**Challenge- I**

We want to automate the deployment of mediawiki on docker/K8/OpenShift, we’d like to see code file.

We want to assess your learnability and your current knowledge of containerizing an application using an orchestration platform like K8/OpenShift.

Details

we’ll start with one instance of mediawiki running.

That’ll be backed by a database server (MySql/postgres) running on a separate container.

We expect this to be installed using these steps: [Installing MediaWiki](https://www.mediawiki.org/wiki/Manual:Installing_MediaWiki)

**Expectations:**

* Automated setup for the problem statement including the infrastructure setup using any IAC code (Terraform/Azure ARM Template/AWS Cloudformation/GCP) of your choice
* Adopt best practices in the tools which you are using. For example- Proper syntax and naming, Modular code etc.
* We expect your solution to be designed using orchestration tool of your choice e.g Openshift, Kubernetes (Use Any Hyperscaler PaaS)
* Brownie points if the application is running.

**Solution :**

To automate the deployment of MediaWiki on Docker/K8, we will use infrastructure-as-code (IAC) tool - Terraform.

I will be using Terraform code that will create an instance of MediaWiki running on Docker, backed by a MySQL database running on a separate container.

First, we need to create a Docker image for MediaWiki. We can use the official MediaWiki Docker image available on Docker Hub.

**The following Dockerfile will build the image:**

**FROM mediawiki:latest**

Next, we can create a Terraform configuration file that will create the necessary infrastructure:

****

**This Terraform configuration file will:**

* Use the Docker provider to create the necessary infrastructure.
* Build the MediaWiki Docker image using the Dockerfile.
* Create a MySQL container with a root password.
* Create a MediaWiki container linked to the MySQL container with the necessary environment variables.
* Expose port 8080 on the host machine to access MediaWiki.

**To deploy the infrastructure, run the following commands:**

**terraform init**

**terraform apply**

This will initialize the Terraform configuration and create the infrastructure. Once the infrastructure is created, we can access MediaWiki by navigating to http://localhost:8080 in our web browser.

**Challenge-ll**

Background: A k8s/Openshift cluster is shared among multiple teams to host their containerized applications and related components wherein each Team deploys their application components in their respective Namespace. Each Namespace comes with  set quotas for the total amount of memory and CPU that can be used by all Pods running in a [namespace](https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces).

**Situation:**

Referring Challenge I example, create 3 replica of Pod and name them as POD A, POD B and POD C

Write a code in YAML

1. To deal with the resource crunch issue and ensure that POD A consumes 400 MB, POD B 200 MB and POC C 400 MB memory, so that all the pods can get the appropriate resources.
2. MediaWiki Pod wants to access sample data stored in form of a .csv file ( You can create a sample CSV and put some max 2 rows and 2 column worth data), write a YAML script to make this .csv available to the running application pod. (We want to see where do you store the .csv file to make it accessible to the running application pod)

**Solution :**

To deal with the resource crunch issue and ensure that POD A consumes 400 MB, POD B 200 MB, and POC C 400 MB memory, we can use Kubernetes/Openshift resource quotas and limits, please find the YAML code to create three replicas of Pods named POD A, POD B, and POD C, with specific memory requirements, and to make a sample CSV file available to the MediaWiki Pod:



This YAML file will create three Pods, POD A, POD B, and POD C, with specific memory requirements. POD A will consume 400 MB, POD B will consume 200 MB, and POD C will consume 400 MB memory.

The YAML file also creates a ConfigMap called sample-data, which contains the sample CSV file. The file is stored in the data field of the ConfigMap.

Finally, the YAML file creates a MediaWiki Pod that uses the sample-data ConfigMap to make the sample CSV file available to the running application Pod. The sample CSV file is mounted to the /data directory inside the container using a volume mount.