

MSc Computer Network Administration and Management
Final Year Project Presentation

Proposed Virtual Private Cloud (VPC) for Pallisa Hospital, implemented using Terraform as an Infrastructure as code (IaC) tool in Amazon Web Services (AWS)

By	: Otilia Linore Kawuma
Module Codes	: M32616-2023/24-PAYEAR
Supervised by	: Dr. Mani Ghahremani
Date	: 16 th /September/2024

Project objectives

- The goal is to design a VPC cloud infrastructure in AWS cloud.
- Write Terraform code to provision a robust, secure, and scalable VPC environment.
- The VPC should enhance data management, optimize resource utilisation and strengthen security within the Hospital.

Existing Infrastructure System challenges

- Pallisa experiences uncontrolled and chaotic growth of data due to manual legacy systems
- Delays in resource, data and information access.
- Vulnerabilities in data access and Information management.
- Lack of remote access to Hospital computers

Project contributions and value added

➤ Operational excellence

Easy to manage and monitor compared to console and on-prem infrastructure due to availability of a **state file**.

➤ Security

The VPC is a logically isolated network that further segregates resources in subnets to avoid public access, use of NACLs, **Security groups**, **IAM roles to run specific tasks**.

➤ Reliability

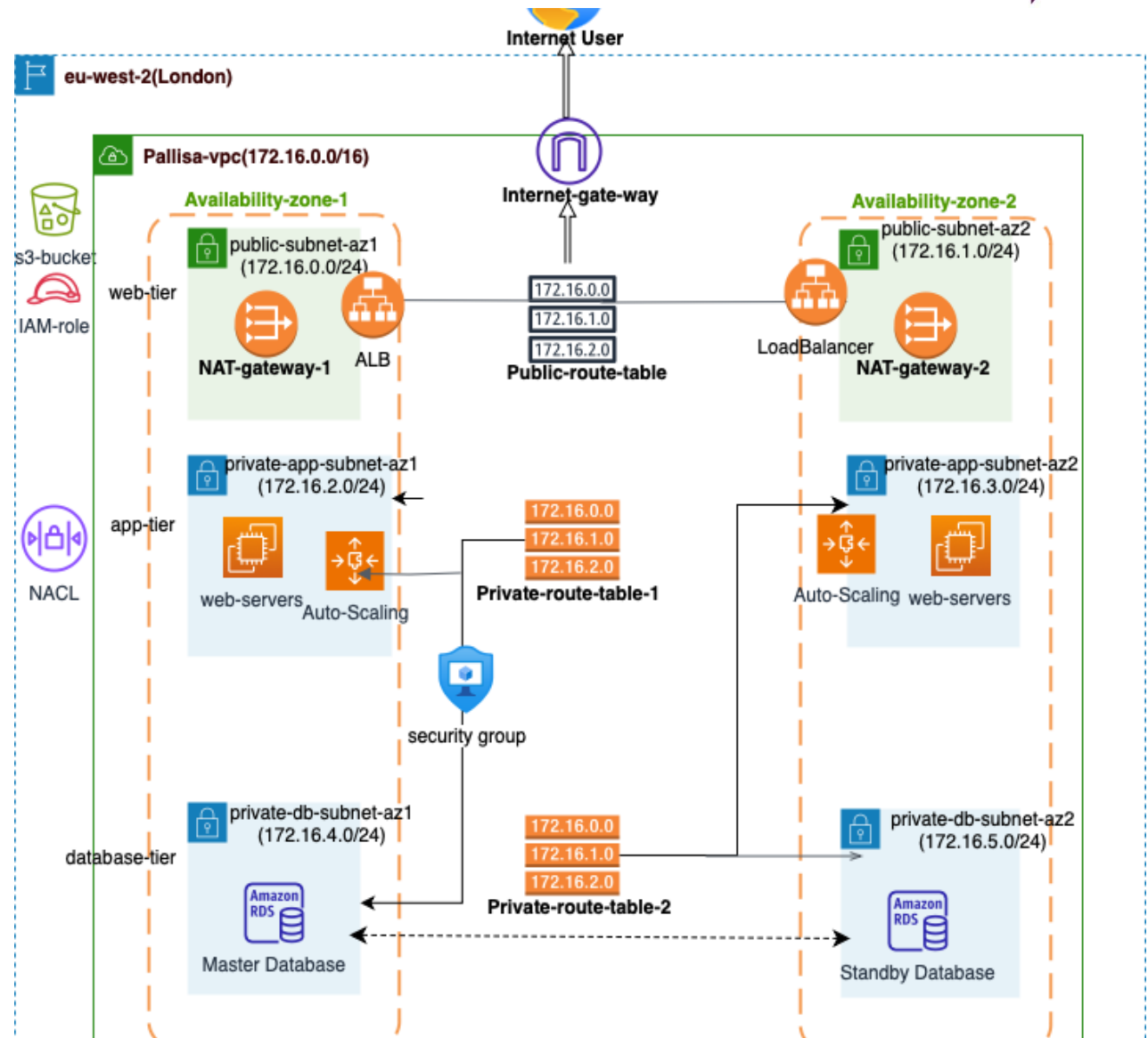
AWS provides a strong SLA for its services, guaranteeing up to of 99.00% for many of its services like EC2, s3, RDS etc.

➤ Performance Efficiency

Elastic load balancing for optimal traffic distribution, Auto scaling for dynamic resource management based on demand, broad range of EC2 types, each optimized for different workloads.

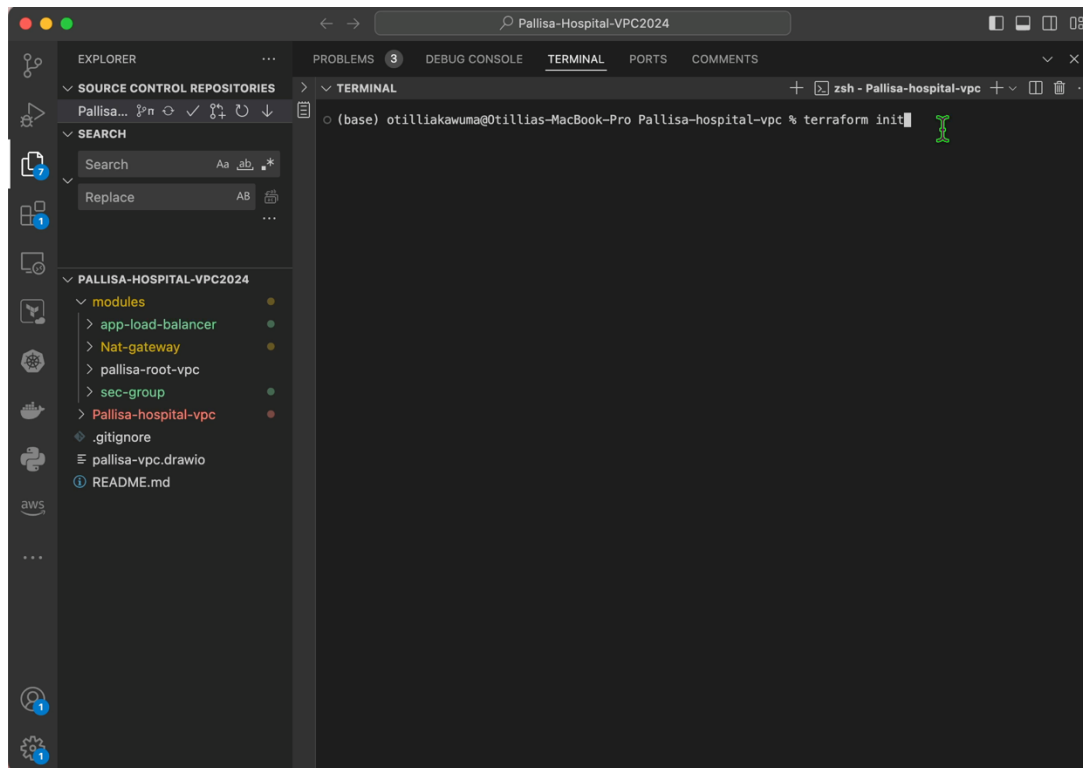
➤ Cost Optimization

Through auto scaling and EC2 spot instances that offer un-used AWS capacity at reduced prices, ideal for flexible workloads without uptime guarantees.



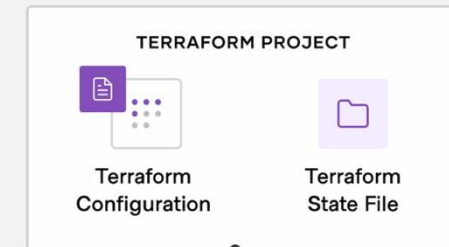
How Terraform works in AWS (terraform init, plan and apply)

A video demo of how the VPC code is executing.



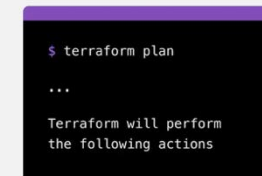
Write

Define infrastructure in configuration files



Plan

Review the changes
Terraform will make to
your infrastructure

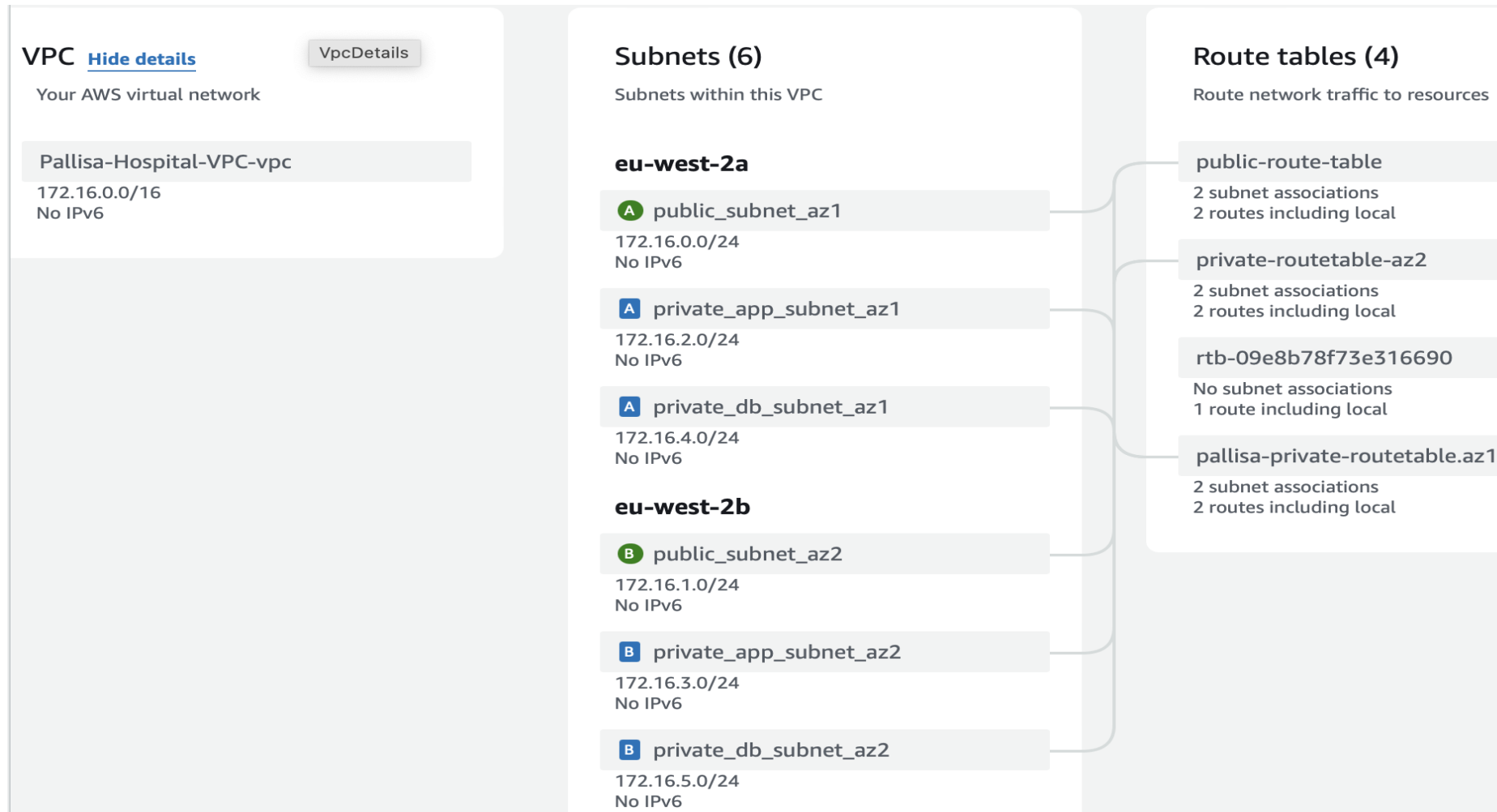


Apply

Terraform provisions
your infrastructure and
updates the state file.



A screen shot showing Pallisa Hospital VPC running in AWS provisioned using Terraform, with six subnets in two Availability Zones and four route tables



Main results of the project

Representing cloud infrastructure as code using tools like terraform has been on the rise due to the need to automate mundane tasks and repetitive IT processes.

- ❖ Development of a robust, secure, and scalable three-tier VPC architecture using terraform.
- ❖ Streamlined the provisioning of cloud resources. Knowledge of tiered resources.
- ❖ Infrastructure is consistent, easily standardized and repeatable across different environments like staging, testing and production.
- ❖ Allows for collaboration and knowledge sharing among diverse teams, there's need for code review and testing

Methodology used

An agile approach was used, we leveraged on iterations; Plan, Implement and Test.

In view is part of the Kanban board used to visualize and manage workflows



Projects / Pallisa-VPC

KAN board

Q Search

OK

GROUP BY No

TO DO 3

Create the Application Load Balancer for the root module

✓ KAN-10

OK

Create NAT-GATEWAY and Security group.

✓ KAN-11

OK

Initialize, validate and review the Terraform project folder.

✓ KAN-8

OK

+ Create issue

IN PROGRESS 2

Verify resources created in AWS and push both the root and child module folders to GitHub.

✓ KAN-9

OK

Access any applications deployed in the VPC environment

✓ KAN-12

OK

+ Create issue

DONE 6 ✓

Open an AWS account, Download and install the Terraform and AWS command line interfaces (CLI).

✓ KAN-1

✓ OK

Build VPC manually in the AWS console.

✓ KAN-3

✓ OK

Install plugins for Terraform and AWS. Configure AWS credentials

✓ KAN-2

✓ OK

Configure s3 bucket for state file locking.

✓ KAN-4

✓ OK

Skills and knowledge developed from taught modules and their application in the project.

Networking skills

- Use of public and private subnets
- Multi AZ-deployment for high availability
- Internet gateway, route tables and Load balancers

Security in the cloud

- Use of least privilege access like (IAM roles)
- Security groups at web, app and database tiers
 - Network access control lists

Contributions

First attempt of proposing the use of automation tools like terraform to design and build infrastructure in a local government department in Pallisa Hospital.

If this work is adopted by many other hospitals, departments and other government ministries, Infrastructure deployment time would massively reduce by 60%

With code, its quick and easy to deploy environments especially for testing that need destruction when application testing is done.

Lessons learned from the project



Thorough requirements gathering and understanding the project aims wholistically is key in determining project execution speed.



The value of using Terraform modules i.e., build once and use multiple times.



A good understanding of how terraform works, helps in quicker troubleshooting and development of robust infrastructure code



Need for continuous monitoring for early error detection and cutting costs

What would have been done differently if asked to complete a similar project

- ❖ Use of the scrum framework instead of Kanban, I realised that I am more productive with short fixed goals delivered within a scrum team ie, scrum master, product owner and sprint ceremonies like Sprint Planning, Daily Stand-ups, Sprint Reviews, and Retrospectives. I feel like these would have allowed me to talk through my work and help me understand it better.
- ❖ Though Kanban uses a visual board to manage workflow, it does not prescribe specific roles, lack of specific roles reduces the need for additional training or restructuring which I later realised I actually needed.

Challenges

- Deprecated arguments eg use of domain vs vpc
- Need to keep up with provider specific configurations
- Learning curve, terraform is syntax specific (HCL)
- S3 bucket issues with storing state file, should be explicitly mentioned in the terraform settings block.

Future works and areas to explore.

- Develop custom Terraform modules for healthcare-specific resources like EC2, Kubernetes clusters, Nat-gateways etc.
- Explore serverless architectures like use of lambda functions and provisioners for specific workflow tasks.
- Explore options for running infrastructure pipeline to facilitate continuous integration, development, deployment(CICD), monitoring and testing.

Why were these areas not explored?

- Knowledge curve
- Limited time availed