Oracle® Cloud

Using Oracle Globally Distributed Autonomous Database

F72333-14
May 2025

Oracle Cloud Using Oracle Globally Distributed Autonomous Database,

F72333-14

Contents

1 Overview of Oracle Globally Distributed Autonomous Database

2 Getting Started With Globally Distributed Autonomous Database

3 Create and Manage a Globally Distributed Autonomous Database

4 Create and Manage Private Endpoints

5 Monitoring a Globally Distributed Database

6 Globally Distributed 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

Globally Distributed Autonomous Database brings the power of distributed (sharded) databases to Oracle Autonomous Database on Dedicated Exadata Infrastructure.

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 distributed database features supported in Oracle Database, see Oracle Sharding Overview for Oracle Database 19c and Oracle Globally Distributed Database Overview for Oracle Database 23ai.

Globally Distributed Database Concepts

To gain a greater understanding of Globally Distributed Database concepts, familiarize yourself with the following terminology.

- **Catalog** - an Oracle Database that supports automated shard deployment, centralized management of the distributed database, and multi-shard queries.

A Catalog serves following purposes:

  - Serves as an administrative server for the entire distributed 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 -** A distributed database is a collection of **shards**.

  Each shard in a distributed database is an independent Oracle Database instance that hosts subset of the distributed 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.

  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.

- **Shardspace -** A shardspace is a shard that stores data corresponding to a range or list of key values in a user-managed data distribution configuration. A shardspace consists of a shard and its replica.

- **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 processes known collectively as a Global Service Manager (GSM) that acts as a regional listener for clients that connect to a Globally Distributed Database.

  The shard director maintains a current topology map of the distributed 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 the distributed 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 distributed database components and schema objects see Architecture and Concepts in *Oracle Globally Distributed Database*.

Data Replication Solutions

Oracle's Globally Distributed Database services offer data replication solutions to ensure high availability, disaster recovery, and additional scalability for reads.

Globally Distributed Database offers shard-level replication with Oracle Data Guard on Oracle Database releases 19c and 23ai. Raft replication is available with Oracle Database beginning in release 23ai.

Oracle Globally Distributed Database automatically deploys the specified replication topology to the procured systems, and enables data replication.

Shard-level Replication with Oracle Data Guard

A shard is a database. Oracle Data Guard replication of shards to physical standby databases can be used to provide individual shard-level high availability. Replication is automatically configured and deployed when the distributed database is created.

Oracle Data Guard is tightly integrated with Oracle's Globally Distributed Database services to provide high availability and disaster recovery with strict data consistency and zero data loss. Oracle Data Guard replication maintains synchronized copies (standby databases) of shards (the primary databases) for high availability and data protection. Standbys can be deployed locally or remotely.

Chunk Set-level Replication with Raft Replication

Instead of replication at the whole shard level using additional databases for standbys, the Raft replication feature in a Globally Distributed Database creates sets of chunks of data from each shard and distributes them automatically among the shards to handle chunk assignment, chunk movement, workload distribution, and balancing upon scaling (addition or removal of shards), including planned or unplanned shard availability changes.

Raft replication is built into the Globally Distributed Database to provide a consensus-based, high-performance, low-overhead availability solution, with distributed replicas and fast failover with zero data loss, while automatically maintaining the replication factor if shards fail. With Raft replication management overhead does not increase with the number of shards. If you are used to NoSQL databases and do not expect to know anything about how replication works, native replication just works.

Unlike Data Guard replication, Raft replication does not need to be reconfigured when shards are added or removed, and replicas do not need to be actively managed.

For more details about how Raft replication works see Using Raft Replication in Oracle Globally Distributed Database.

Resource Identifiers

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

Globally Distributed Autonomous Database resources are listed here.

Resource	Identifier
Distributed Autonomous Database	osddistributedautonomousdb
Distributed Database Private Endpoint	osddistributeddbprivateendpoint
OSD Work Request	osdworkrequest

For example, the OCID format for a Distributed Autonomous Database resource is `ocid1.osddistributedautonomousdb.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.

> **Note:**
>
> Once you tag a cluster for use in a Globally Distributed Database, it will continue to bill for the Globally Distributed Database SKU until the cluster is deleted.

Service Limits

Globally Distributed Database Service Limits can be set for Distributed Database Count and Distributed 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

Oracle's Globally Distributed Database services are integrated with various Oracle Cloud Infrastructure services and features.

- IAM
- Work Requests
- Monitoring

IAM

Oracle Globally Distributed Database services are integrated 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.

To monitor work requests see Monitoring 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 Database resources and alarms.

Globally Distributed 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.

See Monitoring a Globally Distributed Database for more details about monitoring the health and performance of a distributed database.

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.

- Configuring the Tenancy
  Before you can use Oracle's Globally Distributed Database services to create and manage a distributed 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.
- 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.

Configuring the Tenancy

Before you can use Oracle's Globally Distributed Database services to create and manage a distributed 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
- Task 2. Create Compartments
- Task 3. Create User Access Constraints
- Task 4. Configure Network Resources
- Task 5. Configure Security Resources
- Task 6. Create Exadata Resources
- Task 7. Upload the Cloud Autonomous VM Cluster Certificates
- (Optional) Create API Key and User Constraints

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 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 Oracle's Globally Distributed Database services.

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 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 the 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 **gdd** in the examples.
- "Child" compartments for each of the various kinds of resources:
    - **gdd_certs_vaults_keys** for certificate authorities, certificates, certificate bundles, vaults, and keys
    - **gdd_clusters** for Cloud Autonomous VM Clusters
    - **gdd_databases** for databases, VCNs, subnets, private endpoints, and Globally Distributed Database resources.
    - **gdd_exadata** for Exadata Infrastructures
    - **gdd_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 /
     gdd /
          gdd_certs_vaults_keys
          gdd_clusters
          gdd_databases
          gdd_exadata
          gdd_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 distributed 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 distributed 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.

- Understanding Role Separation
- Dynamic Groups

- User Groups
- Policies

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 for Globally Distributed 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
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 Databases using UI and APIs

Dynamic Groups

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

See Creating a Dynamic Group for instructions.

Dynamic Group Name	Description	Rules
gdd-cas-dg	Certificate authority resources	All resource.type='certificateauthority' resource.compartment.id = 'OCID of compartment tenant root / gdd / gdd_certs_vaults_keys'
gdd-clusters-dg	Autonomous VM cluster resources	All resource.compartment.id = 'OCID of compartment tenant root / gdd / gdd_clusters'
gdd-instances-dg	Compute instance resources	All resource.compartment.id = 'OCID of compartment tenant root / gdd / gdd_instances'

User Groups

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

See Creating a Group for instructions.

User Group Name	Description
gdd-certificate-admins	Certificate administrators that create and manage keys and vaults.
gdd-infrastructure-admins	Infrastructure administrators that create and manage cloud network and infrastructure resources
gdd-users	Users that create and manage Globally Distributed 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.

gdd-certificate-admins-tenant-level

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

- Statements:

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

gdd-infrastructure-admins-tenant-level

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

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

gdd-users-tenant-level

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

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

gdd-certificate-admins

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

  ```
  Allow group 'Default' / 'gdd-certificate-admins' to MANAGE certificate-
  authority-family in compartment gdd
  Allow group 'Default' / 'gdd-certificate-admins' to MANAGE keys in
  compartment gdd
  Allow group 'Default' / 'gdd-certificate-admins' to MANAGE distributed-
  autonomous-database in compartment gdd
  Allow group 'Default' / 'gdd-certificate-admins' to MANAGE vaults in
  compartment gdd
  Allow group 'Default' / 'gdd-certificate-admins' to READ buckets in
  compartment gdd
  Allow group 'Default' / 'gdd-certificate-admins' to READ instances in
  compartment gdd
  ```

```
Allow group 'Default' / 'gdd-certificate-admins' to READ distributed-
database-work-requests in compartment gdd
Allow group 'Default' / 'gdd-certificate-admins' to USE key-delegate in
compartment gdd
Allow group 'Default' / 'gdd-certificate-admins' to USE subnets in
compartment gdd
```

gdd-infrastructure-admins

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

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

gdd-users

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

```
Allow group 'Default' / 'gdd-users' to MANAGE autonomous-backups in
compartment gdd
Allow group 'Default' / 'gdd-users' to MANAGE autonomous-container-
databases in compartment gdd
Allow group 'Default' / 'gdd-users' to MANAGE autonomous-databases in
compartment gdd
Allow group 'Default' / 'gdd-users' to MANAGE instance-family in
compartment gdd
Allow group 'Default' / 'gdd-users' to MANAGE distributed-autonomous-
database in compartment gdd
Allow group 'Default' / 'gdd-users' to MANAGE tags in compartment gdd
Allow group 'Default' / 'gdd-users' to READ dns-records in compartment gdd
Allow group 'Default' / 'gdd-users' to READ dns-zone in compartment gdd
Allow group 'Default' / 'gdd-users' to READ keys in compartment gdd
Allow group 'Default' / 'gdd-users' to READ distributed-database-work-
requests in compartment gdd
Allow group 'Default' / 'gdd-users' to READ vaults in compartment gdd
Allow group 'Default' / 'gdd-users' to READ vcns in compartment gdd
Allow group 'Default' / 'gdd-users' to USE autonomous-exadata-
infrastructures in compartment gdd
Allow group 'Default' / 'gdd-users' to USE cloud-autonomous-vmclusters in
compartment gdd
Allow group 'Default' / 'gdd-users' to USE network-security-groups in
compartment gdd
Allow group 'Default' / 'gdd-users' to USE private-ips in compartment gdd
Allow group 'Default' / 'gdd-users' to USE subnets in compartment gdd
Allow group 'Default' / 'gdd-users' to USE vnics in compartment gdd
Allow group 'Default' / 'gdd-users' to USE volumes in compartment gdd
```

gdd-dg-cas

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

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

gdd-dg-clusters

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

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

gdd-kms

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

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

Task 4. Configure Network Resources

As the infrastructure administrator, create the network resources and enable the connectivity needed by the distributed database.

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

- Common Network Resources
- Additional Network Resources Based on Your Topology

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 gdd_iad and subnet gdd_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 = gdd / gdd_databases • Region = Ashburn (IAD) • Subnet name = gdd_subnet • Subnet Type = Regional The subnet must be regional, spanning all availability domains
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 gdd_pe • **Compartment:** gdd/gdd_databases This should be the compartment containing the Ashburn region subnet you created above. • **Subnet:** gdd_subnet If you don't see the subnet listed, verify that it was created as a **Regional** subnet. • **Virtual cloud network:** gdd_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 Database topology, create additional network resources as described below.

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

All network resources should be created in the gdd/gdd_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 gdd_iad. • In region Ashburn (IAD), create service gateway gdd_sgw_iad	Required Peering None Required Connectivity Unrestricted connectivity with subnet gdd_subnet (created for private endpoint)
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 gdd_R1 with subnet osd-database-subnet-R1 • In region R1, create service gateway gdd_sgw_R1	Required Peering gdd_iad ⇔ gdd_R1 Required Connectivity Unrestricted between gdd_iad.gdd_subnet (created for private endpoint) and gdd_R1.osd-database-subnet-R1
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 gdd_R1 with subnet osd-database-subnet-R1 • In region R2, create VCN gdd_R2 with subnet osd-database-subnet-R2 ... • In region Rn, create VCN gdd_Rn with subnet osd-database-subnet-Rn Service gateways: • In region R1, create service Gateway gdd_sgw_R1 • In region R2, create Service gateway gdd_sgw_R2 ... • In region Rn, create service Gateway gdd_sgw_Rn	Required Peering gdd_iad ⇔ gdd_R1 gdd_iad ⇔ gdd_R2 gdd_iad ⇔ gdd_Rn gdd_R1 ⇔ gdd_R2 gdd_R1 ⇔ gdd_Rn gdd_R2 ⇔ gdd_Rn Required Connectivity Unrestricted and bi-directional between gdd_iad.gdd_subnet (created for private endpoint) and gdd_R1.osd-database-subnet-R1 gdd_R2.osd-database-subnet-R2 gdd_Rn.osd-database-subnet-Rn Unrestricted and bi-directional between gdd_R1.osd-database-subnet-R1 and gdd_R2.osd-database-subnet-R2 gdd_Rn.osd-database-subnet-Rn Unrestricted and bi-directional between gdd_R2.osd-database-subnet-R2 and gdd_Rn.osd-database-subnet-Rn

*The Globally Distributed 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 Database resources in their respective regions.

Task 5. Configure Security Resources

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

All security resources are created in the gdd/gdd_certs_vaults_keys compartment.

> **Caution:**
>
> After creating a Globally Distributed 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 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 Database implementation.

- Automatic Data Distribution, Single Region
- Automatic Data Distribution, Primary and Standby Regions
- User-Managed Data Distribution, Single Region
- User-Managed Data Distribution, Multiple Regions

Automatic Data Distribution, Single Region

In this 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 gdd_vault_R1 Instructions: Creating a Vault
Certificate Authority Key	• In region R1, create master encryption key gdd_ca_key_R1, in vault gdd_vault_R1 Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 2048 Instructions: Create a Master Encryption Key

Resource	Instructions and Examples
TDE Key	• In region R1, create master encryption key gdd_TDE_key-oraspace in vault gdd_vault_R1 Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256 Instructions: Create a Master Encryption Key
Certificate Authority	Create a CA for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region R1, using key gdd_ca_key_R1, create CA gdd_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. Instructions: Creating a Certificate Authority
Certificate	Create a Certificate for upload to Cloud Autonomous VM Clusters. • In region R1, using CA gdd_ca_R1, create Certificate gdd_cert Instructions: Creating a Certificate
CA Bundle	Create a CA Bundle for upload to Cloud Autonomous VM Clusters. • In region R1, create a CA Bundle gdd_cert_bundle containing the certificate chain for Certificate gdd_cert Instructions: Creating a CA Bundle

Automatic Data Distribution, Primary and Standby Regions

This topology results when primary and standby databases are placed in different regions. In this 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 gdd_vault_Rp • In region Rs, create vault gdd_vault_Rs Instructions: Creating a Vault
Replicated Virtual Vault	Create a replicated virtual vault for the Transparent Data Encryption (TDE) master encryption key. • In region Rp, create virtual vault gdd_vault_Rp_Rs that is replicated to region Rs Instructions: Replicating a Vault and Keys

Resource	Instructions and Examples
Certificate Authority Keys	- In region Rp, create master encryption key gdd_ca_key_Rp in vault gdd_vault_Rp - In region Rs, create master encryption key gdd_ca_key_Rs in vault gdd_vault_Rs Required attribute values: - Protection Mode = HSM - Key Shape: Algorithm = RSA - Length = 2048 Instructions: Create a Master Encryption Key
TDE Key	- In region Rp, create master encryption key gdd_TDE_key-oraspace in replicated virtual vault gdd_vault_Rp_Rs Required attribute values: - Protection Mode = Software - Key Shape: Algorithm = AES - Length = 256 Instructions: Create a Master Encryption Key
Certificate Authorities	Create CAs for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. - In region Rp, using key gdd_ca_key_Rp, create CA gdd_ca_Rp - In region Rs, using key gdd_ca_key_Rs, create CA gdd_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. Instructions: Creating a Certificate Authority
Certificates	Create the Certificates for 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 gdd_ca_Rp, create Certificate gdd_cert - In region Rs, using CA gdd_ca_Rs, create Certificate gdd_cert Instructions: Creating a Certificate
CA Bundles	Create the CA Bundles for upload to Cloud Autonomous VM Clusters. - In region Rp, create CA Bundle gdd_cert_bundle containing the certificate chain for Certificates gdd_cert in regions Rp and Rs - In region Rs, create CA Bundle gdd_cert_bundle containing he certificate chain for Certificates gdd_cert in regions Rp and Rs Instructions: Creating a CA Bundle

User-Managed Data Distribution, Single Region

In this 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 gdd_vault_R1 Instructions: Creating a Vault
Certificate Authority Key	• In region R1, create key gdd_ca_key_R1 in vault gdd_vault_R1 Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 2048 Instructions: Create a Master Encryption Key
TDE Keys	• In region R1, create key gdd_TDE_key-catalog in vault gdd_vault_R1 for encrypting the catalog • In region R1, create key gdd_TDE_key-spaceN in vault gdd_vault_R1 for encrypting the shards in shard space N Required attribute values: • Protection Mode = Software • Key Shape: Algorithm = AES • Length = 256 Instructions: Create a Master Encryption Key
Certificate Authority	Create a CA for issuing certificates for Cloud Autonomous VM Clusters and GSM compute instances. • In region R1, using key gdd_ca_key_R1, create CA gdd_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. Instructions: Creating a Certificate Authority
Certificate	Create a Certificate for upload to Cloud Autonomous VM Clusters. • In region R1, using CA key gdd_ca_R1, create Certificate gdd_cert Instructions: Creating a Certificate
CA Bundle	Create a CA Bundle for upload to Cloud Autonomous VM Clusters. • In region R1, create a CA Bundle gdd_cert_bundle containing the certificate chain for Certificate gdd_cert Instructions: Creating a CA Bundle

User-Managed Data Distribution, Multiple Regions

In this 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 gdd_vault_R1 • In region R2, create vault gdd_vault_R2 ... • In region Rn, create vault gdd_vault_Rn Instructions: Creating a Vault
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, gdd_vault_Rp_Rs, in the database's primary region, Rp, that is replicated to the database's standby region, Rs. Replicating a Vault and Keys
Certificate Authority Keys	• In region R1, create key gdd_ca_key_R1 in vault gdd_vault_R1 • In region R2, create key gdd_ca_key_R2 in vault gdd_vault_R2 ... • In region Rn, create key gdd_ca_key_Rn in vault gdd_vault_Rn Required attribute values: • Protection Mode = HSM • Key Shape: Algorithm = RSA • Length = 2048 Instructions: Create a Master Encryption Key
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 gdd_TDE_key-catalog for the catalog database in the vault in the region where the catalog's database is placed • Create key gdd_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 gdd_TDE_key-catalog in the replicated virtual vault in the region where the catalog's primary database is placed • Create key gdd_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 Instructions: Create a Master Encryption Key

Resource	Instructions and Examples
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 gdd_ca_key_R1, create CA gdd_ca_R1 • In region R2, using key gdd_ca_key_R2, create CA gdd_ca_R2 ... • In region Rn, using key gdd_ca_key_Rn, create CA gdd_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. Instructions: Creating a Certificate Authority
Certificates	Create Certificates in each region for 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 gdd_ca_R1, create Certificate gdd_cert • In region R2, using CA gdd_ca_R2, create Certificate gdd_cert ... • In region Rn, using CA gdd_ca_Rn, create Certificate gdd_cert Instructions: Creating a Certificate
CA Bundles	Create the CA Bundles for upload to Cloud Autonomous VM Clusters. • In region R1, create CA Bundle gdd_cert_bundle containing the certificate chain for Certificates gdd_cert in regions R1, R2, ..., Rn • In region R2, create CA Bundle gdd_cert_bundle containing the certificate chain for Certificates gdd_cert in regions R1, R2, ..., Rn ... • In region Rn, create CA Bundle gdd_cert_bundle containing the certificate chain for Certificates gdd_cert in regions R1, R2, ..., Rn Instructions: Creating a CA Bundle

Task 6. Create Exadata Resources

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

- Exadata Resource Considerations
- Create Exadata Infrastructure Instances
- Import Oracle-ApplicationName Tag Namespace
- Create Cloud Autonomous VM Clusters

Exadata Resource Considerations

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 gdd/gdd_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 a cluster for each database in the Globally Distributed Database topology.

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

While creating the clusters make sure to do the following:

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

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

  Before you can add the tag to an Autonomous Exadata VM Cluster, you must import the tag’s namespace.

> **Note:**
>
> Once you tag a cluster for use in a Globally Distributed Database, it will continue to bill for the Globally Distributed Database SKU until the cluster is deleted.

- Create clusters in gdd/gdd_clusters compartment.
- **For release 23ai:** If you plan to use 23ai databases, check the prerequisites section in Create an Autonomous Exadata VM Cluster for 23ai database software version requirements.
- When the clusters are set up they need to be set to the same time zone.
- It is recommended that you use one VM cluster per database (shard or catalog).

Task 7. Upload the Cloud Autonomous VM Cluster Certificates

As the certificate administrator, you created the certificate authority, certificates, and CA bundle in the gdd/gdd_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.

(Optional) Create API Key and User Constraints

Create an OCI API key pair if you intend to directly use the Globally Distributed 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-database/latest/

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

Create and Manage a 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 topics that follow explain how to configure, deploy, and perform operations on 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.
- 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.
- Listing Globally Distributed Databases
- Viewing Globally Distributed Autonomous Database Details
  You view Globally Distributed Autonomous Database configuration, backup, and maintenance information by going to its Details page.
- Managing Certificates
  You must upload signed certificates for the shard directors (GSMs) to the Globally Distributed Autonomous Database before you can deploy the configuration.
- 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.
- Downloading Client Credentials
  You need the client credentials and connection information to connect to your Globally Distributed Autonomous Database. The client credentials include the wallet.
- Adding Shards
  Add shards to scale out your Globally Distributed Autonomous Database
- Modifying Shards
  You can modify a shard's ECPU count, auto-scaling setting, and storage allocation.
- 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.
- Stopping a Globally Distributed Autonomous Database
- Starting a Globally Distributed Autonomous Database
- 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.

- Moving Globally Distributed Autonomous Database Resources
  You can move a Globally Distributed Autonomous Database from one compartment to another.
- 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.
- Updating the Display Name
  You can change the display name of a Globally Distributed Autonomous Database from its details page.
- Managing Tags
  Tags help you locate resources within your tenancy.
- Globally Distributed Autonomous Database REST APIs
  The following REST APIs are used to interact with the Globally Distributed Autonomous Database (distributed-autonomous-database) resource.

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. **Note:** Deployment must take place within 7 days of completing the operation in Creating a Globally Distributed Autonomous Database Resource, or you must terminate the resources	Deploying Globally Distributed Autonomous Database

Task	Description	More Information
	and start again.	

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.
Deployment type	This setting is not configurable. Only Dedicated Infrastructure is supported.
Database version	You can select release 19c or 23ai
Workload type	This setting is not configurable. Only Transaction Processing is supported.

4. In step 2, Configure Shards and Catalog, in **Configure Shards**, provide the following information according to the release you selected previously.

   19c Configuration Settings

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. **Note:** It is recommended that you use one VM cluster per database (shard or catalog).
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. **Note:** You can select a cluster that contains a primary shard for a Data Guard standby database; however, it is recommended that you use one VM cluster per database (shard or catalog).
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)** . **Note:** The **ONS port (remote)** number must be unique to each Globally Distributed Autonomous Database. Do not reuse a port number used in another Globally Distributed Autonomous Database unless a delete operation is fully processed on the original.

Setting	Description and Notes
TLS	**TLS port** - TLS port number **Note:** The **TLS port** number must be unique to each Globally Distributed Autonomous Database. Do not reuse a port number used in another Globally Distributed Autonomous Database until a delete operation is fully processed on the original. **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.

23ai Configuration Settings

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. Note that when Raft is selected in **Replication type** the User managed option is disabled.

Setting	Description and Notes
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.
Replication type	**Raft** replication creates replication units consisting of sets of chunks and distributes them automatically among the shards to handle chunk assignment, chunk movement, workload distribution, and balancing upon scaling. Note that when Raft is selected the User managed data distribution option is disabled. **Data Guard** is a shard-level replication solution which instantiates Oracle Data Guard standby databases for each shard.
Replication factor	If **Raft** replication type is selected, you can set the **Replication factor**. Replication factor is the number of replicas in a replication unit. This number includes the leader replica and its followers.
Shard	Shows the display name for each shard or shardspace in the configuration. Once you select a region the name is populated.
Region/**Primary region**	Select the region where you would like to host your shard If Data Guard is the selected replication type this is the **Primary region**. Automated data distribution with Data Guard replication type does not support shards in multiple regions.
VM cluster/**Primary VM cluster**	Select a cluster available in the selected region. If Data Guard is the selected replication type this is the **Primary VM cluster**. **Note:** It is recommended that you use one VM cluster per database (shard or catalog).

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
Data Guard	If **Data Guard** is the selected replication type, this toggle enables or disables Data Guard replication on the selected shard. If enabled, an Oracle Data Guard standby database is instantiated for the shard.
Data Guard region	If **Data Guard** is the selected replication type, select the region where you would like to host the shard's Data Guard standby
Data Guard VM Cluster	If **Data Guard** is the selected replication type, select a cluster available in the selected Data Guard region. **Note:** You can select a cluster that contains a primary shard for a Data Guard standby database; however, it is recommended that you use one VM cluster per database (shard or catalog).
Configure Catalog	You can choose to use the same configuration that is applied to the shards, or uncheck the **Same as Shard's configuration** box and make selections that apply only to the catalog database. The same fields are as described above for Shards. Note that Raft replication type does not apply to the catalog. You can uncheck **Same as Shard's configuration** and configure Data Guard if you want catalog replication.

Setting	Description and Notes
Create administrator credentials	Create the user that will be able to access the shard catalog and all of the shards in the configuration.
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)** . **Note:** The **ONS port (remote)** number must be unique to each Globally Distributed Autonomous Database. Do not reuse a port number used in another Globally Distributed Autonomous Database unless a delete operation is fully processed on the original.

Setting	Description and Notes
TLS	**TLS port** - TLS port number **Note:** The **TLS port** number must be unique to each Globally Distributed Autonomous Database. Do not reuse a port number used in another Globally Distributed Autonomous Database until a delete operation is fully processed on the original. **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: Replication unit	Available for release 23ai only If **Raft** replication type is selected, you can configure **Replication unit**. Under Advanced Options you can optionally configure the number of replication units created for the Globally Distributed Autonomous Database. When Raft replication is enabled, a Globally Distributed Autonomous Database contains multiple **replication units**. A replication unit is a set of chunks that have the same replication topology.
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.

Listing Globally Distributed Databases

- Listing Globally Distributed Autonomous Database Resources

Listing Globally Distributed Autonomous Database Resources

Open the **navigation menu** and select **Oracle Database**. Then select **Globally Distributed Autonomous Database**.

> **Note:**
>
> The **navigation menu** is the main menu located in the upper-left corner of the Oracle Cloud Console. Use the menu to navigate to OCI services, dashboards, and marketplace.

The list of distributed databases is shown by default.

Viewing Globally Distributed Autonomous Database Details

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

Finding the 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 (Shown for release 19c)

The Shards and Catalog tab displays 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.

Shards Tab (Shown for release 23ai)

The Shards tab displays a list of all of the shards with their configuration settings.

If Raft replication type is configured you can toggle **Show replication units** to see the status of the replication unit leaders and followers on each shard.

Catalog Tab (Shown for release 23ai)

The Catalog tab displays the configuration settings for the catalog database.

Replication Unit Tab (Shown for release 23ai)

If your Globally Distributed Autonomous Database was configured with Raft replication type, this tab displays a list of the replication units by ID number. An icon indicates the status of the individual replication unit members and each member is labeled with the shard it resides on.

Work Requests (Shown for all releases)

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

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 there is no certificate available for download, go back to the Manage Certificates panel and click **Generate CSR**. In the Generate CSR dialog select the region and CA bundle with which to generate the CSR, and click **Generate CSR**.
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.

> **Note:**
>
> Deployment must take place within 7 days of completing the operation in Creating a Globally Distributed Autonomous Database Resource or Adding Shards, or you must terminate the resources and start again.

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 Globally Distributed Autonomous 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.

Adding Shards

Add shards to scale out your Globally Distributed Autonomous Database

You can add shards when:

- You have completed Creating a Globally Distributed Autonomous Database Resource, but have not yet completed Deploying Globally Distributed Autonomous Database.
- You have completed Deploying Globally Distributed Autonomous Database and want to scale up your Globally Distributed Autonomous Database with more shards.

1. On the **Details** page, on the **Shards and Catalog** tab, select **Add Shard**.
2. 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. You can add up to 10 shards in each set to deploy, and then add more after deployment if needed.

   - **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.

> **Note:**
>
> It is recommended that you use one VM cluster per database (shard or catalog).

- **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.

> **Note:**
>
> You can select a cluster that contains a primary shard for a Data Guard standby database; however, it is recommended that you use one VM cluster per database (shard or catalog).

3. In **Create administrator credentials**, set the password for the shard database ADMIN user.
4. 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.
5. Click **Validate** to run checks to make sure the new shards are valid.
6. Once any validation errors are addressed and validation is successful, click **Add Shards** to deploy the new shards.

> **Note:**
>
> There is a time limit for deploying new shards.
>
> - When scaling up a deployed Globally Distributed Autonomous Database, you must complete Deploying Globally Distributed Autonomous Database within 7 days of completing this procedure or you will get an error and must terminate the new shard resources and start again.
> - When adding shards to an undeployed Globally Distributed Autonomous Database, you have 7 days from completing Creating a Globally Distributed Autonomous Database Resource to add any shards and complete Deploying Globally Distributed Autonomous Database.

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

Modifying Shards

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:**
>
> Move resource is not allowed if any GSM, shard, or catalog is in a failed state.

> **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**.

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.

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 (distributed-autonomous-database) resource.

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

REST API	Description
`AddDistributedAutonomousDatabaseGdsControlNode`	Adds a new Global Data Services control node for running GDSCTL commands on the Globally Distributed Autonomous Database
`ChangeDistributedAutonomousDatabaseCompartment`	Moves the specified Globally Distributed Autonomous Database and its dependent resources to the specified compartment
`ConfigureDistributedAutonomousDatabaseGsms`	Lets you configure new shard director (GSM) instances for the Globally Distributed Autonomous Database
`ConfigureDistributedAutonomousDatabaseSharding`	Lets you complete deployment of the specified Globally Distributed Autonomous Database
`CreateDistributedAutonomousDatabase`	Creates a new Globally Distributed Autonomous Database resource.
`DeleteDistributedAutonomousDatabase`	Deletes the specified Globally Distributed Autonomous Database
`DownloadDistributedAutonomousDatabaseGsmCertificateSigningRequest`	Downloads the common certificate signing request for GSMs as a <globalautonomousdb-prefix>.csr file, which can be generated using `GenerateDistributedAutonomousDatabaseGsmCertificateSigningRequest` Use this CSR file to generate the CA signed certificate, then as a next step use `UploadDistributedAutonomousDatabaseSignedCertificateAndGenerateWallet` to upload the CA signed certificate to the GSM, and generate wallets for the GSM instances of the Globally Distributed Autonomous Database.
`GenerateDistributedAutonomousDatabaseGsmCertificateSigningRequest`	Generates a common certificate signing request (CSR file) for the Globally Distributed Autonomous Database GSM instances. Use `DownloadDistributedAutonomousDatabaseGsmCertificateSigningRequest` to download the file.
`GenerateDistributedAutonomousDatabaseWallet`	Generates the wallet associated with the specified Globally Distributed Autonomous Database
`GetDistributedAutonomousDatabase`	Gets the details of the specified Globally Distributed Autonomous Database
`PatchDistributedAutonomousDatabase`	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.
`RotateDistributedAutonomousDatabasePasswords`	Rotate passwords for different components of the Globally Distributed Autonomous Database.
`StartDistributedAutonomousDatabase`	Starts the specified Globally Distributed Autonomous Database
`StopDistributedAutonomousDatabase`	Stops the specified Globally Distributed Autonomous Database

REST API	Description
UpdateDistributedAutonomousDatabase	Lets you change the display name and edit tags associated with a Globally Distributed Autonomous Database resource.
UploadDistributedAutonomousDatabaseSignedCertificateAndGenerateWallet	Uploads the CA signed certificate to the Globally Distributed Autonomous Database GSM instances, and generate wallets for the GSM instances.
ValidateDistributedAutonomousDatabaseNetwork	Validates the network connectivity between components of the Globally Distributed Autonomous Database
ListDistributedAutonomousDatabases	Gets a list of Globally Distributed Autonomous Databases

See Private Endpoint REST APIs for descriptions of the private endpoint REST APIs.

4 Create and Manage Private Endpoints

A private endpoint is required in the Ashburn region to connect Oracle Cloud databases running in a customer VCN to the Globally Distributed Database services.

You create the private endpoint as part of setting up your network resources in Task 4. Configure Network Resources. For general information about private endpoints, see About Private Endpoints.

The topics that follow describe the steps for creating a private endpoint for a Globally Distributed Database and the life cycle operations on an existing private endpoint.

- Creating a Private Endpoint
- Listing Private Endpoints
- Viewing Private Endpoint Details
- Editing Private Endpoints
- Moving Private Endpoints
- Private Endpoint REST APIs

Creating a Private Endpoint

You create a private endpoint in the Private Endpoints list page. To find the Private Endpoints list page, see Listing Private Endpoints.

1. In the **Private Endpoints** list page select **Create private endpoint**.
2. 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 Ashburn region subnet that you created in Task 4. Configure Network Resources.
   - **Subnet in *compartment*:** Choose the subnet you created in Task 4. Configure Network Resources.
   - **Virtual cloud network in *compartment*:** Select a VCN
3. Optionally, you can select tags for this resource by clicking **Show Tagging Options**.

Listing Private Endpoints

- Listing Private Endpoints for Globally Distributed Autonomous Database

Listing Private Endpoints for Globally Distributed Autonomous Database

1. Open the **navigation menu** and select **Oracle Database**. Then select **Globally Distributed Autonomous Database**.

   > **Note:**
   >
   > The **navigation menu** is the main menu located in the upper-left corner of the Oracle Cloud Console. Use the menu to navigate to OCI services, dashboards, and marketplace.

2. On the left side of the screen, select **Private Endpoints**.

   A list of existing private endpoints is displayed.

Viewing Private Endpoint Details

To find a private endpoint's details, go the the Private Endpoints list page and select a private endpoint from the list. To find the Private Endpoints list page, see Listing Private Endpoints.

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

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 sections (tabs) which show configuration information and tags.

The details page also lets you view private endpoint-related related Work Requests and any Distributed Databases that use this private endpoint.

Editing Private Endpoints

You can edit a private endpoint in the Private Endpoints list page. To find the Private Endpoints list page, see Listing Private Endpoints.

In the list, select **Edit private endpoint** from the Actions (three dots) menu for the private endpoint you want to make changes to.

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

Moving Private Endpoints

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

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

   To find the Private Endpoints list page, see Listing Private Endpoints.
   You can also select **Move Resource** on the private endpoint's 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.

Private Endpoint REST APIs

The following REST APIs are used to interact with the Distributed Database Private Endpoint resource.

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

REST API	Description
`ChangeDistributedDatabasePrivateEndpointCompartment`	Moves the private endpoint to the specified compartment.
`CreateDistributedDatabasePrivateEndpoint`	Creates a private endpoint.
`DeleteDistributedDatabasePrivateEndpoint`	Deletes a private endpoint.
`GetDistributedDatabasePrivateEndpoint`	Gets a private endpoint.
`ReinstateProxyInstance`	Reinstates the proxy instance associated with the private endpoint
`UpdateDistributedDatabasePrivateEndpoint`	Updates private endpoint configuration details.
`ListDistributedDatabasePrivateEndpoints`	Lists private endpoints.

See Globally Distributed Database Policies for API permissions and policy guidelines.

5 Monitoring a Globally Distributed Database

Monitoring Work Requests

Globally Distributed Databases use their own APIs for Work Requests.

Using the Console:

Work request status is displayed in a Globally Distributed Database's details page.

From the Globally Distributed Database list page, click any database name and go to its details page. To find the Globally Distributed Database list page, see Listing Globally Distributed Databases.

The **Work requests** section 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.

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.

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

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 distributed 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 5-1 Event Types for Globally Distributed Autonomous Database

Friendly Name	Event Type
Distributed Autonomous Database - Add GDSCTL Node Begin	`com.oraclecloud.globaldb.adddistributedautonomousdatabasegdscontrolnode.begin`
Distributed Autonomous Database - Add GDSCTL Node End	`com.oraclecloud.globaldb.adddistributedautonomousdatabasegdscontrolnode.end`
Distributed Autonomous Database - Change Compartment Begin	`com.oraclecloud.globaldb.changedistributedautonomousdatabasecompartment.begin`
Distributed Autonomous Database - Change Compartment End	`com.oraclecloud.globaldb.changedistributedautonomousdatabasecompartment.end`
Distributed Autonomous Database - Configure Sharding Begin	`com.oraclecloud.globaldb.configuredistributedautonomousdatabasesharding.begin`
Distributed Autonomous Database - Configure Sharding End	`com.oraclecloud.globaldb.configuredistributedautonomousdatabasesharding.end`
Distributed Autonomous Database - Configure GSMs Begin	`com.oraclecloud.globaldb.configuredistributedautonomousdatabasegsms.begin`
Distributed Autonomous Database - Configure GSMs End	`com.oraclecloud.globaldb.configuredistributedautonomousdatabasegsms.end`
Distributed Autonomous Database - Create Begin	`com.oraclecloud.globaldb.createdistributedautonomousdatabase.begin`

Table 5-1 (Cont.) Event Types for Globally Distributed Autonomous Database

Friendly Name	Event Type
Distributed Autonomous Database - Create End	com.oraclecloud.globaldb.createdistributedautonomousdatabase.end
Distributed Autonomous Database - Delete Begin	com.oraclecloud.globaldb.deletedistributedautonomousdatabase.begin
Distributed Autonomous Database - Delete End	com.oraclecloud.globaldb.deletedistributedautonomousdatabase.end
Distributed Autonomous Database - Download GSM Certificate Signing Request	com.oraclecloud.globaldb.downloaddistributedautonomousdatabasegsmcertificatesigningrequest
Distributed Autonomous Database - Fetch Cloud Autonomous VM Clusters	com.oraclecloud.globaldb.fetchdistributedautonomousdatabasevmclusters
Distributed Autonomous Database - Generate GSM Certificate Signing Request Begin	com.oraclecloud.globaldb.generatedistributedautonomousdatabasegsmcertificatesigningrequest.begin
Distributed Autonomous Database - Generate GSM Certificate Signing Request End	com.oraclecloud.globaldb.generatedistributedautonomousdatabasegsmcertificatesigningrequest.end
Distributed Autonomous Database - Generate Wallet	com.oraclecloud.globaldb.generatedistributedautonomousdatabasewallet
Distributed Autonomous Database - Patch Begin	com.oraclecloud.globaldb.patchdistributedautonomousdatabase.begin
Distributed Autonomous Database - Patch End	com.oraclecloud.globaldb.patchdistributedautonomousdatabase.end
Distributed Autonomous Database - Prevalidate	com.oraclecloud.globaldb.prevalidatedistributedautonomousdatabase
Distributed Autonomous Database - Start Begin	com.oraclecloud.globaldb.startdistributedautonomousdatabase.begin
Distributed Autonomous Database - Start End	com.oraclecloud.globaldb.startdistributedautonomousdatabase.end
Distributed Autonomous Database - Stop Begin	com.oraclecloud.globaldb.stopdistributedautonomousdatabase.begin
Distributed Autonomous Database - Stop End	com.oraclecloud.globaldb.stopdistributedautonomousdatabase.end
Distributed Autonomous Database - Update	com.oraclecloud.globaldb.updatedistributedautonomousdatabase
Distributed Autonomous Database - Upload Signed Certificate And Generate Wallet Begin	com.oraclecloud.globaldb.uploaddistributedautonomousdatabasesignedcertificateandgeneratewallet.begin
Distributed Autonomous Database - Upload Signed Certificate And Generate Wallet End	com.oraclecloud.globaldb.uploaddistributedautonomousdatabasesignedcertificateandgeneratewallet.end
Distributed Autonomous Database - Validate Network Begin	com.oraclecloud.globaldb.validatedistributedautonomousdatabasenetwork.begin
Distributed Autonomous Database - Validate Network End	com.oraclecloud.globaldb.validatedistributedautonomousdatabasenetwork.end

Table 5-2 Event Types for Distributed Database Private Endpoint

Friendly Name	Event Type
Distributed Database Private Endpoint - Change Compartment Begin	`com.oraclecloud.globaldb.changedistributeddatabaseprivateendpointcompartment.begin`
Distributed Database Private Endpoint - Change Compartment End	`com.oraclecloud.globaldb.changedistributeddatabaseprivateendpointcompartment.end`
Distributed Database Private Endpoint - Create Begin	`com.oraclecloud.globaldb.createdistributeddatabaseprivateendpoint.begin`
Distributed Database Private Endpoint - Create End	`com.oraclecloud.globaldb.createdistributeddatabaseprivateendpoint.end`
Distributed Database Private Endpoint - Delete Begin	`com.oraclecloud.globaldb.deletedistributeddatabaseprivateendpoint.begin`
Distributed Database Private Endpoint - Delete End	`com.oraclecloud.globaldb.deletedistributeddatabaseprivateendpoint.end`
Distributed Database Private Endpoint - Update	`com.oraclecloud.globaldb.updatedistributeddatabaseprivateendpoint`

6

Globally Distributed Database Policies

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

- Giving Permissions to Users
- Required Policies
- Resource-Types
- Resource-Permissions Model
- Permissions for Globally Distributed Autonomous Database APIs
- Details for Verbs + Resource-Type Combinations
- Supported Variables

Giving Permissions to Users

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

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

Create user groups to manage Globally Distributed 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 distributed databases.

Required Policies

Several users, groups, and policies are required to set up and run a Globally Distributed Database.

See Task 3. Create User Access Constraints for complete nistructions and lists.

Resource-Types

Oracle's Globally Distributed Autonomous Database service offers individual resource-types for writing policies.

Resource-Type	Description
`distributed-autonomous-database`	Configuration of the Globally Distributed Autonomous Database, including the data distribution model and information for connecting to the shards and catalog databases.

Resource-Type	Description
`distributed-database-privateendpoint`	A private endpoint used to connect databases running in a customer VCN to the Globally Distributed Database services.
`distributed-database-workrequest`	Monitor for long-running operations, such as shard creation, update, or deletion.

Resource-Permissions Model

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

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

Resource	Permissions
`distributed-autonomous-database`	• DISTRIBUTED_DB_INSPECT • DISTRIBUTED_DB_READ • DISTRIBUTED_DB_MANAGE • DISTRIBUTED_DB_MOVE • DISTRIBUTED_DB_CREATE • DISTRIBUTED_DB_DELETE
`distributed-database-privateendpoint`	• DISTRIBUTED_DB_PRIVATE_ENDPOINT_INSPECT • DISTRIBUTED_DB_PRIVATE_ENDPOINT_READ • DISTRIBUTED_DB_PRIVATE_ENDPOINT_MANAGE • DISTRIBUTED_DB_PRIVATE_ENDPOINT_MOVE • DISTRIBUTED_DB_PRIVATE_ENDPOINT_CREATE • DISTRIBUTED_DB_PRIVATE_ENDPOINT_DELETE
`distributed-database-work-requests`	• DISTRIBUTED_DB_WORK_REQUEST_LIST • DISTRIBUTED_DB_WORK_REQUEST_READ

Permissions for Globally Distributed Autonomous Database APIs

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

- Distributed-autonomous-database API permissions
- Distributed-database-privateendpoint API permissions
- Distributed-database-workrequest API permissions

Distributed-autonomous-database API permissions

API names and permissions for distributed-autonomous-database resource-type

Table 6-1 Distributed-autonomous-database API permissions

API Operation	Permission
`AddDistributedAutonomousDatabaseGdsControlNode`	DISTRIBUTED_DB_MANAGE
`ChangeDistributedAutonomousDatabaseCompartment`	DISTRIBUTED_DB_MOVE
`ConfigureDistributedAutonomousDatabaseGsms`	DISTRIBUTED_DB_MANAGE
`ConfigureDistributedAutonomousDatabaseSharding`	DISTRIBUTED_DB_MANAGE
`CreateDistributedAutonomousDatabase`	DISTRIBUTED_DB_CREATE
`DeleteDistributedAutonomousDatabase`	DISTRIBUTED_DB_DELETE
`DownloadDistributedAutonomousDatabaseGsmCertificateSigningRequest`	DISTRIBUTED_DB_MANAGE
`GenerateDistributedAutonomousDatabaseGsmCertificateSigningRequest`	DISTRIBUTED_DB_MANAGE
`GenerateDistributedAutonomousDatabaseWallet`	DISTRIBUTED_DB_READ
`GetDistributedAutonomousDatabase`	DISTRIBUTED_DB_READ
`PatchDistributedAutonomousDatabase`	DISTRIBUTED_DB_MANAGE
`RotateDistributedAutonomousDatabasePasswords`	DISTRIBUTED_DB_MANAGE
`StartDistributedAutonomousDatabase`	DISTRIBUTED_DB_MANAGE
`StopDistributedAutonomousDatabase`	DISTRIBUTED_DB_MANAGE
`UpdateDistributedAutonomousDatabase`	DISTRIBUTED_DB_MANAGE
`UploadDistributedAutonomousDatabaseSignedCertificateAndGenerateWallet`	DISTRIBUTED_DB_MANAGE
`ValidateDistributedAutonomousDatabaseNetwork`	DISTRIBUTED_DB_MANAGE
`ListDistributedAutonomousDatabases`	DISTRIBUTED_DB_INSPECT

Distributed-database-privateendpoint API permissions

API names and permissions for distributed-database-privateendpoint resource-type

Table 6-2 Distributed-database-privateendpoint API permissions

API Operation	Permissions
`ChangeDistributedDatabasePrivateEndpointCompartment`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_MOVE
`CreateDistributedDatabasePrivateEndpoint`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_CREATE
`DeleteDistributedDatabasePrivateEndpoint`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_DELETE
`GetDistributedDatabasePrivateEndpoint`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_READ
`ReinstateProxyInstance`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_MANAGE
`UpdateDistributedDatabasePrivateEndpoint`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_MANAGE
`ListDistributedDatabasePrivateEndpoints`	DISTRIBUTED_DB_PRIVATE_ENDPOINT_INSPECT

Distributed-database-workrequest API permissions

API names and permissions for distributed-database-workrequest resource-type

Table 6-3 Distributed-database-workrequest API permissions

API Operation	Permission
`GetWorkRequest`	DISTRIBUTED_DB_WORK_REQUEST_READ
`ListWorkRequests`	DISTRIBUTED_DB_WORK_REQUEST_LIST
`ListWorkRequestErrors`	DISTRIBUTED_DB_WORK_REQUEST_READ
`ListWorkRequestLogs`	DISTRIBUTED_DB_WORK_REQUEST_READ

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 Database.

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

- Distributed-autonomous-database
- Distributed-database-privateendpoint
- Distributed-database-workrequest

Distributed-autonomous-database

Permission	APIs Fully Covered
INSPECT	
DISTRIBUTED_DB_INSPECT	`ListDistributedAutonomousDatabases`
READ	
INSPECT +	INSPECT+

DISTRIBUTED_DB_READ	`DownloadDistributedAutonomousDatabaseGsmCertificateSigningRequest`
	`GenerateDistributedAutonomousDatabaseWallet`
	`GetDistributedAutonomousDatabase`

MANAGE	
READ +	READ +

DISTRIBUTED_DB_MANAGE	`AddDistributedAutonomousDatabaseGdsControlNode`
	`ConfigureDistributedAutonomousDatabaseGsms`
	`ConfigureDistributedAutonomousDatabaseSharding`
	`GenerateDistributedAutonomousDatabaseGsmCertificateSigningRequest`
	`PatchDistributedAutonomousDatabase`
	`RotateDistributedAutonomousDatabasePasswords`
	`StartDistributedAutonomousDatabase`
	`StopDistributedAutonomousDatabase`
	`UpdateDistributedAutonomousDatabase`
	`UploadDistributedAutonomousDatabaseSignedCertificateAndGenerateWallet`
	`ValidateDistributedAutonomousDatabaseNetwork`

DISTRIBUTED_DB_MOVE	`ChangeDistributedAutonomousDatabaseCompartment`
CREATE	
UPDATE+	UPDATE+
DISTRIBUTED_DB_CREATE	`CreateDistributedAutonomousDatabase`
DELETE	
CREATE+	CREATE+
DISTRIBUTED_DB_DELETE	`DeleteDistributedAutonomousDatabase`

Distributed-database-privateendpoint

Permission	APIs Fully Covered
INSPECT	
DISTRIBUTED_DB_PRIVATE_ENDPOINT_INSPECT	`ListDistributedDatabasePrivateEndpoints`
READ	
INSPECT +	INSPECT+
DISTRIBUTED_DB_PRIVATE_ENDPOINT_READ	`GetDistributedDatabasePrivateEndpoint`
MANAGE	
READ +	READ +
DISTRIBUTED_DB_PRIVATE_ENDPOINT_MANAGE	`UpdateDistributedDatabasePrivateEndpoint` `ReinstateProxyInstance`
DISTRIBUTED_DB_PRIVATE_ENDPOINT_MOVE	`ChangeDistributedDatabasePrivateEndpointCompartment`
CREATE	
UPDATE+	UPDATE+
DISTRIBUTED_DB_PRIVATE_ENDPOINT_CREATE	`CreateDistributedDatabasePrivateEndpoint`
DELETE	
CREATE+	CREATE+
DISTRIBUTED_DB_PRIVATE_ENDPOINT_DELETE	`DeleteDistributedDatabasePrivateEndpoint`

Distributed-database-workrequest

Permission	APIs Fully Covered
INSPECT	
DISTRIBUTED_DB_WORK_REQUEST_LIST	`ListWorkRequests`
READ	
INSPECT +	INSPECT+

DISTRIBUTED_DB_WORK_REQUEST_READ	`GetWorkRequest`
	`ListWorkRequestErrors`
	`ListWorkRequestLogs`

Supported Variables

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

Oracle's Globally Distributed Database services support all general variables. For more information, see general variables for all requests.