

[**https://github.com/Abhishek-Kraj/Dataplex-Universal-Catalog/tree/feature/iss-foundation**](https://github.com/Abhishek-Kraj/Dataplex-Universal-Catalog/tree/feature/iss-foundation) **this my repo**

Based on the attached image showing Dataplex Universal Catalog's use cases and features, and referencing your GitHub repository, I can see you're building a comprehensive Terraform solution for Dataplex Universal Catalog implementation. Let me provide recommendations to enhance your repository structure and implementation.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)

**Repository Structure Recommendations**

Your repository should follow Terraform best practices with a modular structure similar to Google's recommended approach:[[3]](#fn3)[[4]](#fn4)[[5]](#fn5)

Dataplex-Universal-Catalog/  
├── modules/  
│ ├── entry-group/  
│ │ ├── main.tf  
│ │ ├── variables.tf  
│ │ ├── outputs.tf  
│ │ └── versions.tf  
│ ├── aspect-type/  
│ ├── entry-type/  
│ ├── catalog-entry/  
│ ├── data-quality-scan/  
│ ├── data-profiling-scan/  
│ └── business-glossary/  
├── environments/  
│ ├── dev/  
│ │ ├── terraform.tfvars  
│ │ ├── backend.tf  
│ │ └── main.tf  
│ └── prod/  
│ ├── terraform.tfvars  
│ ├── backend.tf  
│ └── main.tf  
├── examples/  
│ ├── basic-catalog/  
│ ├── custom-metadata/  
│ ├── data-quality-rules/  
│ └── bigquery-discovery/  
├── rules/  
│ └── yaml/  
│ └── data-quality-rules.yaml  
├── schemas/  
│ └── table-schemas/  
├── docs/  
│ ├── architecture.md  
│ └── deployment-guide.md  
└── README.md

**Key Features to Implement**

Based on the image, your implementation should cover these Dataplex Universal Catalog features:[[2]](#fn2)[[6]](#fn6)[[1]](#fn1)

**Metadata Cataloging**

Automatic metadata retrieval from BigQuery, Cloud SQL, Spanner, Vertex AI, Pub/Sub, Dataform, and Dataproc Metastore. Implement discovery modules for each source type.[[7]](#fn7)[[1]](#fn1)

**Data Discovery**

Scan Cloud Storage buckets for structured and unstructured data to extract metadata. Create modules for automatic discovery configuration.[[1]](#fn1)

**Data Insights**

Integrate AI-powered capabilities to generate natural language questions about data and perform statistical analyses.[[6]](#fn6)[[1]](#fn1)

**Data Profiling**

Create reusable modules for profiling BigQuery tables to identify common characteristics of column data.[[1]](#fn1)

**Data Quality**

Implement Terraform modules managing data quality rules as code using YAML configuration files. This follows the best practice of version-controlling data quality rules.[[8]](#fn8)[[9]](#fn9)[[1]](#fn1)

**Business Glossary**

Build modules for managing business terminology and attaching terms to table columns for consistent data usage understanding.[[2]](#fn2)[[1]](#fn1)

**Data Lineage**

Enable tracking of data movement through systems with source, destination, and transformation metadata.[[2]](#fn2)[[1]](#fn1)

**Terraform Module Implementation Examples**

**Entry Group Module**

# modules/entry-group/main.tf  
resource "google\_dataplex\_entry\_group" "this" {  
 entry\_group\_id = var.entry\_group\_id  
 project = var.project\_id  
 location = var.region  
   
 description = var.description  
 display\_name = var.display\_name  
   
 labels = var.labels  
}

**Data Quality Scan Module**

Following Google's recommended approach for managing data quality rules as code:[[8]](#fn8)

# modules/data-quality-scan/main.tf  
locals {  
 rules = yamldecode(file(var.rules\_file\_path))  
}  
  
resource "google\_dataplex\_datascan" "data\_quality" {  
 data\_scan\_id = var.scan\_id  
 project = var.project\_id  
 location = var.region  
   
 data {  
 resource = var.bigquery\_table\_resource  
 }  
   
 data\_quality\_spec {  
 dynamic "rules" {  
 for\_each = local.rules.rules  
 content {  
 dimension = rules.value.dimension  
 threshold = lookup(rules.value, "threshold", null)  
   
 dynamic "non\_null\_expectation" {  
 for\_each = lookup(rules.value, "non\_null\_expectation", null) != null ? [^1] : []  
 content {  
 column = rules.value.non\_null\_expectation.column  
 }  
 }  
   
 dynamic "range\_expectation" {  
 for\_each = lookup(rules.value, "range\_expectation", null) != null ? [^1] : []  
 content {  
 column = rules.value.range\_expectation.column  
 min\_value = lookup(rules.value.range\_expectation, "min\_value", null)  
 max\_value = lookup(rules.value.range\_expectation, "max\_value", null)  
 }  
 }  
   
 dynamic "uniqueness\_expectation" {  
 for\_each = lookup(rules.value, "uniqueness\_expectation", null) != null ? [^1] : []  
 content {  
 column = rules.value.uniqueness\_expectation.column  
 }  
 }  
 }  
 }  
 }  
   
 execution\_spec {  
 trigger {  
 schedule {  
 cron = var.scan\_schedule  
 }  
 }  
 }  
}

**YAML Rules Configuration**

# rules/yaml/customer-table-quality.yaml  
rules:  
 - dimension: "COMPLETENESS"  
 non\_null\_expectation:  
 column: "customer\_id"  
 threshold: 0.99  
   
 - dimension: "VALIDITY"  
 regex\_expectation:  
 column: "email"  
 regex: "^[a-zA-Z0-9+\_.-]+@[a-zA-Z0-9.-]+$"  
 threshold: 0.95  
   
 - dimension: "UNIQUENESS"  
 uniqueness\_expectation:  
 column: "customer\_id"  
 threshold: 1.0  
   
 - dimension: "ACCURACY"  
 range\_expectation:  
 column: "age"  
 min\_value: "0"  
 max\_value: "120"

**Best Practices for Your Repository**

**Service Account Permissions**

Use least-privilege service accounts for Terraform deployments. Create dedicated service accounts with only necessary Dataplex permissions.[[4]](#fn4)

**State Management**

Store Terraform state in Google Cloud Storage with versioning and encryption enabled. Implement state locking for concurrent operations.[[3]](#fn3)[[4]](#fn4)

**Environment Separation**

Maintain separate dev and prod environments with distinct tfvars files. Use workspace-based or directory-based separation.[[4]](#fn4)[[8]](#fn8)

**CI/CD Integration**

Implement Cloud Build triggers for automated Terraform deployments. Configure terraform plan for feature branches and terraform apply for protected branches.[[8]](#fn8)

**Module Versioning**

Tag module releases consistently and document changes. Store modules in separate repositories if shared across multiple projects.[[4]](#fn4)

**CMEK Implementation**

Enable Customer-Managed Encryption Keys for Dataplex resources with HSM protection level. Ensure key regions match Dataplex resource regions.[[1]](#fn1)

**Documentation**

Include comprehensive README files for each module with usage examples, input/output descriptions, and architecture diagrams.[[5]](#fn5)

**Integration Architecture**

Based on the unified data management architecture shown in the image, your implementation should support:[[1]](#fn1)

**Integrated Analytics Experience**: Connect catalog to Data Warehouses, Data Lakes, and Data Marts for unified governance.[[1]](#fn1)

**Unified Data Management**: Implement metadata, intelligence, lifecycle, governance, and security layers across all data sources.[[1]](#fn1)

This approach enables end-to-end data lifecycle management from distributed discovery to business insights with comprehensive governance through BigQuery integration.[[6]](#fn6)[[2]](#fn2)[[1]](#fn1)

⁂

1. image.jpg

1. <https://cloud.google.com/dataplex>

1. <https://cloud.google.com/dataplex/docs/terraform>

1. <https://controlmonkey.io/terraform-gcp-provider-best-practices/>

1. <https://www.env0.com/blog/terraform-modules>

1. <https://cloud.google.com/dataplex/docs/introduction>

1. <https://cloud.google.com/dataplex/docs/catalog-overview>

1. <https://cloud.google.com/dataplex/docs/manage-data-quality-rules-as-code>

1. <https://discuss.google.dev/t/dataplex-dq-rules-as-code-with-terraform/168404/2>

1. <https://github.com/GoogleCloudPlatform/dataplex-catalog-transfer-tooling>

1. <https://github.com/abhishek-ch/around-dataengineering>

1. <https://cloud.google.com/dataplex/docs/samples>

1. <https://googleapis.github.io/genai-toolbox/resources/sources/dataplex/>

1. <https://github.com/pradhan-abhishek/gcp>

1. <https://dev.to/ipt/data-governance-with-dbt-terraform-and-dataplex-a-practical-guide-to-bigquery-policy-tags-5f7d>

1. <https://www.coursera.org/learn/data-governance-with-dataplex-universal-catalog>

1. <https://dev.to/devopsfundamentals/gcp-fundamentals-cloud-dataplex-api-53mj>

1. <https://discuss.google.dev/t/subject-unable-to-programmatically-access-dataplex-universal-catalog-term-to-column-links-and-aspects-tags-via-api/255636>

1. <https://registry.terraform.io/modules/drandell/dataplex/google/latest>

1. <https://www.youtube.com/watch?v=t-rQ7tduYcY>

1. <https://www.cloudskillsboost.google/course_templates/1410/video/575422?locale=zh_TW>

1. <https://registry.terraform.io/providers/hashicorp/google/latest/docs/guides/provider_reference>

1. <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/data_catalog_entry>