

Prefect ECS Deployment - Report

1. Why I Chose Terraform

I chose Terraform over CloudFormation mainly due to its simplicity and flexibility. Having used both, I found Terraform's syntax much more readable and reusable. It's open-source and works well across different cloud platforms. It also makes managing infrastructure changes easier with features like plan/apply and state file tracking.

Terraform has a huge community, great module support, and is widely adopted in DevOps workflows, which made it the ideal choice for this assignment.

2. Key Learnings

◆ Infrastructure as Code (IaC)

Writing infrastructure in code makes it repeatable, version-controlled, and easier to review. I learned how these resources could be modularized in Terraform for better reuse and scalability, though for this assignment I kept the configuration flat for simplicity.

◆ Amazon ECS & Fargate

I understood how Fargate allows containerized workloads to run without managing servers. Learned how to configure task definitions, services, IAM roles, networking (VPC, subnets), and logging (CloudWatch) for ECS.

◆ Prefect Cloud Worker Integration

Registered a Prefect 2.0 worker that connects to a work pool in Prefect Cloud. Managed the Prefect API key securely via AWS Secrets Manager. Learned how the Prefect worker stays online, polls for work, and executes flows on demand.

◆ Terraform Data Sources and References

Understood the difference between data blocks and resource blocks. Learned how to fetch dynamic values like AWS Account ID using data sources and reference secrets securely.

◆ Handling Sensitive Variables

Used Terraform variable files (.tfvars) and locals to organize configuration. Managed sensitive inputs (like Prefect API key) using AWS Secrets Manager instead of hardcoding.

3. Challenges and Resolutions

◆ Secret Not Passing to ECS Task

I tried referencing the secret ARN directly in the ECS container definition but it didn't work. With help, I changed it to use secrets block properly in the container definition:

```
secrets = [  
  {  
    name      = "PREFECT_API_KEY"  
    valueFrom = data.aws_secretsmanager_secret_version.prefect_api_key.arn  
  }  
]
```

◆ Worker Not Connecting to Prefect Cloud

Initially, the worker wouldn't connect. After checking logs in CloudWatch, I noticed missing environment variables. The fix was to pass in both static and secret-based environment variables using the correct environment and secrets configuration blocks.

◆ Task Definition Confusion

I wasn't sure where to add the secrets block in task or container definition. I fixed it after verifying it must go into the container definition block.

4. Suggestions for Improvement

- ◆ Add Monitoring & Alerts using CloudWatch Alarms.
- ◆ Use Terraform modules to make the setup reusable across environments.
- ◆ Add CI/CD to automate deployment on commit.