Cloud IaC Deployment Architecture

This document describes the Cloud Infrastructure as Code (IaC) deployment architecture leveraging Terraform, AWS, and Azure DevOps. The architecture ensures consistent, secure, and automated infrastructure deployment across multiple environments with AWS Control Tower and Terraform AFT.

# 1. Components

* Terraform as Infrastructure as Code (IaC) tool
* AWS Cloud as the deployment target
* Azure DevOps for CI/CD pipelines
* AWS Control Tower for account management and governance
* Terraform AFT (Account Factory for Terraform) for standardized deployments
* Multiple environments: Development, Testing, Pre-Production, Production
* AWS Data Services: Redshift, Bedrock, SageMaker, Athena

# 2. Deployment Architecture

The deployment architecture is designed to ensure standardization, compliance, and automation. The following flow describes the deployment process:

1. 1. Developers write Terraform modules and configuration files.
2. 2. Code is stored in Azure Repos (Git repository).
3. 3. Azure DevOps pipelines (CI/CD) are triggered on commits.
4. 4. CI pipeline runs Terraform linting, validation, and security scans.
5. 5. CD pipeline applies Terraform plans into target AWS environments.
6. 6. AWS Control Tower manages accounts via Organizational Units (OUs).
7. 7. Terraform AFT ensures standardized account provisioning and deployment.
8. 8. Workloads and AWS Data Services (Redshift, Bedrock, SageMaker, Athena) are deployed in respective environments.

# 3. Environments

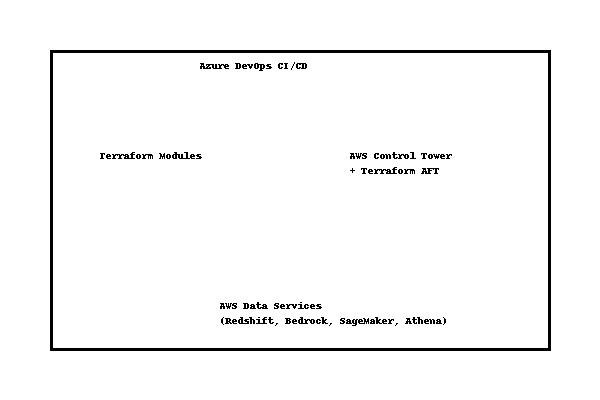
The architecture includes four environments to ensure proper testing and promotion of workloads:  
- Development: For initial development and experimentation.  
- Testing: For QA and functional validation.  
- Pre-Production: For staging and performance testing.  
- Production: For live workloads and customer-facing applications.

# 4. Governance and Security

AWS Control Tower ensures centralized governance and security guardrails across accounts. Terraform AFT enforces standard account baselines. Azure DevOps integrates with IAM roles for secure deployment into AWS. Security groups, VPC design, and private endpoints are applied to ensure data services (like Redshift, Bedrock, SageMaker, Athena) are not exposed publicly.

# 5. Deployment Architecture Diagram

The diagram below illustrates the high-level architecture:



# Updated Deployment Architecture Diagram

