

# IaC Compulsory Assignment 2 Report

## Folder structure

For this assignment I choose the second folder structure. This is since I felt this gave the most efficient and concise way of describing our infrastructure. Here we only have one file describing the infrastructure, instead of multiple with individual setups for each file. This ensures consistency across the different environments as we are sure of that all the environments are the same as the environment described in the main branch. This works especially well with my implementation of a branch specific dev environment for each branch. This allows for an almost full implementation of our repository as the single source of truth as it is consistent across all of the different environments in all branches. This makes the infrastructure fully documented in all its stages, ensuring a understandable infrastructure in all its stages.

In addition, I also think that the implementation of different tfvars files is smart in this folder structure. I used this in a way where I allowed for having different resource allocation in the same resources across multiple environments. For instance I usually defaulted to having the dev environments use a cheaper sku, than the prod and staging environments. This is helpful saving on costs, where we most likely don't need a very resource heavy dev environment. I did not use these files for changing the names of resources to have different names in each environment as I thought this would lead to unclarities with inconsistent naming conventions.

There are of course some drawbacks to this folder structure. One of them is that it can be difficult to understand at first. Since the same file is describing all of the environments, it can be difficult to be able to abstract this file into three environments. This could also be problematic in a case where one wants, for whatever reason, to have a different setups for each environment, as this is something this folder structure does not easily allow. Lastly, my environment almost demands the workflows I have created to be manageable. Deploying this manually, would involve running a lot of commands to create and switch to the correct workspace and running terraform apply with the correct tfvars file, which could be time consuming.

## Difficulties

The main difficulty I had in this assignment was that I tried to implement a fully working infrastructure containing a web app that used all of the resources in the same way that it was described that they were to be used by OperaTerra in the task. This turned out to be rather difficult, and I therefore used a lot of time on this. In the end I did not manage to create a working solution for this as I am very restricted on how much time I can afford to use on this assignment.

Otherwise I also had some difficulties creating the workflows. Especially the branch related workflows, as this forced me to learn a lot about how one triggers a workflow to run at a specific time, and also how to find and use the branch name to create a workspace with the same name. I also used a lot of time figuring out the best way of using other workflows in a workflow. The final solution here was to make the terraform lint and tfsec workflows callable, and then call them from the other workflows. Then I could make the rest of the workflow depend on these calls, meaning that they only run after the other workflows have passed.