**CHAPTER 8**

**OCI Best Practice Architectures**

In this chapter, you will learn how to

•   Design Highly Available Disaster Recovery (HADR) OCI solutions

•   Leverage OCI security features to protect your cloud infrastructure

High Availability (HA) and Disaster Recovery (DR) are interrelated concepts and go hand-in-hand with each other. A robust design that is fault tolerant, uses redundant components, and ensures application or system availability in a manner that is transparent to end users is considered an HADR architecture. The best HADR is when users do not even know that anything has failed because they experienced zero downtime and no loss of service. Disaster Recovery refers to the ability for systems to continue operating after an outage with zero or minimal loss of data. We define HA as availability in relation to service while DR is defined as availability in relation to data. These concepts represent availability and recovery goals. Their definitions overlap as technology converges, resulting in the hybrid term HADR used to collectively describe design for both HA and DR.

Good examples of HADR are Oracle RAC databases with Data Guard standby databases. A RAC database is a single set of database files concurrently accessed by more than one database instance running on more than one node. If a RAC node fails, database availability is not affected, and applications being serviced by the RAC database continue uninterrupted, oblivious to the fact that a node failure just occurred. If the entire primary site where all RAC instances reside is lost, a Data Guard failover occurs to the standby database at a secondary site. Service is temporarily impacted while the failover and data recovery complete following a disaster.

HADR designs include eliminating single points of failure by leveraging redundant components. In the previous example, RAC protects availability against loss of database nodes or instances while Data Guard protects against loss of the entire primary database because it effectively keeps a redundant copy of the database at the standby location. This configuration is usually further protected using database backups. If a database server experiences a catastrophic failure and a database is lost, you may have to restore the server and database from backups, which are yet other redundant offline copies of the database. Service is often interrupted while data is being restored and recovered.

This chapter covers OCI design principles in relation to HADR. A discussion of good security practices in OCI follows later in the chapter. Security in OCI must be considered when designing cloud-based solutions. Understanding the deceptively powerful security features available in OCI allows you to leverage these features in designing secure cloud architectures. OCI gives many organizations an opportunity for a fresh start when designing their security posture. You may have inherited archaic security policies that are no longer working, exposing your systems to unnecessary risk, and now is the time to learn from past mistakes and design secure architectures with the wisdom of hindsight based on learning from your on-premises security approach. Contrary to early security-related fears that may have led to slow cloud adoption, the technology industry recognizes that cloud vendors have invested heavily in securing their offerings to the point that public clouds today ironically have security that is superior to many on-premises systems.

As you read through this chapter, you will encounter familiar OCI terms and concepts introduced in earlier chapters but described here through an HADR lens.

**Design Highly Available Disaster Recovery (HADR) OCI Solutions**

The basic design principle behind HADR is to eliminate single points of failure by leveraging redundant components. HADR is associated with two additional terms that reflect the organizational context within which these technical architectures are situated: RPO (Recovery Point Objective) and RTO (Recovery Time Objective).

RPO refers to how much data loss is tolerable for the organization in the event of a disaster. Ideally, you should strive for a zero data loss architecture. This ideal is accompanied by increased costs for redundant components and advanced software that reduces or eliminates data loss. The financial realities of an organization often dictate the RPO. It is common to see financial systems with zero data loss architectures, whereas non-production systems or systems where data can be recreated from downstream sources have less stringent RPO requirements.

RTO refers to the duration of a service outage. If the organizational requirement is to have zero downtime for some systems, this has implications for the underlying design and cost of the solution. Many systems can tolerate short outages. It is not ideal but can reduce the implementation costs significantly. Critical systems, usually governing life support and other high-risk environments, require zero downtime, and implementation costs for these designs tend to be material.



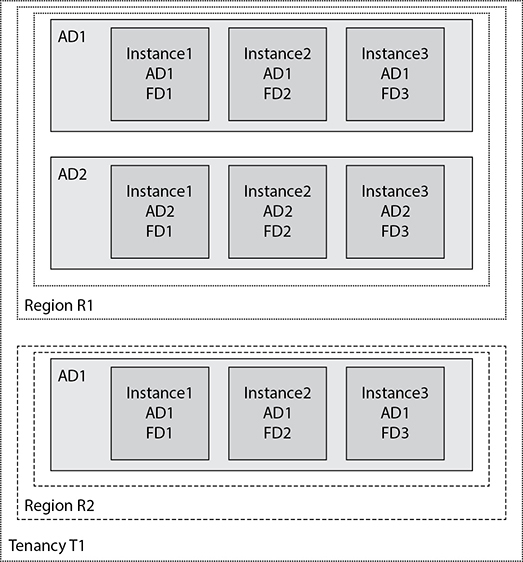
**CAUTION**    Understanding RPO and RTO for various systems in your organization is crucial and forms the basis for your solution. It is your duty to gather the input realities, design the most optimal HADR architecture given these realities, and highlight the risks and points of failure. It is equally important to present alternative (usually more costly) options where risks are mitigated to key stakeholders and to get written acceptance of the final solution and consequent service level agreement.

Any infrastructure resource is at risk of failure. It is your task to mitigate that risk by architecting a solution that is fault tolerant while meeting RPO and RTO requirements. This is an art that requires a solid understanding of the fault-tolerant options available in OCI.

**Regions and Availability Domains**

Fault tolerance may be found along the entire spectrum of OCI resources. At the highest level of the OCI resource hierarchy are regions. Your cloud account is your tenancy and upon creation is designated a home region. Your tenancy may subscribe to multiple regions. This permits you to create OCI resources in data centers across the globe. Consider a website serving customers on opposite ends of the earth. It is entirely conceivable to register your website domain name with a DNS provider that resolves to IP addresses of load balancers or compute instances in the Phoenix region in the United States and in the Bombay region in India, providing both HADR for your website and shorter network distance that requesting clients closer to these host regions need to traverse, leading to superior website access performance.

Regions comprise one or more availability domains. Regions with a single AD are less fault tolerant than regions with multiple ADs. [Figure 8-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8fig1) describes a tenancy subscribed to two regions. Region R1 has two ADs while region R2 has one AD.



**Figure 8-1**   Tenancy, regions, availability, and fault domains

Loss of the AD in region R2 equates to the loss of an entire region while in the multi-AD region R1; loss of an AD does not bring down the entire region. The relative seriousness of losing an AD must be placed in context. While losing an AD is a big deal, it is extremely uncommon and a low probability event. For many organizations, their current on-premises infrastructure still resides in a single data center. Loss of an entire data center or AD is usually the result of a large-scale disaster or wide-reaching sustained power outage. The nature of multi-AD regions is such that they are located relatively close to one another. This physical proximity is needed to sustain low network latency between the ADs in the region. Therefore, a disaster that knocks out an AD in a region could be large enough to wipe out all the ADs in that region. If there are multiple ADs in your region, you should architect your infrastructure in such a way that critical systems are protected from an AD failure. A common use case involves a regional load balancer distributing traffic to middleware webservers or application servers located in separate ADs within a region to ensure HA.

Single AD regions are not without protection. Redundancy has been engineered into the ADs through the provision of fault domains. Three fault domains exist per AD. A fault domain is a set of fault-tolerant isolated physical infrastructure within an AD. By choosing different fault domains for VM instances, you ensure these are hosted on separate physical hardware, thus increasing your intra-availability domain resilience. It is a good HADR DBaaS solution to use the Oracle RAC option and place RAC compute nodes in separate fault domains. In [Figure 8-1](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8fig1), three instances are shown in the single AD region R2. Instance1 in AD1 in FD1 could be one RAC node, and Instance2 in AD1 in FD2 could be the second node of a two-node DBaaS RAC cluster. Instance3 in AD1 in FD3 could be a single-instance Data Guard standby database system. Each instance is isolated from hardware failure by placing them in separate fault domains. Architecting redundant instances across fault domains, availability domains, and regions provides a multi-tiered approach to HA.

**VCNs, Load Balancers, and Compute Instances**

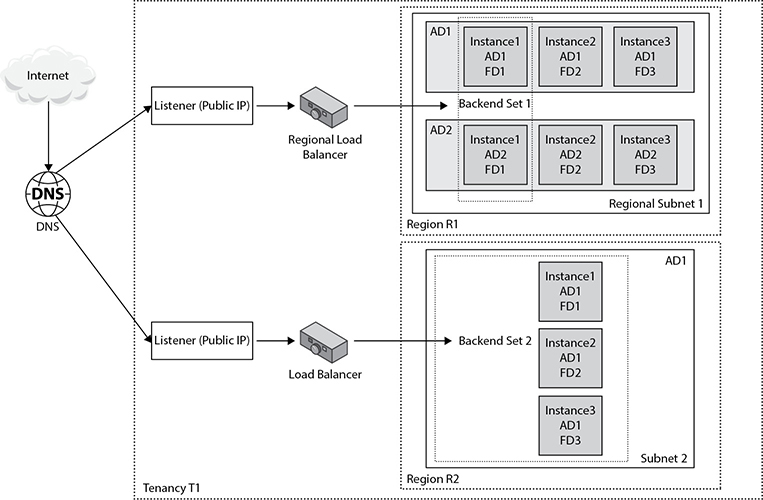
Your tenancy may subscribe to multiple regions. A VCN spans all the ADs in a region. Remote VCN peering can be used to connect VCNs securely and reliably across regions. VCNs are composed of subnets. Subnets and load balancers may be created at the AD or region level. Both subnet and load balancer HADR is improved when you use regional resources in a multi-AD region. Load balancers are commonly used for optimizing the utilization of backend resources as well as to provide scaling and high availability. A load balancer may be public or private and accepts incoming TCP or HTTP network traffic on a single IP address and distributes it to a backend set that comprises one or more compute instances in your VCN. In this context, the compute instances are known as backend servers. Each of these compute instances resides in either a public or private subnet.

When a private load balancer is created, you specify the VCN and private subnet to which it belongs. An active (primary) private load balancer obtains a private IP address from the CIDR range of a private subnet. A passive (standby) private load balancer is created automatically for failover purposes and also receives a private IP address from the same subnet. A floating private IP address serves as a highly available address of the load balancer. The active and passive private load balancers are highly available within a single AD. If the primary load balancer fails, the listener directs traffic to the standby load balancer and availability is maintained. A private load balancer is accessible from compute instances within the VCN where the subnet of the load balancer resides. You may improve the service availability across ADs by setting up multiple private load balancers and make use of private DNS servers to set up a round-robin DNS configuration with their IP addresses.

A public load balancer is allocated a floating public IP address that is routable from the Internet. When a public load balancer is created, two subnets are required, one for the active (primary) public load balancer and another for the passive load balancer. Incoming traffic from the public Internet on allowed ports and protocols is directed to the floating public IP address associated with the active load balancer. If the primary load balancer fails, the passive device is automatically made active. The public load balancer is a regional resource as opposed to a private load balancer, which is an AD-level resource. In regions with multiple ADs, it is mandatory to specify public subnets in different ADs for the active and passive load balancers.

Services are often accessed via DNS hostname resolution to an IP address of a load balancer or compute instance. Subnets provide primary private IP addresses to resident compute instances and private load balancers. Secondary private or reserved public IP addresses may be additionally allocated to compute instances. It is good HADR practice to bind services to DNS targets that resolve to secondary or floating IP addresses. In the event of a loss of a compute instance, the secondary IP address may be reassigned to a standby compute instance enabling a continuation of service. Secondary private IP addresses may only be assigned to standby compute instances resident in the same subnet as the primary instance. Reserved public IP addresses offer more flexibility and may be assigned to any standby compute instance and persist beyond the lifetime of the compute instance to which it is assigned.

[Figure 8-2](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8fig2) depicts a potential website architecture. You may register your domain name in a public DNS registry that resolves to multiple public IP addresses that belong to public load balancer listeners located in different regions. The regional load balancer in region R1 forwards traffic to a backend set of instances that reside in separate ADs but share a common regional subnet. The public load balancer in region R2 routes traffic to a backend set of instances that despite residing in a single AD are guaranteed to run on separate hardware.



**Figure 8-2**   HADR architecture using regional network resources across multiple regions

In this architecture, the website is protected from single server failure, entire AD failure, and even region failure, ensuring uninterrupted service and meeting stringent organizational HADR goals.



**NOTE**    When a compute instance using floating IP addresses fails, it is often possible to automate the allocation of the floating IP address to a standby instance to minimize downtime.

To failover to a standby compute instance using a floating IP address still leaves you with the problem of data being out of sync between the primary and standby compute instances. There are several approaches to maintaining data availability and ensuring data integrity that involve using shared file systems or keeping file systems in sync. Databases are discussed later in this chapter.

**VPN and FastConnect**

Connecting a VCN to your on-premises network creates a new point of failure that must be mitigated. As discussed in [Chapter 3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch3.xhtml#ch3), you may make use of IPSec VPN and FastConnect to connect discrete networks. You may set up a connection between your on-premises network’s edge router (CPE or customer-premises equipment) and an OCI Dynamic Routing Gateway (DRG). High availability is provided for the network path between the CPE and DRG through a set of redundant IPSec VPN tunnels. When an IPSec VPN is set up, two tunnels are created for redundancy. These may be used in active-active mode or active-passive mode.

OCI provides FastConnect as a means to create a dedicated high-speed private connection between on-premises networks and OCI. FastConnect provides consistent, predictable, secure, and reliable performance. FastConnect supports the following uses:

•   Private peering extends your on-premises network into a VCN and may be used to create a hybrid cloud. On-premises connections can be made to the private IP addresses of instances as if they were coming from instances in the VCN. Private peering can also occur between instances in VCNs in other regions.

•   Public peering allows you to connect from resources outside the VCN, such as an on-premises network, to public OCI services such as object storage, over FastConnect without traversing the Internet.

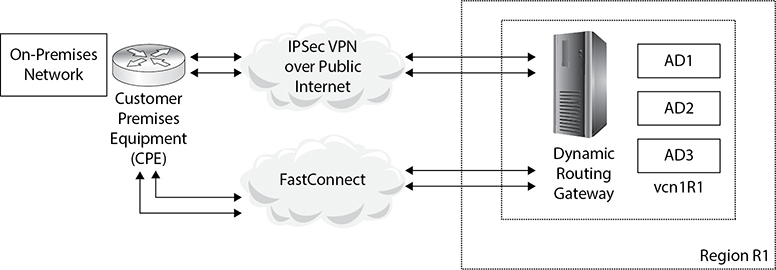
FastConnect is actualized using several connectivity models:

•   Colocation with Oracle allows direct physical cross-connects between your network and Oracle’s FastConnect edge devices. HADR is achieved by ensuring that at least two cross-connects are set up, each connecting to a different router. New virtual circuits should be provisioned on both redundant links.

•   Using an Oracle Network Provider or Exchange Partner, you can set up a FastConnect connection from your network to the provider or partner network that has a high bandwidth connection into Oracle’s FastConnect edge devices. To mitigate loss of a FastConnect location, you can set up redundant circuits using a different FastConnect location provided by either the same provider or a different provider. This enables redundancy at both the circuit and data center levels.

Planned maintenance on routers involved in the FastConnect setup can impact availability. This may be mitigated by providing multiple paths and changing traffic patterns to use preferred paths by configuring Border Gateway Protocol (BGP) route preferences.

An alternative VPN HADR architecture involves using both IPSec VPN and FastConnect to provide redundancy in case one of these connection options fails, as shown in [Figure 8-3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8fig3). When both IPSec VPN and FastConnect virtual circuits connect to the same DRG, consider that IPSec VPN uses static routing while FastConnect uses BGP. You need to test your configuration to ensure that if BGP routes overlap with IPSec VPN routes, then BGP routes are preferred by OCI as long as they are available.



**Figure 8-3**   FastConnect and IPSec VPN HADR connecting VCN to an on-premises network

There has also been an increase in the availability of software-defined wide area networking (SD-WAN) solutions in the realm of cloud networking that are worth considering. SD-WAN appliances reside at your network edge locations—for example, in your VCN and on-premises networks—and evaluate the performance of multiple network routes using analytics and routing algorithms to dynamically route network traffic along the most efficient path.

**Storage and Compute Instances**

File systems may be shared by compute instances in the same AD by using OCI file storage services, which provide an NFS-compliant shared network attached storage. If critical files are kept in a shared file storage location then loss of a compute instance may be tolerated with no service loss. An example of this may be a website where the static content or files with session state information are placed on shared file system. If multiple webserver compute instances reference the shared file storage, then a website may remain highly available even if an underlying webserver compute instance has failed.



**NOTE**    The file storage service also provides a snapshot-based backup mechanism that supports the immediate restoration of files accidentally removed due to user error, which further supports RPO and RTO.

When primary and standby compute instances are in separate ADs, file systems may be synchronized to provide some measure of HADR. There are two basic approaches to synchronizing file systems:

•   Synchronous replication refers to changes being shipped from the primary to the standby instance. Only once the changes are confirmed to have arrived at the standby are they committed on the primary site. Synchronous replication is dependent on network and block volume resources and can lead to queuing delays and waits, which could have a negative impact on performance. Instances in ADs in the same region benefit from the low-latency network backbone between the ADs and you may find that synchronous replication is a feasible approach that leads to synchronization between the two sites when no data loss can be tolerated.

•   Asynchronous replication refers to changes to the primary instance being shipped to the standby with no need to wait for confirmation that the changes have been received and applied. There is no blocking or waiting with this approach and it is suitable for synchronizing data between instances in different regions. There is a risk of data loss, however, because a change may be written to a primary instance, which fails before the change is sent to the standby instance.



**CAUTION**    DenseIO compute shapes include support for direct attached NVMe disks. This storage is not SAN-based and is presented as raw storage to the compute instance. There is no redundancy built in and it is your responsibility to set up appropriate redundancy. This could take the format of ASM redundancy if the NVMe volumes are used for Oracle databases or some sort of RAID configuration if they are used for generic file system storage. Configuring redundancy for locally attached NVMe storage provides redundancy and HADR in case of storage failure.

Object storage provides petabyte-scale storage that is resilient and has become a standard for general-purpose file storage in the cloud. The object storage system is Internet accessible, and you control the permissions and whether a bucket is publicly accessible or not. OCI object storage integrates with OCI’s Identity and Access Management (IAM) to control permissions on object storage. Object storage is not suitable for high-speed computing storage requirements (such as those required to run databases) but provides flexible and scalable options for unstructured data storage and sharing as well as being great for big data and content repositories, backups and archives, log data, and other large datasets. Object storage is also not bound to an instance or an AD but is a region-level construct that provides a highly available file storage solution.

**Performance-Based HADR**

Service availability is sometimes impacted by poorly performing systems. Common causes of performance-related service outage include undersized compute infrastructure for the actual workload, distributed denial of service (DDoS) attacks, and software quality. Buggy or problematic software quality cannot be solved through HADR design but through a systematic quality assurance test and release cycle. Although this topic is beyond the scope of this book, it has happened too often that poorly tested code is released into the wild, exposing enterprises to avoidable critical security- and performance-related bugs. The undersized compute infrastructure problem is one that is dealt with quite easily.

The cloud computing paradigm revolves around sizing resources for average workloads to contain costs while planning for bursting or scaling during peak periods. Imagine a website hosted on a single compute instance running an HTTP server. You can create a custom image from this compute instance and use it as the basis for automatically provisioning additional preconfigured web server compute instances when your website is overloaded. Not all resource classes are capable of autoscaling but two key resources benefit from this automation: compute instances and autonomous databases, discussed later in the chapter. DBaaS instances benefit from explicit CPU scaling to increase CPU capacity.

Instance configuration and pools provide the basis for autoscaling, which is the dynamic addition or removal of instances from an instance pool based on an autoscaling policy. Instance configuration metadata forms the basis for pools of compute instances created using these regional templates. You can configure the automatic spawning and removal of compute instances based on a common instance configuration as the workload changes. Instance pools are a regional construct and may contain instances from multiple ADs. By attaching load balancers to instance pools, new auto-provisioned instances are added to the relevant backend sets. This is a useful feature to consider when designing for performance-based compute instance HADR.

**Database HADR**

[Chapter 6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch6.xhtml#ch6) outlines the available database options in OCI. Oracle Database on OCI is a key driver to the adoption of OCI for many enterprises. Oracle databases usually support critical workloads and HADR is a primary consideration. Database backups are a key component of any HADR solution and are not discussed here. This section assumes you have read [Chapter 6](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch6.xhtml#ch6) or are familiar with these options for running Oracle databases on OCI.

**Single-Instance Databases (SI)**

You can manually configure and run a single-instance database on a compute instance or use DBaaS to have OCI provision a single-instance server. Automatically created single-instance DBaaS environments are preconfigured with Oracle Grid Infrastructure (GI). Because there are many points of failure similar to manually configured single-instance databases and similar approaches to mitigate these issues, these two methods of running single-instance databases are discussed together next. Manually configured instances offer both familiarity for on-premises DBAs as well as certain migration flexibility, but have limited HADR options. There are many points of failure, some of which are detailed next with suggestions to mitigate these issues.

•   With a single-instance database manually configured on a compute instance, once you lose your compute instance, the database is no longer accessible. You could clone your boot image to create a custom image and use this to create a new standby instance that is available in the same AD as the source instance. A floating IP address could be attached to the source instance. In the event of an unrecoverable loss of the source instance, you could reallocate the floating IP address, detach the block volumes with the Oracle Home and Oracle database files from the source instance, and attach it to the standby instance.

•   Access to the database is tied to the availability and accessibility of the database processes and listener process to database clients. If a listener fails, no new remote database connections are accepted. If a database process fails, so does the database. Oracle offers a technology (for single-instance databases) known as Oracle Restart, which is part of GI. GI should be installed for many reasons, especially for ASM. GI is installed with DBaaS instances. Oracle Restart periodically checks if key processes such as DB listeners, DB server processes, and ASM disk groups are available and restarts processes if a problem is detected. For Oracle RAC databases, similar functionality is managed by the Clusterware-related services.

•   Sufficient disk storage must be allocated to allow for growth of database, audit, trace, and log files, as well as any redo logs. If the database is running in archivelog mode (as it should be in most cases), there must be sufficient space for growth of archived redo logs. Use Oracle Enterprise Manager (OEM) or Oracle Management Cloud (OMC) to monitor your server and alert you proactively to disk and other issues that could affect availability.

•   Loss of key database files will also halt the database. If GI is installed, ASM disk groups may be configured with normal or high levels of redundancy, configuring either two or three copies of ASM extents respectively across the disks in the disk group. Placing key, or ideally all, database files in redundant disk groups ensures database availability.

**Real Application Clusters (RAC)**

RAC is an option that allows a set of database files to be mounted by multiple database instances concurrently. RAC relies on shared storage accessible by each instance participating in the cluster. RAC is included with all Exadata Cloud services, supporting up to eight nodes and is optional on VM-based DBaaS supporting a two-node format. RAC provides HADR for a database through two primary mechanisms:

•   A RAC database is concurrently mounted by one or more database instances, each running on a separate compute node within an AD. RAC databases can tolerate the loss of a RAC node. As long as there is at least one RAC node available, the database remains accessible.

•   Database listener services are provided by a special set of processes known as scan listeners that provide HADR for local database listeners. Three scan listeners are deployed on floating IP addresses and route network connection traffic to available local listeners on each RAC node based on their workload and availability. When a node is offline, the scan listeners relocate dynamically to available nodes to provide connection HADR to database clients.

RAC nodes or compute instances are created in separate fault domains within the same AD to protect the database from hardware failure. RAC additionally provides an HADR benefit during planned maintenance activities. Firmware on each RAC compute node may be updated, and even database software patches may be applied in a rolling manner to maximize database availability.

**Data Guard**

While RAC mitigates against node failure and ensures HA, Data Guard mitigates against node, shared storage, and even AD failure in multi-AD regions. Data Guard is a mature replication technology available with all Oracle Enterprise Edition versions. You may manually configure Data Guard between two manually configured databases (RAC or SI) or use the managed Data Guard DBaaS for OCI. The managed option is straightforward to set up and uses automation behind the scenes to ensure the standby is correctly configured. As of this writing, the managed Data Guard service is limited to inter-AD replication within a multi-AD region. You may choose to manually build a Data Guard standby for a manually configured database that is located in a different region or even on-premises. Keep in mind that network latency must be considered and may limit the replication mode, as discussed later in this section.

All Data Guard configurations consist of a primary database and at least one standby database. Each system is a fully operational Oracle server with nodes, instances, and independent sets of database files. The primary and standby systems are almost exclusively on separate infrastructure, in different fault domains, to provide business continuity in case there is a failure of the primary system. Two modes of Data Guard replication may be configured, physical and logical standby. An important differentiator is that with physical Data Guard, the entire database is replicated with no exception. With logical replication such as SQL Apply or even Oracle GoldenGate, only a subset of the database is replicated—user-specified schemas and objects.

Each database in a Data Guard architecture is designated a role. One database occupies the primary role while one or more databases occupy the standby role. Data Guard physical standby replication may be configured in one of three modes that determines how the redo stream of changes from the primary database is shipped and applied on the standby:

•   **Maximum Performance**   (optimized for performance) Replication is asynchronous and the primary database does not wait for confirmation that the captured redo stream has been successfully shipped or applied on the standby. Replication interruptions or issues do not impact the primary database.

•   **Maximum Availability**   (optimized for availability and protection) Under normal operation, changes are not committed on the primary database until all changes are written to the redo logs and confirmation that the captured redo stream has been successfully shipped or applied on the standby has been received. Issues with the replication cause the system to downgrade automatically to operate as if in maximum performance mode, and the availability of the primary database is not impacted.

•   **Maximum Protection**   (optimized for data protection and integrity of the standby) Zero data loss is possible. Replication is synchronous and the primary database waits for confirmation that the captured redo stream has been successfully shipped or applied on the standby. Replication interruptions or issues cause the primary database to shut down.

A role transition occurs when one of the standby databases assumes the primary role. The following are two types of role transitions:

•   **Switchover**   A planned role transition often to perform OS patching or hardware maintenance on the primary. No data loss is incurred during a switchover because there is a clean stop of changes in the primary, the standby is in sync, and the roles have transitioned. There is a small window (a few minutes at most, usually less than a minute) of unavailability during a switchover. The “old” standby database is now the primary (open read-write) while the “old” primary acts in a standby role. Managed recovery is paused, while maintenance is performed without interruption to the business. This provides HADR in a planned manner.

•   **Failover**   This is an unplanned role transition and, as the name suggests, a failure has occurred. One of the standby databases is designated as the new primary. Point-in-time recovery (PITR) occurs. Depending on the Data Guard configuration discussed earlier, minimal (max performance or max availability) or no data loss (max protection) may occur during failover. Max protection mode ensures that any last in-flight transactions are only committed on the primary if they are committed on the standby, so no data loss occurs.

Ideally, each database system resides in a separate availability domain in a multi-AD region, but at the very least, each system should reside in a separate fault domain.

In a two-system Data Guard configuration, one is designated the primary role while the other takes on the standby role. Applications and users connect to the primary database and as changes are made on the primary database, they are captured and shipped over the network to the standby database, where they are applied. This is a classical physical standby configuration.

Several important considerations for Data Guard systems include setting up a regional subnet in multi-AD regions. This simplifies networking configuration, and primary and standby database systems can be placed in different ADs. Data Guard also provides a monitoring agent known as an observer that ideally runs on different infrastructure from both the primary and secondary DB systems. The observer can be configured to determine when a failover is required and to automatically perform the failover. This provides a huge HADR advantage as no human involvement is required.

Data Guard may be enhanced by licensing the additional Active Data Guard (ADG) option. ADG is included with the Enterprise Edition—Extreme Performance on OCI. ADG allows standby databases to be opened in read-only mode, supporting the offloading of expensive queries and read-intensive activities such as exports and backups to the standby. ADG in 19c and later supports limited read-write operations on the standby database, which exposes some exciting HADR possibilities.



**NOTE**    Consider the us-ashburn-1 region, which has at least three ADs. A best-practice HADR architecture may include a primary RAC database system in AD1 with each RAC node in a separate fault domain, with a standby RAC database system in AD2, again with both standby nodes in separate fault domains. Active Data Guard has been implemented for backup and query offloading. A Data Guard observer has been configured in AD3 to orchestrate a fast-start-fail-over (FSFO) if issues with the primary RAC database system are detected.

**GoldenGate**

Oracle GoldenGate provides logical replication between multiple databases using many different topologies. Each database may reside in separate ADs or even in separate regions if the network bandwidth is sufficient for the DB IOPS. These databases are kept in sync by transactional replication and potentially support updates from multiple master databases. The following GoldenGate topologies are supported on OCI:

•   **Unidirectional**   Source database transactions are sent to one target database. Typically used for offloading read-intensive operations such as Reporting, while providing a read-only synchronized standby database.

•   **Broadcast**   This is similar to Unidirectional but source database transactions are sent to multiple target databases, typically to distribute data to downstream systems.

•   **Bi-directional**   Source database transactions are sent to one target databases and vice versa, using predefined conflict resolution strategies to avoid data conflicts. This topology provides instant failover as both databases are active processing read and write transactions.

•   **Peer-to-Peer**   The same as Bi-directional with multiple master databases. So DB1 replicates to DB2, which replicates to DB3, and so on. This topology also provides excellent zero downtime HADR.

•   **Consolidation**   This is conceptually the opposite of Broadcast because transactions from many source databases are replicated to a single target database. It is typically used in a datamart or datawarehouse scenario.

•   **Cascading**   This conceptually consists of Unidirectional and Broadcast topologies.

Each topology has pros and cons and may be leveraged to design an appropriate HADR solution. Conflict-resolution strategies are usually designed when implementing GoldenGate. This solution can provide zero downtime for RPO/RPO requirements but comes at a financial and complexity cost.

**Autonomous Databases**

Oracle Autonomous Database (ADB) systems offer a hosted and managed option with an underlying Exadata service and the ability to dynamically scale up and scale down both the CPUs and storage allocated to your VM. This supports sizing your environment for average workload, scaling up during peak periods and scaling back down once the workload normalizes, thereby providing database-level performance-based HADR. Autonomous databases are opened by multiple nodes (RAC) when more than 16 OCPUs are enabled and are hosted on highly redundant Exadata infrastructure managed and monitored by Oracle.



**NOTE**    Infrastructure-as-Code solutions, including Terraform and OCI Resource Manager, may also be leveraged in an HADR solution to rapidly provision infrastructure in a secondary region. Budgetary constraints and RPO/RTO requirements may determine if you set up a warm standby that is ready to go with minor CPU scaling adjustments or a cold standby that takes a longer time to provision.

**Leverage OCI Security Features to Protect Your Cloud Infrastructure**

Practically securing your OCI tenancy demands the consolidation of lessons on IAM, networking, compute instances, storage, and databases discussed in detail in previous chapters dedicated to each of these topics. The following is a set of high-level guidelines to consider when securing your tenancy.

**IAM**

OCI resources may be organized into compartments to logically separate applications and systems. IAM supports a flexible framework for granting the least-privilege required to secure access and govern OCI resources. IAM policies allow only authorized groups of IAM users and dynamic groups of principal instances to interact with OCI resources and are an effective mechanism for managing IAM.

Try to ensure that all resources are tagged with defined tags, enabling cost-tracking tags to implement a chargeback system. While your organization may not be ready for a chargeback system today, this approach also helps reduce unnecessary spend and allocates an owner to each resource. Each resource owner is the person or department that pays for a resource. This paradigm may encourage security-compliant behaviors that will contribute to the overall security of your tenancy and reduce the sprawl of resources commonly found in less well-governed environments.

There are many keys involved when accessing OCI resources. Consider rotating SSH keys periodically and allocating individual SSH or API keys to specific individuals who must have access to OCI resources. Corporate key stores should also be leveraged to hold master keys, TDE keys, and other key pairs to reduce resource access issues and to boost security posture. Authentication mechanisms granting users access to the OCI console, API access using API keys, and object storage access using Auth tokens must be formalized for IAM user management.

Recent collaboration with other cloud vendors, notably Azure, supports the federation of IAM systems across multi-cloud environments. While this is great for flexible multi-cloud architectures, you now have an additional level of IAM to carefully consider when designing your security posture. As discussed in [Chapter 3](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch3.xhtml#ch3), IAM federation is not limited to IDCS and can include AD, G Suite, and other SAML 2–compliant directory services. Federating identity providers from the start of your cloud journey is well advised.

**Networking**

When designing security for your VCN, many OCI networking components are available out of the box to support good practice, starting with private and public subnets. It is usually permissible to expose a public load balancer to the Internet for HTTP traffic to your public website or web applications. These points of entry need to be secured and protected. Sensitive databases should be located in private subnets. NAT gateways may be used for instances in private networks to gain one-way access to Internet resources and still be protected. Use the route table infrastructure to ensure that Dynamic Routing Gateways only route allowed traffic between your VCN and your on-premises network or other cloud networks.

Each subnet can have one route table comprising rules that route traffic to one or more targets. It is good practice for hosts that have similar routing requirements to use the same route tables across multiple ADs. It is also recommended that private subnets have individual route tables to control traffic flow. Traffic between all compute instances within a VCN is routable, but the VCN route table limits routing of traffic into and out of the VCN.

Each subnet may have multiple security lists, each of which supports multiple stateful and stateless ingress and egress rules. You must ensure that security lists behave like firewalls, managing traffic into and out of the VCN (known as North–South traffic) as well as managing internal VCN traffic between multiple subnets (known as East–West traffic).

If access to compute resources is required from the public Internet, it is good practice to create a Bastion host. A Bastion host, or server, is colloquially known as a jump-box designed and configured to withstand attacks.

**Compute Instances**

Security in OCI is a shared responsibility between you and Oracle. Using DBaaS and compute instances based on Oracle-supplied images is a safer bet than using custom images and manual software and database installations because the default images are already security hardened by the OCI security specialists at Oracle. It is highly recommended to start your customized compute and database system using a supplied image. Bare metal instances are available with no Oracle-managed software. These instances do not benefit from the security hardening and must be hardened by your internal security team.

**Storage and Databases**

Data is your biggest asset and must be protected. Place sensitive data in encrypted databases, which are encrypted at rest and in-transit. Set up auditing to track who accesses and updates data in both your file systems and databases. When backups are made, ensure these are encrypted, keeping encryption keys in corporate key vaults.

Formulate and formalize a data strategy for data stored in your tenancy that explicitly encourages the use of best practices when using local file systems on block volumes, object storage buckets, shared file storage, and databases to eliminate the risk of errant files with sensitive data being exposed.

**Chapter Review**

This chapter discussed high availability and disaster recovery architectural considerations as well as a consolidated high-level outlook on designing a future cloud security posture by embracing best practices and leveraging lessons from your own organizational security legacy.

There are many permutations of OCI resources that will address any HADR requirement. In reality, we are governed by three main considerations: budget, RPO, and RTO. Understanding that the HADR design exercise is constrained by these three considerations is likely to yield an optimal architecture. Not all systems require a zero downtime, zero data loss, sufficiently engineered solution. But some do.

Cloud is the future and cloud vendors continue to innovate and compete for your business. HADR and security are key areas to design correctly. But it is not over when you build it. These topics are highly visible, and it is highly recommended that you attend to the latest developments and innovations available to evolve your design to ensure a safe and highly available infrastructure that is resilient to disaster.

**Questions**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans1)**.**   RTO is an important concept related to high availability and disaster recovery. What sentiment is associated with RTO?

**A.**   RTO specifies the amount of data loss that is tolerable for a system without impacting the business too negatively.

**B.**   RTO specifies the amount of time it takes to restore service for a system without impacting the business too negatively.

**C.**   Oracle redo transactions are shipped to the Data Guard standby database.

**D.**   Real-time Oracle is a logical replication solution that mirrors an existing system.

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans2)**.**   RPO is an important concept related to high availability and disaster recovery. What sentiment is associated with RPO?

**A.**   RPO specifies the amount of data loss that is tolerable for a system without impacting the business too negatively.

**B.**   RPO specifies the amount of time it takes to restore service for a system without impacting the business too negatively.

**C.**   The Oracle redo process generates redo logs essential to instance recovery.

**D.**   The Oracle replication process is a solution that clones an existing system.

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans3)**.**   Which type of IP address may be unassigned and reassigned between compute instances?

**A.**   Public IP

**B.**   Private IP

**C.**   IPv6 IP

**D.**   Floating IP

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans4)**.**   DenseIO compute shapes include support for direct attached NVMe disks. What steps, if any, are required to ensure redundancy for this type of storage?

**A.**   This storage is mirrored at the SAN level. No further steps are required.

**B.**   Object storage mirrors must be configured.

**C.**   Direct attached NVMe disks are preconfigured as highly available storage.

**D.**   Some RAID configuration must be implemented to support redundancy for generic file system storage.

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans5)**.**   Which of the following statements is true?

**A.**   All OCI regions have three availability domains.

**B.**   HADR is not possible in a region with only one AD.

**C.**   Each AD has three fault domains.

**D.**   All OCI regions have three fault domains.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans6)**.**   What process coordinates a fast-start-fail-over event in a Data Guard setup that automates a primary database failover to its standby?

**A.**   Observer

**B.**   Watcher

**C.**   Listener

**D.**   Active Data Guard

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans7)**.**   Which of the following statements is true?

**A.**   You can only interact with OCI resources using the CLI and Terraform.

**B.**   You can only interact with OCI resources using the CLI, Terraform and SDKs.

**C.**   You can only interact with OCI resources using the CLI, Terraform, SDKs, and the OCI console.

**D.**   You can interact with OCI using any tool through the OCI API endpoints.

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans8)**.**   Which of the following statements is true? (Choose all that apply.)

**A.**   A RAC database is concurrently mounted by one or more database instances, each running on a separate compute node.

**B.**   RAC databases can tolerate the loss of a RAC node.

**C.**   As long as there is at least one RAC node available, the database remains accessible.

**D.**   Both primary and standby databases in a Data Guard configuration may be RAC databases.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans9)**.**   Which of these options may provide zero data loss solutions for Oracle databases?

**A.**   Oracle RAC

**B.**   Oracle Data Guard in Maximum Performance mode

**C.**   Oracle Data Guard in Maximum Protection mode

**D.**   Oracle Data Guard in Maximum Availability mode

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#ch8ans10)**.**   List two basic approaches to synchronizing general purpose file systems for HADR.

**A.**   Private peering

**B.**   Synchronous replication

**C.**   File Storage Service snapshots

**D.**   Asynchronous replication

**Answers**

[**1**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans1)**.**   **B.** RTO specifies the amount of time it takes to restore service for a system without impacting the business too negatively.

[**2**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans2)**.**   **A.** RPO specifies the amount of data loss that is tolerable for a system without impacting the business too negatively.

[**3**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans3)**.**   **D.** Floating IP addresses may be unassigned from one compute instance and reassigned to another. It is often possible to automate the allocation of a floating IP address to a standby instance to minimize downtime.

[**4**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans4)**.**   **D.** DenseIO compute shapes include support for direct attached NVMe disks. This storage is not SAN-based. There is no redundancy built in and it is your responsibility to set up appropriate redundancy using some sort of RAID configuration if they are used for generic file system storage.

[**5**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans5)**.**   **C.** Each AD has three fault domains providing physical server isolation for VMs created in separate FDs in the same AD.

[**6**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans6)**.**   **A.** A Data Guard observer is configured to orchestrate a fast-start-fail-over (FSFO) if issues with the primary database system are detected.

[**7**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans7)**.**   **D.** You can interact with OCI using any tool through the OCI API endpoints. You are not confined to using any specific tools.

[**8**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans8)**.**   **A, B, C, D.** RAC and Data Guard form a potent pair in providing HADR for Oracle databases.

[**9**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans9)**.**   **C.** Data Guard in Maximum Protection mode ensures synchronous replication achieving zero data loss at the cost of potential waits on the primary database for confirmation that the captured redo stream has been successfully shipped and applied on the standby.

[**10**](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781260452600/ch8.xhtml#r_ch8ans10)**.**   **A, D.** Synchronous replication supports zero data loss but incurs waits while transported IOs are acknowledged and remote IOs are confirmed. Asynchronous replication risks data loss in the event of an outage, but changes to the primary site are shipped to the standby with no need to wait for confirmation that the changes have been received and applied. There is no blocking or waiting with this approach and it is suitable for synchronizing data between instances in different regions.