CloudTuner Virtual Tagging Schema

Unified Multi-Cloud Resource Classification System

Version: 1.0
Date: October 28, 2025
Document Type: Technical Specification

Executive Summary

The **CloudTuner Virtual Tagging Schema** provides a unified framework for detecting, classifying, and tagging cloud resources across AWS, Azure, and Google Cloud Platform. This schema enables consistent cost allocation, compliance tracking, and resource management regardless of cloud provider.

Key Features

- **33 unified resource types** mapped across AWS, Azure, and GCP
- **44 standardized metadata fields** for comprehensive resource tracking
- **17 tag definitions** (7 required, 10 optional) for organizational governance
- **Provider-agnostic architecture** for seamless multi-cloud operations

1. CloudTuner Unified Resource Types

1.1 Compute Resources

CloudTuner Type	AWS	Azure	GCP
Compute_VirtualMachine	AWS::EC2::Instance	Microsoft.Compute/virtualMachines	compute.googleapis.com/Instance
Compute_Function	AWS::Lambda::Function	Microsoft.Web/sites	cloudfunctions.googleapis.com/Function
Compute_ContainerCluster	AWS::EKS::Cluster	Microsoft.ContainerService/managedClusters	container.googleapis.com/Cluster
Compute_ContainerService	AWS::ECS::Service	Microsoft.ContainerInstance/containerGroups	run.googleapis.com/Service
Compute_AutoScalingGroup	AWS::AutoScaling::AutoScalingGroup	Microsoft.Compute/virtualMachineScaleSets	compute.googleapis.com/InstanceGroupManager

1.2 Storage Resources

CloudTuner Type	AWS	Azure	GCP
Storage_ObjectStorage	AWS::S3::Bucket	Microsoft.Storage/storageAccounts (Blob)	storage.googleapis.com/Bucket
Storage_BlockVolume	AWS::EC2::Volume	Microsoft.Compute/disks	compute.googleapis.com/Disk
Storage_FileSystem	AWS::EFS::FileSystem	Microsoft.Storage/storageAccounts (File)	file.googleapis.com/Instance
Storage_Archive	AWS::Glacier::Vault	Microsoft.Storage/storageAccounts (Archive)	storage.googleapis.com/Bucket (Archive)

1.3 Database Resources

CloudTuner Type	AWS	Azure	GCP
Database_RelationalDB	AWS::RDS::DBInstance	Microsoft.Sql/servers/databases	sqladmin.googleapis.com/Instance
Database_NoSQL	AWS::DynamoDB::Table	Microsoft.DocumentDB/databaseAccounts	firestore.googleapis.com/Database
Database_Cache	AWS::ElastiCache::CacheCluster	Microsoft.Cache/Redis	redis.googleapis.com/Instance
Database_DataWarehouse	AWS::Redshift::Cluster	Microsoft.Synapse/workspaces	bigquery.googleapis.com/Dataset

CloudTuner Type	AWS	Azure	GCP
Database_Cluster	AWS::RDS::DBCluster	Microsoft.DBforPostgreSQL/serverGroupsv2	spanner.googleapis.com/Instance

1.4 Networking Resources

CloudTuner Type	AWS	Azure	GCP
Network_VirtualNetwork	AWS::EC2::VPC	Microsoft.Network/virtualNetworks	compute.googleapis.com/Network
Network_Subnet	AWS::EC2::Subnet	Microsoft.Network/virtualNetworks/subnets	compute.googleapis.com/Subnetwork
Network_LoadBalancer	AWS::ElasticLoadBalancingV2::LoadBalancer	Microsoft.Network/loadBalancers	compute.googleapis.com/ForwardingR...
Network_SecurityGroup	AWS::EC2::SecurityGroup	Microsoft.Network/networkSecurityGroups	compute.googleapis.com/Firewall
Network_Gateway	AWS::EC2::InternetGateway	Microsoft.Network/virtualNetworkGateways	compute.googleapis.com/VpnGateway...
Network_IPAddress	AWS::EC2::EIP	Microsoft.Network/publicIPAddresses	compute.googleapis.com/Address

1.5 Security & Identity Resources

CloudTuner Type	AWS	Azure	GCP
Security_IAMRole	AWS::IAM::Role	Microsoft.Authorization/roleAssignments	iam.googleapis.com/Role
Security_IAMUser	AWS::IAM::User	Microsoft.ManagedIdentity/userAssignedIdentities	iam.googleapis.com/ServiceAccount
Security_EncryptionKey	AWS::KMS::Key	Microsoft.KeyVault/vaults/keys	cloudkms.googleapis.com/CryptoKey
Security_Secret	AWS::SecretsManager::Secret	Microsoft.KeyVault/vaults/secrets	secretmanager.googleapis.com/Secret

2. Standard Metadata Fields

2.1 Resource Identification (Required)

- **resource_id**: Unique identifier within cloud provider
- **resource_name**: Human-readable resource name
- **resource_type**: CloudTuner unified resource type
- **resource_arn_uri**: Full resource identifier (ARN/URI)
- **creation_timestamp**: Resource creation date/time
- **last_modified_timestamp**: Last modification date/time

2.2 Cloud Provider Context (Required)

- **cloud_provider**: AWS | Azure | GCP
- **account_id**: AWS account ID
- **subscription_id**: Azure subscription ID
- **project_id**: GCP project ID
- **region**: Cloud provider region
- **availability_zone**: Specific availability zone
- **resource_group**: Azure resource group (if applicable)

2.3 Cost & Billing

- **cost_center**: Organizational cost center code
- **billing_account**: Billing account identifier
- **budget_code**: Budget allocation code
- **charge_code**: Chargeback code

- **cost_allocation_tag**: Custom cost allocation identifier

2.4 Organization

- **organization_id**: Parent organization identifier
- **business_unit**: Business unit name
- **department**: Department name
- **team**: Team name
- **owner_email**: Resource owner email
- **technical_contact**: Technical support contact

2.5 Environment

- **environment**: production | staging | development | test | qa | sandbox
- **lifecycle_stage**: active | deprecated | decommissioned
- **deployment_method**: terraform | cloudformation | manual | ansible
- **configuration_management**: Configuration tool used

2.6 Application

- **application_name**: Application identifier
- **application_id**: Unique application ID
- **application_version**: Application version
- **service_name**: Microservice name
- **component**: Application component
- **tier**: frontend | backend | database | cache

2.7 Compliance

- **compliance_framework**: SOC2 | GDPR | HIPAA | PCI-DSS | ISO27001
- **data_classification**: public | internal | confidential | restricted
- **encryption_status**: encrypted | unencrypted
- **backup_policy**: Backup frequency
- **retention_period**: Data retention period

2.8 Operations

- **monitoring_enabled**: Boolean flag
- **alerting_enabled**: Boolean flag
- **logging_level**: critical | high | medium | low
- **backup_enabled**: Boolean flag
- **auto_scaling_enabled**: Boolean flag

3. Required Tags for CloudTuner Schema

3.1 Mandatory Tags (Must be present on all resources)

Tag Key	Format	Example	Purpose	Validation
CostCenter	String	IT-OPS-001	Track costs by cost center	Alphanumeric with hyphens, 3-20 chars
BillingAccount	String	BA-12345	Associate with billing account	Alphanumeric, 5-15 chars

Tag Key	Format	Example	Purpose	Validation
Owner	Email	team@company.com	Identify resource owner	Valid email format
BusinessUnit	String	Engineering	Categorize by business unit	Predefined list
Team	String	CloudOps	Identify owning team	Alphanumeric, 3-50 chars
Environment	String	production	Deployment environment	Must be: production, staging, development, test, qa, sandbox
ApplicationName	String	cloudtuner-api	Associate with application	Alphanumeric with hyphens, 3-50 chars
DataClassification	String	confidential	Data sensitivity level	Must be: public, internal, confidential, restricted

3.2 Recommended Optional Tags

Tag Key	Format	Example	Purpose
ApplicationID	String	APP-001	Unique application identifier
ServiceTier	String	frontend	Service tier/layer
ComplianceFramework	String	SOC2,GDPR	Compliance requirements
BackupPolicy	String	daily	Backup frequency
MonitoringLevel	String	critical	Monitoring importance
CreatedDate	ISO 8601	2025-10-28	Resource creation date
ExpirationDate	ISO 8601	2026-01-15	Planned decommission date
ManagedBy	String	terraform	Infrastructure management
ProjectCode	String	PROJ-2025-001	Project association

4. Virtual Tagging Rules Engine

4.1 Rule Structure

Virtual tags are applied based on **detection rules** that evaluate resource metadata:

```
virtual_tag: Environment
rules:
  - value: production
    filters:
      - resource_name: contains "prod"
      - account_id: matches "prod-*"
      - native_tag.environment: equals "production"
    priority: 1

  - value: development
    filters:
      - resource_name: contains "dev"
      - account_id: matches "dev-*"
    priority: 2

  - value: untagged
    filters:
      - native_tag.environment: absent
    default: true
```

4.2 Detection Methods

Pattern-Based Detection:

- Resource naming conventions (e.g., "prod-web-server-01" → Environment: production)
- Account/subscription naming patterns
- Region-based categorization

Metadata-Based Detection:

- Resource type classification
- Cost threshold analysis
- Network topology inference

Native Tag Inheritance:

- Existing AWS/Azure/GCP tags
- Tag propagation from parent resources
- Default tag application

5. Implementation for CloudTuner.ai

5.1 Data Collection Layer

AWS Integration:

- AWS Config for resource discovery
- Cost and Usage Reports (CUR 2.0) for billing data
- Resource Groups Tagging API for native tags

Azure Integration:

- Azure Resource Graph for inventory
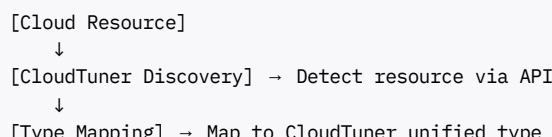- Cost Management API for billing
- Azure Resource Manager for metadata

GCP Integration:

- Cloud Asset Inventory for resources
- BigQuery Billing Export for costs
- Resource Manager API for metadata

5.2 CloudTuner Schema Mapping

1. **Ingest native resource data** from AWS/Azure/GCP APIs
2. **Map to CloudTuner unified types** using resource type mappings
3. **Extract metadata fields** into standardized schema
4. **Apply virtual tagging rules** to untagged resources
5. **Assign cost allocation tags** for FinOps reporting
6. **Store in CloudTuner database** for querying and analysis

5.3 Virtual Tag Assignment Workflow

```
[Cloud Resource]
     ↓
[CloudTuner Discovery] → Detect resource via API
     ↓
[Type Mapping] → Map to CloudTuner unified type
```

```
       ↓
[Metadata Extraction] → Extract standard metadata fields
       ↓
[Native Tag Check] → Does resource have native tags?
       ↓
[Virtual Tag Rules] → Apply detection rules
       ↓
[Tag Assignment] → Assign virtual tags
       ↓
[Cost Allocation] → Associate costs with tags
       ↓
[CloudTuner Database] → Store for reporting
```

6. Untagged Resource Detection

6.1 Detection Strategy

For resources missing native tags, CloudTuner applies:

1. **Naming Convention Analysis**
   - Extract patterns from resource names
   - Match against organizational standards
   - Example: "prod-api-server-us-east" → Environment: production, Region: us-east

2. **Account/Project Context**
   - Map account IDs to environments
   - Associate subscriptions with business units
   - Link projects to cost centers

3. **Resource Relationships**
   - Infer tags from parent resources
   - Apply VPC/network-level tags
   - Propagate from resource groups

4. **Default Tag Assignment**
   - Apply "untagged" category
   - Flag for manual review
   - Trigger notification to resource owner

6.2 Example Virtual Tagging Rules

Rule 1: Environment Detection

```
If resource_name contains "prod" OR account_id = "123456789012"
   → Environment: production

If resource_name contains "dev" OR account_id = "987654321098"
   → Environment: development

Else
   → Environment: untagged
```

Rule 2: Cost Center Assignment

```
If BusinessUnit = "Engineering" AND Team = "CloudOps"
   → CostCenter: IT-OPS-001

If BusinessUnit = "Sales"
   → CostCenter: SALES-001
```

```
Else
   → CostCenter: UNALLOCATED
```

Rule 3: Application Mapping

```
If resource_type = "Compute_VirtualMachine" AND resource_name contains "api"
   → ApplicationName: api-service, ServiceTier: backend

If resource_type = "Database_RelationalDB"
   → ServiceTier: database
```

7. Schema Benefits

7.1 For Cost Management

- **Unified cost allocation** across all three cloud providers
- **Granular cost attribution** by business unit, team, application
- **Automatic tagging** eliminates manual effort
- **Consistent reporting** with standardized tag schema

7.2 For Compliance

- **Automated classification** of data sensitivity
- **Compliance framework tracking** (SOC2, GDPR, HIPAA)
- **Audit trail** with metadata history
- **Policy enforcement** through required tags

7.3 For Operations

- **Resource inventory** across multi-cloud environments
- **Ownership identification** for incident response
- **Automated lifecycle management** based on tags
- **Improved resource governance** with consistent naming

8. Next Steps for CloudTuner.ai

Phase 1: Foundation (Weeks 1-2)

1. Implement resource discovery for AWS, Azure, GCP
2. Build CloudTuner unified type mapping
3. Create metadata extraction pipeline
4. Design virtual tagging rule engine

Phase 2: Rule Engine (Weeks 2-3)

1. Implement YAML-based rule parser
2. Build pattern detection algorithms
3. Create default tagging rules
4. Test with sample data from all three clouds

Phase 3: Integration (Weeks 3-4)

1. Connect to existing CloudTuner database
2. Build API endpoints for tag management
3. Create UI for rule configuration
4. Implement cost allocation logic

Phase 4: Testing & Launch (Week 4)

1. Test with pilot clients
2. Validate tag accuracy
3. Collect feedback
4. Launch MVP

Appendix A: Tag Validation Rules

Tag Key	Validation Rule	Error Message
CostCenter	Regex: `^[A-Z0-9-]{3,20}$`	"Cost center must be 3-20 alphanumeric characters with hyphens"
Owner	Email validation	"Owner must be a valid email address"
Environment	Enum validation	"Environment must be: production, staging, development, test, qa, sandbox"
DataClassification	Enum validation	"Data classification must be: public, internal, confidential, restricted"
BusinessUnit	Enum validation	"Business unit must be: Engineering, Sales, Marketing, Finance, HR, IT"

Appendix B: Cloud Provider Tag Limits

Provider	Max Tags per Resource	Max Key Length	Max Value Length
AWS	50	128 characters	256 characters
Azure	50	512 characters	256 characters
GCP	64 (labels)	63 characters	63 characters

CloudTuner Note: Virtual tags are stored in CloudTuner's database and are not subject to provider limits.

Document Information

Author: CloudTuner.ai Product Team
Classification: Internal Use
Version History:

- v1.0 (2025-10-28): Initial schema definition

Contact: For questions about this schema, contact the CloudTuner.ai engineering team.