***Chapter 1*: Introduction to Oracle Cloud Infrastructure**

**Oracle Cloud Infrastructure** (**OCI**) is an **Infrastructure-as-a-Service** (**IaaS**) cloud platform that allows consumers to create resources, such as compute instances, databases, networks, containers, functions, and storage, in order to run their applications and workloads. A number of different parties can interact with the OCI cloud. Some of these are actual users, while others are external systems that OCI services communicate with.

So, what do you need from the cloud? Well, an enterprise always looks for scalable, available, and on-demand solutions when they want to move their workload to the cloud. However, for critical enterprise applications, you need no-compromise security and performance guarantees. Remember, you want to offer the same, or better, **Service-Level Agreements** (**SLAs**) for your business.

In this chapter, we will cover the following topics:

* Regions and **Availability Domains** (**ADs**)
* Off-box virtualization
* Fault domains

**Regions and ADs**

**OCI** has been built using the concept of **regions**. A region is simply a physical location in the world where OCI hosts data centers. In a nutshell, a region is a localized geographic area. Within a region, OCI hosts one or more physical data centers and calls this an **Availability Domain** (**AD**).

In this section, we will look at the main concepts of OCI in more detail, such as regions, ADs, and fault domains. Additionally, we will learn how to subscribe to other regions.

A lot of OCI services are regional; for example, **Virtual Cloud Networks** (**VCNs**). If you create a VCN, it will span across the AD. Other services are AD-specific, such as compute resources. You can create a compute instance that has access to a specific AD. Additionally, there is a very strong interconnectivity between the ADs within a region and across regions. Within an AD, interconnected traffic is encrypted as well.

As of August 2020, there are 26 regions and 6 interconnected Azure regions that are live. The following map shows the different regions that are currently live across the globe:



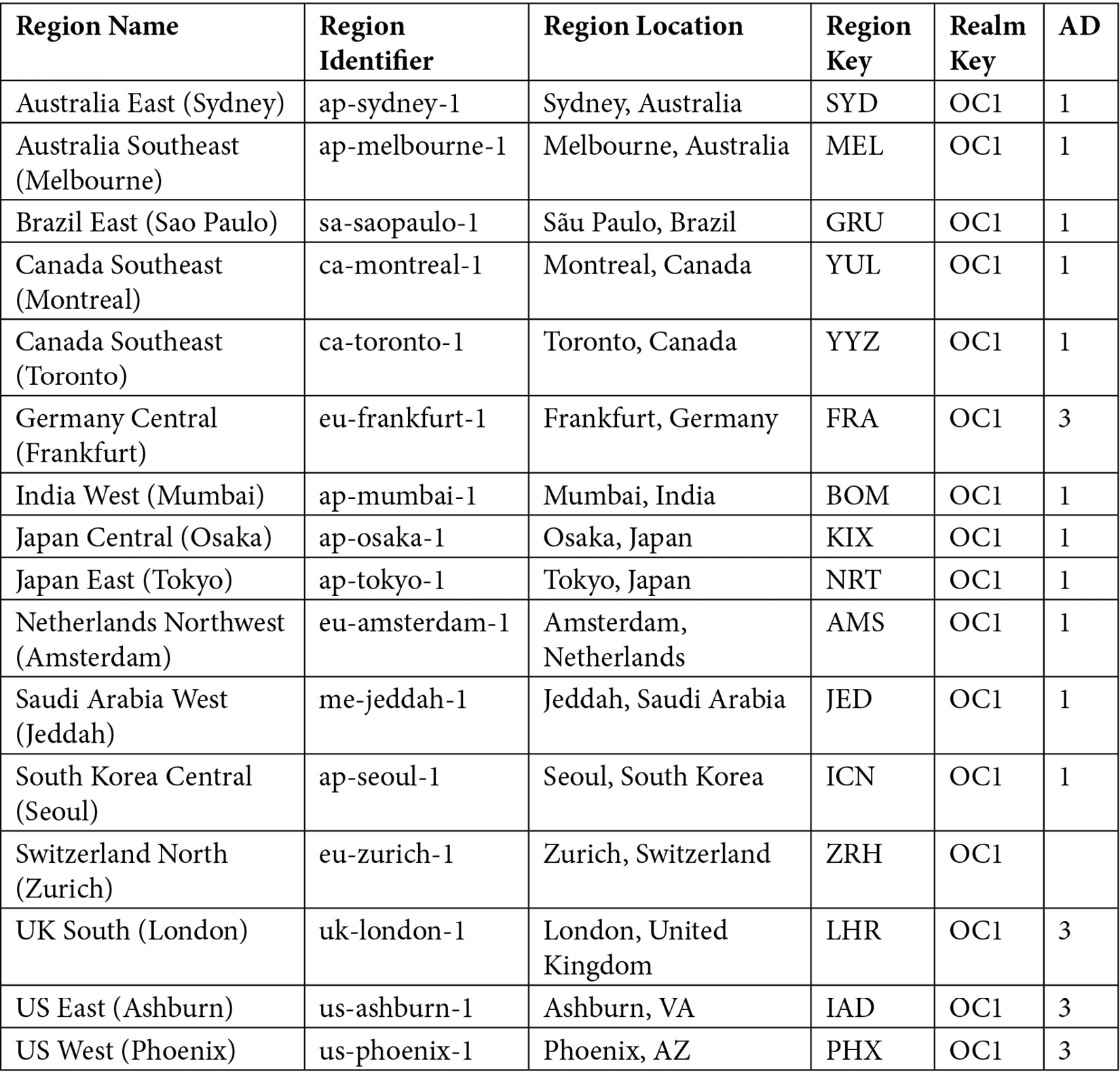
Figure 1.1 – OCI regions

Oracle's strategy is to add new regions around the world in order to give customers local access to its cloud resources. To speed up the process and still provide high availability, OCI has launched one AD region that has three fault domains inside the physical AD. We will discuss fault domains in more detail later in this chapter.

OCI regions are dispersed via vast distances across countries and even continents. When a customer deploys their application, they typically put that application in the region where it is most heavily used. However, there are multiple reasons why someone might choose to put their applications in different regions, such as the following:

* A natural calamity could affect a whole country or continent.
* Data jurisdiction drives data locality requirements.

The following table identifies a region, its identifiers, location, region key, realm key, and the number of ADs within it:



*IMPORTANT NOTE*

*The data in the table is accurate as of August 2020. However, it may not remain accurate as Oracle is rapidly adding new regions and interconnected Azure regions. You can refer to Oracle's public documentation to find the latest information on the available regions at*[*https://docs.cloud.oracle.com/en-us/iaas/Content/General/Concepts/regions.htm?*](https://docs.cloud.oracle.com/en-us/iaas/Content/General/Concepts/regions.htm?)*.*

**Managing regions from the OCI console**

You can subscribe to any of the commercially available regions from the OCI console. However, doing so requires administrative privileges.

During the sign-up process, OCI will create a **tenancy** and assign a region to you. This is called the **home region**. You cannot change this home region later. However, you can subscribe to other available regions. By doing so, OCI will replicate your identity resources to the new region.

To view the subscribed regions, follow these steps:

1. Log in to the OCI console.
2. Open the **Regions** menu.
3. View the subscribed region(s). Please note that all the region names that appear in the **Regions** menu are the regions that you are subscribed to already.

To subscribe to other regions, perform these steps:

1. Log in to the OCI console.
2. Open the **Regions** menu, and then click on **Manage Regions**, as highlighted in the following screenshot:

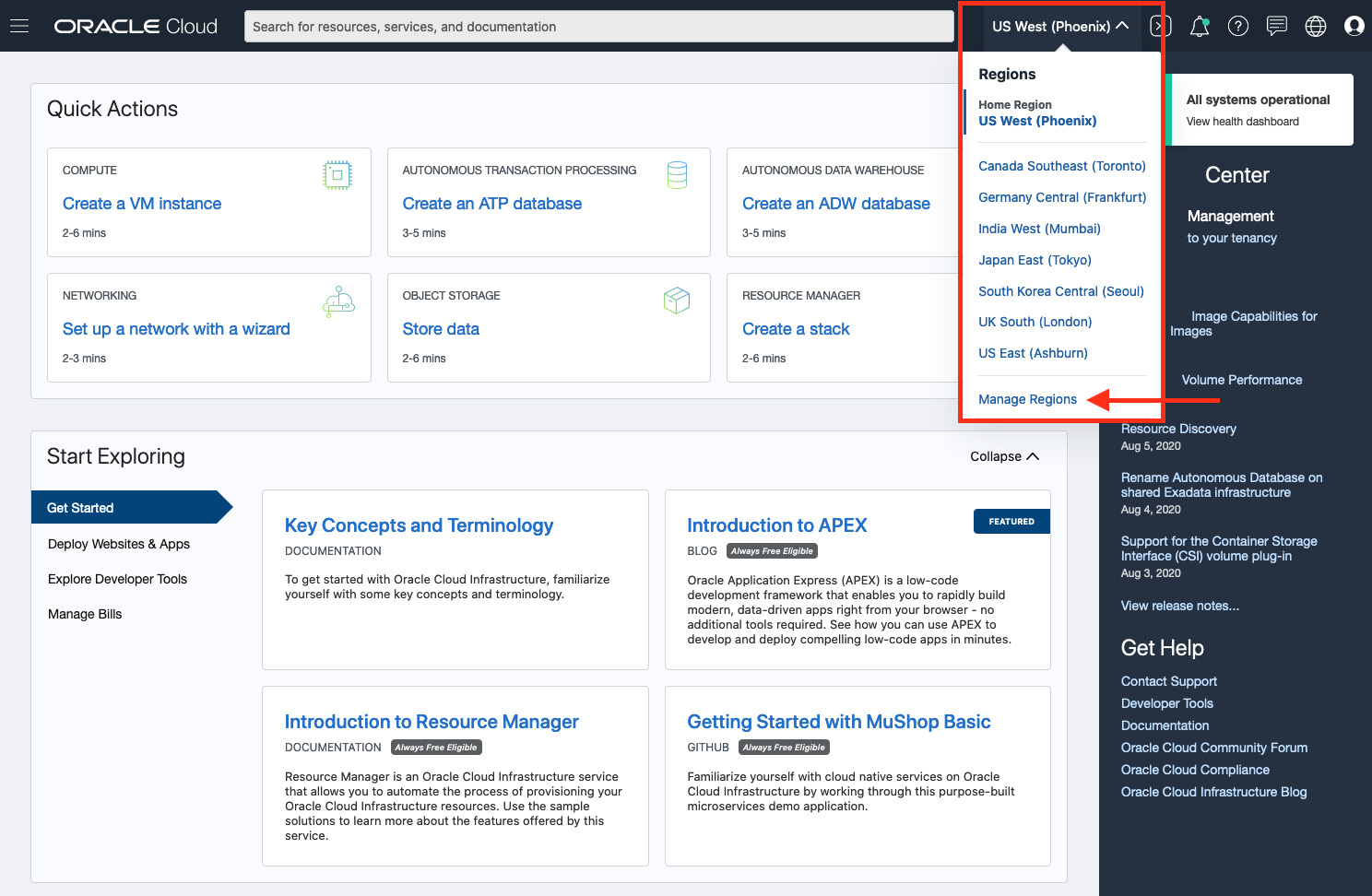


Figure 1.2 – The list of subscribed regions

1. Check which region you want to subscribe to, and then click on **Subscribe**, as shown in the following screenshot:

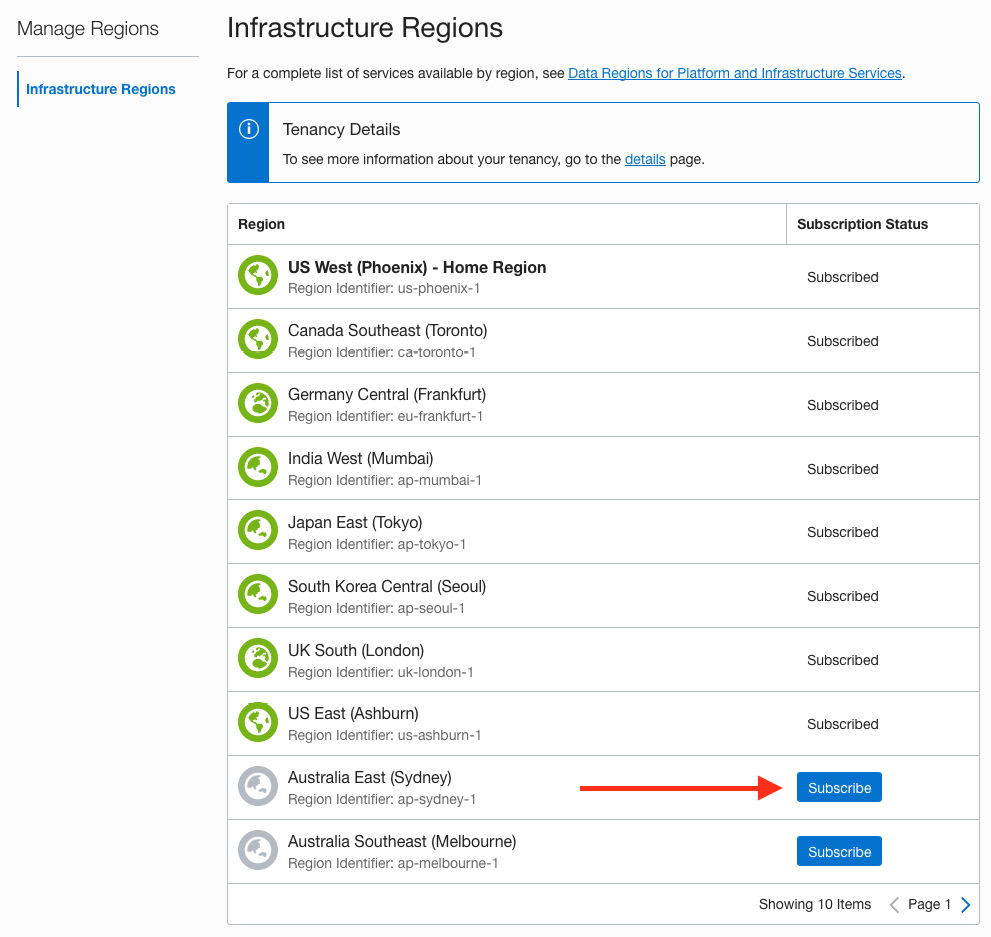


Figure 1.3 – The Infrastructure Regions subscription page

So, you can see how simple it is to subscribe to regions where you want to consume cloud resources.

**Logical view of Oracle Cloud Infrastructure components**

OCI regions are part of a realm. A **realm** is a logical collection of regions. Realms do not share any data. While regions within a realm can share data via replication, regions in separate realms are completely isolated from each other.

**Tenancies**

OCI users live in a tenancy, which is a logical grouping for a business customer that contains users, groups, and compartments. A tenancy is based in a home region but can be subscribed to other regions. When a tenancy is subscribed to another region, tenancy data created in the home region is automatically replicated to the subscribed region. Replication of this data is required to call services in that region. Identity data can only be modified in the tenant's home region.

**Bootstrapping**

A tenancy is created when the accounts service receives a request to create an account entitlement. The account service coordinates with the **Identity and Access Management** (**IAM**) service to generate several resources, such as the tenancy (root compartment), a default access policy, a user account, and an administrators group to which the user is added.

When the user account is created, a **one-time password** (**OTP**) is generated. With this password, the user can sign in to the web console and upload the public part of the key pair they generated. Once this is done, the user can start making signed calls to the OCI APIs using **command-line interface** (**CLI**) tools.

**Compartments**

Compartments are the logical containers of resources. Compartments typically have a policy attached to them to control access to the resources inside. Compartments can be nested as well. They can span regions, which makes it possible to add, for example, compute instances from different regions to the same compartment and have them guarded by the same policy.

**Oracle Cloud Identifiers (OCIDs)**

Resources in OCI are identified by unique **Oracle Cloud Identifiers** (**OCIDs**). An OCID consists of different parts separated by dots:

ocid1.<resource type>.<realm>.[region][.future use].<unique ID>

Let's take a look at the various parts that make up the OCID:

* **ocid1**: This indicates the version of the OCID.
* **resource type**: This is the type of resource. For example, a resource can be an instance, a volume, a VCN, a tenancy, a user, or a group.
* **realm**: This indicates which realm the resource is in. For example, all production regions use **oc1**.
* **region**: This indicates where this resource is located.
* **future use**: This is reserved for future use; therefore, you are likely to see a blank space here.
* **unique ID**: This section is the unique portion of the ID. You might notice a different format depending on the type of resource or service.

Here are a couple of examples:

ocid1.tenancy.oc1..aaaaaaaaba3pv6wkcr4jqae5f44n2b2m2yt2j6rx32uzr4h25vqstifsfdsq

ocid1.instance.oc1.phx.abuw4ljrlsfiqw6vzzxb43vyypt4pkodawglp3wqxjqofakrwvou52gb6s5a

OCI's regions are physically divided into ADs, which are named by numbering them (for example, **phx-ad-1**, **phx-ad-2**, and **phx-ad-3**). It is a human tendency to pick the first AD from a given list when it comes to creating a compute instance in an AD. In order to stop you from doing this, OCI gives each tenancy a random set of logical ADs. This is called **AD obfuscation**. Logical AD names look like **SQPR:PHX-AD-1**, and physical AD names look like **phx-ad-1**.

AD, which is nothing but physical data centers, are far away enough that they are completely independent, from a failure perspective, but are close enough to have very low-latency connectivity. Customers get to choose what AD they create resources in, such as compute instances, databases, and more. This selection, however, is randomly mapped to a physical AD to prevent the uneven usage of ADs. In the following screenshot, you can see a logical view of the mapping of ADs in a region and how that maps to a fault domain inside it:

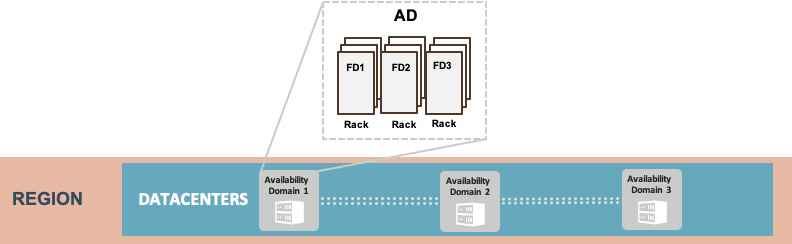


Figure 1.4 – An OCI AD

Inside an AD, OCI runs a highly scalable and high-performance network, which is not oversubscribed. Due to this design of non-oversubscription, there is no noisy neighbor problem. In terms of scalability, this AD can scale up to approximately 1 million ports. Additionally, because of the no noisy neighbors and non-oversubscription network, this AD has predictable low latency and high-speed interconnectivity between hosts that don't traverse more than two hops. In the following logical diagram, you can see a mapping of how the physical network infrastructure connects to the regions:

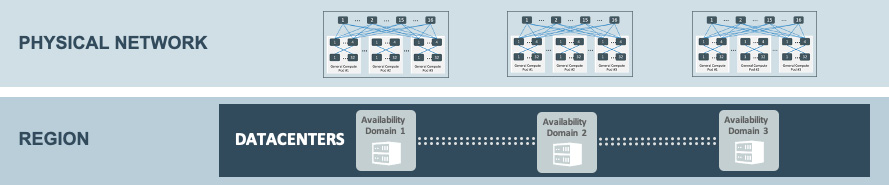


Figure 1.5 – The layout of OCI's physical infrastructure

OCI's first four regions (Phoenix, Ashburn, London, and Frankfurt) consist of three ADs. Each AD is in a physically separate data center.

In these initial regions, Oracle built many foundational services that were specifically tied to a single AD. This is so that there would be no dependencies outside of a single AD. A compute service is the most prominent example of this.

After the first four regions, Oracle shifted their strategy. The majority of customer workloads do not take advantage of ADs for high availability, and, instead, they rely on high availability within an AD and use multiple regions to support disaster recovery. Therefore, OCI adapted to this by launching a larger number of single-AD regions.

**Off-box virtualization**

If you look at any traditional cloud provider, they are all made up of **Virtual Machines** (**VMs**) running on top of a hypervisor. A hypervisor's job is to isolate these VMs by sharing the same CPUs, and then capture I/O from each VM to ensure that they are abstracted from the hardware. The VM is, therefore, secure and portable, as it sees only a software-defined network interface card. The hypervisor can inspect all of the packets between the VMs and enable features such as IP whitelists and access control lists. This is depicted in the following diagram:

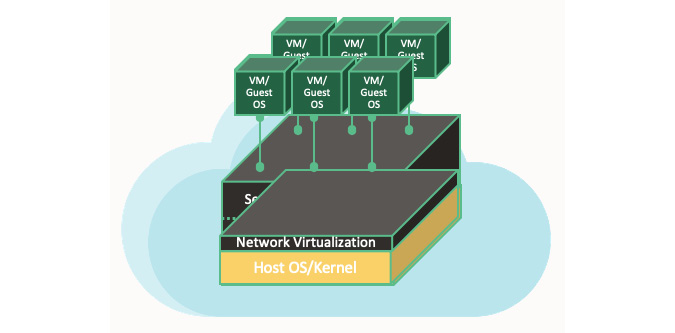


Figure 1.6 – In-kernel network virtualization

In in-kernel virtualization, whenever there is a need to inspect packets to and from a VM, you can put pressure on the host hypervisor by taking away its CPU cycles. This is mainly because this type of hypervisor performs packet switching, encapsulation, and enforces stateful firewall rules. However, this is not the only risk. There is another risk of having noisy neighbors. A noisy neighbor VM monopolizes bandwidth, disk I/O, and CPUs at the expense of its neighbors.

The fundamental purpose of off-box virtualization is that it no longer commits I/O virtualization into the hypervisor but to the network outside of the physical box. You can't reach the control plane that runs the virtual network from the public internet. However, you can create an explicit tunnel to reach the virtual network, which can be monitored, audited, and, in the case of an emergency, switched off as well. This is shown in the following diagram, where you can clearly see that the network I/O virtualization is not being done at the host hypervisor level:

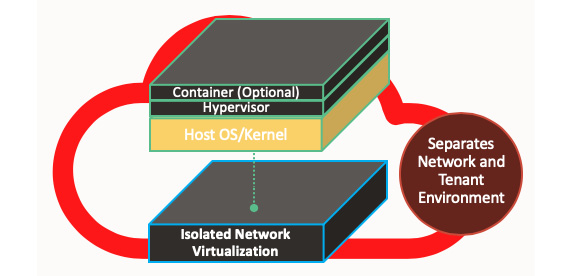


Figure 1.7 – Off-box network virtualization

When you move network virtualization from in-kernel virtualization to off-box virtualization, it results in a dramatic change in performance and improved security posture. This is because you are no longer getting any performance overhead associated with the hypervisor. Another benefit of off-box virtualization is that you retain the flexibility to plug anything into the virtual network. You can perhaps add another bare metal host, an **Non-Volatile Memory Express** (**NVMe**) storage system, a VM, a container, or even an engineered system, such as Exadata. All of them can run on this virtual network and reach each other within two hops. Take a look at the following diagram:

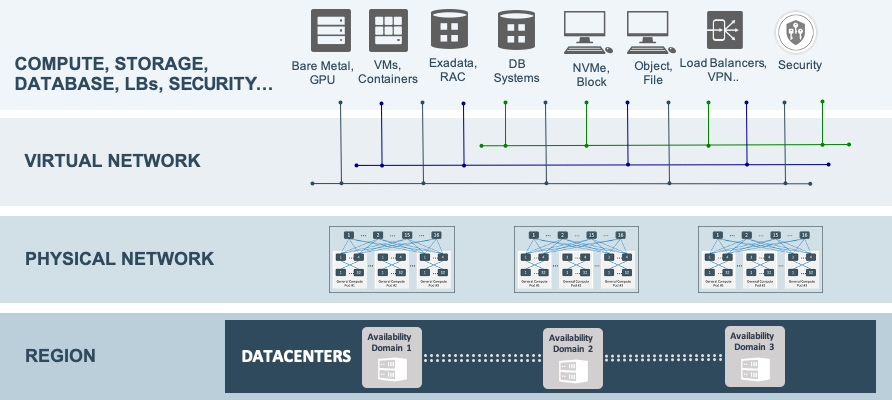


Figure 1.8 – OCI's holistic architecture

OCI's unique offering of bare metal servers doesn't come with a pre-installed operating system or any software. This increases the level of security over traditional virtualization. You can choose any hypervisor that you want to install on top of the OCI bare metal instance and then deploy VMs and install applications on top of that. OCI doesn't offer any access to the memory space of these bare metal instances, allowing complete physical isolation.

As you can see, these bare metal instances are running without any adjacent co-tenants; therefore, they boost both IOPS and bandwidth.

**The security benefits of off-box virtualization**

Traditional server virtualization comes with an abstraction layer. This layer abstracts the application that is running on the virtualized server from compute resources, storage, and networking hardware. You can deploy this virtual infrastructure without any disruption as it has nothing to do with user experience. You have to virtualize the CPU, main memory, network access, and I/O if you want to take advantage of partitioning, isolation, and hardware independence, which all result from the virtualization process.

Traditional server virtualization in first-generation public cloud infrastructures might come at a cost to the performance overhead and lead to weaker security. The performance overhead is mainly because the hypervisor needs to manage network traffic and I/O for all of the VMs that are running on a host, causing noisy neighbor problems. Additionally, the level of security is inherently weaker because, in traditional virtualization architectures, the hypervisor has complete trust and makes access decisions on behalf of the VM. This means a hostile actor that compromises the hypervisor can easily spread beyond the single hypervisor to other systems in the same cloud. More specifically, the traditional model implies that the attacker of a compromised host/hypervisor can access *any* VCN because the host/hypervisor is trusted to do so. You can see an example of in-kernel network virtualization in the following diagram:

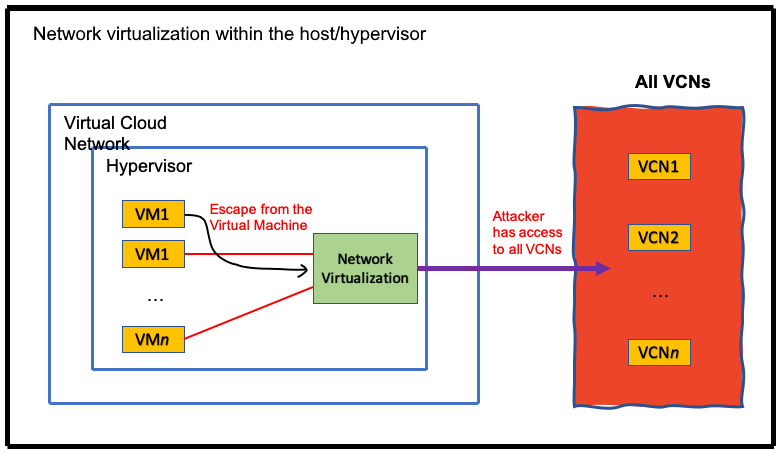


Figure 1.9 – In-kernel virtualization

The security isolation between one customer's compute resources from other customers' compute resources is critical. Fundamentally, OCI designed its security architecture with the assumption that customer-controlled compute resources can be hostile. It has a multi-pronged *defense in-depth* security architecture, which has been designed to minimize the security risks of traditional virtualization.

Oracle's Gen 2 Cloud infrastructure uses a unique approach that eliminates some of the disadvantages of traditional server virtualization. OCI uses off-box network virtualization, which takes network virtualization out of the software stack (hypervisor) and places it in the infrastructure. Oracle uses off-box network virtualization technology in both bare metal instances (for example, physical servers dedicated to a single customer) and VM instances.

Isolated network virtualization limits the attacker surface to *only* a VCN connected to the hypervisor by the control plane. OCI moves the trust from the hypervisor to the isolated network virtualization, which is implemented outside of the hypervisor. In the following diagram, you can see how this is done:

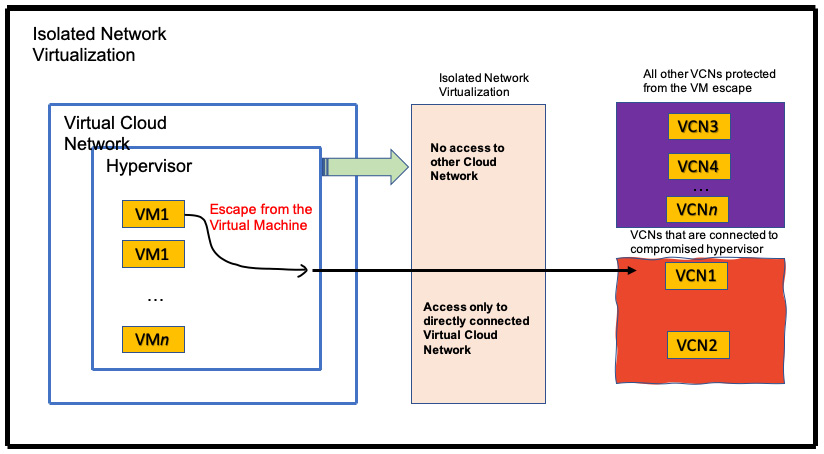


Figure 1.10 – Off-box network virtualization

Bare metal is unique as no hypervisor is needed to run resource virtualization for the network and I/O. Instead, OCI moves network virtualization into the infrastructure, resulting in dramatic performance and security gains. This is because the performance overhead associated with traditional virtualization (in the hypervisor) is eliminated.

OCI effectively uses a coprocessor in the infrastructure, which performs network virtualization, and thus allows the hypervisor to focus on other virtualization tasks. Offloading network virtualization eliminates a number of traditional hypervisor security risks by creating a security boundary between the hypervisor running on one host and the virtual networks of other VMs running on other hosts.

OCI engineered this solution in an attempt to reduce the attack surface and potential damage that an attacker might cause in the case of a compromised hypervisor.

OCI has also implemented additional layers of network isolation, which prevent malicious actors from sending unauthorized network traffic even in the extremely unlikely case of an attacker breaking through the first line of defense that is provided by off-box network virtualization.

For bare metal instances, off-box network virtualization provides a security boundary for the virtual network. This boundary prevents an attacker on a bare metal instance from gaining access to the virtual networks of other bare metal instances and VMs running on other hypervisors.

**Fault domains**

OCI has achieved high availability by distributing resources to regions and ADs. A region is a geographically distributed area where one or more ADs are placed. During the initial process of OCI deployment, Oracle created multiple ADs inside a single region, such as Ashburn, Phoenix, Frankfurt, and the UK. These ADs are simply physically separated data centers in a single region.

To further segregate one single AD into more physically isolated areas, OCI created fault domains. A fault domain is a group of rack hardware that has been physically isolated within an AD. Each AD contains three fault domains. You can further choose which fault domain to put your cloud resources into, creating a high-availability structure even when you have just one AD within a region. Fault domains provide anti-affinity rules for your cloud resources. The physical hardware in a fault domain also has its own power supplies, which are redundant, to provide a further layer of availability. You can view a high-level logical diagram of the physically separated fault domain structure within a single AD here:

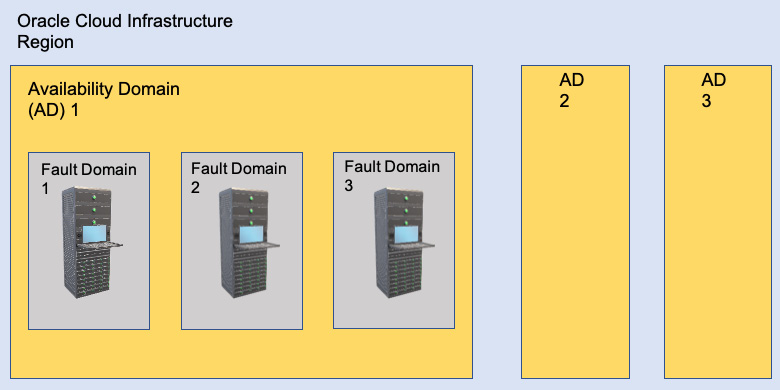


Figure 1.11 – OCI's fault domain

Fault domains are based on the compute racks within an AD. All of the resources that share a rack will also share a fault domain, and resources in different fault domains cannot exist on the same rack. Customers can choose which fault domain they want to create resources in. This selection, similarly to ADs, is randomly mapped to a physical fault domain per tenancy in order to prevent the uneven usage of fault domains.

In [*Chapter 4*](https://learning.oreilly.com/library/view/oracle-cloud-infrastructure/9781800566460/B16798_04_Final_NM_ePub.xhtml#_idTextAnchor083), *Compute Choices on Oracle Cloud Infrastructure*, we will show you how to choose a fault domain while creating an instance to distribute your workload across physical racks.

**Summary**

In this chapter, we have learned about the fundamentals of Oracle's second-generation cloud infrastructure and its building blocks, such as regions, ADs, fault domains, and off-box virtualization. These foundational pillars help Oracle to uniquely distinguish itself from other cloud providers in the market.

In the next chapter, we will go through the first and most important block of OCI's foundation, which is IAM. We will examine how OCI creates a logical separation of its resources using compartments, and we will learn how to assign roles and accesses using policy definitions.