Final project

The Cloud Module 2022 final project will be in two phases:

1. Infrastructure
2. DevOps.

Completing this project should provide you with at least the fundamentals knowledge of the role of a real-life Cloud/ DevOps engineer. The completed project will also be a very good demonstration for any interviews or even a template for an application so it is extremely worth doing.

First phase - Infrastructure

This is similar to what you have done the last couple of weeks but wrapping it all together. This is an important role of a Cloud engineer. Orchestrating new/existing infrastructure, proactive monitoring, managing resources, disaster recovery etc.

Infrastructure resources:

- Turn what you currently have into 3-tier (frontend, backend or APIs, database) application;
- Working DB
    - Retrieve data from a DB Server of your choice (RDS or Mongo) - any type of data, we just want to see some data being retrieved by your application
- Make sure nothing is exposed to the internet unnecessarily by adding internal networking

Security:

- Assess how you can improve your current infrastructure by thinking about how it can be attacked.
- Implement changes or write down what needs to be done.

Monitoring:

- Blackbox monitoring
    - Is the Frontend alive? Is the backend alive? Is the database responding? Make sure alerts are generated
- TLS monitoring (check if the certificate is about to expire)
- Any other monitoring that you find suitable

Infrastructure as a Code

- All infrastructure should be on Terraform
- Use an S3 bucket to store your **versioned** Terraform state
    - Can also use S3 for storing static resources

Second phase - DevOps

This phase you will have touched on during the first 2 weeks. This is very important to understand for any DevOps, Cloud and even Developer roles these days. It is about automating and optimising the Software Development Lifecycle while at the same time maintaining quality and allowing a faster feedback loop.

Environments:

- Create a dev/test and production environment (tip: create one using code and then just easily replicate the second one)

CI/ CD:

- The pipeline should have the following workflow:

    1. Run tests (or at least a lint) on the JS/PHP/Python/ whatever languages you like
    2. Deploy to first environment
    3. Run Smoke tests
    4. Wait user approval
    5. Deploy to second environment
    6. Run Smoke Tests

Use whatever CI/CD tools you want. If not sure, we recommend using github action.

Final project deliverables (in 4 weeks - date TBC)

Presentation:

- Architectural diagrams:
    - Diagram for workflows or lifecycle
    - Diagram for infrastructure

- Business cases to justify why to implement phase 1 and phase 2 respectively
- Costs management
    - Estimated costs of resources
    - Cost management strategy

Demo:

- **Phase 1**: Launch whole infrastructure with Terraform
    - `terraform apply` to launch infrastructure or update infrastructure
    - `terraform apply` to a new cloud region

- **Phase 2:** A working CI/CD pipeline for multiple environments.
    - CI/CD for the lower environment.
    - Approval to promote build to an upper environment.

Documentation & instructions:

- Phase 1
- Phase 2