ORACLE

WebLogic For OKE
An Architecture Example in OCI

Solution Definition and Design

Version 0.2 30 June 2023

Contents

Document Control

1.1 Version Control

Version	Authors	Date	Comments
0.1	Cristiano Ghirardi	29 June 2023	Document creation
0.2	Cristiano Ghirardi	30 June 2023	Review

1.2 Common Acronyms

Acronym	Meaning
IaaS	Infrastructure as a Service
DRG	Dynamic Routing Gateway
OCI	Oracle Cloud Infrastructure
LBaaS	Load Balancer as a Service
LB	Load Balancer
NSG	Network Security Group
OCIR	Oracle Container Registry
VCN	Virtual Cloud Network
AD	Availability Domain
FD	Fault Domain
ExaCS	Exadata Cloud Service
App	Application
WAF	Web Application Firewall
VM	Virtual Machine

1.3 Document Purpose

This document provides a high-level solution definition for the Oracle solution and aims at describing the current state and to-be state as well as an example of physical implementable solution.

We will, specifically, consider an OCI architecture whose purpose is to deploy a WebLogic workload in OCI using the OCI For OKE Marketplace packaged solution. To introduce the workload and describe the solution we will consider a fictional *ACME* company which needs to move one of its most complex WebLogic workloads to OCI.

Workload Description

ACME is a transport company based in Europe which manages land and maritime transportation services. The company has developed internally a number of J2EE applications and microservices which purpose is to manage tickets, travellers and to collect IOT data from Bus. The application is currently deployed on WebLogic 12.1.0.4 which is hosted in on-premises Linux bare-metal hosts. *ACME* is looking to consolidate all the applications, WebLogic, and non-WebLogic (microservices), into the same Oracle Kubernetes Engine (OKE) cluster.

Workload Requirements and Architecture

3.1 Overview

We will build the architecture with the goal of consolidating all WebLogic and microservices applications into the same highly available Kubernetes cluster. Oracle Container Engine for Kubernetes (OKE) is the most suitable platform to accomplish that goal moving to OCI, in fact, OKE allows to:

- Migrate WebLogic applications into OCI in straightforward way
- Migrate your WebLogic licenses to OCI (some requirements must be met)
- Create the OKE WebLogic cluster with the Marketplace Terraform stack
- Use worker pools in the same cluster or in separate clusters to host different non-WebLogic applications (for example, microservices)
- Integrate with and automatically deploys a Jenkins platform (if created by the Marketplace Stack) which allows to setup CI/CD pipelines not only for WebLogic but also for all kind of applications that support a CI/CD development model
- Migrate straightforwardly WebLogic applications from on-premises to OCI using the WebLogic tooling and features (see the Livelabs)
- Directly migrate microservices containers to OKE

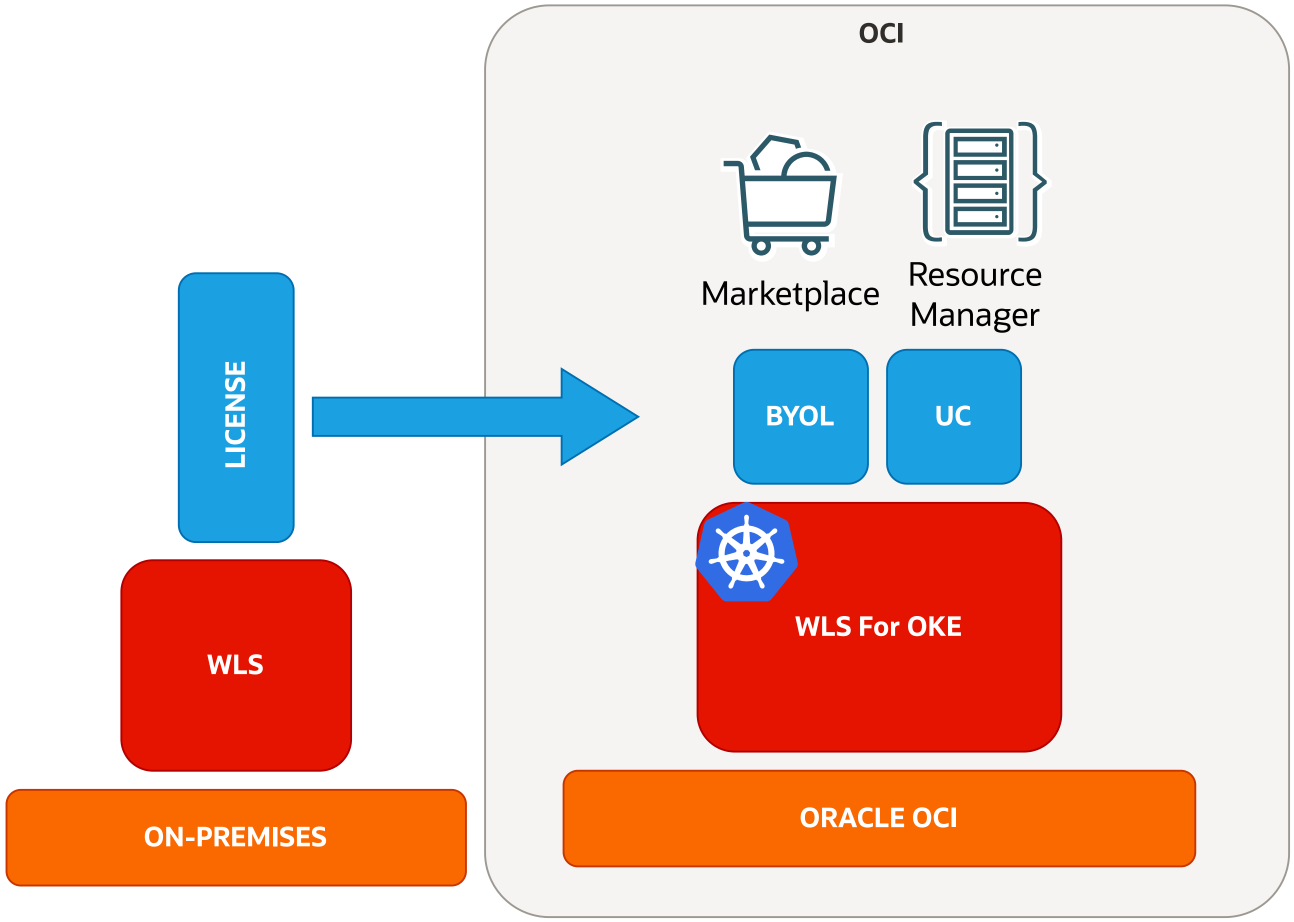

Figure 1 WebLogic for OKE - Marketplace

3.2 Future State Architecture

3.2.1 Logical Architecture

The logical architecture of our fictional *ACME* company is quite simple and, as for the physical one, we will focus mainly on OKE and WebLogic deployment inside it. In this logical schema we just present the main resources and integrations, the important aspects to consider are:

- Some applications will be accessed from Internet or others from *ACME* Intranet.
- *ACME* will have admin access from the Intranet on separate, privileged channels (VPN and/or OCI Fastconnect).
- Oracle Container Registry (OCIR) will be used for both WebLogic images and *ACME* Microservices images.
- Kubernetes Persistent Volumes will use Oracle File System Storage (FSS).
- Data layer will be an OCI DB Base System. DB Base Systems leverage the Oracle Database Multitenant frameworks so multiple datastores can be placed in the same DB System, moreover, it supports High Availability with a possible two-nodes RAC deployment and Disaster Recovery with an associated *Dataguard* DB System.

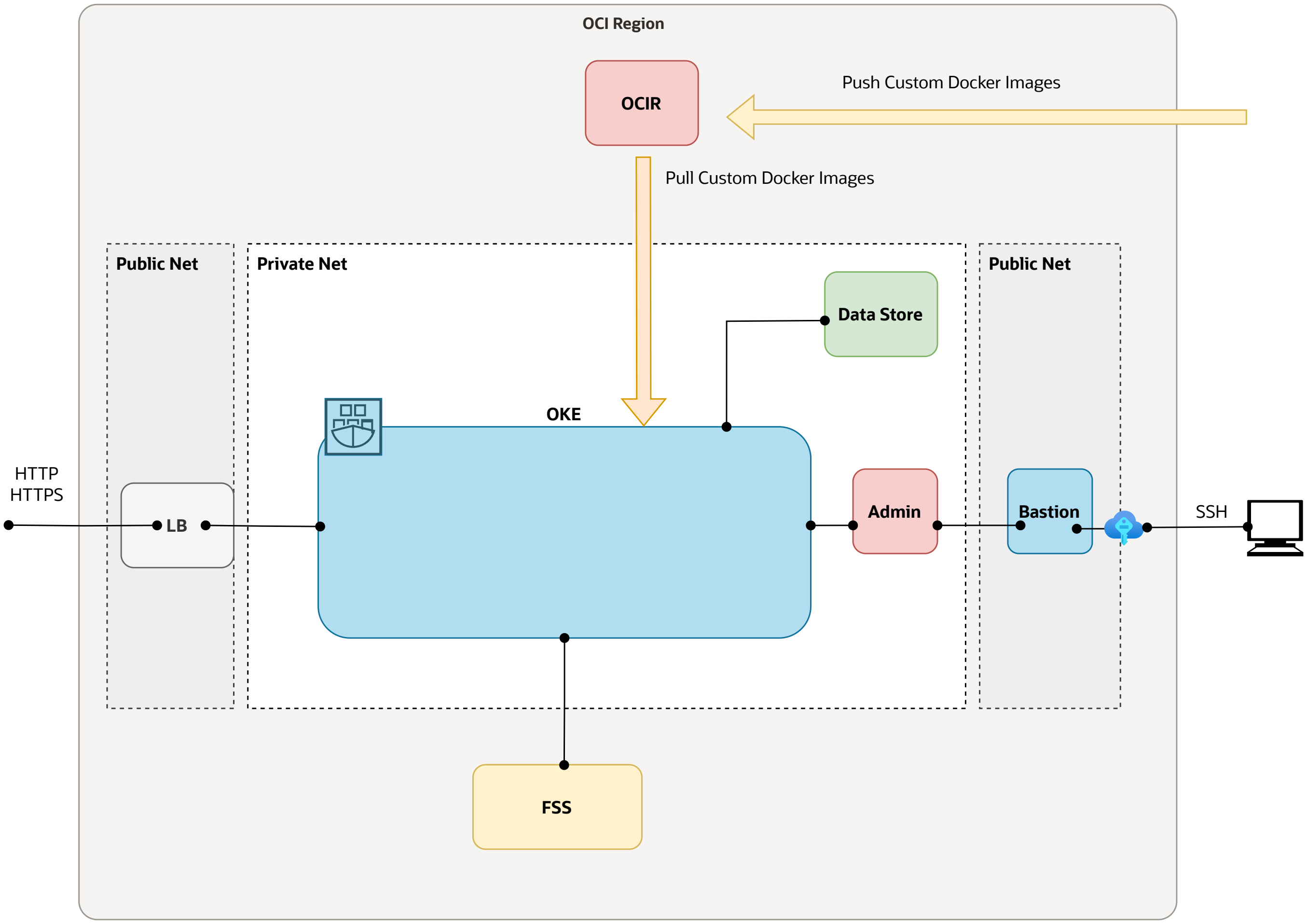

Figure 2 ACME Logical Architecture

In the next chapters we will consider very high-level aspects to be evaluated during the *ACME* migration to OCI. The goal is to present an example of all the topics to consider during the deployment and to give a sort of workflow to follow during the project. We don't expect that this example will be detailed enough during a real project, but it should be helpful as a set of guidelines.

3.2.2 OCI Cloud Landing Zone Architecture

The design considerations for an OCI Cloud Landing Zone have to do with OCI and industry architecture best practices, along *ACME* specific architecture requirements that reflect the Cloud Strategy (hybrid, multi-cloud, etc.). An OCI Cloud Landing zone involves a variety of fundamental aspects that have a broad level of sophistication. A good summary of a Cloud Landing Zone has been published in the OCI User Guide. For this architecture example we assume the Landing Zone will be deployed following OCI best practices.

3.2.2.1 Naming Convention

A naming convention is an important part of any deployment to ensure consistency as well as security within your tenancy. Hence, we jointly agree on a naming convention, matching Oracle's best practices and ACME requirements.

Please find the agreed naming convention in the chapter Resource Naming Convention.

3.2.2.2 Security and Identity Management

This chapter covers the Security and Identity Management definitions and resources which will be implemented for ACME.

3.2.2.2.1 Universal Security and Identity and Access Management Principles

- Groups will be configured at the tenancy level and access will be governed by policies configured in OCI.
- Any new project deployment in OCI will start with the creation of a new compartment. Compartments follow a hierarchy, and the compartment structure will be decided as per the application requirements.
- It is also proposed to keep any shared resources, such as Object Storage, Networks etc. in a shared services compartment. This will allow the various resources in different compartments to access and use the resources deployed in the shared services compartment and user access can be controlled by policies related to specific resource types and user roles.
- Policies will be configured in OCI to maintain the level of access / control that should exist between resources in different compartments. These will also control user access to the various resources deployed in the tenancy.
- The tenancy will include a pre-provisioned Identity Cloud Service (IDCS) instance (the primary IDCS instance) or, where applicable, the Default Identity Domain. Both provide access management across all Oracle cloud services for IaaS, PaaS and SaaS cloud offerings.
- The primary IDCS or the Default Identity Domain will be used as the access management system for all users administrating (OCI Administrators) the OCI tenant.

3.2.2.2.2 Authentication and Authorization for OCI

Provisioning of respective OCI administration users will be handled by *ACME*.

3.2.2.2.2.1 User Management

Only OCI Administrators are granted access to the OCI Infrastructure. As a good practice, these users are managed within the pre-provisioned and pre-integrated Oracle Identity Cloud Service (primary IDCS) or, where applicable, the OCI Default Identity Domain, of OCI tenancy. These users are members of groups. IDCS Groups can be mapped to OCI groups while Identity Domains groups do not require any mapping. Each mapped group membership will be considered during login.

Local Users

The usage of OCI Local Users is not recommended for most users and is restricted to a few users only. These users include the initial OCI Administrator created during the tenancy setup, and additional emergency administrators.

Local Users are considered as Emergency Administrators and should not be used for daily administration activities!

No additional users are to be, nor should be, configured as local users.

ACME is responsible to manage and maintain local users for emergency use cases.

Federated Users

Unlike Local Users, Federated Users are managed in the Federated or Enterprise User Management system. In the OCI User list Federated Users may be distinguished by a prefix which consists of the name of the federated service in lower case, a '/' character followed by the username of the federated user, for example:

```
oracleidentityservicecloud/user@example.com
```

In order to provide the same attributes (OCI API Keys, Auth Tokens, Customer Secret Keys, OAuth 2.0 Client Credentials, and SMTP Credentials) for Local and *Federated Users* federation with third-party Identity Providers should only be done in the pre-configured primary IDCS or the Default Identity Domain where applicable.

All users have the same OCI-specific attributes (OCI API Keys, Auth Tokens, Customer Secret Keys, OAuth 2.0 Client Credentials, and SMTP Credentials).

OCI Administration user should only be configured in the pre-configured primary IDCS or the Default Identity Domain where applicable.

Note: Any federated user can be a member of 100 groups only. The OCI Console limits the number of groups in a SAML assertion to 100 groups. User Management in the Enterprise Identity Management system will be handled by ACME.

Authorization

In general, policies hold permissions granted to groups. Policy and Group naming follows the Resource Naming Conventions.

Tenant Level Authorization

The policies and groups defined at the tenant level will provide access to administrators and authorized users, to manage or view resources across the entire tenancy. Tenant level authorization will be granted to tenant administrators only.

These policies follow the recommendations of the CIS Oracle Cloud Infrastructure Foundations Benchmark v1.1.0, recommendations 1.1, 1.2, 1.3.

Service Policy

A Service Policy is used to enable services at the tenancy level. It is not assigned to any group.

Shared Compartment Authorization

Compartment level authorization for the cmp-shared compartment structure uses the following specific policies and groups.

Apart from tenant level authorization, authorization for the cmp-shared compartment provides specific policies and groups. In general, policies will be designed that lower-level compartments are not able to modify resources of higher-level compartments.

Policies for the cmp-shared compartment follow the recommendations of the CIS Oracle Cloud Infrastructure Foundations Benchmark v1.1.0, recommendations 1.1, 1.2, 1.3.

Compartment Level Authorization

Apart from tenant level authorization, compartment level authorization provides compartment structure specific policies and groups. In general, policies will be designed that lower-level compartments are not able to modify resources of higher-level compartments.

Authentication and Authorization for Applications and Databases

Application (including Compute Instances) and Database User management is separate of and done outside of the primary IDCS or Default Identity Domain. The management of these users is the sole responsibility of ACME using the application, compute instance and database specific authorization.

3.2.2.2.3 Security Posture Management

Oracle Cloud Guard

Oracle Cloud Guard Service will be enabled using the pcy-service policy and with the following default configuration. Customization of the Detector and Responder Recipes will result in clones of the default (Oracle Managed) recipes.

Cloud Guard default configuration provides several good settings. It is expected that these settings may not match with ACME's requirements.

Targets

In accordance with the CIS Oracle Cloud Infrastructure Foundations Benchmark, v1.1.0, Chapter 3.15, Cloud Guard will be enabled in the root compartment.

Detectors

The Oracle Default Configuration Detector Recipes and Oracle Default Activity Detector Recipes are implemented. To better meet the requirements, the default detectors must be cloned and configured by ACME.

Responder Rules

The default Cloud Guard Responders will be implemented. To better meet the requirements, the default detectors must be cloned and configured by ACME.

Vulnerability Scanning Service

In accordance with the CIS Oracle Cloud Infrastructure Foundations Benchmark, v1.1.0, OCI Vulnerability Scanning will be enabled using the pcy-service policy.

Compute instances which should be scanned *must* implement the *Oracle Cloud Agent* and enable the *Vulnerability Scanning plugin*.

OCI OS Management Service

Required policy statements for OCI OS Management Service are included in the pcy-service policy.

By default, the *OS Management Service Agent plugin* of the *Oracle Cloud Agent* is enabled and running on current Oracle Linux 6 and Oracle Linux 7 platform images.

3.2.2.2.4 Monitoring, Auditing and Logging

In accordance with the CIS Oracle Cloud Infrastructure Foundations Benchmark, v1.1.0, Chapter 3 Logging and Monitoring the following configurations will be made:

- OCI Audit log retention period set to 365 days.
- At least one notification topic and subscription to receive monitoring alerts.
- Notification for Identity Provider changes.
- Notification for IdP group mapping changes.

- Notification for IAM policy changes.
- Notification for IAM group changes.
- Notification for user changes.
- Notification for VCN changes.
- Notification for changes to route tables.
- Notification for security list changes.
- Notification for network security group changes.
- Notification for changes to network gateways.
- VCN flow logging for all subnets.
- Write level logging for all Object Storage Buckets.
- Notification for Cloud Guard detected problems.
- Notification for Cloud Guard remedied problems.

For IDCS or OCI Identity Domain Auditing events, the respective Auditing API can be used to retrieve all required information.

3.2.2.2.5 Data Encryption

All data will be encrypted at rest and in transit. Encryption keys can be managed by Oracle or the customer and will be implemented for identified resources.

3.2.2.2.5.1 Key Management

All keys for **OCI Block Volume**, **OCI Container Engine for Kubernetes**, **OCI Database**, **OCI File Storage**, **OCI Object Storage**, and **OCI Streaming** are centrally managed in a shared or a private virtual vault will be implemented and placed in the compartment cmp-security.

Object Storage Security

For Object Storage security the following guidelines are considered.

- **Access to Buckets** – Assign least privileged access for IAM users and groups to resource types in the object-family (Object Storage Buckets & Object)
- **Encryption at rest** – All data in the Object Storage is encrypted at rest using AES-256 and is on by default. This cannot be turned off and objects are encrypted with a master encryption key.

Data Residency

It is expected that data will be held in the respective region and additional steps will be taken when exporting the data to other regions to comply with the applicable laws and regulations. This should be review for every project onboard into the tenancy.

3.2.2.2.6 Operational Security

Security Zones

Whenever possible OCI Security Zones will be used to implement a security compartment for Compute instances or Database resources. For more information on Security Zones refer to the in the *Oracle Cloud Infrastructure User Guide* chapter on Security Zones.

Remote Access to Compute Instances or Private Database Endpoints

To allow remote access to Compute Instances or Private Database Endpoints, the OCI Bastion will be implemented for defined compartments.

To be able to use OCI services to for OS management, Vulnerability Scanning, Bastion Service, etc. it is highly recommended to implement the Oracle Cloud Agent as documented in the *Oracle Cloud Infrastructure User Guide* chapter Managing Plugins with Oracle Cloud Agent.

3.2.2.2.7 Network Time Protocol Configuration for Compute Instance

Synchronized clocks are a necessity for securely operating environments. OCI provides a Network Time Protocol (NTP) server using the OCI global IP number 169.254.169.254. All compute instances should be configured to use this NTP service.

3.2.2.2.8 Regulations and Compliance

ACME is responsible for setting the access rules to services and environments that require stakeholders' integration to the tenancy to comply with all applicable regulations. Oracle will support in accomplishing this task.

OCI Solution Architecture sample BOM In this chapter we show an example of a list of all the resources necessary to build the architecture. Once again, this want to be just an example and more detailed list should be created to properly evaluate the requirements and the associate costs.

Description	Resource Type	Number
Bastion Server Host	Compute Instance	1
Admin Host	Compute Instance	1
Non-WebLogic Worker Pools Nodes	Compute Instance	2
WebLogic Worker Pools Nodes	Compute Instance	3
Microservices Worker Pools Nodes	Compute Instance	3
Oracle Database	Base DB System on VMs	1
Internal Private Load Balancer for Administration	Load Balancer	1
Public Load Balancer for Internet Access	Load Balancer	1
Private Load Balancer for Intranet Access	Load Balancer	1
File System for Persistent Volumes	File Storage Service	1
Web Application Firewall for Internet Access	Web Application Firewall	1
Optional Fast Connect	Fastconnect	1

3.3 Physical Future State Architecture

The following is the physical more detailed schema of *ACME* architecture.

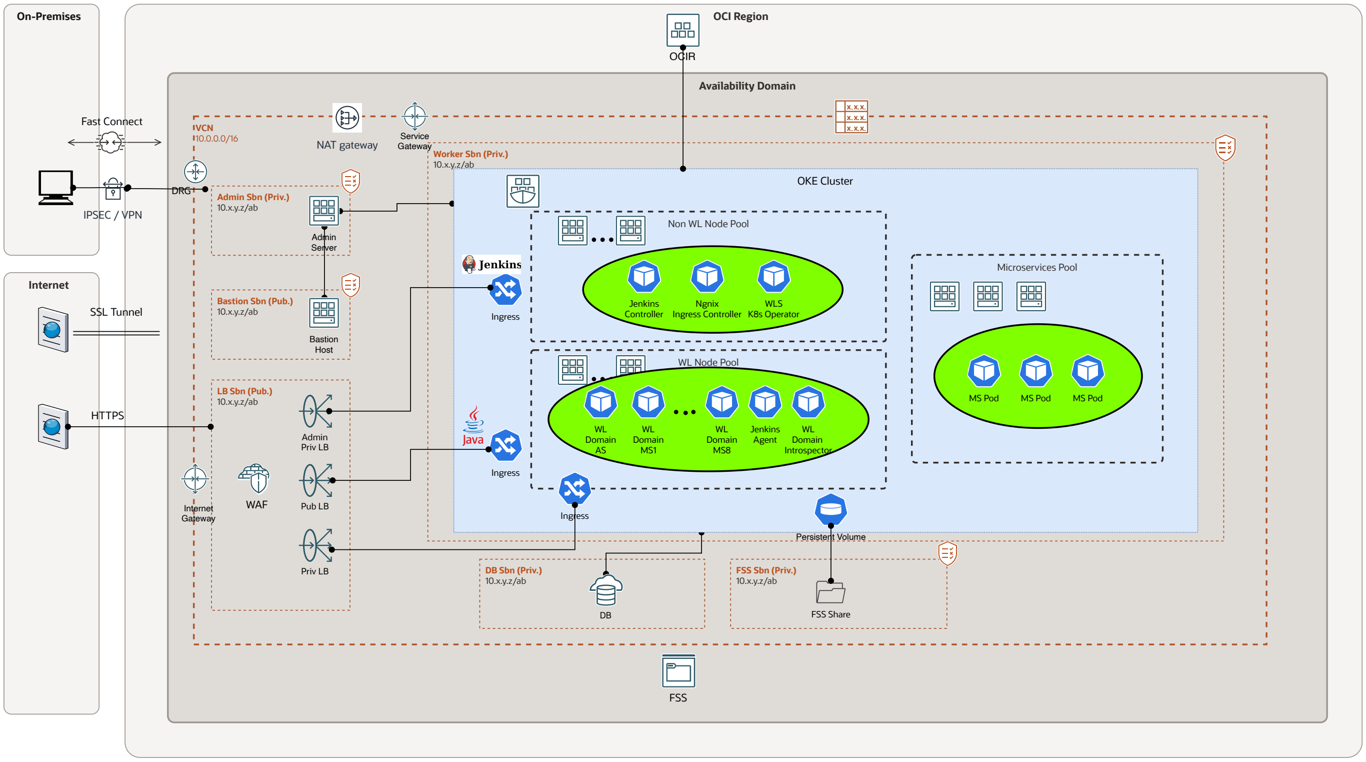

Figure 3 ACME Physical Architecture

Where the glossary for the terminology used is the following:

Region: a collection of availability domains located in a single geographic location.

Availability Domain: one or more isolated, fault-tolerant Oracle data centers that host cloud resources such as instances, volumes, and subnets. A region contains one or more availability domains.

VCN (Virtual Cloud Network): a virtual version of a traditional network—including CIDRs, subnets, route tables, and gateways—on which your instance runs.

NAT Gateway: an optional virtual router that you can add to your virtual cloud network (VCN) to perform Network Address Translation (NAT). A NAT gateway gives cloud resources without public IP addresses access to the internet without exposing those resources to incoming internet connections.

Service Gateway: an optional virtual router that you can add to your virtual cloud network (VCN). The gateway enables on-premises hosts or VCN hosts to privately access Oracle services (such as Object Storage and Autonomous Database) without exposing the resources to the public internet.

Internet Gateway: an optional virtual router that you can add to a virtual cloud network (VCN). It provides a path for network traffic between the VCN and the internet.

WAF (Web Application Firewall): protect applications from malicious and unwanted internet traffic with a cloud-based, PCI-compliant, global web application firewall service.

Since one of the most important aspects is deploying WebLogic for OKE using the Marketplace Stack is the network layout and the subnets sizing we want also to show an example of a possible network configuration and subnetting where we consider also to have Non-Production and Production pools in the same OKE cluster. The deployment of those pools can be managed using the pre-build Jenkins pipelines during the new WebLogic domains creation or manually. If a stronger separation and isolation is necessary different life-cycle environment can be created on separate OKE clusters using different VCNs and compartments but adopting the same *VCN slicing* approach.

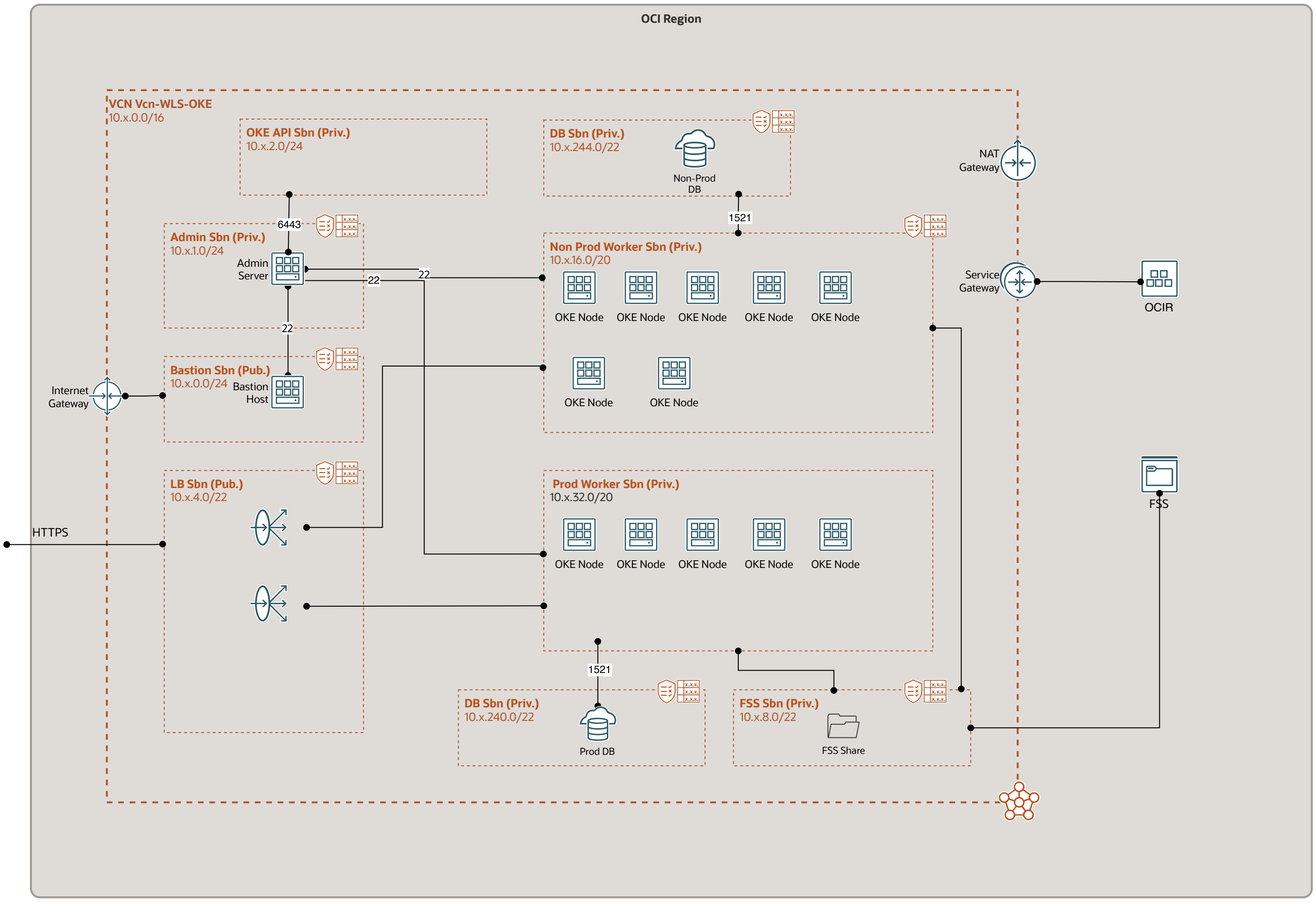

Figure 4 Example of VCN Subnetting in a Multi-Environment WebLogic for OKE Cluster