DevOps Intern Assignment Report

Candidate Name: Jayant Patil

Date: 25/05/2025

Assignment Title: Infrastructure as Code Setup for Prefect Worker on Amazon ECS

Tool Choice and Justification

I chose Terraform as the Infrastructure as Code (IaC) tool for this project due to its:

- Cloud-agnostic design: It supports multiple providers beyond AWS, offering future flexibility.
- Strong community support: Extensive modules and active forums significantly speed up development and troubleshooting.
- Modular architecture: Its support for modules and remote backends encourages clean, scalable codebases.
- Better ecosystem for iteration and testing: It integrates well with CI/CD and supports state management via Terraform Cloud or S3.

Key Learnings

- Provisioning and managing an ECS Fargate-based architecture using Terraform.
- Setting up IAM roles, security groups, NAT gateway, and subnet architectures in a multi-AZ VPC.
- Storing and retrieving secrets from AWS Secrets Manager securely for use in containerized services.
- Understanding Prefect Clouds work pool types, worker models, and their limitations.
- Debugging ECS task behavior using CloudWatch logs.

Challenges Faced and Resolutions

1. ECS Task Failing to Pull Image

Issue: ECS Fargate tasks failed to download the Docker image from DockerHub.

Root Cause: ECS tasks were using the default security group which had no inbound rules, and were

launched in private subnets without NAT access.

Solution:

- Created a custom ECS security group with inbound access from within the VPC and full outbound access.
- Deployed a NAT Gateway in the public subnet to allow internet-bound traffic from private subnets.
- Updated the ECS service to use the custom security group.

2. Secrets Handling

Issue: Initially attempted to hard-code the Prefect API key in environment variables.

Resolution: Moved the API key to AWS Secrets Manager and passed it to the ECS task securely using the secrets field in the task definition.

3. Prefect Work Pool Limitations

Issue: The default prefect:managed work pool does not require an external worker. As a result, the deployed ECS worker did not actively fetch or execute flows.

Root Cause: The free tier of Prefect Cloud only allows use of managed (push-based) pools.

Solution: Documented this limitation and validated that the worker was correctly deployed and connected, despite being inactive. For real usage, upgrading to a paid Prefect plan is required to create pull-based (worker-required) pools.

Demo and Verification

- ECS Cluster (prefect-cluster) and Fargate worker service (dev-worker) were successfully created.
- Verified log output in CloudWatch Logs, confirming successful container startup and connection to the Prefect API.
- Confirmed presence of Secrets Manager secret and its proper injection into the container.
- Verified ECS service was running in private subnets with internet access through NAT.
- Prefect Cloud confirmed workers connection, although inactive due to managed pool type.

Suggestions for Improvement

- Switch to pull-based pools: Upgrade to Prefects paid tier to fully utilize custom workers.

- Auto-scaling ECS tasks: Integrate ECS service auto-scaling based on CPU/memory metrics.

- CI/CD Integration: Use GitHub Actions or Terraform Cloud for automated infrastructure provisioning.

- Enhanced Monitoring: Add CloudWatch alarms or integrate external observability tools like Datadog or Prometheus.

- Logging enhancement: Enable ECS container insights and structured JSON logging.