Oracle® Cloud

Using Oracle Cloud Infrastructure Globally Distributed Autonomous Database

F72333-06
April 2024

Oracle Cloud Using Oracle Cloud Infrastructure Globally Distributed Autonomous Database,

F72333-06

Contents

3 Using Globally Distributed Autonomous Database

4 Create and Manage Private Endpoints

5 Lifecycle Operations

6 Monitoring Globally Distributed Autonomous Database

7 Globally Distributed Autonomous Database Policies

1 Overview of Oracle Globally Distributed Autonomous Database

Learn about the Oracle Cloud Infrastructure Globally Distributed Autonomous Database service.

The following topics explain key capabilities of Globally Distributed Autonomous Database and describe the concepts you need to know about the service.

About Oracle Globally Distributed Autonomous Database

Oracle Globally Distributed Autonomous Database is a cloud-based, fully-managed database service that enables the sharding of data across globally distributed converged databases.

It is designed to support large-scale, mission-critical applications. It is a highly available, fault-tolerant, and scalable database service that enables organizations to store and process massive amounts of data with high performance and reliability.

The Globally Distributed Autonomous Database is built on top of Oracle's autonomous technology, which means that it is self-driving, self-securing, and self-healing. This allows automation of many of the routine tasks associated with managing a database, such as patching, tuning, and backup and recovery, which can help reduce the risk of human error and improve system uptime.

For a detailed discussion of Oracle Sharding, see Oracle Sharding Overview for Oracle Database 19c.

About the Components of Globally Distributed Autonomous Database

Globally Distributed Autonomous Database is comprised of the following components:

- **Catalog** - an Oracle Database that supports automated shard deployment, centralized management of Globally Distributed Autonomous Database, and multi-shard queries.

  A Catalog serves following purposes:

  - Serves as an administrative server for the entire Globally Distributed Autonomous Database

  - Stores a gold copy of the database schema
  - Manages multi-shard queries with a multi-shard query coordinator
  - Stores a gold copy of duplicated table data
- **Shard** - Globally Distributed Autonomous Database is a collection of **shards**.

  Each shard in Globally Distributed Autonomous Database is an independent Oracle Database instance that hosts subset of Globally Distributed Autonomous Database data. Shared storage is not required across the shards.

  Shards can all be placed in one region or can be placed in different regions. A region in the context of Globally Distributed Autonomous Database represents a data center or multiple data centers that are in close network proximity.

  Shards are replicated for high availability and disaster recovery with Oracle Data Guard. For high availability, Data Guard standby shards can be placed in the same region where the primary shards are placed. For disaster recovery, the standby shards can be located in another region.
- **Shard director** - A network listener that enable high performance connection routing based on a sharding key. In addition, a shard director is a set of process known collectively as a Global Service Manager (GSM) that acts as a regional listener for clients that connect to Globally Distributed Autonomous Database.

  The shard director maintains a current topology map of Globally Distributed Autonomous Database. Based on the sharding key passed during a connection request, the director routes the connections to the appropriate shard.
- **Global service** - A database service that is used to access data in Globally Distributed Autonomous Database.

  A global service is an extension to the notion of the traditional database service. All of the properties of traditional database services are supported for global services.

For more in depth information about Globally Distributed Autonomous Database components and schema objects see Oracle Sharding Architecture and Concepts in *Using Oracle Sharding*.

Resource Identifiers

Globally Distributed Autonomous Database resources have a unique, Oracle-assigned identifier called an Oracle Cloud ID (OCID).

Globally Distributed Autonomous Database resources are `Sharded-database` and `Sharded-database-work-requests`.

For example, the OCID format for `Sharded-database` is `ocid1.osdshardeddatabase.oc1.iad.<UNIQUE ID>`.

For information about the OCID format and other ways to identify your resources, see Resource Identifiers.

Metering and Billing

Metering and billing for Globally Distributed Autonomous Database is based on the number of ECPU per hour.

Because ECPUs are allocated in the Autonomous Database, see Compute Management and Billing for details.

Service Limits

Globally Distributed Autonomous Database Service Limits can be set for Sharded Database Count and Sharded Database Private Endpoint Count.

Autonomous Database instances, ECPU count, and storage need to have limits set for Autonomous Database service.

See Plan and Monitor Capacity for details.

Integrated Services

Globally Distributed Autonomous Database is integrated with various Oracle Cloud Infrastructure services and features.

IAM

Globally Distributed Autonomous Database integrates with the Identity and Access Management (IAM) service for authentication and authorization for the Console, SDK, CLI, and REST API.

To learn more about IAM, see IAM Overview.

Work Requests

Globally Distributed Autonomous Database uses its own APIs for Work Requests.

The permissions required for using the APIs are documented in Permissions for Globally Distributed Autonomous Database APIs.

Monitoring

Oracle Cloud Infrastructure Monitoring lets you actively and passively monitor your Globally Distributed Autonomous Database resources and alarms.

Globally Distributed Autonomous Database metrics capture CPU utilization, OCPU consumption, memory utilization, deployment health, and inbound and outbound lag. You can view these metrics using the Monitoring service.

2

Getting Started With Globally Distributed Autonomous Database

The following topics give you the information and prerequisites you need to get started with Globally Distributed Autonomous Database.

- Understanding Role Separation
- Configuring a Tenancy for Globally Distributed Autonomous Database
- Interfaces to Globally Distributed Autonomous Database

Understanding Role Separation

You need to ensure that your cloud users have access to use and create only the appropriate kinds of cloud resources to perform their job duties. A best practice of Globally Distributed Autonomous Database is to define roles for the purposes of role separation.

The roles and responsibilities described in the following table should guide your understanding of how to define user groups, dynamic groups, and policies for your Globally Distributed Autonomous Database implementation. The example roles presented here are used throughout the environment setup, resource creation, and management instructions.

Roles	Responsibilities
Tenant administrator	Subscribe to regions Create compartments Create dynamic groups, user groups, and policies
Infrastructure administrator	Create/Update/Delete virtual-network-family Create/Update/Delete Autonomous Exadata Infrastructure Create/Update/Delete Autonomous Exadata VM Clusters Tag Autonomous Exadata VM Clusters Create/Update/Delete Globally Distributed Autonomous Database Private Endpoints

Roles	Responsibilities
Certificate administrator	Create/Update/Delete Vault Create/Update/Delete Keys Create/Update/Delete Certificate Authority Create/Update/Delete Certificate Create/Update/Delete CA Bundle Upload Certificate and Certificate Bundles to Autonomous Exadata VM Clusters Download GSM Certificate Signing Request (CSR) Create a GSM Certificate based on GSM CSR Upload GSM Certificate
User	Create and manage Globally Distributed Autonomous Databases using UI and APIs

Configuring a Tenancy for Globally Distributed Autonomous Database

Before you can use the Globally Distributed Autonomous Database service to create and manage a sharded database, you must perform these preparatory tasks to organize your tenancy, create policies for the various resources, and then procure and configure the network, security, and infrastructure resources.

Task 1. Subscribe to Ashburn Region

As the tenant administrator, subscribe to Ashburn (IAD) region and all of the regions required to run your Globally Distributed Autonomous Database implementation.

1. Subscribe to the Ashburn (IAD) region.
   - To use the service, you must subscribe to the Ashburn region.
   - Your tenancy Home Region does not have to be the Ashburn region, but you must subscribe to the Ashburn region to use Globally Distributed Autonomous Database.
2. Subscribe to any other region where you will be placing a database.
   - Subscribe to any regions where you plan to place databases for your Globally Distributed Autonomous Database implementation; this includes databases for the catalog, shards, and Oracle Data Guard standby databases.

For more information, see Managing Regions.

Task 2. Create Compartments

As the tenant administrator, create compartments in your tenancy for all of the resources required by Globally Distributed Autonomous Database.

Oracle recommends the following structure, and these compartments are referenced throughout the setup tasks:

- A "parent" compartment for the entire deployment. This is **gdad** in the examples.
- "Child" compartments for each of the various kinds of resources:
  - **gdad_certs_vaults_keys** for certificate authorities, certificates, certificate bundles, vaults, and keys
  - **gdad_clusters** for Cloud Autonomous VM Clusters
  - **gdad_databases** for databases, VCNs, subnets, private endpoints, and Globally Distributed Autonomous Database resources.
  - **gdad_exadata** for Exadata Infrastructures
  - **gdad_instances** for compute instances for application servers (edge node/jump host to act as bastion to connect to the database)

The resulting compartment structure will resemble the following:

```
tenant /
     gdad /
          gdad_certs_vaults_keys
          gdad_clusters
          gdad_databases
          gdad_exadata
          gdad_instances
```

For more information, see Working with Compartments.

Task 3. Create User Access Constraints

Formulate an access control plan, and then institute it by creating appropriate IAM (Identity and Access Management) resources. Accordingly, access control within a Globally Distributed Autonomous Database is implemented at various levels, which are defined by the groups and policies here.

The user groups, dynamic groups, and policies described in the following tables should guide the creation of your own user access control plan for your Globally Distributed Autonomous Database implementation.

As the tenant administrator, create the following recommended groups, dynamic groups, and policies to grant permissions to the previously defined roles. The examples and documentation links assume that your tenancy uses identity domains.

Dynamic Groups

Create the following dynamic groups to control access to resources created in the Globally Distributed Autonomous Database compartments.

See Creating a Dynamic Group for instructions.

Dynamic Group Name	Description	Rules
gdad-cas-dg	Certificate authority resources	All resource.type='certificateauthority' resource.compartment.id = 'OCID of compartment tenant root / gdad / gdad_certs_vaults_keys'

Dynamic Group Name	Description	Rules
gdad-clusters-dg	Autonomous VM cluster resources	All resource.compartment.id = 'OCID of compartment tenant root / gdad / gdad_clusters'
gdad-instances-dg	Compute instance resources	All resource.compartment.id = 'OCID of compartment tenant root / gdad / gdad_instances'

User Groups

Create the following groups to give users permissions to use resources in the Globally Distributed Autonomous Database compartments.

See Creating a Group for instructions.

User Group Name	Description
gdad-certificate-admins	Certificate administrators that create and manage keys, vaults, CAs, and certificates
gdad-infrastructure-admins	Infrastructure administrators that create and manage cloud network and infrastructure resources
gdad-users	Users that create and manage Globally Distributed Autonomous Database resources using the APIs and UI

Policies

Create IAM policies to grant the groups access to resources created in the Globally Distributed Autonomous Database compartments.

The following example policies, which are based on the compartment structure and groups created previously, should guide the creation of your own IAM policies for your Globally Distributed Autonomous Database implementation.

The identity domain (for example, Default) should be the identity domain you created the groups in.

See Creating a Policy for instructions.

gdad-certificate-admins-tenant-level

- Description: Tenant-level privileges for group gdad-certificate-admins
- Compartment: tenant
- Statements:

```
Allow group 'Default' / 'gdad-certificate-admins' to INSPECT
tenancies in tenancy
Allow group 'Default' / 'gdad-certificate-admins' to INSPECT work-
requests in tenancy
```

gdad-infrastructure-admins-tenant-level

- Description: Tenant-level privileges for group gdad-infrastructure-admins
- Compartment: tenant
- Statements:

```
Allow group 'Default' / 'gdad-infrastructure-admins' to INSPECT tenancies
in tenancy
Allow group 'Default' / 'gdad-infrastructure-admins' to INSPECT work-
requests in tenancy
Allow group 'Default' / 'gdad-infrastructure-admins' to READ limits in
tenancy
Allow group 'Default' / 'gdad-infrastructure-admins' to READ tag-
namespaces in tenancy
```

gdad-users-tenant-level

- Description: Tenant-level privileges for group gdad-users
- Compartment: tenant
- Statements:

```
Allow group 'Default' / 'gdad-users' to INSPECT tenancies in tenancy
Allow group 'Default' / 'gdad-users' to INSPECT work-requests in tenancy
Allow group 'Default' / 'gdad-users' to READ limits in tenancy
Allow group 'Default' / 'gdad-users' to READ Sharded-database in tenancy
Allow group 'Default' / 'gdad-users' to READ tag-namespaces in tenancy
```

gdad-certificate-admins

- Description: Compartment-level privileges for group gdad-certificate-admins
- Compartment: tenant/gdad
- Statements:

```
Allow group 'Default' / 'gdad-certificate-admins' to MANAGE certificate-
authority-family in compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to MANAGE keys in
compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to MANAGE Sharded-
database in compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to MANAGE vaults in
compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to READ buckets in
compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to READ instances in
compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to READ Sharded-
database-work-requests in compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to USE key-delegate in
compartment gdad
Allow group 'Default' / 'gdad-certificate-admins' to USE subnets in
compartment gdad
```

gdad-infrastructure-admins

- Description: Compartment-level privileges for group gdad-infrastructure-admins
- Compartment: tenant/gdad
- Statements:

```
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE
autonomous-exadata-infrastructures in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE
cloud-autonomous-vmclusters in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE
instance-family in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE
Sharded-database in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE tags
in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to MANAGE
virtual-network-family in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to READ
autonomous-container-databases in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to READ
autonomous-virtual-machines in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins' to READ leaf-
certificate-family in compartment gdad
Allow group 'Default' / 'gdad-infrastructure-admins" to READ
Sharded-database-work-requests in compartment gdad
```

gdad-users

- Description: Compartment-level privileges for group gdad-users
- Compartment: tenant/gdad
- Statements:

```
Allow group 'Default' / 'gdad-users' to MANAGE autonomous-backups
in compartment gdad
Allow group 'Default' / 'gdad-users' to MANAGE autonomous-container-
databases in compartment gdad
Allow group 'Default' / 'gdad-users' to MANAGE autonomous-databases
in compartment gdad
Allow group 'Default' / 'gdad-users' to MANAGE instance-family in
compartment gdad
Allow group 'Default' / 'gdad-users' to MANAGE Sharded-database in
compartment gdad
Allow group 'Default' / 'gdad-users' to MANAGE tags in compartment
gdad
Allow group 'Default' / 'gdad-users' to READ dns-records in
compartment gdad
Allow group 'Default' / 'gdad-users' to READ dns-zone in
compartment gdad
Allow group 'Default' / 'gdad-users' to READ keys in compartment
gdad
Allow group 'Default' / 'gdad-users' to READ Sharded-database-work-
requests in compartment gdad
Allow group 'Default' / 'gdad-users' to READ vaults in compartment
gdad
```

```
Allow group 'Default' / 'gdad-users' to READ vcns in compartment gdad
Allow group 'Default' / 'gdad-users' to USE autonomous-exadata-
infrastructures in compartment gdad
Allow group 'Default' / 'gdad-users' to USE cloud-autonomous-vmclusters
in compartment gdad
Allow group 'Default' / 'gdad-users' to USE network-security-groups in
compartment gdad
Allow group 'Default' / 'gdad-users' to USE private-ips in compartment
gdad
Allow group 'Default' / 'gdad-users' to USE subnets in compartment gdad
Allow group 'Default' / 'gdad-users' to USE vnics in compartment gdad
Allow group 'Default' / 'gdad-users' to USE volumes in compartment gdad
```

gdad-dg-cas

- Description: Compartment-level privileges for dynamic group gdad-cas-dg
- Compartment: tenant/gdad
- Statements:

```
Allow dynamic-group 'Default' / 'gdad-cas-dg' to MANAGE objects in
compartment gdad
Allow dynamic-group 'Default' / 'gdad-cas-dg' to USE keys in compartment
gdad
```

gdad-dg-clusters

- Description: Compartment-level privileges for dynamic group gdad-clusters-dg
- Compartment: tenant/gdad
- Statements:

```
Allow dynamic-group 'Default' / 'gdad-clusters-dg' to MANAGE keys in
compartment gdad_certs_vaults_keys
Allow dynamic-group 'Default' / 'gdad-clusters-dg' to READ vaults in
compartment gdad_certs_vaults_keys
```

gdad-kms

- Description: Compartment-level privileges for Key Management Service
- Compartment: tenant/gdad
- Statements:

```
Allow service keymanagementservice to MANAGE vaults in compartment
gdad_certs_vaults_keys
```

Task 4. Configure Network Resources

As the infrastructure administrator, create the network resources and enable the connectivity needed by the Globally Distributed Autonomous Database implementation.

Example resources are named throughout these instructions to simplify tracking and relationships. For example, the name "gdad_iad" refers to the VCN created in the Ashburn (IAD) region.

Instructions for creating the resources are available at:

- VCNs and Subnets
- Private Endpoints
- Creating a Service Gateway
- Peering VCNs in different regions through a DRG

Common Network Resources

All Globally Distributed Autonomous Database implementations require a VCN, subnet, and a private endpoint in the Ashburn (IAD) region.

As the infrastructure administrator, create the resources as described in the following table.

Resource	Instructions
Virtual Cloud Network (VCN)+ subnet	In Ashburn (IAD), create VCN gdad_iad and subnet osd-gsm-proxy-subnet. This VCN and subnet are required to enable connectivity between the Globally Distributed Autonomous Database service and databases in the Globally Distributed Autonomous Database topology. Use the following values: • Compartment = gdad / gdad_databases • Region = Ashburn (IAD) • Subnet name = osd-gsm-proxy-subnet **Note:** the subnet name must be named osd-gsm-proxy-subnet or Globally Distributed Autonomous Database will not work properly. • Subnet Type = Regional The subnet must be regional, spanning all availability domains

Resource	Instructions
Private Endpoint	Create a private endpoint in the Ashburn (IAD) region to enable connectivity between the Globally Distributed Autonomous Database service and the databases in the Globally Distributed Autonomous Database topology. 1. Open the navigation menu, click**Oracle Database**, then click **Globally Distributed Autonomous Database**. 2. Click **Private Endpoints** in the navigation pane. 3. Click **Create private endpoint**. 4. Enter the following information. • **Name:** For example gdad_pe • **Compartment:** gdad/gdad_databases This should be the compartment you chose when creating the subnet named osd-gsm-proxy-subnet in the previous step. • **Subnet:** osd-gsm-proxy-subnet If you don't see the subnet listed, verify that it was created as a **Regional** subnet. • **Virtual cloud network:** gdad_iad • **Add tags (optional):** you can select tags for this resource by clicking Show Tagging Options.

Additional Network Resources Based on Your Topology

Depending on your Globally Distributed Autonomous Database topology, create additional network resources as described below.

Note that databases for the Globally Distributed Autonomous Database topology include the catalog, shards, and Oracle Data Guard standby databases.

All network resources should be created in the gdad/gdad_databases compartment.

Use Case	Network Resources	Peering and Connectivity
All databases are placed in the Ashburn (IAD) region	Create a subnet and service gateway in Ashburn (IAD) region for your Cloud Autonomous VM Clusters. • In region Ashburn (IAD), create subnet osd-databases-subnet-iad in VCN gdad_iad. • In region Ashburn (IAD), create service gateway gdad_sgw_iad	Required Peering None Required Connectivity Unrestricted connectivity with subnet osd-gsm-proxy-subnet
All databases are placed in a single region, R1, that is not Ashburn (IAD)*	Create a subnet and service gateway in the region for your Cloud Autonomous VM Clusters. • In region R1, create VCN gdad_R1 with subnet osd-database-subnet-R1 • In region R1, create service gateway gdad_sgw_R1	Required Peering gdad_iad ↔ gdad_R1 Required Connectivity Unrestricted between gdad_iad.osd-gsm-proxy-subnet and gdad_R1.osd-database-subnet-R1

Use Case	Network Resources	Peering and Connectivity
Databases are placed in multiple regions R1, R2, ..., RN	Create subnets and service gateways in each region for your Cloud Autonomous VM Clusters. Subnet: • In region R1, create VCN gdad_R1 with subnet osd-database-subnet-R1 • In region R2, create VCN gdad_R2 with subnet osd-database-subnet-R2 ... • In region Rn, create VCN gdad_Rn with subnet osd-database-subnet-Rn Service gateways: • In region R1, create service Gateway gdad_sgw_R1 • In region R2, create Service gateway gdad_sgw_R2 ... • In region Rn, create service Gateway gdad_sgw_Rn	Required Peering gdad_iad ↔ gdad_R1 gdad_iad ↔ gdad_R2 gdad_iad ↔ gdad_Rn gdad_R1 ↔ gdad_R2 gdad_R1 ↔ gdad_Rn gdad_R2 ↔ gdad_Rn Required Connectivity Unrestricted and bi-directional between gdad_iad.osd-gsm-proxy-subnet and gdad_R1.osd-database-subnet-R1 gdad_R2.osd-database-subnet-R2 gdad_Rn.osd-database-subnet-Rn Unrestricted and bi-directional between gdad_R1.osd-database-subnet-R1 and gdad_R2.osd-database-subnet-R2 gdad_Rn.osd-database-subnet-Rn Unrestricted and bi-directional between gdad_R2.osd-database-subnet-R2 and gdad_Rn.osd-database-subnet-Rn

*The Globally Distributed Autonomous Database service control plane exists only in the Ashburn (IAD) region. The private endpoint your created in a previous step in the Ashburn (IAD) region is used to communicate with the Globally Distributed Autonomous Database resources in their respective regions.

Task 5. Configure Security Resources

As the Globally Distributed Autonomous Database certificate administrator, create the vault, key, certificate authority, certificate, and CA bundle resources.

All security resources are created in the gdad/gdad_certs_vaults_keys compartment.

> **Caution:**
>
> After creating a Globally Distributed Autonomous Database that references a key, you cannot move the vault or keys to a new compartment without also restarting the autonomous container databases that reference the moved vault or key.

Depending on your Globally Distributed Autonomous Database topology, create security resources as described in the following tables.

The example resource names used in the following tables should guide the creation of your own security resources for your Globally Distributed Autonomous Database implementation.

Instructions for creating the resources are available at:

- Creating a Vault
- Create a Master Encryption Key
- Replicating a Vault and Keys
- Creating a Certificate Authority
- Creating a Certificate
- Creating a CA Bundle

Automatic Data Distribution, Single Region

In this Globally Distributed Autonomous Database use case, security resources are created in a singe region.

In the examples below, all resources are created in region R1.

Resource	Instructions and Examples
Vault	Create a vault for the Certificate Authority (CA) and the Transparent Data Encryption (TDE) master encryption keys. • In region R1, create vault gdad_vault_R1
Certificate Authority Key	• In region R1, create master encryption key gdad_ca_key_R1, in vault gdad_vault_R1 Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 512
TDE Key	• In region R1, create master encryption key gdad_TDE_key-oraspace in vault gdad_vault_R1 Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256
Certificate Authority	Create a CA for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region R1, using key gdad_ca_key_R1, create CA gdad_ca_R1 You can use a third party CA to create a certificate, but you must import the certificate issued by 3rd Party CA to OCI Certificate Service.
Certificate	Create a Certificate for bundling and uploading to Cloud Autonomous VM Clusters. • In region R1, using CA gdad_ca_R1, create Certificate gdad_cert in vault gdad_vault_R1

Resource	Instructions and Examples
CA Bundle	Create a CA Bundle for uploading to Cloud Autonomous VM Clusters. • In region R1, create a CA Bundle gdad_cert_bundle containing the certificate chain for Certificate gdad_cert

Automatic Data Distribution, Primary and Standby Regions

This topology results when primary and standby databases are placed in different regions. In this Globally Distributed Autonomous Database use case, security resources are created in a the primary database and standby database regions.

In the examples below, resources are created in regions Rp (primary) and Rs (standby).

Resource	Instructions and Examples
Vaults	Create the vaults for the Certificate Authority (CA) master encryption keys. • In region Rp, create vault gdad_vault_Rp • In region Rs, create vault gdad_vault_Rs
Replicated Virtual Vault	Create a replicated virtual vault for the Transparent Data Encryption (TDE) master encryption key. • In region Rp, create virtual vault gdad_vault_Rp_Rs that is replicated to region Rs
Certificate Authority Keys	• In region Rp, create master encryption key gdad_ca_key_Rp in vault gdad_vault_Rp • In region Rs, create master encryption key gdad_ca_key_Rs in vault gdad_vault_Rs Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 512
TDE Key	• In region Rp, create master encryption key gdad_TDE_key-oraspace in replicated virtual vault gdad_vault_Rp_Rs Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256
Certificate Authorities	Create CAs for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region Rp, using key gdad_ca_key_Rp, create CA gdad_ca_Rp • In region Rs, using key gdad_ca_key_Rs, create CA gdad_ca_Rs You can use a third party CA to create a certificate, but you must import the certificate issued by 3rd Party CA to OCI Certificate Service.

Resource	Instructions and Examples
Certificates	Create the Certificates for bundling and upload to Cloud Autonomous VM Clusters. Note: You must use the **same common name** for the certificates in regions Rp and Rs. • In region Rp, using CA gdad_ca_Rp, create Certificate gdad_cert in vault gdad_vault_Rp • In region Rs, using CA gdad_ca_Rs, create Certificate gdad_cert in vault gdad_vault_Rs
CA Bundles	Create the CA Bundles for uploading to Cloud Autonomous VM Clusters. • In region Rp, create CA Bundle gdad_cert_bundle containing the certificate chain for Certificates gdad_cert in regions Rp and Rs • In region Rs, create CA Bundle gdad_cert_bundle containing he certificate chain for Certificates gdad_cert in regions Rp and Rs

User-Managed Data Distribution, Single Region

In this Globally Distributed Autonomous Database use case, security resources are created in a singe region

In the examples below, all resources are created in region R1.

Resource	Instructions and Examples
Vault	Create a vault for the Certificate Authority (CA) and the Transparent Data Encryption (TDE) master encryption keys. • In region R1, create vault gdad_vault_R1
Certificate Authority Key	• In region R1, create key gdad_ca_key_R1 in vault gdad_vault_R1 Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 512
TDE Keys	• In region R1, create key gdad_TDE_key-catalog in vault gdad_vault_R1 for encrypting the catalog • In region R1, create key gdad_TDE_key-spaceN in vault gdad_vault_R1 for encrypting the shards in shard space N Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256

Resource	Instructions and Examples
Certificate Authority	Create a CA for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region R1, using key gdad_ca_key_R1, create CA gdad_ca_R1 You can use a third party CA to create a certificate, but you must import the certificate issued by 3rd Party CA to OCI Certificate Service.
Certificate	Create a Certificate for bundling and uploading to Cloud Autonomous VM Clusters. • In region R1, using CA key gdad_ca_R1, create Certificate gdad_cert in vault gdad_vault_R1
CA Bundle	Create a CA Bundle for uploading to Cloud Autonomous VM Clusters. • In region R1, create a CA Bundle gdad_cert_bundle containing the certificate chain for Certificate gdad_cert

User-Managed Data Distribution, Multiple Regions

In this Globally Distributed Autonomous Database use case, security resources are created in every region where a database will be placed.

This topology can result when either, or both, of the following are true:

- The primary catalog and shard databases are placed in different regions
- The databases within a shard space are placed in different regions

Security resources are created in each region, R1, ..., Rn, where a database will be placed.

Resource	Instructions and Examples
Vaults	Create a vault in each region for the Certificate Authority (CA) master encryption keys. • In region R1, create vault gdad_vault_R1 • In region R2, create vault gdad_vault_R2 ... • In region Rn, create vault gdad_vault_Rn
Replicated Virtual Vaults	Create replicated virtual vaults for the Transparent Data Encryption (TDE) master encryption keys. For each database, catalog or shard, with a primary region, Rp, that is different from its standby region, Rs: • Create a virtual vault, gdad_vault_Rp_Rs, in the database's primary region, Rp, that is replicated to the database's standby region, Rs.

Resource	Instructions and Examples
Certificate Authority Keys	• In region R1, create key gdad_ca_key_R1 in vault gdad_vault_R1 • In region R2, create key gdad_ca_key_R2 in vault gdad_vault_R2 ... • In region Rn, create key gdad_ca_key_Rn in vault gdad_vault_Rn Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 512
TDE Keys	For each database, catalog, or shard, that either has no standby database, or has a standby region that is the same as its primary region: • Create key gdad_TDE_key-catalog for the catalog database in the vault in the region where the catalog's database is placed • Create key gdad_TDE_key-spaceN for a shard space database in the vault in the region where the shard's database is placed For each database, catalog or shard, with a primary region that is different from its stand by region: • Create key gdad_TDE_key-catalog in the replicated virtual vault in the region where the catalog's primary database is placed • Create key gdad_TDE_key-spaceN in the replicated virtual vault in the region where the shard's primary database is placed Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256
Certificate Authorities	Create a Certificate Authority (CA) in each region for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region R1, using key gdad_ca_key_R1, create CA gdad_ca_R1 • In region R2, using key gdad_ca_key_R2, create CA gdad_ca_R2 ... • In region Rn, using key gdad_ca_key_Rn, create CA gdad_ca_Rn You can use a third party CA to create a certificate, but you must import the certificate issued by 3rd Party CA to OCI Certificate Service.

Resource	Instructions and Examples
Certificates	Create Certificates in each region for bundling and upload to Cloud Autonomous VM Clusters. **Note:** You must use the **same common name** for the certificates in all regions. • In region R1, using CA gdad_ca_R1, create Certificate gdad_cert in vault gdad_vault_R1 • In region R2, using CA gdad_ca_R2, create Certificate gdad_cert in vault gdad_vault_R2 ... • In region Rn, using CA gdad_ca_Rn, create Certificate gdad_cert in vault gdad_vault_Rn
CA Bundles	Create the CA Bundles for upload to Cloud Autonomous VM Clusters. • In region R1, create CA Bundle gdad_cert_bundle containing the certificate chain for Certificates gdad_cert in regions R1, R2, ..., Rn • In region R2, create CA Bundle gdad_cert_bundle containing the certificate chain for Certificates gdad_cert in regions R1, R2, ..., Rn ... • In region Rn, create CA Bundle gdad_cert_bundle containing the certificate chain for Certificates gdad_cert in regions R1, R2, ..., Rn

Task 6. Create Exadata Resources

As the infrastructure administrator, configure the Globally Distributed Autonomous Database topology.

Keep the following in mind:

- The Globally Distributed Autonomous Database service supports only two node, quarter rack Exadata.
- An Exadata Infrastructure is region specific. This means that each region in which you plan to place a catalog or shard database will require an Exadata Infrastructure.
- You must create a Cloud Autonomous VM Cluster for each catalog and shard database you plan to deploy in the Globally Distributed Autonomous Database.
- Shards and catalog databases can be co-located on a given Cloud Autonomous VM Cluster. However, using a common Cloud Autonomous VM Cluster for catalog and shard database has the potential to cause a processing bottleneck.

Create Exadata Infrastructure Instances

Create Exadata Infrastructure resources in the gdad/gdad_exadata compartment.

Follow the instructions in Create an Exadata Infrastructure Resource.

Import Oracle-ApplicationName Tag Namespace

Import the Oracle-ApplicationName tag namespace in the root compartment of your tenancy.

1. From the Cloud console navigation menu, select **Governance & Administration**, then **Tag Namespaces** (under the Tenancy Management category).
2. In the Tag Namespaces panel, check if the Oracle-ApplicationName namespace exists in the root compartment of your tenancy.

   Make sure the root compartment of your tenancy is selected under **List Scope**.
3. If you don't see Oracle-ApplicationName in the list, do the following:
   a. Click **Import Standard Tags** (located above the list).
   b. Select the checkbox next to the Oracle-ApplicationName namespace and click **Import**.

Create Cloud Autonomous VM Clusters

Create clusters in gdad/gdad_clusters compartment.

It is required that you define the following tag as you create each cluster:

```
Oracle-ApplicationName.Other_Oracle_Application: Sharding
```

> **Note:**
>
> Before you can add the tag to an Autonomous Exadata VM Cluster, you must import the tag's namespace as described in the previous step.

See Create an Autonomous Exadata VM Cluster for steps to create the clusters.

Task 7. Upload the Cloud Autonomous VM Cluster Certificates

As the certificate administrator, you created the certificate authority, certificates, and CA bundle in the gdad/gdad_certs_vaults_keys compartment. Now you upload the CA Bundle to each Autonomous Exadata VM Cluster.

Important:

- The CA bundle you upload should be **identical** for all Autonomous Exadata VM Clusters.
- The certificate common name should be **identical** for all Autonomous Exadata VM Clusters.

For more information, see Manage Security Certificates for an Autonomous Exadata VM Cluster Resource.

Task 8. (Optional) Create API Key and User Constraints

Create an OCI API key pair if you intend to directly use the Globally Distributed Autonomous Database REST API, OCI Software Development Kits, and Command Line Interface.

Follow the instructions in Required Keys and OCIDs.

If you want to set user controls on the APIs see Permissions for Globally Distributed Autonomous Database APIs.

Interfaces to Globally Distributed Autonomous Database

You can use Oracle Cloud Infrastructure Globally Distributed Autonomous Database service through the Oracle Cloud Interface Console (a browser based interface), REST APIs, or Oracle Cloud Infrastructure Software Development Kits and Command Line Interface.

Using the Console

To access Globally Distributed Autonomous Database using the Console:

1. Use a supported browser to access the Console.

   See Signing In to the Console for details.
2. Enter your cloud tenant, user name, and password, when prompted.
3. Click **Sign in**.
4. In the upper-right corner of the window, select a region that offers the Globally Distributed Autonomous Database service enabled; for example, **US East (Ashburn)**.
5. From the navigation menu, select **Oracle Database**, then **Globally Distributed Autonomous Database**.

   The home page for Globally Distributed Autonomous Database is displayed.

Using Globally Distributed Autonomous Database APIs

You can find the complete Globally Distributed Autonomous Database REST API reference at https://docs.oracle.com/iaas/api/#/en/globally-distributed-autonomous-database/latest/

REST APIs available for the operations relevant to each Globally Distributed Autonomous Database resource are listed in Globally Distributed Autonomous Database REST APIs and Globally Distributed Autonomous Database Private Endpoint REST APIs.

See REST APIs and Software Development Kits and Command Line Interface for more information about using REST APIs and the OCI Software Development Kits and Command Line Interface.

3 Using Globally Distributed Autonomous Database

You create a Globally Distributed Autonomous Database configuration, which is used as a blueprint for the service to procure VMs, deploy the Globally Distributed Autonomous Database software components on systems you designate in the configuration and start required services. You can then monitor and perform life cycle operations on the database.

The following topics explain how to configure, deploy, and perform operations on Globally Distributed Autonomous Database:

- Creating a Globally Distributed Autonomous Database Resource
- Managing Certificates
- Deploying Globally Distributed Autonomous Database
- Downloading Client Credentials
- Viewing Globally Distributed Autonomous Database Details
- Add a Shard
- Terminating (Deleting) a Shard
- Stopping a Globally Distributed Autonomous Database
- Starting a Globally Distributed Autonomous Database
- Terminating (Deleting) a Globally Distributed Autonomous Database
- Moving Globally Distributed Autonomous Database Resources
- Monitoring Work Requests
- Updating the Display Name
- Managing Tags

Globally Distributed Autonomous Database Creation and Deployment Workflow

To get started with Globally Distributed Autonomous Database, you must create the configuration, ensure signed certificates are uploaded, and then deploy the configuration.

Task	Description	More Information
Create Globally Distributed Autonomous Database configuration	Configure the connectivity, security, and topology details of the shards and shard catalog databases.	Creating a Globally Distributed Autonomous Database Resource
Download, Sign, and Upload the certificate	When using TLS you must upload a signed certificate before you can deploy the configuration.	Managing Certificates

Task	Description	More Information
Deploy Globally Distributed Autonomous Database	Deploy the configuration and start the services.	Deploying Globally Distributed Autonomous Database

Creating a Globally Distributed Autonomous Database Resource

A Globally Distributed Autonomous Database resource contains the connectivity and configuration details of the shards and shard catalog databases.

You create the resource in the Globally Distributed Autonomous Database home page.

1. Log in to the Console as a user with permissions to create Globally Distributed Autonomous Database resources, and navigate to the Globally Distributed Autonomous Database home page.
2. Click **Create Globally Distributed Autonomous Database**.

   This will open a three step wizard.
3. In step 1, Configure Globally Distributed Autonomous Database:

   Provide the following information.

Setting	Description and Notes
Compartment	Select a compartment to host the Globally Distributed Autonomous Database resource
Display name	Enter a user-friendly description or other information that helps you easily identify the Autonomous Database. Avoid entering confidential information. You can modify this name after resource creation.
Database name prefix	This prefix is appended to all of the database names in the configuration for ease of use.

   Note the information in the Configure Globally Distributed Autonomous Database pane.

Setting	Value
Deployment type	Dedicated Infrastructure
Database version	19c
Workload type	Transaction Processing
4. In step 2, Configure Shards and Catalog, in **Configure Shards**, provide the following information.

Setting	Description and Notes
Automated	Data is automatically distributed across shards using partitioning by consistent hash. The partitioning algorithm evenly and randomly distributes data across shards.
User managed	Lets you explicitly specify the mapping of data to individual shards. It is used when, because of performance, regulatory, or other reasons, certain data needs to be stored on a particular shard, and the administrator needs to have full control over moving data between shards. **Note:** When you choose User managed data distribution, your **Shards** configuration settings apply to the shardspace rather than the shard itself.
Shard count	Enter the total number of shards to initially deploy in the Globally Distributed Autonomous Database. You can configure up to 10, and then add more later if needed.
Shards	In the upper right corner of the Configure Shards pane, you can toggle between a default list view and a Map view. The **Map** view filters and shows the available Exadata clusters where shards could be deployed. To create shards in the map, click on the available regions, then click **Configure Shards**. If you wish, you can toggle to the form view and refine the configuration.
Primary region	Select the primary region where you would like to host your shard
Primary VM cluster	Select a cluster available in the selected primary region.
Shard/Shardspace name	Shows the display name for each shard or shardspace in the configuration. Once you select a region the name is populated.

Setting	Description and Notes
ECPU	Enter the number of ECPU cores to enable for each shard. Specify the number of ECPUs as an integer. Available cores are subject to your tenancy’s service limits. You must enter a minimum of 2 ECPUs per shard. ECPUs are based on the number of cores, elastically allocated, from the shared pool of Exadata database servers and storage servers. Aggregated ECPU consumption on a given cluster is 1.5 times the ECPU count. Note that a number of ECPUs are consumed in overhead and are not available to the shards. See Oracle Cloud Infrastructure Documentation for more information.
ECPU auto scaling	Enable automatic scaling based on workload per shard/shardspace. This value is passed on to the Autonomous Database so that it can manage ECPU auto scaling.
Storage	GB of storage to allocate to your database
Enable Data Guard	Instantiates Oracle Data Guard standby databases for each shard.
Data Guard region	Select the region where you would like to host the shard's Data Guard standby
Data Guard VM Cluster	Select a cluster available in the selected Data Guard region.
Configure Catalog	You can choose to use the same configuration that is applied to the shards, or uncheck the box and make selections that apply only to the catalog database. The same fields are as described above for Shards.
Create administrator credentials	Create the user that will be able to access the shard catalog and all of the shards in the configuration.

Setting	Description and Notes
Encryption key	The encryption key settings you configure depend on the data distribution type you chose above. **Automated** - All shards have the same encryption vault and encryption key, and is mandatory. **User managed** - Each shard can have the same or different encryption key details, and is optional. For both cases: • Based on the primary region that you selected for the first shard, you select the vaults and encryption key available in that region and selected compartment. • If Data Guard is enabled for a shard, and if the standby region is not the same as the primary region for that shard, you can select virtual private vaults that are replicated in the standby region.
Select character sets	Select the Character sets and National character sets that will be used in all of the shard and shard catalog databases. The AL32UTF8 character set is recommended by default for character sets and the AL16UTF16 character set is recommended by default for National character sets.
Select ports	Enter the **Listener port**, **ONS port (local)**, and **ONS port (remote)**.
TLS	**TLS port** - TLS port number **Cluster certificate common name** - Identifies a similar group of clusters. Enter a name that is 3 to 64 characters and can contains letters, numbers, hyphens(-), underscores(_), and dots(.) The Cluster certificate common name must match the certificate common name that was used when the clusters were created.
Advanced options: Chunks	Under Advanced Options you can optionally configure the number of chunks per shard. This setting is only applicable when Automated data distribution is selected.
Advanced options: Tags	Under Advanced Options you can add tags to the Globally Distributed Autonomous Database resource. These can also be added after creation.

5. Click **Next** to review the configuration details.
6. If everything on the summary page is correct, click **Validate** to run validation against the configuration.
7. Once any validation errors are addressed and validation is successful, click **Create**.

   After you click **Create**, the Globally Distributed Autonomous Database display name appears in the list while the creation operation runs.

The creation operation can take a while, because several tasks are performed as part of the create operation, including host procurement, installing software, and generating certificates for the shard directors (GSMs).

You can monitor the operation status in the State column and track progress in the Work request tab. When the shard status is Available, Globally Distributed Autonomous Database creation is complete and successful.

> **Caution:**
>
> After a user creates a Globally Distributed Autonomous Database, do not move vaults and keys or the Globally Distributed Autonomous Database will not work.

8. When the Create process is complete you can continue to Managing Certificates, so you can download, sign, and upload the certificates for the GSMs.

Managing Certificates

You must upload signed certificates for the shard directors (GSMs) to the Globally Distributed Autonomous Database before you can deploy the configuration.

When you create a Globally Distributed Autonomous Database, a certificate signing request (CSR) is generated.

Using the Console

You can manage certificates on the Globally Distributed Autonomous Database details page.

1. Sign in to your Oracle Cloud Account at cloud.oracle.com, navigate to the Globally Distributed Autonomous Database home page, and select the Globally Distributed Autonomous Database for which you want to manage certificates.
2. Click **Manage certificate** on the Globally Distributed Autonomous Database details page.
3. Click **Download CSR** in the Manage Certificates panel.

   If for some reason there is no certificate available for download, go back to the Manage Certificates panel and click **Generate Certificate**.
4. Create a certificate with the CSR using the same Certificate Authority (CA) used to create certificates for the Autonomous VM clusters.

   The Certificate Authority (CA) that is the issuer of the GSM certificate should be the same as that used for the Exadata Autonomous VM Cluster certificate (see Task 5. Configure Security Resources).

   If there are multiple issuers for Exadata Autonomous VM Cluster certificates, make sure only **one** of the issuers of Exadata Autonomous VM Clusters signs the GSM certificate.
5. Upload the Certificate.

   To upload the signed certificate, click **Manage Certificates** on the Globally Distributed Autonomous Database details page as described above, then choose an option on the lower half of the panel to upload the certificate.

You can upload the signed certificate as a Certificate file (.crt), or paste the content into the field.

See Creating a Certificate for more information.

Now you are ready to deploy and start the Globally Distributed Autonomous Database. See Deploying Globally Distributed Autonomous Database

Deploying Globally Distributed Autonomous Database

You deploy a Globally Distributed Autonomous Database after uploading signed certificates and any time you make changes to the configuration, such as add a shard.

1. Sign in to your Oracle Cloud Account at cloud.oracle.com, and navigate to the Globally Distributed Autonomous Database details page for which you want to complete the deployment.
2. Click **Configure Sharding**.
3. Select **Rebalance** to automatically redistribute data among the shards.
   This is typically done after adding or removing shards from the configuration in case of Automated Sharding type.
4. Click **Configure Sharding** to start the deployment.

Downloading Client Credentials

You need the client credentials and connection information to connect to your database. The client credentials include the wallet.

Oracle client credentials (wallet files) are downloaded from Globally Distributed Autonomous Database by a service administrator. If you are not a Globally Distributed Autonomous Database administrator, your administrator should provide you with the client credentials.

1. Navigate to the Globally Distributed Autonomous Database details page.
2. Click **Database connection**.
3. On the Database Connection panel click **Download Wallet**.
4. In the **Download Wallet** dialog, enter a wallet password in the **Password** field and confirm the password in the **Confirm Password** field.

   The password must be at least 8 characters long and must include at least 1 letter and either 1 numeric character or 1 special character.

   > **Note:**
   >
   > This password protects the downloaded Client Credentials wallet. This wallet is not the same as the Transparent Data Encryption (TDE) wallet for the database; therefore, use a different password to protect the Client Credentials wallet.

5. Click **Download** to save the client security credentials zip file.

   By default the file name is: `Wallet_`*databasename*`.zip`. You can save this file as any file name you want.

   You must protect this file to prevent unauthorized database access.

The zip file includes the following:

- `tnsnames.ora` and `sqlnet.ora`: Network configuration files storing connect descriptors and SQL*Net client-side configuration.
- `cwallet.sso` and `ewallet.p12`: Auto-open SSO wallet and PKCS12 file. PKCS12 file is protected by the wallet password provided in the UI.
- `truststore.jks`: Java truststore file that is protected by the wallet password provided while downloading the wallet.
- `ojdbc.properties`: Contains the wallet related connection property required for JDBC connection. This should be in the same path as `tnsnames.ora`.
- `hostinfo.json`: Host information file with a list of IP addresses that are part of the cluster used by the Globally Distributed Autonomous Database.

Viewing Globally Distributed Autonomous Database Details

Finding the Details Page

You view Globally Distributed Autonomous Database configuration, backup, and maintenance information by going to its **Details** page.

1. Sign in to your Oracle Cloud Account at cloud.oracle.com.
2. Click the ☰ menu icon in the top left corner to display the navigation menu.
3. Click **Oracle Database** in the navigation menu.
4. Choose **Globally Distributed Autonomous Database** under Oracle Database. The Globally Distributed Autonomous Database **home page** opens.
5. If needed, switch to the compartment hosting the database. See Understanding Compartments for information about using and managing compartments.
6. In the list of databases, select the name of the database you want. The **Details** page for the selected database is displayed.

   The **Globally Distributed Autonomous Database information** tab shows some configuration information.

There are a few places to look for information depending on what you are looking for.

Resource Information

The **Database information** panel, which is accessed when you click **Show all**, gives the following details:

- **Name:** Display name
- **Compartment**
- **OCID:** Here you can view the full OCID or copy it
- **Deployment type:** Dedicated Infrastructure
- **Workload type:** Transaction Processing
- **Data distribution:** Automated or User managed
- **Database version:** Oracle Database release number (for example, 19.18.0.1.0)

- **Created:** Creation date (for example, Fri, May 12, 2023, 20:02:40 UTC)
- **Lifecycle state:** Available, Failed
- **Listener port:** Default 1522
- **ONS ports (local):** Default 6123
- **ONS ports (remote):** Default 6234
- **TLS port**
- **Cluster certificate common name**
- **Character set:** For example, AL16UTF16
- **National character set:** For example, AL16UTF16
- **Time zone** For example, UTC
- **Last updated**

Configuration Summary

The **Summary** panel, accessed by choosing **Summary** from the **More actions** menu, displays some of the same information as the Database information panel, but in addition you will find:.

- **Database name prefix**
- **Username** Administrator user name
- **Shards and Catalog details:** Shard name, ECPU, ECPU auto scaling, Storage, Primary region, Primary VM cluster, Data Guard enabled, Data Guard region, and Data Guard VM cluster
- **Tags** such as Oracle-Tags.CreatedBy and Oracle-Tags.CreatedOn

Shard and Catalog Tab

The Shards and Catalog tab display a searchable, filterable summary of each database in the Global Scale Autonomous Database configuration, which includes:

- State of the database (Available or Failed)
- Allocated ECPUs and storage
- Shard group or shard space membership
- Region of deployment
- Availability domain
- VM cluster

In addition you can click on the Disaster Recovery arrow at the right end of each row to display any Data Guard configuration information.

Work Requests

The work requests tab displays the status of ongoing operations on the databases.

Add a Shard

For more information about the concepts and considerations of adding shards to a Globally Distributed Autonomous Database see Shard Management in *Using Oracle Sharding*.

You can add shards to the Globally Distributed Autonomous Database from its **Details** page.

1. Go to the **Details** page of the Globally Distributed Autonomous Database to which you want to add a shard.
2. On the **Details** page, on the **Shards and Catalog** tab, select **Add Shard**.
3. On the **Add Shards** pane configure the new shard.

   In **Shard Count** indicate the number of shards you want to add, then configure them in the table below.

   - **Shard/Shardspace name** - Shows the display name for each shard or shardspace in the configuration. Once you select a region the name is populated.
   - **Primary region** - Select the primary region where you would like to host your shard
   - **Primary VM cluster** - Select a cluster available in the selected primary region.
   - **ECPU count** - The number of ECPU cores to enable. Specify the number of ECPUs for your shard as an integer. Available cores are subject to your tenancy's service limits.
   - **ECPU auto scaling** - Enable automatic scaling based on workload per shard/ shardspace
   - **Storage** - GB of storage to allocate to your database
   - **Enable Data Guard** - Instantiates Oracle Data Guard standby instances for each shard
   - **Data Guard region** - Select the region where you would like to host the shard's Data Guard standby
   - **Data Guard VM Cluster** - Select a cluster available in the selected Data Guard region.
4. In **Create administrator credentials**, set the password for the shard database ADMIN user.
5. Select the **Encryption key** details for the new shards.

   The encryption key settings you configure depend on the data distribution method configured for the Globally Distributed Autonomous Database when it was created.

   Automated - All shards have the same encryption vault and encryption key, and is mandatory.

   User managed - Each shard can have the same or different encryption key details, and is optional.

   For both cases:

   - Based on the primary region that you selected for the first shard, you select the vaults and encryption key available in that region and selected compartment.
   - If Data Guard is enabled for a shard, and if the standby region is not the same as the primary region for that shard, you can select virtual private vaults that are replicated in the standby region.
6. Click **Validate** to run checks to make sure the new shards are valid.

7. Once any validation errors are addressed and validation is successful, click **Add Shards** to deploy the new shards.

Modifying a Shard

You can modify a shard's ECPU count, auto-scaling setting, and storage allocation.

You can modify shards in a Globally Distributed Autonomous Database from its **Details** page.

1. Go to the **Details** page of the Globally Distributed Autonomous Database in which you want to modify a shard.
2. In the **Details** page, on the **Shards and Catalog** tab, select **Modify** from the Actions (three dots) menu for the shard you want to make changes to.

   On the **Modify Shard** pane you can configure the ECPU and storage settings.

   - **ECPU** - The number of ECPU cores to enable. Specify the number of ECPUs for the shard as an integer. Available cores are subject to your tenancy's service limits.
   - **ECPU auto scaling** - Enable automatic scaling based on workload per shard/shardspace.
   - **Storage** - GB of storage to allocate to your shard.
   - **Data Guard** - Indicates if an Oracle Data Guard standby instance is deployed for this shard.
3. Click **Apply** to save the changes to the shard.

Terminating (Deleting) a Shard

Terminating a shard in a Globally Distributed Autonomous Database configuration permanently deletes it and removes all automatic backups. You cannot recover a terminated shard.

For more information about the concepts and considerations of removing shards see Shard Management in *Using Oracle Sharding*.

1. Go to the **Details** page of the Globally Distributed Autonomous Database from which you want to remove a shard.
2. On the **Details** page, on the **Shards and Catalog** tab select a checkbox for the shard, and then select **Terminate Shard**.
3. For Globally Distributed Autonomous Database configured for Automated data distribution, you can select **Rebalance the data** to evenly redistribute the data from this shard among the remaining shards.
4. On the **Terminate Shards** dialog enter the Globally Distributed Autonomous Database name to confirm that you want to remove the shard.
5. Click **Remove**.

Stopping a Globally Distributed Autonomous Database

> **Note:**
>
> When you stop Globally Distributed Autonomous Database, the following details apply:
>
> - Tools are no longer able to connect to the database.
> - In-flight database transactions and queries are stopped.
> - ECPU billing is halted.

1. Go to the **Details** page of the Globally Distributed Autonomous Database you want to stop.
2. On the **Details** page, select **Actions** and then select **Stop**.
3. Click **Stop** to confirm.

Starting a Globally Distributed Autonomous Database

> **Note:**
>
> When you start Globally Distributed Autonomous Database, CPU billing is initiated, billed by the second with a minimum usage period of one minute.

1. Go to the **Details** page of the Globally Distributed Autonomous Database you want to start.
2. On the **Details** page, select **Actions** and then select **Start**.

   Start is only shown for a stopped Globally Distributed Autonomous Database.
3. Click **Start** to confirm.

Terminating (Deleting) a Globally Distributed Autonomous Database

Terminating Globally Distributed Autonomous Database permanently deletes it and removes all automatic backups. You cannot recover a terminated Globally Distributed Autonomous Database.

1. Go to the **Details** page of the Globally Distributed Autonomous Database you want to terminate.
2. On the **Details** page, select **Actions** and then select **Terminate**.

3. On the Terminate Database page enter the Globally Distributed Autonomous Database name to confirm that you want to terminate the database.
4. Click **Terminate**.

Moving Globally Distributed Autonomous Database Resources

You can move a Globally Distributed Autonomous Database from one compartment to another.

> **Caution:**
>
> If you need to move a Globally Distributed Autonomous Database resource, please contact Oracle customer support first. There may be unintended consequences to moving any resource within the Globally Distributed Autonomous Database configuration. See Moving Resources to a Different Compartment for more information.

> **Note:**
>
> As soon as you move the Globally Distributed Autonomous Database to a different compartment, the policies that govern the new compartment apply immediately and affect access to the database. Therefore, your access to the database may change, depending on the policies governing your Oracle Cloud user account's access to resources.
>
> After the Globally Distributed Autonomous Database move to a new compartment is successful, any work request logs associated with the Globally Distributed Autonomous Database from the original compartment are no longer available.

To move Globally Distributed Autonomous Database you must have the right to manage Globally Distributed Autonomous Database in its current compartment and in the compartment you are moving it to.

1. Select **Move resource** on the Globally Distributed Autonomous Database details page.
2. In the **Move Globally Distributed Autonomous Database to a different compartment** dialog, select the compartment to move the Globally Distributed Autonomous Database to from the dropdown.
3. Click **Move Globally Distributed Autonomous Database**.

Updating the Display Name

You can change the display name of a Globally Distributed Autonomous Database from its **Details** page.

1. Go to the **Details** page of the Globally Distributed Autonomous Database you want to update.
2. On the **Details** page, select **Actions** and then select **Update display name**.

3. Enter the new display name in the **New display name** field.
4. Enter the current name in the field below to confirm the name change.
5. Click **Update display name**.

Managing Tags

Tags help you locate resources within your tenancy.

You can add and view tags from the Globally Distributed Autonomous Database home page and details page.

On the Globally Distributed Autonomous Database home page, from the Globally Distributed Autonomous Database home Actions (three dots) menu, select you can select **Add Tags**.

On the Globally Distributed Autonomous Database details page, you can select **Add Tags** from the **More actions** menu, or click the **Tags** tab to add, view, and edit tags.

See Managing Tags and Tag Namespaces to learn more about tagging.

Globally Distributed Autonomous Database REST APIs

The following REST APIs are used to interact with the Globally Distributed Autonomous Database (sharded-database) resource.

These APIs are documented in the Globally Distributed Autonomous Database REST API reference at https://docs.oracle.com/iaas/api/#/en/globally-distributed-autonomous-database/latest/

REST API	Description
ChangeShardedDatabaseCompartment	Moves the Globally Distributed Autonomous Database identified by `shardedDatabaseId` and its dependent resources to the specified compartment.
ConfigureSharding	Lets you complete the Globally Distributed Autonomous Database deployment identified by `shardedDatabaseId`.
CreateShardedDatabase	Creates a new Globally Distributed Autonomous Database(sharded-database) resource. t
DeleteShardedDatabase	Deletes the Globally Distributed Autonomous Database identified by `shardedDatabaseId`.
DownloadGsmCertificateSigningRequest	Generates the common certificate signing request for the Globally Distributed Autonomous Database GSM instances.
FetchConnectionString	Gets the Globally Distributed Autonomous Database connection string for application connections.
FetchShardableCloudAutonomousVmClusters	Gets a list of `cloudAutonomousVMClusters` for the given tenancy, that can be sharded.

REST API	Description
GenerateGsmCertificateSigningRequest	Generates the certificate signing request for Globally Distributed Autonomous Database GSM instances. Once the certificate signing request is generated, you can download it using `downloadGsmCertificateSigningRequest` .
GenerateWallet	Generates the wallet for application connections with Globally Distributed Autonomous Database.
GetShardedDatabase	Gets the details of Globally Distributed Autonomous Database resources specified with `shardedDatabaseId`.
PatchShardedDatabase	Lets you add, remove, or update shards in the Globally Distributed Autonomous Database topology. You can add, remove, or update multiple shards in a single patch operation; however, combinations of inserts, updates, and removes in a single operation are not allowed.
PrevalidateShardedDatabase	Prevalidates a Globally Distributed Autonomous Database configuration before creating it.
StartShardedDatabase	Starts the Globally Distributed Autonomous Database identified by `shardedDatabaseId`.
StopShardedDatabase	Stops the Globally Distributed Autonomous Database identified by `shardedDatabaseId`.
UpdateShardedDatabase	Lets you change the display name and edit tags associated with a Globally Distributed Autonomous Database resource.
UploadSignedCertificateAndGenerateWallet	Uploads the CA signed certificate to the Globally Distributed Autonomous Database GSM instances, and generate wallets for the GSM instances.
ValidateNetwork	Validates the network connectivity between components of the Globally Distributed Autonomous Database.
ListShardedDatabases	Gets a list of Globally Distributed Autonomous Databases.

See Globally Distributed Autonomous Database Private Endpoint REST APIs for descriptions of the private endpoint REST APIs.

4 Create and Manage Private Endpoints

Private endpoints are created during Globally Distributed Autonomous Database environment set up. Operations on existing private endpoint resources are described in the topics below.

Create a Private Endpoint

Create a private endpoint to configure the sharded database topology for Oracle Cloud databases running in a private VCN (including Oracle Database on compute) or in Oracle on-premises databases that have a FastConnect or IPSec VPN connection to OCI.

A private endpoint is created in the Private Endpoints page which is linked from the Globally Distributed Autonomous Database home page.

1. Log in to the Console as a user with permissions to create Globally Distributed Autonomous Database resources.
2. From the Console hamburger menu, select **Oracle Database**, then **Globally Distributed Autonomous Database**.

   The home page for Globally Distributed Autonomous Database is displayed.
3. Click **Private Endpoints** in the left navigation panel to open the Private Endpoints page.
4. Click **Create private endpoint**.
5. In the Create private endpoint panel, enter the following information.
   - **Name:** Enter a name.
   - **Description:** Optionally, enter a description.
   - **Choose compartment:** Choose the compartment containing the subnet "osd-gsm-proxy-subnet" that you created in Additional Network Resources Based on Your Topology.
   - **Subnet in *compartment***: Choose the "osd-gsm-proxy-subnet" subnet.
   - **Virtual cloud network in *compartment***: Select a VCN
6. Optionally, you can select tags for this resource by clicking **Show Tagging Options**.

See Create and Manage Private Endpoints for lifecycle operations.

About Private Endpoints

Private endpoints are used to connect to Oracle cloud databases running in a private VCN (including Oracle Database on compute) and to connect to Oracle on-premises databases that have a FastConnect or IPSec VPN connection to OCI.

Private Endpoints Home Page

You can find a link to the Private Endpoints home page on the Globally Distributed Autonomous Database home page.

On the home page you can create and terminate (delete) private endpoints using the buttons at the top of the page.

Using the Actions (three dots) menu on a private endpoint in the list, you can perform a variety of operations, such as add and view tags, copy OCID, move the resource, update the display name, edit the configuration details, or terminate it.

Private Endpoints Details Page

You can find information about private endpoints, run operations, and make changes to them on the Private Endpoint Details page for each private endpoint resource.

You can find a link to the Private Endpoints page in the Globally Distributed Autonomous Database home page. From there, click on the name of a private endpoint to open its details page.

At the top of the details page there are buttons to run operations on the private endpoint, such as update the display name, move resource, add tags and terminate. On this page there are also tabs which show configuration information and tags.

The details page also lets you view Work Requests and any Sharded Databases (Globally Distributed Autonomous Database) that use this private endpoint.

Edit Private Endpoints

You can change the name and description of a private endpoint.

1. Log in to the Console as a user with permissions to create Globally Distributed Autonomous Database resources.
2. Click **Private Endpoints** to open the Private Endpoints page.
3. In the list, select **Edit private endpoint** from the Actions (three dots) menu for the private endpoint you want to make changes to.

Move Private Endpoints

You can move a private endpoint resource from one compartment to another.

1. In the list of private endpoints on the Private Endpoints home page, select **Move Resource** from the Actions (three dots) menu for the private endpoint you want to move.

   You can also select **Move Resource** on the Private Endpoint Details page.
2. In the **Move resource** dialog, select the compartment to move the private endpoint to from the dropdown.
3. Click **Move Resource**.

After you move the private endpoint to the new compartment, inherent policies apply immediately and may affect access to the private endpoint through the Console. For more information, see Managing Compartments.

Globally Distributed Autonomous Database Private Endpoint REST APIs

The following REST APIs are used to interact with the Globally Distributed Autonomous Database Private Endpoint (sharded-database-private-endpoint) resource.

These APIs are documented in the Globally Distributed Autonomous Database REST API reference at https://docs.oracle.com/iaas/api/#/en/globally-distributed-autonomous-database/20230301/PrivateEndpoint/

REST API	Description
ChangePrivateEndpointCompartment	Moves the private endpoint to the specified compartment.
CreatePrivateEndpoint	Creates a private endpoint.
DeletePrivateEndpoint	Deletes a private endpoint.
GetPrivateEndpoint	Gets a private endpoint.
UpdatePrivateEndpoint	Updates private endpoint configuration details.
ListPrivateEndpoints	Lists private endpoints.

See Globally Distributed Autonomous Database REST APIs for descriptions of the `sharded-database` REST APIs.

5
Lifecycle Operations

Backing Up and Restoring a Globally Distributed Autonomous Database

Backup and restore is done at the shard (and catalog) database level and is managed by the underlying Autonomous Database. There is no backup management at the Globally Distributed Autonomous Database level.

Recovery is also done using Autonomous Database flows.

Manual backup is also done from Autonomous Database. Click on a shard in your Globally Distributed Autonomous Database configuration and it takes you to the Autonomous Database page where you can manage backups.

See Backup and Restore Autonomous Database on Dedicated Exadata Infrastructure for information.

6 Monitoring Globally Distributed Autonomous Database

Monitor Databases with Performance Hub

You can use Performance hub to view real-time and historical performance data for Globally Distributed Autonomous Database. Performance Hub shows Shard Status, Data Distribution, and Performance information.

Performance Hub is displayed only for users with Admin privileges.

Accessing the Performance Hub

1. Go to the **Details** page of the Globally Distributed Autonomous Database you want to monitor with Performance hub.
2. On the **Details** page, select **Performance hub**.

On the Performance hub page you will find:

- A banner that displays the number of catalogs and shards, primary and standby, and a summary with number of regions, shardspaces, storage and ECPUs, and global services.
- Tabs for **Shards** and **Catalogs** with graphs depicting performance metrics, such as CPU utilization, Storage utilization, Sessions, Execute count, Running statements, and Queued statements.

> **Note:**
>
> If you are using default database metrics then you will not see data from any undiscovered shards in the chart.
> If you are using enhanced metrics, the data for all shards is displayed because the shards are discovered by the shard catalog.

Monitoring Work Requests

Using the Console:

Globally Distributed Autonomous Database work request status is displayed in the Details page.

From the Globally Distributed Autonomous Database home page, click any database name and go to the Details page.

The lower set of tabs include the **Work requests** tab, which displays the status of ongoing operations.

Using the REST APIs

You can use the GetWorkRequest and ListWorkRequests APIs to get work request status.

See Work Request Reference for details.

Globally Distributed Autonomous Database Events

Globally Distributed Autonomous Database emits events in Oracle Cloud Infrastructure (OCI), which are structured messages that indicate changes in the sharded database resource.

You can define rules in the OCI Event Service to get notified of events happening in an OCI native service and use the Notification Service (ONS) to send emails or other notifications from these events.

Table 6-1 Event Types for Sharded Database

Friendly Name	Event Type
Global Autonomous Database - Change Compartment Begin	`com.oraclecloud.globaldb.changeshardeddatabasecompartment.begin`
Global Autonomous Database - Change Compartment End	`com.oraclecloud.globaldb.changeshardeddatabasecompartment.end`
Global Autonomous Database - Configure GSMs Begin	`com.oraclecloud.globaldb.configureshardeddatabasegsms.begin`
Global Autonomous Database - Configure GSMs End	`com.oraclecloud.globaldb.configureshardeddatabasegsms.end`
Global Autonomous Database - Configure Sharding Begin	`com.oraclecloud.globaldb.configuresharding.begin`
Global Autonomous Database - Configure Sharding End	`com.oraclecloud.globaldb.configuresharding.end`
Global Autonomous Database - Create Begin	`com.oraclecloud.globaldb.createshardeddatabase.begin`
Global Autonomous Database - Create End	`com.oraclecloud.globaldb.createshardeddatabase.end`
Global Autonomous Database - Delete Begin	`com.oraclecloud.globaldb.deleteshardeddatabase.begin`
Global Autonomous Database - Delete End	`com.oraclecloud.globaldb.deleteshardeddatabase.end`
Global Autonomous Database - Download GSM Certificate Signing Request	`com.oraclecloud.globaldb.downloadgsmcertificatesigningrequest`
Global Autonomous Database - Fetch Connection String	`com.oraclecloud.globaldb.fetchconnectionstring`
Global Autonomous Database - Fetch Shardable Cloud Autonomous VM Clusters	`com.oraclecloud.globaldb.fetchshardablecloudautonomousvmclusters`
Global Autonomous Database - Generate GSM Certificate Signing Request Begin	`com.oraclecloud.globaldb.generategsmcertificatesigningrequest.begin`

Table 6-1 (Cont.) Event Types for Sharded Database

Friendly Name	Event Type
Global Autonomous Database - Generate GSM Certificate Signing Request End	`com.oraclecloud.globaldb.generategsmcertificatesigningrequest.end`
Global Autonomous Database - Generate Wallet	`com.oraclecloud.globaldb.generatewallet`
Global Autonomous Database - Patch Begin	`com.oraclecloud.globaldb.patchshardeddatabase.begin`
Global Autonomous Database - Patch End	`com.oraclecloud.globaldb.patchshardeddatabase.end`
Global Autonomous Database - Prevalidate	`com.oraclecloud.globaldb.prevalidateshardeddatabase`
Global Autonomous Database - Start Begin	`com.oraclecloud.globaldb.startshardeddatabase.begin`
Global Autonomous Database - Start End	`com.oraclecloud.globaldb.startshardeddatabase.end`
Global Autonomous Database - Stop Begin	`com.oraclecloud.globaldb.stopshardeddatabase.begin`
Global Autonomous Database - Stop End	`com.oraclecloud.globaldb.stopshardeddatabase.end`
Global Autonomous Database - Update	`com.oraclecloud.globaldb.updateshardeddatabase`
Global Autonomous Database - Upload Signed Certificate And Generate Wallet Begin	`com.oraclecloud.globaldb.uploadsignedcertificateandgeneratewallet.begin`
Global Autonomous Database - Upload Signed Certificate And Generate Wallet End	`com.oraclecloud.globaldb.uploadsignedcertificateandgeneratewallet.end`
Global Autonomous Database - Validate Network Begin	`com.oraclecloud.globaldb.validatenetwork.begin`
Global Autonomous Database - Validate Network End	`com.oraclecloud.globaldb.validatenetwork.end`

Table 6-2 Event Types for Sharded Database Private Endpoint

Friendly Name	Event Type
Global Database Private Endpoint - Change Compartment Begin	`com.oraclecloud.globaldb.changeprivateendpointcompartment.begin`
Global Database Private Endpoint - Change Compartment End	`com.oraclecloud.globaldb.changeprivateendpointcompartment.end`
Global Database Private Endpoint - Create Begin	`com.oraclecloud.globaldb.createprivateendpoint.begin`
Global Database Private Endpoint - Create End	`com.oraclecloud.globaldb.createprivateendpoint.end`
Global Database Private Endpoint - Delete Begin	`com.oraclecloud.globaldb.deleteprivateendpoint.begin`

Table 6-2 (Cont.) Event Types for Sharded Database Private Endpoint

Friendly Name	Event Type
Global Database Private Endpoint - Delete End	`com.oraclecloud.globaldb.deleteprivateendpoint.end`
Global Database Private Endpoint - Update	`com.oraclecloud.globaldb.updateprivateendpoint`

Globally Distributed Autonomous Database Metrics

Because Globally Distributed Autonomous Database is a collection of database instances and services, you monitor metrics for those resources which make up the Globally Distributed Autonomous Database topology.

See also: Monitor Databases with Autonomous Database Metrics

7
Globally Distributed Autonomous Database Policies

To control access to Globally Distributed Autonomous Database and the type of access each user group has, you must create policies.

The following topics explain how to create policies for Globally Distributed Autonomous Database.

Giving Permissions to Users

Use IAM policies to grant certain capabilities to a Globally Distributed Autonomous Database user group.

You can configure group and group permissions so that members can manage Globally Distributed Autonomous Database resources.

Create user groups to manage Globally Distributed Autonomous Database resources with role-based levels of access, and then add users that require access to these resources to the groups.

Remember that only resources within the same compartment can access each other, unless the proper permissions are granted. Ensure that you have the proper permissions to view and select the appropriate VCN and subnet when creating sharded databases.

See Globally Distributed Autonomous Database Policies for recommended user groups and policies as well as information about resource-types, details for Verbs + Resource-Type combinations, and permissions required for Globally Distributed Autonomous Database API operations.

Globally Distributed Autonomous Database Resource-Types

Globally Distributed Autonomous Database offers individual resource-types for writing policies.

Resource-Type	Description
`Sharded-database`	Configuration of the Globally Distributed Autonomous Database, including the data distribution model and information for connecting to the shards and catalog databases.
`Sharded-database-work-requests`	Monitor for long-running operations, such as shard creation, update, or deletion.

About Creating Globally Distributed Autonomous Database Policies

Required policies

For the list of required policies for Globally Distributed Autonomous Database, see Task 3. Create User Access Constraints.

Policy best practices

As a best practice you should create policies for the following personas.

Persona	Role	Role Privilege	Permission Name
Super Administrator / Fleet Administrator	Admin	1. Read,Write, and Admin access to Sharded database 2. Ability to add and remove users	SDB_Manage
Database Administrator / DBA	User	1. Use Access	SDB_USE
Developer	Read-only	Read Access	SDB_READ

You will need to add policies to enable access to these services as well:

- Oracle Autonomous Databases for your shard and catalog databases
- Oracle Vault to store secrets

Allowing Globally Distributed Autonomous Database resource management

The following statements give a group of super administrator users, `gdadAdminGroup`, permission to *manage* Globally Distributed Autonomous Database `Sharded-database` resources.

```
allow group gdadAdminGroup to manage Sharded-database in compartment
gdadCompartment
```

The `manage` permission lets users in this group create and delete Globally Distributed Autonomous Database resources.

Limiting users to only "use" capability

If you want a group of users that only have the ability to use Globally Distributed Autonomous Database resources, but not create and delete them, then create a separate group for those users, such as `GD-ADUserGroup`, and replace `manage` with `use` as shown here.

```
allow group gdadUserGroup to use Sharded-database in compartment
gdadCompartment
```

Users with the `use` permission can edit resources, but cannot create or delete the resources.

Allowing network resource management

Globally Distributed Autonomous Database requires you to provide VCN and subnet information when creating sharded database resources. In order to provide this information, you need to have the ability to view cloud network information.

The following example gives the group `gdadGroup` permission to `inspect` network resources in the compartment and select them when creating resources.

```
allow group gdadGroup to inspect virtual-network-family in compartment
gdadCompartment
```

To let users of `gdadGroup` manage the network resources for Globally Distributed Autonomous Database resources, grant the group `manage virtual-network-family` permission as shown here.

```
allow group gdadGroup to manage virtual-network-family in compartment
gdadCompartment
```

If the `manage virtual-network-family` policy is restricted because of security reasons then the following policies are required:

```
allow group gdadGroup to inspect vcns in compartment gdadCompartment
allow group gdadGroup to use subnets in compartment gdadCompartment
allow group gdadGroup to manage vnic in compartment gdadCompartment
```

This way, you can view the list of existing VCNs, view and work with subnets, and have all of the permissions on VNIC.

Creating a Tagging Policy

The following statement gives group `aghdGroup` permission to manage `tag-namespaces` and `tags` for workspaces:

```
allow group gdadGroup to manage tag-namespaces in compartment gdadCompartment
```

Related Topics

Learn more about:

- Policies

- Permissions
- Resource Tags

Creating a Policy

You create policies using the Console.

1. In the Console navigation menu, under **Governance and Administration**, go to **Identity**, and then click **Policies**.
2. Click **Create Policy**.
3. Enter a name and description for the policy.
4. In the **Statement** field, enter a policy rule in the following format:

   ```
   allow <subject> to <verb> <resource-type> in <location> where
   <condition>
   ```

   Conditions are optional.
5. (Optional) To add another statement, click **+ Another Statement**.
6. Click **Create**.

For more information about policies, see how policies work, policy syntax, and policy reference.

Details for Verbs + Resource-Type Combinations

There are various Oracle Cloud Infrastructure verbs and resource-types that you can use when you create a policy. The topics in this section show the permissions and API operations covered by each verb for Globally Distributed Autonomous Database.

The level of access is cumulative as you go from `inspect` to `read` to `use` to `manage`.

sharded-database

Permission	APIs Fully Covered
INSPECT	
SDB_INSPECT	ListShardedDatabase
READ	
INSPECT +	INSPECT+
SDB_READ	GetShardedDatabase GenerateShardedDatabaseWallet GetConnectionString
UPDATE	
READ +	READ +

Permission	APIs Fully Covered
SBD_UPDATE	UpdateShardedDatabase RotateShardedDatabaseGsms ValidateNetwork StartShardedDatabase StopShardedDatabase
SDB_MOVE	ChangeShardedDatabaseCompartment
CREATE	
UPDATE+	UPDATE+
SDB_CREATE	CreateShardedDatabase
DELETE	
CREATE+	CREATE+
SDB_DELETE	DeleteShardedDatabase

sharded-database-work-request

Permission	APIs Fully Covered
INSPECT	
SDB_WORK_REQUEST_LIST	ListWorkRequests
READ	
INSPECT +	INSPECT+
SDB_WORK_REQUEST_READ	GetWorkRequest ListWorkRequestErrors ListWorkRequestLogs

Supported Variables

When you add conditions to your policies, you can use either Globally Distributed Autonomous Database general or service specific variables.

Globally Distributed Autonomous Database supports all general variables. For more information, see general variables for all requests.

Resource-Permissions Model for Globally Distributed Autonomous Database

Each resource defines its own permissions model. This permissions model forms the basis of how a policy is defined to allow for authorized access to resources.

These permissions are intended to be mapped to Operations (list, get, update delete, and so on) to allow for fine grained access control.

- **Read** (read-only)– allows the user to view `sharded-database` details

- **Update** – grants View permission, plus allows the user to edit an existing `sharded-database` resource, including move, add shard, remove shard
- **Create** – grants Update permission, plus allows the user to create new `sharded-database` resources
- **Delete** – grants Create permission, plus allows the user to delete a `sharded-database`

The following table details the permissions model for Globally Distributed Autonomous Database resources.

Resource	Permissions
`sharded-database`	- SDB_INSPECT - SDB_READ - SDB_CREATE - SDB_UPDATE (update, add, remove) - SDB_DELETE - SDB_MOVE
`sharded-database-work-requests`	- SDB_WORK_REQUEST_READ - SDB_WORK_REQUEST_LIST

Permissions for Globally Distributed Autonomous Database APIs

Here's a list of the API operations mapped to permissions for Globally Distributed Autonomous Database in logical order, grouped by resource-type.

The resource-types are `Sharded-database` and `Sharded-database-work-request`.

API Operation	Permission
`ListShardedDatabase`	SDB_INSPECT
`GetShardedDatabase`	SDB_READ
`GenerateShardedDatabaseWallet`	SDB_READ
`GetConnectionString`	SDB_READ
`UpdateShardedDatabase`	SDB_UPDATE
`RotateShardedDatabaseGsms`	SDB_UPDATE
`ValidateNetwork`	SDB_UPDATE
`StartShardedDatabase`	SDB_UPDATE
`StopShardedDatabase`	SDB_UPDATE
`ChangeShardedDatabaseCompartment`	SDB_MOVE
`CreateShardedDatabase`	SDB_CREATE
`DeleteShardedDatabase`	SDB_DELETE
`ListWorkRequests`	SDB_WORK_REQUEST_LIST
`GetWorkRequest`	SDB_WORK_REQUEST_READ
`ListWorkRequestErrors`	SDB_WORK_REQUEST_READ
`ListWorkRequestLogs`	SDB_WORK_REQUEST_READ