kubectl services command.

r. You should be able to provide evidence when you execute the Web locally as shown below.

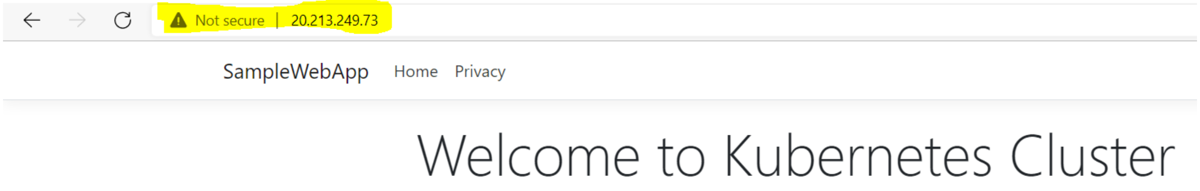


https://localhost:5001

SampleWebApp Home Privacy

Welcome to Kubernetes Cluster

s. Another evidence would be when the Web Application is deployed in the Kubernetes Cluster with the EXTERNAL IP address.



Not secure | 20.213.249.73

SampleWebApp Home Privacy

Welcome to Kubernetes Cluster

t. Perform the SCM collaboration practices and make changes in the code. Make a commit to the code on the local repository and pushes it to the remote repository or you can make a straight change in the remote file and Commit.

u. Committing the change in the .NET application should automatically trigger a build, publish an artifact and again ask for approval before deploying.

v. Evidence for the number of images getting created with a unique in the Azure Container Registries (ACR) repository.

w. Provide a **<u>short video</u>** showing that these practices were followed accordingly. ***(Please see the Assignment Requirements section for more details.)***

Task 2: Discussing and Implementing Security (Research Work)

1. DevSecOps
   a) Discuss a five-step process to successfully implement DevSecOps.
   b) DevSecOps talks about six best practices. Select one of the best practices and discuss how it can help your organization successfully integrate DevSecOps into its daily operations.
2. Hashicorp's Vault
   a) Managing Secrets and protecting sensitive data. Explain how HashiCorp Vault works.

(2000-3000 words)

Assignment Requirements

Below is your checklist to be completed before you submit your Assignment 2.

Deliverables	Completed
▪ Short Video with your commentary. - Your video should NOT be more than 30 minutes. - The video should only contain your voice. It is your choice if you wish to appear in the video. ▪ Source Code / YAML files / Docker file	
▪ A **project documentation** containing: - Description of the scenario where DevOps process can be implemented - Screenshots to be provided in the documentation wherever it is mentioned that evidence is required. - Constraints (Limitations / Errors) you came across and strategies applied to solve it.	
▪ DevSecOps documentation (2000 – 3000 words)	
Format ▪ You are required to present your assessment in a report format. Therefore, you will need to include a title page, an executive summary a table of contents, a conclusion and a reference list (use APA referencing).	

Assignment 2 Marking Scheme

Criteria	Marks
• **Part 1 – Designing CI/CD Pipeline**	**60**
– Software Application, Docker & YAML files.	3+3+4
– Configured the remote repo and link to repo	2
– Building pipeline in Azure DevOps using correct template or tasks.	5
– Successful Build and Deployment	8
– Evidence of image pushed to ACR	5
– Integration of kuebctl in Azure Portal Powershell.	5
– Evidence of the pods and service running from Azure Portal Powershell using the kubectl commands	5
– Evidence of making a commit – Executed an automatic trigger to the Build and publish of an artifact – Evidence of the deploy stage in blue before permitting the approval – Successful Deployment to the EXTERNAL IP address.	5
– Explanation of constraints (Limitations / Errors) you came across and strategies applied to solve it.	5
– Short Video	10
• **Part 2 DevSecOps**	35
– Discussion of the five-step process in DevSecOps	10
– Integration of one of the practices in DevSecOps	15
– Working of HashiCorp's Vault	10
• **Overall format (including references)**	5
Total	**100**