# DevOps Kitchen Workshop

## Deliver DevSecOps

### *#week_fifteen&sixteen - Terraform code scanning with checkov*

***duration: 2 weeks***

ForgTech company wanna test your ability to Deliver HCL code scanning using Checkov, This will help you to build a good reputation.

The FrogTech Security team requests you implement a scanning stage of the company main AWS Pipeline, with the criteria below:

1. The scan should match intended changes (*i.e. do not scan* `.tf` *files instead scan the plan result)*
2. The scan step action should run in parallel with the plan step.
3. The Checkov scan should run on JSON plan output, providing a human-readable format.
4. The Checkov scan should run before the manual approval stage.

As well as Build a personal document consisting of what you learn with deep details and resources *i.e. this will assist you in getting back and refreshing your knowledge later.*

By following The ForgTech deployment policy, you should deliver this deployment in an automated pipeline using AWS Pipeline and follow the DevOps Team pipeline structure standards as The Pipeline stages should be as follows:

1. Preparation stage: This includes the installation and preparation steps *(i.e. install AWS CLI, Terraform, Python, Checkov, and Terraform Initialization)*
2. Plan Stage: This includes the terraform validation and plan commands; The plan must done by using the output file *(i.e. tfplan file)*
3. Scan step action: This includes the Checkov scan, and should be run on the plan result, as well as in parallel with Plan stage.
4. Manual approval: Pause the pipeline until Reviewed & Approved by the checker engineer.
5. Terraform apply: This includes Starting provision resources.

Consider the below requirements specifications.

1. Resources must be created at the us-east-1 region.
2. Store state file backend into HCP | S3.
3. Resources must have common tags combination as below:
4. Common tags:
   a. Key: “Environment”, Value: “terraformChamps”
   b. Key: “Owner”, Value: <“Your_first_name“>

**Bonus**

1. Build an Architecture diagram of the Pipeline stages.
2. Append an extra step command of the Plan stage reviewing specific parts of attributes using `jq` command.
3. Crafting multiple HCP workspaces.

## References:

- AWS CodePipeline Automate IaC provisioning
- AWS CodePipeline - Deliver DevSecOps
- What is checkov
- Custom policy overview
- Terraform plan scanning
- eraki code sample