Faysal Aziz

Terraforming the Modern Data Platform

Production-Grade AWS Data Engineering Infrastructure with Terraform and GitLab CI/CD

FA-0443 · 2 Days

Software Development · DevOps, Cloud & Infrastructure · Cybersecurity · Data & Analytics · Automation, RPA & Power Platform

---

Introduction

- 2 Days

Modern data platforms are no longer built through manual console operations or isolated engineering efforts. They are assembled as continuously evolving systems where infrastructure, orchestration, governance, automation, and operational reliability must work together with precision. As organizations scale data ingestion, analytics, machine learning, and governance initiatives, Infrastructure-as-Code and CI/CD pipelines become foundational capabilities rather than optional tooling choices.

This intensive two-day training is designed specifically for Data Engineers, Data Platform Engineers, and Cloud Infrastructure teams responsible for operating AWS-based data ecosystems. Rather than teaching Terraform as a generic DevOps utility, the course focuses on real-world data engineering architectures and production operating models. Participants will learn how Terraform, GitLab CI/CD, and AWS-native data services can be integrated into repeatable, scalable, and governable deployment workflows for modern ETL and lakehouse environments.

The program is built around practical enterprise patterns including batch ETL pipelines, S3-based data lakehouse architectures using Iceberg, orchestration using Step Functions, infrastructure modularization, environment promotion strategies, Terraform state governance, IAM provisioning, secrets management, and operational cost optimization. The instructor brings over 30 years of industry experience and delivers the course using production-driven implementation practices aligned with current enterprise adoption patterns and operational realities rather than purely academic examples.

Recent industry practices strongly emphasize GitLab-integrated Terraform pipelines, remote state governance using S3 and DynamoDB, policy validation, security scanning, drift detection, and multi-environment promotion workflows in production cloud environments.

Learning outcomes

By the end of this training, participants will be able to:

- Design production-grade AWS data platform infrastructure using Terraform
- Build reusable Terraform modules for enterprise data engineering workloads
- Provision and manage AWS data services using Infrastructure-as-Code
- Implement multi-environment Terraform deployment strategies
- Integrate Terraform workflows into GitLab CI/CD pipelines
- Design controlled infrastructure promotion workflows across environments
- Implement Terraform validation, security scanning, and approval workflows
- Manage Terraform remote state securely using AWS-native services
- Provision IAM roles and policies for data platform components
- Implement secrets management strategies for Terraform and CI/CD pipelines
- Deploy and manage data lakehouse infrastructure using Amazon S3 and Iceberg concepts
- Provision Glue ETL jobs, crawlers, workflows, and orchestration components
- Automate infrastructure governance and operational consistency
- Implement infrastructure drift detection and operational controls
- Apply cost optimization strategies for AWS data infrastructure and CI/CD operations
- Operate Terraform infrastructure safely in production-grade environments

Prerequisites

- Basic AWS operational knowledge
- Familiarity with Linux command-line operations
- Basic scripting knowledge
- Understanding of data engineering and ETL concepts
- General understanding of cloud networking and IAM concepts
- Familiarity with Git-based workflows
- Exposure to CI/CD concepts is beneficial but not mandatory

Training outline

1. Modern Data Platform Infrastructure and Infrastructure-as-Code Foundations

Evolution of Modern Data Platforms

1. Cloud-native data platform architecture
2. Infrastructure automation in data engineering
3. Operational challenges in large-scale ETL environments
4. Enterprise Infrastructure-as-Code adoption patterns
5. Infrastructure governance and repeatability

Terraform Fundamentals for Data Engineering

1. Terraform architecture and execution model
2. Declarative infrastructure concepts
3. Terraform workflow lifecycle
4. Providers, resources, modules, and data sources
5. State management fundamentals
6. Dependency management across AWS services
7. Infrastructure graph and execution planning

Terraform Ecosystem and Current Industry Practices

1. Terraform 1.x enterprise capabilities
2. GitOps and Infrastructure-as-Code operational models
3. Enterprise module standardization
4. Security and compliance integration
5. Infrastructure drift management
6. Policy-as-Code concepts
7. Infrastructure lifecycle governance

2. AWS Data Platform Reference Architectures

Batch ETL Platform Architecture

1. AWS DMS ingestion workflows
2. Amazon S3 landing zones
3. AWS Glue ETL processing
4. Redshift analytics integration
5. BI and downstream consumption layers
6. Operational monitoring architecture

Data Lakehouse Architecture

1. Amazon S3 data lake design
2. Iceberg table architecture concepts
3. Glue Data Catalog integration
4. Metadata management workflows
5. Partitioning and optimization considerations
6. Lakehouse operational patterns

Workflow Orchestration Architecture

1. AWS Step Functions orchestration patterns
2. Glue workflow coordination
3. Event-driven orchestration concepts
4. Failure handling strategies
5. Retry and compensation workflows
6. Monitoring and observability integration

Production Environment Design

1. Development, staging, and production isolation
2. Shared services architecture
3. Environment promotion models
4. Multi-account AWS strategies
5. Naming standards and tagging policies
6. Infrastructure governance structures

3. Terraform Environment Setup and Project Structuring

Development Environment Preparation

1. Terraform installation and configuration
2. AWS CLI integration
3. GitLab repository structure
4. IDE integration and tooling
5. Terraform formatting and validation tools
6. Secure credential handling

Terraform Project Organization

1. Monorepo versus multi-repository structures
2. Root module design
3. Child module structuring
4. Environment directory organization
5. Variable hierarchy management
6. Shared module strategies

Terraform Backend and State Management

1. Remote state architecture
2. S3 backend configuration
3. DynamoDB state locking
4. State isolation across environments
5. State security considerations
6. State recovery and troubleshooting
7. Drift detection concepts

4. Terraform Core Concepts Applied to Data Platforms

Variables and Configuration Management

1. Input variables
2. Local values
3. Output variables

4. tfvars organization
5. Dynamic configuration strategies
6. Environment parameterization

Modular Infrastructure Design

1. Reusable Terraform modules
2. Data platform component modularization
3. Shared infrastructure abstractions
4. Module dependency management
5. Versioning strategies
6. Enterprise module governance

Dependency and Resource Orchestration

1. Explicit and implicit dependencies
2. Resource lifecycle handling
3. Ordered provisioning
4. Cross-service dependencies
5. Managing asynchronous AWS services
6. Handling infrastructure changes safely

5. Provisioning AWS Data Platform Services with Terraform

Amazon S3 Data Lake Provisioning

1. Bucket provisioning strategies
2. Data lake zone structures
3. Lifecycle policies
4. Versioning and encryption
5. Replication considerations
6. Access control implementation

AWS DMS Infrastructure Provisioning

1. Replication instance deployment
2. Source and target endpoint provisioning

3. Migration task configuration
4. Networking and security setup
5. Monitoring integration

AWS Glue Infrastructure as Code

1. Glue database provisioning
2. Glue crawler deployment
3. Glue ETL job configuration
4. Glue workflow orchestration
5. Job parameterization
6. IAM role integration
7. Glue connection management

AWS Step Functions Orchestration Deployment

1. State machine provisioning
2. Workflow definitions
3. Service integrations
4. Error handling patterns
5. EventBridge integration
6. Operational monitoring

Amazon Redshift Infrastructure Provisioning

1. Redshift cluster deployment
2. Networking and subnet groups
3. Parameter groups
4. Workload management concepts
5. Security and encryption
6. Operational considerations

Amazon DynamoDB Configuration Stores

1. Configuration table provisioning
2. Scaling configuration
3. Partition key strategies

4. Encryption and access controls
5. Operational governance

Amazon EC2 for Data Processing Workloads

1. Compute provisioning patterns
2. IAM instance profiles
3. Security groups
4. Bootstrap and initialization
5. Operational monitoring

Monitoring and Logging Infrastructure

1. CloudWatch log groups
2. CloudWatch metrics integration
3. Alarm provisioning
4. Dashboard configuration
5. Operational observability patterns

Optional SageMaker Infrastructure Integration

1. Notebook provisioning concepts
2. IAM integration
3. Data access integration
4. Security considerations

6. IAM and Security Architecture for Data Platforms

IAM Design for Data Engineering Workloads

1. Least privilege architecture
2. Cross-service IAM roles
3. Trust relationships
4. Glue service roles
5. DMS access policies
6. Redshift access controls

Terraform-Based IAM Provisioning

1. IAM users and roles
2. Managed policies
3. Inline policies
4. Role assumption workflows
5. Federated access concepts

Secrets Management Strategies

1. GitLab CI/CD secret handling
2. AWS Secrets Manager integration
3. Secure variable injection
4. Avoiding credential hardcoding
5. Secret rotation considerations

Security and Compliance Validation

1. tfsec integration
2. Checkov scanning concepts
3. Terraform validation pipelines
4. Policy enforcement
5. Infrastructure compliance workflows

7. GitLab CI/CD for Terraform-Based Data Platforms

GitLab CI/CD Foundations

1. GitLab runner architecture
2. Pipeline execution models
3. Infrastructure repository workflows
4. Branching strategies
5. Merge request governance

Production-Grade Terraform Pipelines

1. terraform fmt

2. terraform validate
3. terraform plan
4. Plan artifact handling
5. terraform apply workflows
6. Controlled deployment execution

Multi-Environment Deployment Pipelines

1. Dev environment pipelines
2. Staging environment promotion
3. Production deployment workflows
4. Environment isolation
5. Pipeline parameterization
6. Environment approval gates

Infrastructure Change Governance

1. Manual approval stages
2. Merge request validation
3. Change management integration
4. Separation of duties
5. Controlled production deployment

Terraform State and Pipeline Integration

1. Remote backend integration
2. State locking workflows
3. Concurrent deployment handling
4. State integrity protection
5. Pipeline failure recovery

Security Integration within Pipelines

1. Secret injection practices
2. CI/CD variable protection
3. OIDC and temporary credentials
4. Infrastructure scanning stages

5. Artifact security considerations

Drift Detection and Operational Automation

1. Scheduled plan execution
2. Drift identification
3. Infrastructure reconciliation
4. Automated notifications
5. Operational reporting

8. Production Simulation Workshop

End-to-End Data Platform Deployment Workflow

1. Repository initialization
2. Module deployment
3. Environment provisioning
4. CI/CD pipeline execution
5. Infrastructure validation
6. Promotion across environments

Production Incident and Change Scenarios

1. Drift remediation simulation
2. Failed deployment recovery
3. State conflict handling
4. IAM troubleshooting
5. Infrastructure rollback scenarios

Disclaimer

This outline is intended as a professional training framework and planning guideline. The trainer reserves the right to amend, reorganize, expand, condense, or substitute topics, demonstrations, tooling approaches, and practical exercises where necessary to accommodate participant proficiency levels, operational priorities, platform updates, time constraints, or evolving industry practices without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur