

Cloud Computing - Cloud Computing Resources and Services

Group Members:

Linda Sekhoasha (222004139)

Wandile Ngobese (222056013)

Samukelo Mkhize (220009930)

Khonzinkosi Mkhize (219005273)

Purpose and Scope of Ontology

Our chosen topic is "Cloud Computing Resources and Services".

In today's digital era, cloud computing has become a cornerstone of modern infrastructure, enabling organizations to dynamically scale resources, improve efficiency, and innovate faster. However, the rapid development of the cloud computing landscape has led to inconsistencies in the way cloud resources are described, classified, and managed across different providers like AWS, Azure, Google Cloud, IBM Cloud, Alibaba Cloud, among others (Cloud Standards Customer Council, 2017). This disparity creates challenges for interoperability, migration, resource optimization, and unified management.

An ontology for cloud resources and services would serve as a standardized, conceptual framework to understand and organize critical information in this domain. Cloud computing covers a wide range of service models and resources, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and more. Within these, there are multiple types of resources like compute resources, storage resources, networking resources, databases, among other things (Buyya et al, 2013).

We decided to focus on our ontology's six major categories of cloud resources: Compute Resources, Storage Resources, Networking Resources, Database Resources, Service Metadata & Attributes, and Ontology use cases.

The purpose of this ontology is to provide an introductory overview of Cloud Service Category and standardize the terminology, hierarchies, and associations among different resource types.

An ontology like this would be useful to organizations developing multi-cloud strategies, cloud management platforms, among other things. It would also be helpful to academic research focused on cloud interoperability and cloud security frameworks.

Our ontology is aimed at making the process of cloud adoption for businesses and governments, promoting standardization across vendors, and encouraging innovation by making cloud environments more understandable and manageable. Companies building hybrid or multi-cloud architectures and developers creating cloud-native applications will find this ontology especially useful.

The scope of this ontology was limited to identifying major types of cloud resources, describing their properties and capabilities, listing any possible subtypes or variants within

each category, and outlining how each type of resource fits into broader cloud computing scenarios.

Cloud Computing Ontology

Competency Questions

Cluster 1: Service Models & Resource Classification

- 1. What Cloud_Computing_Models are defined?
- Which Cloud_Resources use the laaS service model?
- 3. Which Cloud Resources use the PaaS service model?
- 4. Which Cloud_Resources use the SaaS service model?
- 5. What are the subtypes of laaS_Resource?
- 6. What are the subtypes of PaaS_Resource?
- 7. What are the subtypes of SaaS_Resource?
- 8. Given a resource class (e.g., CRM_Resource), what service model does it use?
- 9. What service model does an laaS Resource use?
- 10. What service model does a PaaS Resource use?
- 11. What service model does a SaaS_Resource use?
- 12. Which classes are direct subclasses of Cloud Resource?
- 13. Which resource types are defined overall?

Cluster 2: Cloud Providers & Top-Level Services

- 14. Which organizations are modeled as Cloud Service Providers (CSPs)?
- 15. How many CSPs are represented in the ontology?
- 16. What services does each CSP (AWS, Microsoft Azure, GCP, IBM Cloud, Oracle Cloud, Alibaba Cloud) offer?
- 17. Which cloud provider offers the service AWS EC2 and Azure Functions?
- 18. What services are categorized under Virtual Machine, Object Storage, Container Orchestration, and other top-level services?
- 19. Which CSPs provide function-as-a-service and database management services (e.g., AWS_Lambda, Amazon_RDS)?

- 20. What networking services (e.g., VPN Gateway, VPC) are offered by CSPs?
- 21. Can a service (e.g., Amazon_EC2) be provided by multiple CSPs, and what services are unique to each provider?
- 22. How are cloud resources connected to service models or providers in the ontology?

Cluster 3: Concrete Service Instances & Provisioning

- 23. Which services are classified as Virtual_Machine_Service?
- 24. Which resources are instances of Object_Storage_Resource?
- 25. What cloud services are instances of Container_Orchestration_Service?
- 26. Which services are categorized under Database_Resource?
- 27. What services are considered Serverless_Compute_Service or Function_as_a_Service?
- 28. Which services are modeled as VPN_Gateway_Resource?
- 29. Is Amazon_EC2 modeled as a Virtual_Machine or a Virtual_Machine_Service?
- 30. Which DNS-related services are represented in the ontology?
- 31. Are there any services classified as Auto_Scaling_Service? If so, which ones?
- 32. Which providers have services that fall under Load_Balancer_Resource?

Cluster 4: Standardized Terms & Mappings

- 33. Which concepts are modeled as Standardized_Term?
- 34. What instances are defined as Interoperability_Mapping?
- 35. Is Compute_Instance a Standardized_Term?
- 36. Is there a mapping between ObjectStorage and S3?
- 37. Which API standards are included in the ontology?
- 38. Is gRPC classified as an API_Standard?
- 39. Does the ontology model a mapping for Compute_Instance to EC2?

- 40. Which concepts are modeled as Performance_Metric?
- 41. Is CPU_Utilization a type of Performance_Metric?
- 42. Does the ontology include Cold_Start_Latency as a performance metric?
- 43. How many Performance_Metric instances are modeled in the ontology?
- 44. Are Disk_IOPS and Query_Response_Time both categorized as performance metrics?

Cluster 6: Cost Models

- 45. Which concepts are classified as instances of Cost_Model?
- 46. Is Pay_as_you_go a valid cost model in the ontology?
- 47. Does the ontology include both Spot_Pricing and Reserved_Instance_Pricing as cost models?
- 4. What cost models are associated with AWS_EC2?
- 5. Which services use Storage_Tier_Pricing as their cost model?
- 6. Does Azure_Functions use a PayAsYouGo_Pricing model?
- 7. Which cloud services are billed using Data_Transfer_Pricing?
- 8. Are both Oracle_Cloud_Database_Service and Amazon_RDS offered under Reserved_Pricing?
- 48. How many services use OnDemand_Pricing as a cost model?
- 49. Which services use more than one cost model?
- 50. Is Alibaba_ECS associated with both Spot_Pricing and OnDemand_Pricing?

Cluster 7: Security & Compliance Attributes

- 51. What security features does AWS_EC2 have?
- 52. Which cloud services comply with Regulatory_Compliance?
- 53. Does IBM_Cloud_Object_Storage support Encryption_At_Rest?
- 54. Does GCP_Compute_Engine have Access_Control as a compliance feature?
- 55. Which services offer Role_Based_Access_Control?
- 56. Does Azure_LB provide SSL_Offloading?
- 57. Does Alibaba_OSS support Data_Encryption?
- 58. Which services have Key_Pair_Authentication?
- 59. Is Transparent_Data_Encryption a security or compliance feature?

- 60. Which services support DNSSEC?
- 61. Does Azure_Functions use Managed_Identity?
- 62. What compliance features are associated with IBM_Cloud_Database?

Cluster 8: Availability & Reliability Attributes

- 63. Which services offer Auto_Scaling as an availability feature?
- 64. Does AWS_RDS support Failover_Cluster as a reliability feature?
- 65. What services provide Multi_Region_Redundancy?
- 66. Which services use Active_Geo_Replication for reliability?
- 67. Does GCS_Bucket support Multi_Region_Storage?
- 68. Is Azure_SQL_Database equipped with Failover_Cluster?
- 69. Which services support Regional_Failover?
- 70. Does Amazon_EC2 offer Auto_Scaling?
- 71. What services provide Redundancy_Across_AZs?
- 72. Which services offer Availability_Sets as an availability feature?
- 73. Does Alibaba_OSS support Multi_Region_Redundancy?
- 74. What services have Uptime_SLA as an availability feature?

Cluster 9: Usage, Composition & Recommendations

- 75. Which services use Compute_Instance as a standardized term?
- 76. What services use Serverless_Computing as a standardized term?
- 77. Which services are categorized as Object_Storage?
- 78. What services are categorized under Virtual_Network?
- 79. Which services are classified as Load_Balancer?
- 80. Which services use VPN_Gateway as a standardized term?
- 81. Which services are defined as DNS_Service?
- 82. Which services are categorized as Relational_Database?
- 83. What services use Data Warehouse as a standardized term?
- 84. Which services are classified as Kubernetes_Service?
- 85. What services are classified as Auto_Scaling?
- 86. Which services use Block_Storage as a standardized term?
- 87. What services are classified as Database_Service?

88. Which services are categorized as Function_as_a_Service?

Cluster 10: Composite Services & Patterns

- 89. What are the composite services in the system?
- 90. What service components make up Multi_Cloud_Web_App_Stack?
- 91. What service components are part of Hybrid_Storage_Service?
- 92. What pattern does Multi_Cloud_Web_App_Stack follow?
- 93. What pattern does Hybrid_Storage_Service follow?

Cluster 11: Deployment & Tenancy

- 94. What are the cloud deployment models in the system?
- 95. Who operates each cloud deployment model (Public, Private, Hybrid, Community)? What tenancy model does each cloud deployment model have?
- 96. Who manages and owns the Private Cloud?

Vocabulary

Classes

1. Cloud Models & Layers

- 1.1. Cloud_Computing_Model
- 1.2. laaS
- 1.3. PaaS
- 1.4. SaaS
- 1.5. Cloud_Resource
- 1.6. Cloud Service
- 1.7. Cloud_Service_Provider

2. <u>Cloud Services and Components</u>

- 2.1. Virtual_Machine_Service
- 2.2. Object_Store_Service

- 2.3. Container_Orchestration_Service
- 2.4. Kubernetes_Service

3. <u>laaS Resources</u>

- 3.1. laaS_Resource
- 3.2. Compute_Resource
- 3.3. Virtual_Machine
- 3.4. Virtual_Machine_Service
- 3.5. Storage_Resource
- 3.6. Block_Storage
- 3.7. Object_Storage_Resource
- 3.8. Block_Storage
- 3.9. Network_Resource
- 3.10. Virtual_Network_Resource
- 3.11. Load_Balancer_Resource
- 3.12. VPN_Gateway_Resource
- 3.13. DNS_Resource
- 3.14. Function_As_A_Resource

4. <u>PaaS Resource</u>

- 4.1. PaaS_Resource
- 4.2. Container_Orchestration_Service
- 4.3. Kubernetes_Service

5. <u>SaaS Resource</u>

- 5.1. SaaS_Resource
- 5.2. CRM_Resource
- 5.3. Email_Resource
- 5.4. Analytics_Resource

6. <u>Data Related</u>

- 6.1. Data_Resource
- 6.2. Database_Resource
- 6.3. Data_Warehouse_Resource

7. <u>Security and Compliance</u>

- 7.1. Security_Resource
- 7.2. Security_Feature
- 7.3. Compliance_Feature
- 7.4. Security_And_Compliance_Attribute

8. <u>Deployment Models</u>

- 8.1. Public_Cloud
- 8.2. Private_Cloud
- 8.3. Hybrid_Cloud

9. <u>Tenancy</u>

- 9.1. Tenancy_Model
- 9.2. Single_Tenant
- 9.3. Multi_Tenant

10. <u>Metadata and Description</u>

- 10.1. Service_Metadata_Attribute
- 10.2. Performance_Metric
- 10.3. Cost_Model
- 10.4. Availability_And_Reliability_Attribute
- 10.5. Standardized_Term
- 10.6. Interoperability_Mapping

10.7. API_Standard

11. Workload and Optimization

- 11.1. Workload
- 11.2. Workload_Profile
- 11.3. Optimization_Goal
- 11.4. Recommendation_Engine
- 11.5. Composite_Function
- 11.6. Service_Component
- 11.7. Composition_Pattern

12. <u>Organizational Entities</u>

- 12.1. Organization
- 12.2. Region

Properties

1. Object Properties

- 1.1. managedBy
- 1.2. builtBy
- 1.3. ownedBy
- 1.4. operatedBy
- 1.5. isProvidedBy
- 1.6. instanceOf
- 1.7. isA
- 1.8. includes
- 1.9. includesComponent
- 1.10. composedOf / hasComponent
- 1.11. hasProperty
- 1.12. hasServiceModel
- 1.13. hasPerformanceMetric

- 1.14. hasCostModel
- 1.15. hasSecurityFeature
- 1.16. hasComplianceFeature
- 1.17. hasAvailabilityFeature
- 1.18. hasReliabilityFeature
- 1.19. hasTenancy
- 1.20. followsPattern
- 1.21. optimizedFor
- 1.22. suitableFor
- 1.23. usesStandardizedTerm

2. <u>Data Properties</u>

- 2.1. hasMemory
- 2.2. hasCPUCount
- 2.3. hasBandwidth
- 2.4. hasLatency
- 2.5. hasPrice
- 2.6. hasSLA
- 2.7. hasRequestDuration
- 2.8. hasStorageTier
- 2.9. hasColdStartLatency
- 2.10. hasMemoryConsumption
- 2.11. hasNetworkThroughput

Instances

Cloud Services and Components:

- 1. AWS_EC2
- 2. AWS_S3
- 3. AWS_Lambda
- 4. AWS_RDS
- 5. AWS_EKS
- 6. AWS_VPC
- 7. AWS VGW
- 8. Route_53
- 9. AWS_RedShift
- 10. ELB (Elastic Load Balancer)
- 11. Azure_VM

- 12. Azure_LB
- 13. Azure_VPN_Gateway
- 14. Azure DNS
- 15. Azure_SQL_Database
- 16. Azure_AKS
- 17. Azure_Scale_Sets
- 18. Azure_Functions
- 19. GCP_Compute_Engine
- 20. GCS_Bucket
- 21. GCP GKE
- 22. IBM_Cloud_Virtual_Server
- 23. IBM_Cloud_Functions
- 24. IBM_Cloud_Object_Storage
- 25. IBM_Cloud_Kubernetes_Service
- 26. IBM_Cloud_Database
- 27. IBM_Cloud_Load_Balancer
- 28. IBM Cloud VPC
- 29. IBM_Cloud_Transit_Gateway
- 30. Oracle_Cloud_VM
- 31. Oracle_Cloud_Functions
- 32. Oracle_Cloud_Block_Storage
- 33. Oracle_Cloud_Object_Storage
- 34. Oracle_Cloud_Kubernetes_Engine
- 35. Oracle_Cloud_Database_Service
- 36. Oracle_Cloud_Load_Balancer
- 37. Oracle_Cloud_VCN
- 38. Alibaba_ECS
- 39. Alibaba_Function_Compute
- 40. Alibaba_OSS
- 41. Alibaba_Container_Service
- 42. Alibaba_ApsaraDB
- 43. Alibaba SLB
- 44. Alibaba_VPC
- 45. Alibaba_SAG

Other Services:

- 46. Google_Cloud
- 47. Salesforce

- 48. HubSpot
- 49. Zoho_CRM
- 50. Microsoft_365_Email
- 51. Gmail
- 52. SendGrid
- 53. Google_Analytics
- 54. Adobe_Analytics
- 55. Mixpanel

Performance Metrics:

- 56. CPU_Utilization
- 57. RAM_Usage
- 58. Network_Latency
- 59. Data_Throughput

Pricing Models:

- 60. Pay_as_you_go
- 61. Subscription_Based
- 62. Spot_Pricing
- 63. Storage_Tier_Pricing
- 64. vCore_Pricing
- 65. DTU_Pricing
- 66. Storage_Class_Pricing
- 67. Pay_per_Execution
- 68. Reserved_Instance_Pricing
- 69. PayAsYouGo_Pricing
- 70. OnDemand_Pricing
- 71. Reserved_Pricing
- 72. Data_Transfer_Pricing
- 73. Request_Pricing
- 74. Preemptible_Pricing

Security and Compliance Attributes:

- 75. Data_Encryption
- 76. Access_Control
- 77. Regulatory_Compliance
- 78. Uptime_SLA
- 79. Failover Mechanism
- 80. System_Redundancy

Cloud Platforms:

- 81. AWS
- 82. Azure
- 83. GCP (Google_Cloud_Platform)
- 84. IBM Cloud
- 85. Alibaba Cloud

Cloud Services and Components:

- 86. Mapping_ComputeInstance_to_EC2
- 87. Mapping_ObjectStorage_to_S3
- 88. Open_API
- 89. gRPC
- 90. OCCI

Workloads and Selectors:

- 91. High_CPU_Workload
- 92. Memory_Optimized_Workload
- 93. Storage_Optimized_Workload
- 94. General_Purpose_Workload
- 95. GPU_Workload
- 96. Network_Optimized_Workload
- 97. Latency_Sensitive_Workload

- 98. ML_Based_Selector
- 99. Rules_Based_Selector

Optimisation and Patterns:

- 100. Cost_Optimization
- 101. Performance_Maximization
- 102. Compliance_Alignment
- 103. Hybrid_Storage_Service
- 104. Multi_Cloud_Web_App_Stack
- 105. Frontend_Hosting
- 106. Backend_Compute
- 107. Database_Layer
- 108. Data_Replication_Pattern
- 109. Burst_To_Cloud_Pattern
- 110. Failover_Composition_Pattern

Axioms

Cluster 1: Service Models & Resource Classification

```
1. IaaS is Cloud_Computing_Model
FOL: ∀x(IaaS(x) → Cloud_Computing_Model(x))
DL: IaaS ⊆ Cloud_Computing_Model
```

2. PaaS is Cloud_Computing_Model
FOL: ∀x(PaaS(x) → Cloud_Computing_Model(x))

```
DL: PaaS ⊆ Cloud_Computing_Model
```

3. SaaS is Cloud_Computing_Model

```
FOL: \forall x(SaaS(x) \rightarrow Cloud\_Computing\_Model(x))
```

DL: SaaS ⊆ Cloud_Computing_Model

4. IaaS_Resource has a service model IaaS FOL: $\forall x(IaaS_Resource(x) \rightarrow \exists y(IaaS(y) \land hasServiceModel(x, y)))$

```
DL: IaaS Resource ⊆ ∃hasServiceModel.IaaS
5. PaaS Resource has a service model PaaS
   FOL: \forall x (PaaS_Resource(x) \rightarrow \exists y (PaaS(y) \land hasServiceModel(x, y)))
   DL: PaaS Resource ⊆ ∃hasServiceModel.PaaS
6. SaaS Resource has a service model SaaS
   FOL: \forall x(SaaS\_Resource(x) \rightarrow \exists y(SaaS(y) \land hasServiceModel(x, y)))
   DL: SaaS_Resource ⊆ ∃hasServiceModel.SaaS
7. IaaS_Resource is Cloud_Resource
   FOL: \forall x(IaaS\_Resource(x) \rightarrow Cloud\_Resource(x))
   DL: IaaS Resource ⊆ Cloud Resource
8. PaaS Resource is Cloud Resource
   FOL: \forall x (PaaS_Resource(x) \rightarrow Cloud_Resource(x))
   DL: PaaS_Resource ⊆ Cloud_Resource
9. SaaS_Resource is Cloud_Resource
   FOL: \forall x(SaaS_Resource(x) \rightarrow Cloud_Resource(x))
   DL: SaaS_Resource ⊆ Cloud_Resource
10. Compute_Resource is IaaS_Resource
   FOL: \forall x (Compute\_Resource(x) \rightarrow IaaS\_Resource(x))
   DL: Compute_Resource ⊆ IaaS_Resource
11. Storage_Resource is IaaS_Resource
   FOL: \forall x (Storage\_Resource(x) \rightarrow IaaS\_Resource(x))
   DL: Storage_Resource ⊆ IaaS_Resource
12. Network Resource is IaaS Resource
   FOL: \forall x (Network\_Resource(x) \rightarrow IaaS\_Resource(x))
   DL: Network Resource ⊆ IaaS Resource
13. Data Resource is PaaS Resource
   FOL: \forall x(Data\_Resource(x) \rightarrow PaaS\_Resource(x))
   DL: Data Resource ⊆ PaaS Resource
14. Function_as_a_Resource is IaaS_Resource
   FOL: \forall x (Function\_As\_A\_Resource(x) \rightarrow IaaS\_Resource(x))
```

```
DL: Function as a Resource ⊆ IaaS Resource
15. Auto_Scaling_Service is PaaS_Resource
   FOL: \forall x (Auto\_Scaling\_Service(x) \rightarrow PaaS\_Resource(x))
   DL: Auto_Scaling_Service ⊆ PaaS_Resource
16. AI ML Resource is PaaS Resource
  FOL: \forall x(AI\_ML\_Resource(x) \rightarrow PaaS\_Resource(x))
  DL: AI ML Resource ⊆ PaaS Resource
17. CRM_Resource is SaaS_Resource
  FOL: \forall x (CRM\_Resource(x) \rightarrow SaaS\_Resource(x))
  DL: CRM_Resource ⊆ SaaS_Resource
18. Email Resource is SaaS Resource
  FOL: \forall x (Email\_Resource(x) \rightarrow SaaS\_Resource(x))
  DL: Email_Resource ⊆ SaaS_Resource
19. Analytics_SaaS_Resource is SaaS_Resource
  FOL: \forall x (Analytics\_SaaS\_Resource(x)) \rightarrow SaaS\_Resource(x))
   DL: Analytics_SaaS_Resource ⊆ SaaS_Resource
```

Cluster 2: Cloud Providers & Top-Level Services

AWS is an instance of Cloud_Service_Provider
 FOL: Cloud_Service_Provider(AWS)
 DL: Cloud_Service_Provider(AWS)

 Azure is an instance of Cloud_Service_Provider
 FOL: Cloud_Service_Provider(Azure)
 DL: Cloud_Service_Provider(Azure)

 Google_Cloud_Platform(GCP) is an instance of Cloud_Service_Provider
 FOL: Cloud_Service_Provider(Google_Cloud_Platform)
 DL: Cloud_Service_Provider(Google_Cloud_Platform)
 DL: Cloud_Service_Provider(Google_Cloud_Platform)

```
IBM Cloud is an instance of Cloud Service Provider
   23.
      FOL: Cloud_Service_Provider(IBM_Cloud)
      DL: Cloud_Service_Provider(IBM_Cloud)
   24.
         Oracle Cloud is an instance of Cloud Service Provider
      FOL: Cloud_Service_Provider(Oracle_Cloud)
      DL: Cloud_Service_Provider(Oracle_Cloud)
   25. Alibaba Cloud is an instance of Cloud Service Provider
      FOL: Cloud_Service_Provider(Alibaba_Cloud)
      DL: Cloud_Service_Provider(Alibaba_Cloud)
AWS (Amazon Web Services)
   26. AWS_EC2 is provided by AWS
      FOL: \forall x (AWS_EC2(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x, y)))
      DL: isProvidedBy(AWS_EC2, AWS)
   27. AWS_Lambda is provided by AWS
      FOL: \forall x (AWS_Lambda(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y)))
      DL: isProvidedBy(AWS_Lambda, AWS)
   28. AWS_S3 is provided by AWS
      FOL: \forall x(AWS_S3(x) \rightarrow \exists y(AWS(y) \land isProvidedBy(x,y)))
      DL: isProvidedBy(AWS_S3, AWS)
   29. AWS_VPC is provided by AWS
      FOL: \forall x (AWS_VPC(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(AWS_VPC, AWS)
   30. ELB (Elastic Load Balancer) is provided by AWS
      FOL: \forall x (ELB(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(ELB, AWS)
   31. AWS_VGW (VPN Gateway) is provided by AWS
      FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(AWS_VGW, AWS)
   32. Route_53 is provided by AWS
      FOL: \forall x (Route_53(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
```

```
DL: isProvidedBy(Route_53, AWS)
   33. AWS_RDS is provided by AWS
      FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(AWS_RDS, AWS)
   34. AWS_RedShift is provided by AWS
      FOL: \forall x (AWS_RedShift(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(AWS_RedShift, AWS)
        AWS_EKS (Elastic Kubernetes Service) is provided by AWS
      FOL: \forall x (AWS_EKS(x) \rightarrow \exists y (AWS(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(AWS_EKS, AWS)
Azure (Microsoft Azure)
   36. Azure_VM is provided by Azure
      FOL: \forall x(Azure_VM(x) \rightarrow \exists y(Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_VM, Azure)
   37. Azure_VNet is provided by Azure
      FOL: \forall x(Azure\_VNet(x) \rightarrow \exists y(Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_VNet, Azure)
   38. Azure_LB is provided by Azure
      FOL: \forall x(Azure\_LB(x) \rightarrow \exists y(Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_LB, Azure)
   39. Azure_VPN_Gateway is provided by Azure
      FOL: \forall x (Azure_VPN_Gateway(x) \rightarrow \exists y (Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_VPN_Gateway, Azure)
   40. Azure_DNS is provided Azure
      FOL: \forall x (Azure\_DNS(x) \rightarrow \exists y (Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_DNS, Azure)
   41. Azure_SQL_Database is provided Azure
      FOL: \forall x (Azure\_SQL\_Database(x) \rightarrow \exists y (Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_SQL_Database, Azure)
```

```
42. Azure_AKS (Azure Kubernetes Service) is provided Azure
       FOL: \forall x(Azure\_AKS(x) \rightarrow \exists y(Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_AKS, Azure)
   43. Azure_Scale_Sets is provided Azure
      FOL: \forall x(Azure\_Scale\_Sets(x) \rightarrow \exists y(Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_Scale_Sets, Azure)
   44. Azure Functions is provided Azure
      FOL: \forall x (Azure\_Functions(x) \rightarrow \exists y (Azure(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(Azure_Functions, Azure)
Google Cloud Platform (GCP)
   45. GCP_Compute_Engine is provided GCP
      FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (GCP(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(GCP_Compute_Engine, GCP)
   46. GCS_Bucket (Google Cloud Storage) is provided GCP
      FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (GCP(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(GCS_Bucket, GCP)
   47. GCP_GKE (Google Kubernetes Engine) is provided GCP
       FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (GCP(y) \land isProvidedBy(x,y))
      DL: isProvidedBy(GCP_GKE, GCP)
IBM Cloud
          IBM_Cloud_Virtual_Server is provided IBM_Cloud
      FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
       isProvidedBy(x,y))
       DL: isProvidedBy(IBM_Cloud_Virtual_Server, IBM_Cloud)
          IBM_Cloud_Functions is provided IBM_Cloud
      FOL: \forall x (IBM\_Cloud\_Functions(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
       isProvidedBy(x,y)
      DL: isProvidedBy(IBM_Cloud_Functions, IBM_Cloud)
   50.
          IBM_Cloud_Object_Storage is provided IBM_Cloud
      FOL: \forall x (IBM\_Cloud\_Object\_Storage(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
```

```
isProvidedBy(x,y))
       DL: isProvidedBy(IBM_Cloud_Object_Storage, IBM_Cloud)
          IBM_Cloud_Kubernetes_Service is provided IBM_Cloud
       FOL: \forall x (IBM\_Cloud\_Kubernetes\_Service(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
       isProvidedBy(x,y)
       DL: isProvidedBy(IBM_Cloud_Kubernetes_Service, IBM_Cloud)
   52.
          IBM_Cloud_Database is provided IBM_Cloud
       FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
       isProvidedBy(x,y)
       DL: isProvidedBy(IBM_Cloud_Database, IBM_Cloud)
   53.
          IBM_Cloud_Load_Balancer is provided IBM_Cloud
       FOL: \forall x (IBM\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (IBM\_Cloud(y) \land \exists y (IBM\_Cloud(y)))
       isProvidedBy(x,y))
       DL: isProvidedBy(IBM_Cloud_Load_Balancer, IBM_Cloud)
          IBM_Cloud_VPC is provided IBM_Cloud
       FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (IBM\_Cloud(y) \land isProvidedBy(x,y))
       DL: isProvidedBy(IBM_Cloud_VPC, IBM_Cloud)
          IBM_Cloud_Transit_Gateway is provided IBM_Cloud
       FOL: \forall x (IBM\_Cloud\_Transit\_Gateway(x) \rightarrow \exists y (IBM\_Cloud(y) \land
       isProvidedBy(x,y))
       DL: isProvidedBy(IBM_Cloud_Transit_Gateway, IBM_Cloud)
Oracle Cloud
          Oracle_Cloud_VM is provided Oracle_Cloud
       FOL: \forall x (0racle\_Cloud\_VM(x) \rightarrow \exists y (0racle\_Cloud(y) \land \exists y (0racle\_Cloud(y)))
       isProvidedBy(x,y)
       DL: isProvidedBy(Oracle_Cloud_VM, Oracle_Cloud)
          Oracle_Cloud_Functions is provided Oracle_Cloud
       FOL: \forall x(Oracle\_Cloud\_Functions(x) \rightarrow \exists y(Oracle\_Cloud(y) \land
       isProvidedBy(x,y)
       DL: isProvidedBy(Oracle_Cloud_Functions, Oracle_Cloud)
   58. Oracle_Cloud_Block_Storage is provided Oracle_Cloud
       FOL: \forall x (0 \text{ racle\_Cloud\_Block\_Storage}(x) \rightarrow \exists y (0 \text{ racle\_Cloud}(y) \land
```

```
isProvidedBy(x,y))
                                         DL: isProvidedBy(Oracle_Cloud_Block_Storage, Oracle_Cloud)
                                                            Oracle_Cloud_Kubernetes_Engine is provided Oracle_Cloud
                                        FOL: \forall x (Oracle\_Cloud\_Kubernetes\_Engine(x) \rightarrow \exists y (Oracle\_Cloud(y) \land \exists y)
                                         isProvidedBy(x,y)
                                         DL: isProvidedBy(Oracle_Cloud_Kubernetes_Engine, Oracle_Cloud)
                     60.
                                                            Oracle_Cloud_Database_Service is provided Oracle_Cloud
                                        FOL: \forall x(Oracle\_Cloud\_Database\_Service(x) \rightarrow \exists y(Oracle\_Cloud(y) \land \exists y(Oracle\_Cloud(y)) \land 
                                         isProvidedBy(x,y))
                                         DL: isProvidedBy(Oracle_Cloud_Database_Service, Oracle_Cloud)
                     61.
                                                            Oracle_Cloud_Load_Balancer is provided Oracle_Cloud
                                        FOL: \forall x(0racle\_Cloud\_Load\_Balancer(x) \rightarrow \exists y(0racle\_Cloud(y) \land \exists y(0racle\_Cloud(y)))
                                         isProvidedBy(x,y))
                                         DL: isProvidedBy(Oracle_Cloud_Load_Balancer, Oracle_Cloud)
                                                            Oracle_Cloud_VCN is provided Oracle_Cloud
                                        FOL: \forall x (0racle\_Cloud\_VCN(x) \rightarrow \exists y (0racle\_Cloud(y) \land \exists y (0racle\_Cloud(y)))
                                         isProvidedBy(x,y))
                                         DL: isProvidedBy(Oracle_Cloud_VM, Oracle_Cloud)
Alibaba Cloud
                                                            Alibaba_ECS is provided Alibaba_Cloud
                                        FOL: \forall x(Alibaba\_ECS(x) \rightarrow \exists y(Alibaba\_Cloud(y) \land \exists y(Alibaba\_Cloud(y)) \land y(Alibaba
                                         isProvidedBy(x,y)
                                         DL: isProvidedBy(Alibaba_ECS, Alibaba_Cloud)
                                                            Alibaba_Function_Compute is provided Alibaba_Cloud
                                         FOL: \forall x(Alibaba\_Function\_Compute(x) \rightarrow \exists y(Alibaba\_Cloud(y) \land
                                         isProvidedBy(x,y))
                                         DL: isProvidedBy(Alibaba_Function_Compute, Alibaba_Cloud)
                     65. Alibaba_OSS is provided Alibaba_Cloud
                                        FOL: \forall x (Alibaba_OSS(x) \rightarrow \exists y (Alibaba_Cloud(y) \land \exists y (Alibaba_Cloud(y)) \land \exists y (Alibaba_Cloud
                                         isProvidedBy(x,y))
                                         DL: isProvidedBy(Alibaba_OSS, Alibaba_Cloud)
```

```
Alibaba_Container_Service is provided Alibaba_Cloud
                     FOL: \forall x(Alibaba\_Container\_Service(x) \rightarrow \exists y(Alibaba\_Cloud(y) \land
                     isProvidedBy(x,y))
                     DL: isProvidedBy(Alibaba_OSS, Alibaba_Cloud)
67. Alibaba_ApsaraDB is provided Alibaba_Cloud
                    FOL: \forall x(Alibaba\_ApsaraDB(x) \rightarrow \exists y(Alibaba\_Cloud(y) \land
                     isProvidedBy(x,y)
                     DL: isProvidedBy(Alibaba_ApsaraDB, Alibaba_Cloud)
                                        Alibaba_SLB is provided Alibaba_Cloud
                    FOL: \forall x (Alibaba\_SLB(x) \rightarrow \exists y (Alibaba\_Cloud(y) \land \exists y (Alibaba\_Cloud(y)) \land \exists y (AlibabaCloud(y)) \land \exists
                     isProvidedBy(x,y))
                     DL: isProvidedBy(Alibaba_SLB, Alibaba_Cloud)
69. Alibaba_VPC is provided Alibaba_Cloud
                    FOL: \forall x (Alibaba_VPC(x) \rightarrow \exists y (Alibaba_Cloud(y) \land \exists y (Alibaba_Cloud(y)) \land \exists y (Alibaba_VPC(x)) \rightarrow \exists y (Alibaba_
                     isProvidedBy(x,y)
                     DL: isProvidedBy(Alibaba_VPC, Alibaba_Cloud)
                                       Alibaba_SAG is provided Alibaba_Cloud
70.
                    FOL: \forall x (Alibaba\_SAG(x) \rightarrow \exists y (Alibaba\_Cloud(y) \land \exists y (Alibaba\_Cloud(y)) \land \exists y (AlibabaCloud(y)) \land \exists
                     isProvidedBy(x,y)
                     DL: isProvidedBy(Alibaba_SAG, Alibaba_Cloud)
71. Cloud Service is Cloud Resource
                     FOL: \forall x(Cloud\_Service(x) \rightarrow Cloud\_Resource(x))
                     DL: Cloud Serviceis ⊆ Cloud Resource
72. Virtual Machine Service is Cloud Service
                     FOL: \forall x (Virtual\_Machine\_Service(x) \rightarrow Cloud\_Service(x))
                     DL: Virtual_Machine_Service ⊆ Cloud_Resource
                                         Object_Storage_Service is Cloud_Service
                     FOL: \forall x (0bject\_Storage\_Service(x) \rightarrow Cloud\_Service(x))
                     DL: Object_Storage_Service ⊆ Cloud_Service
                                         Container Orchestration Service is Cloud Service
                     FOL: \forall x (Container\_Orchestration\_Service(x) \rightarrow Cloud\_Service(x))
                     DL: Container_Orchestration_Service ⊆ Cloud_Service
```

```
Composite Service is Cloud Service
     FOL: \forall x (Composite\_Service(x) \rightarrow Cloud\_Service(x))
     DL: Composite_Service ⊆ Cloud_Service
Cluster 3: Concrete Service Instances & Provisioning
        AWS_EC2 is an instance of Virtual_Machine_Service
  76.
     FOL: Virtual_Machine_Service(AWS_EC2)
     DL: Virtual_Machine_Service(AWS_EC2)
  77. AWS Lambda is an instance of Function as a Resource
     FOL: Function_as_a_Service(AWS_Lambda)
     DL: Function_as_a_Service(AWS_Lambda)
  78. AWS_S3 is an instance of Object_Storage_Resource
     FOL: Object_Storage_Resource(AWS_S3)
     DL: Object_Storage_Resource(AWS_S3)
  AWS VPC is an instance of Virtual Network Resource
     FOL: Virtual_Network_Resource(AWS_VPC)
     DL: Virtual_Network_Resource(AWS_VPC)
  80. ELB is an instance of Load_Balancer_Resource
     FOL: Load_Balancer_Resource(ELB)
     DL: Load_Balancer_Resource(ELB)
  81. AWS_VGW is an instance of VPN_Gateway_Resource
     FOL: VPN_Gateway_Resource(AWS_VGW)
     DL: VPN_Gateway_Resource(AWS_VGW)
        Route_53 is an instance of DNS_Resource
     FOL: DNS_Resource(Route_53)
     DL: DNS_Resource(Route_53)
```

AWS_RDS is an instance of Database_Resource

FOL: Database_Resource(AWS_RDS)

```
DL: Database_Resource(AWS_RDS)
84. AWS_RedShift is an instance of Data_Warehouse_Resource
  FOL: Data_Warehouse_Resource(AWS_RedShift)
  DL: Data_Warehouse_Resource(AWS_RedShift)
85. AWS EKS is an instance of Container Orchestration Service
  FOL: Container Orchestration Service(AWS EKS)
  DL: Container_Orchestration_Service(AWS_EKS)
86. Azure_VM is an instance of Virtual_Machine_Service
  FOL: Virtual_Machine_Service(Azure_VM)
  DL: Virtual_Machine_Service(Azure_VM)
87. Azure_VNet is an instance of Virtual_Network_Resource
  FOL: Virtual_Network_Resource(Azure_VNet)
  DL: Virtual_Network_Resource(Azure_VNet)
88. Azure_LB is an instance of Load_Balancer_Resource
  FOL: Load_Balancer_Resource(Azure_LB)
  DL: Load_Balancer_Resource(Azure_LB)
89. Azure_VPN_Gateway is an instance of VPN_Gateway_Resource
  FOL: VPN_Gateway_Resource(Azure_VPN_Gateway)
  DL: VPN_Gateway_Resource(Azure_VPN_Gateway)
90. Azure_DNS is an instance of DNS_Resource
  FOL: DNS_Resource(Azure_DNS)
  DL: DNS_Resource(Azure_DNS)
91. Azure SOL Database is an instance of Database Resource
  FOL: Database_Resource(Azure_SQL_Database)
  DL: Database_Resource(Azure_SQL_Database)
92. Azure_AKS is an instance of Container_Orchestration_Service
  FOL: Container_Orchestration_Service(Azure_AKS)
  DL: Container_Orchestration_Service(Azure_AKS)
     Azure_Scale_Sets is an instance of Auto_Scaling_Service
  FOL: Auto_Scaling_Service(Azure_Scale_Sets)
```

```
DL: Auto_Scaling_Service(Azure_Scale_Sets)
94. Azure_Functions is an instance of Function_as_a_Service
  FOL: Function_as_a_Service(Azure_Functions)
  DL: Function_as_a_Service(Azure_Functions)
95. GCP_Compute_Engine is an instance of Virtual_Machine_Service
  FOL: Virtual_Machine_Service(GCP_Compute_Engine)
  DL: Virtual_Machine_Service(GCP_Compute_Engine)
96. GCS_Bucket is an instance of Object_Storage_Resource
  FOL: Object_Storage_Resource(GCS_Bucket)
  DL: Object_Storage_Resource(GCS_Bucket)
97. GCP_GKE is an instance of Container_Orchestration_Service
  FOL: Container_Orchestration_Service(GCP_GKE)
  DL: Container_Orchestration_Service(GCP_GKE)
98.
     IBM_Cloud_Virtual_Server is an instance of
  Virtual_Machine_Service
  FOL: Virtual_Machine_Service(IBM_Cloud_Virtual_Server)
  DL: Virtual_Machine_Service(IBM_Cloud_Virtual_Server)
     IBM_Cloud_Functions is an instance of
  Serverless_Compute_Service
  FOL: Serverless_Compute_Service(IBM_Cloud_Functions)
  DL: Serverless_Compute_Service(IBM_Cloud_Functions)
100. IBM_Cloud_Object_Storage is an instance of
  Object_Storage_Resource
  FOL: Object_Storage_Resource(IBM_Cloud_Object_Storage)
  DL: Object_Storage_Resource(IBM_Cloud_Object_Storage)
101. IBM Cloud Kubernetes Service is an instance of
  Container Orchestration Service
  FOL:
  Container_Orchestration_Service(IBM_Cloud_Kubernetes_Service)
  DL: Container_Orchestration_Service(IBM_Cloud_Kubernetes_Service)
102. IBM_Cloud_Database is an instance of Database_Resource
  FOL: Database_Resource(IBM_Cloud_Database)
```

```
DL: Database_Resource(IBM_Cloud_Database)
103. IBM_Cloud_Load_Balancer is an instance of
  Load Balancer Resource
  FOL: Load_Balancer_Resource(IBM_Cloud_Load_Balancer)
  DL: Load_Balancer_Resource(IBM_Cloud_Load_Balancer)
104. IBM Cloud VPC is an instance of Virtual Network Resource
  FOL: Virtual_Network_Resource(IBM_Cloud_VPC)
  DL: Virtual_Network_Resource(IBM_Cloud_VPC)
105. IBM_Cloud_Transit_Gateway is an instance of
  VPN_Gateway_Resource
  FOL: VPN_Gateway_Resource(IBM_Cloud_Transit_Gateway)
  DL: VPN_Gateway_Resource(IBM_Cloud_Transit_Gateway)
106. Oracle_Cloud_VM is an instance of Virtual_Machine_Service
  FOL: Virtual_Machine_Service(Oracle_Cloud_VM)
  DL: Virtual_Machine_Service(Oracle_Cloud_VM)
107. Oracle_Cloud_Functions is an instance of
  Serverless_Compute_Service
  FOL: Serverless_Compute_Service(Oracle_Cloud_Functions)
  DL: Serverless_Compute_Service(Oracle_Cloud_Functions)
108. Oracle_Cloud_Block_Storage is an instance of
  Block_Storage_Resource
  FOL: Block_Storage_Resource(Oracle_Cloud_Block_Storage)
  DL: Block_Storage_Resource(Oracle_Cloud_Block_Storage)
109. Oracle_Cloud_Object_Storage is an instance of
  Object_Storage_Resource
  FOL: Object_Storage_Resource(Oracle_Cloud_Object_Storage)
  DL: Object_Storage_Resource(Oracle_Cloud_Object_Storage)
110. Oracle_Cloud_Kubernetes_Engine is an instance of
  Container Orchestration Service
  FOL:
  Container_Orchestration_Service(Oracle_Cloud_Kubernetes_Engine)
  DL:
```

```
Container_Orchestration_Service(Oracle_Cloud_Kubernetes_Engine)
111. Oracle_Cloud_Database_Service is an instance of
  Database Resource
  FOL: Database_Resource(Oracle_Cloud_Database_Service)
  DL: Database_Resource(Oracle_Cloud_Database_Service)
112. Oracle Cloud Load Balancer is an instance of
  Load Balancer Resource
  FOL: Load_Balancer_Resource(Oracle_Cloud_Load_Balancer)
  DL: Load_Balancer_Resource(Oracle_Cloud_Load_Balancer)
113. Oracle_Cloud_VCN is an instance of Virtual_Network_Resource
  FOL: Virtual_Network_Resource(Oracle_Cloud_VCN)
  DL: Virtual_Network_Resource(Oracle_Cloud_VCN)
114. Alibaba_ECS is an instance of Virtual_Machine_Service
  FOL: Virtual_Machine_Service(Alibaba_ECS)
  DL: Virtual_Machine_Service(Alibaba_ECS)
115. Alibaba_Function_Compute is an instance of
  Serverless_Compute_Service
  FOL: Serverless_Compute_Service(Alibaba_Function_Compute)
  DL: Serverless_Compute_Service(Alibaba_Function_Compute)
116. Alibaba_OSS is an instance of Object_Storage_Resource
  FOL: Object_Storage_Resource(Alibaba_OSS)
  DL: Object_Storage_Resource(Alibaba_OSS)
117. Alibaba Container Service is an instance of
  Container Orchestration Service
  FOL: Container_Orchestration_Service(Alibaba_Container_Service)
  DL: Container_Orchestration_Service(Alibaba_Container_Service)
118. Alibaba_ApsaraDB is an instance of Database_Resource
  FOL: Database_Resource(Alibaba_ApsaraDB)
  DL: Database_Resource(Alibaba_ApsaraDB)
119. Alibaba_SLB is an instance of Load_Balancer_Resource
  FOL: Load_Balancer_Resource(Alibaba_SLB)
```

```
DL: Load_Balancer_Resource(Alibaba_SLB)
  120. Alibaba_VPC is an instance of Virtual_Network_Resource
     FOL: Virtual_Network_Resource(Alibaba_VPC)
     DL: Virtual_Network_Resource(Alibaba_VPC)
  121. Alibaba_SAG is an instance of VPN_Gateway_Resource
     FOL: VPN_Gateway_Resource(Alibaba_SAG)
     DL: VPN_Gateway_Resource(Alibaba_SAG)
Cluster 4: Standardized Terms & Mappings
```

```
122. Compute_Instance is an instance of Standardized_Term
  FOL: Standardized_Term(Compute_Instance)
  DL: Standardized_Term(Compute_Instance)
123. Block_Storage is Standardized_Term
  FOL: \forall x (Block\_Storage(x) \rightarrow Standardized\_Term(x))
  DL: Block_Storage ⊆ Standardized_Term
124. Kubernetes_Service is Standardized_Term
  FOL: \forall x (Kubernetes\_Service(x) \rightarrow Standardized\_Term(x))
  DL: Kubernetes Service ⊆ Standardized Term
125. gRPC is an instance of Interoperability_Mapping
  FOL: Interoperability_Mapping(gRPC)
  DL: Interoperability_Mapping(gRPC)
126. OCCI is an instance of Interoperability_Mapping
  FOL: Interoperability_Mapping(OCCI)
  DL: Interoperability_Mapping(OCCI)
127. Open_API is an instance of Interoperability_Mapping
  FOL: Interoperability_Mapping(Open_API)
  DL: Interoperability_Mapping(Open_API)
```

```
128. Mapping_ObjectStorage_to_S3 is an instance of
  Standardized_Term
  FOL: Standardized_Term(Mapping_ObjectStorage_to_S3)
  DL: Standardized_Term(Mapping_ObjectStorage_to_S3)
129. Mapping_ComputeInstance_to_EC2 is an instance of
  Standardized Term
  FOL: Standardized_Term(Mapping_ComputeInstance_to_EC2)
  DL: Standardized_Term(Mapping_ComputeInstance_to_EC2)
130. Open_API is an instance of API_Standard
  FOL: API_Standard(Open_API)
  DL: API_Standard(Open_API)
131. gRPC is an instance of API_Standard
  FOL: API_Standard(gRPC)
  DL: API_Standard(gRPC)
132. OCCI is an instance of API_Standard
  FOL: API_Standard(OCCI)
  DL: API_Standard(OCCI)
```

Cluster 5: Performance Metrics

133. CPU_Utilization is an instance of Performance_Metric
 FOL: Performance_Metric(CPU_Utilization)
 DL: Performance_Metric(CPU_Utilization)

134. RAM_Usage is an instance of Performance_Metric
 FOL: Performance_Metric(RAM_Usage)
 DL: Performance_Metric(RAM_Usage)

135. Network_Latency is an instance of Performance_Metric
 FOL: Performance_Metric(Network_Latency)
 DL: Performance_Metric(Network_Latency)

```
136. Storage_Latency is an instance of Performance_Metric
  FOL: Performance_Metric(Storage_Latency)
  DL: Performance_Metric(Storage_Latency)
137. Cold_Start_Latency is an instance of Performance_Metric
  FOL: Performance_Metric(Cold_Start_Latency)
  DL: Performance_Metric(Cold_Start_Latency)
138. Data_Throughput is an instance of Performance_Metric
  FOL: Performance_Metric(Data_Throughput)
  DL: Performance_Metric(Data_Throughput)
139. Memory_Consumption is an instance of Performance_Metric
  FOL: Performance_Metric(Memory_Consumption)
  DL: Performance_Metric(Memory_Consumption)
140. Network_Throughput is an instance of Performance_Metric
  FOL: Performance_Metric(Network_Throughput)
  DL: Performance_Metric(Network_Throughput)
141. Request_Duration is an instance of Performance_Metric
  FOL: Performance_Metric(Request_Duration)
  DL: Performance_Metric(Request_Duration)
142. Query_Response_Time is an instance of Performance_Metric
  FOL: Performance_Metric(Query_Response_Time)
  DL: Performance_Metric(Query_Response_Time)
143. Disk_IOPS is an instance of Performance_Metric
  FOL: Performance_Metric(Disk_IOPS)
  DL: Performance_Metric(Disk_IOPS)
```

Cluster 6: Cost Models

144. PayAsYouGo_Pricing is an instance of Cost_Model
FOL: Cost_Model(PayAsYouGo_Pricing)

```
DL: Cost_Model(PayAsYouGo_Pricing)
145. Subscription_Based is an instance of Cost_Model
  FOL: Cost_Model(Subscription_Based)
  DL: Cost_Model(Subscription_Based)
146. Reserved_Instance_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Reserved_Instance_Pricing)
  DL: Cost_Model(Reserved_Instance_Pricing)
147. Spot_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Spot_Pricing)
  DL: Cost_Model(Spot_Pricing)
148. vCore_Pricing is an instance of Cost_Model
  FOL: Cost_Model(vCore_Pricing)
  DL: Cost_Model(vCore_Pricing)
149. Pay_per_Execution is an instance of Cost_Model
  FOL: Cost_Model(Pay_per_Execution)
  DL: Cost_Model(Pay_per_Execution)
150. OnDemand_Pricing is an instance of Cost_Model
  FOL: Cost_Model(OnDemand_Pricing)
  DL: Cost_Model(OnDemand_Pricing)
151. Reserved_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Reserved_Pricing)
  DL: Cost_Model(Reserved_Pricing)
152. Preemptible_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Preemptible_Pricing)
  DL: Cost_Model(Preemptible_Pricing)
153. Storage_Tier_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Storage_Tier_Pricing)
  DL: Cost_Model(Storage_Tier_Pricing)
154. Data_Transfer_Pricing is an instance of Cost_Model
  FOL: Cost_Model(Data_Transfer_Pricing)
```

```
DL: Cost_Model(Data_Transfer_Pricing)
155. Request_Pricing is an instance of Cost_Model
                          FOL: Cost_Model(Request_Pricing)
                          DL: Cost_Model(Request_Pricing)
156. AWS_EC2 has a cost model OnDemand_Pricing
                          FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land (OnDemand\_Pricing(y)) 
                        hasCostModel(x,y))
                          DL: hasCostModel(AWS_EC2, OnDemand_Pricing)
157. AWS_EC2 has a cost model Reserved_Pricing
                          FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (Reserved\_Pricing(y) \land \exists y (Reserved\_Pricing(y)) \land \exists y (Reserved\_
                        hasCostModel(x,y))
                        DL: hasCostModel(AWS_EC2, Reserved_Pricing)
158. AWS_Lambda has a cost model PayAsYouGo_Pricing
                        FOL: \forall x (AWS\_Lambda(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y)) \land \exists y (PayA
                        hasCostModel(x,y))
                          DL: hasCostModel(AWS_Lambda, PayAsYouGo_Pricing)
159. AWS_S3 has a cost model PayAsYouGo_Pricing
                          FOL: \forall x (AWS_S3(x) \rightarrow \exists y (PayAsYouGo_Pricing(y) \land \exists y (PayAsYouGo_Pricing(y)) \land \exists y (PayAsYouGo_P
                        hasCostModel(x,y))
                          DL: hasCostModel(AWS_S3, PayAsYouGo_Pricing)
160. AWS_S3 has a cost model Storage_Tier_Pricing
                          FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Storage_Tier_Pricing(y)) \land
                        hasCostModel(x,y))
                          DL: hasCostModel(AWS_S3, Storage_Tier_Pricing)
161. AWS_VPC has a cost model Data_Transfer_Pricing
                          FOL: \forall x (AWS_VPC(x) \rightarrow \exists y (Data_Transfer_Pricing(y)) \land
                          hasCostModel(x,y))
                          DL: hasCostModel(AWS_VPC, Data_Transfer_Pricing)
162. ELB has a cost model PayAsYouGo_Pricing
                          FOL: \forall x (ELB(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land hasCostModel(x,y))
                          DL: hasCostModel(ELB, PayAsYouGo_Pricing)
```

```
163. AWS_VGW has a cost model Data_Transfer_Pricing
                                 FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (Data_Transfer_Pricing(y)) \land
                                hasCostModel(x,y)))
                                 DL: hasCostModel(AWS_VGW, Data_Transfer_Pricing)
164. Route_53 has a cost model Request_Pricing
                                 FOL: \forall x (Route_{53}(x) \rightarrow \exists y (Request_{Pricing}(y) \land \exists z (Request_{Pricing}(y) \land \exists 
                                 hasCostModel(x,y))
                                 DL: hasCostModel(Route_53, Request_Pricing)
165. AWS_RDS has a cost model PayAsYouGo_Pricing
                                FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (PayAsYouGo_Pricing(y) \land \exists y (PayAsYouGo_Pricing(y)) \land \exists y (PayAsYouGo_Pricing(y) \land \exists y (PayAsYouGo_Pricing(y)) \land \exists y (PayAsYouGo_Pri
                                hasCostModel(x,y))
                                 DL: hasCostModel(AWS_RDS, PayAsYouGo_Pricing)
166. AWS_RDS has a cost model Reserved_Pricing
                                 FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (Reserved_Pricing(y)) \land
                                hasCostModel(x,y))
                                DL: hasCostModel(AWS_RDS, Reserved_Pricing)
167. AWS_RedShift has a cost model OnDemand_Pricing
                                FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land \exists y (
                                hasCostModel(x,y))
                                DL: hasCostModel(AWS_RedShift, OnDemand_Pricing)
168. AWS_EKS has a cost model PayAsYouGo_Pricing
                                 FOL: \forall x (AWS_EKS(x) \rightarrow \exists y (PayAsYouGo_Pricing(y) \land Australian A
                                hasCostModel(x,y))
                                 DL: hasCostModel(AWS_EKS, PayAsYouGo_Pricing)
169. Azure_VM has a cost model Spot_Pricing
                                 FOL: \forall x(Azure_VM(x) \rightarrow \exists y(Spot_Pricing(y) \land hasCostModel(x,y)))
                                 DL: hasCostModel(Azure_VM, Spot_Pricing)
170. Azure_VM has a cost model Reserved_Pricing
                                 FOL: \forall x (Azure_VM(x) \rightarrow \exists y (Reserved_Pricing(y) \land \exists y (Reserved_Pricing(y)) \land \exists y (Reserved_Pricing(y) \land \exists y (Reserved_Pricing(y)) \land \exists y (Reserved_P
                                 hasCostModel(x,y))
                                 DL: hasCostModel(Azure_VM, Reserved_Pricing)
171. Azure_VNet has a cost model Data_Transfer_Pricing
                                 FOL: \forall x(Azure\_VNet(x) \rightarrow \exists y(Data\_Transfer\_Pricing(y) \land \exists y(Data\_Transfer\_Pricing(y)) \land \exists y(
```

```
hasCostModel(x,y))
                                  DL: hasCostModel(Azure_VNet, Data_Transfer_Pricing)
172. Azure_LB has a cost model PayAsYouGo_Pricing
                                  FOL: \forall x (Azure\_LB(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y)) \land
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Azure_LB, PayAsYouGo_Pricing)
173. Azure_VPN_Gateway has a cost model Data_Transfer_Pricing
                                FOL: \forall x (Azure_VPN_Gateway(x) \rightarrow \exists y (Data_Transfer_Pricing(y)) \land
                                hasCostModel(x,y)))
                                  DL: hasCostModel(Azure_VPN_Gateway, Data_Transfer_Pricing)
174. Azure_DNS has a cost model Request_Pricing
                                FOL: \forall x (Azure\_DNS(x) \rightarrow \exists y (Request\_Pricing(y) \land \exists y (Request\_Pricing(y)) \land \exists y (Request\_Pricing(y) \land \exists y (Request\_Pricing(y)) \land \exists y (R
                                hasCostModel(x,y))
                                  DL: hasCostModel(Azure_DNS, Request_Pricing)
175. Azure_SQL_Database has a cost model PayAsYouGo_Pricing
                                FOL: \forall x(Azure\_SQL\_Database(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land \exists y(PayAsYouGo\_Pricing(y)) \land \exists y(PayAsYo
                                hasCostModel(x,y))
                                  DL: hasCostModel(Azure_SQL_Database, PayAsYouGo_Pricing)
176. Azure_SQL_Database has a cost model Reserved_Pricing
                                  FOL: \forall x(Azure\_SQL\_Database(x) \rightarrow \exists y(Reserved\_Pricing(y) \land \exists y(Reserved\_Pricing(y)) \land \exists y(Reserved\_P
                                hasCostModel(x,y))
                                  DL: hasCostModel(Azure_SQL_Database, Reserved_Pricing)
177. Azure_AKS has a cost model PayAsYouGo_Pricing
                                  FOL: \forall x(Azure\_AKS(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land \exists y(PayAsYouGo\_Pricing(y)) \land \exists y(PayAsYouGo\_Prici
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Azure_AKS, PayAsYouGo_Pricing)
178. Azure_Scale_Sets has a cost model Reserved_Pricing
                                  FOL: \forall x(Azure\_Scale\_Sets(x) \rightarrow \exists y(Reserved\_Pricing(y) \land \exists y(Reserved\_Pricing(y)) \land \exists y(Reserved\_Pri
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Azure_Scale_Sets, Reserved_Pricing)
179. Azure_Functions has a cost model PayAsYouGo_Pricing
                                  FOL: \forall x(Azure\_Functions(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land \exists y(PayAsYouGo\_Pricing(y)) \land \exists y(PayAsYouGo
                                  hasCostModel(x,y))
```

```
DL: hasCostModel(Azure_Functions, PayAsYouGo_Pricing)
180. GCP_Compute_Engine has a cost model OnDemand_Pricing
                                     FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land \exists y (OnDemand\_Pricing(y
                                     hasCostModel(x,y))
                                     DL: hasCostModel(GCP_Compute_Engine, OnDemand_Pricing)
181. GCP_Compute_Engine has a cost model Preemptible_Pricing
                                   FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (Preemptible\_Pricing(y) \land \exists y (Preemptible\_Pricing(y)) \land \exists y (Preemptible\_Pricing(y)
                                   hasCostModel(x,y))
                                     DL: hasCostModel(GCP_Compute_Engine, Preemptible_Pricing)
182. GCS_Bucket has a cost model Storage_Tier_Pricing
                                   FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Storage\_Tier\_Pricing(y) \land \exists y (Storage\_Tier\_Pricing(y)) \land \exists y
                                   hasCostModel(x,y))
                                   DL: hasCostModel(GCS_Bucket, Storage_Tier_Pricing)
183. GCS_Bucket has a cost model PayAsYouGo_Pricing
                                     FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y)) \land \exists y (PayA
                                   hasCostModel(x,y))
                                     DL: hasCostModel(GCS_Bucket, PayAsYouGo_Pricing)
184. GCP_GKE has a cost model PayAsYouGo_Pricing
                                     FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y)) \land \exists y (PayAsYouG
                                   hasCostModel(x,y))
                                     DL: hasCostModel(GCP_GKE, PayAsYouGo_Pricing)
185. IBM_Cloud_Virtual_Server has a cost model OnDemand_Pricing
                                     FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land (OnDemand
                                   hasCostModel(x,y))
                                     DL: hasCostModel(IBM_Cloud_Virtual_Server, OnDemand_Pricing)
186. IBM_Cloud_Virtual_Server has a cost model Reserved_Pricing
                                     FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (Reserved\_Pricing(y) \land \exists y (Reserved\_Pricing(y)) \land \exists y (Reserved\_Pri
                                     hasCostModel(x,y))
                                     DL: hasCostModel(IBM_Cloud_Virtual_Server, Reserved_Pricing)
187. IBM_Cloud_Functions has a cost model PayAsYouGo_Pricing
                                     FOL: \forall x (IBM\_Cloud\_Functions(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y)) \land
                                     hasCostModel(x,y))
```

```
DL: hasCostModel(IBM_Cloud_Functions, PayAsYouGo_Pricing)
188. IBM_Cloud_Object_Storage has a cost model Storage_Tier_Pricing
                      FOL: \forall x (IBM\_Cloud\_Object\_Storage(x) \rightarrow \exists y (Storage\_Tier\_Pricing(y))
                     ∧ hasCostModel(x,y)))
                      DL: hasCostModel(IBM_Cloud_Object_Storage, Storage_Tier_Pricing)
189. IBM Cloud Kubernetes Service has a cost model
                      PayAsYouGo_Pricing
                      FOL: ∀x(IBM_Cloud_Kubernetes_Service(x) →
                       \exists y (PayAsYouGo\_Pricing(y) \land hasCostModel(x,y)))
                      DL: hasCostModel(IBM_Cloud_Kubernetes_Service,
                      PayAsYouGo_Pricing)
190. IBM_Cloud_Database has a cost model PayAsYouGo_Pricing
                      FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y)) \land \exists
                     hasCostModel(x,y))
                      DL: hasCostModel(IBM_Cloud_Database, PayAsYouGo_Pricing)
191. IBM_Cloud_Database has a cost model Reserved_Pricing
                      FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (Reserved\_Pricing(y) \land \exists y (Reserved\_Pricing(y)) \land \exists y (Reserved\_Pricing(y
                     hasCostModel(x,y))
                      DL: hasCostModel(IBM_Cloud_Database, Reserved_Pricing)
192. IBM_Cloud_Load_Balancer has a cost model PayAsYouGo_Pricing
                      FOL: \forall x (IBM\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y) \land \exists y (PayAsYouGo\_Pricing(y)) \land \exists y (PayAsYouGo\_Pricing(y)
                      hasCostModel(x,y))
                      DL: hasCostModel(IBM_Cloud_Load_Balancer, PayAsYouGo_Pricing)
193. IBM_Cloud_VPC has a cost model Data_Transfer_Pricing
                      FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (Data\_Transfer\_Pricing(y) \land \exists y (Data\_Transfer\_Pricing(y)) \land \exists y
                      hasCostModel(x,y))
                      DL: hasCostModel(IBM_Cloud_VPC, Data_Transfer_Pricing)
194. IBM_Cloud_Transit_Gateway has a cost model
                      Data_Transfer_Pricing
                      FOL: \forall x(IBM\_Cloud\_Transit\_Gateway(x) \rightarrow
                       \exists y (Data\_Transfer\_Pricing(y) \land hasCostModel(x,y)))
                      DL: hasCostModel(IBM_Cloud_Transit_Gateway,
                      Data_Transfer_Pricing)
```

```
195. Oracle Cloud VM has a cost model Reserved Pricing
                   FOL: \forall x (Oracle\_Cloud\_VM(x) \rightarrow \exists y (Reserved\_Pricing(y) \land \exists y (Reserved\_Pricing(y)) \land \exists 
                   hasCostModel(x,y))
                   DL: hasCostModel(Oracle_Cloud_VM, Reserved_Pricing)
196. Oracle_Cloud_VM has a cost model OnDemand_Pricing
                   FOL: \forall x (Oracle\_Cloud\_VM(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land (OnDemand\_Pric
                   hasCostModel(x,y))
                   DL: OhasCostModel(Oracle_Cloud_VM, OnDemand_Pricing)
197. Oracle_Cloud_Functions has a cost model PayAsYouGo_Pricing
                   FOL: \forall x(0racle\_Cloud\_Functions(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land \exists y(PayAsYouGo\_Pricing(y)) \land \exists y(Pay
                  hasCostModel(x,y))
                   DL: hasCostModel(Oracle_Cloud_Functions, PayAsYouGo_Pricing)
198. Oracle_Cloud_Block_Storage has a cost model
                   Storage_Tier_Pricing
                   FOL: ∀x(Oracle_Cloud_Block_Storage(x) →
                    ∃y(Storage_Tier_Pricing(y) ∧ hasCostModel(x,y)))
                   DL: OhasCostModel(Oracle_Cloud_Block_Storage,
                   Storage_Tier_Pricing)
199. Oracle_Cloud_Object_Storage has a cost model
                   Storage_Tier_Pricing
                   FOL: ∀x(Oracle_Cloud_Object_Storage(x) →
                    ∃y(Storage_Tier_Pricing(y) ∧ hasCostModel(x,y)))
                   DL: hasCostModel(Oracle_Cloud_Object_Storage,
                   Storage_Tier_Pricing)
200. Oracle_Cloud_Kubernetes_Engine has a cost model
                   PayAsYouGo_Pricing
                   FOL: \forall x (Oracle\_Cloud\_Kubernetes\_Engine(x) \rightarrow
                     \exists y (PayAsYouGo\_Pricing(y) \land hasCostModel(x,y)))
                   DL: hasCostModel(Oracle_Cloud_Kubernetes_Engine,
                   PayAsYouGo_Pricing)
201. Oracle_Cloud_Database_Service has a cost model
                   OnDemand_Pricing
                   FOL: ∀x(Oracle_Cloud_Database_Service(x) →
                     \exists y (OnDemand\_Pricing(y) \land hasCostModel(x,y)))
```

```
DL: hasCostModel(Oracle_Cloud_Database_Service, OnDemand_Pricing)
 202. Oracle_Cloud_Database_Service has a cost model
                           Reserved_Pricing
                           FOL: \forall x(Oracle\_Cloud\_Database\_Service(x) \rightarrow
                             \exists y (Reserved\_Pricing(y) \land hasCostModel(x,y)))
                           DL: hasCostModel(Oracle_Cloud_Database_Service, Reserved_Pricing)
 203. Oracle_Cloud_Load_Balancer has a cost model PayAsYouGo_Pricing
                           FOL: \forall x (Oracle\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y))
                          ∧ hasCostModel(x,y)))
                           DL: hasCostModel(Oracle_Cloud_Load_Balancer, PayAsYouGo_Pricing)
 204. Oracle_Cloud_VCN has a cost model Data_Transfer_Pricing
                           FOL: \forall x(Oracle\_Cloud\_VCN(x) \rightarrow \exists y(Data\_Transfer\_Pricing(y) \land \exists y(Data\_Transfer\_Pricing(y)) 
                          hasCostModel(x,y))
                           DL: hasCostModel(Oracle_Cloud_VCN, OnDemand_Pricing)
 205. Alibaba_ECS has a cost model OnDemand_Pricing
                           FOL: \forall x (Alibaba\_ECS(x) \rightarrow \exists y (OnDemand\_Pricing(y) \land \exists y (OnDemand\_Pricing(y)) \land \exists 
                          hasCostModel(x,y))
                           DL: hasCostModel(Alibaba_ECS, OnDemand_Pricing)
 206. Alibaba_ECS has a cost model Spot_Pricing
                           FOL: \forall x (Alibaba\_ECS(x) \rightarrow \exists y (Spot\_Pricing(y) \land \exists y (Spot\_Pricing(y)) \land \exists 
                           hasCostModel(x,y))
                           DL: hasCostModel(Alibaba_ECS, Spot_Pricing)
 207. Alibaba_Function_Compute has a cost model PayAsYouGo_Pricing
                           FOL: \forall x (Alibaba_Function_Compute(x) \rightarrow \exists y (PayAsYouGo_Pricing(y) \land \exists y (PayAsYouGo_Pricing(y)) \land \exists y (PayAsYouGo_Pricing(y
                           hasCostModel(x,y))
                           DL: hasCostModel(Alibaba_Function_Compute, PayAsYouGo_Pricing)
208. Alibaba_OSS has a cost model Storage_Tier_Pricing
                           FOL: \forall x (Alibaba_OSS(x) \rightarrow \exists y (Storage_Tier_Pricing(y) \land \exists y (Storage_Tier_Pricing(y)) \land \exists 
                           hasCostModel(x,y))
                           DL: hasCostModel(Alibaba_OSS, Storage_Tier_Pricing)
 209. Alibaba_Container_Service has a cost model PayAsYouGo_Pricing
                           FOL: \forall x (Alibaba\_Container\_Service(x) \rightarrow \exists y (PayAsYouGo\_Pricing(y))
                          ∧ hasCostModel(x,y)))
```

```
DL: hasCostModel(Alibaba_Container_Service, PayAsYouGo_Pricing)
210. Alibaba_ApsaraDB has a cost model PayAsYouGo_Pricing
                                  FOL: \forall x(Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y)) \land Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y)) \land Alibaba\_ApsaraDB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y)) \land Alibaba\_ApsaraDB(x) \rightarrow AlibabaApsaraDB(x) \rightarrow Aliba
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Alibaba_ApsaraDB, PayAsYouGo_Pricing)
211. Alibaba_ApsaraDB has a cost model Reserved_Pricing
                                FOL: \forall x (Alibaba\_ApsaraDB(x) \rightarrow \exists y (Reserved\_Pricing(y) \land \exists y (Reserved\_Pricing(y)) \land \exists y (Reserved\_Pricing(y))
                                hasCostModel(x,y))
                                  DL: hasCostModel(Alibaba_ApsaraDB, Reserved_Pricing)
212. Alibaba_SLB has a cost model PayAsYouGo_Pricing
                                FOL: \forall x(Alibaba\_SLB(x) \rightarrow \exists y(PayAsYouGo\_Pricing(y) \land Alibaba\_SLB(x))
                                hasCostModel(x,y))
                                  DL: hasCostModel(Alibaba_SLB, PayAsYouGo_Pricing)
213. Alibaba_VPC has a cost model Data_Transfer_Pricing
                                  FOL: \forall x (Alibaba\_VPC(x) \rightarrow \exists y (Data\_Transfer\_Pricing(y) \land \exists y (Data\_Transfer\_Pricing(y)) \land \exists y (
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Alibaba_VPC, Data_Transfer_Pricing)
214. Alibaba_SAG has cost a model Data_Transfer_Pricing
                                  FOL: \forall x (Alibaba\_SAG(x) \rightarrow \exists y (Data\_Transfer\_Pricing(y) \land \exists y (Data\_Transfer\_Pricing(y)) \land \exists y (
                                  hasCostModel(x,y))
                                  DL: hasCostModel(Alibaba_SAG, Data_Transfer_Pricing)
```

Cluster 7: Security & Compliance Attributes

```
215. Security_Feature is a Security_and_Compliance_Attribute
   FOL: ∀x(Security_Feature(x) →
    Security_and_Compliance_Attribute(x))
   DL: Security_Feature ⊆ Security_and_Compliance_Attribute

216. Compliance_Feature is a Security_and_Compliance_Attribute
   FOL: ∀x(Compliance_Feature(x) →
    Security_and_Compliance_Attribute(x))
```

```
DL: Compliance_Feature ⊆ Security_and_Compliance_Attribute
217. Encryption_At_Rest is an instance of Security_Feature
  FOL: Security_Feature(Encryption_At_Rest)
  DL: Security_Feature(Encryption_At_Rest)
218. IAM_Role_Execution is an instance of Security_Feature
  FOL: Security_Feature(IAM_Role_Execution)
  DL: Security_Feature(IAM_Role_Execution)
219. Security_Groups is an instance of Security_Feature
  FOL: Security_Feature(Security_Groups)
  DL: Security_Feature(Security_Groups)
220. Key_Pair_Authentication is an instance of Security_Feature
  FOL: Security_Feature(Key_Pair_Authentication)
  DL: Security_Feature(Key_Pair_Authentication)
221. Role_Based_Access_Control is an instance of Security_Feature
  FOL: Security_Feature(Role_Based_Access_Control)
  DL: Security_Feature(Role_Based_Access_Control)
222. Network_Access_Control_List is an instance of Security_Feature
  FOL: Security_Feature(Network_Access_Control_List)
  DL: Security_Feature(Network_Access_Control_List)
223. SSL_Offloading is an instance of Security_Feature
  FOL: Security_Feature(SSL_Offloading)
  DL: Security_Feature(SSL_Offloading)
224. VPN_Tunneling is an instance of Security_Feature
  FOL: Security_Feature(VPN_Tunneling)
  DL: Security_Feature(VPN_Tunneling)
225. DNSSEC is an instance of Security_Feature
  FOL: Security_Feature(DNSSEC)
  DL: Security_Feature(DNSSEC)
226. Bucket_Lifecycle_Management is an instance of Security_Feature
  FOL: Security_Feature(Bucket_Lifecycle_Management)
```

```
DL: Security_Feature(Bucket_Lifecycle_Management)
227. Managed_Identity is an instance of Security_Feature
  FOL: Security_Feature(Managed_Identity)
  DL: Security_Feature(Managed_Identity)
228. Azure_Security_Center_Protection is an instance of
  Security_Feature
  FOL: Security_Feature(Azure_Security_Center_Protection)
  DL: Security_Feature(Azure_Security_Center_Protection)
229. Azure_Disk_Encryption is an instance of Security_Feature
  FOL: Security_Feature(Azure_Disk_Encryption)
  DL: Security_Feature(Azure_Disk_Encryption)
230. Network_Security_Group is an instance of Security_Feature
  FOL: Security_Feature(Network_Security_Group)
  DL: Security_Feature(Network_Security_Group)
231. Identity_and_Access_Management is an instance of
  Security_Feature
  FOL: Security_Feature(Identity_and_Access_Management)
  DL: Security_Feature(Identity_and_Access_Management)
232. Transparent_Data_Encryption is an instance of Security_Feature
  FOL: Security_Feature(Transparent_Data_Encryption)
  DL: Security_Feature(Transparent_Data_Encryption)
233. Data_Encryption is an instance of Compliance_Feature
  FOL: Compliance_Feature(Data_Encryption)
  DL: Compliance_Feature(Data_Encryption)
234. Access_Control is an instance of Compliance_Feature
  FOL: Compliance_Feature(Access_Control)
  DL: Compliance_Feature(Access_Control)
235. IAM_Access_Control is an instance of Compliance_Feature
  FOL: Compliance_Feature(IAM_Access_Control)
  DL: Compliance_Feature(IAM_Access_Control)
```

```
236. Regulatory_Compliance is an instance of Compliance_Feature
                           FOL: Compliance_Feature(Regulatory_Compliance)
                           DL: Compliance_Feature(Regulatory_Compliance)
237. AWS_S3 has a security feature Encryption_At_Rest
                           FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Encryption_At_Rest(y) \land \exists z (Encryption_At_Re
                           hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_S3, Encryption_At_Rest)
238. AWS_Lambda has a security feature IAM_Role_Execution
                           FOL: \forall x (AWS\_Lambda(x) \rightarrow \exists y (IAM\_Role\_Execution(y) \land \exists y (IAM\_Role\_Execution(y)) \land \exists y (IAM\_
                         hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_Lambda, IAM_Role_Execution)
239. AWS_EC2 has a security feature Security_Groups
                           FOL: \forall x (AWS_EC2(x) \rightarrow \exists y (Security_Groups(y) \land \exists z (Security_Group
                         hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_EC2, Security_Groups)
240. AWS_EC2 has a security feature Key_Pair_Authentication
                           FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (Key\_Pair\_Authentication(y) \land \exists y (Key\_Pair\_Authentication(y)) \land \exists y (Key\_Pair\_Authenticatio
                           hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_EC2, Key_Pair_Authentication)
241. AWS_EKS has a security feature Role_Based_Access_Control
                           FOL: \forall x (AWS_EKS(x) \rightarrow \exists y (Role_Based_Access_Control(y)) \land
                           hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_EKS, Role_Based_Access_Control)
242. AWS_VPC has a security feature Network_Access_Control_List
                           FOL: \forall x (AWS_VPC(x) \rightarrow \exists y (Network_Access_Control_List(y) \land Access_Control_List(y))
                           hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(AWS_VPC, Network_Access_Control_List)
243. ELB has a security feature SSL_Offloading
                           FOL: \forall x (ELB(x) \rightarrow \exists y (SSL\_0ffloading(y) \land hasSecurityFeature(x,y))
                           DL: hasSecurityFeature(ELB, SSL_Offloading)
244. AWS_VGW has a security feature VPN_Tunneling
                           FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (VPN_Tunneling(y) \land \exists y (VPN_Tunneling(y)) \land \forall y (VPN_Tunneling(y)) \land \forall y (VPN_Tunneling(y))
                           hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(AWS_VGW, VPN_Tunneling)
245. Route_53 has a security feature DNSSEC
                  FOL: \forall x (Route_{53}(x) \rightarrow \exists y (DNSSEC(y) \land hasSecurityFeature(x,y))
                  DL: hasSecurityFeature(Route_53 , DNSSEC)
246. AWS_RedShift has a security feature Encryption_At_Rest
                  FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (Encryption\_At\_Rest(y) \land \exists y (Encryption\_At\_Rest(y)) \land \exists y (En
                 hasSecurityFeature(x,y))
                  DL: hasSecurityFeature(AWS_RedShift, Encryption_At_Rest)
247. AWS_RDS has a security feature Encryption_At_Rest
                  FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (Encryption_At_Rest(y)) \land
                 hasSecurityFeature(x,y))
                 DL: hasSecurityFeature(AWS_RDS, Encryption_At_Rest)
248. Azure_VM has a security feature
                 Azure_Security_Center_Protection
                  FOL: \forall x(Azure\_VM(x) \rightarrow \exists y(Azure\_Security\_Center\_Protection(y) \land
                 hasSecurityFeature(x,y))
                 DL: hasSecurityFeature(Azure_VM,
                 Azure_Security_Center_Protection)
249. Azure_VM has a security feature Azure_Disk_Encryption
                  FOL: \forall x(Azure_VM(x) \rightarrow \exists y(Azure_Disk_Encryption(y) \land \exists y(Az
                 hasSecurityFeature(x,y))
                 DL: hasSecurityFeature(Azure_VM, Azure_Disk_Encryption)
250. Azure_VNet has a security feature Network_Security_Group
                  FOL: \forall x(Azure\_VNet(x) \rightarrow \exists y(Network\_Security\_Group(y) \land \exists y(Network\_Security\_Group(y)) \land \exists y(Network\_Group(y)) \land \exists y(Network\_Group(y
                  hasSecurityFeature(x,y))
                  DL: hasSecurityFeature(Azure_VNet, Network_Security_Group)
251. Azure_LB has a security feature SSL_Offloading
                  FOL: \forall x (Azure\_LB(x) \rightarrow \exists y (SSL\_Offloading(y)) \land
                  hasSecurityFeature(x,y))
                  DL: hasSecurityFeature(Azure_LB, SSL_Offloading)
252. Azure_VPN_Gateway has a security feature VPN_Tunneling
                  FOL: \forall x (Azure_VPN_Gateway(x) \rightarrow \exists y (VPN_Tunneling(y)) \land
                  hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(Azure_VPN_Gateway, VPN_Tunneling)
253. Azure_DNS has a security feature DNSSEC
                   FOL: \forall x (Azure\_DNS(x) \rightarrow \exists y (DNSSEC(y) \land hasSecurityFeature(x,y))
                   DL: hasSecurityFeature(Azure_DNS, DNSSEC)
254. Azure_SQL_Database has a security feature
                   Transparent_Data_Encryption
                   FOL: ∀x(Azure_SQL_Database(x) →
                    \exists y (Transparent\_Data\_Encryption(y) \land hasSecurityFeature(x,y))
                   DL: hasSecurityFeature(Azure_SQL_Database,
                   Transparent_Data_Encryption)
255. Azure_AKS has a security feature
                  Azure_Kubernetes_Security_Policies
                  FOL: ∀x(Azure_AKS(x) → ∃y(Azure_Kubernetes_Security_Policies(y)
                  ∧ hasSecurityFeature(x,y))
                  DL: hasSecurityFeature(Azure_AKS,
                  Azure_Kubernetes_Security_Policies)
256. Azure_Scale_Sets has a security feature Network_Security_Group
                   FOL: \forall x(Azure\_Scale\_Sets(x) \rightarrow \exists y(Network\_Security\_Group(y) \land \exists y(Network\_Security\_Group(y)) \land \exists
                   hasSecurityFeature(x,y))
                   DL: hasSecurityFeature(Azure_Scale_Sets, Network_Security_Group)
257. Azure_Functions has a security feature Managed_Identity
                   FOL: \forall x (Azure\_Functions(x) \rightarrow \exists y (Managed\_Identity(y) \land \exists y (Managed\_Identity(y)) 
                   hasSecurityFeature(x,y))
                   DL: hasSecurityFeature(Azure_Functions, Managed_Identity)
258. GCS_Bucket has a security feature
                   Identity_and_Access_Management
                   FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Identity\_and\_Access\_Management(y)) \land
                  hasSecurityFeature(x,y))
                   DL: hasSecurityFeature(GCS_Bucket,
                   Identity_and_Access_Management)
259. GCP_Compute_Engine has a security feature Security_Groups
                   FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (Security\_Groups(y) \land \exists y (Security\_Groups(y)) \land \exists y (Security
                   hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(GCP_Compute_Engine, Security_Groups)
260. GCP_Compute_Engine has a security feature
            Key_Pair_Authentication
            FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (Key\_Pair\_Authentication(y) \land
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(GCP_Compute_Engine,
            Key_Pair_Authentication)
261. GCP_GKE has a security feature Role_Based_Access_Control
            FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (Role\_Based\_Access\_Control(y) \land \exists y (Role\_Based\_Access\_Control(y)) \land \exists y (Role\_Based\_Access\_C
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(GCP_GKE, Role_Based_Access_Control)
262. GCS_Bucket has a security feature Bucket_Lifecycle_Management
            FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Bucket\_Lifecycle\_Management(y)) \land
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(GCS_Bucket, Bucket_Lifecycle_Management)
263. IBM_Cloud_Virtual_Server has a security feature
            Security_Groups
            FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (Security\_Groups(y) \land \exists y (Security\_Groups(y)) \land \exists y (Se
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(IBM_Cloud_Virtual_Server, Security_Groups)
264. IBM_Cloud_Functions has a security feature IAM_Role_Execution
            FOL: \forall x (IBM\_Cloud\_Functions(x) \rightarrow \exists y (IAM\_Role\_Execution(y)) \land
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(IBM_Cloud_Functions, IAM_Role_Execution)
265. IBM_Cloud_Object_Storage has a security feature
            Encryption_At_Rest
            FOL: \forall x (IBM\_Cloud\_Object\_Storage(x) \rightarrow \exists y (Encryption\_At\_Rest(y) \land
            hasSecurityFeature(x,y))
            DL: hasSecurityFeature(IBM_Cloud_Object_Storage,
            Encryption_At_Rest)
266. IBM_Cloud_Kubernetes_Service has a security feature
            Role_Based_Access_Control
            FOL: ∀x(IBM_Cloud_Kubernetes_Service(x) →
              \exists y(Role\_Based\_Access\_Control(y) \land hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(IBM_Cloud_Kubernetes_Service,
        Role Based Access Control)
267. IBM_Cloud_Database has a security feature
        Transparent_Data_Encryption
        FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow
        \exists y (Transparent\_Data\_Encryption(y) \land hasSecurityFeature(x,y))
        DL: hasSecurityFeature(IBM_Cloud_Database,
        Transparent_Data_Encryption)
268. IBM_Cloud_Load_Balancer has a security feature SSL_Offloading
        FOL: \forall x (IBM\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (SSL\_0ffloading(y)) \land
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(IBM_Cloud_Load_Balancer, SSL_Offloading)
269. IBM_Cloud_VPC has a security feature
       Network_Access_Control_List
       FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (Network\_Access\_Control\_List(y) \land \exists y (Network\_Access\_Control\_List(y)) \land \exists y (Network\_Access\_Control
       hasSecurityFeature(x,y))
       DL: hasSecurityFeature(IBM_Cloud_VPC,
       Network_Access_Control_List)
270. IBM_Cloud_Transit_Gateway has a security feature VPN_Tunneling
        FOL: \forall x (IBM\_Cloud\_Transit\_Gateway(x) \rightarrow \exists y (VPN\_Tunneling(y) \land
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(IBM_Cloud_Transit_Gateway, VPN_Tunneling)
271. Oracle_Cloud_VM has a security feature Key_Pair_Authentication
        FOL: \forall x(Oracle\_Cloud\_VM(x) \rightarrow \exists y(Key\_Pair\_Authentication(y) \land
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_VM, Key_Pair_Authentication)
272. Oracle_Cloud_Functions has a security feature
        IAM Role Execution
        FOL: \forall x(Oracle\_Cloud\_Functions(x) \rightarrow \exists y(IAM\_Role\_Execution(y) \land \exists y(IAM\_Role\_Execution(y))
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Functions,
        IAM_Role_Execution)
273. Oracle_Cloud_Block_Storage has a security feature
        Encryption_At_Rest
```

```
FOL: \forall x (Oracle\_Cloud\_Block\_Storage(x) \rightarrow \exists y (Encryption\_At\_Rest(y))
        ∧ hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Block_Storage,
        Encryption_At_Rest)
274. Oracle_Cloud_Object_Storage has a security feature
        Encryption_At_Rest
        FOL: \forall x(0racle\_Cloud\_Object\_Storage(x) \rightarrow
         \exists y (Encryption\_At\_Rest(y) \land hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Object_Storage,
        Encryption_At_Rest)
275. Oracle_Cloud_Kubernetes_Engine has a security feature
        Role_Based_Access_Control
        FOL: \forall x (Oracle\_Cloud\_Kubernetes\_Engine(x) \rightarrow
         \exists y (Role\_Based\_Access\_Control(y) \land hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Kubernetes_Engine,
        Role_Based_Access_Control)
276. Oracle_Cloud_Database_Service has a security feature
        Transparent_Data_Encryption
        FOL: ∀x(Oracle_Cloud_Database_Service(x) →
         \exists y (Transparent\_Data\_Encryption(y) \land hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Database_Service,
        Transparent_Data_Encryption)
277. Oracle_Cloud_Load_Balancer has a security feature
        SSL_Offloading
        FOL: \forall x(Oracle\_Cloud\_Load\_Balancer(x) \rightarrow \exists y(SSL\_Offloading(y)) \land
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_Load_Balancer,
        SSL_Offloading)
278. Oracle_Cloud_VCN has a security feature Network_Security_Group
        FOL: \forall x(Oracle\_Cloud\_VCN(x) \rightarrow \exists y(Network\_Security\_Group(y) \land \exists y(Network\_Security\_Group(y))
        hasSecurityFeature(x,y))
        DL: hasSecurityFeature(Oracle_Cloud_VCN, Network_Security_Group)
279. Alibaba_ECS has a security feature Security_Groups
        FOL: \forall x(Alibaba\_ECS(x) \rightarrow \exists y(Security\_Groups(y) \land \exists y(Security\_Groups(y)) \land \exists y(Security\_Grou
        hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(Alibaba_ECS, Security_Groups)
280. Alibaba_Function_Compute has a security feature
              IAM Role Execution
              FOL: \forall x (Alibaba_Function_Compute(x) \rightarrow \exists y (IAM_Role_Execution(y) \land
              hasSecurityFeature(x,y))
              DL: hasSecurityFeature(Alibaba_Function_Compute,
              IAM_Role_Execution)
281. Alibaba_OSS has a security feature Encryption_At_Rest
              FOL: \forall x(Alibaba_OSS(x) \rightarrow \exists y(Encryption_At_Rest(y) \land Alibaba_OSS(x))
              hasSecurityFeature(x,y))
              DL: hasSecurityFeature(Alibaba_OSS, Encryption_At_Rest)
282. Alibaba_Container_Service has a security feature
              Role_Based_Access_Control
              FOL: ∀x(Alibaba_Container_Service(x) →
              \exists y (Role\_Based\_Access\_Control(y) \land hasSecurityFeature(x,y))
              DL: hasSecurityFeature(Alibaba_Container_Service,
              Role_Based_Access_Control)
283. Alibaba_ApsaraDB has a security feature
             Transparent_Data_Encryption
             FOL: \forall x (Alibaba_ApsaraDB(x) \rightarrow \exists y (Transparent_Data_Encryption(y))
             ∧ hasSecurityFeature(x,y))
             DL: hasSecurityFeature(Alibaba_ApsaraDB,
             Transparent_Data_Encryption)
284. Alibaba_SLB has a security feature SSL_Offloading
              FOL: \forall x (Alibaba\_SLB(x) \rightarrow \exists y (SSL\_Offloading(y) \land \exists y (SSL\_Offloading(y)) \land \exists y (SSL\_Offload
              hasSecurityFeature(x,y))
              DL: hasSecurityFeature(Alibaba_SLB, SSL_Offloading)
285. Alibaba_VPC has a security feature Network_Access_Control_List
              FOL: \forall x(Alibaba\_VPC(x) \rightarrow \exists y(Network\_Access\_Control\_List(y) \land
              hasSecurityFeature(x,y))
              DL: hasSecurityFeature(Alibaba_VPC, Network_Access_Control_List)
286. Alibaba_SAG has a security feature VPN_Tunneling
              FOL: \forall x (Alibaba\_SAG(x) \rightarrow \exists y (VPN\_Tunneling(y) \land \exists y (VPN\_Tunneling(y)) \land \exists y (VPN\_Tunneling
              hasSecurityFeature(x,y))
```

```
DL: hasSecurityFeature(Alibaba_SAG, VPN_Tunneling)
287. AWS_S3 has a compliance feature Data_Encryption
                       FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Data_Encryption(y) \land \exists y (Data_Encryption(y)) 
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_S3, Data_Encryption)
288. AWS_S3 has a compliance feature Regulatory_Compliance
                       FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Regulatory_Compliance(y) \land \exists x (R
                      hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_S3, Regulatory_Compliance)
289. AWS_Lambda has a compliance feature IAM_Access_Control
                      FOL: \forall x (AWS\_Lambda(x) \rightarrow \exists y (IAM\_Access\_Control(y) \land Access\_Control(y))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_Lambda, IAM_Access_Control)
290. AWS_Lambda has a compliance feature Regulatory_Compliance
                       FOL: \forall x (AWS_Lambda(x) \rightarrow \exists y (Regulatory_Compliance(y) \land \exists y (Regulatory_Compliance(y)) \land \exists y (R
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_Lambda, Regulatory_Compliance)
291. AWS_EC2 has a compliance feature Access_Control
                       FOL: \forall x (AWS_EC2(x) \rightarrow \exists y (Access_Control(y) \land \exists y (Access_Control(y
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_EC2, Access_Control)
292. AWS_EC2 hasComplianceFeature IAM_Access_Control
                       FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (IAM\_Access\_Control(y) \land Access\_Control(y))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_EC2, IAM_Access_Control)
293. AWS_RDS has a compliance feature Data_Encryption
                       FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (Data_Encryption(y) \land \exists y (Data_Encryption(y)))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_RDS, Data_Encryption)
294. AWS_RDS hasComplianceFeature Regulatory_Compliance
                       FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (Regulatory_Compliance(y)) \land
                       hasComplianceFeature(x,y))
```

```
DL: hasComplianceFeature(AWS_RDS, Regulatory_Compliance)
295. AWS_RedShift has a compliance feature Data_Encryption
                       FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (Data\_Encryption(y) \land \exists y (Data\_Encryption(y)) \land \exists y (DataEncryption(y)) \land \exists y (D
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_RedShift, Data_Encryption)
296. AWS_RedShift has a compliance feature Regulatory_Compliance
                       FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (Regulatory\_Compliance(y) \land \exists y (Regulatory\_Compliance(y)))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_RedShift, Regulatory_Compliance)
297. AWS_EKS has a compliance feature Access_Control
                      FOL: \forall x(AWS\_EKS(x) \rightarrow \exists y(Access\_Control(y) \land \exists y(Ac
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_EKS, Access_Control)
298. AWS_EKS has a compliance feature Regulatory_Compliance
                       FOL: \forall x (AWS_EKS(x) \rightarrow \exists y (Regulatory_Compliance(y) \land \exists y (Regulatory_Compliance(y)))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_EKS, Regulatory_Compliance)
299. AWS_VPC has a compliance feature Access_Control
                       FOL: \forall x(AWS_VPC(x) \rightarrow \exists y(Access_Control(y) \land \exists y(Ac
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_VPC, Access_Control)
300. AWS_VGW has a compliance feature Regulatory_Compliance
                       FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (Regulatory_Compliance(y) \land \exists y (Regulatory_Compliance(y)))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(AWS_VGW, Regulatory_Compliance)
301. Route_53 has a compliance feature Regulatory_Compliance
                       FOL: \forall x (Route_{53}(x) \rightarrow \exists y (Regulatory_{Compliance(y)} \land \exists y (Regulatory_{Compliance(y)}))
                       hasComplianceFeature(x,y))
                       DL: hasComplianceFeature(Route_53, Regulatory_Compliance)
302. Azure_VM has a compliance feature Access_Control
                       FOL: \forall x (Azure_VM(x) \rightarrow \exists y (Access_Control(y) \land \exists y (Access_Control(
                       hasComplianceFeature(x,y))
```

```
DL: hasComplianceFeature(Azure_VM, Access_Control)
303. Azure_VM has a compliance feature Regulatory_Compliance
                                 FOL: \forall x (Azure_VM(x) \rightarrow \exists y (Regulatory_Compliance(y) \land \exists y (Regulatory_Compliance(y)) \land \exists y (Reg
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_VM, Regulatory_Compliance)
304. Azure_LB has a compliance feature Access_Control
                                 FOL: \forall x (Azure\_LB(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_LB, Access_Control)
305. Azure_VPN_Gateway has a compliance feature
                                 Regulatory_Compliance
                                 FOL: \forall x (Azure_VPN_Gateway(x) \rightarrow \exists y (Regulatory_Compliance(y) \land
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_VPN_Gateway,
                                 Regulatory_Compliance)
306. Azure_DNS has a compliance feature Regulatory_Compliance
                                 FOL: \forall x(Azure\_DNS(x) \rightarrow \exists y(Regulatory\_Compliance(y) \land \exists y(Regulatory\_Compliance(y)) \land \exists y(R
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_DNS, Regulatory_Compliance)
307. Azure_SQL_Database has a compliance feature Data_Encryption
                                 FOL: \forall x (Azure\_SQL\_Database(x) \rightarrow \exists y (Data\_Encryption(y) \land \exists y (Data\_Encryption(y)) \land \exists y (Data_Encryption(y)) \land \exists y (Data_Enc
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_SQL_Database, Data_Encryption)
308. Azure_SQL_Database has a compliance feature
                                 Regulatory_Compliance
                                 FOL: \forall x(Azure\_SQL\_Database(x) \rightarrow \exists y(Regulatory\_Compliance(y) \land \exists y(Regulatory\_Compliance(y)) \land \exists y(Regulatory\_Compliance(y)
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_SQL_Database,
                                 Regulatory_Compliance)
309. Azure_AKS has a compliance feature Access_Control
                                 FOL: \forall x(Azure\_AKS(x) \rightarrow \exists y(Access\_Control(y) \land \exists y(Access\_Control(y)) \land \exists y(Access\_Control(y)
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Azure_AKS, Access_Control)
```

```
310. Azure_Scale_Sets has a compliance feature Access_Control
                                FOL: \forall x(Azure\_Scale\_Sets(x) \rightarrow \exists y(Access\_Control(y) \land \exists y(Access\_Control(y)
                               Azure_Scale_Sets (x,y))
                                DL: hasComplianceFeature(Azure_Scale_Sets, Access_Control)
311. Azure_Functions has a compliance feature IAM_Access_Control
                                FOL: \forall x(Azure\_Functions(x) \rightarrow \exists y(IAM\_Access\_Control(y) \land \exists y(IAM\_Access\_Control(y)) \land \exists y(IAM\_Access
                                hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(Azure_Functions, IAM_Access_Control)
312. GCP_Compute_Engine has a compliance feature Access_Control
                                FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y)) \land \exists y (Access\_C
                                hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(GCP_Compute_Engine, Access_Control)
313. GCS_Bucket has a compliance feature Data_Encryption
                                FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Data\_Encryption(y) \land \exists y (Data\_Encryption(y)))
                               hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(GCS_Bucket, Data_Encryption)
314. GCS_Bucket has a compliance feature Regulatory_Compliance
                                FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Regulatory\_Compliance(y) \land \exists y (Regulatory\_Compliance(y)) \land \exists y (R
                                hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(GCS_Bucket, Regulatory_Compliance)
315. GCP_GKE has a compliance feature Access_Control
                                FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y
                                hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(GCP_GKE, Access_Control)
316. IBM_Cloud_Virtual_Server has a compliance feature
                                Access_Control
                                FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y)) \land \exists y
                                hasComplianceFeature(x,y))
                                DL: hasComplianceFeature(IBM_Cloud_Virtual_Server,
                               Access_Control)
317. IBM_Cloud_Functions has a compliance feature
                                IAM_Access_Control
                                FOL: \forall x (IBM\_Cloud\_Functions(x) \rightarrow \exists y (IAM\_Access\_Control(y) \land
                                hasComplianceFeature(x,y))
```

```
DL: hasComplianceFeature(IBM_Cloud_Functions, IAM_Access_Control)
 318. IBM_Cloud_Object_Storage has a compliance feature
                              Data_Encryption
                              FOL: \forall x (IBM\_Cloud\_Object\_Storage(x) \rightarrow \exists y (Data\_Encryption(y) \land \exists y (Data\_Encryption(y)) \land \exists y (Da
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_Object_Storage,
                              Data_Encryption)
 319. IBM_Cloud_Kubernetes_Service has a compliance feature
                             Access_Control
                             FOL: \forall x (IBM\_Cloud\_Kubernetes\_Service(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y)) \land
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_Kubernetes_Service,
                             Access_Control)
 320. IBM_Cloud_Database has a compliance feature Data_Encryption
                              FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (Data\_Encryption(y) \land \exists y (Data\_Encryption(y)) \land \exists y (DataEncryption(y)) \land \exists
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_Database, Data_Encryption)
 321. IBM_Cloud_Database has a compliance feature
                              Regulatory_Compliance
                              FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (Regulatory\_Compliance(y) \land
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_Database,
                              Regulatory_Compliance)
322. IBM_Cloud_Load_Balancer has a compliance feature
                              Access Control
                              FOL: \forall x (IBM\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y)) \land \exists x (Access\_Control(x)) \land \exists x 
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_Load_Balancer, Access_Control)
 323. IBM_Cloud_VPC has a compliance feature Access_Control
                              FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (Access\_Control(y) \land \exists x (Access\_Control(y) \land \exists y (Access\_Control(y) \land \exists y (Access\_Control(y) \land \exists y (Access\_Con
                              hasComplianceFeature(x,y))
                              DL: hasComplianceFeature(IBM_Cloud_VPC, Access_Control)
 324. IBM_Cloud_Transit_Gateway has a compliance feature
                              Regulatory_Compliance
```

```
FOL: \forall x(IBM\_Cloud\_Transit\_Gateway(x) \rightarrow
         \exists y (Regulatory\_Compliance(y) \land hasComplianceFeature(x,y))
         DL: hasComplianceFeature(IBM_Cloud_Transit_Gateway,
         Regulatory_Compliance)
325. Oracle_Cloud_VM has a compliance feature Access_Control
         FOL: \forall x(Oracle\_Cloud\_VM(x) \rightarrow \exists y(Access\_Control(y) \land \exists y(Access\_Control(y)) \land \exists y(Access\_Cont
         hasComplianceFeature(x,y))
         DL: hasComplianceFeature(Oracle_Cloud_VM, Access_Control)
326. Oracle_Cloud_Functions has a compliance feature
         IAM Access Control
         FOL: \forall x(Oracle\_Cloud\_Functions(x) \rightarrow \exists y(IAM\_Access\_Control(y) \land
         hasComplianceFeature(x,y))
         DL: hasComplianceFeature(Oracle_Cloud_Functions,
         IAM Access Control)
327. Oracle_Cloud_Block_Storage has a compliance feature
         Data_Encryption
         FOL: \forall x(Oracle\_Cloud\_Block\_Storage(x) \rightarrow \exists y(Data\_Encryption(y) \land \exists y(Data\_Encryption(y)) \land \exists y(Data\_Encryption(y))
         hasComplianceFeature(x,y))
         DL: hasComplianceFeature(Oracle_Cloud_Block_Storage,
         Data_Encryption)
328. Oracle_Cloud_Object_Storage has a compliance feature
         Data_Encryption
         FOL: \forall x(0racle\_Cloud\_Object\_Storage(x) \rightarrow \exists y(Data\_Encryption(y) \land \exists y(Data\_Encryption(y)) \land \exists y(Data\_Encryption(y))
         hasComplianceFeature(x,y))
         DL: hasComplianceFeature(Oracle_Cloud_Object_Storage,
         Data_Encryption)
329. Oracle_Cloud_Kubernetes_Engine has a compliance feature
        Access_Control
        FOL: \forall x (Oracle\_Cloud\_Kubernetes\_Engine(x) \rightarrow \exists y (Access\_Control(y))
        ∧ hasComplianceFeature(x,y))
         DL: hasComplianceFeature(Oracle_Cloud_Kubernetes_Engine,
        Access_Control)
330. Oracle_Cloud_Database_Service has a compliance feature
         Data_Encryption
```

```
FOL: \forall x (Oracle\_Cloud\_Database\_Service(x) \rightarrow \exists y (Data\_Encryption(y))
                  ∧ hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Oracle_Cloud_Database_Service,
                   Data_Encryption)
331. Oracle_Cloud_Database_Service has a compliance feature
                   Regulatory_Compliance
                   FOL: ∀x(Oracle_Cloud_Database_Service(x) →
                     \exists y (Regulatory\_Compliance(y) \land hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Oracle_Cloud_Database_Service,
                   Regulatory_Compliance)
332. Oracle_Cloud_Load_Balancer has a compliance feature
                  Access Control
                   FOL: \forall x(Oracle\_Cloud\_Load\_Balancer(x) \rightarrow \exists y(Access\_Control(y) \land
                   hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Oracle_Cloud_Load_Balancer,
                   Access_Control)
333. Oracle_Cloud_VCN has a compliance feature Access_Control
                   FOL: \forall x (Oracle\_Cloud\_VCN(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Control(y)) \land \exists y (Access
                   hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Oracle_Cloud_VCN, Access_Control)
334. Alibaba_ECS has a compliance feature Access_Control
                   FOL: \forall x (Alibaba\_ECS(x) \rightarrow \exists y (Access\_Control(y) \land \exists y (Access\_Contr
                   hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Alibaba_ECS, Access_Control)
335. Alibaba_Function_Compute has a compliance feature
                   IAM Access Control
                   FOL: \forall x (Alibaba_Function_Compute(x) \rightarrow \exists y (IAM_Access_Control(y) \land
                   hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Alibaba_Function_Compute,
                   Data_Encryption)
336. Alibaba_OSS has a compliance feature Data_Encryption
                   FOL: \forall x (Alibaba_0SS(x) \rightarrow \exists y (Data_Encryption(y) \land \exists y (Data_Encryption(y)) \land \exists y (Data_Encryption
                   hasComplianceFeature(x,y))
                   DL: hasComplianceFeature(Alibaba_OSS, Data_Encryption)
```

```
337. Alibaba_Container_Service has a compliance feature
                                 Access Control
                                 FOL: \forall x(Alibaba\_Container\_Service(x) \rightarrow \exists y(Access\_Control(y) \land \exists y(Access\_Control(y)) \land \exists y(A
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Alibaba_Container_Service,
                                 Access_Control)
338. Alibaba_ApsaraDB has a compliance feature Data_Encryption
                                 FOL: \forall x (Alibaba_ApsaraDB(x) \rightarrow \exists y (Data_Encryption(y) \land \exists y (Data_Encryption(y)) \land \exists y (Data_Encry
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Alibaba_ApsaraDB, Data_Encryption)
339. Alibaba_SLB has a compliance feature Access_Control
                                 FOL: \forall x(Alibaba_SLB(x) \rightarrow \exists y(Access_Control(y) \land \exists 
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Alibaba_SLB, Access_Control)
340. Alibaba_VPC has a compliance feature Access_Control
                                 FOL: \forall x (Alibaba_VPC(x) \rightarrow \exists y (Access_Control(y) \land \exists y (Access_Contr
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Alibaba_VPC, Access_Control)
341. Alibaba_SAG has a compliance feature Regulatory_Compliance
                                 FOL: \forall x (Alibaba\_SAG(x) \rightarrow \exists y (Regulatory\_Compliance(y) \land
                                 hasComplianceFeature(x,y))
                                 DL: hasComplianceFeature(Alibaba_SAG, Regulatory_Compliance)
```

Cluster 8: Availability & Reliability Attributes

342. Availability_Feature is an

Availability_and_Reliability_Attribute

FOL: ∀x(Availability_Feature(x) →

Availability_and_Reliability_Attribute(x))

DL: Availability_Feature ⊆

Availability_and_Reliability_Attribute

```
343. Reliability_Feature is an
  Availability_and_Reliability_Attribute
  FOL: ∀x(Reliability_Feature(x) →
  Availability_and_Reliability_Attribute(x))
  DL: Reliability_Feature ⊆ Availability_and_Reliability_Attribute
344. Uptime_SLA is an instance of Availability_Feature
  FOL: Availability_Feature(Uptime_SLA)
  DL: Availability_Feature(Uptime_SLA)
345. Azure_Availability_Sets is an instance of Availability_Feature
  FOL: Availability_Feature(Azure_Availability_Sets)
  DL: Availability_Feature(Azure_Availability_Sets)
346. Availability_Sets is an instance of Availability_Feature
  FOL: Availability_Feature(Availability_Sets)
  DL: Availability_Feature(Availability_Sets)
347. Multi_Region_Storage is an instance of Availability_Feature
  FOL: Availability_Feature(Multi_Region_Storage)
  DL: Availability_Feature(Multi_Region_Storage)
348. Auto_Scaling is an instance of Availability_Feature
  FOL: Availability_Feature(Auto_Scaling)
  DL: Availability_Feature(Auto_Scaling)
349. Failover_Mechanism is an instance of Reliability_Feature
  FOL: Reliability_Feature(Failover_Mechanism)
  DL: Reliability_Feature(Failover_Mechanism)
350. System_Redundancy is an instance of Reliability_Feature
  FOL: Reliability_Feature(System_Redundancy)
  DL: Reliability_Feature(System_Redundancy)
351. Multi_Region_Redundancy is an instance of Reliability_Feature
  FOL: Reliability_Feature(Multi_Region_Redundancy)
  DL: Reliability_Feature(Multi_Region_Redundancy)
352. Failover_Cluster is an instance of Reliability_Feature
  FOL: Reliability_Feature(Failover_Cluster)
```

```
DL: Reliability_Feature(Failover_Cluster)
353. Regional_Failover is an instance of Reliability_Feature
                    FOL: Reliability_Feature(Regional_Failover)
                    DL: Reliability_Feature(Regional_Failover)
354. Active_Geo_Replication is an instance of Reliability_Feature
                    FOL: Reliability_Feature(Active_Geo_Replication)
                    DL: Reliability_Feature(Active_Geo_Replication)
355. Redundancy_Across_AZs is an instance of Reliability_Feature
                    FOL: Reliability_Feature(Redundancy_Across_AZs)
                   DL: Reliability_Feature(Redundancy_Across_AZs)
356. AWS_S3 has a reliability feature Redundancy_Across_AZs
                    FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Redundancy_Across_AZs(y) \land \exists y (Redundancy_Across_AZs(y)) \land \exists x (Redundancy_ACross_AZs(y)) \land \exists x (Redundancy_ACross_AZs(y)) \land \exists x (Redundancy_ACross_AZs(y)) \land \exists x (Redundancy_ACross_AZs(x)) \land \exists x (Redundancy_ACross_AZs(x)) \land \exists x (Redundancy_ACross_AZs(x)) \land \exists x (Redundancy_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_ACross_A
                   has Reliability Feature(x,y)
                    DL: hasReliabilityFeature(AWS_S3, Redundancy_Across_AZs)
357. AWS_S3 has a reliability feature Multi_Region_Redundancy
                    FOL: \forall x (AWS_S3(x) \rightarrow \exists y (Multi_Region_Redundancy(y)) \land
                   has Reliability Feature(x,y)
                    DL: hasReliabilityFeature(AWS_S3, Multi_Region_Redundancy)
358. AWS_Lambda has a reliability feature Regional_Failover
                    FOL: \forall x (AWS\_Lambda(x) \rightarrow \exists y (Regional\_Failover(y) \land Australian Australian
                    hasReliabilityFeature(x,y))
                    DL: hasReliabilityFeature(AWS_Lambda, Regional_Failover)
359. AWS_EC2 has an availability feature Auto_Scaling
                    FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)))
                    has Availability Feature (x,y)
                    DL: hasAvailabilityFeature(AWS_EC2, Auto_Scaling)
360. AWS_RDS has a reliability feature Failover_Cluster
                    FOL: \forall x (AWS_RDS(x) \rightarrow \exists y (Failover_Cluster(y) \land \exists y (Failover_Cluster(y)) \land \exists y (Failover_Cluster(y) \land \exists y (Failover_Cluster(y)) \land \exists y (Failover_Cl
                    hasReliabilityFeature(x,y))
                    DL: hasReliabilityFeature(AWS_RDS, Failover_Cluster)
361. AWS_EKS has an availability feature Auto_Scaling
                    FOL: \forall x (AWS\_EKS(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) \land \exists y (A
```

```
has Availability Feature (x,y)
          DL: hasAvailabilityFeature(AWS_EKS, Auto_Scaling)
362. AWS_VPC has a reliability feature Redundancy_Across_AZs
          FOL: \forall x (AWS_VPC(x) \rightarrow \exists y (Redundancy_Across_AZs(y) \land Across_AZs(y))
          has Reliability Feature(x,y)
          DL: hasReliabilityFeature(AWS_VPC, Redundancy_Across_AZs)
363. AWS_VGW has a reliability feature Regional_Failover
          FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (Regional_Failover(y) \land Automatical A
          hasReliabilityFeature(x,y))
          DL: hasReliabilityFeature(AWS_VGW, Regional_Failover)
364. Route_53 has a reliability feature Multi_Region_Redundancy
          FOL: \forall x (Route_{53}(x) \rightarrow \exists y (Multi_{Region_{Redundancy}(y)}) \land
          has Reliability Feature(x,y)
          DL: hasReliabilityFeature(Route_53, Multi_Region_Redundancy)
365. AWS_RedShift has a reliability feature Multi_Region_Redundancy
          FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (Multi\_Region\_Redundancy(y) \land
          has Reliability Feature(x,y)
          DL: hasReliabilityFeature(AWS_RedShift, Multi_Region_Redundancy)
366. Azure_VM has an availability feature Azure_Availability_Sets
          FOL: ∀x(Azure_VM(x) → ∃y(Azure_Availability_Sets(y) ∧
          has Availability Feature (x,y)
          DL: hasAvailabilityFeature(Azure_VM, Azure_Availability_Sets)
367. Azure_VM has an availability feature Availability_Sets
          FOL: \forall x(Azure_VM(x) \rightarrow \exists y(Availability_Sets(y) \land \forall y(Availability_Sets(y) \land y(Availab
          has Availability Feature (x, y)
          DL: hasAvailabilityFeature(Azure_VM, Availability_Sets)
368. Azure_LB has a reliability feature Multi_Region_Redundancy
          FOL: \forall x(Azure\_LB(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land Autority)
          hasReliabilityFeature(x,y))
          DL: hasReliabilityFeature(Azure_LB, Multi_Region_Redundancy)
369. Azure_VPN_Gateway has a reliability feature
          Multi_Region_Redundancy
          FOL: \forall x(Azure\_VPN\_Gateway(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land
```

```
has Reliability Feature(x,y)
                      DL: hasReliabilityFeature(Azure_VPN_Gateway,
                      Multi_Region_Redundancy)
370. Azure_DNS has a reliability feature Multi_Region_Redundancy
                      FOL: \forall x(Azure\_DNS(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land
                      hasReliabilityFeature(x,y))
                      DL: hasReliabilityFeature(Azure_DNS, Multi_Region_Redundancy)
371. Azure_SQL_Database has a reliability feature
                     Active_Geo_Replication
                     FOL: \forall x (Azure\_SQL\_Database(x) \rightarrow \exists y (Active\_Geo\_Replication(y) \land \exists y (Active\_Geo\_Replication(y)) \land \exists y (
                      hasReliabilityFeature(x,y))
                     DL: hasReliabilityFeature(Azure_SQL_Database,
                     Active_Geo_Replication)
372. Azure_SQL_Database has a reliability feature Failover_Cluster
                      FOL: \forall x(Azure\_SQL\_Database(x) \rightarrow \exists y(Failover\_Cluster(y) \land \exists y(Failover\_Cluster(y)) \land \exists y(Failover\_C
                      hasReliabilityFeature(x,y))
                      DL: hasReliabilityFeature(Azure_SQL_Database, Failover_Cluster)
373. Azure_AKS has a reliability feature Multi_Region_Redundancy
                      FOL: \forall x(Azure\_AKS(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land
                      hasReliabilityFeature(x,y))
                      DL: hasReliabilityFeature(Azure_AKS, Multi_Region_Redundancy)
374. Azure_Scale_Sets has an availability feature Auto_Scaling
                      FOL: \forall x (Azure\_Scale\_Sets(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) \land \exists y (Auto\_Scaling(y))
                      has Availability Feature (x, y)
                      DL: hasAvailabilityFeature(Azure_Scale_Sets, Auto_Scaling)
375. Azure_Functions has a reliability feature Regional_Failover
                      FOL: \forall x (Azure\_Functions(x) \rightarrow \exists y (Regional\_Failover(y) \land \exists y (Regional\_Failover(y)))
                      hasReliabilityFeature(x,y))
                      DL: hasReliabilityFeature(Azure_Functions, Regional_Failover)
376. GCP_Compute_Engine has an availability feature Auto_Scaling
                      FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) \land \exists y (Auto\_Scaling(y
                      has Availability Feature (x,y)
                      DL: hasAvailabilityFeature(GCP_Compute_Engine, Auto_Scaling)
```

```
377. GCS_Bucket has an availability feature Multi_Region_Storage
            FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Multi\_Region\_Storage(y) \land \exists y (Multi\_Region\_Storage(y)) \land \exists y
            hasAvailabilityFeature(x,y))
            DL: hasAvailabilityFeature(GCS_Bucket, Multi_Region_Storage)
378. GCS_Bucket has a reliability feature Multi_Region_Redundancy
            FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Multi\_Region\_Redundancy(y) \land
            has Reliability Feature(x,y)
            DL: hasReliabilityFeature(GCS_Bucket, Multi_Region_Redundancy)
379. GCP_GKE has a reliability feature Multi_Region_Redundancy
            FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (Multi\_Region\_Redundancy(y) \land
            hasReliabilityFeature(x,y))
            DL: hasReliabilityFeature(GCP_GKE, Multi_Region_Redundancy)
380. IBM_Cloud_Virtual_Server has an availability feature
            Auto_Scaling
            FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) \land \exists y (Auto\_Sca
            hasAvailabilityFeature(x,y))
            DL: hasAvailabilityFeature(IBM_Cloud_Virtual_Server,
            Auto_Scaling)
381. IBM_Cloud_Object_Storage has a reliability feature
            Multi_Region_Redundancy
            FOL: ∀x(IBM_Cloud_Object_Storage(x) →
             \exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
            DL: hasReliabilityFeature(IBM_Cloud_Object_Storage,
            Multi_Region_Redundancy)
382. IBM_Cloud_Kubernetes_Service has a reliability feature
            Multi_Region_Redundancy
            FOL: ∀x(IBM_Cloud_Kubernetes_Service(x) →
             \exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
            DL: hasReliabilityFeature(IBM_Cloud_Kubernetes_Service,
            Multi_Region_Redundancy)
383. IBM_Cloud_Database has a reliability feature Failover_Cluster
            FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (Failover\_Cluster(y) \land A)
            has Reliability Feature(x,y)
            DL: hasReliabilityFeature(IBM_Cloud_Database, Failover_Cluster)
```

```
384. IBM_Cloud_Load_Balancer has an availability feature
                      Auto_Scaling
                       FOL: \forall x (IBM\_Cloud\_Load\_Balancer(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) \land \exists y (Auto\_Scal
                       has Availability Feature (x, y)
                       DL: hasAvailabilityFeature(IBM_Cloud_Load_Balancer, Auto_Scaling)
385. IBM_Cloud_VPC has a reliability feature
                       Multi_Region_Redundancy
                       FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (Multi\_Region\_Redundancy(y) \land \exists y (Multi\_Region\_Redundancy(y)) \land \exists y (Multi\_Redundancy(y)) \land \exists y (Mu
                       has Reliability Feature(x,y)
                       DL: hasReliabilityFeature(IBM_Cloud_VPC, Multi_Region_Redundancy)
386. IBM_Cloud_Transit_Gateway has a reliability feature
                       Regional_Failover
                       FOL: \forall x (IBM\_Cloud\_Transit\_Gateway(x) \rightarrow \exists y (Regional\_Failover(y) \land \exists y (Regional\_Failover(y)) \land \exists y
                      has Reliability Feature(x,y)
                      DL: hasReliabilityFeature(IBM_Cloud_Transit_Gateway,
                       Regional_Failover)
387. Oracle_Cloud_VM has an availability feature Auto_Scaling
                       FOL: \forall x (Oracle\_Cloud\_VM(x) \rightarrow \exists y (Auto\_Scaling(y) \land \exists y (Auto\_Scaling(y)) 
                       has Availability Feature (x,y)
                       DL: hasAvailabilityFeature(Oracle_Cloud_VM, Auto_Scaling)
388. Oracle_Cloud_Block_Storage has a reliability feature
                       Multi_Region_Redundancy
                       FOL: \forall x(Oracle\_Cloud\_Block\_Storage(x) \rightarrow
                        \exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
                       DL: hasReliabilityFeature(Oracle_Cloud_Block_Storage,
                       Multi_Region_Redundancy)
389. Oracle_Cloud_Object_Storage has a reliability feature
                       Multi_Region_Redundancy
                       FOL: \forall x(0racle\_Cloud\_Object\_Storage(x) \rightarrow
                        \exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
                       DL: hasReliabilityFeature(Oracle_Cloud_Object_Storage,
                       Multi_Region_Redundancy)
390. Oracle_Cloud_Kubernetes_Engine has a reliability feature
                       Multi_Region_Redundancy
                       FOL: ∀x(Oracle_Cloud_Kubernetes_Engine(x) →
```

```
\exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
                   DL: hasReliabilityFeature(Oracle_Cloud_Kubernetes_Engine.
                   Multi_Region_Redundancy)
391. Oracle_Cloud_Database_Service has a reliability feature
                   Failover Cluster
                   FOL: ∀x(Oracle_Cloud_Database_Service(x) →
                     ∃y(Failover_Cluster(y) ∧ hasReliabilityFeature(x,y))
                   DL: hasReliabilityFeature(Oracle_Cloud_Database_Service,
                   Failover_Cluster)
392. Oracle_Cloud_Load_Balancer has an availability feature
                  Auto_Scaling
                   FOL: \forall x(Oracle\_Cloud\_Load\_Balancer(x) \rightarrow \exists y(Auto\_Scaling(y) \land \exists y(Auto\_Scaling(y)) \land \exists y
                   has Availability Feature (x,y)
                  DL: hasAvailabilityFeature(Oracle_Cloud_Load_Balancer,
                  Auto_Scaling)
393. Oracle_Cloud_VCN has a reliability feature
                   Multi_Region_Redundancy
                   FOL: \forall x (Oracle\_Cloud\_VCN(x) \rightarrow \exists y (Multi\_Region\_Redundancy(y) \land
                   hasReliabilityFeature(x,y))
                   DL: hasReliabilityFeature(Oracle_Cloud_VCN,
                   Multi_Region_Redundancy)
394. Alibaba_ECS has an availability feature Auto_Scaling
                   FOL: \forall x(Alibaba\_ECS(x) \rightarrow \exists y(Auto\_Scaling(y) \land \exists y(Auto\_Scaling(y)) \land \exists y(Auto\_Scaling(y
                   hasAvailabilityFeature(x,y))
                   DL: hasAvailabilityFeature(Alibaba_ECS, Auto_Scaling)
395. Alibaba_Function_Compute has a reliability feature
                   Regional_Failover
                   FOL: \forall x (Alibaba_Function_Compute(x) \rightarrow \exists y (Regional_Failover(y) \land \exists y (Regional_Failover(y)) \land \exists x (Regional_Failover(y)) \land \exists x (Regional_Failover(y)) \land \exists x (Regional_Failover(y)) \land \exists x 
                   hasReliabilityFeature(x,y))
                   DL: hasReliabilityFeature(Alibaba_Function_Compute,
                   Regional_Failover)
396. Alibaba_OSS has a reliability feature Multi_Region_Redundancy
                   FOL: \forall x (Alibaba_OSS(x) \rightarrow \exists y (Multi_Region_Redundancy(y) \land
                   hasReliabilityFeature(x,y))
```

```
DL: hasReliabilityFeature(Alibaba_OSS, Multi_Region_Redundancy)
397. Alibaba_Container_Service has a reliability feature
         Multi_Region_Redundancy
         FOL: ∀x(Alibaba_Container_Service(x) →
          \exists y (Multi\_Region\_Redundancy(y) \land hasReliabilityFeature(x,y))
         DL: hasReliabilityFeature(Alibaba_Container_Service,
         Multi_Region_Redundancy)
398. Alibaba_ApsaraDB has a reliability feature Failover_Cluster
         FOL: \forall x(Alibaba\_ApsaraDB(x) \rightarrow \exists y(Failover\_Cluster(y) \land
         hasReliabilityFeature(x,y))
         DL: hasReliabilityFeature(Alibaba_ApsaraDB, Failover_Cluster)
399. Alibaba_SLB has an availability feature Auto_Scaling
         FOL: \forall x (Alibaba_SLB(x) \rightarrow \exists y (Auto_Scaling(y) \land \exists y (Auto_Scalin
         has Availability Feature (x,y)
         DL: hasAvailabilityFeature(Alibaba_SLB, Auto_Scaling)
400. Alibaba_VPC has a reliability feature Multi_Region_Redundancy
         FOL: \forall x(Alibaba\_VPC(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land
         has Reliability Feature(x,y)
         DL: hasReliabilityFeature(Alibaba_VPC, Multi_Region_Redundancy)
401. Alibaba_SAG has a reliability feature Multi_Region_Redundancy
         FOL: \forall x(Alibaba\_SAG(x) \rightarrow \exists y(Multi\_Region\_Redundancy(y) \land
         hasReliabilityFeature(x,y))
         DL: hasReliabilityFeature(Alibaba_SAG, Multi_Region_Redundancy)
```

Cluster 9: Usage, Composition & Recommendations

```
402. Compute_Instance is referenced by AWS_EC2 FOL: \forall x (Compute_Instance(x) \rightarrow \exists y (AWS_EC2(y) \land isReferencedBy(x,y)))
DL: Compute_Instance \subseteq isReferencedBy.{AWS_EC2}
```

```
403. Serverless_Computing is referenced by AWS_Lambda
               FOL: \forall x (Serverless\_Computing(x) \rightarrow \exists y (AWS\_Lambda(y)) \land
               isReferencedBy(x,y))
               DL: Serverless_Computing ⊆ isReferencedBy.{AWS_Lambda}
404. Object_Storage is referenced by AWS_S3
               FOL: \forall x (0bject\_Storage\_Resource(x) \rightarrow \exists y (AWS\_S3(y) \land Augustation Augustat
               isReferencedBy(x,y))
               DL: Object_Storage_Resource ⊆ isReferencedBy.{AWS_S3}
405. Virtual_Network is referenced by AWS_VPC
              FOL: \forall x(Virtual\_Network(x) \rightarrow \exists y(AWS\_VPC(y) \land \exists y(AWS\_VPC(y)))
               isReferencedBy(x,y))
               DL: Virtual_Network \subseteq isReferencedBy.{AWS_VPC}
406. Load_Balancer is referenced by ELB
              FOL: \forall x (Load\_Balancer(x) \rightarrow \exists y (ELB(y) \land isReferencedBy(x,y)))
               DL: Load_Balancer ⊆ isReferencedBy.{ELB}
407. VPN_Gateway is referenced by AWS_VGW
              FOL: \forall x (VPN\_Gateway(x) \rightarrow \exists y (AWS\_VGW(y) \land isReferencedBy(x,y)))
               DL: VPN_Gateway ⊆ isReferencedBy.{AWS_VGW}
408. DNS_Service is referenced by Route_53
               FOL: \forall x (DNS\_Service(x) \rightarrow \exists y (Route\_53(y) \land isReferencedBy(x,y)))
               DL: DNS_Service ⊆ isReferencedBy.{Route_53}
409. Data_Warehouse is referenced by AWS_RedShift
               FOL: \forall x(Data\_Warehouse(x) \rightarrow \exists y(AWS\_RedShift(y) \land \exists y(AWS\_RedShift(y)))
               isReferencedBy(x,y))
               DL: Data_Warehouse ⊆ isReferencedBy.{AWS_RedShift}
410. Kubernetes_Service is referenced by AWS_EKS
               FOL: \forall x (Kubernetes\_Service(x) \rightarrow \exists y (AWS\_EKS(y) \land Australian A
               isReferencedBy(x,y))
               DL: Kubernetes_Service ⊆ isReferencedBy.{AWS_EKS}
411. Relational_Database is referenced by AWS_RDS
               FOL: \forall x (Relational\_Database(x) \rightarrow \exists y (AWS\_RDS(y) \land Australian 
               isReferencedBy(x,y))
```

```
DL: Relational_Database ⊆ isReferencedBy.{AWS_RDS}
412. Compute_Instance is referenced by Azure_VM
              FOL: \forall x (Compute\_Instance(x) \rightarrow \exists y (Azure\_VM(y) \land Azure\_VM(y))
              isReferencedBy(x,y))
              DL: Compute_Instance ⊆ isReferencedBy.{Azure_VM}
413. Virtual_Network is referenced by Azure_VNet
              FOL: \forall x(Virtual\_Network(x) \rightarrow \exists y(Azure\_VNet(y) \land \exists y(Azure\_VNet(y)))
              isReferencedBy(x,y))
              DL: Virtual_Network ⊆ isReferencedBy.{Azure_VNet}
414. Load_Balancer is referenced by Azure_LB
              FOL: \forall x(Load\_Balancer(x) \rightarrow \exists y(Azure\_LB(y) \land isReferencedBy(x,y))
              DL: Load_Balancer ⊆ isReferencedBy.{Azure_LB}
415. VPN_Gateway is referenced by Azure_VPN_Gateway
              FOL: \forall x (VPN\_Gateway(x) \rightarrow \exists y (Azure\_VPN\_Gateway(y) \land \exists y (Azure\_VPN\_Gateway(y)) \land \exists y (Azure\_VPN\_Ga
              isReferencedBy(x,y))
              DL: VPN_Gateway ⊆ isReferencedBy.{Azure_VPN_Gateway}
416. DNS_Service is referenced by Azure_DNS
              FOL: \forall x(DNS\_Service(x) \rightarrow \exists y(Azure\_DNS(y) \land isReferencedBy(x,y))
              DL: DNS_Service ⊆ isReferencedBy.{Azure_DNS}
417. Relational_Database is referenced by Azure_SQL_Database
              FOL: \forall x (Relational\_Database(x) \rightarrow \exists y (Azure\_SQL\_Database(y)) \land
              isReferencedBy(x,y))
              DL: Relational_Database ⊆ isReferencedBy.{Azure_SQL_Database}
418. Kubernetes_Service is referenced by Azure_AKS
              FOL: \forall x (Kubernetes\_Service(x) \rightarrow \exists y (Azure\_AKS(y) \land Azure\_AKS(y))
              isReferencedBy(x,y))
              DL: Kubernetes_Service ⊆ isReferencedBy.{Azure_AKS}
419. Auto_Scale is referenced by Azure_Scale_Sets
              FOL: \forall x (Auto\_Scale(x) \rightarrow \exists y (Azure\_Scale\_Sets(y) \land \exists y (Azure\_Sets(y) \land
              isReferencedBy(x,y))
              DL: Auto_Scale ⊆ isReferencedBy.{Azure_Scale_Sets}
```

```
420. Serverless_Computing is referenced by Azure_Functions
                  FOL: \forall x (Serverless\_Computing(x) \rightarrow \exists y (Azure\_Functions(y) \land \exists y (Azure\_Functions(y)) \land \exists y (Azure\_
                  isReferencedBy(x,y))
                  DL: Serverless_Computing ⊆ isReferencedBy.{Azure_Functions}
421. Compute_Instance is referenced by GCP_Compute_Engine
                  FOL: \forall x (Compute\_Instance(x) \rightarrow \exists y (GCP\_Compute\_Engine(y) \land \exists y (GCP\_Compute\_Engine(y)) \land \exists y
                  isReferencedBy(x,y))
                  DL: Compute_Instance ⊆ isReferencedBy.{GCP_Compute_Engine}
422. Block_Storage is referenced by GCS_Bucket
                  FOL: \forall x (Block\_Storage(x) \rightarrow \exists y (GCS\_Bucket(y)) \land
                  isReferencedBy(x,y))
                  DL: Block_Storage ⊆ isReferencedBy.{GCS_Bucket}
423. Kubernetes_Service is referenced by GCP_GKE
                 FOL: \forall x (Kubernetes\_Service(x) \rightarrow \exists y (GCP\_GKE(y) \land A)
                  isReferencedBy(x,y))
                  DL: Kubernetes_Service ⊆ isReferencedBy.{GCP_GKE}
424. Compute_Instance is referenced by IBM_Cloud_Virtual_Server
                  FOL: \forall x (Compute\_Instance(x) \rightarrow \exists y (IBM\_Cloud\_Virtual\_Server(y) \land
                  isReferencedBy(x,y))
                  DL: Compute_Instance ⊆ isReferencedBy.{IBM_Cloud_Virtual_Server}
425. Function_as_a_Service is referenced by IBM_Cloud_Functions
                  FOL: \forall x (Function_as_a\_Service(x) \rightarrow \exists y (IBM\_Cloud\_Functions(y) \land \exists y (IBM\_Cloud\_Functions(y)) \land \exists y (IBM\_Cloud\_Functions
                  isReferencedBy(x,y))
                  DL: Function_as_a_Service ⊆ isReferencedBy.{IBM_Cloud_Functions}
426. Object_Storage is referenced by IBM_Cloud_Object_Storage
                  FOL: \forall x (0bject\_Storage(x) \rightarrow \exists y (IBM\_Cloud\_0bject\_Storage(y) \land
                  isReferencedBy(x,y))
                  DL: Object_Storage ⊆ isReferencedBy.{IBM_Cloud_Object_Storage}
427. Kubernetes_Service is referenced by
                  IBM Cloud Kubernetes Service
                  FOL: ∀x(Kubernetes_Service(x) →
                    \exists y(IBM\_Cloud\_Kubernetes\_Service(y) \land isReferencedBy(x,y)))
                  DL: Kubernetes_Service ⊆
```

```
isReferencedBy.{IBM_Cloud_Kubernetes_Service}
428. Database_Service is referenced by IBM_Cloud_Database
             FOL: \forall x(Database\_Service(x) \rightarrow \exists y(IBM\_Cloud\_Database(y) \land
             isReferencedBy(x,y))
             DL: Database_Service ⊆ isReferencedBy.{IBM_Cloud_Database}
429. Load_Balancer is referenced by IBM_Cloud_Load_Balancer
            FOL: \forall x(Load\_Balancer(x) \rightarrow \exists y(IBM\_Cloud\_Load\_Balancer(y) \land
             isReferencedBy(x,y))
             DL: Load_Balancer ⊆ isReferencedBy.{IBM_Cloud_Load_Balancer}
430. Virtual_Network_Service is referenced by IBM_Cloud_VPC
            FOL: \forall x(Virtual\_Network\_Service(x) \rightarrow \exists y(IBM\_Cloud\_VPC(y) \land \exists y(IBM\_Cloud\_VPC(y)))
             isReferencedBy(x,y))
             DL: Virtual_Network_Service ⊆ isReferencedBy.{IBM_Cloud_VPC}
431. VPN_Gateway is referenced by IBM_Cloud_Transit_Gateway
            FOL: \forall x (VPN\_Gateway(x) \rightarrow \exists y (IBM\_Cloud\_Transit\_Gateway(y) \land
             isReferencedBy(x,y))
             DL: VPN_Gateway ⊆ isReferencedBy.{IBM_Cloud_Transit_Gateway}
432. Compute_Instance is referenced by Oracle_Cloud_VM
             FOL: \forall x (Compute\_Instance(x) \rightarrow \exists y (Oracle\_Cloud\_VM(y) \land \exists y (Oracle\_Cloud\_VM(y)) \land \exists y (Oracle\_Cloud\_Cloud\_VM(y)) \land \exists y (Oracle\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_
             isReferencedBy(x,y))
             DL: Compute_Instance ⊆ isReferencedBy.{Oracle_Cloud_VM}
433. Function_as_a_Service is referenced by Oracle_Cloud_Functions
             FOL: \forall x (Function_as_a\_Service(x) \rightarrow \exists y (Oracle\_Cloud\_Functions(y))
            \land isReferencedBy(x,y)))
             DL: Function as a Service ⊆
             isReferencedBy.{Oracle_Cloud_Functions}
434. Block_Storage is referenced by Oracle_Cloud_Block_Storage
             FOL: \forall x (Block\_Storage(x) \rightarrow \exists y (Oracle\_Cloud\_Block\_Storage(y) \land
             isReferencedBy(x,y))
             DL: Block_Storage ⊆ isReferencedBy.{Oracle_Cloud_Block_Storage}
435. Object_Storage is referenced by Oracle_Cloud_Object_Storage
             FOL: \forall x(0bject\_Storage\_Resource(x) \rightarrow \exists y(0racle\_Cloud(y) \land \exists y(0racle\_Cloud(y)) \land \exists y(0r
             isReferencedBy(x,y))
```

```
DL: Object_Storage ⊆
       isReferencedBy.{Oracle_Cloud_Object_Storage}
436. Kubernetes_Service is referenced by
       Oracle_Cloud_Kubernetes_Engine
       FOL: ∀x(Kubernetes_Service(x) →
        \exists y(Oracle\_Cloud\_Kubernetes\_Engine(y) \land isReferencedBy(x,y)))
       DL: Kubernetes Service ⊆
       isReferencedBy.{Oracle_Cloud_Kubernetes_Engine}
437. Database_Service is referenced by
       Oracle_Cloud_Database_Service
       FOL: ∀x(Database_Service(x) →
        \exists y (Oracle\_Cloud\_Database\_Service(y) \land isReferencedBy(x,y)))
       DL: Database_Service ⊆
       isReferencedBy.{Oracle_Cloud_Database_Service}
438. Load_Balancer is referenced by Oracle_Cloud_Load_Balancer
       FOL: \forall x(Load\_Balancer(x) \rightarrow \exists y(Oracle\_Cloud\_Load\_Balancer(y)) \land
       isReferencedBy(x,y))
       DL: Load_Balancer ⊆ isReferencedBy.{Oracle_Cloud_Load_Balancer}
439. Virtual_Network_Service is referenced by Oracle_Cloud_VCN
       FOL: \forall x(Virtual\_Network\_Service(x) \rightarrow \exists y(Oracle\_Cloud\_VCN(y) \land \exists y(Oracle\_Cloud\_VCN(y)) \land \exists y(Oracle\_Cloud\_Cloud\_VCN(y)) \land \exists y(Oracle\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_Cloud\_C
       isReferencedBy(x,y))
       DL: Virtual_Network_Service ⊆ isReferencedBy.{Oracle_Cloud_VCN}
440. Compute_Instance is referenced by Alibaba_ECS
       FOL: \forall x (Compute\_Instance(x) \rightarrow \exists y (Alibaba\_ECS(y) \land Alibaba\_ECS(y))
       isReferencedBy(x,y))
       DL: Compute_Instance ⊆ isReferencedBy.{Alibaba_ECS}
441. Function_as_a_Service is referenced by Alibaba_Function_Compute
       FOL: \forall x(Function_as_a\_Service(x) \rightarrow
        \exists y (Alibaba_Function_Compute(y) \land isReferencedBy(x,y)))
       DL: Function_as_a_Service ⊆
       isReferencedBy.{Alibaba_Function_Compute}
442. Object_Storage is referenced by Alibaba_OSS
       FOL: \forall x (0bject\_Storage\_Resource(x) \rightarrow \exists y (Alibaba\_OSS(y) \land Alibaba\_OSS(y))
       isReferencedBy(x,y))
```

```
DL: Object_Storage ⊆ isReferencedBy.{Alibaba_OSS}
443. Kubernetes_Service is referenced by Alibaba_Container_Service
   FOL: \forall x (Kubernetes\_Service(x) \rightarrow \exists y (Alibaba\_Container\_Service(y))
   \land isReferencedBy(x,y)))
   DL: Kubernetes Service ⊆
   isReferencedBy.{Alibaba_Container_Service}
444. Database_Service is referenced by Alibaba_ApsaraDB
   FOL: \forall x(Database\_Service(x) \rightarrow \exists y(Alibaba\_ApsaraDB(y) \land
   isReferencedBy(x,y))
   DL: Database_Service ⊆ isReferencedBy.{Alibaba_ApsaraDB}
445. Load_Balancer is referenced by Alibaba_SLB
   FOL: \forall x (Load\_Balancer(x) \rightarrow \exists y (Alibaba\_SLB(y) \land Alibaba\_SLB(y))
   isReferencedBy(x,y))
   DL: Load_Balancer ⊆ isReferencedBy.{Alibaba_SLB}
446. Virtual_Network_Service is referenced by Alibaba_VPC
   FOL: \forall x(Virtual\_Network\_Service(x) \rightarrow \exists y(Alibaba\_VPC(y) \land \exists y(Alibaba\_VPC(y))
   isReferencedBy(x,y))
   DL: Virtual_Network_Service ⊆ isReferencedBy.{Alibaba_VPC}
447. VPN_Gateway is referenced by Alibaba_SAG
   FOL: \forall x (VPN\_Gateway(x) \rightarrow \exists y (Alibaba\_SAG(y) \land Alibaba\_SAG(y))
   isReferencedBy(x,y))
   DL: VPN_Gateway ⊆ isReferencedBy.{Alibaba_SAG}
448. High_CPU_Workload is an instance of Workload
   FOL: Workload(High_CPU_Workload)
   DL: Workload(High_CPU_Workload)
449. Memory_Optimized_Workload is an instance of Workload
   FOL: Workload(Memory_Optimized_Workload)
   DL: Workload(Memory_Optimized_Workload)
450. Storage_Optimized_Workload is an instance of Workload
   FOL: Workload(Storage_Optimized_Workload)
   DL: Workload(Storage_Optimized_Workload)
```

```
451. General Purpose Workload is an instance of Workload
                        FOL: Workload(General_Purpose_Workload)
                        DL: Workload(General_Purpose_Workload)
452. GPU Workload is an instance of Workload
                        FOL: Workload(GPU_Workload)
                        DL: Workload(GPU_Workload)
453. Network_Optimized_Workload is an instance of Workload
                       FOL: Workload(Network_Optimized_Workload)
                       DL: Workload(Network_Optimized_Workload)
454. AWS_EC2 is suitable for High_CPU_Workload
                        FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (High\_CPU\_Workload(y) \land suitableFor(x,y))
                       DL: suitableFor(AWS_EC2, High_CPU_Workload)
455. AWS_EC2 is suitable for General_Purpose_Workload
                        FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (General\_Purpose\_Workload(y)) \land
                        suitableFor(x,y))
                        DL: suitableFor(AWS_EC2, General_Purpose_Workload)
456. AWS_EC2 is suitable for Network_Optimized_Workload
                        FOL: \forall x (AWS\_EC2(x) \rightarrow \exists y (Network\_Optimized\_Workload(y) \land \exists y (Network\_Optimized\_Workload(y)) \land \exists y (Network\_Optimized\_Work(y)) \land \exists y (Network(y)) \land \exists y (Netw
                        suitableFor(x,y))
                        DL: suitableFor(AWS_EC2, Network_Optimized_Workload)
457. AWS_Lambda is suitable for General_Purpose_Workload
                        FOL: \forall x (AWS_Lambda(x) \rightarrow \exists y (General_Purpose_Workload(y) \land \exists y (General_Purpose_Workload(y)) \land \exists y
                        suitableFor(x,y))
                        DL: suitableFor(AWS_Lambda, General_Purpose_Workload)
458. AWS_VPC is suitable for General_Purpose_Workload
                        FOL: \forall x (AWS_VPC(x) \rightarrow \exists y (General_Purpose_Workload(y) \land \exists y (General_Purpose_Workload(y)) \land \exists y (G
                        suitableFor(x,y))
                        DL: suitableFor(AWS_VPC, General_Purpose_Workload)
459. ELB is suitable for General_Purpose_Workload
                        FOL: \forall x (ELB(x) \rightarrow \exists y (General\_Purpose\_Workload(y) \land \exists y (General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y) \land General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y) \land General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y) \land General\_Purpose\_Workload(y)) \land General\_Purpose\_Workload(y) \land
                        suitableFor(x,y))
```

```
DL: suitableFor(ELB, General_Purpose_Workload)
460. AWS_VGW is suitable for Network_Optimized_Workload
            FOL: \forall x (AWS_VGW(x) \rightarrow \exists y (Network_Optimized_Workload(y) \land \exists y (Network_Optimized_Workload(y)) \land \exists y (Network_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Optimized_Work_Opti
            suitableFor(x,y))
            DL: suitableFor(AWS_VGW, Network_Optimized_Workload)
461. Route_53 is suitable for General_Purpose_Workload
            FOL: \forall x (Route_53(x) \rightarrow \exists y (General_Purpose_Workload(y)) \land
            suitableFor(x,y))
            DL: suitableFor(Route_53, General_Purpose_Workload)
462. AWS_RedShift is suitable for Storage_Optimized_Workload
            FOL: \forall x (AWS\_RedShift(x) \rightarrow \exists y (Storage\_Optimized\_Workload(y) \land
            suitableFor(x,y))
            DL: suitableFor(AWS_RedShift, Storage_Optimized_Workload)
463. AWS_EKS is suitable for General_Purpose_Workload
            FOL: \forall x (AWS_EKS(x) \rightarrow \exists y (General_Purpose_Workload(y) \land \exists y (General_Purpose_Workload(y)) \land \exists y (G
            suitableFor(x,y))
            DL: suitableFor(AWS_EKS, General_Purpose_Workload)
464. Azure_VM is suitable for Memory_Optimized_Workload
            FOL: \forall x(Azure_VM(x) \rightarrow \exists y(Memory_Optimized_Workload(y) \land \exists y(Memory_Optimized_Workload(y)))
            suitableFor(x,y))
            DL: suitableFor(Azure_VM, Memory_Optimized_Workload)
465. Azure_VM is suitable for High_CPU_Workload
            FOL: \forall x (Azure_VM(x) \rightarrow \exists y (High_CPU_Workload(y) \land \exists y (High_CPU_Workload(y)))
            suitableFor(x,y))
            DL: suitableFor(Azure_VM, High_CPU_Workload)
466. Azure_VNet is suitable for General_Purpose_Workload
            FOL: \forall x (Azure_VNet(x) \rightarrow \exists y (General_Purpose_Workload(y)) \land
            suitableFor(x,y))
            DL: suitableFor(Azure_VNet, General_Purpose_Workload)
467. Azure_LB is suitable for General_Purpose_Workload
            FOL: \forall x (Azure\_LB(x) \rightarrow \exists y (General\_Purpose\_Workload(y)) \land
            suitableFor(x,y))
```

```
DL: suitableFor(Azure_LB, General_Purpose_Workload)
468. Azure_VPN_Gateway is suitable for Network_Optimized_Workload
                 FOL: \forall x (Azure_VPN_Gateway(x) \rightarrow \exists y (Network_Optimized_Workload(y))
                \land suitableFor(x,y))
                 DL: suitableFor(Azure_VPN_Gateway, Network_Optimized_Workload)
469. Azure_DNS is suitable for General_Purpose_Workload
                 FOL: \forall x(Azure\_DNS(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land \exists y(General\_Purpose\_Workload(y)) \land \exists y(General\_Purpose\_Worklo
                 suitableFor(x,y))
                 DL: suitableFor(Azure_DNS, General_Purpose_Workload)
470. Azure_SQL_Database is suitable for Memory_Optimized_Workload
                 FOL: \forall x (Azure\_SQL\_Database(x) \rightarrow \exists y (Memory\_Optimized\_Workload(y))
                \land suitableFor(x,y))
                DL: suitableFor(Azure_SQL_Database, Memory_Optimized_Workload)
471. Azure_AKS is suitable for General_Purpose_Workload
                 FOL: \forall x(Azure\_AKS(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land \exists y(General\_Purpose\_Workload(y)) \land \exists y(General\_Purpose\_Worklo
                 suitableFor(x,y))
                 DL: suitableFor(Azure_AKS, General_Purpose_Workload)
472. Azure_Scale_Sets is suitable for High_CPU_Workload
                 FOL: \forall x(Azure\_Scale\_Sets(x) \rightarrow \exists y(High\_CPU\_Workload(y) \land \exists y(High\_CPU\_Workload(y)) \land y(High\_CPU\_W
                 suitableFor(x,y))
                 DL: suitableFor(Azure_Scale_Sets, High_CPU_Workload)
473. Azure_Functions is suitable for General_Purpose_Workload
                 FOL: \forall x(Azure\_Functions(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land
                 suitableFor(x,y))
                 DL: suitableFor(Azure_Functions, General_Purpose_Workload)
474. GCP_Compute_Engine is suitable for High_CPU_Workload
                 FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (High\_CPU\_Workload(y) \land \exists y (High\_CPU\_Workload(y)))
                 suitableFor(x,y))
                 DL: suitableFor(GCP_Compute_Engine, High_CPU_Workload)
475. GCP_Compute_Engine is suitable for General_Purpose_Workload
                 FOL: \forall x (GCP\_Compute\_Engine(x) \rightarrow \exists y (General\_Purpose\_Workload(y) \land
                 suitableFor(x,y))
```

```
DL: suitableFor(GCP_Compute_Engine, General_Purpose_Workload)
476. GCS_Bucket is suitable for Storage_Optimized_Workload
                  FOL: \forall x (GCS\_Bucket(x) \rightarrow \exists y (Storage\_Optimized\_Workload(y) \land \exists y (Storage\_Optimized\_Workload(y)) \land \exists y (
                  suitableFor(x,y)
                  DL: GsuitableFor(GCS_Bucket, Storage_Optimized_Workload)
477. GCP_GKE is suitable for General_Purpose_Workload
                  FOL: \forall x (GCP\_GKE(x) \rightarrow \exists y (General\_Purpose\_Workload(y) \land \exists y (General\_Purpose\_Workload(y) \land \exists y (General\_Purpose\_Workload(y)) \land \exists y (Ge
                  suitableFor(x,y))
                  DL: suitableFor(GCP_GKE, General_Purpose_Workload)
478. IBM_Cloud_Virtual_Server is suitable for High_CPU_Workload
                  FOL: \forall x (IBM\_Cloud\_Virtual\_Server(x) \rightarrow \exists y (High\_CPU\_Workload(y) \land \exists y (High\_CPU\_Workload(y)) \land \exists y 
                  suitableFor(x,y)
                  DL: suitableFor(IBM_Cloud_Virtual_Server, High_CPU_Workload)
479. IBM_Cloud_Virtual_Server is suitable for
                 General_Purpose_Workload
                  FOL: ∀x(IBM_Cloud_Virtual_Server(x) →
                   \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
                  DL: suitableFor(IBM_Cloud_Virtual_Server,
                 General_Purpose_Workload)
480. IBM_Cloud_Functions is suitable for General_Purpose_Workload
                  FOL: \forall x (IBM\_Cloud\_Functions(x) \rightarrow \exists y (General\_Purpose\_Workload(y))
                 \land suitableFor(x,y))
                 DL: suitableFor(IBM_Cloud_Functions, General_Purpose_Workload)
481. IBM_Cloud_Object_Storage is suitable for
                  Storage_Optimized_Workload
                  FOL: \forall x(IBM\_Cloud\_Object\_Storage(x) \rightarrow
                   \exists y (Storage_Optimized_Workload(y) \land suitableFor(x,y))
                  DL: suitableFor(IBM_Cloud_Object_Storage,
                  Storage_Optimized_Workload)
482. IBM_Cloud_Kubernetes_Service is suitable for
                  General_Purpose_Workload
                  FOL: ∀x(IBM_Cloud_Kubernetes_Service(x) →
                   \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
                  DL: suitableFor(IBM_Cloud_Kubernetes_Service,
```

```
General_Purpose_Workload)
483. IBM_Cloud_Database is suitable for Memory_Optimized_Workload
        FOL: \forall x (IBM\_Cloud\_Database(x) \rightarrow \exists y (Memory\_Optimized\_Workload(y))
       \land suitableFor(x,y))
        DL: suitableFor(IBM_Cloud_Database, Memory_Optimized_Workload)
484. IBM_Cloud_Load_Balancer is suitable for
        General Purpose Workload
        FOL: ∀x(IBM_Cloud_Load_Balancer(x) →
        \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
        DL: suitableFor(IBM_Cloud_Load_Balancer,
       General_Purpose_Workload)
485. IBM_Cloud_VPC is suitable for General_Purpose_Workload
        FOL: \forall x (IBM\_Cloud\_VPC(x) \rightarrow \exists y (General\_Purpose\_Workload(y) \land \exists y (General\_Purpose\_Workload(y)) \land 
        suitableFor(x,y))
        DL: suitableFor(IBM_Cloud_VPC, General_Purpose_Workload)
486. IBM_Cloud_Transit_Gateway is suitable for
        Network_Optimized_Workload
        FOL: \forall x(IBM\_Cloud\_Transit\_Gateway(x) \rightarrow
        \exists y (\text{Network\_Optimized\_Workload}(y) \land \text{suitableFor}(x,y))
        DL: suitableFor(IBM_Cloud_Transit_Gateway,
        Network_Optimized_Workload)
487. Oracle_Cloud_VM is suitable for High_CPU_Workload
        FOL: \forall x(Oracle\_Cloud\_VM(x) \rightarrow \exists y(High\_CPU\_Workload(y) \land \exists y(High\_CPU\_Workload(y)))
        suitableFor(x,y))
        DL: suitableFor(Oracle_Cloud_VM, High_CPU_Workload)
488. Oracle_Cloud_VM is suitable for General_Purpose_Workload
        FOL: \forall x(0racle\_Cloud\_VM(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land
        suitableFor(x,y))
        DL: suitableFor(Oracle_Cloud_VM, General_Purpose_Workload)
489. Oracle Cloud Functions is suitable for
        General_Purpose_Workload
        FOL: \forall x(Oracle\_Cloud\_Functions(x) \rightarrow
         \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
```

```
DL: suitableFor(Oracle_Cloud_Functions, General_Purpose_Workload)
490. Oracle_Cloud_Block_Storage is suitable for
   Storage_Optimized_Workload
   FOL: \forall x(Oracle\_Cloud\_Block\_Storage(x) \rightarrow
   \exists y (Storage_Optimized_Workload(y) \land suitableFor(x,y))
   DL: suitableFor(Oracle_Cloud_Block_Storage,
   Storage_Optimized_Workload)
491. Oracle_Cloud_Object_Storage is suitable for
   Storage_Optimized_Workload
   FOL: \forall x(0racle\_Cloud\_Object\_Storage(x) \rightarrow
   \exists y (Storage_Optimized_Workload(y) \land suitableFor(x,y))
   DL: suitableFor(Oracle_Cloud_Object_Storage,
   Storage_Optimized_Workload)
492. Oracle_Cloud_Kubernetes_Engine is suitable for
   General_Purpose_Workload
   FOL: ∀x(Oracle_Cloud_Kubernetes_Engine(x) →
   \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
   DL: suitableFor(Oracle_Cloud_Kubernetes_Engine,
   General_Purpose_Workload)
493. Oracle_Cloud_Database_Service is suitable for
   Memory_Optimized_Workload
   FOL: \forall x(Oracle\_Cloud\_Database\_Service(x) \rightarrow
   \exists y (Memory_Optimized_Workload(y) \land suitableFor(x,y))
   DL: suitableFor(Oracle_Cloud_Database_Service,
   Memory_Optimized_Workload)
494. Oracle Cloud Load Balancer is suitable for
   General_Purpose_Workload
   FOL: ∀x(Oracle_Cloud_Load_Balancer(x) →
   \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
   DL: suitableFor(AWS_EC2, General_Purpose_Workload)
495. Oracle_Cloud_VCN is suitable for Network_Optimized_Workload
   FOL: \forall x(Oracle\_Cloud\_VCN(x) \rightarrow \exists y(Network\_Optimized\_Workload(y) \land
   suitableFor(x,y))
   DL: suitableFor(Oracle_Cloud_VCN, Network_Optimized_Workload)
```

```
496. Alibaba_ECS is suitable for High_CPU_Workload
                      FOL: \forall x (Alibaba\_ECS(x) \rightarrow \exists y (High\_CPU\_Workload(y) \land \exists y (High\_CPU\_Workload(y)) \land \exists y (High\_CPU\_Wor
                      suitableFor(x,y))
                      DL: suitableFor(Alibaba_ECS, High_CPU_Workload)
497. Alibaba_ECS is suitable for General_Purpose_Workload
                      FOL: \forall x(Alibaba\_ECS(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land \exists y(General\_Purpose\_Workload(y)) \land \exists y(General\_Purpose\_Work
                      suitableFor(x,y)
                      DL: suitableFor(Alibaba_ECS, General_Purpose_Workload)
498. Alibaba_Function_Compute is suitable for
                      General_Purpose_Workload
                      FOL: \forall x (Alibaba_Function_Compute(x) \rightarrow
                       ∃y(General_Purpose_Workload(y) ∧ suitableFor(x,y))
                      DL: suitableFor(Alibaba_Function_Compute,
                     General_Purpose_Workload)
499. Alibaba_OSS is suitable for Storage_Optimized_Workload
                      FOL: \forall x(Alibaba_OSS(x) \rightarrow \exists y(Storage_Optimized_Workload(y) \land
                      suitableFor(x,y)
                      DL: suitableFor(Alibaba_OSS, Storage_Optimized_Workload)
500. Alibaba Container Service is suitable for
                      General_Purpose_Workload
                      FOL: ∀x(Alibaba_Container_Service(x) →
                       \exists y (General\_Purpose\_Workload(y) \land suitableFor(x,y))
                      DL: suitableFor(Alibaba_Container_Service,
                     General_Purpose_Workload)
501. Alibaba_ApsaraDB is suitable for Memory_Optimized_Workload
                      FOL: \forall x(Alibaba\_ApsaraDB(x) \rightarrow \exists y(Memory\_Optimized\_Workload(y) \land
                      suitableFor(x,y)
                      DL: suitableFor(Alibaba_ApsaraDB, Memory_Optimized_Workload)
502. Alibaba_SLB is suitable for General_Purpose_Workload
                      FOL: \forall x(Alibaba\_SLB(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land \exists y(General\_Purpose\_Workload(y)) \land \exists y(General\_Purpose\_Work
                      suitableFor(x,y))
                      DL: suitableFor(Alibaba_SLB, General_Purpose_Workload)
503. Alibaba_VPC is suitable for General_Purpose_Workload
                      FOL: \forall x(Alibaba\_VPC(x) \rightarrow \exists y(General\_Purpose\_Workload(y) \land \exists y(General\_Purpose\_Workload(y)) \land \exists y(General\_Purpose\_Work
```

```
suitableFor(x,y))
             DL: suitableFor(Alibaba_VPC, General_Purpose_Workload)
504. Alibaba_SAG is suitable for Network_Optimized_Workload
             FOL: \forall x(Alibaba\_SAG(x) \rightarrow \exists y(Network\_Optimized\_Workload(y) \land
             suitableFor(x,y)
             DL: suitableFor(Alibaba_SAG, Network_Optimized_Workload)
505. Cloud Service has component Standardized Term
             FOL: \forall x(Cloud\_Service(x) \rightarrow \exists y(Standardized\_Term(y) \land \exists y(Standardized\_Term(y)))
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.Standardized_Term
506. Cloud_Service has component Interoperability_Mapping
             FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (Interoperability\_Mapping(y) \land
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.Interoperability_Mapping
507. Cloud_Service has component API_Standard
             FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (API\_Standard(y) \land \exists y (API\_Standard(y)))
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.API_Standard
508. Cloud_Service has component Workload_Profile
             FOL: \forall x(Cloud\_Service(x) \rightarrow \exists y(Workload\_Profile(y) \land \exists y(Workload\_Profile(y)) \land \exists y(Workload\_Profil
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.Workload_Profile
509. Cloud_Service has component Recommendation_Engine
             FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (Recommendation\_Engine(y) \land
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.Recommendation_Engine
510. Cloud_Service has component Optimization_Goal
             FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (Optimization\_Goal(y) \land \exists y (Optimization\_Goal(y)))
             hasComponent(x,y))
             DL: Cloud_Service ⊆ ∃hasComponent.Optimization_Goal
511. Cloud_Service has component Composite_Function
             FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (Composite\_Function(y) \land \exists y (Composite\_Function(y)) \land \exists y (C
             hasComponent(x,y))
```

```
DL: Cloud_Service ⊆ ∃hasComponent.Composite_Function
512. Cloud_Service has component Service_Component
                     FOL: \forall x(Cloud\_Service(x) \rightarrow \exists y(Service\_Component(y) \land \exists y(Service\_Component(y)) \land \exists
                     hasComponent(x,y))
                     DL: Cloud_Service ⊆ ∃hasComponent.Service_Component
513. Cloud_Service has component Composition_Pattern
                     FOL: \forall x (Cloud\_Service(x) \rightarrow \exists y (Composition\_Pattern(y) \land \exists y (Composition\_Pattern(y)) \land \exists
                     hasComponent(x,y))
                     DL: Cloud_Service ⊆ ∃hasComponent.Composition_Pattern
514. ML_Based_Selector is optimized for Perfomance_Maximization
                     FOL: \forall x (ML\_Based\_Selector(x) \rightarrow \exists y (Perfomance\_Maximization(y) \land
                     optimizedFor(x,y))
                     DL: ML_Based_Selector ⊆ ∃optimizedFor.Perfomance_Maximization
515. Rules_Based_Selector is optimized for Cost_Optimization
                     FOL: \forall x (Rules\_Based\_Selector(x) \rightarrow \exists y (Cost\_Optimization(y)) \land
                     optimizedFor(x,y))
                     DL: Rules_Based_Selector ⊆ ∃optimizedFor.Cost_Optimization
```

Cluster 10: Composite Services & Patterns

```
516. Hybrid_Storage_Service is an instance of Composite_Service
   FOL: Composite_Service(Hybrid_Storage_Service)
   DL: Composite_Service(Hybrid_Storage_Service)

517. Multi_Cloud_Web_App_Stack is an instance of Composite_Service
   FOL: Composite_Service(Multi_Cloud_Web_App_Stack)
   DL: Composite_Service(Multi_Cloud_Web_App_Stack)

518. Frontend_Hosting is an instance of Service_Component
   FOL: Service_Component(Frontend_Hosting)
```

```
DL: Service_Component(Frontend_Hosting)
519. Backend_Compute is an instance of Service_Component
             FOL: Service_Component(Backend_Compute)
             DL: Service_Component(Backend_Compute)
520. Databased_Layer is an instance of Service_Component
             FOL: Service_Component(Databased_Layer)
             DL: Service_Component(Databased_Layer)
521. Multi_Cloud_Web_App_Stack is composed of Frontend_Hosting
             FOL: \forall x (Multi\_Cloud\_Web\_App\_Stack(x) \rightarrow \exists y (Frontend\_Hosting(y) \land \exists y (Frontend\_Hosting(y)) \land \exists y (Frontend\_Ho
             composedOf(x,y))
             DL: composedOf(Multi_Cloud_Web_App_Stack, Frontend_Hosting)
522. Multi_Cloud_Web_App_Stack is composed of Backend_Compute
             FOL: \forall x (Multi\_Cloud\_Web\_App\_Stack(x) \rightarrow \exists y (Backend\_Compute(y)) \land
            composedOf(x,y))
             DL: composedOf(Multi_Cloud_Web_App_Stack, Backend_Compute)
523. Multi_Cloud_Web_App_Stack is composed of Databased_Layer
             FOL: \forall x (Multi\_Cloud\_Web\_App\_Stack(x) \rightarrow \exists y (Databased\_Layer(y) \land \exists y (Databased\_Layer(y)) \land \exists y (D
             composedOf(x,y))
             DL: composedOf(Multi_Cloud_Web_App_Stack, Databased_Layer)
524. Hybrid_Storage_Service is composed of Databased_Layer
             FOL: \forall x(Hybrid\_Storage\_Service(x) \rightarrow \exists y(Databased\_Layer(y) \land
             composedOf(x,y))
             DL: composedOf(Hybrid_Storage_Service, Databased_Layer)
525. Multi_Cloud_Web_App_Stack follows pattern
             Burst_To_Cloud_Pattern
             FOL: \forall x (Multi\_Cloud\_Web\_App\_Stack(x) \rightarrow
             ∃y(Burst_To_Cloud_Pattern(y) ∧ followsPattern(x,y))
             DL: followsPattern(Multi_Cloud_Web_App_Stack, Frontend_Hosting)
526. Hybrid_Storage_Service follows pattern
             Data_Replication_Pattern
             FOL: ∀x(Hybrid_Storage_Service(x) →
              \exists y (Data\_Replication\_Pattern(y) \land followsPattern(x,y))
```

```
DL: followsPattern(Multi_Cloud_Web_App_Stack,
Data_Replication_Pattern)
```

Cluster 11: Deployment & Tenancy

```
527. Public_Cloud is a Cloud_Deployment_Model
         FOL: \forall x (Public\_Cloud(x) \rightarrow Cloud\_Deployment\_Model(x))
         DL: Public_Cloud ⊆ Cloud_Deployment_Model
528. Private_Cloud is a Cloud_Deployment_Model
         FOL: \forall x (Private\_Cloud(x) \rightarrow Cloud\_Deployment\_Model(x))
         DL: Private_Cloud ⊆ Cloud_Deployment_Model
529. Hybrid_Cloud is a Cloud_Deployment_Model
         FOL: \forall x (Hybrid\_Cloud(x) \rightarrow Cloud\_Deployment\_Model(x))
         DL: Hybrid_Cloud ⊆ Cloud_Deployment_Model
530. Community_Cloud is a Cloud_Deployment_Model
         FOL: \forall x (Community\_Cloud(x) \rightarrow Cloud\_Deployment\_Model(x))
         DL: Community_Cloud ⊆ Cloud_Deployment_Model
531. Public_Cloud is operated by a Cloud_Service_Provider
         FOL: \forall x (Public\_Cloud(x) \rightarrow \exists y (Cloud\_Service\_Provider(y) \land \exists y (Cloud\_Service\_Provider(y)))
         isOperatedBy(x,y)
         DL: Public_Cloud ⊆ ∃isOperatedBy.Cloud_Service_Provider
532. Private_Cloud is operated by a Cloud_Service_Provider
         FOL: \forall x (Private\_Cloud(x) \rightarrow \exists y (Cloud\_Service\_Provider(y) \land \exists y (Cloud\_Service\_Provider(y)))
         isOperatedBy(x,y)
         DL: Private_Cloud ⊆ ∃isOperatedBy.Cloud_Service_Provider
533. Hybrid_Cloud is operated by a Cloud_Service_Provider
         FOL: \forall x (Hybrid\_Cloud(x) \rightarrow \exists y (Cloud\_Service\_Provider(y) \land \exists y (Cloud\_Service\_Provider(y)) \land \exists y (Cloud\_
         isOperatedBy(x,y)
         DL: Hybrid_Cloud ⊆ ∃isOperatedBy.Cloud_Service_Provider
```

```
534. Community_Cloud is operated by a Cloud_Service_Provider
   FOL: \forall x (Community\_Cloud(x) \rightarrow \exists y (Cloud\_Service\_Provider(y)) \land
   isOperatedBy(x,y)
   DL: Community_Cloud ⊆ ∃isOperatedBy.Cloud_Service_Provider
535. Public_Cloud has a tenancy Multi_Tenant
   FOL: \forall x (Public\_Cloud(x) \rightarrow \exists y (Multi\_Tenant(y) \land hasTenancy(x,y))
   DL: Public_Cloud ⊆ ∃hasTenancy.Multi_Tenant
536. Private_Cloud has a tenancy Single_Tenant
   FOL: \forall x (Private\_Cloud(x) \rightarrow \exists y (Single\_Tenant(y) \land hasTenancy(x,y))
   DL: Private_Cloud ⊆ ∃hasTenancy.Single_Tenant
537. Private_Cloud is managed by an Organization
   FOL: \forall x (Private\_Cloud(x) \rightarrow \exists y (Organization(y) \land isManagedBy(x,y))
   DL: Private_Cloud ⊆ ∃isManagedBy.Organization
538. Private_Cloud is built by an Organization
   FOL: \forall x (Private\_Cloud(x) \rightarrow \exists y (Organization(y) \land isBuiltBy(x,y))
   DL: Private_Cloud ⊆ ∃isBuiltBy.Organization
539. Private_Cloud is owned by an Organization
   FOL: \forall x (Private\_Cloud(x) \rightarrow \exists y (Organization(y) \land isOwnedBy(x,y))
   DL: Private_Cloud ⊆ ∃isOwnedBy.Organization
540. Every Compute_Resource must have a memory value
   FOL: \forall x (Compute\_Resource(x) \rightarrow \exists m (hasMemory(x, m)))
   DL: Compute_Resource ⊆ ∃hasMemory.xsd:decimal
541. High_Memory_Resource instances have memory greater than or
   equal to 64 GB
   FOL: \forall x (High\_Memory\_Resource(x) \rightarrow hasMemory(x, m) \land m \ge 64)
   DL: High_Memory_Resource \subseteq \exists hasMemory.(xsd:integer[\ge 64])
542. A resource cannot have more than one memory value
   FOL: \forall x, m1, m2((hasMemory(x, m1) \land hasMemory(x, m2)) \rightarrow m1 = m2)
   DL: Functional(hasMemory)
543. AWS_EC2 has 16 GB memory
   FOL: hasMemory(AWS_EC2, 16)
```

```
DL: hasMemory(AWS_EC2 16)
544. A resource with less than 8 GB of memory is a
   Low_Memory_Resource
   FOL: \forall x (Low\_Memory\_Resource(x) \rightarrow hasMemory(x, m) \land m < 8)
   DL: Low_Memory_Resource \subseteq \exists hasMemory.(xsd:integer[<8])
545. Every Compute_Resource must declare its CPU count
   FOL: \forall x (Compute\_Resource(x) \rightarrow \exists c (hasCPUCount(x, c)))
   DL: Compute_Resource ⊆ ∃hasCPUCount.xsd:integer
546. A resource with more than 16 CPUs is considered
   Multi Core Resource
   FOL: \forall x (hasCPUCount(x, c) \land c > 16 \rightarrow Multi\_Core\_Resource(x))
   DL: \exists hasCPUCount.(xsd:integer[> 16]) \subseteq Multi_Core_Resource
547. AWS_EC2 has 8 CPUs
   FOL: hasCPUCount(AWS_EC2, 8)
   DL: hasCPUCount(AWS_EC2, 8)
548. CPU count is a Functional
   FOL: \forall x, c1, c2((hasCPUCount(x, c1) \land hasCPUCount(x, c2)) \rightarrow c1 =
   c2)
   DL: Functional(hasCPUCount)
549. No Storage_Resource should have a CPU count
   FOL: \forall x (Storage\_Resource(x) \rightarrow \neg \exists c (hasCPUCount(x, c)))
   DL: Storage_Resource ⊆ ¬∃hasCPUCount.xsd:integer
550. Every Network_Resource must have bandwidth
   FOL: \forall x (Network\_Resource(x) \rightarrow \exists b (hasBandwidth(x, b)))
   DL: Network Resource ⊆ ∃hasBandwidth.xsd:decimal
551. High_Bandwidth_Resource has at least 1000 Mbps bandwidth
   FOL: \forall x (High\_Bandwidth\_Resource(x) \rightarrow hasBandwidth(x, b) \land b \ge
   DL: High_Bandwidth_Resource \subseteq \exists hasBandwidth.(xsd:decimal[>=
   10001)
552. hasBandwidth is Functional
   FOL: \forall x, b1, b2((hasBandwidth(x, b1) \land hasBandwidth(x, b2)) \rightarrow b1 =
```

```
b2)
   DL: Functional(hasBandwidth)
553. Google_Network has 2000 Mbps bandwidth
   FOL: hasBandwidth(Google_Network, 2000)
   DL: hasBandwidth(Google_Network, 2000)
554. Any resource with bandwidth less than 100 is a
   Low Bandwidth Resource
   FOL: \forall x (hasBandwidth(x, b) \land b < 100 \rightarrow
   Low_Bandwidth_Resource(x))
   DL: \exists has Bandwidth.(xsd:decimal[< 100]) \subseteq Low_Bandwidth_Resource
555. Every Network_Resource must define its latency
   FOL: \forall x (Network\_Resource(x) \rightarrow \exists 1 (hasLatency(x, 1)))
   DL: Network_Resource ⊆ ∃hasLatency.xsd:decimal
556. Low_Latency_Service has latency less than 10 ms
   FOL: \forall x (hasLatency(x, 1) \land 1 < 10 \rightarrow Low_Latency_Service(x))
   DL: \exists hasLatency.(xsd:decimal[< 10]) \subseteq Low_Latency_Service
557. hasLatency is Functional
   FOL: \forall x, 11, 12((hasLatency(x, 11) \land hasLatency(x, 12)) \rightarrow 11 = 12)
   DL: Functional(hasLatency)
558. Azure_Network has a latency of 8 ms
   FOL: hasLatency(Azure_Network, 8)
   DL: hasLatency(Azure_Network, 8)
559. Any Service with latency above 100 ms is High_Latency_Service
   FOL: \forall x (hasLatency(x, 1) \land 1 > 100 \rightarrow High\_Latency\_Service(x))
   DL: \exists hasLatency.(xsd:decimal[> 100]) \subseteq High_Latency_Service
560. Every Cloud_Service must have a price
   FOL: \forall x (Cloud\_Service(x) \rightarrow \exists p (hasPrice(x, p)))
   DL: Cloud Service ⊆ ∃hasPrice.xsd:decimal
561. hasPrice is Functional
   FOL: \forall x, p1, p2((hasPrice(x, p1) \land hasPrice(x, p2)) \rightarrow p1 = p2)
   DL: Functional(hasPrice)
```

```
562. AWS_EC2 has a price of R1.80/hour
   FOL: hasPrice(AWS_EC2, 1.80
   DL: hasPrice(AWS_EC2, 1.80)
563. Free_Service has price equal to 0
   FOL: \forall x (hasPrice(x, p) \land p = 0 \rightarrow Free\_Service(x))
   DL: \exists hasPrice.(xsd:decimal[= 0]) \subseteq Free_Service
564. Premium Service costs more than R8.50/hour
   FOL: \forall x (hasPrice(x, p) \land p > 8.50 \rightarrow Premium\_Service(x))
   DL: ∃hasPrice.(xsd:decimal[> 8.50]) ⊆ Premium_Service
565. Each Cloud_Service must have an SLA value (as a percentage)
   FOL: \forall x (Cloud\_Service(x) \rightarrow \exists s (hasSLA(x, s)))
   DL: Cloud_Service ⊆ ∃hasSLA.xsd:decimal
566. Mission_Critical_Service must have SLA ≥ 99.99%
   FOL: \forall x (Mission\_Critical\_Service(x) \rightarrow hasSLA(x, s) \land s \ge 99.99)
   DL: Mission_Critical_Service \subseteq \exists hasSLA.(xsd:decimal[>= 99.99])
567. hasSLA is Functional_Data_Property
   FOL: \forall x, s1, s2((hasSLA(x, s1) \land hasSLA(x, s2)) \rightarrow s1 = s2)
   DL: Functional_Data_Property(hasSLA)
568. Azure_Functions has SLA of 99.95%
   FOL: hasSLA(Azure_Functions, 99.95)
   DL: hasSLA(Azure_Functions, 99.95)
569. Services with SLA less than 90% are Non_Standard_Services
   FOL: \forall x(hasSLA(x, s) \land s < 90 \rightarrow Non_Standard_Service(x))
   DL: \exists hasSLA.(xsd:decimal[< 90]) \subseteq Non_Standard_Service
570. Every Serverless_Compute_Service must define request duration
   FOL: \forall x (Serverless\_Compute\_Service(x) \rightarrow \exists d(hasRequestDuration(x,
   d)))
   DL: Serverless_Compute_Service ⊆ ∃hasRequestDuration.xsd:decimal
571. Long_Duration_Service has request duration above 30 seconds
   FOL: \forall x (hasRequestDuration(x, d) \land d > 30 \rightarrow
   Long_Duration_Service(x))
   DL: \exists hasRequestDuration.(xsd:decimal[> 30]) \subseteq
```

```
Long_Duration_Service
572. AWS_Lambda has duration of 5 seconds
   FOL: hasRequestDuration(AWS_Lambda, 5)
   DL: hasRequestDuration(AWS_Lambda, 5)
573. hasRequestDuration is Functional
   FOL: \forall x, d1, d2((hasRequestDuration(x, d1) \land hasRequestDuration(x, d1))
   d2)) \rightarrow d1 = d2)
   DL: Functional(hasRequestDuration)
574. Services with duration ≤ 1 second are considered
  Fast Execution Service
  FOL: \forall x (hasRequestDuration(x, d) \land d \le 1 \rightarrow
  Fast_Execution_Service(x))
  DL: \exists has Request Duration. (xsd:decimal[<= 1]) \subseteq
  Fast_Execution_Service
575. Every Storage_Resource has a defined tier
   FOL: \forall x (Storage\_Resource(x) \rightarrow \exists t (hasStorageTier(x, t)))
   DL: Storage_Resource ⊆ ∃hasStorageTier.xsd:string
576. Archive_Storage must have tier equal to "Archive"
   FOL: \forall x (hasStorageTier(x, t) \land t = "Archive" \rightarrow
  Archive_Storage(x))
  DL: ∃hasStorageTier.(xsd:string[= "Archive"]) ⊆ Archive_Storage
577. hasStorageTier is Functional
  FOL: \forall x \forall t 1 \forall t 2 ((hasStorageTier(x, t1) \land hasStorageTier(x, t2))
  \rightarrow t1 = t2)
  DL: Functional(hasStorageTier)
578. AWS_S3_Glacier has tier "Cold"
   FOL: hasStorageTier(AWS_S3_Glacier, "Cold")
   DL: hasStorageTier(AWS_S3_Glacier, "Cold")
```

```
579. Tiers labeled "Hot" are for frequently accessed data
  FOL: \forall x (hasStorageTier(x, t) \land t = "Hot" \rightarrow
  Hot_Access_Storage(x))
  DL: ∃hasStorageTier.(xsd:string[= "Hot"]) ⊆ Hot_Access_Storage
580. Every Serverless_Compute_Service may have cold start latency
  FOL: ∀x(Serverless_Compute_Service(x) →
   \exists c(hasColdStartLatency(x, c)))
  DL: Serverless_Compute_Service ⊆
   ∃ hasColdStartLatency.xsd:decimal
581. AWS_Lambda has cold start latency of 200 ms
  FOL: hasColdStartLatency(AWS_Lambda, 200)
  DL: hasColdStartLatency(AWS_Lambda, 200)
582. hasColdStartLatency is Functional
  FOL: \forall x, c1, c2((hasColdStartLatency(x, c1)) \land
  hasColdStartLatency(x, c2)) \rightarrow c1 = c2)
  DL: Functional(hasColdStartLatency)
583. If latency is less than 100 ms, then it is Fast_Start_Service
  FOL: \forall x (hasColdStartLatency(x, 1) \land 1 < 100 \rightarrow
  Fast_Start_Service(x))
  DL: ∃hasColdStartLatency.(xsd:decimal[< 100]) ⊆
  Fast Start Service
584. Services with cold start latency greater than 300 ms are
  Slow_Start_Service
  FOL: \forall x (hasColdStartLatency(x, 1) \land 1 > 300 \rightarrow
  Slow_Start_Service(x))
  DL: ∃hasColdStartLatency.(xsd:decimal[> 300]) ⊆
  Slow Start Service
585. Every Cloud_Application must have a memory consumption value
  FOL: \forall x (Cloud\_Application(x) \rightarrow \exists m (hasMemoryConsumption(x, m)))
  DL: Cloud_Application ⊆ ∃hasMemoryConsumption.xsd:decimal
586. Application_X consumes 512 MB of memory
  FOL: hasMemoryConsumption(Application_X, 512)
  DL: hasMemoryConsumption(Application_X, 512)
```

```
587. hasMemoryConsumption is Functional_Data_Property
   FOL: \forall x, m1, m2((hasMemoryConsumption(x, m1)) \land
   has Memory Consumption(x, m2)) \rightarrow m1 = m2)
   DL: Functional_Data_Property(hasMemoryConsumption)
588. Applications consuming more than 1024 MB are High_Memory_App
   FOL: \forall x (hasMemoryConsumption(x, m) \land m > 1024 \rightarrow
  High_Memory_App(x))
   DL: \exists has Memory Consumption. (xsd:decimal[> 1024]) \subseteq
   High_Memory_App
589. Applications with memory consumption below 256 MB are
  Lightweight_App
  FOL: \forall x (has Memory Consumption(x, m) \land m < 256 \rightarrow
  Lightweight\_App(x)
  DL: ∃hasMemoryConsumption.(xsd:decimal[< 256]) ⊆ Lightweight_App
590. Each Network_Resource must have network throughput defined
   FOL: \forall x (Network\_Resource(x) \rightarrow \exists t (hasNetworkThroughput(x, t)))
   DL: Network_Resource ⊆ ∃hasNetworkThroughput.xsd:decimal
591. High_Throughput_Network has throughput above 1 Gbps
   FOL: \forall x (has Network Throughput(x, t) \land t > 1000 \rightarrow
  High_Throughput_Network(x))
   DL: ∃hasNetworkThroughput.(xsd:decimal[> 1000]) ⊆
  High_Throughput_Network
592. hasNetworkThroughput is Functional
   FOL: \forall x, t1, t2((hasNetworkThroughput(x, t1)) \land
   hasNetworkThroughput(x, t2)) \rightarrow t1 = t2)
   DL: Functional(hasNetworkThroughput)
593. AWS_Enhanced_Networking has throughput of 10 Gbps
   FOL: hasNetworkThroughput(AWS_Enhanced_Networking, 10000)
   DL: hasNetworkThroughput(AWS_Enhanced_Networking, 10000)
594. Services with throughput below 100 Mbps are Low_Throughput
   FOL: \forall x (hasNetworkThroughput(x, t) \land t < 100 \rightarrow
   Low_Throughput(x))
   DL: \exists hasNetworkThroughput.(xsd:decimal[< 100]) \subseteq Low_Throughput
```

Literature Review

This literature review is aimed at laying the ground-work for the Cloud Computing Ontology, thereby providing comparisons with existing ontologies in the field, and it will also outline the benefits and challenges of the Cloud Computing Ontology.

One specific type of distributed computing is cloud computing. Without requiring physical storage, it presents autonomous computer resources that may be dispersed and scaled dynamically as needed. Users can utilize them whenever they want and can access them via the internet (Mahmood et al., 2013).

According to NIST, cloud computing is a model that makes it possible to have easy, anytime, anywhere network access to a range of shared resources. Additionally, it lists key features and several cloud computing service models (Mell et al., 2011). An alternative description of cloud computing is suggested by the International Organization for Standardization (ISO) standard, which provides standardized vocabulary and concepts. Interoperability among various cloud providers is emphasized by these conceptual frameworks (International Organization for Standardization, 2014).

Varying Ontologies on cloud computing have been proposed. An Ontology-Based Cloud Resource Management Framework (Gao and Yu, 2013) presents a formal ontology model that captures varying resource types such as network, storage, etc., it further provides semantic descriptions of these resources to facilitate the relations between user requests and available cloud services. It uses inference engines based on resource properties and Quality of Service requirements for automated decision-making. These studies demonstrate that ontology-based management can dramatically improve and optimize scheduling, dynamically as well (Li et al., 2013).

Cloud providers are well-structured and well-documented but lack the formalized ontological representation across services. Microsoft Azure has similar patterns to the ones appearing in Azure's virtual machines (Microsoft Azure, 2025), in which services are described as efficient and well structured but lack formal and semantic connection (Microsoft, 2025). Similarly in Amazon Web Services (AWS), the computer resources, storage and pricing models are well-described through varying documentation, but no unified semantic model has been implemented (Amazon Web Services, 2025). In Google, documentations such as Compute Engine and Cloud Storage provide technical descriptions, but like the above-mentioned Cloud providers, have no explicit or specialized ontology (Google Cloud, 2025).

Despite the advancements and futuristic researches made in Cloud Ontology Development, a variety of challenges persist. Service names are inconsistent, the capabilities and pricing differ, hence complicating ontology unification. Ontologies are unable to sufficiently adapt and be maintainable to the rapidly evolving and ever-so growing Cloud services in Cloud

Computing. The Ontological frameworks must handle the large-scale nature of the "clouds" (Microsoft, 2025; TechTrend, 2025).

References

- [1] Amazon Web Services, "Amazon EC2 Documentation," Amazon Web Services, 2025. [Online]. Available: https://docs.aws.amazon.com/ec2/. [Accessed: Apr. 1, 2025].
- [2] Microsoft Azure, "Azure Virtual Machines Documentation," Microsoft, 2025. [Online]. Available: https://learn.microsoft.com/azure/virtual-machines/. [Accessed: Apr. 10, 2025].
- [3] Google Cloud, "Compute Engine Documentation," Google Cloud, 2025. [Online]. Available: https://cloud.google.com/compute/docs. [Accessed: Apr. 10, 2025].
- [4] Amazon Web Services, "Amazon S3 Documentation," Amazon Web Services, 2025. [Online]. Available: https://docs.aws.amazon.com/s3/. [Accessed: Apr. 10, 2025].
- [5] Microsoft Azure, "Azure Blob Storage Documentation," Microsoft, 2025. [Online]. Available: https://learn.microsoft.com/azure/storage/blobs/. [Accessed: Apr. 10, 2025].
- [6] Google Cloud, "Cloud Storage Documentation," Google Cloud, 2025. [Online]. Available: https://cloud.google.com/storage/docs. [Accessed: Apr. 10, 2025].
- [7] Amazon Web Services, "AWS Pricing," Amazon Web Services, 2025. [Online]. Available: https://aws.amazon.com/pricing/. [Accessed: Apr. 10, 2025].
- [8] Microsoft Azure, "Azure Pricing," Microsoft, 2025. [Online]. Available: https://azure.microsoft.com/pricing/. [Accessed: Apr. 10, 2025].
- [9] Google Cloud, "Google Cloud Pricing," Google Cloud, 2025. [Online]. Available: https://cloud.google.com/pricing/. [Accessed: Apr. 10, 2025].
- [10] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (NIST Special Publication 800-145)," National Institute of Standards and Technology, Gaithersburg, MD, Sep. 2011.
- [11] International Organization for Standardization, "ISO/IEC 17788:2014 Information technology Cloud computing Overview and vocabulary," ISO, Geneva, Switzerland, 2014.
- [12] T. Erl, Z. Mahmood, and R. Puttini, "Cloud Computing: Concepts, Technology & Architecture, 2nd ed." Upper Saddle River, NJ: Prentice Hall, 2013.

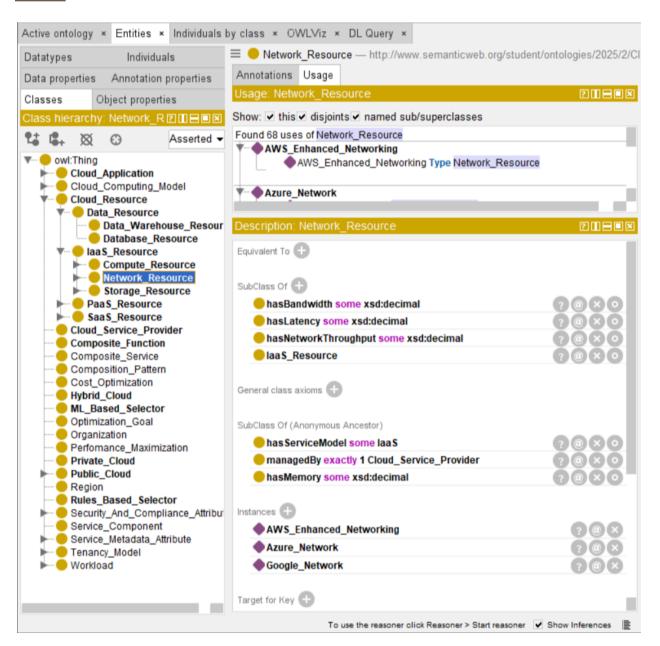
- [13] Y. J. Li, M. Gao, and Y. Yu, "An Ontology-based Cloud Resource Management Framework," IEEE Trans. Cloud Comput., vol. 1, no. 2, pp. 1–14, Apr.–Jun. 2013.
- [14] What is Vendor Management Benefit Process & Best Practices. TechTrend, 2025 Available: https://the-tech-trend.com/supply-chain-management/what-is-vendor-management-benefit-process-best-practices/ [Accessed: Apr. 1, 2025].
- [15] Cloud Standards Customer Council, "Interoperability and portability for cloud computing: A guide," Object Management Group, 2017.
- [16] R. Buyya, C. Vecchiola, and S. T. Selvi, "Mastering Cloud Computing: Foundations and Applications Programming." Burlington, MA, USA: Morgan Kaufmann, 2013.
- [17] D. Petcu et al., "Portable Cloud Applications–From Theory to Practice," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1417–1430, 2013.
- [18] G. Lewis, "The Role of Standards in Cloud-Computing Interoperability," *Software Engineering Institute*, Carnegie Mellon University, 2012.
- [19] DMTF, "Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP," *Distributed Management Task Force*, 2012.
- [20] OASIS, "Topology and Orchestration Specification for Cloud Applications (TOSCA)," Version 1.0, 2013.
- [21] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Key Challenges in Cloud Computing: Enabling the Future Internet of Services," *IEEE Internet Computing*, vol. 17, no. 4, pp. 18–25, 2013.
- [22] M. Villari, A. Puliafito, M. Fazio, and S. Dustdar, "Containers and Microservices for Fog Computing," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 10, 2018.
- [23] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to Enhance Cloud Architectures to Enable Cross-Federation," in *Proceedings of the 3rd International Conference on Cloud Computing*, IEEE, 2010.
- [24] E. P. Duarte Jr., C. Geyer, and L. C. E. de Bona, "A Systematic Review of Cloud Computing Ontologies," *Grid Computing*, vol. 16, pp. 279–296, 2018.
- [25] M. Comerio, E. D. Nitto, D. Petcu, and R. D. Calheiros, "Foundations of Cloud Service Selection," in *Service-Oriented and Cloud Computing*, Springer, 2014.

- [26] F. Zhang et al., "Cloud Computing Resource Scheduling and a Survey of Its Evolution," *Journal of Cloud Computing*, vol. 9, no. 1, p. 20, 2020.
- [27] T. Metsch, A. Edmonds, and A. Papaspyrou, "Open Cloud Computing Interface Core," *Open Grid Forum*, 2011.
- [28] S. R. Krishnan, "Cloud Security and Privacy," O'Reilly Media, 2009.
- [29] M. Paolucci, M. Di Penta, D. Di Ruscio, and A. Pierantonio, "Ontology-Driven Discovery and Integration of Services in Cloud Environments," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1652–1663, 2018.
- [30] S. B. Mokhtar et al., "Knowledge-as-a-Service (KaaS): A Cloud-based Approach for Delivering Knowledge Services," *Journal of Internet Services and Applications*, vol. 5, no. 1, pp. 1–18, 2014.

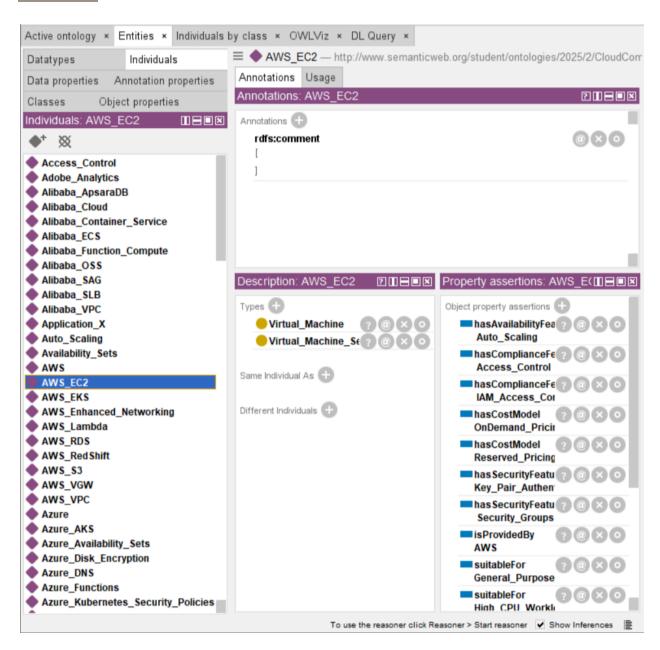
Screenshots

Vocabulary

Entities

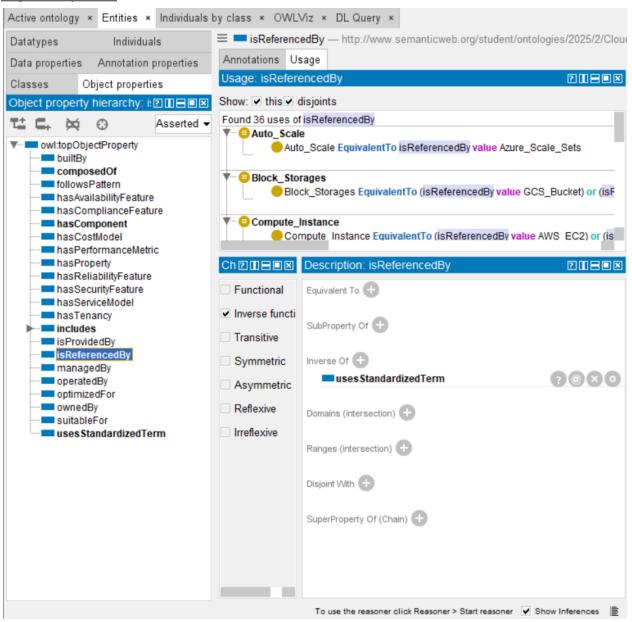


Individuals

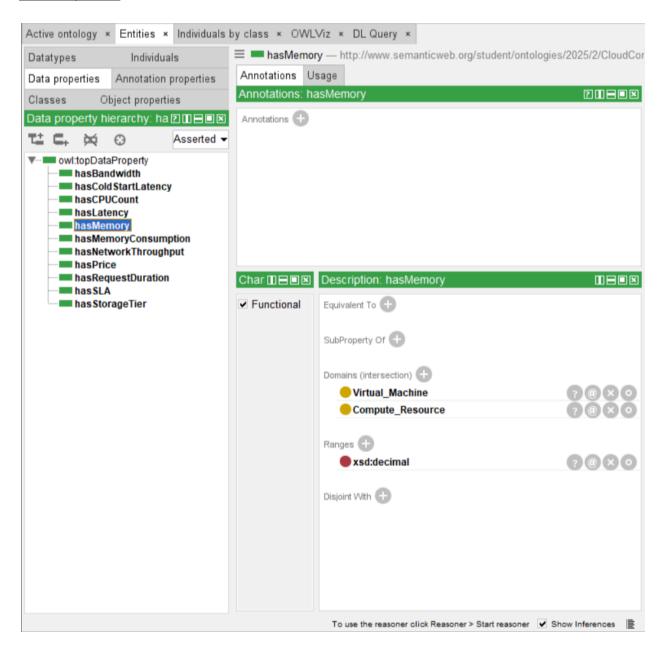


Properties

Object Properties



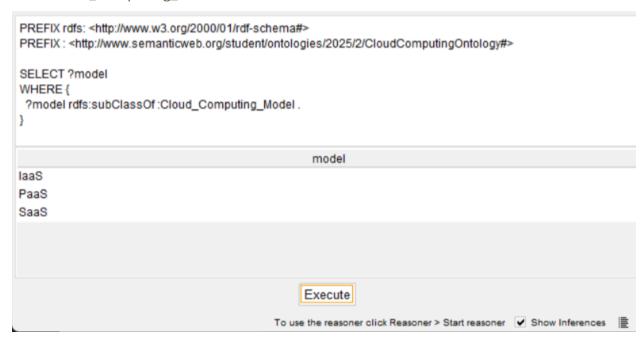
Data Properties



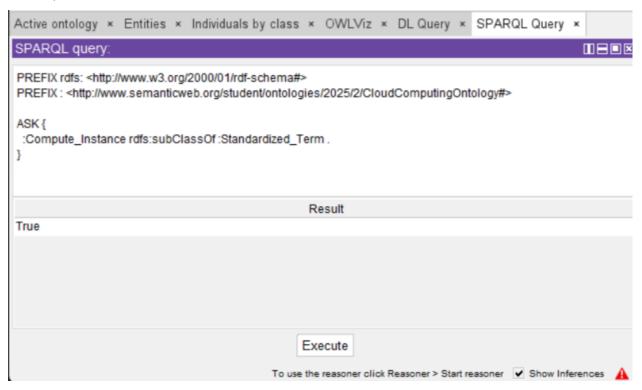
Competency Questions (Answerd by SPARQL Query)

Since this project has many competency questions, we decided to show only 10 of them.

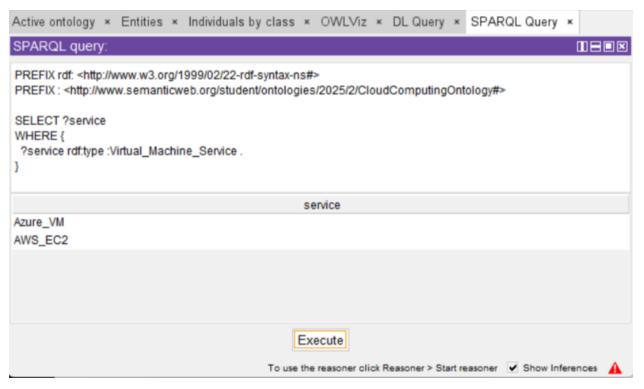
1. What Cloud_Computing_Models are defined?



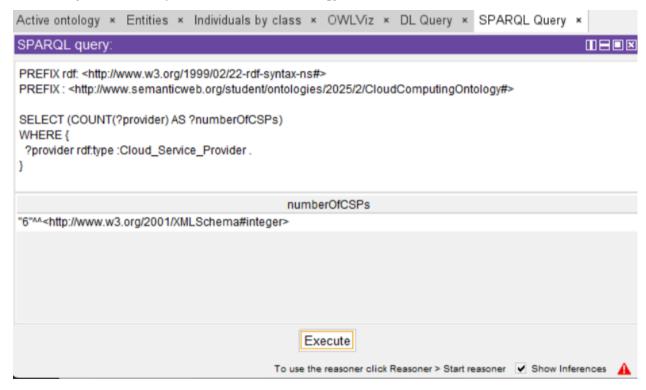
2. Is Compute_Instance a Standardized_Term?



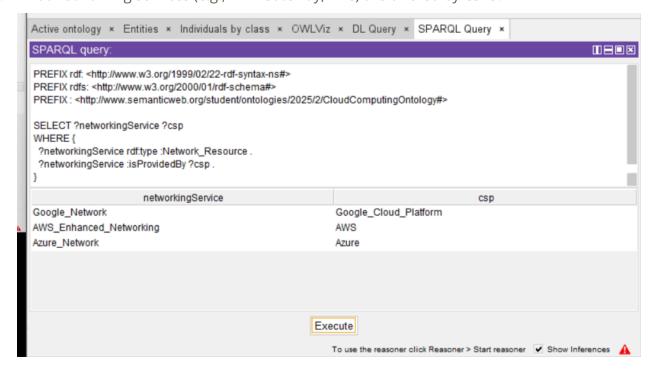
3. Which services are classified as Virtual_Machine_Service?



4. How many CSPs are represented in the ontology?



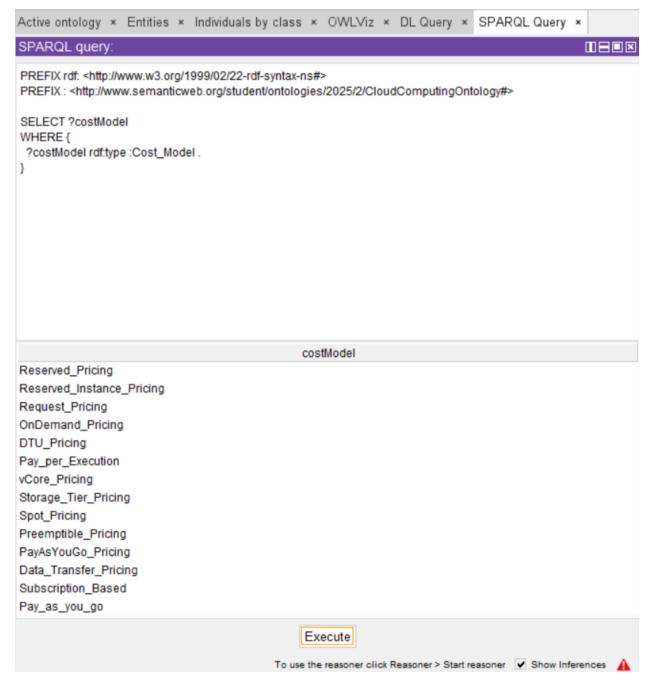
5. What networking services (e.g., VPN Gateway, VPC) are offered by CSPs?



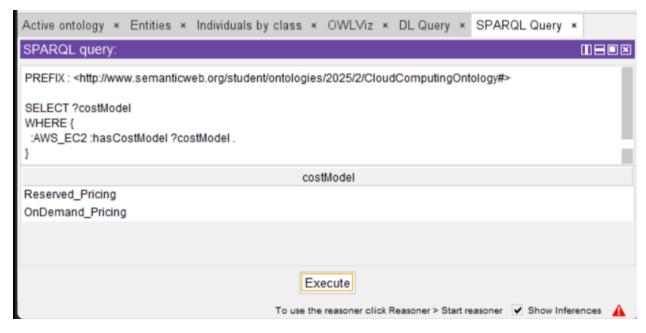
6. Which concepts are modeled as Performance_Metric?



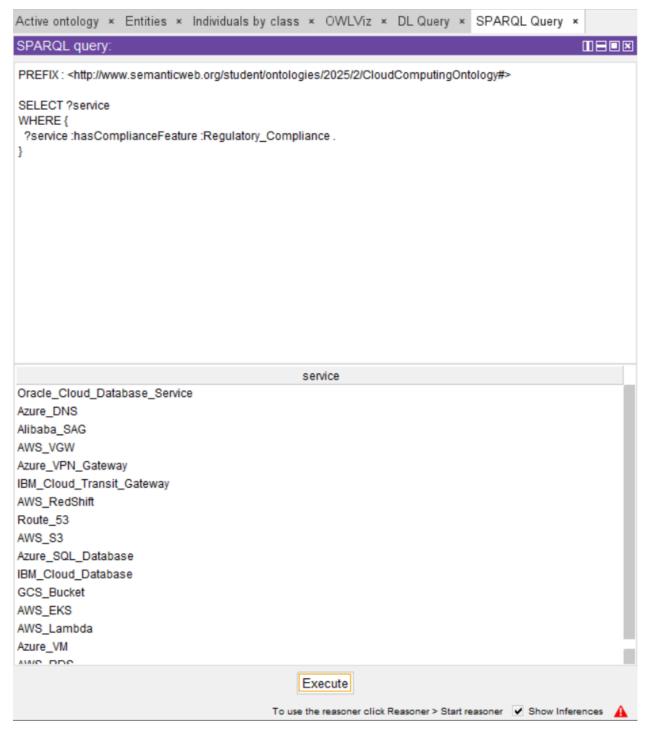
7. Which concepts are classified as instances of Cost_Model?



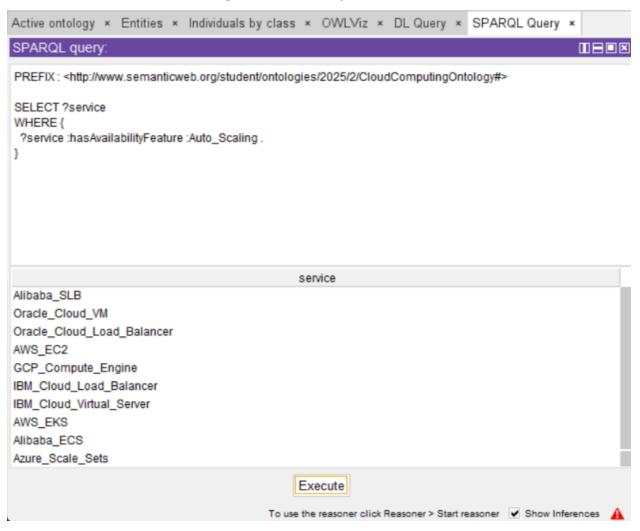
8. What cost models are associated with AWS_EC2?



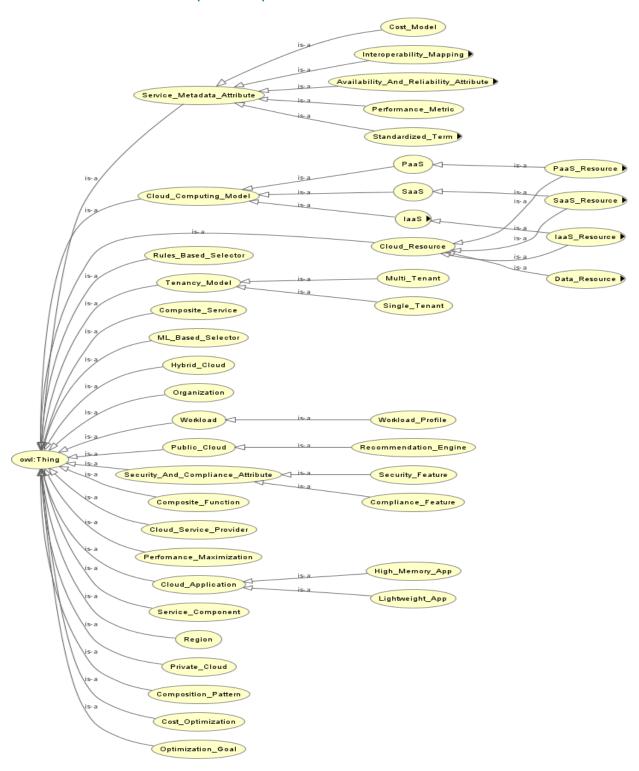
9. Which cloud services comply with Regulatory_Compliance?



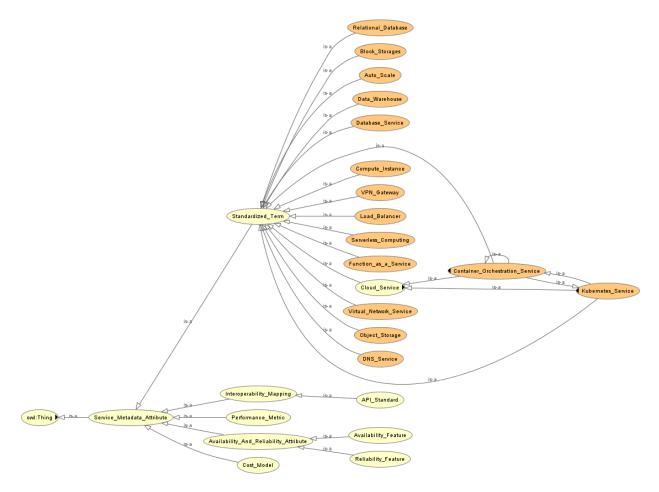
10. Which services offer Auto_Scaling as an availability feature?



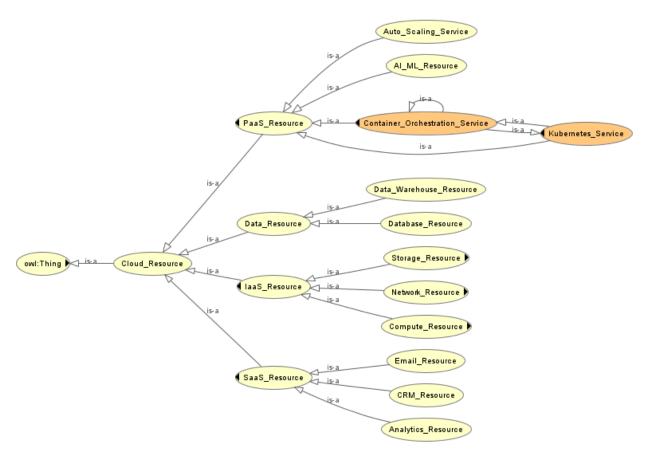
The Structure of Classes (OWL Viz)



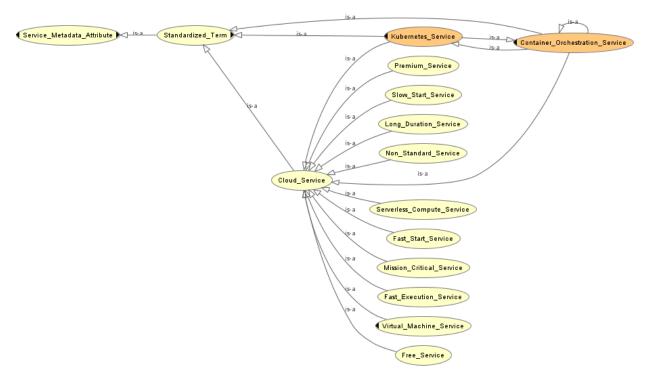
Generalised structure of cloud computing.



Service metadata attribute structure.



Cloud resources structure.



Standardized terms structure