

## Terraform Assignment

### Learning Outcomes:

This assignment focuses on bringing IaC concepts together in a business solution via a mini-network. Best practice is required throughout. A class period will be allocated to initial discussion on the assignment to enable team work in teasing out the problem(s) at hand. The assignment due date is as shown in Blackboard.

LO's covered by this assignment are:

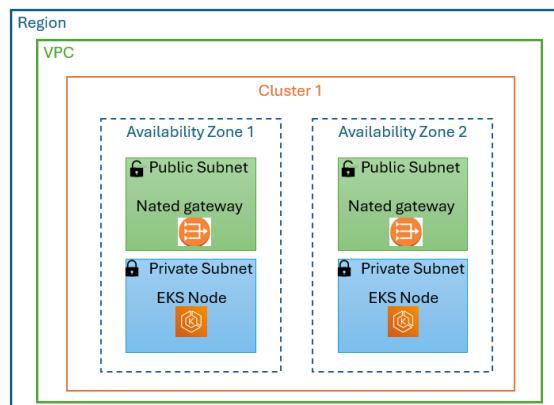
1. Engage in practitioner-based research to compose solutions to deployment pipeline efficiency issues
2. Conceptualise, research and critique techniques and technologies to automate processes and job lifecycles
3. Undertake an analysis of incomplete, incorrect or contradictory solutions to a problem and communicate in a professional manner an alternative approach underpinned by justifications informed by clearly articulated research.
4. Critical awareness of the importance of coding best practice in disseminating information to enhance teamwork.
5. Analyse the impact of scripted pipelines on enhancing team work in delivering software across a pipeline.
6. Devise a scripted solution for a software delivery pipeline including design, implementation and test phases.
7. Review and refactor scripts via peer-review retrospective and refinement meetings.

Create a project using terraform for the following to host an application across clusters in two availability zones. You may use an architecture of your choice but you should explain your decisions as you go.

Make sure to host your solution on github making the repository publicly available. Ensure that you show multiple prs to the repo to show continuous improvement.

A single cluster example architecture might look like this, but don't forget to consider the load balancer, networking, etc.

A multi-cluster architecture is different. A single cluster is required as a minimal solution.



Use a scenario to help you frame the example. You might host a website on a node and replicate it across your cluster(s) for high availability. Keep the example simple.

- a) A VPC – Private cloud – similar to the Kay McNulty Data Centre resources. Some of you may have recently reviewed with Danny.
- b) Use cidr block 10.0.0.0/16
- c) Region(s): (as preferred)
- d) Availability zones: 2
- e) Public Subnets: 2
  - a. 10.0.1.0/24
  - b. 10.0.2.0/24
- f) Private Subnets: 2
  - a. 10.0.10.0/24
  - b. 10.0.20.0/24
- g) Example - Web servers: 2
  - a. 1 per availability zone.
  - b. Install nginx, apache httpd or other server on each
  - c. Webservers to be added to the public subnet
- h) ALB
  - a. Application load balancer
- i) Networking
  - a. Routing tables

This part is tricky so take your time!
  - b. Consider the following pseudocode. Is it good or weak?

```
Resource "aws_route_table" "ws_route_table" {
  Route { cidr_block = "0.0.0.0/24"
    Gateway_id = aws_internet_gateway.internet_gateway.id
  }
}
```

A key part of this assignment is the application of best practice so take your time.
  - c. Internet Gateway
  - d. Consider:

```
vpc_id = aws_vpc.startup_vpc.id
```
- j) Security Groups
  - a. Firewalls for both inbound and outbound traffic. Consider ports and protocols.

#### Final submission:

- Code, with comments justifying decisions, through a link to your public github repo
- Your lab write up –
  - conclusions.
  - Max. 5 images of the final running network with your name or lnumber showing clearly in the running code



© R.G. Lennon

### Note:

Time will be given in class for group discussion to aid progress but an individual solution and individual submissions are required. The discussion in groups will aid the experience of team work on solutions but without individual development.

The rubric will be available on blackboard so follow it carefully to spend your time where you can gain the most marks.