In a rapidly evolving digital landscape, DevOps has become the driving force behind innovation and efficiency in the world of software development and IT operations.

However, as the DevOps field expands, professionals are finding it increasingly challenging to connect with like-minded individuals, share best practices, and keep up with the latest industry trends. This gap in communication and collaboration has sparked the creation of DevConnect, a revolutionary social web app designed to bring the global DevOps community together.

With a clear vision in mind, the founders assembled a dedicated team of DevOps engineers, software developers, and designers to bring DevConnect to life.

The founders chose you as a DevOps engineer for their biggest product The DevConnect web app.

You are an essential part of this ambitious project, entrusted with the responsibility to ensure the smooth deployment, scaling, and maintenance of the platform. Your expertise in automation, continuous integration, and continuous delivery is critical for the success of DevConnect.

Go "DevOps" your way through this project, following best practices and GCP technologies to create an outstanding platform that unites the global DevOps community.

**Dockerization**

* For the packages, use all the latest versions.
* Run the application and make sure everything is working.
* Create init.sh and delete.sh for automation.
* Run the application with volume and make it persistent - check by signing up to the app, delete the container and then start the container and log in without signing up.
* Create a new Private Github repository and push your code to it.

**Deployment**

* Create an artifact repository called <your-name>-artifacts

in the me-west1 region and automate a deployment of the web app image to it.

* Deploy a zonal GKE **standart** cluster called <your-name>-cluster with the following specifications:
* IF NOT MENTIONED LEAVE AT DEFAULT

**Zone:** me-west1-\*, us-west1-\*,northamerica-northeast-\*

**Node pool 1:**

1. name - devconnect-app
2. nodes - 1
3. machine type - e2-micro (2 vCPU, 1 core, 1 GB memory)
4. service account - assign DevOps-sa
5. boot disk - 10 gb.
6. node taints - NO\_EXECUTE, key=webapp, value=mywebapp.(read and understand taints)

* In a namespace called production, create 1 replica deployment to the app.
* Expose it using load balancer service and access it through a browser.
* Fix the bug and upload to the artifact repository a new version with the corrected bugfix.
* Rollout the new version deployment.

**CI/CD**

* Create a Compute engine instance with the following specs:

1. Name - <your-name>-jenkins.
2. Region - me-west1(Tel-Aviv)
3. Machine type - e2-medium (2 vCPU, 1 core, 4 GB memory)
4. service account - assign DevOps-sa
5. boot disk - 10 gb.
6. Automation - install docker engine.

* Create a new local repository called jenkins\_lab and use it to create an automation deployment from your local laptop that builds your jenkins image from freestyle project, uploads it to the artifact registry and runs it inside the compute engine instance, make sure to run with volume for persistence.

Make sure the container can use docker!

* Add to the jenkins image gcloud sdk installation and upload the new version.
* Access jenkins through the web and configure it(Install suggested plugins, create user, etc…)
* Create a CI/CD pipeline(jenkinsfile) that do the following:
* Build is triggered by checking if change(push) has been made every 10 seconds.
* Build the application
* Test - run django tests and check for 200(OK) response when trying to access the app.

If build succeeded:

* Push the image new version to artifact registry repository(The version must be the commit message)
* **BONUS:** Deploy the updated app to production cluster

If build failed:

* print “the pipeline failed :(“.
* Always Clean up all resources and workspace when you're done.
* Change Application environment to production push the new version, wait for CI to automate deployment and rollout the new version manually
* Sketch an architecture of the DevConnect project using google architecture tool