ORACLE UNIVERSITY

Integrated Cloud Applications & Platform Services

Oracle Cloud IaaS: Compute and Storage Fundamentals

Sudent Guide

D96893GC10

Edition 2.0 | November 2016 | D97030

Learn more from Oracle University at **education.oracle.com**

ORACLE

Authors

Anamika Mukherjee

Mamatha Srinath

Victor Zamora

Technical Contributors and Reviewers

Matt Taylor

Susan Jang

Lachlan Williams

Aparna Nagaraj

Editors

Nikita Abraham

Kavita Saini

Graphic Editor

Maheshwari Krishnamurthy

Publishers

Michael Sebastian

Jayanthy Keshavamurthy

Asief Baig

Joseph Fernandez

Contents

1 Course Overview
Course Objectives 1-2
Target Audience 1-3
Schedule 1-4
Course Practices 1-7
Classroom Guidelines 1-8
Resource: Links 1-9

2 Introduction to Oracle Linux
Lesson Objectives 2-2
Linux Kernel 2-3
Linux Kernel Development Model 2-4
Linux Distributions 2-5
Oracle Linux 2-6
Oracle's Technical Contributions to Linux 2-7
Oracle Linux: Compatible with Red Hat Enterprise Linux (RHEL) 2-8
Unbreakable Enterprise Kernel 2-9
How Do I Learn More? 2-10
Quiz 2-11
Summary 2-12

3 Introduction to Virtualization
Objectives 3-2
What Is Virtualization? 3-3
What Does Virtualization Look Like? 3-4
Physical Components 3-5
Software Component: Hypervisor 3-6
Does the Host Machine Need an OS to Run a Hypervisor? 3-7
Xen Technology 3-8
Why Use Virtualization? 3-9
How Does Virtualization Help in Resource Optimization? 3-10
How Does Virtualization Help in Consolidation? 3-11
How Does Virtualization Help to Maximize Uptime? 3-12
How Does Virtualization Help to Easily Migrate Workloads? 3-13

How Does Virtualization Help to Protect Investment in Existing Legacy Systems? 3-14
Quiz 3-15
Summary 3-20

4 Introduction to IaaS
Objectives 4-2
What Is Infrastructure as a Service (IaaS)? 4-3
What Are the Benefits of IaaS? 4-4
Role of IaaS in Cloud Computing 4-5
In IaaS, Who Controls What? 4-6
What IaaS Services Does Oracle Offer? 4-7
Oracle Compute Cloud Service 4-8
Oracle Storage Cloud Service 4-9
Oracle Network Cloud Service: VPN for Dedicated Compute 4-10
Oracle Network Cloud Service: FastConnect 4-11
Quiz 4-12
Summary 4-16

5 Introduction to Oracle Compute Cloud Service
Objectives 5-2
What Is Oracle Compute Cloud Service? 5-3
Features of Oracle Compute Cloud Service 5-4
Create Instances 5-5
Assign Processor and Memory Resources 5-6
Use a Persistent Boot Disk 5-7
Attach High-Capacity Block Storage to Instances 5-8
Practice Fine-Grained Control over Network Traffic 5-9
Reserve and Assign a Persistent Public IP Address 5-10
Ensure Secure Access to Instances 5-11
Monitor and Manage Your Resources 5-12
Automate Provisioning and Management Workflows 5-13
Migrate Applications to the Cloud 5-14
Summary of Oracle Compute Cloud Service Features 5-15
What Else Can I Do with Compute? 5-16
Who Uses Oracle Compute Cloud Service? 5-17
Quiz 5-18
Summary 5-20
Resource: Links 5-21

6 Introduction to Storage Concepts
Objectives 6-2
What Is Storage? 6-3
What Differentiates Cloud Storage from Local Storage? 6-4
Local Storage 6-5
Cloud Storage 6-6
What Are the Different Types of Cloud Storage? 6-7
So, Which Is Better: Local Storage or Cloud Storage? 6-8
What Are the Different Types of Storage? 6-9
Tell Me About Block Storage 6-10
Tell Me About Object Storage 6-11
Tell Me About Ephemeral Storage 6-12
Storage: The Big Picture 6-13
Quiz 6-14
Summary 6-19

7 Introduction to Oracle Storage Cloud Service
Objectives 7-2
Oracle Cloud: Overview 7-3
So What Exactly Is Object Storage on Oracle Cloud? 7-4
How Does Object Storage Work? 7-5
Object Storage Elements 7-6
Other Storage Solutions 7-7
What Are the Benefits of Oracle Storage Cloud Service? 7-8
What Features Does Oracle Storage Cloud Service Provide? 7-9
Architectural Overview 7-11
Oracle Storage Cloud Service - Replication Policy 7-12
Interface to Oracle Storage Cloud Service 7-13
How Do I Access Object Storage on Cloud? 7-14
Installing Cygwin with cURL (1/3) 7-15
Installing Cygwin with cURL (2/3) 7-16
Installing Cygwin with cURL (3/3) 7-17
Starting Cygwin 7-18
REST URLs: URL for the Account 7-19
REST URLs: URL for Containers and Objects 7-21
Quiz 7-22
Summary 7-26

8 Oracle Compute Cloud Service Instances
Objectives 8-2
What Is an Instance? 8-3

How Do I Create an Instance from the Web Console? 8-5
What Is an SSH Key? 8-6
How Do I Use an SSH Key? 8-7
How Do I Generate an SSH Key Pair on Linux? 8-8
How Do I Generate an SSH Key Pair on Windows? 8-10
Saving the SSH Private Key on Windows 8-12
Saving the SSH Public Key on Windows 8-13
I've Generated the SSH Key Pair. What's Next? 8-14
Create Instance Wizard - Image Page 8-15
Create Instance Wizard - Shape Page 8-16
Create Instance Wizard – Instance Page 8-17
Create Instance Wizard – Storage Page 8-19
Can I Create an Instance Using an App from Oracle Cloud Marketplace? 8-20
How Do I Know If My Instance Is Running? 8-21
Instance Life Cycle 8-22
How Can I Log in to My Instance from Linux? 8-23
How Can I Log in to My Instance from Windows? 8-24
What's on My Instance? 8-25
How Do I Stop, Restart, and Delete My Instance? 8-27
Quiz 8-28
Summary 8-31
Resource: Links 8-32

9 Oracle Compute Cloud Service Network Settings
Objectives 9-2
How Do I Control Access to My Instance? 9-3
How Do Security Lists Work? 9-4
How Do Security Rules Work? 9-5
What Should I Know About Security Lists? 9-6
Security Lists: An Example 9-8
What Should I Know About Security Rules? 9-9
Security Rules: An Example 9-11
Is That How SSH Access to My Instance Works? 9-12
So, Does My Instance Need a Public IP Address? 9-13
Does My Instance Need a Private IP Address? 9-14
I Get It. Can I Configure My Network Settings Now? 9-15
How Do I Create a Security List? 9-16
How Do I Add My Instance to a Security List? 9-17
How Do I Create a Security IP List? 9-18
Which Security Application Do I Use? 9-19
How Do I Create a Security Rule? 9-20

But I Don't Want My Instance to be Accessible by Everybody. How Do I Disable Access to My Instance? 9-21
How Do I Remove an Instance from a Security List? 9-22
How Do I Disable a Security Rule? 9-23
Quiz 9-24
Summary 9-27
Resource: Links 9-28

10 Oracle Compute Cloud Service Block Storage Volumes
Objectives 10-2
How Can I Use Block Storage with My Instance? 10-3
Workflow for Using a Block Storage Volume 10-4
How Do I Create a Block Storage Volume? 10-5
How Do I Add a Block Storage Volume to an Existing Instance? 10-6
How Do I Find the Storage Volume on My Instance? 10-7
What Should I Know About File Systems? 10-8
How Do I Create a File System on the Storage Volume? 10-9
What's a Mount Point? 10-10
How Do I Mount the Storage Volume on My Instance? 10-11
Now Can I Use the Storage Volume? 10-12
What If I Don't Need a Storage Volume Anymore? 10-13
I've Unmounted the Storage Volume. Now How Do I Detach It? 10-15
How Do I Delete the Block Storage Volume? 10-16
Quiz 10-17
Summary 10-20
Resource: Links 10-21

11 Oracle Compute Cloud Service Bootable Storage Volumes
Objectives 11-2
What Is a Bootable Storage Volume? 11-3
What Is a Machine Image? 11-4
How Do I Create a Bootable Block Storage Volume? 11-5
How Do I Create an Instance with this Storage Volume? 11-6
How Do I Access the Storage Volume? 11-8
Quiz 11-9
Summary 11-11
Resource: Links 11-12

12 Oracle Compute Cloud Service Orchestrations
Objectives 12-2
What Is an Orchestration? 12-3
Why Should I Use Orchestrations? 12-4
What Does an Orchestration Look Like? 12-6
Which Objects Can I Create Using an Orchestration? 12-7
What If I Don’t Want All My Objects in One Orchestration? 12-8
How Do I Define Dependencies in an Orchestration? 12-9
How Do I Create an Instance Using an Orchestration? 12-12
Steps to Create an Instance Using an Orchestration 12-13
What Is JSON? Why Should I Use It? 12-14
JSON Syntax 12-15
Sample JSON to Build Your First Orchestration – Single Instance 12-16
Steps to Create an Instance Using an Orchestration 12-18
How Do I Upload My Orchestration to Oracle Compute Cloud Service? 12-19
Steps to Create an Instance Using an Orchestration 12-20
What Are the Prerequisite Objects for Creating an Instance? 12-21
Steps to Create an Instance Using an Orchestration 12-22
Everything’s Fine. Can I Start My Orchestration? 12-23
How Do I Know What My Orchestration Is Doing? 12-24
How Do I Know If My Instance Is Running? 12-26
My Orchestration Status Says Ready. Can I Stop It? 12-27
Quiz 12-28
Summary 12-32
Resource: Links 12-33

13 HTTP and REST Fundamentals
Objectives 13-2
Topics 13-3
A Brief History of the Internet and World Wide Web 13-4
How Does It All Work? 13-5
What Are Web Servers? 13-6
How Do Web Servers Work? 13-7
What Is a Client? 13-8
Topics 13-9
What Is HTTP? 13-10
HTTP Request 13-11
HTTP Request: URL 13-12
HTTP Response 13-13
Response Bodies 13-14
An HTTP Response 13-15

Topics 13-16
What Is REST? 13-17
How Do I Use REST on the Web? 13-18
Practical REST 13-19
Can I Use REST from the Command Line? 13-21
Topics 13-22
Exchanging Data with REST 13-23
XML Versus JSON in Web Services 13-24
What Is JSON? 13-25
JSON: Overview 13-26
JavaScript Literals in JSON 13-27
What Do I Need to Know to Use REST? 13-28
HTTP Methods 13-29
RESTful URI 13-30
Quiz 13-31
Summary 13-34

14 Oracle Storage Cloud Service Access and Authentication
Objectives 14-2
How Do I Authenticate Access to Using REST API? 14-3
How Do I Request an Authentication Token? 14-4
Output 14-5
Now How Do I Construct the Authentication URL for My Account? (1/2) 14-6
Now How Do I Construct the Authentication URL for My Account? (2/2) 14-7
Done. What's Next? 14-8
Where Can I Find My User Information? 14-9
How Do I Store an Object? 14-10
Output 14-11
Using Access Control Lists (ACLs) to Protect Data 14-12
So, What Are the Default Values when Creating a New Container? 14-13
Example: Creating a Container with Default Values 14-14
Quiz 14-15
Summary 14-19

15 Oracle Storage Cloud Service Containers and Objects
Objectives 15-2
Two Sections 15-3
Containers 15-4
SECTION I: Creating Containers 15-5
SECTION I: Listing Containers 15-7
SECTION I: Setting Container Metadata 15-10

SECTION I: Setting Container Metadata: ACLs 15-11
SECTION I: Setting Container Quotas 15-16
SECTION I: Setting Custom Metadata 15-19
SECTION I: Deleting Containers 15-21
Objects 15-24
SECTION II: Listing Objects in a Container 15-25
SECTION II: Creating Objects 15-28
SECTION II: Downloading Objects 15-31
SECTION II: Deleting Objects 15-33
SECTION II: Updating Object Metadata 15-35
SECTION II: Scheduling Automatic Deletion of Objects 15-38
Quiz 15-42
Summary 15-47

1

Course Overview

Course Objectives

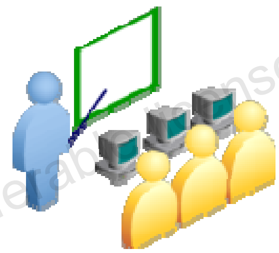
After completing this course, you should be able to:

- Explain virtualization, IaaS, cloud computing, and storage concepts
- Use Oracle Compute Cloud Service to create and customize virtual machines (instances)
- Configure network and security settings for your Oracle Compute Cloud Service instances
- Use Block Storage Volumes and Bootable Storage Volumes for your Oracle Compute Cloud Service instances
- Understand and use REST API
- Access and use Oracle Storage Cloud Service to store and retrieve data

ORACLE

Target Audience

- This course is intended for anyone interested in Compute and Storage (IaaS) areas.
- This course helps you to develop a fundamental knowledge of:
  - Oracle Compute Cloud Service
  - Oracle Storage Cloud service
- A basic knowledge of Linux is a prerequisite for this course.



Schedule

Day	Lessons
Day 1	Lesson 1 – Course Overview Lesson 2 – Introduction to Oracle Linux Lesson 3 – Introduction to Virtualization Lesson 4 – Introduction to IaaS Lesson 5 – Introduction to Oracle Compute Cloud Service Lesson 6 – Introduction to Storage Concepts Lesson 7 – Introduction to Oracle Storage Cloud Service Lesson 8 – Oracle Compute Cloud Service Instances

Schedule

Day	Lessons
Day 2	Lesson 9 – Oracle Compute Cloud Service Network Settings Lesson 10 – Oracle Compute Cloud Service Block Storage Volumes Lesson 11 – Oracle Compute Cloud Service Bootable Storage Volumes Lesson 12 – Oracle Compute Cloud Service Orchestrations Lesson 13 – HTTP and REST Fundamentals

Schedule

Day	Lessons
Day 3	Lesson 14 – Oracle Storage Cloud Service Access and Authentication Lesson 15 – Oracle Storage Cloud Service Containers and Objects

ORACLE

Course Practices

- Lessons are reinforced with hands-on practices, where required.
- The practices include a scripted solution to help you to execute the tasks.
- You can access Oracle Cloud Service to complete the practices.
- You should have received an email with the following:
  - Your user name and password
  - Your identity domain

ORACLE

Classroom Guidelines

- The instructor starts each session at the scheduled time.
- Ask questions, but with respect to the current topic and also in the interest of other students.
- Ensure that cell phones are in the silent mode.

Resource: Links

Resource	URL
Documentation	http://docs.oracle.com/en/
Oracle IaaS	http://docs.oracle.com/cloud/latest/?tab=4
Oracle Linux	http://docs.oracle.com/cd/E37670_01/index.html
Oracle Compute Cloud Service Tutorials	http://docs.oracle.com/cloud/latest/stcomputecs/compute-cloud-tutorials.html
Oracle Storage Cloud Service Tutorials	http://docs.oracle.com/cloud/latest/storagecs_common/storagecs_tutorials.htm

2

Introduction to Oracle Linux

ORACLE

Lesson Objectives

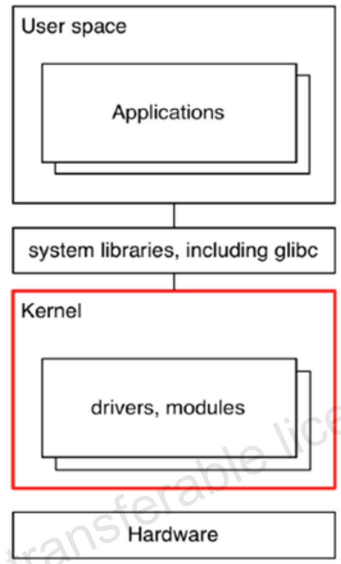
After completing this lesson, you should be able to describe:

- The history of the Linux operating system
- The Linux kernel development model
- Linux distributions
- Oracle's comprehensive Linux solution
- Oracle's contributions to the Linux community
- Oracle Linux's compatibility with Red Hat Enterprise Linux (RHEL)
- Unbreakable Enterprise Kernel

ORACLE

Linux Kernel

- Linux is modular in design:
  - User space
  - Kernel
- Modular design allows for a large development community, better fault isolation, and security.
- Linus Torvalds developed the original Linux kernel.
- Linux version 0.01 was released in September 1991.
- The name Linux is a combination of Linus and UNIX.



The Linux operating system is a modular system. At the lowest level, the kernel interacts with the hardware and controls and schedules access to resources (CPU, memory, storage, network, and so on) on behalf of applications. Applications run in what is called the user space and call only a stable set of system libraries to ask for kernel services. The `glibc` library is the GNU C library that defines the system calls and other basic facilities, such as `open`, `malloc`, and `printf`. Nearly all applications, including Oracle Database, use this library.

This modular design allows components of Linux to originate from different developers, each of which has their own specific design goals in mind. A modular design also means that the Linux kernel is independent of applications and interfaces. The result is that application crashes and security vulnerabilities in applications tend to remain isolated, rather than affecting the system as a whole.

The Windows operating system, alternatively, has a high degree of integration with applications and interfaces. This can have significant security and stability consequences. For example, the Windows kernel is heavily integrated with the graphical user interface.

In Linux, each component is configured separately, typically by using text-based configuration files. Configurations are not in a cryptic database (the Windows Registry). Reading and writing configuration information can be done by scripts or applications by using simple text parsing engines. No special application programming interface (API) is required to interface with the system configuration data.

Linux Kernel Development Model

- Thousands of developers contribute to frequent releases of the kernel.
- Features are pushed upstream through mail lists and IRC.
- New releases deliver stable updates, new features, and performance improvements.
- The Linus Torvalds–led team makes the new releases.
- Mainline kernels are released approximately every three months.
- Kernel branches are available at http://www.kernel.org.
- Linux kernel development uses Git as the source-code control system.

Thousands of developers representing hundreds of corporations contribute to frequent releases of the Linux kernel. The development effort has been called one of the largest cooperative software projects ever attempted. Stable updates are created regularly, which include new features, support for new devices, and performance improvements.

The Linux community collaborates through various mailing lists that are set up to handle kernel development. Features are pushed upstream, through these mail lists and Internet Relay Chat (IRC). Upstream is the term used for a community-owned version of a specific project. This is where the development happens and always has the most recent changes. You can subscribe to some of these development mailing lists at http://vger.kernel.org/vger-lists.html.

Linus Torvalds leads a team that releases new versions, called "vanilla" or "mainline" kernels. A new version of this mainline kernel is officially released approximately every three months. The mainline branch of development incorporates new features, security fixes, and bug fixes. It is not considered a "stable" branch until it undergoes thorough testing. Separate stable branches for each released version exist. The stable branches do not include the latest features, but do include bug fixes.

A number of kernel versions are currently being maintained as stable kernels. These kernels have patches that are backported to them. These patches are primarily driver updates and security fixes. Kernel branches are available at http://www.kernel.org.

Linux Distributions

- Linux distributions:
  - Are built on top of the Linux kernel
  - Are complete operating systems and more
  - Include compiled binaries and source code
- There are hundreds of Linux distributions.
  - Commercially backed distributions
  - Linux community–driven distributions
- Example:
  - Oracle Linux, Debian, Fedora, Red Hat Enterprise Linux (RHEL), Ubuntu, and many others

ORACLE

A Linux distribution (distro for short) is a collection of software built on top of the Linux kernel and offered as a complete package. Distributions are full operating systems plus some additional applications, such as graphics packages, OpenOffice, and others. The kernel is just one component of a Linux distribution.

A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation and a window system, proprietary applications, free applications, distribution-specific applications for configuration and installation, user manuals, and support information. Most of the software is distributed both as compiled binaries and source code. This allows users to modify and compile the original source code.

There are hundreds of Linux distributions, both commercially backed distributions from companies, such as Red Hat and Novell, as well as Linux community–driven distributions. Some of the more well-known distributions include:

- Oracle Linux
- Debian
- Fedora (a Red Hat–sponsored and community-supported distribution)
- Red Hat Enterprise Linux (RHEL): RHEL is the commercial version of Fedora.
- Ubuntu: Canonical is the vendor behind Ubuntu.

Oracle Linux

ORACLE
LINUX

- Brings the latest Linux innovations to customers
- Is the best-performing, most modern and reliable Linux OS
- Tracks mainline closely
- Influences Linux roadmap upstream via direct code contributions
- Provides highest-value, enterprise-class support
- Deployment best practices: Full stack tested with real-world workloads
- Provides comprehensive legal indemnification
- Lowers cost
- Ksplice: Apply kernel updates on a running system

ORACLE

Oracle offers a comprehensive Linux solution including:

- Dedicated development team
- Dedicated QA team
- Dedicated support team
- Dedicated ISV and IHV team
- Oracle Linux training and certification
- Oracle Linux consulting services

Ksplice allows you to do kernel updates without having to reboot the system. A kernel update comes from either Oracle or from the kernel community. The Ksplice team takes the update and works it into a binary patch that is inserted into a running kernel. You apply it by using the Ksplice tools and the patch is up and running.

Security updates are announced to the world, and there is typically a time period between when a security problem is globally known and when system administrators have an opportunity to patch their systems. Ksplice allows you to apply security updates without having to wait for your users to tell you it is okay to take down the system. This problem is even more significant when running a large number of systems. Ksplice allows you to maintain highly available systems that are also very secure.

Oracle's Technical Contributions to Linux

- Oracle has a dedicated Linux kernel development team.
- Oracle's technical contributions to Linux include:
  - ASMLib
  - Asynchronous IO (AIO) Kernel Subsystem
  - Btrfs file system
  - Oracle Cluster Filesystem (OCFS2)
  - Linux data integrity based on the T10-PI standard
  - Xen Hypervisor
- All Oracle Linux code is available to the Linux community.
- The Git source tree with change logs and commit messages is available at: http://oss.oracle.com/git/

ORACLE

Oracle's long-term vision is focused on continuing to enhance and extend the enterprise-class capabilities of Linux, and this vision is manifest through the many projects and code contributions that Oracle shares with the Linux community. Oracle continues to strengthen its involvement in the Linux community by providing enhancements that facilitate the development and deployment of enterprise Linux solutions. With Oracle Linux, regarding the code developed, 100% of that code becomes available to the open source community for Linux.

Oracle Linux: Compatible with Red Hat Enterprise Linux (RHEL)

- Source and binaries are fully compatible with RHEL.
- Applications that run on RHEL run on Oracle Linux.
- Trademarks and logos have been removed, but there are no compatibility issues.
- `/etc/oracle-release` was added to identify code obtained from Oracle.
- Oracle continues to track RHEL releases with Oracle Linux ISO releases and errata stream.

Oracle Linux is fully compatible—both source and binary—with Red Hat Enterprise Linux. Applications that run on RHEL run on Oracle Linux.

Strict Binary Compatibility

Tremendous effort has gone into assuring that there is no divergence from the original Red Hat source code, given that the main goal of Oracle Linux and the Oracle Linux Support program is to not fragment the Linux code base, but to improve Linux quality and support.

Oracle Linux is built from the very same source code as Red Hat Enterprise Linux (RHEL). A byte-by-byte comparison of the source code against RHEL reveals no differences, the only changes being the removal of trademarks and copyrights.

Trademarks and logos have been removed from a small number of the packages. These are non-functional text or graphic changes that in no way affect any program code, and they do not generate any compatibility issues. Oracle has added its own text file, `/etc/oracle-release`, so support teams can easily identify that they obtained the code from Oracle.

RHEL provides a text file called `/etc/redhat-release`, which contains a one-line string identifying the specific distribution release. This file is part of the `redhat-release` package. Oracle Linux also contains a text file called `/etc/redhat-release`, which is installed by a package called `oraclelinux-release`.

Unbreakable Enterprise Kernel

- Oracle announced the Unbreakable Enterprise Kernel in September 2010.
- It is used by Exadata and Exalogic for extreme performance.
- The Unbreakable Enterprise Kernel is available since Oracle Linux version 5.5.
- Since Oracle Linux 5.5, you have a choice:
  - Red Hat Compatible Kernel
  - Unbreakable Enterprise Kernel
- Oracle is committed to offering compatibility with Red Hat.
- Full support is offered for customers running either kernel.
- Existing applications run unchanged.

In September 2010, Oracle announced the new Unbreakable Enterprise Kernel for Oracle Linux as a recommended kernel to deploy with Oracle Linux 5 or Red Hat Enterprise Linux (RHEL) 5. Beginning with Oracle Linux 5.5 (default in 5.6), you have a choice when it comes to the kernel, either the Red Hat Compatible Kernel or the Unbreakable Enterprise Kernel. In Oracle Linux 5.6, the Unbreakable Enterprise Kernel became the default kernel.

The initial motivation for creating the Unbreakable Enterprise Kernel was to have a modern and best-performing Linux kernel for the Exadata and Exalogic engineered systems. The kernel needed to scale with the larger number of CPUs, memory, and InfiniBand connects.

Unbreakable Enterprise Kernel is heavily tested with Oracle workloads and therefore recommended for Oracle deployments and all other enterprise deployments. Oracle is committed to offering compatibility with Red Hat, and continues to release and support the Red Hat Compatible Kernel as part of Oracle Linux, for customers that require strict RHEL compatibility. Under the Oracle Linux Support Program, customers can receive full support for Oracle Linux running with either kernel.

Using the Unbreakable Enterprise Kernel instead of the Red Hat compatible kernel changes only the kernel. Nothing changes in the user space. Existing applications run unchanged regardless of which kernel is used. Using a different kernel does not change system libraries such as `glibc`. The `glibc` version in Oracle Linux 6 is 2.12, regardless of the kernel version.

How Do I Learn More?

To learn more about Oracle Linux System Administration, sign up for an Oracle Linux course here: Oracle University

Quiz

Oracle Linux offers a Red Hat–compatible kernel as well as a kernel that is optimized for Oracle applications.

a. True
b. False

Summary

In this lesson, you should have learned:

- The history of the Linux operating system
- The Linux kernel development model
- Linux distributions
- Oracle’s comprehensive Linux solution
- Oracle’s contributions to the Linux community
- Oracle Linux’s compatibility with Red Hat Enterprise Linux (RHEL)
- Unbreakable Enterprise Kernel

To learn more about Oracle Linux System Administration, sign up for an Oracle Linux course here: Oracle University

3

Introduction to Virtualization

Objectives

After completing this lesson, you should be able to:

- Define virtualization
- Describe virtualization concepts
- Explain why virtualization is relevant and useful
- Explain how virtualization helps companies save costs

ORACLE

What Is Virtualization?

- Virtualization refers to the process of creating a virtual version of something instead of its actual physical component.
- In computing, virtualization applies to:
  - Hardware components
  - CPU or processors
  - Storage
  - Network interfaces that constitute a virtual machine
- The virtual operating system platform has its own user applications.

ORACLE

- Virtualization allows you to have multiple virtual operating systems running at the same time, as part of one physical machine.
- In computing, virtualization generates what is known as “virtual machines.”
- Each virtual machine runs on its assigned virtual platform, and involve virtual hardware:
  - Virtual CPU, also known as a virtual processor, is a physical CPU that is assigned to a virtual machine.
  - Virtual Storage is the pooling of physical storage from multiple sources into what appears to be a single storage source.
  - Virtual Network – The two most common virtual networks are the following: 1) the one that connects virtual machines inside a hypervisor, and 2) protocol-based virtual networks, such as VLANs (Virtual LANs), VPNs (Virtual Private Networks) and VPLSs (Virtual Private LAN Services).
- All virtual machines run independently from one another—each with their own operating system and set of applications.

What Does Virtualization Look Like?

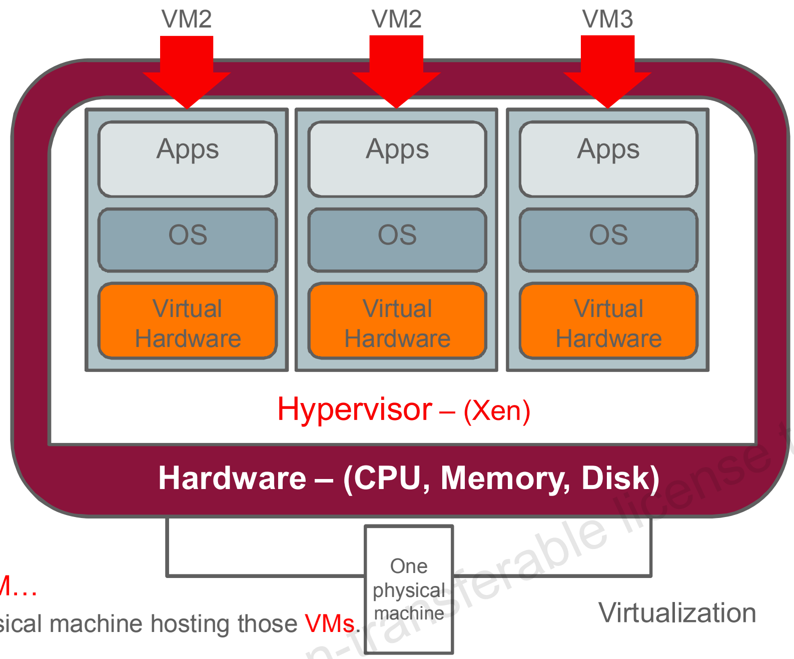
Virtualization is composed of:

- One physical machine
  - Hardware (CPU, memory, hard drive, and so on)
- Hypervisor (Xen)
  - Virtual hardware
  - Software (OS, Apps)

VMs

- VM1, VM2, VM…

VMs share a set of resources available on the physical machine hosting those VMs.



ORACLE

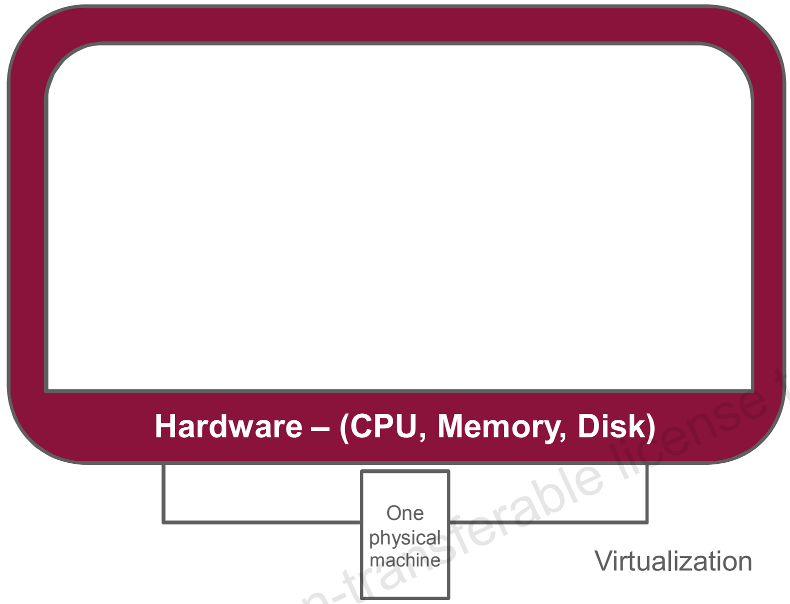
The diagram in the slide illustrates virtualization.

Major components in virtualization include:

- One physical machine that contains the necessary hardware resources
  - CPU, memory, network, hard drives, and so on
- Then, through the hypervisor, a virtual version of each hardware component is created.
  - Virtual CPU, virtual memory, virtual network, virtual hard drives, and so on
- The hypervisor is what allows for virtual machines to run their own operating system and applications.

Physical Components

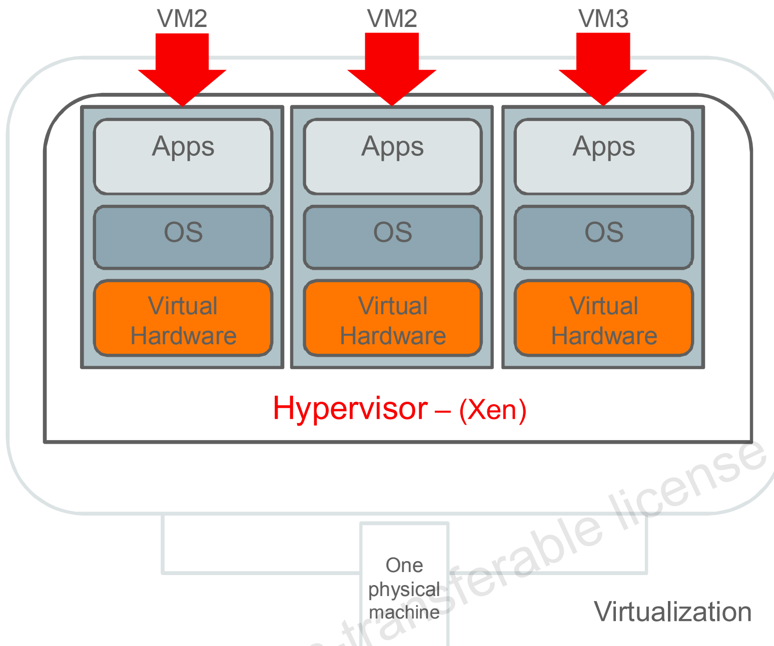
- One physical machine
  - Server
  - PC
- Hardware (CPU, Memory, Disk, Network, and so on)



- One machine can house multiple independent operating systems.
- All virtual machines share the same physical hardware.

Software Component: Hypervisor

- Hypervisor creates a virtual platform on the host computer, on top of which multiple guest operating systems are executed and monitored.
- These guest operating systems run based on virtual hardware configured by the hypervisor.
- Applications are then executed within its operating system.



ORACLE

- Hypervisor
  - Is also called Virtual Machine Monitor (VMM)
  - Creates virtual platforms for (guest) operating systems
  - Controls access to the hardware resources
- Virtual Hardware
  - The hypervisor also provides virtual hardware to meet requirements for each individual operating system.
- Operating System
  - As mentioned earlier, the hypervisor creates and assigns virtual hardware components to each virtual machine. When the user is looking at the hardware information, the guest OS shows what was assigned by the hypervisor (virtual hardware components)—but the guest OS displays them as if they were its native, actual, hardware components.
- Apps
  - Applications are executed and run within their own environment.

Does the Host Machine Need an OS to Run a Hypervisor?

- Type 1: Native or Bare Metal Hypervisor
  - Software that runs directly on the host's hardware.
  - It monitors the guest operating systems.
  - The guest operating system runs on a separate level above the hypervisor.
- Type 2: Hosted Hypervisor
  - Designed to run on top of an existing operating system.
  - Adds a distinct software layer on top of the host operating system.
  - The guest operating system becomes the third software level above the hardware.

Xen Technology

- Xen hypervisor
  - Type 1 bare metal hypervisor
  - Small and lightweight
  - Multiple virtual machines in one host computer
  - Each VM with its own OS

- The Xen hypervisor was originally created by researchers at Cambridge University and derived from work done at the Linux Kernel.
- The Xen hypervisor is a small and lightweight bare metal hypervisor for x86x-compatible computers. It securely executes multiple virtual machines in one host computer.
- Each VM has its own OS with almost native performance.
- The guest operating system runs on a level above the hypervisor.

Why Use Virtualization?

- Most common reason is to reduce cost.
- Other reasons include:
  - Resource optimization
  - Consolidation
  - Maximizing uptime
  - Easily migrating workload as business needs change
  - Protecting investment in existing legacy systems

There are many benefits to virtualization, including saving money, reducing carbon footprint, using computing power more effectively, performing computing tasks more efficiently, and being able to manage, administer, and migrate systems more easily.

Subsequent slides in this lesson describe the benefits of virtualization in more detail.

How Does Virtualization Help in **Resource Optimization**?

- Today's enterprise computer resources are very powerful.
  - Virtualize hardware, and allocate hardware based on needs of users and applications.
  - There is no need to buy additional hardware. Consider creating virtual machines on your existing hardware with unused or underutilized resources.
- Software development environments
  - Each virtual machine is independent and can be isolated from all others.
  - Developers can run and test their applications in a sandbox type of environment on virtual machines without having to worry about product systems or applications.
- Result: Optimized use of your IT resources

ORACLE

Resource Optimization

- Because today's enterprise-level computer resources are so powerful, you can use virtualization to get more out of your one single system.
- Virtualization allows you to use the host computer better, because you now have multiple virtual machines (guests) benefiting from the physical resources (CPU, memory, storage, and so on).
- Virtual machines offer software developers isolated, constrained, test environments.
  - All virtual machines run independently, so it's easy to test and run software applications.

How Does Virtualization Help in **Consolidation**?

- Reduces the number of physical machines in your data center by consolidating them into one and applying virtualization
  - Reduced costs
  - Environment friendly: less heat and power consumption, smaller carbon footprint
- Result: Reduced costs and more environment friendly

Consolidation

- If several applications use a small amount of processing power, many computers can be consolidated into one server running multiple virtual environments.
- For organizations that own hundreds or thousands of servers, consolidation can dramatically reduce physical space, A/C power, and other data center resources.
- Virtualization is also environment friendly, because consolidation results in using fewer resources and less space, which reduces carbon footprint.

How Does Virtualization Help to **Maximize Uptime**?

- Agility:
  - Spin up virtual machines quickly
- Data centers can offer users:
  - Guaranteed uptime of servers and applications
  - Instant deployment of new virtual machines
  - Elasticity, which means resource provisioning when and where required instead of keeping the entire data center in an “always-on” state
  - Reconfiguration of running computing environments without impacting the users
- Result: More efficient and enhanced user experience

Maximizing Uptime

Some other benefits include:

- Speedy disaster recovery if large scale failures do occur
- Aggregated pools of virtual machines via template images

How Does Virtualization Help to **Easily Migrate Workloads**?

- Migration
  - You can move a server environment from one place to another.
  - It is possible to move a virtual machine from one physical machine in the environment to another.
  - A server can be migrated between physical hosts with entirely different hardware configurations.
- Why migrate?
  - To improve reliability and availability
  - In case a virtual machine needs to scale beyond the physical capabilities of the current host and is in need of improved hardware
- Result: Ease of virtual migration

ORACLE

Easy Migration of Workloads as Business Needs Change

- With most virtualization solutions, it is possible to move virtual machines, as well as migrate a server between physical hosts regardless of the difference of their hardware configurations.
- It is important to be able to migrate virtual machines from one physical unit to another. One of the main reasons is hardware improvement and reliability.

How Does Virtualization Help to **Protect Investment in Existing Legacy Systems**?

- What happens when hardware becomes obsolete?
  - Virtualization is an ideal solution.
  - Run a virtual machine on new hardware, while the legacy system itself still behaves as if it were running on the same legacy hardware.
  - Even its performance may benefit from the underlying hardware.
- Result: Ability to execute old software on new hardware

ORACLE

Protecting Investment in Existing Legacy Systems

- Server hardware eventually becomes obsolete, and switching from one system to another can be difficult.
- With VMs, you can continue running legacy systems. From an application perspective, nothing changes.
- This solution gives an organization the time to transition to new processes without worrying about hardware issues, particularly in situations where the manufacturer of the legacy hardware no longer exists or cannot fix broken equipment.

Quiz

Virtualization is when virtual machines are created as part of one physical machine, and share the same hardware resources.

a. True
b. False

Quiz

What does the hypervisor do?

a. Creates platforms for (guest) operating systems
b. Creates virtual machines
c. Provides virtual hardware
d. All of the above
e. None of the above

Quiz

All virtual machines must use the same (guest) operating system.

a. True

b. False

Quiz

Why should you use virtualization?

a. To take full advantage of today's enterprise computing resources
b. To consolidate physical machines
c. To reduce costs
d. All of the above
e. None of the above

Quiz

All virtual machines run independently, so it is easy to test and run software applications.

a. True

b. False

Summary

In this lesson, you should have learned how to:

- Define virtualization
- Describe virtualization concepts
- Explain why virtualization is relevant and useful
- Explain how virtualization helps companies save costs

ORACLE

4

Introduction to IaaS

ORACLE

Objectives

After completing this lesson, you should be able to:

- Define Infrastructure as a Service (IaaS)
- Explain the benefits of IaaS
- Explain how the IaaS layer provides infrastructure for applications and platforms
- Describe Oracle IaaS services

ORACLE

What Is Infrastructure as a Service (IaaS)?

- A form of cloud computing that provides infrastructure services over the Internet
- Required by all applications, databases, and middleware deployments
- Users can access:
  - Computer processors
  - Storage
  - Networks
  - And other infrastructure resources
- One of three "service models" in cloud computing
  - The other two:
    - SaaS – Software as a Service
    - PaaS – Platform as a Service

ORACLE

Infrastructure as a Service (IaaS) is a form of cloud computing that provides infrastructure services over the Internet. One of the main features of IaaS is that users can access their applications and data from anywhere, as long as they can access the Internet. IaaS provides the following benefits:

- The user has access to the foundation level of the computing "task"—compute capacity, network bandwidth, and storage capacity.
- The IaaS provider is responsible for providing the hardware, maintenance, and cloud infrastructure up to the virtualization level.
- The user can create virtual machines as required and decide what operating system and/or apps to use.

The other two service models in cloud computing are:

- **Software as a Service (SaaS):** The user has access only to the applications provided by the SaaS provider. The provider has full control of the entire infrastructure including applications.
  - Example: An email provider
- **Platform as a Service (PaaS):** The user is able to deploy applications using programs and tools provided by the provider. The provider has full control of the infrastructure up to the point where the user is able to modify their applications and environment.
  - Example: A web hosting provider that offers tools and programs to deploy a website

What Are the Benefits of IaaS?

- Customizable application environments
- Pay for what you need
- No maintenance or upgrade hassle for the infrastructure
- Elastic and scalable infrastructure
- High degree of flexibility and control over the environment

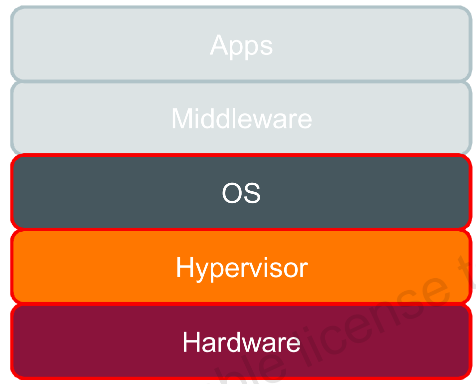
ORACLE

- IaaS allows users to build their application environments without having to buy hardware or rent physical space.
- Subscribers save money because they pay for only the services and resources that they need.
- Maintenance, software and hardware upgrades, and troubleshooting are handled by the IaaS provider, allowing users to focus on more relevant, customizable aspects of their infrastructure.
- Users can create as many VMs as they need, when they need them.
- Users can dynamically control their environment, such as selecting the operating systems of VM and changing the applications that they want to install.
- IaaS provides elasticity and scalability, enabling users to add or remove VMs or other resources easily, whenever required.

All organizations can benefit from IaaS, from small startup businesses to large, well-established corporations. Cloud-based services can be adjusted to meet their varied business needs regardless of size, or computing and storage demands.

Role of IaaS in Cloud Computing

- It provides the infrastructure for:
  - Platforms
  - Applications
- Components of IaaS:
  - Hardware
    - Servers, CPU, storage, network, and so on
  - Software
    - Databases
    - Middleware

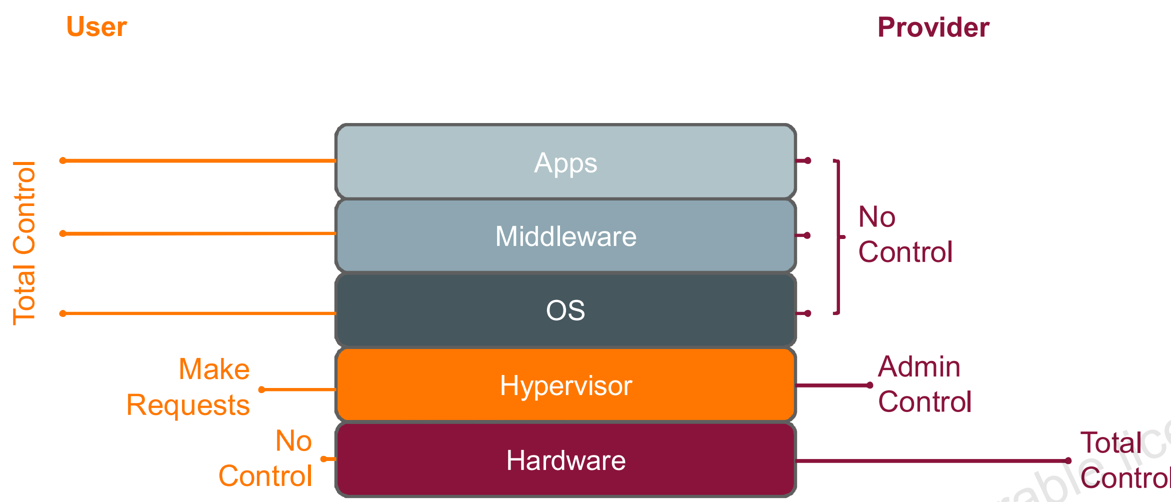


IaaS provides the infrastructure for platforms and applications. IaaS makes it possible for users to deploy virtual machines, install operating systems, and customize their own apps environment.

The main components of IaaS are:

- **Hardware:** Includes as servers, powerful CPUs, large amounts of storage capacity, and network bandwidth
- **Databases:** All the data needs to be administered, controlled, and stored
- **Middleware:** Software that allows communication between servers and users. In other words, this middleware layer makes it possible to input/output data from two different operating systems regardless of what platform they are using.

In IaaS, Who Controls What?



The user has total control starting from the operating system level and above. Users can also make requests at the hypervisor level; however, the IaaS provider has to approve the requests before proceeding.

The provider has total control only at the hardware level, but can approve the user's requests at the hypervisor level. The provider has zero control starting at the operating level and above.

What IaaS Services Does Oracle Offer?

- **Leverage Infrastructure:** Run any workload in the cloud
- **Compute:** Elastic compute capacity to address growing business needs
- **Storage:** Online storage at your fingertips
- **Network:** Connect you datacenter to the cloud

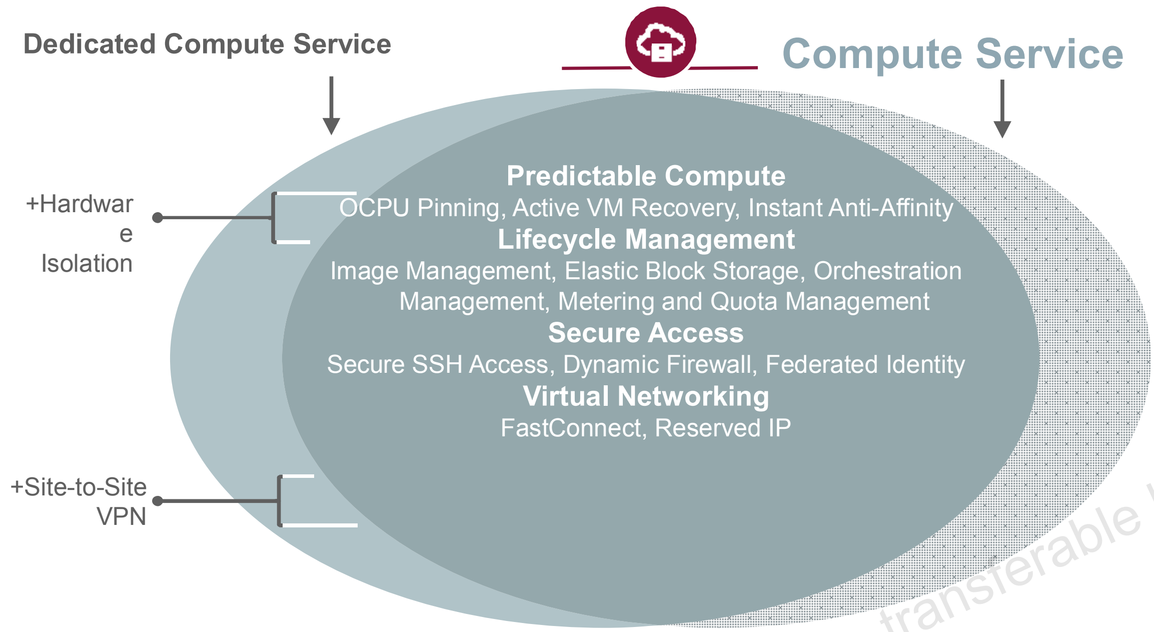
ORACLE

Oracle Cloud Infrastructure as a Service (IaaS) offers a set of core infrastructure capabilities, such as elastic compute and storage to provide customers the ability to run any workload in the cloud.

Specifically for developers, infrastructure services include:

- **Oracle Compute Cloud Service:** To leverage elastic compute capacity to address growing business needs
- **Oracle Storage Cloud Service:** To provide a secure, scalable, reliable, and simple storage solution to meet all of a customer's enterprise needs
- **Oracle Network Cloud Service:** Offers Site-to-Site VPN for dedicated Compute users and FastConnect for all Oracle Cloud customers. Site-to-Site VPN securely extends customers' on-premises network to their dedicated Oracle Compute zone. FastConnect provides a high bandwidth connection between the customers' data centers and Oracle Cloud services.

Oracle Compute Cloud Service



Oracle Compute Cloud Service offers the following:

- **Predictable Compute:**
  - **OCPU Pinning:** Allocate OCPUs to VM instances with no over subscription.
  - **Active VM Recovery:** Configure HA policies to automatically recover failed VMs.
  - **Hardware Isolation (ONLY for Dedicated Compute Service):** Launch dedicated instances on single-tenant hardware.
- **Lifecycle Management:**
  - **Image Management:** Provision virtual machines using pre-packaged images or build your own images.
  - **Elastic Block Storage:** Store data and applications in persistent block storage volumes. Maintain persistence at OS-level using bootable storage volumes.
  - **Orchestration Management:** Automate provisioning and lifecycle operations of virtual compute topologies.
- **Secure Access:**
  - **Secure SSH Access:** Access Oracle Compute Cloud Service instances from a remote host by using a secure shell.
  - **Dynamic Firewall:** Control network traffic among individual instances and/or between group of instances.

The Dedicated Compute offering of Oracle Compute Cloud Service offers everything that Oracle Compute Cloud Service offers. In addition, instances:

- Run on dedicated hardware with network isolation
- Can be accessed using secure site-to-site VPN (more about this later)

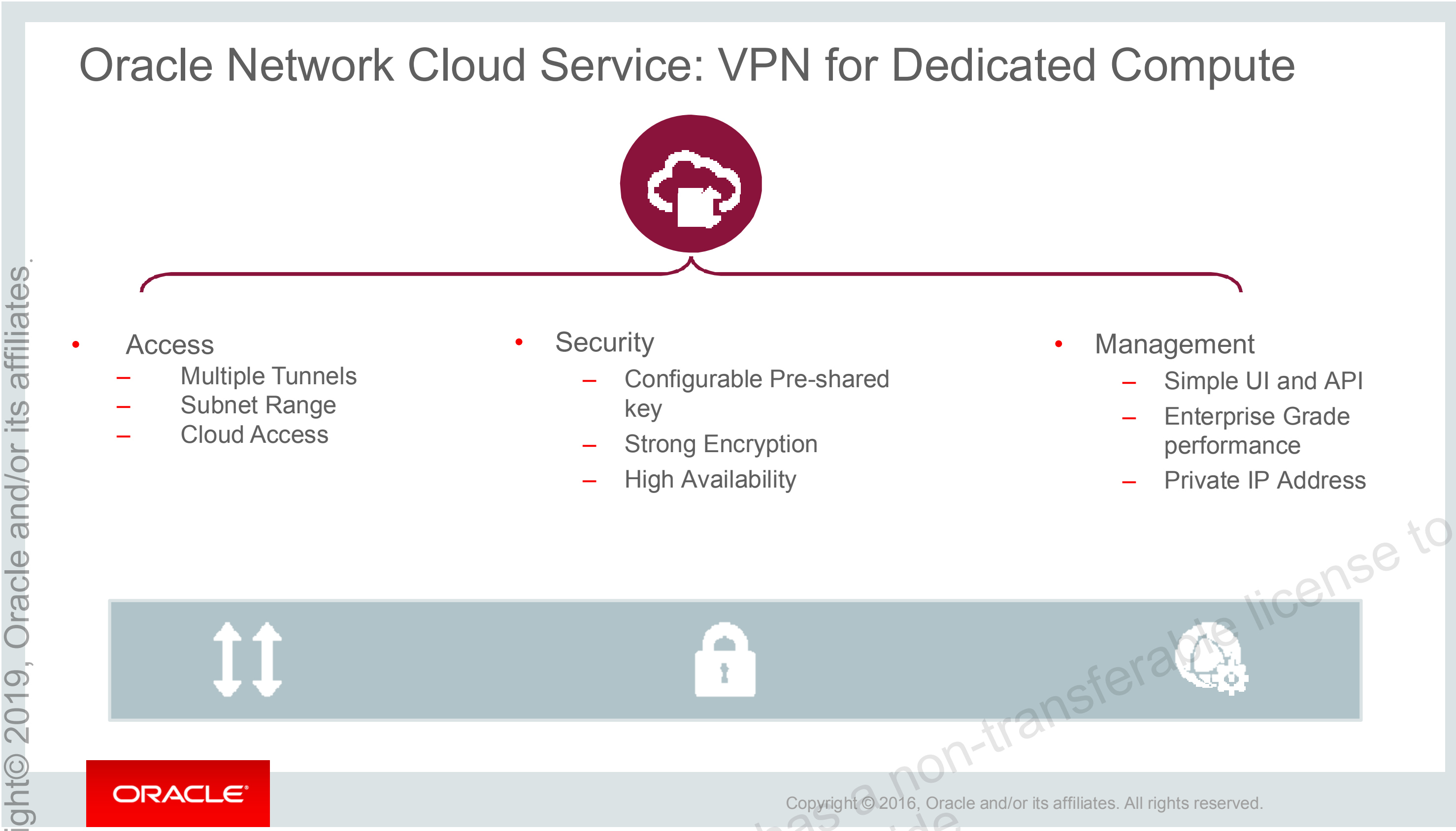
Oracle Storage Cloud Service

- Storage Cloud Service
- Storage Cloud Archive Service
- Shared File Storage Service
- Storage Cloud Software Appliance

ORACLE

Oracle Storage Cloud Service provides secure and scalable, eventually consistent, object storage solution for storing and accessing data from any environment connected to the Internet.

- Oracle Storage Cloud Service
  - Secure and scalable storage
  - Easy to access from anywhere on the Internet
  - Enterprise-class data protection
- Oracle Storage Cloud Archive Service
  - Most cost-effective solution in the industry
  - Designed for large-scale and infrequently accessed data
  - Long-term data retention
  - Rich media content
  - Scientific research archives
  - Cultural preservation



Oracle Storage Cloud Service securely extends your on-premises network to include instances in your Oracle Compute Cloud Service site.

Access

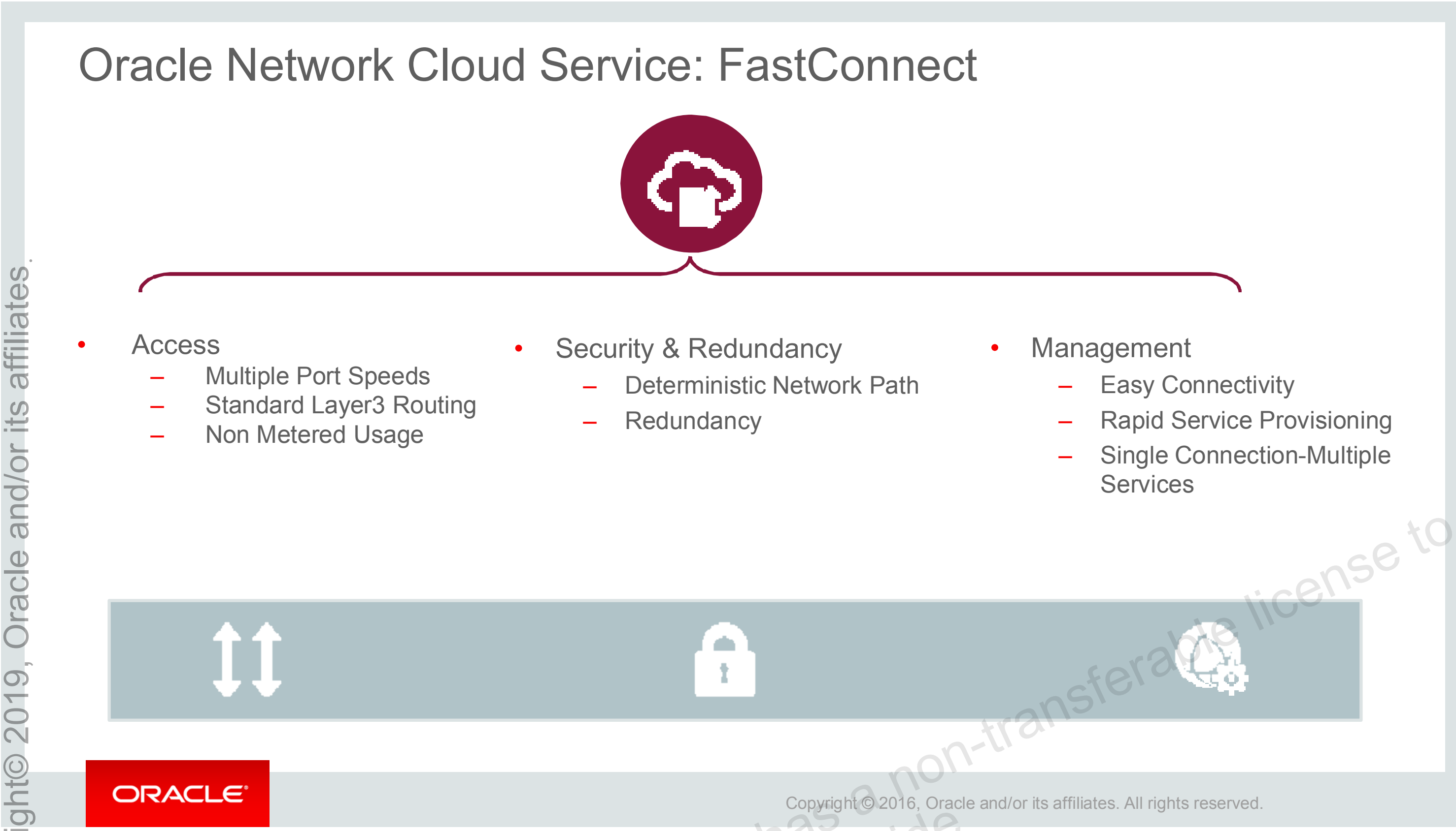
- **Multiple Tunnels:** Multiple site-to-site tunnels can be set up.
- **Subnet Range:** Users can configure a range of private IP addresses for their compute instances.
- **Cloud Access:** Users can access other Oracle services in the cloud.

Security

- **Configurable Pre-shared key:** Symmetric key encryption using Pre-shared key enhances the security as well as overall performance. Users can change it at any time from the services page.
- **Strong Encryption:** All the data between the user's data center and the Oracle Compute Cloud Service site is encrypted using 128 bit AES encryption.
- **High Availability:** VPN devices are configured as a cluster for high availability.

Data Management

- **Simple UI and API:** With few steps, you can create and manage your VPN connections.
- **Enterprise Grade performance:** Ensure that your data is secure while traversing the Internet.
- **Private IP Address:** Private IP addresses can be configured to enable users to access their instances as an extension of their data center.



FastConnect provides a high bandwidth connection between your data center and the Oracle Cloud.

Access

- **Multiple Port Speeds:** Customers can pick between 1 Gbps or 10 Gbps port speeds depending on the use case and the data transfer profile for their usage.
- **Standard Layer3 Routing:** FastConnect Partner Edition leverages industry standard BGP routing to manage the exchange of routes between Oracle Cloud and customers' networks.
- **Non Metered Usage:** FastConnect Partner Edition offers unlimited data transfer between customers' network and Oracle Cloud.

Security and Redundancy

- **Deterministic Network Path:** FastConnect Partner Edition provides predetermined path for data transfer unlike the public Internet thus offering better security as the data never leaves trusted boundaries.
- **Redundancy:** FastConnect Partner Edition is delivered as a fully redundant service with two physical connections from customers' network edge to Equinix Cloud Exchange to ensure high availability.

Data Management

- **Easy Connectivity:** FastConnect Partner Edition is available at Equinix datacenters through a fiber cross connect to the Equinix Cloud Exchange platform.
- **Rapid Service Provisioning:** FastConnect Partner Edition can be turned up rapidly (in minutes) if customers are already connected to Equinix Cloud Exchange.
- **Single Connection-Multiple Services:** FastConnect Partner Edition allows customers to access all supported public facing Oracle Cloud services within a city using a connection through Equinix Cloud Exchange.

Quiz

Infrastructure as a Service (IaaS) is a form of cloud computing that provides only storage services over the Internet.

a. True
b. False

Quiz

IaaS is the only service model in cloud computing.

a. True
b. False

Quiz

How do users benefit when using IaaS?

a. Users save money because they pay for only the services they need.
b. Users can build their application environments without having to buy hardware.
c. Users are not responsible for hardware upgrades.
d. All of the above.
e. None of the above.

Quiz

IaaS provides the infrastructure for platforms and applications. IaaS makes it possible for users to deploy virtual machines, install operating systems, and customize their own applications environment.

a. True

b. False

Summary

In this lesson, you should have learned how to:

- Define Infrastructure as a Service (IaaS)
- Explain the benefits of IaaS
- Explain how the IaaS layer provides infrastructure for applications and platforms
- Describe Oracle IaaS services

5

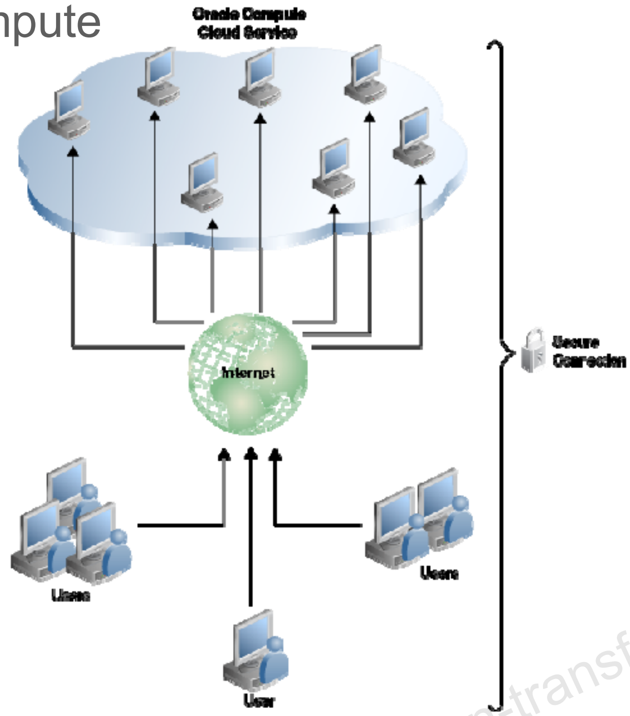
Introduction to Oracle Compute Cloud Service

Objectives

After completing this lesson, you should be able to:

- Provide an overview of Oracle Compute Cloud Service
- Explain the features of Oracle Compute Cloud Service
- Describe the uses and applications of Oracle Compute Cloud Service
- Discuss using apps from Oracle Cloud Marketplace

What Is Oracle Compute Cloud Service?



Oracle Compute Cloud Service is a secure, reliable, low-cost, and standards-based infrastructure service.

You can use Oracle Compute Cloud Service to:

- Rapidly provision virtual machines on Oracle Cloud with all the necessary storage and networking resources
- Manage and scale your virtual machine topology in the cloud easily
- Migrate Oracle and third-party applications to Oracle Cloud

Features of Oracle Compute Cloud Service

F E A T U R E S

1. Create virtual machines using Oracle-provided or custom machine images.
2. Assign processor and memory resources from a range of resource profiles.
3. Use persistent boot disks to start your instance.
4. Attach high-capacity block storage to your instance.
5. Practice fine-grained control over network traffic.
6. Reserve a persistent public IP address and assign it to your instance.
7. Ensure secure access to your instance.
8. Monitor and manage your resources using a web console.
9. Automate provisioning and management workflows using orchestrations.
10. Migrate on-premise workloads and applications to the cloud.

This slide provides an overview of the features of Oracle Compute Cloud Service. Let's look at each of these features in more detail.

Create Instances

F E A T U R E S

1 2 3 4 5 6 7 8 9 10

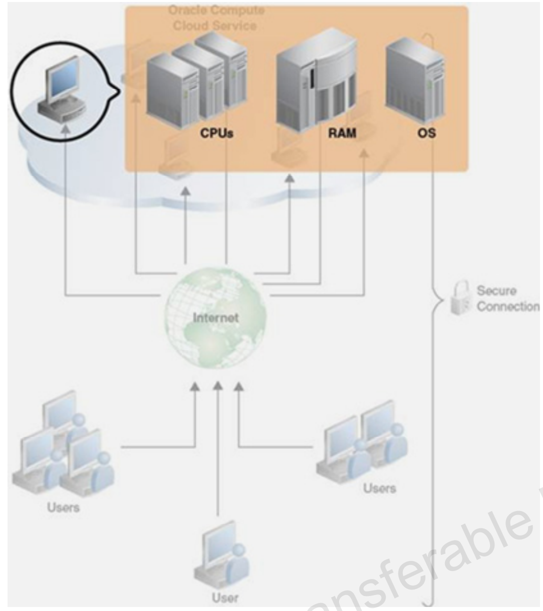
You can create virtual machine instances by using:

- An Oracle-provided machine image*

Or

- A custom machine image that you upload to Oracle Compute Cloud Service

*Machine Image is the template of a virtual hard disk with installed OS and applications.

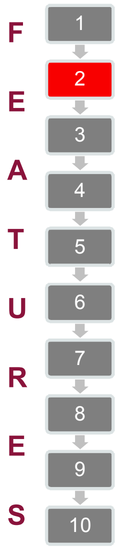


ORACLE

You can use one of several Oracle-provided machine images to quickly provision robust Oracle Linux virtual machines. Alternatively, you can build custom machine images based on the operating system and disk size of your choice and use those images to create virtual machines.

The operating system and software that you use to build machine images must have the required licenses. You are responsible for purchasing the required licenses and ensuring support for any third-party operating systems and software that you run on Oracle Compute Cloud Service instances.

Assign Processor and Memory Resources



Oracle Compute Cloud Service enables you to select from a range of predefined shapes that determine the number of CPUs available in an instance and the amount of RAM available in an instance.

While creating Oracle Compute Cloud Service instances, you can assign CPU and memory resources by selecting from a wide range of resource profiles (called shapes), each of which is a carefully designed combination of processor and memory limits.

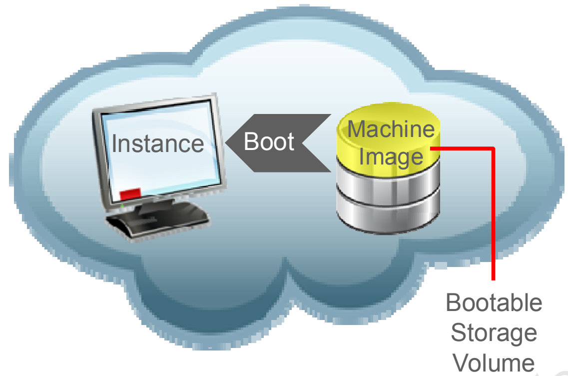
Use a Persistent Boot Disk

Nonpersistent boot disk

- Data is not persistent.
- Changes are lost when the instance is deleted.

Persistent boot disk

- Data is preserved even when the instance is deleted.
- Changes made at operating system-level can be saved and used again later on.



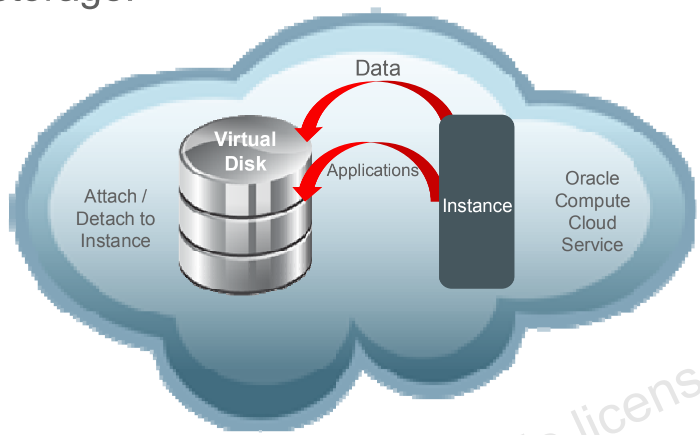
ORACLE

By default, Oracle Compute Cloud Service instances boot from an persistent boot disk. Using a persistent boot disk allows your data to be preserved even when you delete your instance. Any changes that you make at the operating system-level persist, and you can use this boot disk when you re-create the instance or create another instance later on.

If required, you can also boot from a nonpersistent boot disk. If you delete such an instance, the data on the boot disk also gets deleted. If you had made any changes on the boot disk, those changes are lost. Using a nonpersistent boot disk is required when you want to create a snapshot of an instance.

Attach High-Capacity Block Storage to Instances

- Attach persistent block storage:
  - Varying sizes
  - Up to 20 TB
- Store data:
- Store applications:
- Delete instance:
  - Data still intact



You can attach up to 20 TB of block storage to each of your instances for storing data and applications, by creating multiple persistent storage volumes and attaching them to the instances. Even after you delete instances, the data stored in the storage volumes remains intact until you delete the volumes.

Practice Fine-Grained Control over Network Traffic

F E A T U R E S

1 2 3 4 5 6 7 8 9 10

Control traffic between:

- Instance x > Instance y
- Group a of instances > Group b of instances
- Group of instances > external hosts

Also control traffic over:

- Protocols
- Ports

ORACLE

You can control network traffic among individual instances and also between specific groups of instances and external hosts. You can also control traffic to and from instances over specific protocols and ports that you can define.

Reserve and Assign a Persistent Public IP Address

Does your instance need access to/from the Internet?

- Use an auto-generated public IP address.
  - IP address might change if the instance restarts or is deleted and created again.
- Reserve and use a persistent public IP address.
  - Public IP address remains fixed, even if instance restarts or is deleted and created again.

ORACLE

For an instance that requires access to the Internet, you can reserve and use a static public IP address. Each instance can have only one public IP address. When you create an instance, you can select a predefined persistent public IP address, or you can use an auto-generated public IP address. Auto-generated public IP addresses are selected from a pool of public IP addresses and might change if your instance restarts or is deleted and created again. If you need the public IP address of your instance to be static, you can use an IP reservation to specify a fixed public IP address that is used with your instance.

Ensure Secure Access to Instances

F E A T U R E S

1 2 3 4 5 6 7 8 9 10

To ensure secure access to your instances:

- Isolate your instance to prevent unauthorized access
- Specify IP addresses of remote hosts that can be allowed to access instances
- Use an SSH key pair to authenticate access

ORACLE

If you use an Oracle-provided Oracle Linux machine image to create your instance, then you can configure your instance to be accessed securely from remote hosts by using a secure protocol, such as SSH. You can also enable access from only a specific set of external hosts.

Monitor and Manage Your Resources

F	1
E	2
A	3
T	4
U	5
R	6
E	7
S	8
	9
	10

Oracle Compute Cloud Service web console provides:

- An easy-to-use web-based graphical interface
- A list of resources and their status
  - Instances
  - Storage volumes
  - Machine images
  - SSH public keys
  - Network settings
- Details of each object
- The ability to create, update, or delete objects

You can access, administer, and use Oracle Compute Cloud Service through an easy-to-use graphical web console. The console provides a single interface that you can use to monitor and manage all your Oracle Compute Cloud Service resources.

Automate Provisioning and Management Workflows

Using orchestrations, you can:

- Create multiple instances with varied attributes
- Create multiple other objects, such as storage volumes and network settings
- Remove multiple instances or other objects
- Re-create multiple instances or other objects
- Enable high-availability for instances
- Specify dependencies between objects
- Schedule the creation or deletion of a set of objects

You can define all the attributes for multiple, high availability (HA)–enabled virtual machines of varying shapes and machine images in an orchestration. Using the web console, you can then easily create, remove, and reprovision all of the virtual machines and associated resources as required through the orchestration.

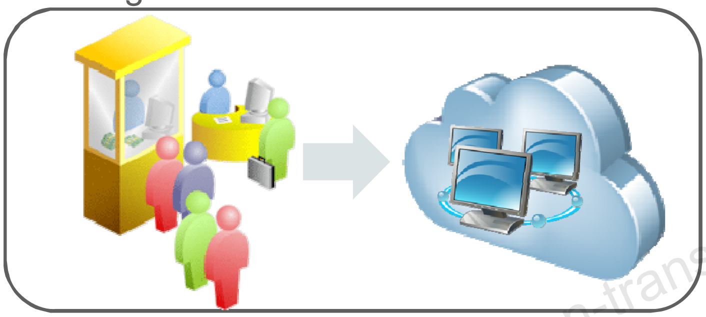
Migrate Applications to the Cloud

F E A T U R E S

1 2 3 4 5 6 7 8 9 10

Oracle Compute Cloud Service:

- Provides high-performance x86 servers
- Enables enterprises to migrate their on-premises applications to virtual machines
- Provides elastic compute, storage, and network capabilities
- Provides predictable performance in the cloud with the Dedicated Compute offering



When enterprises subscribe to Oracle Compute Cloud Service, they can opt for a dedicated environment (called a site) that consists of a large number of high-performance x86 servers reserved for their use. Depending on the configuration subscribed, they get compute power equivalent to 500, 1000, 1500, or 2000 physical cores of a modern Intel Xeon processor with hyperthreading enabled. In a dedicated environment, because they are the only tenant on the site, enterprises can count on predictable performance in the cloud.

Enterprises that have smaller compute requirements can opt for fewer resources, on a site that is shared with other tenants.

In either case, enterprises can migrate their on-premises applications to the virtual machines that they create on the cloud, and take advantage of the elastic compute, storage, and network capabilities that Oracle Compute Cloud Service provides.

Summary of Oracle Compute Cloud Service Features

F E A T U R E S

1 2 3 4 5 6 7 8 9 10

- Create virtual machines (instances) using Oracle-provided or custom machine images.
- Assign processor and memory resources from a range of resource profiles.
- Use persistent boot disks to start your instance.
- Attach high-capacity block storage to your instance.
- Practice fine-grained control over network traffic.
- Reserve a persistent public IP address and assign it to your instance.
- Ensure secure access to your instance.
- Monitor and manage your resources using a web console.
- Automate provisioning and management workflows using orchestrations.
- Migrate on-premise workloads and applications to the cloud.

ORACLE

What Else Can I Do with Compute?

Choose from myriad applications with a variety of Operating Systems.

- Visit Oracle Cloud Marketplace and check out the available apps.
- Browse through the large collection of trusted and innovative apps in many business categories available for free.

Wordpress

actifio

Lamp Stack

django Stack

VIXTERA

Cloud Automation

Ubuntu

drupal

Hundreds More !

ORACLE

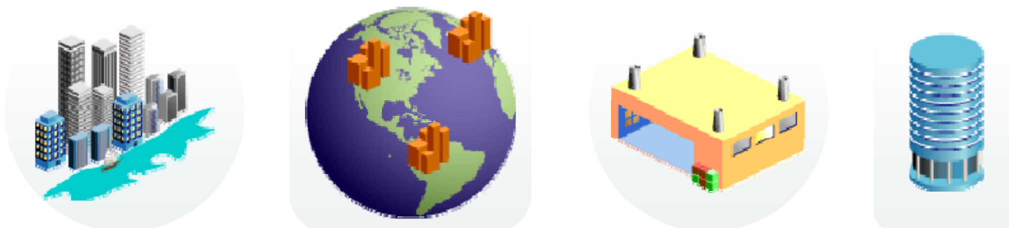
Oracle Cloud Marketplace is an online store—a one-stop shop—selling hundreds of business apps and professional services that complement your existing Oracle Cloud implementation.

All apps and services on the marketplace are offered by approved, registered, and expert partners and developers. Plus, Oracle has vetted, reviewed, and approved each product.

Oracle Cloud Marketplace offers a large collection of trusted and innovative apps in many business categories, including marketing, sales, customer service, social, and talent management. These apps provide enhanced functions and customizations that enable you to set up your Oracle Compute Cloud Service instances and become productive quickly and easily.

Who Uses Oracle Compute Cloud Service?

- Large enterprises that require instant provisioning
- Large enterprises with variable compute requirements
- Small and medium enterprises that want to avoid infrastructure overhead



ORACLE

Examples

- **Online shopping portals:** The shopping portals cater to changing market demands and respond to changing seasons. That establishes the need to plan infrastructure that's elastic and maintains profitability despite ever-changing compute requirements, such as CPUs and storage capacity.
- **Call centers:** These environments typically have high HR churn rates and require instant provisioning and removal of work spaces. Access to elastic compute facility ensures that.
- **Startups:** Startup businesses operate on thin financial resources but can largely benefit from modern infrastructure that offers convenience and edge. Migrating their applications to the cloud provided ease of access and use, and flexible compute environment.
- **Companies on an expansion spree:** Oracle Compute Cloud Service gives the advantage of setting up and tearing down preplanned compute solutions (with the orchestrations) to cater to growing business needs.

Quiz

Which of the following statements are true?

a. Oracle Compute Cloud Service is a vital component of Oracle's IaaS offerings.
b. Oracle Compute Cloud Service can be used to migrate third-party applications to Oracle Cloud.
c. Oracle Compute Cloud Service is for large enterprises and is not for startups.

ORACLE

Quiz

Which of the following can you do while creating an Oracle Compute Cloud Service instance?

a. Attach a persistent boot disk.
b. Select from a large portfolio of CPU and memory resource combinations for the instance.
c. Select an OS.
d. Configure network settings to ensure secure access to the instance even before it is up and running.
e. All the above.

Summary

In this lesson, you should have learned how to:

- Provide an overview of Oracle Compute Cloud Service
- Explain the features of Oracle Compute Cloud Service
- Describe the uses and applications of Oracle Compute Cloud Service
- Discuss using apps at Oracle Cloud Marketplace

Resource: Links

For more information about Oracle Compute Cloud Service, see http://docs.oracle.com/cloud/latest/stcomputecs/index.html

ORACLE

6

Introduction to Storage Concepts

Objectives

After completing this lesson, you should be able to explain:

- The need for storage
- Local storage
- Cloud storage
- Block storage
- Object storage
- Ephemeral storage

What Is Storage?

- Retention of data to retrieve when needed
  - This data can be modified, deleted, updated, and so on.
- Storage options:
  - Local
  - Cloud

ORACLE

- Storage is hugely important in computing, especially now because you generate more content than ever.
- The consumers are able to retrieve, modify, erase, update, and save their data either locally or via the cloud.
  - Saving data locally—personal computer or local network—has been the traditional form for storing data.
  - However, nowadays, cloud storage is gaining popularity due to the increase of mobile devices, wireless technology, application development and new state-of-the-art enterprise computing resources.

What Differentiates Cloud Storage from Local Storage?

- Local Storage
  - Personal: PC stand alone
    - Physical backups, access from one place, storage limitation
  - Business: servers, databases, files, and so on
- Cloud Storage
  - Personal: PC, mobile device, web browser
    - Automatic backups, access from anywhere, unlimited storage
  - Business: servers, databases, files, and so on

- Local storage still remains the traditional form for storing data.
  - At home, users still save their data (such as documents, photos, music, videos, and so on) on their PCs. For users who make backups, most use external drives as their storage device. Although these items are stored locally, there are some storage providers who take user data and upload it to the cloud—for online services.
  - Businesses may use a local network to share, modify, and store data. They store data on their local server, and then perform periodic backups.
- Cloud storage is useful for both individuals and organizations.
  - Users can access data from just about anywhere, and on multiple platforms including mobile devices.
  - It does not matter if the PC's hard drive crashes or a mobile device is lost or damaged. The data can be securely retrieved to a new device by logging in to the cloud storage service.
  - Cloud storage is a great solution for businesses interested in paying for what they use. By subscribing to a cloud storage service, customers pay for the storage capacity that is currently needed. As a customer's business grows, they can increase their storage capacity. Cloud storage also allows interaction with databases as well as making the data available for other computing operations. Cloud storage service providers keep multiple copies of data uploaded by a customer.

Local Storage

- Forms of local storage:
  - Internal hard drive
  - External hard drive
  - Solid State Drive (SSD)
  - USB Drive or flash drive
  - Network Attached Storage (NAS)
  - Optical Drive (CD/DVD)
- Universal facts about local storage devices:
  - Capacity limitation
  - Prone to damage

ORACLE

Local storage is found in many forms including:

- **Internal hard drive:** Found in every computer. Today, many computers have the option to upgrade to a solid state drive for an additional fee.
- **External hard drives:** Used to perform backups or expand storage capacity
- **Solid State Drive (SSD):** Does not have any moving parts; it is able to input/output data faster than traditional hard drives and uses less power
- **USB Drive:** Also known as a flash drive or jump drive. Like the SSD, these are small devices, and are portable and efficient. Unlike the SSD, flash drives are rather inexpensive and very practical for everyday use.
- **Network Attached Storage (NAS):** Is very popular in the workforce. This allows for multiple networked computers to retrieve and save data on storage devices.
- **Optical Drive (CD/DVD):** Are used to store data ranging from basic data files up to music and movies. Soon the optical drive will become obsolete because of revolutionary cloud storage and wireless technology.

The two well-known facts about all local storage devices:

- The user could exceed the storage capacity
- The hard drive will stop working eventually

Cloud Storage

- Main features:
  - Accessible from anywhere
  - Secure
  - Scalable
  - Automatic backups
- Characteristics:
  - Requires Internet connectivity
  - Flexibility to use REST API or other interfaces
  - Options to replicate data to a different location

ORACLE

- Cloud storage accompanied with cloud computing is gaining huge popularity today.
  - Mobile accessibility is a large component of cloud storage.
  - It helps users and organizations (from anywhere) to gain access to data found anywhere in the world.
  - Data is encrypted, keeping it secured when stored accessed.
  - Individual consumers and organizations can go up or down depending on how much storage capacity they need.
  - Cloud storage keeps backups for all the data housed. The backups are distributed throughout different datacenters assuring data security—disaster recovery is no problem.
- Other characteristics include:
  - Internet connectivity is required.
  - You have the flexibility to use REST API or other interfaces.
  - Cloud storage provides the option to replicate data in different locations.

What Are the Different Types of Cloud Storage?

- Public Cloud Storage
- Personal Cloud Storage
- Hybrid Cloud Storage

- **Public Cloud Storage:** Cloud storage resources, maintenance, and management are part of the storage provider's own data center. Customers' data centers and the cloud provider's data center are completely separate.
- **Personal Cloud Storage:** This is intended mostly for individual use. Personal cloud storage is part of the public cloud, except that this service provides mobile friendly features such as content sharing and data synchronization.
- **Hybrid Cloud Storage:** This is a combination of both private and public cloud storage.

So, Which Is Better: Local Storage or Cloud Storage?

- Local Storage
  - **Pros:** Faster, secured
  - **Cons:** Capacity limitation, subject to failure, no automatic backup
- Cloud Storage
  - **Pros:** Accessibility, automatic backups, scalability, synchronization, security, zero maintenance
  - **Cons:** Transfer speed bandwidth-dependent; Internet connectivity required

Both local storage and cloud storage have advantages and disadvantages.

- For local storage, some of the advantages include faster speed and secured connectivity. Some of the disadvantages include limited space; sooner or later the physical disk will fail; and there are no automatic backups.
- The main advantages of cloud storage include being able to access your data from anywhere; automatic backups performed periodically; ability to increase and decrease storage capacity as needed; and that maintenance is done by the cloud provider, not the user. Some disadvantages of cloud storage are that the user must have an Internet connection to access their data, and the transfer speed depends on bandwidth.

What Are the Different Types of Storage?

Main storage types:

- **Block Storage:** Traditional
- **Object Storage:** Trending
- **Ephemeral:** Temporary

ORACLE

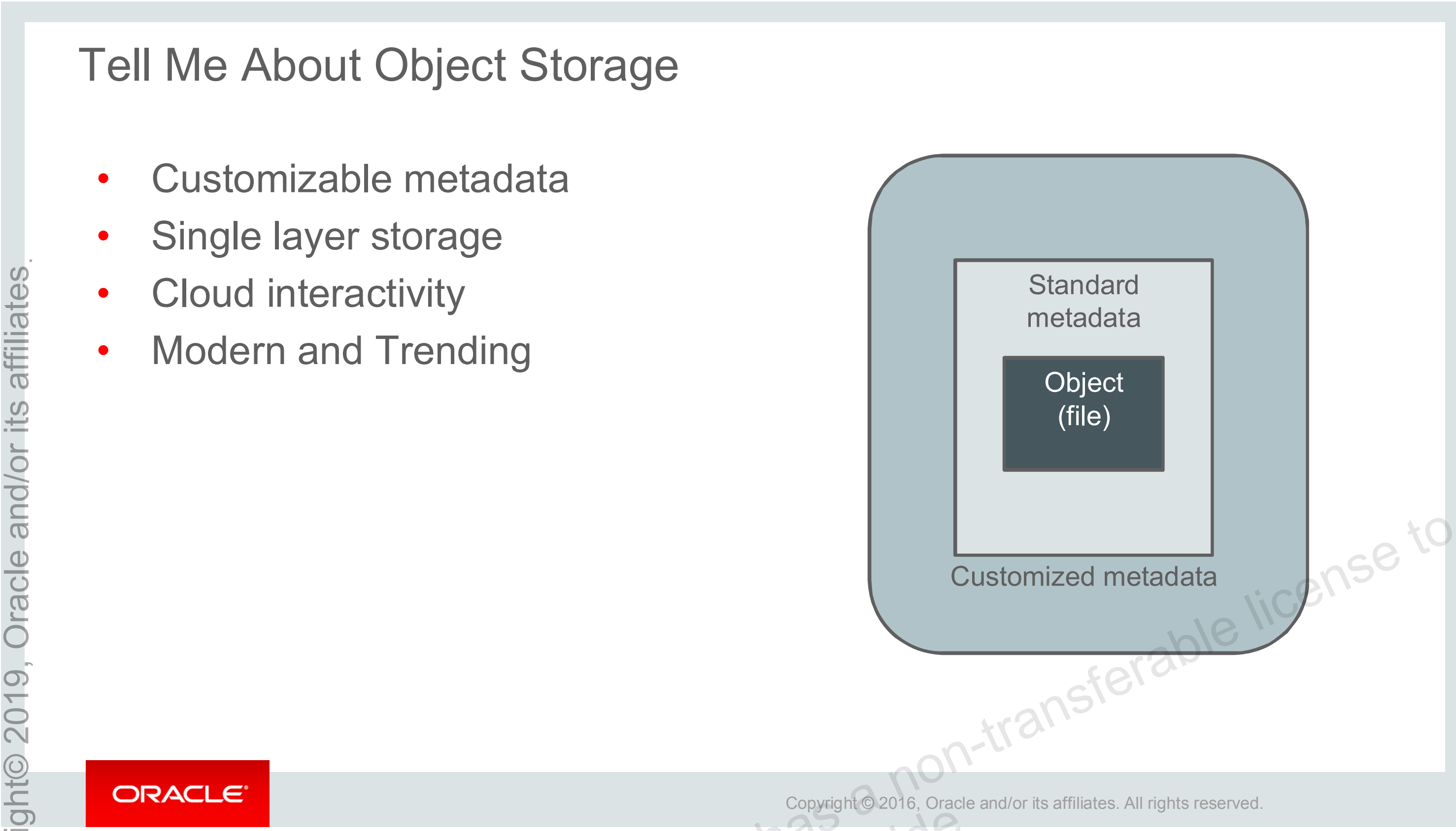
- There are three main types of storage in the physical component.
  - Block storage is the traditional form of storing data.
  - Object storage is gaining popularity due to modern technology and mobile devices.
  - Ephemeral storage exists only temporarily.

Tell Me About Block Storage

- Traditional
  - This storage is in the form of hard drives, optical discs (CDs), and so on.
  - Such physical storage is virtualized in the form of storage volumes and attached to virtual machines.
- File System Structure
- Limited metadata
- Network storage
- Cloud interactivity

ORACLE

- Block storage is a type of data storage used on physical hard drives. Chunks of 1's and 0's are put together to form a file.
- This has been the traditional form of data storage for many years.
- Data is stored and found in the form of file structure. This means that users have to specifically locate the file path and open or execute it.
- Block storage has limited metadata (such as the file name, date created, date modified, date accessed, file size, and so on).
- Large chunks of data can be saved through network storage.
- Cloud storage providers also use block storage on their servers to perform backups and provide cloud services.



- Object storage (or object-based storage) stores and manages data as objects.
- An object is a data file that uses metadata to better describe its characteristics, as well as letting the management system know where this file can be found in the storage pool. If the file is moved, then the location metadata gets updated.
- This metadata is customizable according to the needs of the applications or users.
  - Example: A photo has basic characteristics, such as filename and date created; however, custom metadata allows users or applications to specify additional properties such as the location where the photo was taken, the category, and so on. Then the photos can be retrieved based on location or category information.
- Unlike block storage, object storage is stored in the form of a single layer. The user does not need to specify the file path because its location is part of the metadata.
- This is a very attractive solution in cloud computing. It allows users to upload and access data from computer and mobile devices. The user is able to later retrieve data based on specific searches or filtering data using metadata parameters.

Tell Me About Ephemeral Storage

- Used in cloud computing
  - Virtual Machines
  - Instant accessibility to data

ORACLE

- Ephemeral is defined as *lasting for very short time.*
- Ephemeral storage is mainly used in cloud computing.
- The storage is persistent during the lifetime of a virtual machine. When a virtual machine is terminated, the ephemeral storage disappears as well.
- This is a good option if users need to process huge data files. Hundreds of gigs of original data can be held on ephemeral storage.

Storage: The Big Picture

- Computing ecosystem
- Storage working stand alone
- Storage types team up
- Achieve better results

- Regardless of what activity is being performed, storage is essential.
- Although block storage is the most common form, object storage is quickly becoming popular especially in cloud computing contexts.
- The whole computing ecosystem benefits from one form of storage or another. Businesses may use specific types of storage for specific purposes. Additionally, they may use a combination of storage types to increase performance and improve customer experience.

Quiz

Storage is defined as retention of data to retrieve when needed.

a. True

b. False

Quiz

Which of the following is not part of local storage?

a. Solid State Drive (SSD)
b. Object storage
c. Network attached storage
d. Optical Drive (CD/DVD)
e. None of the above

Quiz

Block storage exists only temporarily and ephemeral is storage in form of 1's and 0's.

a. True

b. False

Quiz

Which of the following storage exists only temporarily?

a. Block storage
b. Object storage
c. Ephemeral storage
d. None of the above

Quiz

Object-based storage is mainly used in:

a. Cloud storage
b. Regular PC storage
c. External hard drives
d. All of the above
e. None of the above

Summary

In this lesson, you should have learned how to explain:

- The need for storage
- Local storage
- Cloud storage
- Block storage
- Object storage
- Ephemeral storage

7

Introduction to Oracle Storage Cloud Service

Objectives

After completing this lesson, you should be able to:

- Discuss object storage on Oracle Cloud
- Explain the basic concepts of object storage
- Explain how object storage on cloud compares with other storage solutions
- Describe how to access object storage on cloud by using the REST API

ORACLE

Oracle Cloud: Overview

- Includes a broad set of industry standards–based, integrated services
- Provides subscription-based access
  - Data storage
- Includes application services
- Provides the following:
  - Service management
  - Hosting
  - Support

ORACLE

- Oracle Cloud includes a broad set of industry standards–based, integrated services that provide customers with subscription-based access to Oracle Platform Services, Application Services, and Social Services, all completely managed, hosted, and supported by Oracle.
- With predictable subscription pricing, Oracle Cloud delivers instant value and productivity for business users, developers, and administrators.
- Oracle Storage Cloud Service is the main focus in this lesson.

So What Exactly Is Object Storage on Oracle Cloud?

- Definition of object storage
  - Components
  - Advantages over traditional storage (block storage)
- Popularity in cloud storage
  - Scalability
  - Customizable attributes and metadata
  - Flexibility
  - Data manipulation
  - Nearly any file type
  - Multiple user accessibility from anywhere

ORACLE

- Object storage is where data is handled as an object, also known as unstructured data. The main differences between object storage and traditional storage (also known as *block storage*), are listed as follows:
  - Stored data contains customized metadata.
  - Data is indexed, allowing for much faster search results.
  - Data can be located by using pointers instead of finding its location based on tracks and sectors on the hard disk (that is, the standard *file system* that we have used for many years).
- This type of storage is used as an essential part of cloud services, in data centers, and it is normally integrated with virtual machines.
- Because object storage allows for additional attributes as part of the "bundle," applications, programs and storage devices are able to better manipulate data.
- Nearly any file type can be stored in the form of object storage. Some popular files include media files (images, videos, music, and photos), documents, PDFs, backups, archives, and so on.
- Multiple users can access the data.

Storage is a fundamental requirement for any enterprise application workload. Traditional storage solutions pose certain scalability, performance, and management challenges that Oracle Storage Cloud Service helps to overcome. Oracle Storage Cloud Service provides a low cost, reliable, secure, and scalable object-storage solution for storing unstructured data and accessing it anytime from anywhere. It is ideal for data backup, archival, and file sharing, and for storing large amounts of unstructured data such as logs, sensor-generated data, and VM images.

How Does Object Storage Work?

- Data stored alongside one another (flat structure)
  - Unique identifiers to locate and retrieve data
  - Infinite storage capacity
- Beneficial when handling large amounts of data
  - Advantages over traditional storage (file system hierarchy)
  - Benefits of searchable metadata
- Storage disks and storage devices

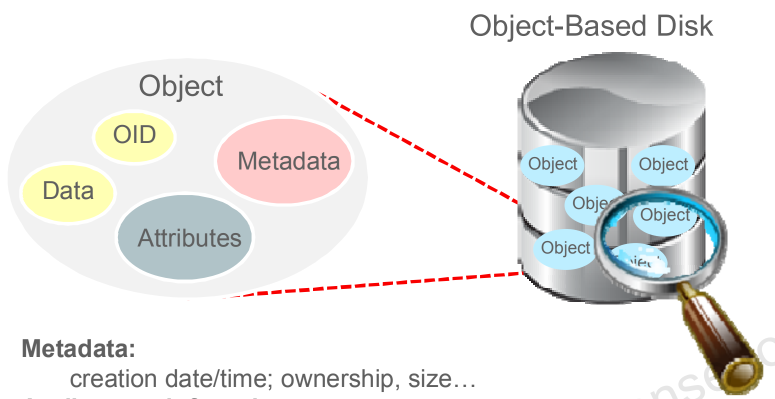
To better understand how object storage works, make a note of the following factors:

- Object storage data is stored alongside one another. That is, all data is stored in the form of a single layer.
- Unlike traditional file systems, with object storage, data is pulled by servers by using unique identifiers to retrieve the desired data. This is particularly helpful for enterprises that interact with large amounts of data because the “large” file is not retrieved until it is found based on its searchable metadata and location.

Object storage is an intelligent evolution of disk drives that can store and serve objects rather than simply place data on tracks and sectors. This task is accomplished by moving low-level storage functions into the storage device and accessing the device through an object interface.

Object Storage Elements

- Object Storage Platform
  – Containers
- Objects
- Disk Volumes
  – Traditional: Block Storage
  – New: Object Storage



Metadata:
creation date/time; ownership, size…
Attributes – inferred:
access patterns, content, indexes…
Attributes – user supplied:
retention, QoS…

ORACLE

- A container is a user-created resource, which can hold an unlimited number of objects, unless you specify a quota for the container. Note that containers cannot be nested.
- When using this form of storage, data is treated as an object. Think of an object as a document file. Users can add additional attributes to each object such as: notes about the file, location where the file was created, compatibility options, and so on.
- Traditional data storage (block storage) does not support additional metadata and attributes. Additionally, the file location must be specified by the user; this way, the operating system calls up that file from the hard drive directly.
- Object storage allows for searchable metadata, automatic indexing, multiple copies/backups of stored data, and the ability to access storage nodes found in different parts of the world.
  - If the storage container is about to reach its capacity limit, a new storage node is created to allow the user to continue adding data.
- A common analogy to better understand object storage is valet parking:
  - Even though you do not know where the car is parked—or if it has been relocated multiple times while you were away—when you are ready to leave, your ticket number is used to trace your car and return it to you. The car is the object; the ticket number is the object's unique identifying number that provides the location of the car; and the valet's parking lot is the container where the vehicles are parked in a flat area.

Other Storage Solutions

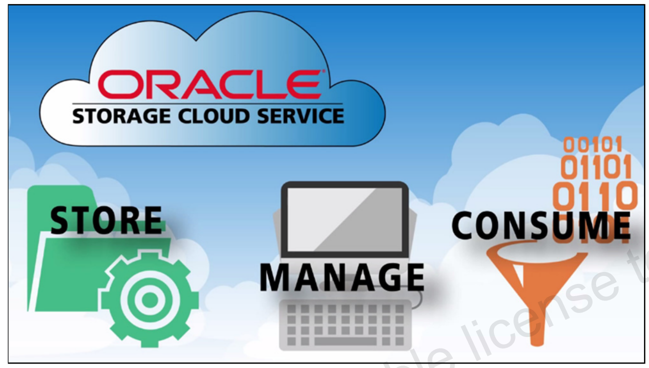
- Object storage has advantages over:
  - Direct-attached storage
  - Network-attached storage (NAS)
  - Block storage

ORACLE

- Other storage solutions include direct-attached storage, network-attached storage, and block storage.
  - With *direct-attached storage*, such as the hard disk drive in a laptop, the operating system that underlies the applications manages data storage, retrieval, and organization through a file system, which is a schema that the operating system uses to organize data on locally attached disks. Direct-attached storage provides convenient, low latency, durable storage. However, because storage capacity is spread between isolated devices, direct-attached storage does not scale well.
  - In *network-attached storage (NAS)*, the storage device is physically separate from the servers that host the applications. To the application hosts, the storage device is available as a network drive. A network file system on the storage device manages data storage, retrieval, and organization. NAS enables applications running on multiple hosts to share storage. It enables centralized management of storage resources and high performance over a local network. But this architecture is feasible only within a limited geographical area, and it offers limited room for scaling. As with direct-attached storage, in NAS as well, applications rely on the underlying operating system and on the network file system of the storage device.
  - *Block storage* enables applications such as online transaction processing (OLTP) databases that have high IOPS (input/output operations per second) requirements to store and retrieve data efficiently, by bypassing the host operating system and interacting directly with the virtual block devices. Chunks of data are stored in blocks, each with an address, but with no other metadata. Applications decide where data is stored, and they retrieve data by calling the appropriate block addresses directly. Block storage optimizes storage for IOPS and block-based access, and provides POSIX-compliant file systems for Oracle Compute Cloud Service instances. It is limited in terms of scalability and does not support the definition of granular metadata for stored data.

What Are the Benefits of Oracle Storage Cloud Service?

- Enterprise solution
  - Cost effective
  - Reliable
  - Scalable
- Public cloud storage solution
  - Secure
  - Elastic
  - Reliable
- Data
  - Enterprise-grade storage capacity
  - Purchase storage now, and buy more later.



- Oracle Storage Cloud Service is a secure, elastic, and on-demand public cloud storage solution.
- It is accessible from anywhere, 24/7, and from any device.
- It provides an easy way to store, manage, and consume large amounts of data.
- It enables users to manage data at a granular level with role-based access control.
- User data is never moved out of the data center without the customer's permission.

What Features Does Oracle Storage Cloud Service Provide?

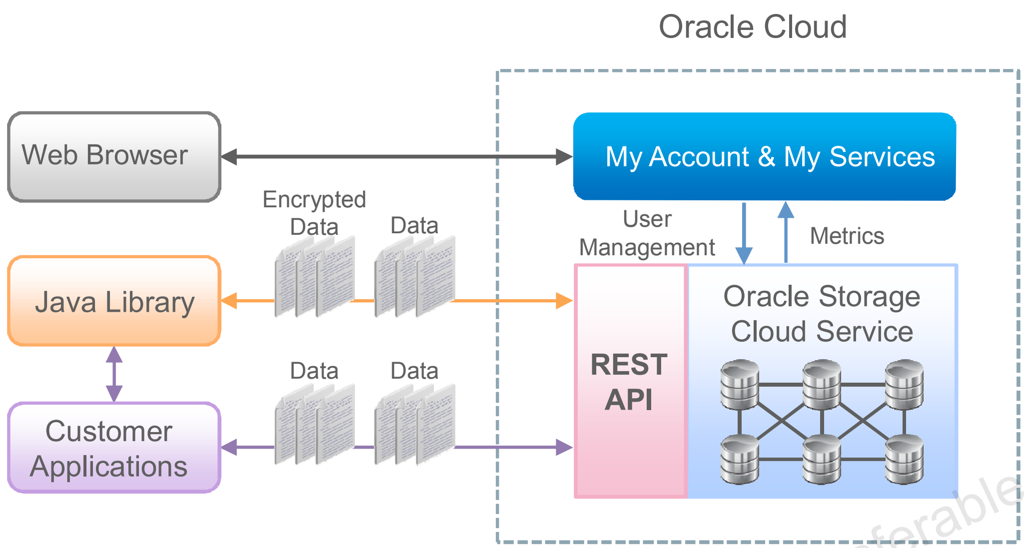
- Object-based storage
- Replication within the data center
- Automatic error detection and healing
- Fine-grained read/write access control to containers
- REST API and Java library interfaces
- Global namespace URL to access the service
- Low-cost data archival

ORACLE

- All objects or containers created in Oracle Storage Cloud Service are replicated to three separate storage nodes in the data center. If one of the three nodes fails, at least two copies of the object or container will continue to be available.
  - Note that by default, data is eventually consistent across the nodes in the data center. When an object or a container is created or modified, the change is not replicated instantaneously to the other two nodes. Until the replication is completed, a container or an object's data may not be consistent across the three nodes. Over time, all changes to all objects or containers are replicated, and the data becomes consistent across the three nodes.
- Object copies are actively scanned for data corruption. If a bad copy is found, it is replaced, automatically, with a new copy.
- Read and write access to an object is controlled via access control lists for its container. Each container can be assigned its own read and write access control lists. By default, access to a container and its objects is private (that is, only the user who created the container can access it), but read access can be made public if required.
- The primary method for accessing Oracle Storage Cloud Service is through a REST web service, which is based on OpenStack Swift. The service can be accessed from anywhere over the Internet, at any time, and from any device.
- A Java library that wraps the REST web service is also available. No special hardware is required to start using the service.

- Regardless of the data center where your service instance is provisioned, you can access Oracle Storage Cloud Service by using a global namespace. Requests sent to the global namespace URL are routed to the data center where your service instance is provisioned.
- In metered accounts, you can create containers of two storage classes, Standard (default) and Archive. You can use Archive containers to store large data sets that you do not need to access frequently, at a fraction of the cost of storing data in Standard containers. Note that to download data stored in Archive containers, you must first restore the objects. The restoration process can take up to four hours depending on the size of the object. A few features, such as dynamic large objects, bulk upload and deletion, and server-side COPY, are not supported for Archive containers. Other specific differences are highlighted in the sections that describe the affected features. Archive containers are ideal for storing data such as email archives, data backups, and digital video masters.

Architectural Overview



- The diagram in the slide presents an architectural overview of Oracle Storage Cloud Service.
- The Oracle Storage Cloud Service architecture is highly available and redundant. It provides support for external access methods, including customer applications, Java SDK, and REST clients.
- When objects are stored in Oracle Storage Cloud Service, data is replicated across three storage nodes in the data center. This replication strategy ensures that the stored object data can survive hardware failure. Note that there can be only one Oracle Storage Cloud Service instance per identity domain.
- In this lesson, we focus only on the REST API.

Oracle Storage Cloud Service - **Replication Policy**

- The service administrator MUST select a replication policy
  - This allows for stored data to be replicated at another data center
- This is required for Oracle Storage Cloud Service for the following subscription types:
  - Commercial IaaS metered
  - Commercial IaaS nonmetered
  - Public sector IaaS nonmetered
- Any user who is assigned the `Storage_Administrator` role can do this task
  - Role name for metered members: `Storage.Storage_Administrator`
  - Role name for nonmetered members: `service-instance-name.Storage_Administrator`

ORACLE

- To be able to use Oracle Storage Cloud Services, a user, with the storage administrator role, must select a replication policy for the service instance.
- Data is written to the primary data center and replicated asynchronously to the secondary data center. The primary and secondary data centers are eventually consistent.
  - If the primary data center is unavailable, read requests to the global namespace URL are routed to the secondary data center. This is known as *failover*. While the primary data center is unavailable, write requests will fail with the 403 – Forbidden error. When the primary data center is available again, all requests to the global namespace URL are routed to the primary data center. This is known as *failback*.
- See Practice 7.1, titled "How To Select a Replication Policy," for instructions on how to select a replication policy.
- **IMPORTANT:** Users will not have access to Oracle Cloud Services if this Replication Policy is not selected.

Interface to Oracle Storage Cloud Service

- REST web service API additions:
  - Centralized identity management across Oracle Cloud
  - Centralized reporting of usage metrics
- No support for some OpenStack Swift features
- REST web service API
  - Over HTTPS only

ORACLE

- Oracle Storage Cloud Service provides REST APIs based on OpenStack Swift. The following major additions have been made:
  - Centralized identity management across Oracle Cloud
  - Centralized reporting of usage metrics
- Oracle Storage Cloud Service does not support the following OpenStack Swift features:
  - Object versioning
  - Static website support
  - Container synchronization
  - Form post
  - Account ACLs
  - Rate limits
- The REST web service API is available only over HTTPS.

How Do I Access Object Storage on Cloud?

- The application or platform must:
  - Completely understand the Hypertext Transfer Protocol (HTTP)
  - Have internet connectivity
- Applications include:
  - cURL
  - Web browsers
- If using a Linux machine, ensure that cURL is installed.
- The following slides describe how to install Cygwin with cURL on a Windows machine

ORACLE

- The REST API can be accessed from any application or programming platform that correctly and completely understands HTTP and has Internet connectivity. The REST API uses advanced facets of HTTP such as secure communication over HTTPS, HTTP headers, and specialized HTTP verbs (`PUT, DELETE`).
- Some applications that meet these requirements are:
  - cURL: cURL is a command-line tool that you can use to invoke REST API calls by sending HTTP requests.
  - Web browsers: Support varies across vendors. Some browser plugins may be needed for full support.
- Many programming platforms (Java, Ruby, Perl, PHP, .NET, and so on) also meet these requirements, although some may require the use of third party libraries for full support. See your programming platform's documentation for guidance.

Installing Cygwin with cURL (1/3)

- The installation tutorial is available at: https://apexapps.oracle.com/pls/apex/f?p=44785:112:::::P112_CONTENT_ID:11571

ORACLE

The next three slides will guide you through the installation process.

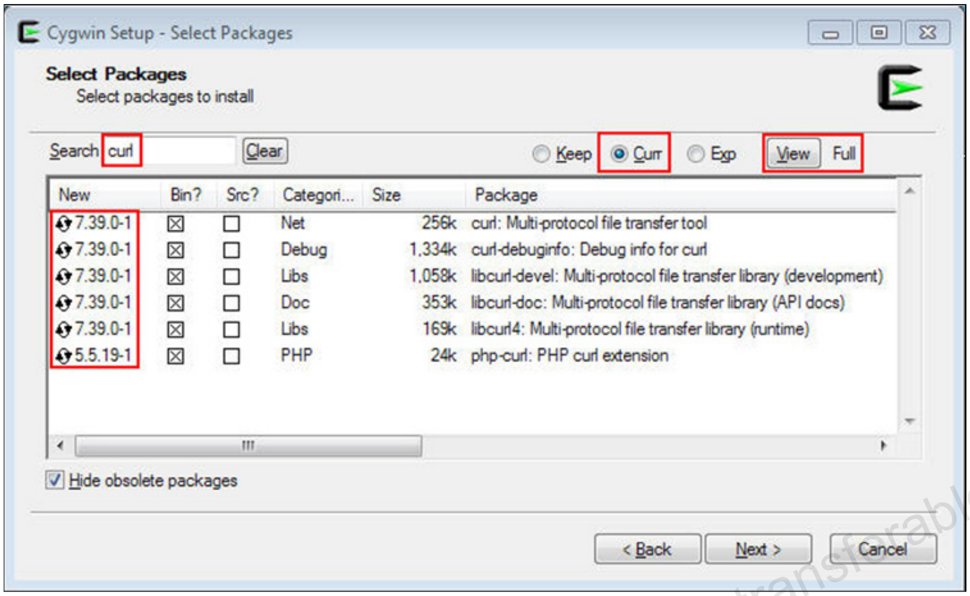
- For more details on the installation process, you may also follow the tutorial online.

Installing Cygwin with cURL (2/3)

1. Download and run the Cygwin installer.
   1. 64-bit: https://cygwin.com/setup-x86_64.exe
   2. 32-bit: http://cygwin.com/setup-x86.exe
2. Follow the prompts by the Cygwin Setup wizard.

- On the Windows system where you want to install Cygwin with cURL, download and run the Cygwin installer:
  - 64-bit: `https://cygwin.com/setup-x86_64.exe`
  - 32-bit: `http://cygwin.com/setup-x86.exe`
- Follow the prompts by the Cygwin Setup wizard.
- You can leave most settings at their default values. Pay specific attention to the following:
  - On the **Choose A Download Source** screen, select "Install from Internet."
  - On the **Select Your Internet Connection** screen, select the appropriate type depending on the Internet connection that you are currently using. For example, if you connect via your company's proxy server, either select the IE5 method or specify the proxy.
  - On the **Choose a download site** screen, select a site from the list, or add your own sites to the list.
  - The **Select Packages** screen displays a list of all available packages and lets you select those that you want to install. By default, only packages in the "Base" category are marked for installation. The Base category does not include tools such as cURL. You should select those explicitly.
  - By default, the packages are grouped by category. Click the View button to toggle to the Full view.
  - Make sure that the Curr radio button remains selected. This ensures that only the most stable version (rather than an experimental version) is selected for each package.
  - In the Search field, enter "*curl*." The curl-related packages are displayed.

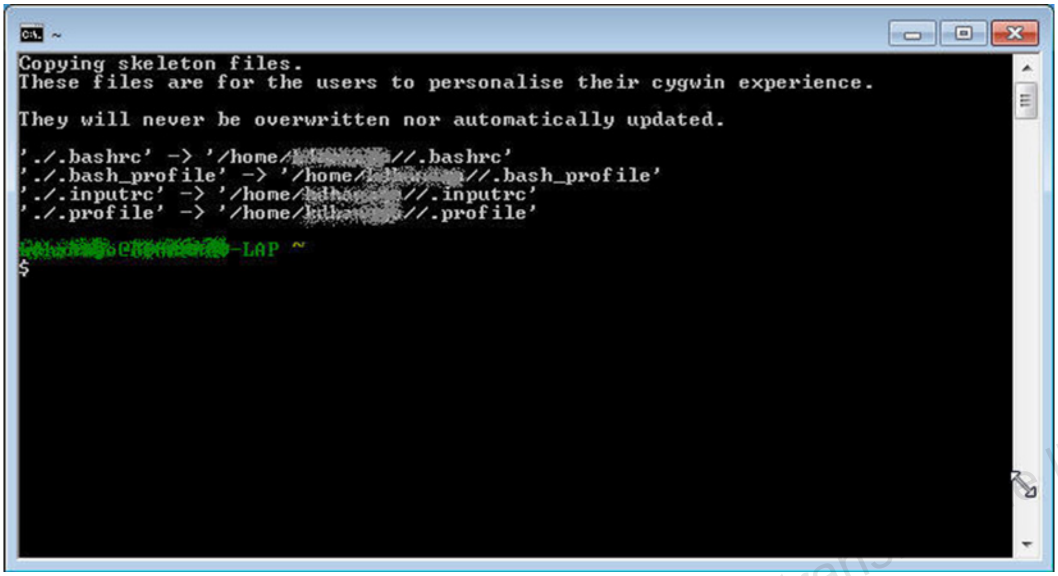
Installing Cygwin with cURL (3/3)



The screenshot in the slide shows the cURL packages that are available for installation.

- Select all the curl packages by clicking the Skip button for each package once. Note how the Skip label changes to show the version number of the selected package. At this point, the Select Packages screen should look like the example in the slide.
- Click Next on the remaining screens of the wizard. The selected packages are downloaded and installed. This may take a while if you have a slow Internet connection.
- On the Installation Status and Create Icons screen, click Finish.

Starting Cygwin



The screenshot in the slide shows the Cygwin command prompt with cURL.

- Go to the directory in which you installed Cygwin (default: `C:\cygwin64`).
- Double-click `Cygwin.bat`. A terminal window opens, as shown in the example in the slide.
  - Verify that cURL was installed by running the command, "`curl`".
  - You should see the following output:

```
curl: Try 'curl --help' or 'curl --manual' for more information
```

- You can now start by using cURL to send HTTP requests.

Tip: Do not delete the `setup*-exe` file. Move it to the `C:\cygwin64` folder. You can use it at any time in the future to update your Cygwin installation, with new or updated packages, for example.

REST URLs: URL for the Account

- REST API endpoint of the service instance
- One of the following formats for the REST API endpoint:
  - Global namespace URL for all customers
  - Data center–specific URL for all customers
- URL in the early releases of Oracle Storage Cloud Service:
  - Metered subscription
  - Nonmetered subscription
- URL formats
- Examples

ORACLE

- The URL for the Oracle Storage Cloud Service REST API endpoint is in one of the following formats:
  - Global namespace URL for all customers: This is the URL that you see in My Services.
  - Data center–specific URL for all customers: https://dataCenterCode.storage.oraclecloud.com/v1/Storage-identityDomainID
- URL in the early releases of Oracle Storage Cloud Service:
  - Metered subscription: https://storage.dataCenterCode.oraclecloud.com/v1/Storage-identityDomainID
  - Nonmetered subscription: https://storage.dataCenterCode.oraclecloud.com/v1/serviceInstanceName-identityDomainID

Note:

  - If your client applications use a URL from an early release of Oracle Storage Cloud Service, you can update the applications to use the global namespace URL.
- In these URL formats:
  - `identityDomainID` is the identity domain in which the service instance is provisioned
  - `dataCenterCode` is the identifier of the data center in which the service instance is provisioned. For example, `dataCenterCode` is us2 for the data center in Chicago, Illinois, U.S.A. and `us6` for Ashburn, Virginia, U.S.A.
  - `serviceInstanceName` is the customer-specified name of the service instance

Examples

For example, for a service instance named `myStorage2` that is provisioned in the `myIdentity3` domain ID in the `us2` data center, the REST API endpoint URLs would be:

- Global namespace URL for all customers: https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3
  - The foo part in this example would be different for each account.
- Data center–specific URLs for all customers: https://us2.storage.oraclecloud.com/v1/Storage-myIdentity3
- URL in the early releases of Oracle Storage Cloud Service:
  - Metered subscription: https://storage.us2.oraclecloud.com/v1/Storage-myIdentity3
  - Nonmetered subscription: https://storage.us2.oraclecloud.com/v1/myStorage2-myIdentity3

REST URLs: URL for Containers and Objects

- URLs for Containers
  - Resources within an account
- URLs for Objects
  - Resources within containers
  - Data center–specific URL for all customers

ORACLE

- URLs for Containers
  - Containers are resources within an account.
  - Given the sample global namespace REST API endpoint URL in the slide, the URL for a container named *myContainer4* would be *https://foo.storage.oraclecloud.com/v1/Storage–myIdentity3/myContainer4*.
- URLs for Objects
  - Objects are resources within containers.
  - Given the sample global namespace REST API endpoint URL in the slide, the URL for an object named *myObject5* in the *myContainer4* container would be https://foo.storage.oraclecloud.com/v1/Storage–myIdentity3/myContainer4/myObject5.

Note

- These URLs are valid for any object in a *Standard* or *Archive* container. To restore an object in an *Archive* container and to track the progress of the restoration, the URL for an object named *myObject5* in the *myArchiveContainer5* container would be *https://foo.storage.oraclecloud.com/v0/Storage–myIdentity3/myArchiveContainer5/myObject5*. Note the *v0* API version in the URL.

Quiz

We already know that object storage is used in cloud computing. Beside this exception, object storage is essentially the same as the *file system* storage (what we have used in PCs for decades now).

a. True

b. False

Quiz

How is object storage stored?

a. Same as the file system architecture
b. File-path form, where the user must specify where on the disk the item is located
c. Alongside one another, in the form of a single layer
d. All of the above
e. None of the above

Quiz

A container is ________.

a. The primary disk on the server
b. What holds an unlimited number of objects
c. The same as a node
d. All of the above
e. None of the above

Quiz

When using object storage, data is treated as objects, giving us the advantage of:

a. Customizing metadata
b. Getting faster results because the metadata is already indexed
c. Filtering search results based on metadata, regardless of where this data is stored
d. All of the above
e. None of the above

Summary

In this lesson, you should have learned how to:

- Discuss object storage on Oracle Cloud
- Explain the basic concepts of object storage
- Explain how object storage on the cloud compares with other storage solutions
- Describe how to access object storage on the cloud by using the REST API

ORACLE

8

Oracle Compute Cloud Service Instances

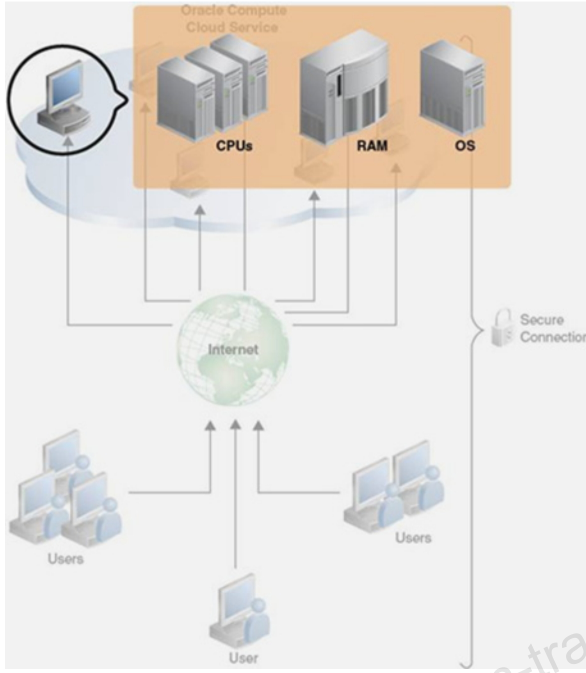
ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe the features and uses of Oracle Compute Cloud Service instances
- Explain how SSH keys are used
- Generate an SSH key pair and upload the public key
- Use the Create Instance wizard to create an instance

What Is an Instance?



In Oracle Compute Cloud Service, an instance is a virtual machine that runs a specific operating system and has the CPU and memory resources that you specify.

Oracle Compute Cloud Service provides a number of machine images that you can use to create an instance. A machine image is a template of a virtual hard disk of a specific size with an installed operating system. For example, you can select a machine image that will create an instance running the Oracle Linux operating system.

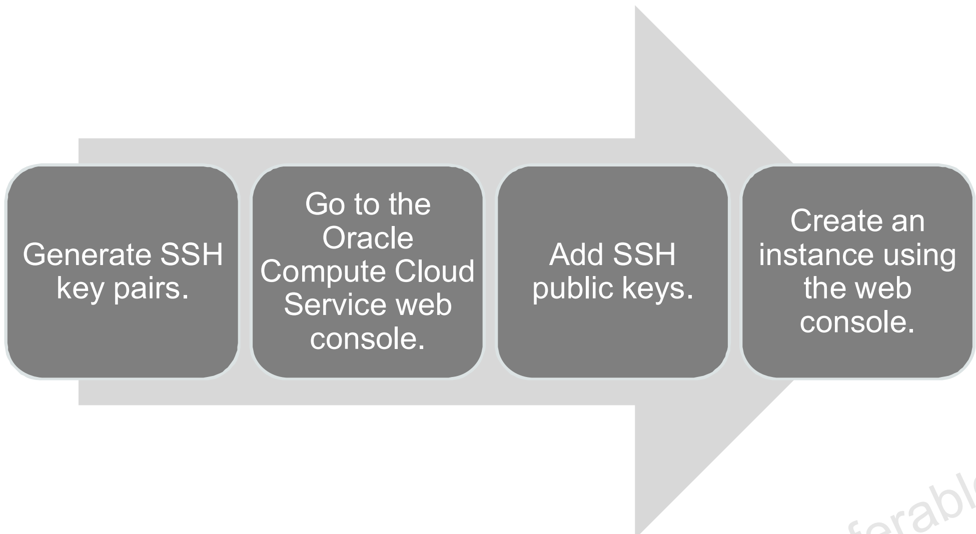
Oracle Compute Cloud Service also provides a number of predefined shapes that you can use. A shape is a carefully designed combination of processor and memory limits that specifies the number of CPUs and the amount of RAM to be allocated to an instance. When you select a shape, your instance is created with the corresponding number of Oracle Compute Units (OCPUs).

An OCPU provides CPU capacity equivalent to one physical core of an Intel Xeon processor with hyper threading enabled. Each OCPU corresponds to two hardware execution threads, known as vCPUs.

You can create instances by using the Oracle Compute Cloud Service web console. While creating your instance, you can specify network access configurations to enable you to log in to your instance remotely. You can also add persistent block storage to your instance.

When your instance is ready, you can log in to your instance remotely from any physical location. You can then configure your instance to add users, mount block storage devices and write data, install applications, and so on. You can run any applications on your instance that are supported on the OS, and that you have licenses for.

How Do I Create an Instance from the Web Console?



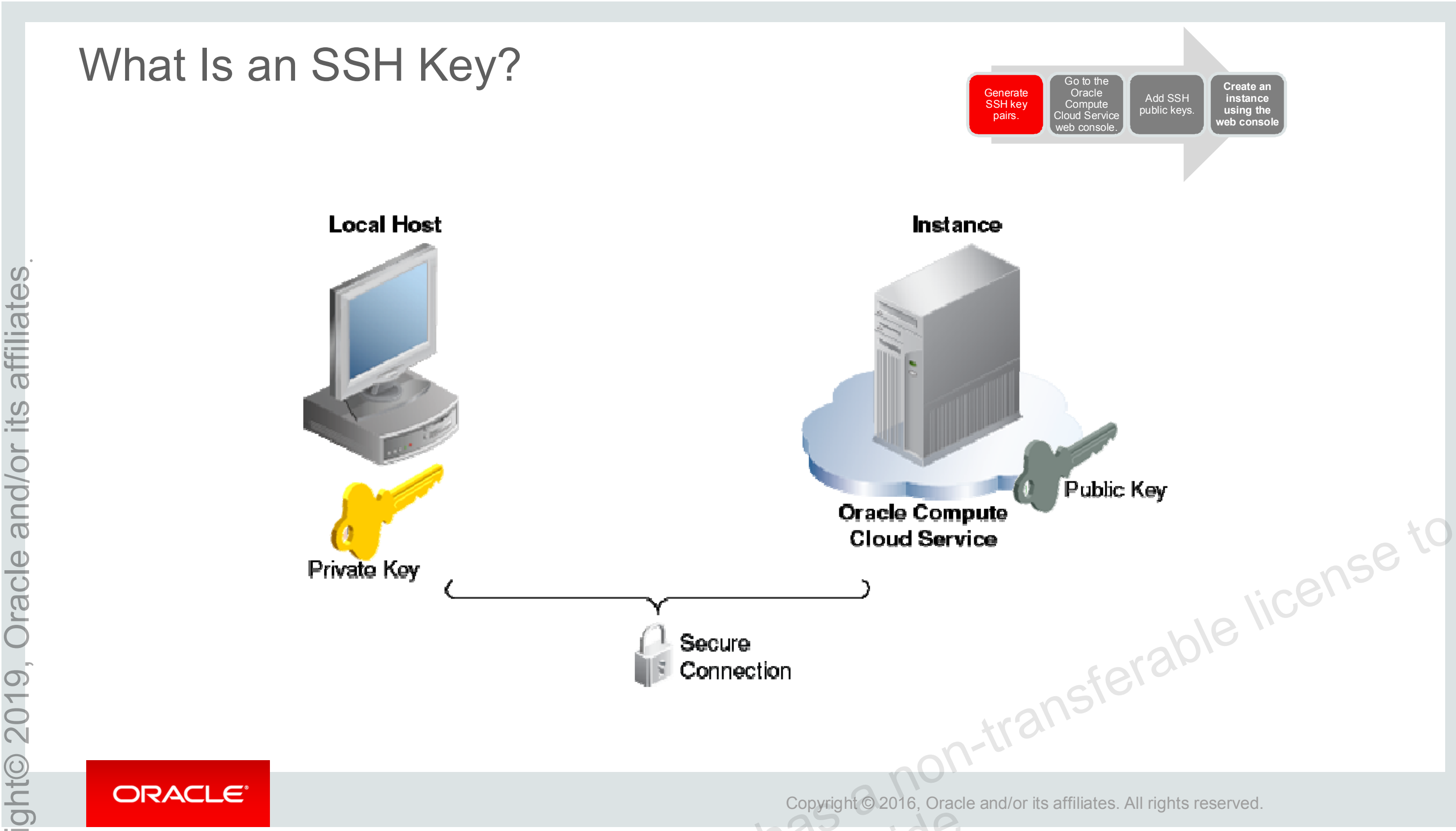
ORACLE

There are many ways that you can create instances and the associated networking and storage resources.

For example, you can create the required storage volumes first, and then create the instances to which the storage volumes should be attached. Alternatively, you can create instances first, and then create and attach the required storage volumes to the instances. Similarly, you can create security lists first, and then create instances and add them to the security lists; or you can create the instances first, and then create security lists and add instances to them.

Let's start with the simplest workflow.

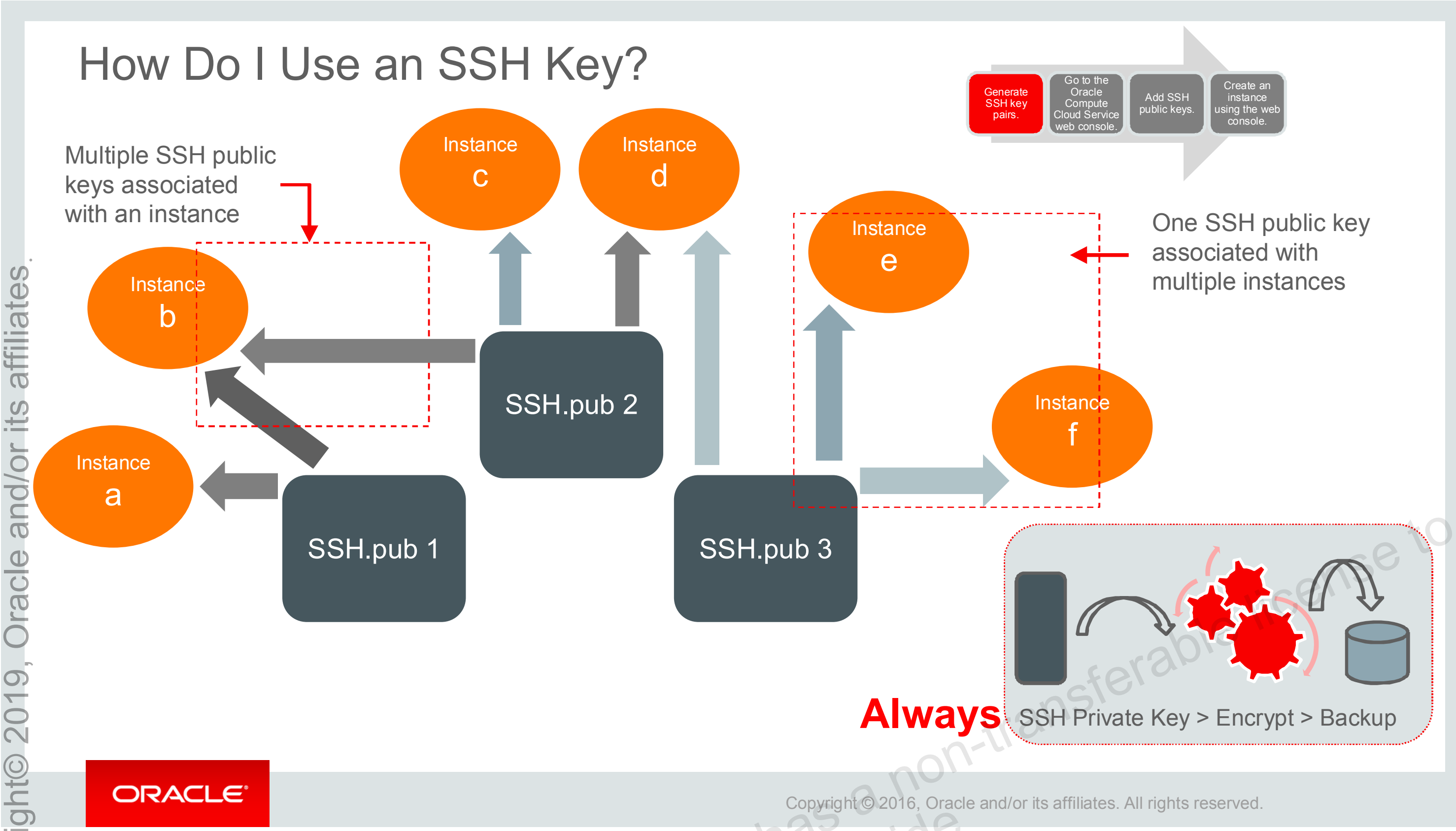
1. Generate SSH key pairs.
2. Go to the Oracle Compute Cloud Service web console.
3. Add your SSH public keys.
4. Create an instance by using the web console.



Using SSH Keys

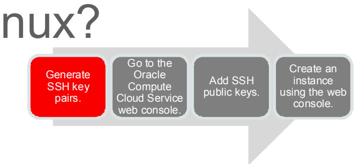
- SSH stands for Secure Shell.
- SSH is a cryptographic network protocol that uses two keys, a public key and a private key, to provide secure communication between two computers.
- Before creating instances, generate at least one SSH key pair and ensure that the private key is available on each host that you will use to access instances.
- The public key must be stored on the instance that you want to access.
- When you log in to the instance by using SSH, you must provide the private key that matches a public key associated with the instance.

How Do I Use an SSH Key?



- You can associate multiple SSH public keys with an instance.
- You can associate an SSH public key with multiple instances.
- Always back up an encrypted copy of your private SSH keys, and keep the keys secure.

How Do I Generate an SSH Key Pair on Linux?



1. Run the ssh-keygen command
   `ssh-keygen -t rsa`
2. Enter the required path and select file name
3. The command prompts you to enter a passphrase

The command generates an SSH key pair

Private key: `example_filename`

Public key : `example_filename.pub`

ORACLE

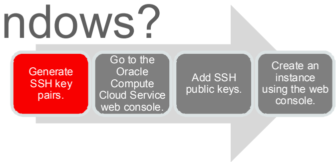
- Run the ssh-keygen command.
  - You can use the -t option to specify the type of key to create.
  - To create an RSA key, run: ssh-keygen -t rsa
  - You can use the -b option to specify the length (bit size) of the key, as shown in the following example:
  - ssh-keygen -b 2048 -t rsa
- The command prompts you to enter the path to the file in which you want to save the key.
  - A default path and file name are suggested in parentheses.
  - For example: /home/user_name/.ssh/id_rsa.
  - Enter the required path and select file name and then press Enter.
- The command prompts you to enter a passphrase.
  - When prompted, enter the passphrase again to confirm it.

The command generates an SSH key pair consisting of a public key and a private key, and saves them in the specified path. The file name of the public key is created automatically by appending .pub to the name of the private key file. For example, if the file name of the SSH private key is id_rsa, the file name of the public key would be id_rsa.pub.

Make a note of

- The file names of private and public keys.
- The path of private and public keys.
- The passphrase.

How Do I Generate an SSH Key Pair on Windows?



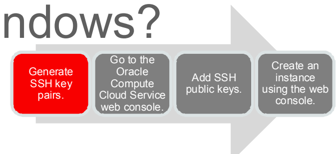
Install PuTTY

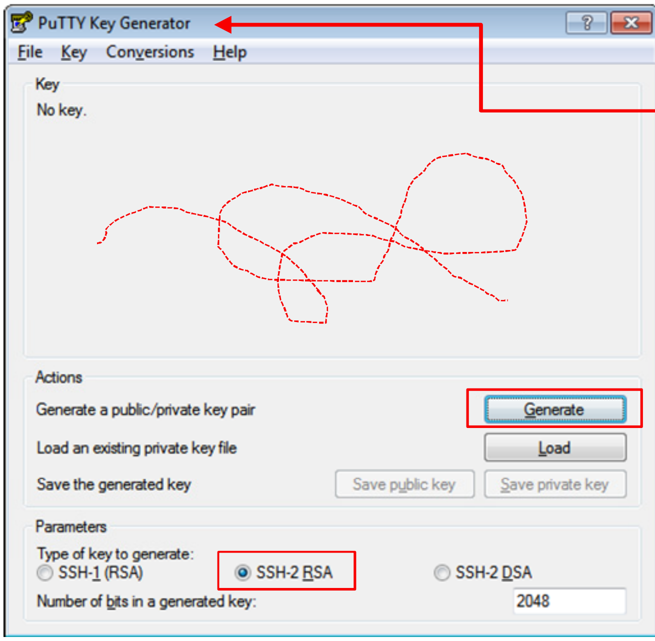
1. Reach the main download site http://www.putty.org
2. Locate `putty.zip`, download it.
3. Unzip the package and install.
4. On your computer, check at the path `C:\Program Files(x86)\PuTTY` for `putty.exe` and `puttygen.exe`, the two utilities we will use in the practices.

Now generate the SSH key pair

ORACLE

How Do I Generate an SSH Key Pair on Windows?





PuTTY Key Generator for Windows

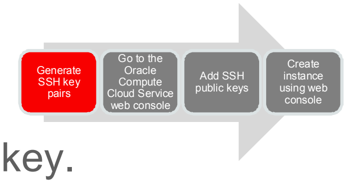
Move your cursor around the blank area to generate randomness.

To generate an SSH key pair by using the PuTTY Key Generator, perform the following steps:

1. Find puttygen.exe in the PuTTY folder on your computer, for example, C:\Program Files (x86)\PuTTY.
2. Double-click puttygen.exe to open it.
3. Accept the default key type, SSH-2 RSA. SSH-2 is the most recent version of the SSH protocol (and it is incompatible with SSH-1). RSA and DSA are algorithms for computing digital signatures.
4. Set the “Number of bits in a generated key” to 2048 bits, if it is not already set to that value. This sets the size of your key and thus the security level. A minimum of 2048 bits is recommended for SSH-2 RSA.
5. Click Generate.
6. Move your cursor around the blank area to generate randomness.

Note: The dotted red line in the image is for illustration purposes only. It does not appear in the generator pane as you move the cursor.

Saving the SSH Private Key on Windows

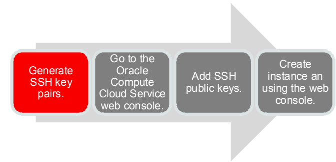


1. The generated key appears under the Public key.
2. The key comment is the name of the key.
3. Password-protect your key, and enter and confirm a key passphrase.
4. Save the private key in PPK format.
5. Save the key in OpenSSH format.
6. Give it the same name as the PPK format. Use the `.ssh` extension.

Saving a Private Key of the SSH Key Pair Using the PuTTY Key Generator on Windows

1. The generated key appears under the Public key for pasting into the OpenSSH authorized_keys file.
2. The key comment is the name of the key that you will use to identify it. You can keep the generated key comment or create your own.
3. Enter a Key passphrase and enter it again for Confirm passphrase.
4. Save the private key of the key pair in PuTTY's Private Key (PPK) format.
5. To save the key in OpenSSH format, open the Conversions menu and select Export SSH key.
6. Give it the same name as the key that you saved in PPK format in the previous step. You can also use any extension (or no extension), but let's use .ssh, to make it clear what format it is.

Saving the SSH Public Key on Windows



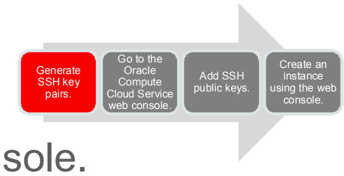
1. Select all the characters under the Public key.
2. Right-click and select Copy from the shortcut menu.
3. Open a text editor and paste the characters.
4. Save the key by using the same root name. Add a `.pub` extension.

ORACLE

Saving a Public Key of the SSH Key Pair Using the PuTTY Key Generator on Windows

1. In the PuTTY Key Generator, select all the characters under "Public key for pasting into OpenSSH authorized_keys file."
2. Right-click somewhere in the selected text and select Copy from the context menu.
3. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.
4. Save the key as a text file, using the same root name as you used for the private key. Add a .pub extension. You can give it any extension you want, but .pub is a useful convention to indicate that this is a public key.

I’ve Generated the SSH Key Pair. What’s Next?



1. Go to the Oracle Compute Cloud Service console.
2. Start the Create Instance wizard.

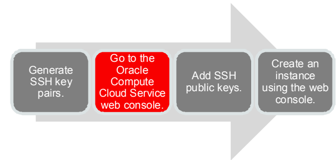
Go to the Oracle Compute Cloud Service console:

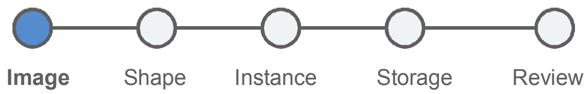
- Sign in to the Oracle Cloud My Services application at https://cloud.oracle.com/sign-in. The Oracle Cloud My Services Dashboard page appears.
- Click near the upper-left corner of the page. The CLOUD SERVICES menu is displayed.
- Select Oracle Compute Cloud Service. The Oracle Compute Cloud Service console is displayed.
- Click Create Instance to start the Create Instance wizard.

When you create an instance using the Create Instance wizard, one or more orchestrations are created automatically to manage the instance and its associated resources. For example, if you use the Create Instance wizard to create an instance and attach a storage volume to it, then two separate orchestrations are created, one for the instance and the other for the storage volume. A master orchestration is also created and both orchestrations are nested in the master orchestration.

You’ll learn more about creating and using orchestrations in Lesson 12, Oracle Compute Cloud Service Orchestrations.

Create Instance Wizard - Image Page





- Select `OL-6.6-20GB-x11-RD` or a similar Oracle-provided Oracle Linux image.
- The latest image list entry is selected by default.
- The image specifies the OS and disk size of the instance.
- Click the button to go to the next page.

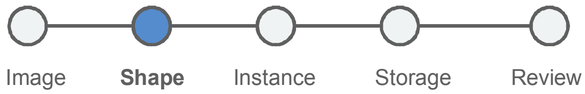
ORACLE

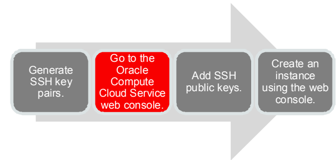
On the Image page of the Create Instance wizard, select `OL-6.6-20GB-x11-RD` or a similar Oracle-provided Oracle Linux image.

If there are multiple entries in an image list, the latest entry is selected by default. You can select an earlier entry from the drop-down list.

After selecting an image, you can click Review and Create to accept the default settings and create your instance. For now, let's see how you can use the other pages in the Create Instance wizard to customize the instance.

Create Instance Wizard - Shape Page

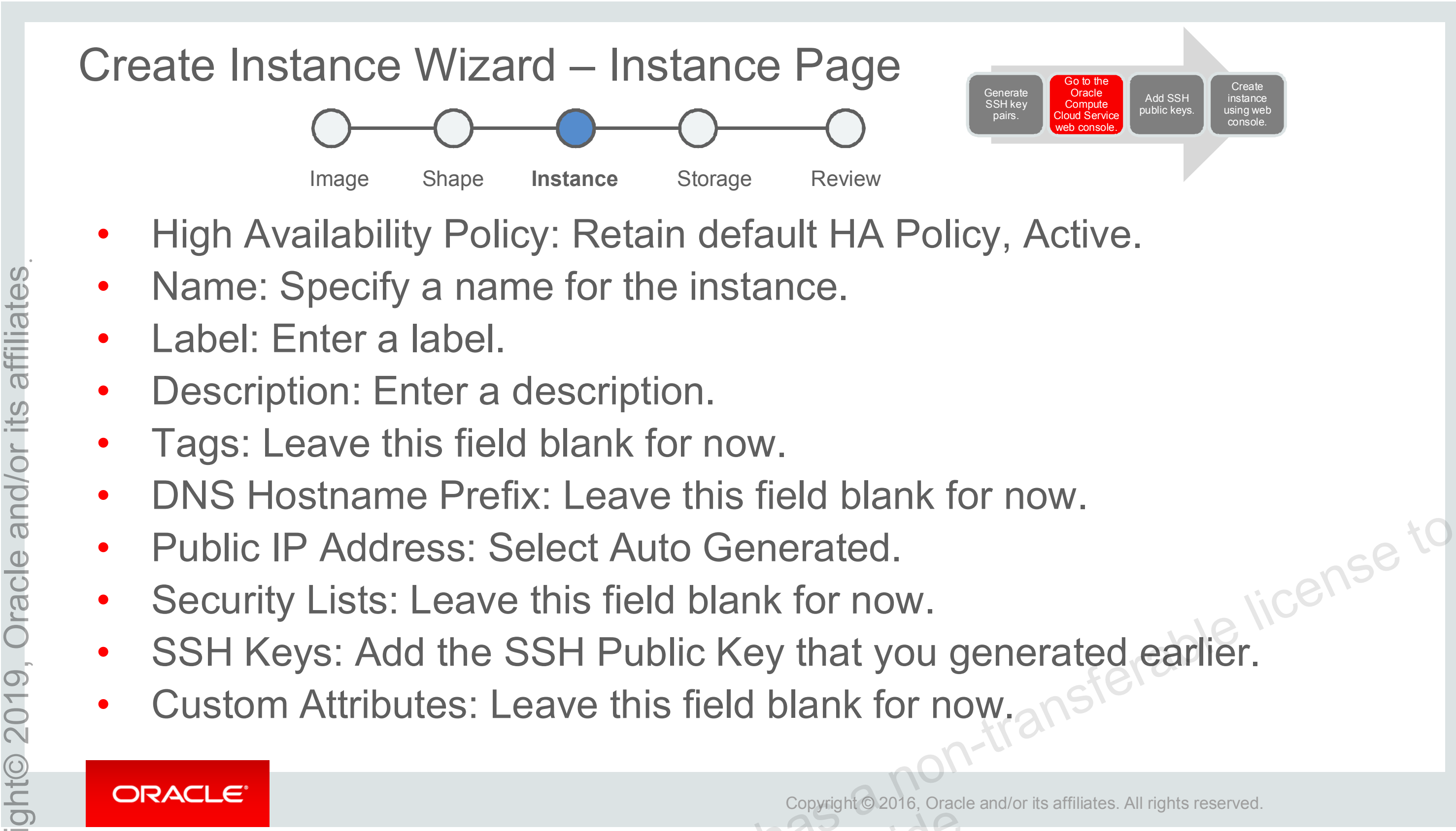




- Select the OC3 shape.
- The shape specifies the OCPU and memory resources to be allocated to the instance.
- Click the button to go to the next page.

ORACLE

On the Shape page, select the shape that you want to use. The shape specifies the OCPU and memory resources to be allocated to the instance.



On the Instance page, select or enter the following information:

Select a High Availability Policy.

- If you retain the default HA policy, Active, then the orchestration monitors the status of your instance and if your instance crashes, the orchestration recreates it automatically.
- If you specify the HA policy Monitor, then the orchestration monitors the status of your instance. If your instance crashes, the status of the orchestration gets updated to Error. However, your instance isn't re-created automatically.
- If you specify the HA policy None, then the orchestration doesn't monitor the status of your instance.

Enter a name for the instance.

Note that the full name of an instance consists of several parts. If you specify a name in the Create Instance wizard, the full name of the instance would be in the format: /Compute identity_domain/user/name_you_specify/id. If you do not specify a name in the wizard, the full name would be in the format: /Compute-identity_domain/user/id. In either case, id is an autogenerated ID.

Enter a label for the instance.

- Enter a label that is meaningful and that you can use to identify the instance easily later. Try to assign a unique label for each instance.

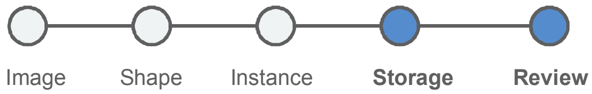
Associate a Public IP Address with the instance.

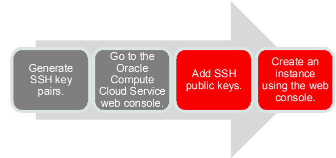
- If you want to connect to this instance over the Internet, then select either Auto Generated or Persistent Public IP Reservation.
- If you select an autogenerated public IP address, the IP address persists while the instance is running, but will change if you delete the instance and create it again later.
- To associate a permanent public IP address with the instance, select Persistent Public IP Reservation.
- To create an IP Reservation, click Create IP Reservation. Enter a name for the IP reservation and then click Create.
- If you don't want your instance to be accessed over the Internet, then you don't need to associate a public IP address with it. In the Public IP Address list, select None.

Specify the SSH keys that you want to associate with this instance.

- To add a new SSH public key, click Add SSH Public Key. Enter a name for the SSH public key, paste the public key in the Value field, and then click Add. The
- SSH public key is added and appears in the list of SSH keys that you want to associate with the instance.

Create Instance Wizard – Storage Page





- Retain the default boot disk that is used to boot the instance.
- Review Page
  - Verify information.
- Click Create
  - Instance is created.

ORACLE

On the Storage page, you can attach data storage volumes and bootable storage volumes to your instance, if required.

The Storage page shows the default boot disk that is used to boot your instance. You can attach existing storage volumes to your instance, if required, or create storage volumes and attach them to the instance. You can also choose to remove the default boot disk. If you do so and if you don't specify a persistent storage volume as the boot drive, a nonpersistent boot disk is used to boot the instance.

For now, retain the default settings on Storage page.

Can I Create an Instance Using an App from Oracle Cloud Marketplace?

1. Sign in to Oracle Cloud Marketplace at https://cloud.oracle.com/marketplace/product/compute.
2. Browse the available apps or search for an app.
3. Select an App.
4. Click Get App.
5. Accept the terms of use.
6. Select your account.
7. Submit the request.
8. After your request is confirmed, to create an instance, click Start Compute Console. The Create Instance wizard starts.

ORACLE

To use an app from Oracle Cloud Marketplace:

1. Go to Oracle Cloud Marketplace at https://cloud.oracle.com/marketplace/product/compute.
2. Search for the app that you want to use, or browse the available apps.
3. Select the app that you want to use by clicking it. You're directed to a page with more information about the selected app.
4. Click Get App.
5. Accept the terms of use and click Next.
   - If you see a message asking you to enable permission settings by clicking Preferences in your Oracle Compute Cloud Service account, follow the instructions to enable the setting. Then return to Oracle Cloud Marketplace and click Get App for your image again.
6. Select your account from the drop-down list.
7. Review the information on the Review screen and click Submit Request.
8. On the Confirmation screen, after your request is confirmed, to create an instance right away, click Start Compute Console. The Create Instance wizard starts.

Now you can follow the steps described earlier in this lesson to create an instance using the app that you selected.

After an app is added to an account, any user in that account (or identity domain) can use the app to create an instance. Just select the app from the list of images while creating your instance or while creating a bootable storage volume.

How Do I Know If My Instance Is Running?

1. Go to the web console.
2. The Instances tab lists your instances.
3. Check that your instance is listed with the status Running.
4. From the menu, select View to see details of your instance.

Monitoring Instances

After creating instances in Oracle Compute Cloud Service, you can view a list of your instances and get details of each instance.

1. Go to the Oracle Compute Cloud Service console and click the Instances tab.
2. The Instances page shows a list of instances, along with information about each instance.
   - **Tip:** You can filter the list of instances according to their category or status. To list instances with a specific status (such as running, error, or stopped), click the Show menu and select the appropriate filter. To view instances of a specific category (such as PaaS, IaaS, or personal), click the Category menu and select the appropriate filter.
3. To view detailed information about an instance, go to the instance that you want to view. From the menu, select View.
4. The instance details page shows all details related to the selected instance, such as the public and private IP addresses, and the storage volumes, security lists, and SSH keys associated with it. You can add or remove storage volumes and security lists from this page.

You can also check the status of the orchestration on the Orchestrations tab. When your instance is running, the orchestration status is Ready.

Instance Life Cycle

Preparing
- Oracle Compute Cloud Service allocates resources.

Initializing
- Oracle Compute Cloud Service installs the image.

Running
- Instance is running.
- You can log in to it.
- You can attach or detach storage volumes and security lists.
- You can restart it or delete it.

ORACLE

An Oracle Compute Cloud Service instance can have one of the following statuses:

- When you create an instance, the initial status is Preparing. Oracle Compute Cloud Service allocates resources and prepares to create the instance.
- While the specified image is being installed, the state changes to Initializing.
- After the image is installed and the instance starts, the status changes to Running.

When an instance is in the Running status, you can log in to it and start using it. You can also attach or detach storage volumes and security lists using the web console. If required, you can restart your instance. When you are done with the instance, you can delete it.

At times, an instance can have the Error status. If this happens, see the error message to understand why the error occurred.

How Can I Log in to My Instance from Linux?

1. Make note of:
   - The public IP address of the instance
   - The path and file name of the private SSH key on your computer
2. Use SSH to log in to your instance as default opc user using the command:

```
ssh opc@ip_address –i private_key
```

3. Key in the passphrase when prompted.

You can log in as the default user.

Note: At this point you will not be able to log in to your instance as you need to configure network settings. This will be explained in a later lesson.

ORACLE

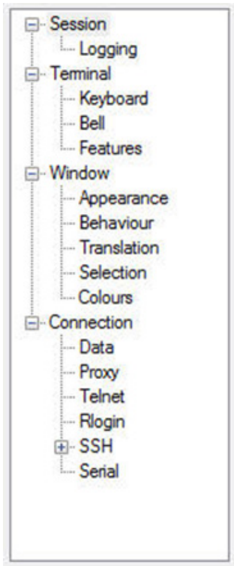
- Use SSH to log in to your instance as the default user, opc, by using the following command:

```
ssh opc@ip_address –i private_key
```

  In this command, ip_address is the public IP address of the instance, and private_key is the full path and file name of the file that contains the private key corresponding to the public key associated with the instance that you want to access.
- When prompted, key in the passphrase.

When you're logged in as the default user, `opc`, use the `sudo` command to run administrative tasks.

How Can I Log in to My Instance from Windows?



1. Open PuTTY.
2. Click Session > Hostname (or IP Address) > enter the public IP address of your instance.
3. Click Connection > expand SSH > click Auth > select the Private SSH Key File.
4. Click Connection > Data > Auto-login username field> enter the default username, `opc`.
5. Click Open.

ORACLE

Accessing an Instance as the opc User

1. Run PuTTY.
2. In the Category pane, click Session and in the Hostname (or IP Address) field, `enter the public IP address of your instance.`
3. In the Category pane, go to Connection, expand SSH, and click Auth. In the Private key file for authentication field, select the Private SSH Key File.
4. In the Category pane, go to Connection and click Data. In the Auto-login username field, enter the default username, `opc`.
5. Click Open.
6. When the terminal opens, key in the passphrase that was used to password-protect the private key.

What's on My Instance?

Your instance has the following default installations or configurations:

- Packages
- A default user
- Remote access
- Disk partitions
- Yum repositories
- Multiple languages

ORACLE

When you use an Oracle-provided Oracle Linux machine image to create your instance, your instance comes with the following packages and configuration:

Packages

- Development tools: Expect Java OpenJDK, GCC suite, GNU utilities, Perl, Ruby, Python, and so on.
- Basic X11 desktop
- Remote X11 access with VNC
- Xterm client
- Security and auditing with OpenSCAP and AIDE
- Integration with name services such as OpenLDAP, Kerberos, and NIS
- System administration tools
- Firefox and Elinks web browsers
- EMACs and vim editors

Users

A user named `opc` is created automatically. The `opc` user has `sudo` privileges and is configured for remote access over the SSH v2 protocol by using RSA keys. The SSH public keys that you specify while creating instances are added to the `/home/opc/.ssh/authorized_keys` file.

Note that `root` login is disabled.

Remote Access

Access to the instance is permitted only over the SSH v2 protocol. All other remote access services are disabled.

Disk Layout

- /boot: 500 MB
- swap: 4 GB
- / (root): Remainder

Oracle Linux Repositories Enabled for Yum Configuration

public_ol6_latest

public_ol6_UEK_latest

public_ol6_UEKR3_latest

Language Support

Arabic

Brazilian Portuguese

Chinese - Simplified

Chinese - Traditional

Czech

Danish

Dutch

Finnish

French

German

Greek

Hebrew

Hungarian

Italian

Japanese

Korean

Norwegian

Polish

Portuguese - Brazilian

Romanian

Russian

Slovak

Spanish

Swedish

Thai

Turkish

How Do I Stop, Restart, and Delete My Instance?

Restarting Your Instance:

1. Go to the Oracle Compute Cloud Service console.
2. On the Instances page, identify the instance that you want to delete.
3. From the menu, select Reboot.

Deleting Your Instance:

1. Go to the Oracle Compute Cloud Service console.
2. On the Instances page, identify the instance that you want to delete.
3. Click the Orchestrations tab.
4. Go to the orchestration that controls the instance that you want to delete. From the menu, select Stop.

ORACLE

Restarting an Instance

1. Go to the Oracle Compute Cloud Service console.
2. On the Instances page, identify the instance that you want to delete.
3. From the menu, select Reboot.

Deleting an Instance

Caution: When you delete an instance that uses a non-persistent boot disk, any changes you may have made to the boot disk after the instance was created are lost.

Note: Any storage volumes that are attached to an instance are detached (but not deleted) when you delete the instance. You must unmount attached storage volumes before deleting an instance.

To delete an instance, perform the following steps:

1. Go to the Oracle Compute Cloud Service console.
2. On the Instances page, identify the instance that you want to delete.
3. Click the Orchestrations tab.
4. Go to the orchestration that controls the instance that you want to delete. From the menu, select Stop.

Quiz

What is an instance? (Select all that apply.)

a. A virtual machine with its own hard disk, CPU, and memory, and with the OS installed on the hard disk

b. A virtual machine configured with your choice of CPU, memory, and OS, and accessed remotely over cloud

c. A virtual machine that runs a specific OS and has CPU, memory, and storage resources available from the cloud and that can be accessed remotely over the cloud

Quiz

Identify the correct sequence of steps for creating an instance.

a. Sign in to the Oracle Compute Cloud Services web console > Create an instance > Generate an SSH key pair > Add the SSH public keys to the instance.

b. Generate an SSH key pair > Use the SSH key pair to sign in to the Oracle Compute Cloud Services web console > Add the SSH public key to Oracle Compute Cloud Service > Create an instance.

c. Generate an SSH key pair > Sign in to the Oracle Compute Cloud Services web console > Add the SSH public keys > Create an instance.

ORACLE

Quiz

Which of the following can you install on your instance?

a. Any licensed software or applications
b. Operating System
c. Firmware
d. All of the above

Summary

In this lesson, you should have learned how to:

- Describe the features and uses of Oracle Compute Cloud Service instances
- Explain how SSH keys are used
- Generate an SSH key pair and upload the public key
- Use the Create Instance wizard to create an instance

Resource: Links

For more information about Oracle Compute Cloud Service, visit http://docs.oracle.com/cloud/latest/stcomputecs/index.html.

ORACLE

9

Oracle Compute Cloud Service Network Settings

ORACLE

Objectives

After completing this lesson, you should be able to:

- Explain networking concepts
- Create security lists and security rules to enable access to your instance
- Disable access to your instance

How Do I Control Access to My Instance?

When you've created an instance, you can enable SSH access to your instance.

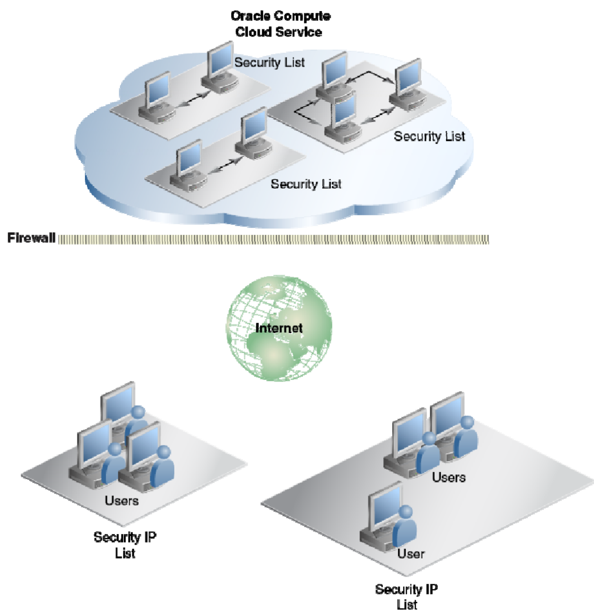
After creating an instance, you can enable or disable access to or from your instance by configuring network settings from the Oracle Compute Cloud Service web console.

You can implement fine-grained control over network access to your Oracle Compute Cloud Service instances, both from other instances as well as from external hosts over the public Internet.

When you've created an instance, you can enable SSH access to the instance. This SSH access is permitted only from hosts that have the SSH private key corresponding to the SSH public key stored on the instance. By default, your instance cannot be accessed from any other external host using any other protocol.

If you want to enable or disable access to your instance from other Oracle Compute Cloud Service instances or from other external hosts over the public Internet, you can use the web console to create security lists and configure network security rules.

How Do Security Lists Work?



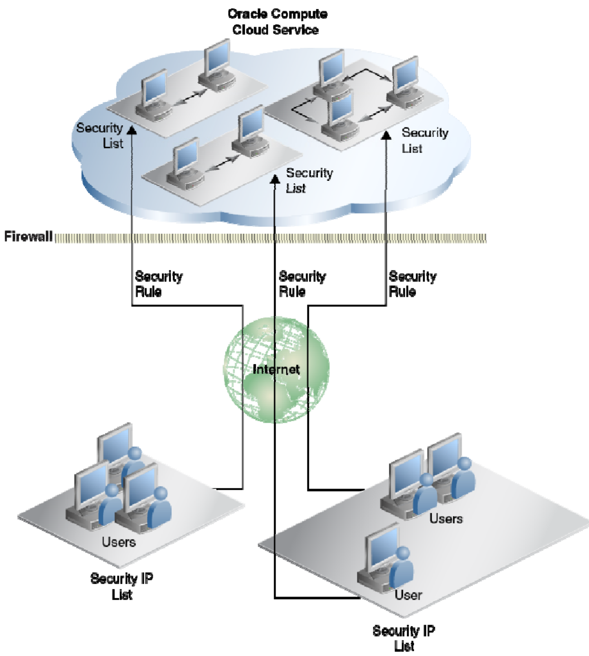
A security list is:

- A grouping of instances
- They can communicate with each other
- They cannot receive traffic from outside the security list

When you create an Oracle Compute Cloud Service instance, your instance is effectively protected by a firewall that controls the traffic to each instance.

If you want your instances to be able to communicate with each other, you can create a security list. A security list is a grouping of instances. When you add an instance to a security list, the instance can communicate with all the other instances in the same security list. However, instances in a security list by default cannot receive traffic from instances or external hosts outside their security list.

How Do Security Rules Work?



Security rules:

- Enable or disable access to instances in a security list
- From other security lists or from external hosts specified in security IP lists
- Over a specified protocol and port

ORACLE

Security rules allow you to enable or disable access to instances in a security list. Each security rule defines a specific source, a destination, and a protocol-port combination over which communication is allowed. If you want instances in two different security lists to be able to communicate with each other, you can create a security rule with one security list as the source and the other security list as the destination.

If you want your instance to be accessible to a specified set of external hosts, you can list the IP addresses of those hosts in a security IP list.

You can then create a security rule that specifies the security IP list as the source. This enables traffic from the specified set of external hosts to access the instances in your security list.

What Should I Know About Security Lists?

- Security lists define inbound and outbound policies.
  - The inbound policy controls the flow of traffic into the security list.
  - The outbound policy controls the flow of traffic out of the security list.
- By default, the inbound policy is deny and the outbound policy is permit.
- When you add an instance to a security list, the inbound and outbound policies of the security list are applicable to that instance.
- If no security rules are defined for a security list, then by default, instances in that security list cannot receive traffic from hosts outside the security list.
- When you remove an instance from a security list, the instance can no longer communicate with other instances in that security list, and it is no longer subject to the security rules defined for that security list.

ORACLE

Security lists define inbound and outbound policies to control the flow of traffic to and from instances.

The inbound policy

The inbound policy controls the flow of traffic into the security list. For example, if the inbound policy is set to permit, packets from all sources using any port or protocol are permitted to the instances in the security list. The default setting for this policy is deny. To control the flow of traffic to the instances in a security list, ensure that the inbound policy is set to deny, and then define security rules to allow only traffic from specified sources to access your instances using specified ports and protocols.

The outbound policy

The outbound policy controls the flow of traffic out of the security list. For example, if the outbound policy is set to deny, packets cannot flow out of the security list. The default setting for this policy is permit. To prevent instances in a security list from communicating with hosts outside the security list, set the outbound policy to deny.

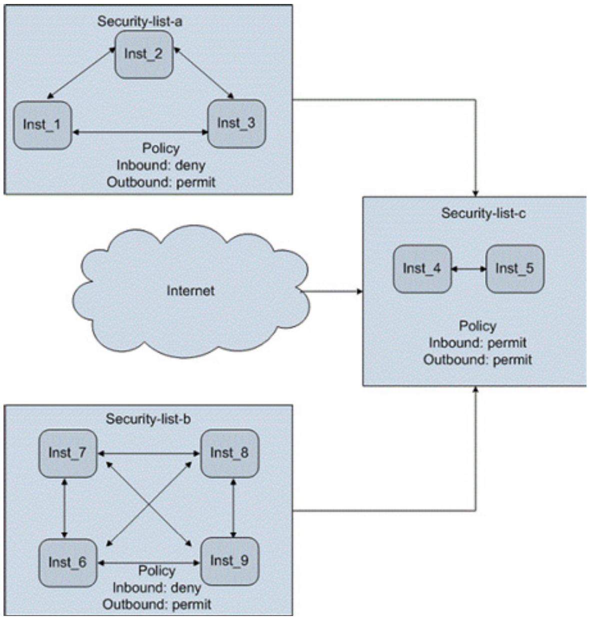
If you retain the default settings of a security list, the instances in that security list cannot receive inbound traffic from hosts outside the security list. You can override this default setting by creating security rules that use this security list as a destination.

A security list can be a source or destination in multiple security rules.

If no security rules are defined for a security list, then by default, instances in that security list cannot receive traffic from hosts outside the security list. However, instances in the security list can still access other instances in the same security list.

When you remove an instance from a security list, the instance can no longer communicate with other instances in that security list, and traffic to and from that instance is no longer controlled by the security rules defined for that security list.

Security Lists: An Example



- Security-list-a cannot receive inbound traffic
- Security-list-b cannot receive inbound traffic
- Security-list-c has inbound policy permit, so it can receive inbound traffic from Security-list-a and Security-list-b, as well as from the public Internet.

ORACLE

This illustration shows three security lists. Security-list-a and Security-list-b have the default inbound and outbound policies. Because the default inbound policy is deny, instances in these two security lists cannot be accessed from hosts outside their own security list. Security-list-c, however, has its inbound policy specified as permit. Instances in this security list can receive inbound traffic from Security-list-a and Security-list-b, as well as from the public Internet.

If an instance is added to multiple security lists that have different policies, then the most restrictive policy is applicable to the instance. For example, in this example, Inst_4 is in Security-list-c, which has the inbound policy permit. If you are to add Inst_4 to Security-list-b as well (inbound policy deny), then the effective inbound policy for Inst_4 would be deny. Inst_4 would still be able to communicate with Inst_5 in Security-list-c, as well as with Inst_6, Inst_7, Inst_8, and Inst_9 in Security-list-b. However, it would not be able to receive traffic from instances in Security-list-a or from external hosts.

What Should I Know About Security Rules?

Security rules consist of a source, a destination, and a security application.

- The source specifies the origin of inbound traffic. You can select either a security list or a security IP list as the source.
- The destination specifies the targeted end point of traffic. Usually, you specify a security list that you have already created as the destination in a security rule. However, if you have created a security list with the outbound policy deny, then you can specify that security list as the source and a security IP list as the destination in a security rule.
- A security application is a protocol-port mapping. When you specify a security application in a security rule, traffic is enabled over the specified protocol and port.

ORACLE

A security rule consists of a source, a destination, and a security application.

Source

The source in a security rule specifies the origin of inbound traffic. You can select either a security list that you have already created, or a predefined security IP list as the source. If you want your instances to receive traffic from any host over the public Internet, then select the predefined public-internet security IP list.

Destination

The destination in a security rule specifies the targeted end point of traffic. Usually, you specify a security list that you have already created as the destination in a security rule. However, if you have created a security list with the outbound policy deny, then you can specify that security list as the source and a security IP list as the destination in a security rule. This allows you to restrict traffic from your instances to a specific set of external hosts, identified by their IP addresses.

Security Application

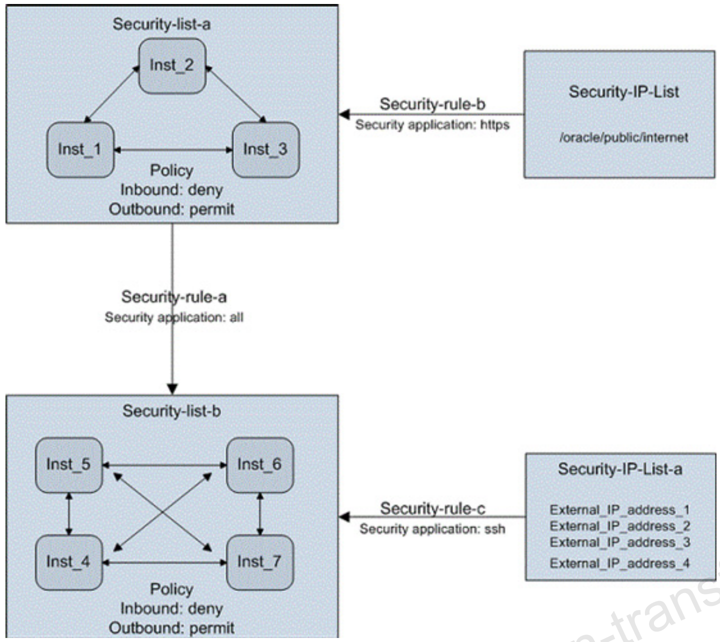
A security application is a protocol-port mapping. Several commonly-used protocols and their default ports are provided as predefined security applications. You can select one of these, or create a security application of your own.

When you specify a security application in a security rule, traffic is enabled over the specified protocol and port. If you want to enable traffic over all protocols and ports, select All.

A security rule acts only on a security list policy that is set to deny. If a security list has its inbound policy set to permit, then you do not need to define security rules to permit traffic to instances in that security list.

You can enable or disable a security rule at any time.

Security Rules: An Example



ORACLE

This illustration shows how you can use security lists and security rules to restrict traffic between your instances and control access to them.

This diagram shows the following communication paths:

- Instances in Security-list-a can send traffic to instances in Security-list-b over any protocol, as defined by Security-rule-a.
- Instances in Security-list-a can receive HTTPS traffic from any host on the public internet, as defined by Security-rule-b.
- Instances in Security-list-b can receive traffic over SSH from any of the IP addresses specified in Security-IP-list-a, as defined by Security-rule-c.

Is That How SSH Access to My Instance Works?

To enable SSH access to your instance:

- Associate a public IP address with your instance
- Add your instance to a security list
- Create a security rule to allow SSH access to the specified security list

To enable SSH access to your instance, add your instance to a security list and create a security rule which allows SSH access. You must also specify a public IP address for your instance.

Remember that instances in a security list can communicate with each other over all protocols and ports.

So, when you add your instance to a security list, if other instances also belong to the same security list, all those instances can communicate with each other.

Also, if you create a security rule with a security list as the source or destination, the security rule applies to all instances in that security list. So if you want to enable SSH access for one instance but not for other instances in the same security list, then create a separate security list for SSH access and add only instances that you want to access over SSH to the new security list.

So, Does My Instance Need a Public IP Address?

If you want to enable access to your instance over the public Internet, you must associate a public IP address with your instance.

1. Go to the Oracle Compute Cloud Service console.
2. Click the Network tab.
3. Click Create IP Reservation.
4. Enter a name for the IP reservation.
5. In the **For Instance** field, select the instance that the IP address must be attached with.
6. Click Create.

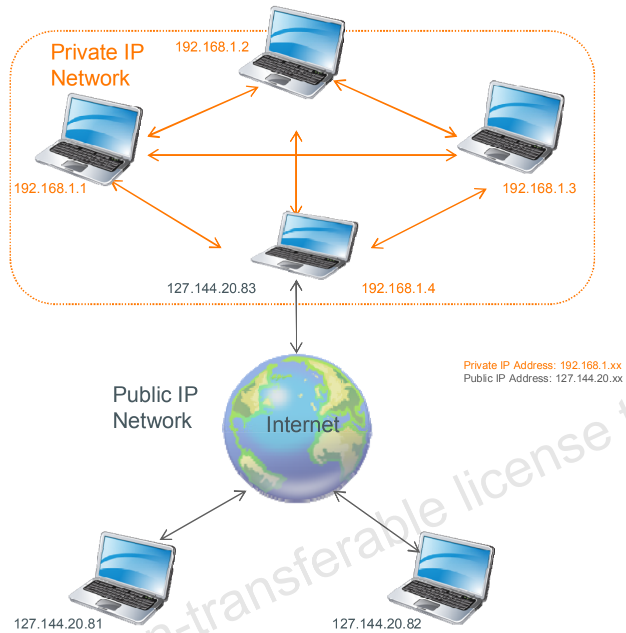
ORACLE

An IP reservation is a public IP address that you can attach to an Oracle Compute Cloud Service instance that requires access to or from the Internet. You can create an IP reservation and associate it with an instance to enable access to the instance from the public Internet. You can do this either while creating an instance, or later on, when your instance is running.

- Go to the web console.
- Click the Network tab.
- Click the IP Reservations tab in the left pane.
- Click Create IP Reservation.
- Enter a name for the IP reservation.
- If your instance is running, then in the For Instance field, you can select the instance that the IP address must be attached with. Alternatively, you can create the IP reservation now without attaching it to any instance, and attach it later, while creating your instance.
- Click Create.

Does My Instance Need a Private IP Address?

- Each instance has a private IP address associated with it.
- When an instance is created, its private IP address is assigned dynamically from a range of private IP addresses.
- You can access an instance from another instance using its private IP address



- Each Oracle Compute Cloud Service instance has a private IP address automatically associated with it.
- When an instance is created, its private IP address is assigned from a range of private IP addresses.
- Private IP addresses aren't constant. When an instance is restarted, its private IP address might change.
- Private IP addresses can be repeated across several private networks, because they are not visible on the public networks.
- If, for reasons of data privacy, you don't want an instance to be accessible over the public Internet, you can opt not to associate a public IP address with that instance. You can then use the private IP address of the instance to access that instance from other instances in the same Oracle Compute Cloud Service account.
- Instances in the same security list also use private IP addresses to access each other. So instances in the same security list can communicate with each other even if they don't have public IP addresses.

I Get It. Can I Configure My Network Settings Now?

To configure your network settings:

1. Create a security list.
2. Add your instance to the security list.
3. Create a security IP list or use the predefined public-internet security IP list.
4. Create a security application or identify a predefined security application.
5. Create a security rule.

How Do I Create a Security List?

Security list > Add instance > Security IP list > Security application > Security rule

1. Go to the Oracle Compute Cloud Service console.
2. Click the Network tab.
3. Click the Security Lists tab in the left pane.
4. Click Create Security List.
5. Enter or select the required details:
   – Name
   – Description
   – Inbound policy
   – Outbound policy
6. Click Create.

ORACLE

To create a security list:

1. Go to the Oracle Compute Cloud Service console.
2. Click the Network tab.
3. Click the Security Lists tab in the left pane.
4. Click Create Security List.
5. Enter or select the required details:
   - **Name:** Enter a name that enables you to identify the purpose of this security rule
   - **Description:** Enter an appropriate description
   - **Inbound policy and Outbound policy:** Select the required inbound and outbound policies. Accept the defaults for now
6. Click Create.

How Do I Add My Instance to a Security List?

Security list > **Add instance** > Security IP list > Security application > Security rule

1. Go to the web console.
2. Click the Instances tab.
3. Go to your instance and from the menu icon, select View.
4. On the instance details page, click Add to Security List.
5. Select the required security list and click Attach.

ORACLE

To add your instance to a security list:

1. Go to the web console.
2. Click the Instances tab.
3. Go to your instance and, from the menu icon, select View.
4. On the instance details page, click Add to Security List.
5. Select the required security list and click Attach.

How Do I Create a Security IP List?

Security list > Add instance > **Security IP list** > Security application > Security rule

1. Go to the web console.
2. Click the Network tab.
3. Click the Security IP Lists tab in the left pane.
4. Click Create Security IP List.
5. Enter the following details:
   – Name
   – IP List
   – Description
6. Click Create.

ORACLE

To create a security IP list:

1. Go to the web console.
2. Click the Network tab.
3. Click the Security IP Lists tab in the left pane.
4. Click Create Security IP List.
5. Enter the following details:
   - **Name:** Enter a name for the security IP list.
   - **IP List:** Enter a comma-separated list of the subnets (in CIDR format) or IPv4 addresses for which you want to create the security IP list. For example, to create a security IP list containing the IP addresses 203.0.113.1 and 203.0.113.2, enter one of the following in the IP List field: **203.0.113.0/30 or 203.0.113.1, 203.0.113.2**
   - **Description:** Enter an appropriate description.
6. Click Create.

Which Security Application Do I Use?

Security list → Add instance → Security IP list → Security application → Security rule

Identify a Predefined Security Application:

1. Go to the web console.
2. Click the Network tab.
3. Click the Security Applications tab in the left pane.
4. Identify the security application that you want to use.

Create a Security Application:

1. Go to the web console.
2. Click the Network tab.
3. Click the Security Applications tab in the left pane.
4. Click Create Security Application.
5. Enter or select the following information:
   - Name
   - Port Type
   - Description
6. Click Create.

ORACLE

Several security applications are predefined for you. You can create a new security application, or use one of the predefined security applications in your security rules. To see the set of predefined security applications:

1. Go to the web console.
2. Click the Network tab.
3. Click the Security Applications tab in the left pane.

For now, identify the http security application and make a note of the name and port number of this security application. You can use this security application to create your security rule.

At any time, if you want to create a security application:

1. Go to the web console.
2. Click the Network tab.
3. Click the Security Applications tab in the left pane.
4. Click Create Security Application.
5. Enter or select the following information:
   - Name: Enter a name for the security application.
   - Port Type: Select the port type.
     - If you select the tcp or udp port type, then enter the port range.
     - If you select the icmp port type, then enter the ICMP type.
   - Description: Enter a meaningful description.
6. Click Create.

How Do I Create a Security Rule?

Security list → Add instance → Security IP list → Security application → **Security rule**

1. Go to the Oracle Compute Cloud Service console.
2. Click the Network tab.
3. Click Create Rule. The Create Security Rule dialog box is displayed.
4. Enter or select the following:
   – Name
   – Status
   – Security Application
   – Source
   – Destination
   – Description
5. Click Create.

ORACLE

To create a security rule:

1. Go to the web console.
2. Click the Network tab.
3. Click Create Rule. The Create Security Rule dialog box is displayed.
4. Enter or select the following:
   - **Name:** Enter a name for the security rule.
   - **Status:** By default, new security rules are enabled. Keep this default setting for now.
   - **Security Application:** Select the protocol that you want this security rule to use. Select the predefined http security application for now.
   - **Source:** Select the security list or security IP list from which traffic over the specified protocol should be allowed. Select public-internet for now.
   - **Destination:** Select the security or security IP list to which traffic should be allowed. Select the security list that you just created.
5. Click Create.

You can now access instances in your security list from any host over the public Internet using http.

But I Don't Want My Instance to be Accessible by Everybody. How Do I Disable Access to My Instance?

You can disable access to an instance in two ways:

- To prevent other hosts from accessing one specific instance, you can remove the instance from specific security lists that it is attached to.
- To prevent other hosts from accessing all the instances in a specific security list, you can disable specific security rules that use that security list.

To prevent other hosts from accessing an instance, you can restrict or disable access to the instance in either of the following ways:

To prevent other hosts from accessing one specific instance, remove the instance from specific security lists that it is attached to. When you remove an instance from a security list, the security rules that are defined for the security list are no longer applicable to the instance, and the instance cannot communicate with other instances in the security list. An instance that isn't associated with any security list is completely inaccessible. You can always use the web console to add such an instance to required security lists.

To prevent other hosts from accessing all the instances in a specific security list, you can disable specific security rules that use that security list. For example, consider a security list that is used as the destination in two security rules, one controlling traffic originating at external hosts listed in a specified security IP list, and the other controlling traffic originating at instances in another security list. You can disable either security rule, without affecting traffic controlled by the other security rule.

How Do I Remove an Instance from a Security List?

1. Go to the web console.
2. Click the Instances tab.
3. On the Instances page, identify the instance that you want to restrict access to. From the menu, select View.
4. On the instance details page, go to the security list that you want to remove your instance from. From the menu, select Remove from Security List.

How Do I Disable a Security Rule?

1. Go to the web console.
2. Click the Network tab.
3. Identify the security rule that you want to disable. From the menu, select Update.
4. In the Update Security Rule dialog box, set the Status to Disabled.
5. Click Update.

Quiz

Which steps are required to enable access to your instance?

a. Create the required security lists and security rules and add your instance to the appropriate security lists.

b. Ensure that you have associated a public IP address with your instance.

c. While creating an instance, select the option to configure the instance for SSH access.

d. Both a) and b)

e. Both b) and c)

Quiz

Which action should you take to prevent an instance from being accessed by an external host?

a. Remove the instance from security lists that it is attached to

b. Disable the security rules that apply to security lists that the instance is attached to

Quiz

An instance is a part of security list that has the inbound policy Deny. That security list used as a destination in a security rule that allows HTTP access from the public Internet. Can you access that instance from your computer using HTTP?

a. Yes

b. No

Summary

In this lesson, you should have learnt how to:

- Explain networking concepts
- Create security lists and security rules to enable access to your Instance
- Disable access to your Instance

ORACLE

Resource: Links

For more information regarding Oracle Compute Cloud Service, visit
http://docs.oracle.com/cloud/latest/stcomputecs/index.html

ORACLE

10

Oracle Compute Cloud Service Block Storage Volumes

ORACLE

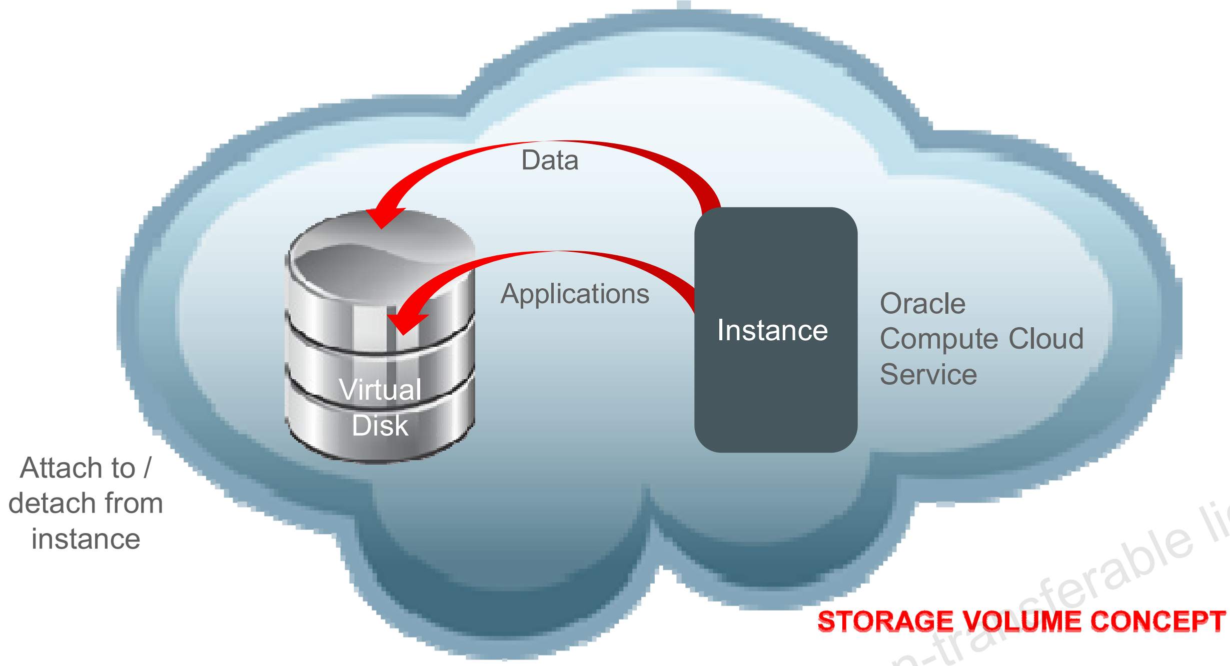
Objectives

After completing this lesson, you should be able to:

- Describe the process of adding block storage to an instance
- Create a block storage volume and add it to an instance
- Remove a block storage volume from an instance and delete it

The virtual machine (VM) is referred to as an instance henceforth in this course.

How Can I Use Block Storage with My Instance?



ORACLE

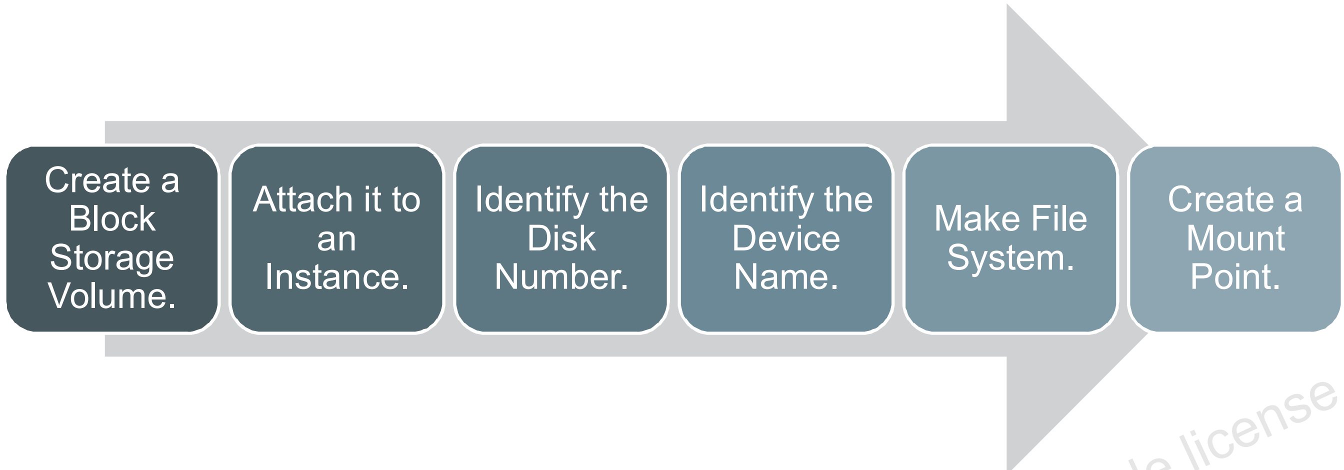
A storage volume is a virtual disk that provides persistent block storage for virtual machines.

You can use storage volumes to store data and applications.

Oracle Compute Cloud Service allows you to:

- Create block storage volumes and attach them to your instances. When you create a storage volume, you can specify the capacity that you need. The permitted range is between 1 GB and 2 TB, in increments of 1 GB.
- Attach up to 10 storage volumes to each Oracle Compute Cloud Service instance. However, note that a storage volume can be attached to only one instance at a time.
- Attach one or more storage volumes to an instance either while creating the instance or later, while the instance is running.
- Scale up or scale down the block storage capacity for the instance by attaching or detaching storage volumes even while the instance is running. However, note that you can detach a storage volume only if it was attached to a running instance. You cannot detach a storage volume that was attached to an instance during instance creation. Also, remember that, when a storage volume is detached from an instance or when the instance is deleted, data stored on the storage volume is not lost.

Workflow for Using a Block Storage Volume



ORACLE

- **CREATE:** You can use the Oracle Compute Cloud Service web console to create block storage volumes.
- **ATTACH:** After creating a block storage volume, you can attach it, or associate it with an instance by using the web console. You can do this either while creating the instance or later, while the instance is running. You can also use the web console to view details of a storage volume, such as its status, size, and the instance to which it is attached.
- **IDENTIFY DISK NUMBER, DEVICE NAME:** When you attach a storage volume to an instance, you must specify an index number. You can use this index number to identify the device name of a storage volume on the instance when you log in to the instance by using SSH.
- **MAKE FILE SYSTEM:** To make a file system, log in to your instance and use a tool such as mkfs to create a file system on the storage volume.
- **CREATE A MOUNT POINT:** On your instance, create a mount point for each storage volume. A mount point is like a directory where the storage volume is mounted. After you have created the mount point, you can cd to this directory to read/write data on the attached storage volume.

How Do I Create a Block Storage Volume?

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Storage tab.
3. Click Create Storage Volume.
4. The Create Storage Volume Wizard starts. Enter the following:
   – Name
   – Boot Image
   – Size, in GB
   – Storage property
   – Description
5. Click Create.

ORACLE

Creating a Storage Volume

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Storage tab.
3. Click Create Storage Volume. The Create Storage Volume wizard starts.
4. Select or enter the required information:
   - Enter a name for the storage volume. Note this name. You'll need it later to search for the storage volume on the Storage page. Select a name that you can use later to quickly identify the key characteristics of the storage volume.
   - If you intend to use this storage volume as a boot disk, choose the boot image from the drop down menu. Later, while creating an instance, you can specify this volume as the boot disk for the instance. Do not select this option for now.
   - Enter the size, in GB, of the storage volume. The allowed range is 1 GB to 2 TB. Consider the storage capacity needs of the applications that you plan to deploy on the instance, and leave some room for attaching more storage volumes in future. This approach helps you to use the available block storage capacity efficiently in the long run.
   - Select a storage property. For storage volumes that require low latency and high IOPS, such as for storing database files, select `/oracle/public/storage/latency`. For all other storage volumes, select `/oracle/public/storage/default`.
   - Enter a description for the storage volume.

Note: The web console might show other storage properties. But do not select any of them.

How Do I Add a Block Storage Volume to an Existing Instance?

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Storage tab.
3. Select the storage volume to attach and from the menu, select Attach Instance.
4. Select the instance to attach the storage volume to.
5. The Attach as Disk # field is auto-populated.
6. Click Attach.

ORACLE

Attaching a Storage Volume to an Instance

You can provide or increase block storage capacity for an instance by attaching storage volumes.

1. Go to the web console.
2. Click the Storage tab.
3. Identify the storage volume that you want to attach. From the menu, select Attach Instance.
4. Select the instance to which you want to attach the volume.
5. The Attach as Disk # field is filled automatically with the next available index at which the volume can be attached. You can leave this field at the automatically selected disk number or enter a higher number up to 10.

   Make a note of the disk number. You will need it when you mount the storage volume on the instance.
6. Click Attach.

How Do I Find the Storage Volume on My Instance?

1. Identify the disk number of the storage volume to mount.
2. Log in to the instance by using SSH.
3. List the devices available on your instance.
4. Identify the device name corresponding to the disk number.

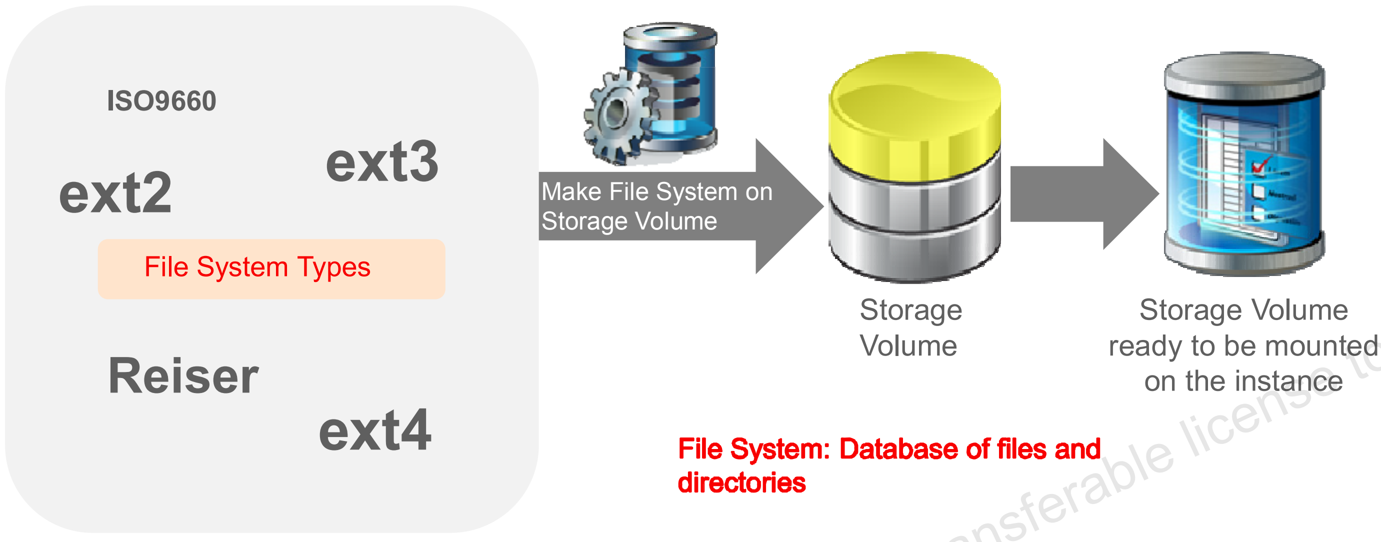
Identifying the New Storage Volume

1. After attaching a storage volume to an instance, identify the disk number of the storage volume.
2. Log in to the instance by using SSH.
3. List the devices available on your instance:

   ```
   ls /dev/xvd*
   ```

   Device names start from `/dev/xvdb` and are determined by the index number that you assigned when you attached the storage volumes. For example, if you attached a storage volume at index 1, the volume gets the device name, `/dev/xvdb`. The storage volume at index 2 would be `/dev/xvdc`; the storage volume at index 3 would be `/dev/xvdd`, and so on.
4. Identify the device name corresponding to the index number that you noted earlier.

What Should I Know About File Systems?



About File Systems

- A file system is a database of files and directories created on a storage volume.
- A file system enables you to access the physical locations on the storage device.
- You cannot store files on a storage volume that does not contain a file system.
- You should create a file system just once for each new storage volume that has no pre-existing data (or data that you want to remove). Creating a new file system on top of an existing file system will effectively remove the old data.
- After creating a file system, you must mount the file system on Linux, before you can access the storage volume and use it for data storage and retrieval.

The following are some of the most common types of file systems for data storage:

- The Second Extended file system (ext2): Nearly every Linux system uses ext2 or its newer version, ext3.
- The Third Extended file system (ext3): This is a newer version of the ext2 file system, augmented with journal support.
- The Fourth Extended file system (ext4)
- ISO9660 (iso9660): A CD-ROM standard, supported on Linux
- The Reiser filesystem (reiserfs): Journaled computer file system, supported by Linux

To create a file system on the target device, use mkfs.

How Do I Create a File System on the Storage Volume?

Use a tool such as `mkfs` to create a file system on the storage volume. For example, to create an ext3 file system on `/dev/xvdd`, run the following command:

```
sudo mkfs -t ext3 /dev/xvdd
```

ORACLE

Creating a File System on Your Storage Volume

1. After attaching a storage volume to an instance, identify the disk number of the storage volume.
2. Log in to the instance by using SSH.
3. List the devices available on your instance:

   ```
   ls /dev/xvd*
   ```

   Device names start from `/dev/xvdb` and are determined by the index number that you assigned when you attached the storage volumes. For example, if you attached a storage volume at index 1, the volume gets the device name, `/dev/xvdb`. The storage volume at index 2 would be `/dev/xvdc`; the storage volume at index 3 would be `/dev/xvdd`, and so on.
4. Identify the device name corresponding to the disk number that you noted earlier.
5. Use a tool such as `mkfs` to create a file system on the storage volume. For example, to create an ext3 file system on `/dev/xvdd`, run the following command:

   ```
   sudo mkfs -t ext3 /dev/xvdd
   ```

Note: If the Extended File System utilities are not available on your instance, a message such as the following is displayed:

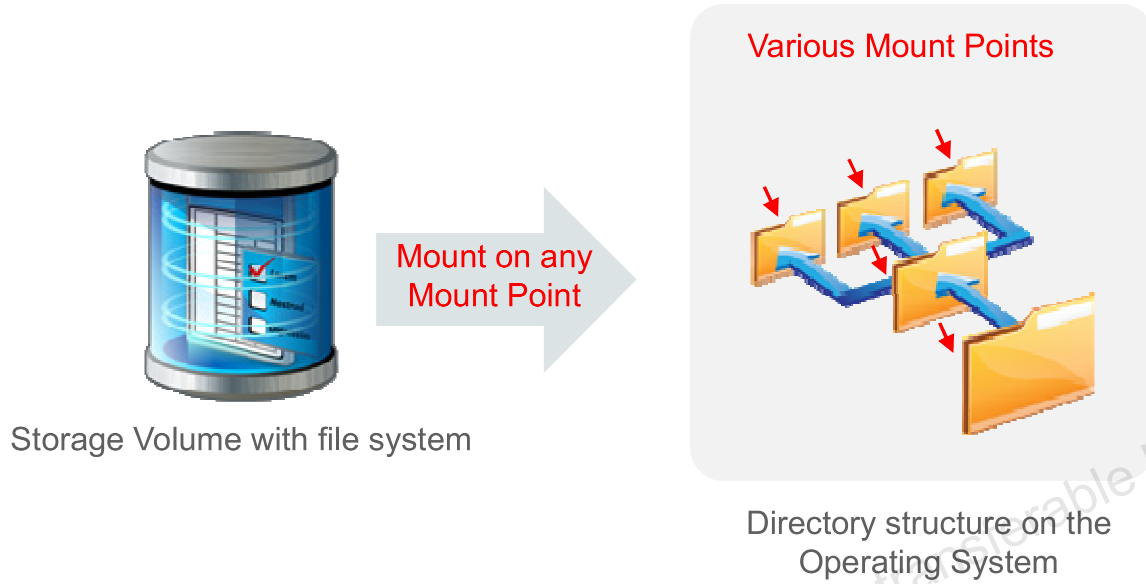
mkfs.ext3: No such file or directory

To install the Extended File System utilities, run the following command:

```
sudo yum install e4fsprogs
```

After the Extended File System utilities are installed, re-run the `mkfs` command.

What's a Mount Point?



About Mount Points

The process of associating a storage volume with an operating system is called **mounting**.

A **mount point** is the place in the current system's directory hierarchy where the storage volume and its file system will be attached. The mount point is always a normal directory. The mount point doesn't have to be created directly at the root (/); it can be created anywhere in the hierarchy of the system.

The `/etc/fstab` File System Table

Linux systems maintain a list of file systems and options in the `/etc/fstab` file. This is a plain text file. To ensure that your storage volumes are mounted at boot time, add the mount point details as an entry in the `/etc/fstab` file.

How Do I Mount the Storage Volume on My Instance?

1. Create a mount point on your instance.
2. Mount the storage volume on the mount point.
3. Make the mount persistent across instance restarts (optional).

ORACLE

Mounting the Storage Volume on Your Instance

1. Create a mount point on your instance. For example, to create the mount point `/mnt/store`, run the following command:

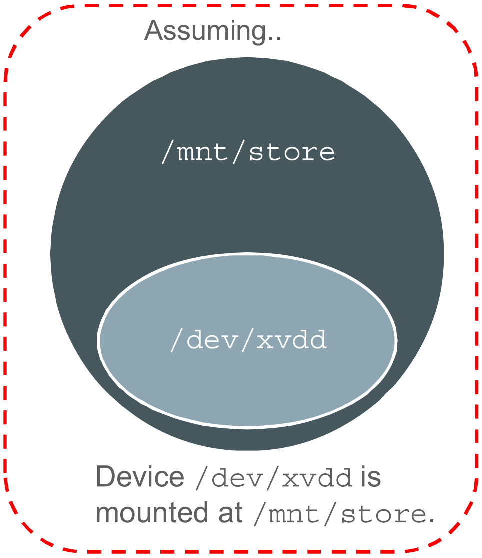
```
sudo mkdir /mnt/store
```

2. Mount the storage volume on the mount point that you created on your instance. For example, to mount the device `/dev/xvdd` at the `/mnt/store` directory, run the following command:

```
sudo mount /dev/xvdd /mnt/store
```

3. To make the mount persist across instance restarts, edit the `/etc/fstab` file and add the mount point as an entry in that file.

Now Can I Use the Storage Volume?



I/O Operations:

- The `sudo cd /mnt/store` command lands you in the storage volume.
- All data storage and retrieval at `/mnt/store` executes at the device.

Storage Volume Capacity:

- To view the size and utilization of the storage volume, use the command `sudo df`.

Accessing the Storage Volume from Your Instance

After the storage volume is mounted, you can access the storage volume by going to the mount point directory.

For example:

If the device `/dev/xvdd` is mounted at the path `/mnt/store`, any access of the storage volume further occurs via the path `/mnt/store`.

So, to check the contents of the device, perform the following operation:

```
sudo cd /mnt/store
sudo ls -l
```

This lists all the files and directories currently available on the device `xvdd`. Further, any file stored in the directory path `/mnt/store` is effectively stored on the `/dev/xvdd` storage volume.

To view the size and utilization of the storage volume, use the command `sudo df`. The output looks similar to the following:

Filesystem	1024-blocks	Used	Available	Capacity	Mounted On
/dev/xvdc	1011928	71400	889124	7%	/usr
/dev/xvdd	17710044	9485296	7325108	56%	/mnt/store

What If I Don’t Need a Storage Volume Anymore?

1. Identify the disk number of the storage volume to unmount.
2. Log in to your instance by using SSH.
3. List the devices available on your instance and their mount points.
4. Identify the device name corresponding to the disk number to unmount.
5. Note the mount point for that device.
6. Run the `umount` command.
7. Edit `/etc/fstab` and remove the mount.

ORACLE

If you no longer need a storage volume, here’s what you can do:

- Unmount the storage volume from your instance.
- Detach the storage volume from your instance.
- Now, you can either keep the storage volume and use it later with the same instance or a different instance, or you can delete the storage volume.

Unmounting a Storage Volume from an Oracle Linux Instance

To detach a storage volume from your instance, or to delete the instance that a storage volume is attached to, you must first unmount the storage volume.

To unmount a storage volume from an Oracle Linux instance, perform the following steps:

1. Identify the disk number of the storage volume that you want to unmount.
2. Log in to the instance by using SSH.
3. List the devices available on your instance and their mount points:

sudo df -hT

```
Filesystem          Type           Size    Used    Avail   Use%
     Mounted on
   /dev/xvdb2       ext4           16G     2.9G    12G     20%    /
   tmpfs            tmpfs          3.7G            0       3.7G     0%
     /dev/shm
   /dev/xvdb1       ext4           194M    90M     94M     49%
     /boot
   /dev/mapper/vg_binaries-lv_tools
                           ext4           9.9G    156M    9.2G    2%
     /u01/app/oracle/tools
   /dev/mapper/vg_backup-lv_backup
                           ext4           20G     4.0G    15G     21%
     /u01/data/backup
   /dev/mapper/vg_domains-lv_domains
                           ext4           9.9G    1.2G    8.3G    12%
     /u01/data/domains
   /dev/mapper/vg_binaries-lv_mw
                           ext4           9.9G    2.0G    7.4G    21%
     /u01/app/oracle/middleware
   /dev/mapper/vg_binaries-lv_jdk
                           ext4           2.0G    334M    1.6G    18%
     /u01/jdk
```

Device names start from /dev/xvdb and are determined by the index number that you assigned when you attached the storage volumes. For example, if you attached a storage volume at index 1, the volume gets the device name, /dev/xvdb. The storage volume at index 2 would be /dev/xvdc; the storage volume at index 3 would be /dev/xvdd; and so on.

Note: For an instance that is set up to boot from a non-persistent boot disk, /dev/xvda is used for the boot disk.

4. Identify the device name corresponding to the disk number that you want to unmount, and note the mount point for that device.

   For example, to unmount the storage volume that is attached at index 3, you must unmount /dev/xvdd.

5. Run the umount command.

   sudo umount mount_point

   For example, to unmount the device that is mounted at /mnt/store, run the following command:

   sudo umount /mnt/store

6. If you had defined this mount point in the /etc/fstab file, edit /etc/fstab and remove the mount.

If you no longer need the volume that you just unmounted, you can detach it from the instance and delete it.

I’ve Unmounted the Storage Volume. Now How Do I Detach It?

1. Go to the web console.
2. Click the Storage tab.
3. Go to the storage volume that you want to detach.
4. From the menu, select Detach Instance.

Detaching a Storage Volume from an Instance

When you no longer require access to a storage volume, you can unmount it and detach it from your instance.

After you detach a storage volume from an instance, you can no longer read from or write data to the storage volume, unless you attach it to any instance.

Note: You cannot detach or delete a storage volume that was attached while creating an instance. If you are sure that a storage volume is no longer required, back up the data elsewhere and delete the storage volume. Perform the following steps:

1. Go to the web console.
2. Click the Storage tab.
3. Go to the storage volume that you want to detach.
4. From the menu, select Detach Instance.

You can also detach a storage volume from the Instances page.

Note: After detaching a block storage from one instance, you can attach and mount it on another instance.

How Do I Delete the Block Storage Volume?

1. Go to the web console.
2. Click the Storage tab.
3. Go to the storage volume that you want to delete.
4. From the menu, select Delete.

ORACLE

Deleting a Block Storage Volume from an Instance

If you delete a storage volume, all the data and applications that were saved on that storage volume are lost. Delete a storage volume only when you are sure that you no longer need any of the data that is stored on that volume.

Ensure that the storage volume that you want to delete is not attached to any instance.

1. Go to the web console.
2. Click the Storage tab.
3. Go to the storage volume that you want to delete.
4. From the menu, select Delete.

Quiz

Which of the following steps is required to ensure that a storage volume is available to an instance when an instance is restarted?

a. Attach the storage volume to the instance while creating the instance.
b. Run the `mount` command after the instance is running.
c. Add an entry in the `/etc/fstab` file with the disk details of the storage volume.

Quiz

Which of the following statements is true?

a. A storage volume must be created before the instance you want to attach it to.

b. You can access a storage volume and perform read/write operations after attaching the storage volume to an instance.

c. You cannot detach a storage volume that was attached to an instance during instance creation.

Quiz

Identify the correct sequence of steps for deleting a storage volume. (Select all that apply.)

a. Detach > Unmount > Remove the entry from `/etc/fstab` > Delete
b. Unmount > Detach > Remove the entry from `/etc/fstab` > Delete
c. Remove the entry from `/etc/fstab` > Unmount > Delete > Detach
d. Remove the entry from `/etc/fstab` > Unmount > Detach > Delete

Summary

In this lesson, you should have learned how to:

- Describe the process of adding block storage to an instance
- Create a block storage volume and add it to an instance
- Remove a block storage volume from an instance and delete it

Resource: Links

For more information about Oracle Compute Cloud Service, visit http://docs.oracle.com/cloud/latest/stcomputecs/index.html.

ORACLE

11

Oracle Compute Cloud Service Bootable Storage Volumes

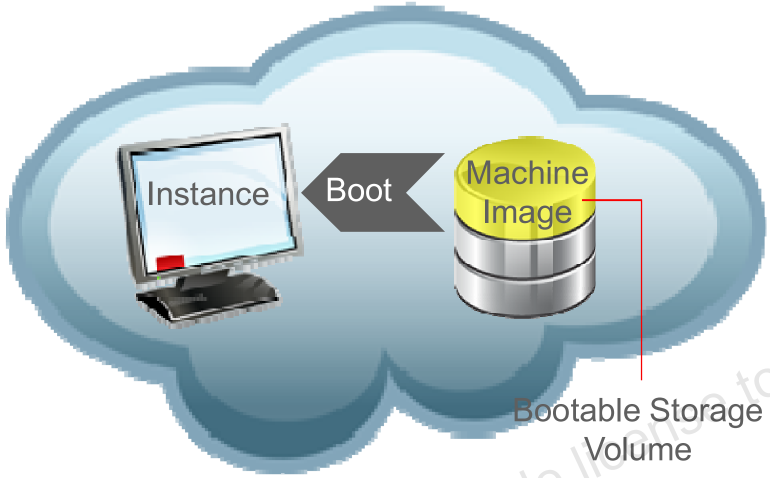
Objectives

After completing this lesson, you should be able to:

- Explain the use of a bootable storage volume
- Create a bootable storage volume
- Create an instance using a bootable storage volume

What Is a Bootable Storage Volume?

- While creating a storage volume, you can select a machine image to be stored on the storage volume.
- You can then use this storage volume as the boot disk for an instance.



Machine Image = OS + Packages

ORACLE

A storage volume is a virtual disk that provides persistent block storage space for instances in Oracle Compute Cloud Service.

To use a persistent boot disk, you can start by creating a bootable storage volume. While creating a storage volume, select the machine image that you want to use. This machine image is then stored on your bootable storage volume. When you boot an instance from such a storage volume, this machine image is used to boot the instance. Alternatively, you can retain the default option to create a boot disk when you create your instance using the web console.

If required, you can select a nonpersistent boot disk while creating an instance. However, if you use a nonpersistent boot disk, then, if you delete and re-create your instance, any changes that you made to the instance are lost. However, if you use a bootable storage volume as the boot disk for an instance, then changes that you make to the boot disk are not lost even when you delete and re-create the instance.

What Is a Machine Image?

- A machine image is a template of a virtual hard disk of a specific size with an installed operating system.
- You use machine images to create virtual machine instances in Oracle Compute Cloud Service.
- While creating instances, you can either use images provided by Oracle, or create your own custom machine images.

ORACLE

- A machine image is a template of a virtual hard disk of a specific size with an installed operating system.
- You use machine images to create virtual machine instances in Oracle Compute Cloud Service.
- While creating instances, you can either use images provided by Oracle, or create your own custom machine images.

Oracle-Provided Images

Oracle provides machine images for Oracle Linux 6.4 and 6.6.

The Oracle-provided images include the essential packages that are necessary to get started using the instance that you create in Oracle Compute Cloud Service. Specifically, they include the basic packages required for the following:

- Development tools: Expect, Java OpenJDK, GCC suite, GNU utilities, Perl, Ruby, Python, and so on
- Basic X11 desktop
- Remote X11 access with VNC
- Xterm client
- Security and auditing with OpenSCAP and AIDE
- Integration with name services such as OpenLDAP, Kerberos, and NIS
- System administration tools
- Firefox and Elinks web browsers
- EMACs and vim editors

How Do I Create a Bootable Block Storage Volume?

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Storage tab.
3. Click Create Storage Volume.
   – The Create Storage Volume wizard starts.
4. On the Storage Details page, enter the following:
   – Name, Boot Image, Size, Storage Property, Description
5. Click Create.

You have already learned how to create a storage volume in the previous lesson. Follow the same steps to launch the Create Storage Volume wizard.

Enter the following values for the fields in the Storage Details Page:

Name: Enter an appropriate name. For example, enter <yourfistname>_bootable_vol_01.

Boot Image: Select the image that you want to use to create a bootable storage volume. For example, select OL-6.6-20GB-x11-RD or similar Oracle provided Oracle Linux image.

Size: Enter an appropriate size, in GB. For example, enter 22.

- **Note:** The size of the storage volume must be at least five percent larger than the selected image size.

Storage Property: Select /storage/default.

Description: Enter an appropriate description. For example, enter Bootable Storage Volume 22 GB.

How Do I Create an Instance with this Storage Volume?

1. Go to the web console.
2. Click Create Instance.
   i. On the Image page: Select an image.
   ii. On the Shape page: Retain the default values.
   iii. On the Instance page: Enter a name, select the SSH Public key generated earlier, and retain the default values in other fields.
   iv. On the Storage page: Click Attach Existing Volume, select the bootable storage volume, select the Boot Drive option, and click Add.
   v. On the Review page: Verify the information and click Create.

ORACLE

1. On the Oracle Compute Cloud Service console, click the Instances tab, and then click Create Instance. The Create Instance wizard starts.
2. On the Image page, select the same image as the one selected while creating the Bootable Storage Volume.
3. On the Shape page, select `OC3`.
4. On the Instance page:
   - High Availability Policy: Retain default HA Policy, Active.
   - Name: Enter name `<yourfirstname>_instance_02`.
   - Label: Enter a label.
   - Description: Enter a description.
   - Tags: Leave this field blank for now.
   - DNS Hostname Prefix: Leave this field blank for now.
   - Public IP Address: Select `Auto Generated`.
   - Security Lists: Leave this field blank for now.
   - SSH Keys: Select the SSH Public Key created earlier.
   - Custom Attributes: Leave this field blank for now.

4. On the Storage page:
   - Attach Storage Volume: Select the storage volume that you want to attach.
   - Attach as Disk #: Retain the default number. The disk number that you specify here determines the device name. The disk attached at index 1 is named /dev/xvdb, the disk at index 2 is /dev/xvdc, the disk at index 3 is /dev/xvdd, and so on. Make a note of the disk number. You'll need it later when you mount the storage volume on the instance. This is automatically set to 1 when the storage volume is specified as a boot disk.
   - Boot Drive: Select this option to use the specified storage volume as the boot disk. When you select this option, the disk number is automatically set to 1.
5. On the Review page, verify the information that you entered and then click Create. Your instance is created.

How Do I Access the Storage Volume?

- Log in to the instance with PuTTY, the SSH client for Windows.
  Device listing:
  `sudo ls -l` command at the path `/dev` lists all the devices associated with the instance.
- Check for persistence of the bootable storage volume (optional)

After logging in to the instance using PuTTY, check the path /dev for the list of devices attached to the instance with the command sudo ls –l. Identify the bootable storage volume and its attributes. The bootable storage volume by default is assigned the least index number. For example,

The result of sudo ls –l /dev/xvd* produces the following result:

brw-r----	1	root	disk	202, 16 Feb	9
05:09	/dev/xvdb				
brw-r----	1	root	disk	202, 17 Feb	9
05:09	/dev/xvdb1				
brw-r----	1	root	disk	202, 18 Feb	9
05:09	/dev/xvdb2				

In the above example, device xvdb is the bootable storage volume.

Check for Persistence

Test the persistence of the bootable storage volume by

- making changes to the boot volume by changing the value of a suitable kernel parameter in the OS image
- deleting the existing instance associated with it
- creating a new instance
- associating the new instance with the same bootable storage volume
- finally checking if the changes made to the boot volume persist

Quiz

Which of the following statements are true?

a. When you create an Oracle Compute Cloud Service instance, a persistent boot volume is used by default.

b. If you use a bootable storage volume as the boot disk for an instance, then changes that you make to the boot disk are not lost even when you delete and re-create the instance.

c. You can create a custom machine image with any OS of your choice and use this machine image to create Oracle Compute Cloud Service instances.

ORACLE

Quiz

If you want your instance to use a persistent boot disk, you can:

a. Go to the Storage tab, click Create Storage Volume to create a bootable storage volume and select an instance to attach it to.
b. Create a bootable storage volume and then click the Create Instance button on the Instances tab. In the Create Instance wizard, select the bootable storage volume while creating the instance.
c. Create a bootable storage volume. Create an instance using the default options in the Create Instance wizard. When the instance is running, select View, and on the instance details page, attach the bootable storage volume.

Summary

In this lesson, you should have learned how to:

- Explain the use of a bootable storage volume
- Create a bootable storage volume
- Create an instance using a bootable storage volume

Resource: Links

For more information regarding Oracle Compute Cloud Service, visit http://docs.oracle.com/cloud/latest/stcomputecs/index.html.

12

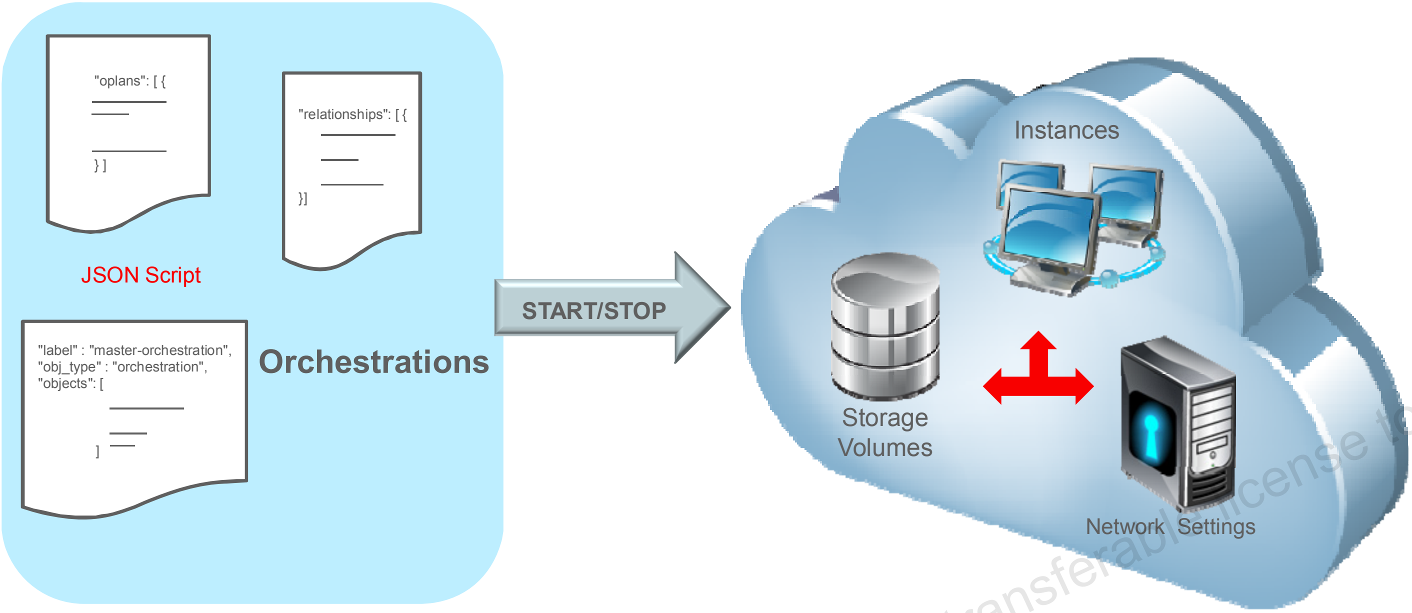
Oracle Compute Cloud Service Orchestrations

Objectives

After completing this lesson, you should be able to:

- Describe how to use orchestrations
- Explain the structure of an orchestration
- List the objects that you can create using an orchestration
- Create an orchestration file using JSON
- Create an instance using an orchestration
- Delete an instance by stopping an orchestration

What Is an Orchestration?



An orchestration is a script that defines the attributes and interdependencies of a collection of compute, networking, and storage resources in Oracle Compute Cloud Service. You can use orchestrations to automate the provisioning and lifecycle operations of an entire virtual compute topology.

For example, you can use orchestrations to create and manage a collection of instances hosting a multi-tiered application stack with all the necessary networking, storage, and security settings.

At any time, you can delete all the instances in an orchestration just by stopping the orchestration. Later on, if required, you can re-create all the objects in that orchestration by starting the orchestration again. Storage attachments, security lists, and any other objects that are created or referenced in the orchestration are then re-created and referenced again, automatically.

Why Should I Use Orchestrations?

- Simplify the process of provisioning and removing objects
- Assign specified objects to specified instances – always
- Define dependencies between objects
- Specify a High Availability policy for instances

Here's how orchestrations are useful.

When you create an orchestration file, you can use it to create not just one instance, but many instances in one file and at the same time. Let's say you want to spin up five instances and you want each instance to use a different SSH key, bootable storage volume, and persistent public IP address. You know that you'll need these instances for – let's say – a few days, and after that you won't want to use them again for several days. If you create these instances one at a time, you'll have to make a note of which resources you used with each instance. If you have another, say, 15 instances in the same environment, then you'll have to make a careful note of which instances you want to delete and start up again later on. It's tedious, right?

Now, if you use an orchestration to do this, it's much simpler. You define all five instances in a single orchestration file. Ensure that the storage volumes, security lists, SSH keys, and IP reservations that you want to use already exist – you can even use separate orchestration files to create those objects. Then just start the orchestration to create all five instances.

When you're done with the instances, stop the orchestration to delete all five instances.

If you created the associated objects, such as IP reservations and storage volumes, in separate orchestrations, then you can stop those orchestrations as well. Later on, when you want to use those instances again, start the orchestrations that create the referenced objects first, then start the orchestration that creates your instances. All your instances will start up using exactly the same resources as they had the first time around.

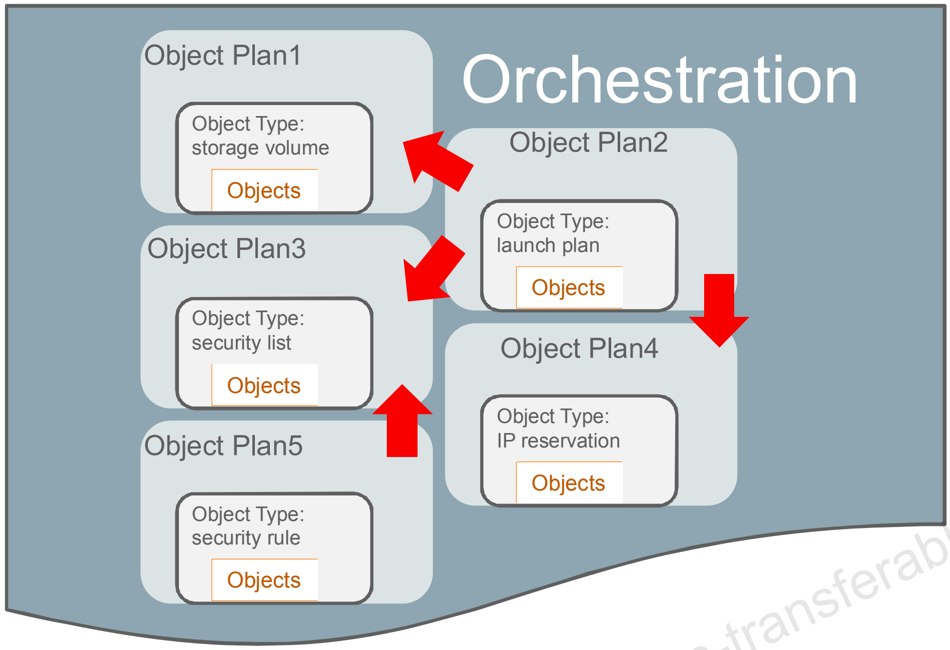
When you create an instance using the web console, you must ensure that you have created the prerequisite objects, such as SSH public keys or bootable storage volumes, before you start the create instance wizard.

When you use an orchestration to create an instance, it is possible to reference an object such as a security list or a public IP address reservation, that is not currently available. You might not have created the referenced object yet, you might have created it and then removed it, or as for an IP reservation, you might have used it with another instance. If this happens, you will not be able to start your orchestration.

To avoid this situation, you can create many of the referenced or prerequisite objects in the same orchestration that you use to create your instance. You can then define dependencies to ensure that the prerequisite objects are always created before the object that refers to them.

Another benefit of using an orchestration to create an instance is that the orchestration can monitor the status of the instances it creates. If you set the High Availability (HA) policy in an orchestration to active, the orchestration continuously monitors the status of its instances. If an instance in such an orchestration goes down, the instance is re-created automatically.

What Does an Orchestration Look Like?



An orchestration is designed with primary building blocks called object plans, or "oplan"s.

Each oplan contains all the attributes for the “object type” defined in that oplan.

An “object type” refers to the Oracle Compute Cloud Service resource that you want to create. For example, if you want to create a storage volume, the “object type” would be storage/volume. If you want to create an instance, the “object type” would be launchplan.

The “objects” attribute defines the properties or characteristics of the Oracle Compute Cloud Service specific object that you want to create. The attributes of the “objects” vary depending on the object type.

For example, if you want to create a storage volume, the “object type” would be storage/volume, and the attributes under “objects” would include size and bootable. If you want to create an instance, the object type would be launch plan, and the objects would include instances, along with instance-specific attributes, such as image list and shape.

Which Objects Can I Create Using an Orchestration?

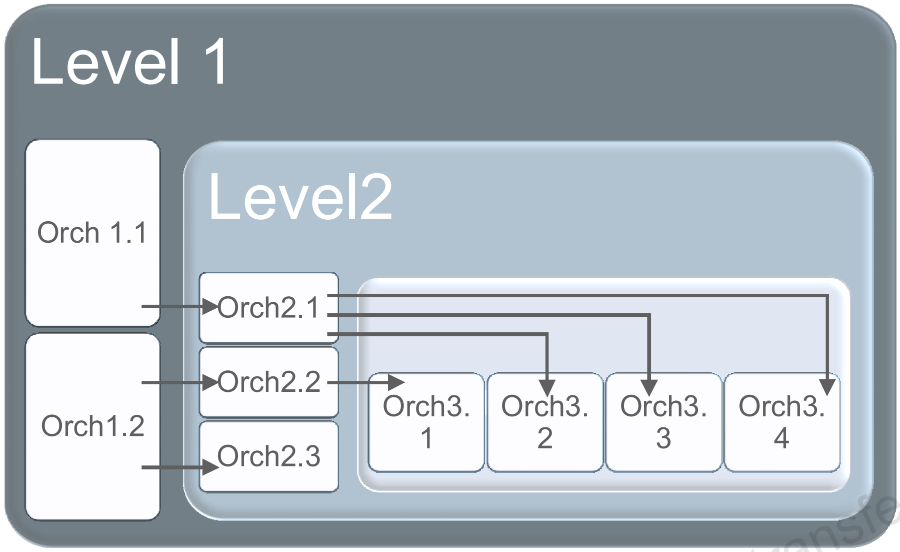
Object Type	Description
ip/reservation	Reserves an IP address.
launchplan	Creates an instance.
orchestration	Starts a set of orchestrations.
storage/volume	Creates a storage volume.
secapplication	Creates a security application.
seciplist	Creates a security IP list.
seclist	Creates a security list.
secrule	Creates a security rule.

ORACLE

You can define the following object types in an orchestration:

- **ip/reservation:** Reserves an IP address. To associate an IP reservation with an instance that is defined in the same orchestration, you must specify a relationship between the ip/reservation and the launchplan object plans.
- **launchplan:** Creates an instance. To add an instance to a security list that is defined in the same orchestration, you must specify a relationship between the launchplan and the seclist object plans.
- **orchestration:** Starts a set of orchestrations.
- **storage/volume:** Creates a storage volume. To attach this storage volume to an instance that is defined in the same orchestration, you must specify a relationship between the storage/volume and the launchplan object plans.
- **secapplication:** Creates a security application. To use this security application in a security rule that is defined in the same orchestration, you must specify a relationship between these objects.
- **seciplist:** Creates a security IP list. To use this security IP list in a security rule that is defined in the same orchestration, you must specify a relationship between these objects.
- **seclist:** Creates a security list. To use this security list in a security rule that is defined in the same orchestration, you must specify a relationship between these objects.
- **secrule:** Creates a security rule. If this security rule uses security applications, security lists, or security IP lists that are defined in the same orchestration, then you must specify a relationship between these objects.

What If I Don't Want All My Objects in One Orchestration?



You can use separate orchestrations to define different sets of objects. For example, you could create separate orchestrations for networking objects, for storage volumes, and for instances.

If you want to orchestrate different sets of interrelated objects together, you can create a master orchestration that references multiple nested orchestrations.

Suppose you want to start two orchestrations that create two different sets of instances. The instances in Orchestration 1.1 will be added to three security lists that are defined in a separate orchestration, say Orch2.1. These security lists are used by security rules defined separately in orchestrations Orch3.2, Orch3.3, and Orch3.4. Also, assume that the instances in Orch1.2 use similar networking objects defined in Orch2.2 and Orch3.1 and that they additionally require storage volumes, which are defined in Orch2.3.

Now you can create a master orchestration that references Orch1.1 and Orch1.2, where Orch1.1 and Orch1.2 in turn reference each of their respective nested orchestrations. You can start or stop an individual orchestration at any level of nesting, or you can start or stop the master orchestration, which causes all the nested orchestrations to start or stop.

Depending on the nature of the objects defined in the orchestrations, you might also need to specify dependencies between the different orchestrations, to ensure that all objects defined in the various orchestrations are created in the appropriate sequence.

How Do I Define Dependencies in an Orchestration?

- Use the `relationships` attribute to specify the sequence in which the objects must be created in the orchestration
- Specify the two objects that have a relationship, identified by their oplan labels
- Specify the relationship type as `depends`

ORACLE

The "relationships" attribute of an orchestration allows you to specify dependencies between oplans.

For example, if you define a storage volume in an orchestration and you also define an instance that the storage volume is attached to, then in the relationships section of the orchestration, you can specify that the launchplan object plan depends on the storage/volume object plan. This ensures that the storage volume is created before the instance is created.

So if you define a storage volume in an orchestration with the oplan label storagevolume1, and a launch plan with the oplan label boot-from-storagevolume1, then the relationship between these objects is as follows:

```
"relationships": [
    {
      "oplan": "boot-from-storagevolume1",
      "to_oplan": "storagevolume1",
      "type": "depends"
    }
 ]
```

For more complex scenarios, you can define multiple relationships.

For example, to create a security list (seclist1), a security application (secapplication1), and a security rule (secrule1) in a single orchestration, define the following relationships to ensure that both the security application and the security list are created before the security rule:

```
"relationships": [
    {
      "oplan": "secrule1",
      "to_oplan": "seclist1",
      "type": "depends"
    },
    {
      "oplan": "secrule1",
      "to_oplan": "secapplication1",
      "type": "depends"
    }
  ]
```

What about specifying dependencies between nested orchestrations in a master orchestration? Consider a master orchestration that references three nested orchestrations with the following labels:

- **instances-orchestration:** Defines instances.
- **network-orchestration:** Defines networking objects, such as security lists and IP addresses that instances must be associated with during instance creation.
- **storage-orchestration:** Defines storage volumes that must be associated with instances during instance creation.

In order for instances-orchestration to be able to create its instances, all objects in network-orchestration and storage-orchestration must exist when the instances-orchestration starts. In this case, you define the following relationships:

```
"relationships": [
    {
        "oplan": "instances-orchestration",
        "to_oplan": "network-orchestration",
        "type": "depends"
    },
    {
        "oplan": "instances-orchestration",
        "to_oplan": "storage-orchestration",
        "type": "depends"
    }
]
```

How Do I Create an Instance Using an Orchestration?

1. Build your orchestration using JSON.
2. Upload the orchestration to Oracle Compute Cloud Service.
3. Check that the prerequisite objects are available.
4. Start the orchestration.

ORACLE

To create instances using orchestrations, you build an orchestration in a JSON-formatted file, upload it to Oracle Compute Cloud Service, and then start the orchestration. Before you start your orchestration, however, ensure that you have created all your prerequisite objects.

Steps to Create an Instance Using an Orchestration

1. Build your orchestration using JSON.
2. Upload the orchestration to Oracle Compute Cloud Service.
3. Check that the prerequisite objects are available.
4. Start the orchestration.

ORACLE

What Is JSON? Why Should I Use It?

- JSON is JavaScript Object Notation (JSON)
- A lightweight format used for data interchange
- Based on a subset of the JavaScript Programming Language
- Easier to use than XML
- Universal data structures

ORACLE

JavaScript Object Notation (JSON), is a syntax for storing and exchanging data. It is a lightweight format, based on a subset of the JavaScript Programming Language.

JSON utilizes two data structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another.

JSON has very simple syntax and is very easy to learn and use. It is generally considered to be an easier-to-use option to XML.

JSON Syntax

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

Example:

```
"storage_attachments":
  [
   {"index": 1, "volume": "/Compute-acme/joe.jonathan@example.com/OL66_boot”},
   {"index": 2, "volume": "/Compute-acme/joe.jonathan@example.com/data1”}
  ]
```

Object

Array

Name / Value pair

ORACLE

JSON data consists of a name and a value written as name/value pairs. JSON values can be:

- A number – either integer or floating point
- A string – enclosed in quotation marks " “
- A Boolean value – 1 / 0
- An array – enclosed in square brackets [ ]
- An object – enclosed in curly braces { }
- Null

Example

```
"storage_attachments":
    [
      {"index": 1, "volume": "/Compute-acme/joe.jonathan@example.com/OL66_boot”},
      {"index": 2, "volume": "/Compute-acme/joe.jonathan@example.com/data1”}
   ]
```

In this example, the object “storage_attachments" is an array containing two objects. Each object is a record of a storage volume. Each object has two name/value pairs to indicate the name of the storage volume and its index number.

You can use a JSON editor to create your orchestration file, and a JSON syntax checker of your choice to validate your JSON syntax.

Sample JSON to Build Your First Orchestration – Single Instance

Now modify this sample script and build your own Orchestration

```
{
  "description": "Simple oplan with an ssh key and a security list",
  "name": "/Compute-acme/joe.jonathan@example.com/simple_orchestration",
  "oplans": [
    {
      "label": "simple_oplan",
      "obj_type": "launchplan",
      "objects": [
        {
          "instances": [
            {
              "imagelist": "/oracle/public/ol_6.6_20GB",
              "label": "OL_6.6_20GB",
              "networking": {
                "eth0": {
                  "seclists": [
                    "/Compute-acme/joe.jonathan@example.com/my_instances"
                  ],
                  "nat": "ipreservation:/Compute-acme/joe.jonathan@example.com/ip1"
                }
              },
              "shape": "oc3",
              "storage_attachments": [
                {
                  "index": 1,
                  "volume": "/Compute-acme/joe.jonathan@example.com/OL66_boot",

                },
                {
                  "index": 2,
                  "volume": "/Compute-acme/joe.jonathan@example.com/data1"
                }
              ],
              "boot_order": [1],
              "sshkeys": [
                "/Compute-acme/joe.jonathan@example.com/ssh-key1"
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Image
Label
Security list
IP reservation
Bootable storage volume
Data storage volume
SSH keys

ORACLE

This sample orchestration does the following:

- Defines an instance with the label OL_6.6_20GB, the oc3 shape, and using the /oracle/public/ol_6.6_20GB image
- Adds the instance to the security list/Compute-acme/joe.jonathan@example.com/my_instances
- Associates the IP reservation /Compute-acme/joe.jonathan@example.com/ip1 with the instance
- Attaches the bootable storage volume /Compute-acme/joe.jonathan@example.com/OL66_boot to the instance
- Attaches the data storage volume /Compute-acme/joe.jonathan@example.com/data1 to the instance
- Associates the SSH public key /Compute-acme/joe.jonathan@example.com/ssh-key1 with the instance

Steps for Building Your First Orchestration

- Copy the sample orchestration to a plain text file, and open the file in any text editor.
- Replace the name of the orchestration with an appropriate three-part name (/Compute-identity_domain/user/object).
- Change the value of the imagelist attribute to the image that you want to use. Select oel_6.4_5GB_RD_10oct2014.
- Under instances, change the value of the label attribute to any label that you want.
- Replace the security list /Compute-acme/joe.jonathan@example.com/my_instances with a security list that you have already created.
- Replace the IP reservation /Compute-acme/joe.jonathan@example.com/ip1 with an IP reservation that you have already created.
- Replace the oc3 shape with the shape that you want to use. For now, use oc3.
- Replace the storage volume /Compute-acme/joe.jonathan@example.com/OL66_boot with the bootable storage volume that your instance should boot from.
- Replace the storage volume /Compute-acme/joe.jonathan@example.com/data1 with a storage volume that you want to attach to the instance. If you want to attach more storage volumes, then specify the index for the storage attachment and the name of the storage volume.
- Replace the SSH key /Compute-acme/joe.jonathan@example.com/ssh-key1 with a key that you have created and added to Oracle Compute Cloud Service.
- Save the orchestration file. You should also validate your JSON file. You can do this by using a third-party tool. If your JSON format is not valid, then an error message is displayed when you upload the orchestration.

Steps to Create an Instance Using an Orchestration

1. Build your orchestration using JSON.
2. Upload the orchestration to Oracle Compute Cloud Service.
3. Check that the prerequisite objects are available.
4. Start the orchestration.

How Do I Upload My Orchestration to Oracle Compute Cloud Service?

1. Go to the Oracle Compute Cloud Service console.
2. Click the Orchestrations tab.
3. Click Upload Orchestration and select the orchestration file that you want to upload.

ORACLE

Before you upload your orchestration JSON, check that your file is complete and use a syntax checker to validate your JSON. To upload the file:

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Orchestrations tab.
3. Click Upload Orchestration and select the orchestration file that you want to upload.

Steps to Create an Instance Using an Orchestration

1. Build your orchestration using JSON.
2. Upload the orchestration to Oracle Compute Cloud Service.
3. Check that the prerequisite objects are available.
4. Start the orchestration.

ORACLE

What Are the Prerequisite Objects for Creating an Instance?

Any object referenced in your orchestration file, such as:

- SSH public key(s)
- Bootable storage volume (optional)
- Storage volumes (optional)
- IP reservation (optional)
- Security lists

When you define your instance in an orchestration, you can reference any of the following objects:

- SSH public key(s) – Use the one you created earlier.
- A bootable storage volume – You have created this earlier.
- One or more storage volumes for the data and applications that you plan to deploy on your instance – You have created this earlier. Make sure that the instance to which this was earlier attached is deleted, to be able to reuse this storage volume.
- An IP reservation, if you want your instance to have a persistent public IP address. So far, you have used an autogenerated IP address to associate with the instance. The procedure to create an IP reservation is discussed in Lesson 10, Configuring Network Settings. Refer to the steps there to create a persistent IP address for your instance at this point.
- One or more security lists, if you want to use security rules to control access to your instance. You have already created a security list. You can reference that security list in your orchestration.

If any object referenced in an orchestration is not available when you start the orchestration, the orchestration will go in to an error state. So ensure that you have created all the objects associated with the instance, and that the objects are available to the instance. For example, if you have specified a storage volume, “Vol1”, to be attached to your instance, you should have created Vol1 and it should not be attached to any other instance.

Steps to Create an Instance Using an Orchestration

1. Build your orchestration using JSON.
2. Upload the orchestration to Oracle Compute Cloud Service.
3. Check that the prerequisite objects are available.
4. Start the orchestration.

Everything's Fine. Can I Start My Orchestration?

1. Go to the web console.
2. Click the Orchestrations tab.
3. Go to the orchestration that you want to start. From the menu, select Start.

To start the orchestration:

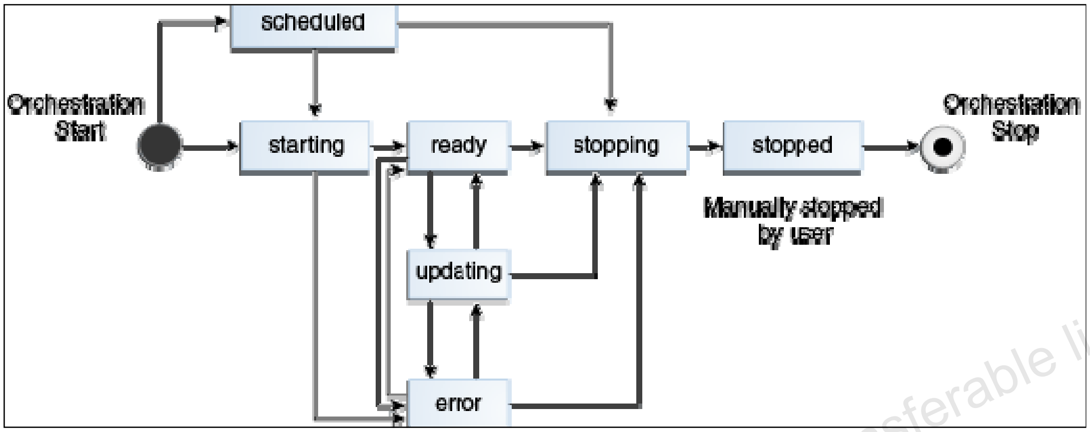
1. Go to the web console.
2. Click the Orchestrations tab.
3. Go to the orchestration that you want to start. From the menu, select Start.

When you start an orchestration, its status changes to Starting and the objects defined in the orchestration are provisioned. When all the objects have been created, the status of the orchestration changes to Ready.

If the orchestration cannot create an object, its status changes to Error. An orchestration might transition from the Error to the Ready state when it completes creating all the specified objects. However, if the status of your orchestration continues to show Error for a long time, then stop the orchestration, identify and fix the issue in an offline copy of the orchestration JSON file, upload the modified orchestration file, and try to start the orchestration again.

How Do I Know What My Orchestration Is Doing?

Orchestration Life Cycle



To check the status of your orchestration:

1. Go to the Oracle Compute Cloud Service web console.
2. Click the Orchestrations tab. All orchestrations are displayed, with information about their description and status.

You can filter the list of orchestrations according to their category or status. To view orchestrations with a specific status (such as ready, error, or stopped), click the Show menu and select the appropriate filter. To view orchestrations of a specific category (such as all or personal), click the Category menu and select the appropriate filter.

An orchestration can be in any of the following states:

Starting

The orchestration is starting.

scheduled

A future start_time has been specified for the orchestration.

- When the current time is equal to or past the start_time value, then the state of the orchestration changes to starting.
- To cancel a current schedule, stop the orchestration. The state of the orchestration then changes to stopping.

ready

The orchestration is running.

- Note that, for any object where the HA policy is not specified or is set to none, you can still update or delete the object using the web console or the API. In this case, the orchestration continues to be in the ready state, even though some or all of the objects created using that orchestration may have been deleted.
- For instances where the HA policy is set to active, if the orchestration is in the ready state, you can update the instance using the web console or the API, but you cannot delete the instance, because it is re-created automatically. To delete such instances, you must stop the orchestration.

updating

The orchestration is being updated.

- When an orchestration is in the ready or error state, you can update it by using the PUT /orchestration/name API call. This causes the state of the orchestration to change to updating.
- When an orchestration is in the updating state, no further updates can be made. Attempts to update such an orchestration are rejected with a validation error.
- If an orchestration in the updating state encounters an error, its state changes error. If no errors are encountered, then the orchestration completes the updates and returns to the ready state.
- When you stop an orchestration that is in the updating state, it transitions to the stopping state.

error

One or more instances in the orchestration have encountered an error.

- The orchestration remains in the error state until all the instances defined in it are running.
- Wait to see if all the instances start running and the state of the orchestration changes automatically to ready. If that does not happen, then stop the orchestration, identify and fix the error, and start the orchestration again.

stopping

The orchestration is stopping.

stopped

The orchestration has stopped. All the objects defined in the orchestration have been deleted.

You can also view details of an orchestration, including return parameters and information about errors, if any. Go to the orchestration that you want to view and, from the menu, select View. The orchestration details page shows you the details of the current state of the orchestration in JSON format.

How Do I Know If My Instance Is Running?

1. Go to the web console.
2. The Instances page shows a list of instances, along with information about each instance.
3. Go to the instance that you want to view. From the menu, select View.

When your orchestration has transitioned to the Ready state, your instance should be running. To verify this:

1. Go to the web console
2. The Instances page shows a list of instances, along with information about each instance. Verify that the instance defined in your orchestration is listed with the status Running.
3. To view the details of this instance, go to the instance and from the menu, select View.

The instance details page shows all the details of the selected instance, such as the public and private IP addresses, and the storage volumes, security lists, and SSH keys associated with it. Verify that your instance was created by the orchestration that you just started, that it uses the appropriate machine image and shape, and that it has the expected resources associated with it, such as the SSH public key, IP address, security lists, storage volumes, and so on.

If there are many instances in your environment, you can filter the list of instances according to their category or status. To list instances with a specific status (such as running, error, or stopped), click the Show menu and select the appropriate filter. To view instances of a specific category (such as PaaS, IaaS, or personal), click the Category menu and select the appropriate filter.

My Orchestration Status Says Ready. Can I Stop It?

1. Go to the Oracle Compute Cloud Service console.
2. Click the Orchestrations tab.
3. Identify the orchestration that you want to stop. From the menu, select Stop.

Remember:

When you stop an orchestration, all the instances and other resources that were provisioned by that orchestration are deleted.

When you stop an orchestration, all the instances and other resources that were provisioned by that orchestration are deleted. So stop an orchestration only if you are sure you do not need any of the objects defined in that orchestration.

To stop an orchestration:

1. Go to the web console.
2. Click the Orchestrations tab.
3. Identify the orchestration that you want to stop. From the menu, select Stop.

When you stop an orchestration, only the resources that are created by the orchestration are deleted. For example, if you use an orchestration to create storage volumes and attach them to your instances, then such storage volumes are deleted when you stop the orchestration, and you lose the data stored on those storage volumes. However, if an orchestration specifies only *attachments* to storage volumes, where the storage volumes themselves are created outside the orchestration, then when you stop the orchestration, the storage volumes are not deleted.

After an orchestration is stopped, its status on the Orchestrations page changes to Stopped. You can still view the orchestration, download it, or start it again later.

Quiz

Which of the following can you do with orchestrations?

a. Create, delete, and re-create multiple instances and other objects with varied attributes
b. Enable high-availability for instances
c. Specify dependencies between objects
d. Schedule the creation or deletion of a set of objects
e. All of the above

Quiz

Which of the following objects can you create or start using orchestrations?

a. Instances
b. Storage volumes
c. SSH keys
d. Security lists and security rules
e. Security applications
f. Other orchestrations
g. All of the above

Quiz

Identify the correct sequence for creating an instance using an orchestration.

a. Generate JSON script > Upload the file using the web console > Create instance > Create other objects such as storage volumes and security lists > Start the orchestration.

b. Generate JSON script > Add orchestration > Start orchestration > Create other objects such as storage volumes and network settings > Create instance.

c. Generate JSON script > Upload the file using the web console > Verify prerequisites > Start the orchestration.

ORACLE

Quiz

What happens when you stop an orchestration?

a. All the objects defined in the orchestration are deleted.

b. All the objects defined in the orchestration are stopped but not deleted.

c. The instances defined in the orchestration are stopped, and other objects are deleted.

Summary

In this lesson, you should have learned how to:

- Use orchestrations
- Explain the structure of an orchestration
- List the objects that you can create using an orchestration
- Create an orchestration file using JSON
- Create an instance using an orchestration
- Delete an instance by stopping an orchestration

ORACLE

Resource: Links

For more information regarding Oracle Compute Cloud Service, visit http://docs.oracle.com/cloud/latest/stcomputecs/index.html

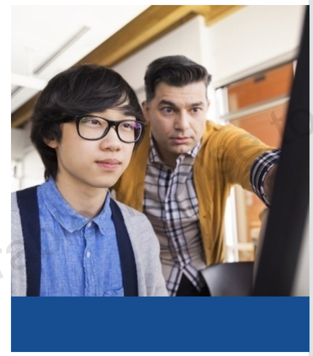
13

HTTP and REST Fundamentals

Objectives

After completing this lesson, you should be able to:

- Define HTTP
- Describe the steps in an HTTP transaction
- List the key HTTP headers in request and response
- List the HTTP Methods
- Define the structure of a URL and a URI
- Describe a REST Web Service
- Describe the JSON format



Topics

- **The Internet and World Wide Web**
- HTTP (HyperText Transfer Protocol)
- REST: Introduction
- REST in Action

A Brief History of the Internet and World Wide Web

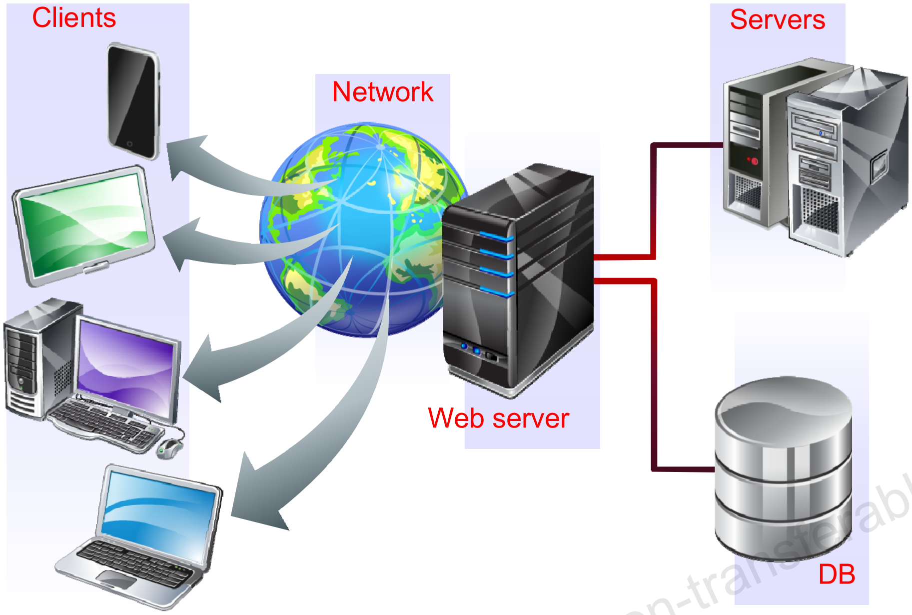
- Internet
  - ARPANET
  - First two nodes connected October 29, 1969
- World Wide Web
  - Tim Berners-Lee released proposal: November 1990
    - HTTP and HTML
  - Netscape released first web browser: November 1994
  - Google incorporates: 1998
  - First Facebook site launched: Feb 2004
  - iPhone unveiled: Jan 2007
- Internet, Web, and mobile phones are fundamental to life
  - But how does it all work?

ORACLE

- First two nodes of what would become the ARPANET were interconnected between UCLA (Los Angeles) and SRI International in Menlo Park, CA on 29 October 1969.
- Tim Berners-Lee published a formal proposal for HTML and HTTP in November 1990. He proposed building a "Hypertext project" he called "WorldWideWeb" as a "web" of "hypertext documents" to be viewed by "browsers" using a client–server architecture.
  - **HTTP:** HyperText Transfer Protocol
  - **HTML:** HyperText Markup Language
- Netscape released the first commercial version of its browser “Mosaic Netscape 0.9” on October 13, 1994.
- Mark Zuckerberg launched his first version of “thefacebook” on Februrary 4, 2004.
- iPhone unveiled by Steve Jobs at MacWorld January 9, 2007.

Note: All information referenced from Wikipedia

How Does It All Work?



On the web, clients connect to servers by using a network, commonly the Internet. Servers can be connected with one another to request information to complete requests from clients.

What Are Web Servers?

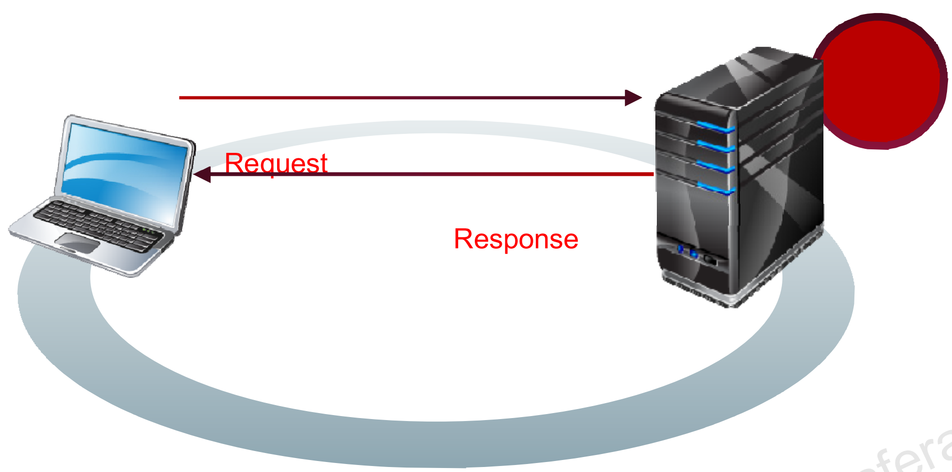
Web applications are usually stored in web servers. They:

- Handle web requests
- Store application files
- Provide access to resources, such as:
  - JSON data
  - HTML pages
  - Dynamic pages
  - Images
  - Video

ORACLE

- A web server is software that listens for requests, provides access to resources, and generates a response.
- A web server can receive requests from multiple clients, and handles them separately.
- When a request is received, the server locates the resource, and sends its contents back to the client. Typically, the response is an HTML file, which the browser displays. In some cases, requests are forwarded to other services that construct the content of the response.
- Dynamic content can be generated in the server or in the client. Web applications use dynamic content in the client with JavaScript and in the server by using an application server.

How Do Web Servers Work?



Web servers work in three stages:

1. A request is always initiated by the client.
2. The server locates the resource and constructs a response.
3. The response with the content of the resource is sent back to the client.

Note that these three steps can be repeated multiple times.

What Is a Client?

A client is an application that runs on a machine that can make requests from a web server and HTML response data,

- **Web browsers**
  - Firefox, Chrome, Internet Explorer, and Safari
- Other applications
  - Weather app on mobiles
  - News readers

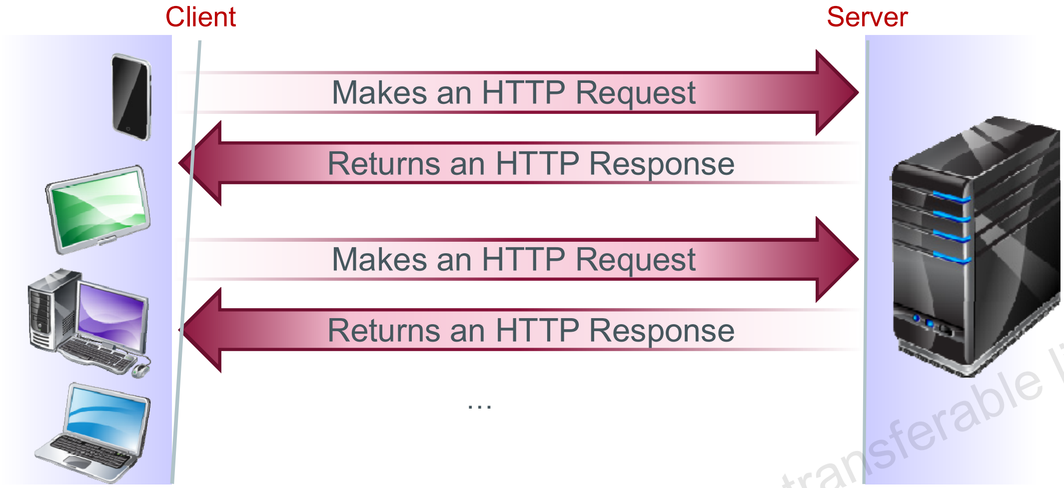
In this course, client applications are created using HTML pages run in your computer.

Topics

- The Internet and World Wide Web
- **HTTP (HyperText Transfer Protocol)**
- REST: Introduction
- REST in Action

What Is HTTP?

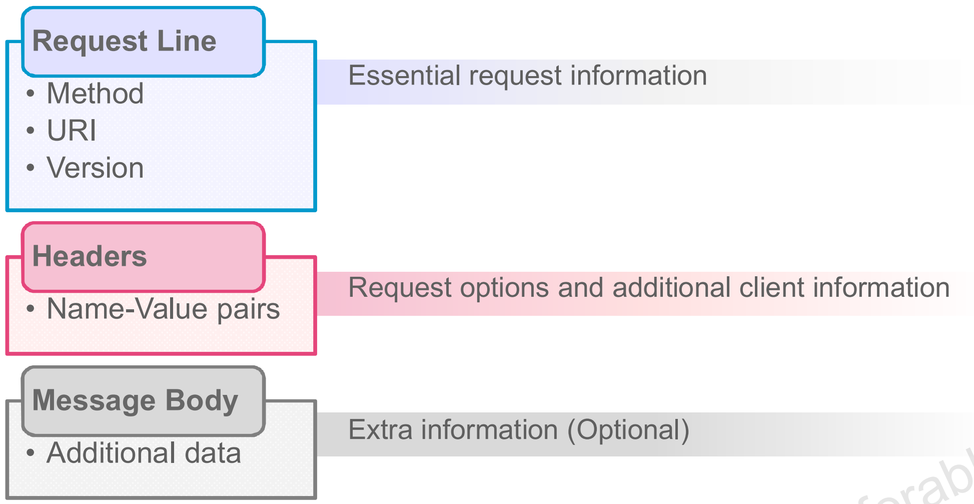
Clients communicate with the server by using HTTP (HyperText Transfer Protocol).



ORACLE

- HTTP is a communication protocol, which relies on TCP connections over IP to connect the client and the server.
- In the HTTP protocol, all requests are independent.
- The client might trigger additional requests, but each one of these requests is completely new to the server and is handled independent of previous requests.
- There is no session management as part of the HTTP protocol.
- HTTP is stateless.
- Session management is accomplished by sending an identifier as part of the request that can relate to a specific session.

HTTP Request



A request contains some information that helps the server generate a response.

Every request contains a Request Line. This contains the Uniform Resource Identifier (URI), which is the name of the requested resource, and the Method used. HTTP defines the following methods: `OPTIONS`, `GET`, `HEAD`, `POST`, `PUT`, `DELETE`, `TRACE`, and `CONNECT`. Web browsers use only `GET` or `POST`.

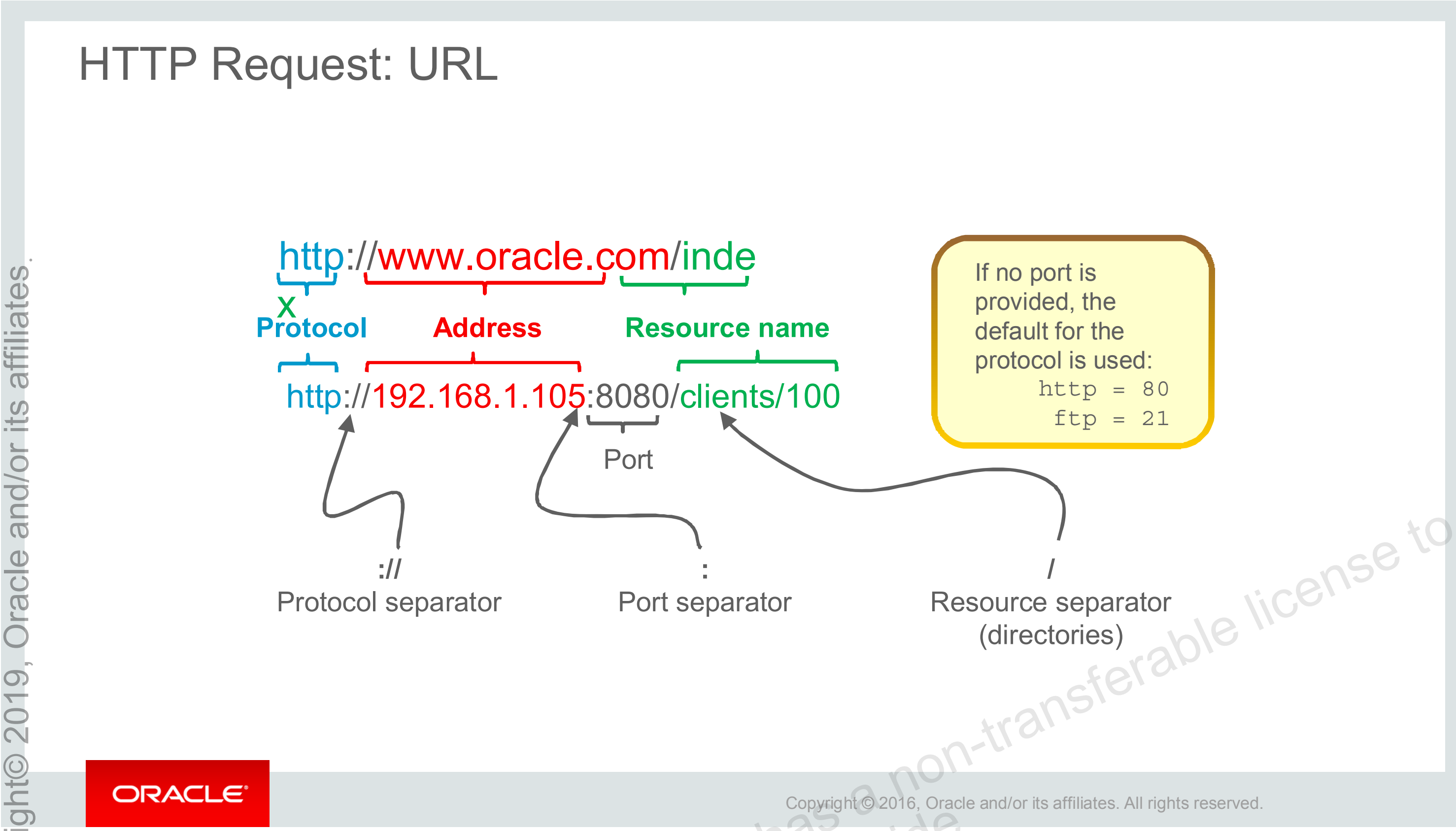
The HTTP request headers are key-value pairs that are used to provide additional information.

- `accept-encoding`: Defines what encoding is expected in the response
- `user-agent`: Specifies the client application being used. When you are using a browser, the name and version of the browser are supplied in this header.
- Session IDs and session information, such as cookies, can be contained in request headers.

For more information about HTTP headers, refer to http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html.

The Message Body can contain additional information supplied by the user, such as form data when using the POST method.

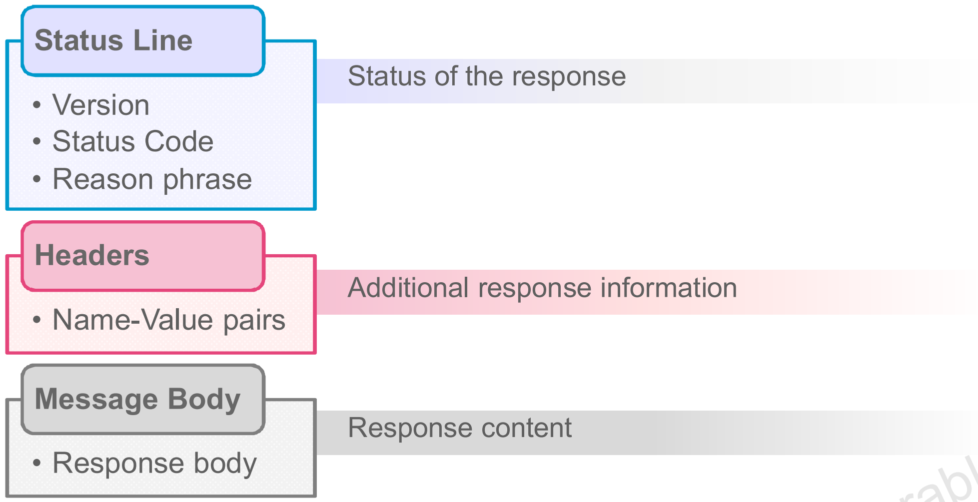
HTTP Request: URL



The slide shows the different components that compose a URL.

Remember that if no port is specified, the default is used. For HTTP, the default port is 80; for FTP, it is 21.

HTTP Response



The HTTP response must contain the status line.

Status codes are grouped in the following ways:

- **1xx:** Informational, for example, 101 Switching Protocols
- **2xx:** Success, for example, 200 OK
- **3xx:** Redirection. Further action is required to complete the request, for example, 301 Moved Permanently.
- **4xx:** Client Error. The request contains a problem and could not be fulfilled, for example, 404 Not Found.
- **5xx:** Server Error. There was a problem in the server, for example, 500 Server Error.

You can find the list of response codes at http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

Response headers contain additional information about the response, such as the content type, content size, user cookies, session ID, and so on.

Between the headers and the body, there needs to be an empty line.

The body contains the actual content of the requested resource. The response content is optional.

Response Bodies

A response body contains the content of a resource, including:

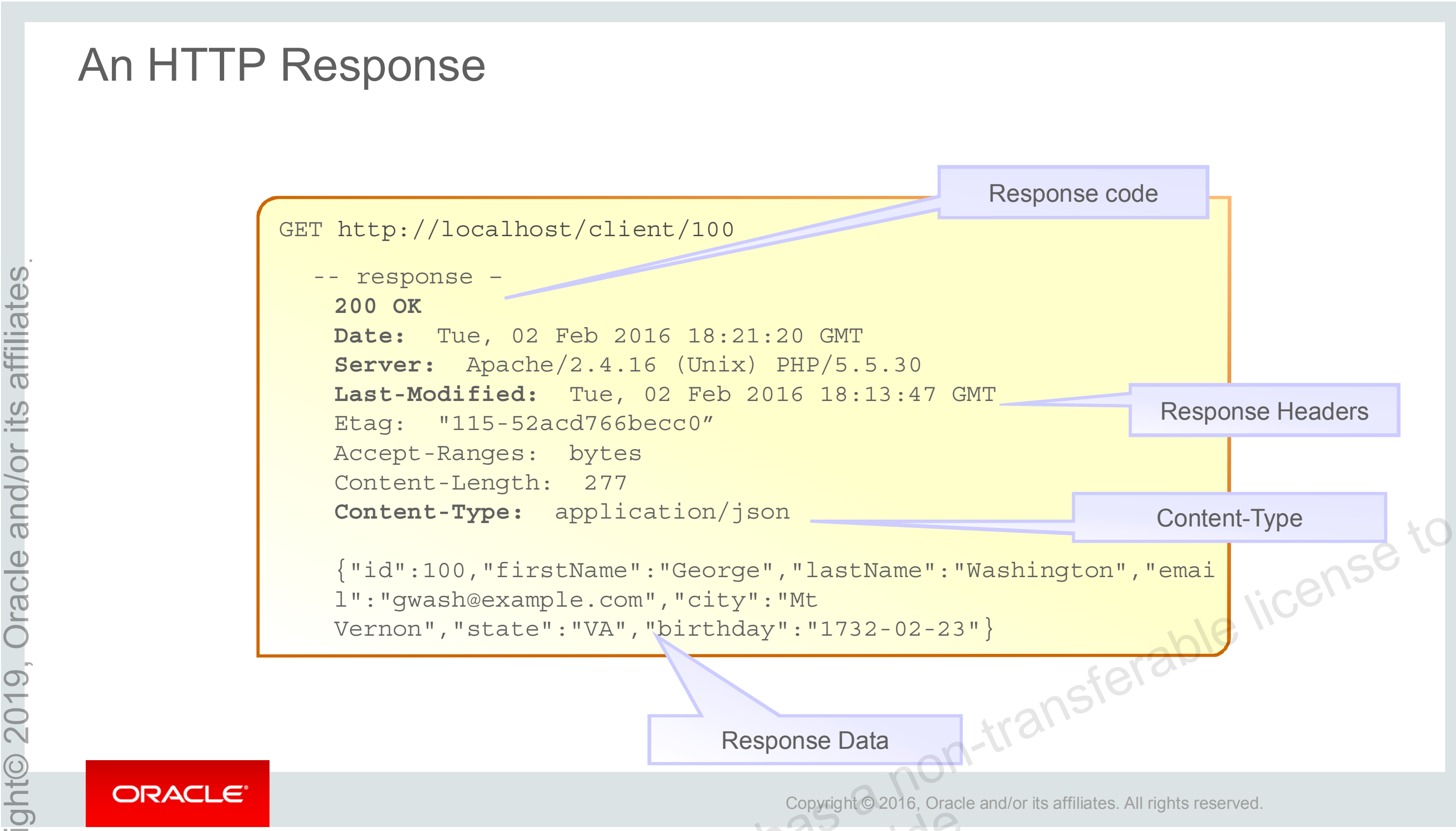
- JSON data *
- Documents
- Images
- Audio
- Video
- JavaScript files

In summary, the response body contains the requested data.

The response body contains the content of the requested resource.

The client uses the response content to display information on the screen.

*JSON is a form of representing data as JavaScript literals.



An HTTP response includes header information along with the body.

- The body includes JSON data.
- Notice that a response code of 200 is returned, which indicates that the HTTP transaction returned OK.
- The Content-Type header indicates what is included in the HTTP response.

Topics

- The Internet and World Wide Web
- HTTP (HyperText Transfer Protocol)
- **REST: Introduction**
- REST in Action

What Is REST?

REST is the architecture that the entire web uses. It is based on the following design principles:

- Client/server interactions
- Uniform interface
- Layered system
- Cache
- Stateless
- Code-on-demand

Representational State Transfer (REST) is a term that has existed for a long time. Roy Fielding describes REST in his doctoral dissertation at UC Irvine, and defines the web architecture for servicing multiple clients in a flexible, scalable, and uniform way.

REST takes full advantage of the HTTP protocol. The complete HTTP protocol complies entirely with the REST proposed architecture:

- Multiple and diverse clients exist in the form of web browsers and applications.
- Uniform Resource Locators and HTTP methods are present to access and modify resources.
- It can be a system that may contain many layers in between the client and the server.
- Caches can be implemented on resources because often the same operations performed over the same resources produce the same result.
- Stateless operations establish no sequence or dependency in REST calls.
- Finally, servers may provide code that needs to be executed. Most often, instead of providing code, the server provides further possible operations in the form of HyperMedia in such a way that clients know what REST operations are possible.

How Do I Use REST on the Web?

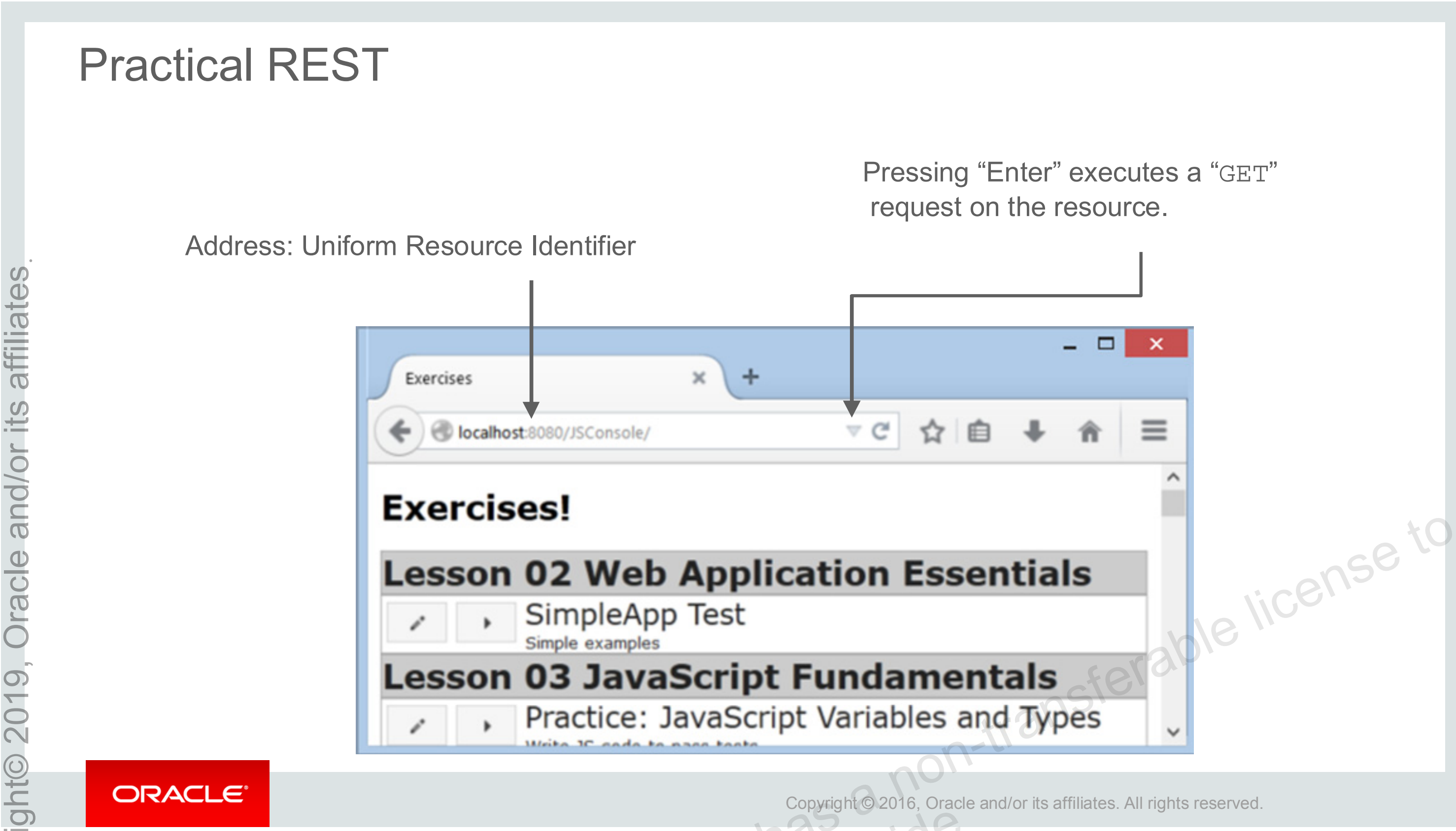
You are already using REST.

- Client/server interactions: Browser – Web Server
- Uniform interface: HTTP, URL
- Layered System: IP: Proxies, Gateways, and so on
- Cache: Web browser cache, intermediary caches
- Stateless: HTTP request-and-response model
- Code-on-demand: JavaScript code

ORACLE

The URI and Method represent the resource to be accessed or modified.

Use the request headers to provide additional information about the request. If needed, add a `requestBody` to provide data in the request.



In the example in the slide, REST begins in the web browser when you enter an address and press Enter.

The browser makes an HTTP `GET` request to get the resource from a server that is specified by the Uniform Resource Identifier provided in the address bar.

Uniform Resource Identifier or Uniform Resource Locator

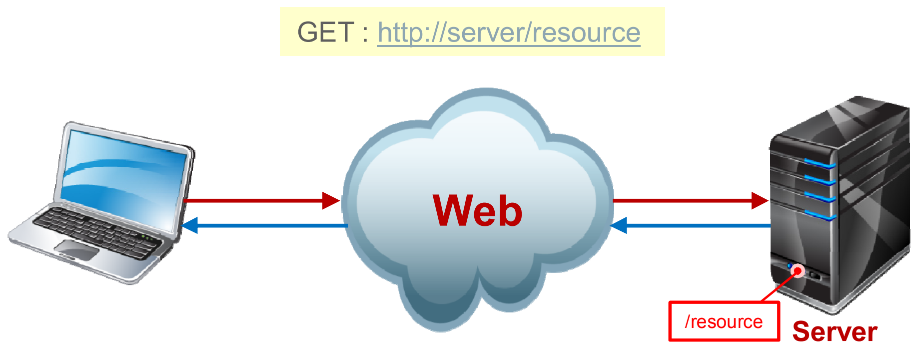
- A URI is composed of the name of the resource, but does not specify the way to access such resource.
- A URL has the protocol or way to get a particular resource.

In the end, all URLs are also URIs but the term URI is more widely accepted when referring to addresses because sometimes the protocol is omitted, for example:

- http://www.oracle.com/education is a valid URL and URI.
- www.oracle.com/education is a valid URI only.

Thus, using URI for both is correct and more “universal.”

Practical REST



The request is sent through the network to the server, which locates the resources and applies the method. In the example in the slide, the method is a `GET` request. Thus, the server will create a representation of the resource and send it back to the client.

Can I Use REST from the Command Line?

- Use cURL:
  - Is an open source command-line tool
  - Enables user authentication, secure connections

Example:

```
curl -v -X PUT \ -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b"
    \ https://foo.storage.oraclecloud.com/v1/Storage-
    myIdentity3/FirstContainer
```

ORACLE

cURL is an open source, command-line tool for transferring data with URL syntax, supporting various protocols including HTTP and HTTPS.

In the example in the slide, cURL is used to create a container, FirstContainer, in the Oracle Storage Cloud Service, using the authentication token specified in the same command. The –v option specifies a verbose output, and the –X option specifies the method to be used, in this case `PUT`. You will learn more about HTTP methods later in this lesson.

Topics

- The Internet and World Wide Web
- HTTP (HyperText Transfer Protocol)
- REST Introduction
- **REST in Action**

ORACLE

Exchanging Data with REST

To create REST applications, you need to define a way to exchange information between clients and servers.

- HTML provides no easy way to identify data types.
- Typically, the choice is between two text formats.

Extensible Markup Language (XML)

- A text file format that marks up text using tags
- Designed to be both human-readable and machine-readable

JavaScript Object Notation (JSON)

- A text file format that stores data in attribute value pairs
- Designed to be both human-readable and machine-readable

ORACLE

Extensible Markup Language (XML)

A text file format that marks up text using tags. For example:

```
<root>
     <name>John Doe</name>
     <email>jdoe@example.com</email>
</root>
```

JavaScript Object Notation (JSON)

A text file format that stores data in attribute value pairs

```
{"name":"John Doe", "email":"jdoe@example.com"}
```

Both formats are designed to be human-readable and machine-readable.

XML Versus JSON in Web Services

Both formats have advantages and disadvantages.

- **Extensible Markup Language (XML)**
  - A large number of processing libraries exist to support XML in almost every language (SAX, DOM, StAX, and JAXB).
  - Representation of data is verbose.
  - May require additional system resources to parse and process
- **JavaScript Object Notation (JSON)**
  - A subset of JavaScript
  - Less verbose than XML
  - More lightweight to transport and process
  - Support evolving in other languages

JSON has become a much more popular option given its compactness and ease to process.

What Is JSON?

JavaScript Object Notation is a simple format to represent objects, arrays, and values as strings.

JavaScript includes functions to convert objects, arrays, and values to JSON strings and vice versa.

http://json.org/

ORACLE

The JavaScript Object Notation (JSON) is a format to represent JavaScript objects, arrays, and values as strings. It is one of the preferred ways of sending and receiving data using REST web services.

JSON allows you to represent objects as key-pair values and arrays as sequential lists of items.

Values can also be represented as JSON, allowing you to create all sorts of objects by using strings, numbers, and Booleans.

JSON is often compared to XML because both can be used to represent structured data. The main difference between JSON and XML is that JSON contains only data without any schema information. Therefore, JSON has no direct validation mechanism.

Also JSON contains only string, number, and Boolean value types and Object and Array data structures. You do not need to define custom data types or nodes as in XML. In JSON, data is represented as it would be inside a JavaScript object, making it extremely flexible at the exchange of validation and schema facilities.

JSON: Overview

JSON is the string (text) representation of JavaScript objects and values.

- JSON is very similar to declaring JavaScript literal values.
- JSON does not have a representation for functions.
- Only values are allowed in JSON.

JSON values are represented in the following manner:

- **Numbers:** Use the number literal.
- **Strings:** Enclose the string literal inside double quotation marks.
- **Booleans:** These are either true or false.

JSON structures are represented in the following manner:

- **Arrays:** Must be enclosed in square brackets "[]" and elements must be comma-separated. They can contain arrays, objects, numbers, strings, or boolean as elements.
- **Objects:** Must be enclosed in curly brackets "{}." Properties are comma-separated key-value pairs. The key for the property must be a string; therefore, it must be enclosed in double quotation marks. Values for properties can be arrays, objects, numbers, strings, or boolean.

There is no way to represent functions in JSON because it is used to represent only data.

JavaScript Literals in JSON

Numbers
- 1
- 1000
- 20.39

Strings
- “string”
- “another String”

Boolean
- true
- false

Arrays

[element1, element 2, …, elementN]

[1, 2, 3 ,4 5]
[“a”,”b”,”c”]
[“a”,1,”c”,4]

Objects

{ “ProperyName”: propertyValue }

{"name":"john","age":31}

{"itemId":["a","b"], "total":2, "active":true}

{ "id":100, "firstName":"George", "lastName":"Washington", "email":"gwash@example.com", "city":"Mt Vernon", "state":"VA", "birthday": "1732-02-23" }

ORACLE

An object property value can be any other JSON literal value including even another object.

Customer example shows JSON data:

- Note that the ID is shown as a number.
- Notice pairs of fieldname/value.
- All other data pairs are strings.

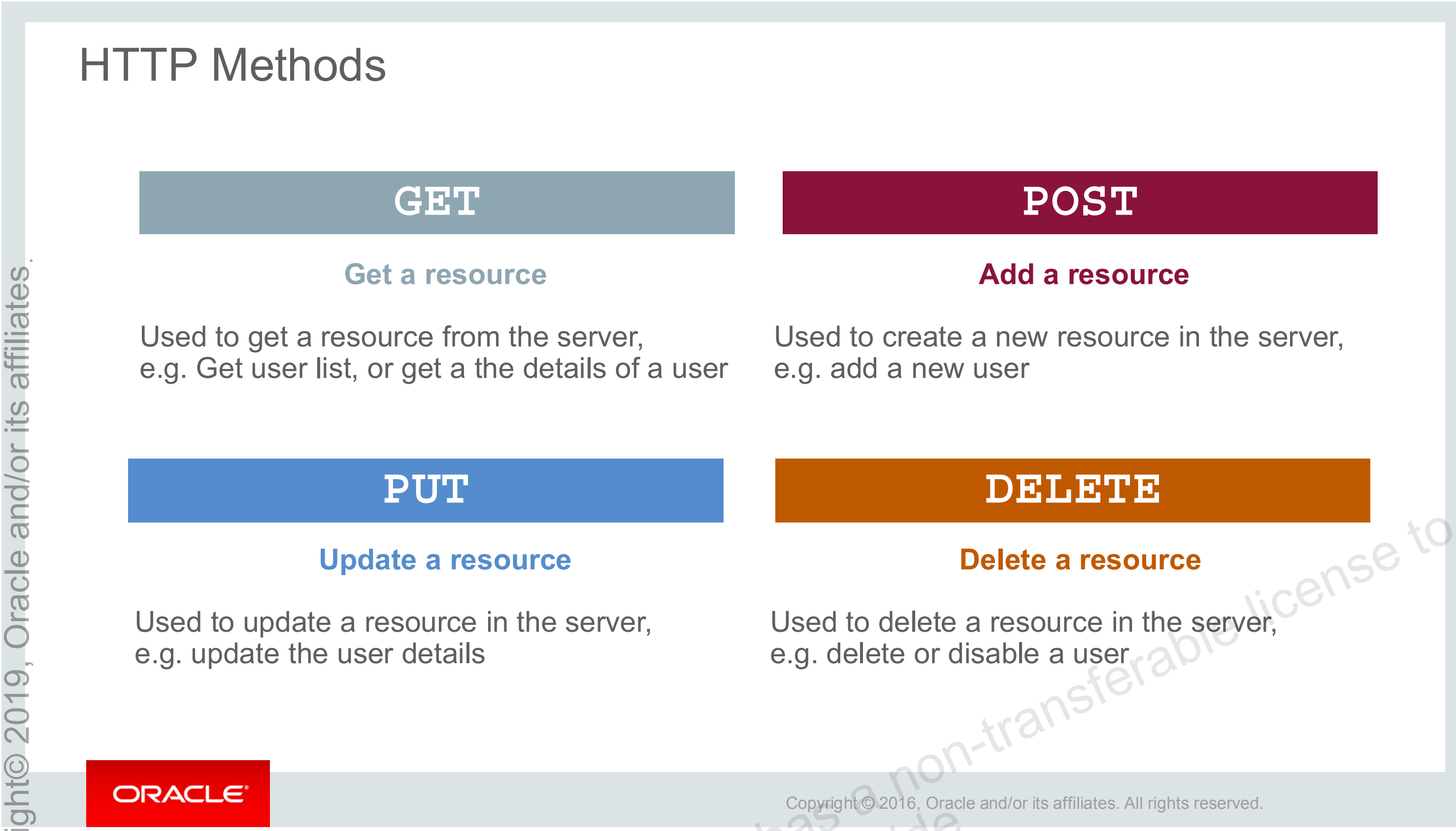
What Do I Need to Know to Use REST?

To use REST in web applications, you need to address the following:

- The HTTP method that you need
- The resource identified by its URI
- The content of the request and parsing of the response accordingly

To know how to use a particular REST service, you need to know what resources you need to access, what methods you can ask for, and what data is going to be exchanged.

The preferred format for the data sent from and to servers is JSON, although the specific format is service dependent.



- `GET` is used to obtain the resource.
- `PUT` is used to update the resource.
- `POST` is used to add a new resource.
- `DELETE` is used to remove a resource.

HTTP defines the `OPTIONS`, `HEAD`, `TRACE`, and `CONNECT` methods as well but their usage in RESTful APIs is not as wide.

Sometimes, you will find the `OPTIONS` method returning a representation of the possible uses of the other methods as well as the URI compositions available. This is not standard and should not be relied upon.

RESTful URI

URIs should describe access to a resource, for instance.

- GET http://www.example.com/users/.
  - Lists all the users
- GET http://www.example.com/users/john.
  - Gives the details of a user identified as “john”
- POST http://www.example.com/users/.
  - Creates a new user
- PUT http://www.example.com/users/john.
  - Updates the user identified as “john”

ORACLE

In the examples in the slide, you can see the common structure of the RESTful API’s URIs.

URIs are usually constructed progressively as you need more details.

For an application that stores places and reviews of such places:

- `/nearby/places`: Refers to the places that are nearby to the user
- `/places`: Refers to the list of places registered
- `/places/toms_pizza`: Refers to the place called Tom’s Pizza
- `/places/toms_pizza/reviews`: Refers to the reviews of the place called Tom’s Pizza

You can see how methods would apply:

- `GET /nearby/places`: Gets a list of the nearby places
- `POST /places`: Adds a new place
- `PUT /places/toms_pizza`: Updates the place called Tom’s Pizza
- `DELETE /places/toms_pizza`: Deletes the place called Tom’s Pizza
- `DELETE /nearby/places`: NOT ALLOWED

Quiz

Which of the following are valid HTTP methods?

a. GET
b. PUSH
c. UPDATE
d. POST
e. DELIVER
f. DELETE
g. All of the above

Quiz

Which of the following are true of REST

a. Uses HTTP
b. Is used by web browsers
c. Provides architecture used by the web
d. Performs stateless operations
e. All of the above

Quiz

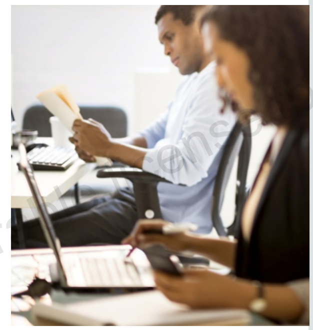
What is JSON?

a. An alternative to XML
b. A machine-readable script that is not human-readable
c. A subset of Java Script
d. Defines key-value pairs
e. Used to represent data and functions
f. All of the above

Summary

In this lesson, you should have learned how to:

- Define HTTP and HTML
- Describe the steps in an HTTP transaction
- List key HTTP headers in requests and responses
- List the HTTP Methods
- Define the structure of a URL and a URI
- Describe a REST Web Service
- Describe the characteristics of JSON



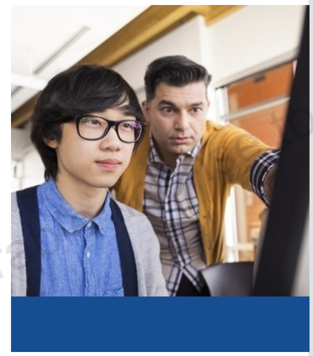
14

Oracle Storage Cloud Service Access and Authentication

Objectives

After completing this lesson, you should be able to:

- Access object storage on Oracle Cloud
- Define Access Control Lists (ACLs)
- Protect data stored in object storage on Oracle Cloud using ACLs



ORACLE

How Do I Authenticate Access to Using REST API?

1. Request an authentication token.
2. Construct the authentication URL for your account.
3. Execute the cURL request commands:
   – `GET`
   – `PUT`

The following are the three main steps involved in accessing object storage on the cloud via the REST API:

1. Request an authentication token.
2. Construct the authentication URL for your account.
3. Execute the cURL commands.
   - `GET`
   - `PUT`

Note: If you do not already have cURL installed, refer to the lesson titled "Introduction to Oracle Storage Cloud Service" for instructions on how to obtain and install it.

How Do I Request an Authentication Token?

- Execute the cURL command:
  - `GET` request

```
curl -v -X GET \
     -H "X-Storage-User: myService-myIdentity3:myUsername" \
     -H "X-Storage-Pass: myPassword" \
     https://foo.storage.oraclecloud.com/auth/v1.0
```


- If your user credentials are not authenticated, the service returns an HTTP response with a status code of 401 and no authentication token is returned.
- This slide shows an example of a cURL command for requesting an authentication token using `GET`.
- The following slide shows the output.

Output

- Output of the GET request:

```
> GET /auth/v1.0 HTTP/1.1
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Storage-User: myService-myIdentity3:myUsername
> X-Storage-Pass: myPassword

< HTTP/1.1 200 OK
< X-Storage-Url: https://foo.storage.oraclecloud.com/v1/myService-myIdentity3
< X-Storage-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
< X-Trans-Id: txba4aa8f776164c33b7aa587554c29fb6
< Content-Length: 0
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Type: text/plain
< Content-Language: en
```


- To use your authentication token, include it as the value of the `X-Auth-Token HTTP` header in every `HTTP` request to the service instance.
- If your authentication token is not valid, or has expired, the service returns an `HTTP` response with the status code 401 and the requested operation will fail.
- If the authentication token has expired, you must request a new token.
- If you are reading publicly accessible objects, you do not need to provide an authentication token in your `HTTP` request; anonymously accessible objects do not need an authentication token.

Now How Do I Construct the Authentication URL for My Account? (1/2)

1. Sign in to the **Oracle Cloud My Services** application.
   – The **My Services** dashboard is displayed. It lists the services that are assigned to your account.
2. Look for **Oracle Storage Cloud Service**.
3. Select **View Details** from the **Actions** menu.
4. Alternatively, click the **Oracle Storage Cloud Service** link on the **Dashboard** page.
   – On the resulting page, the details of your Oracle Storage Cloud Service instance are displayed.

ORACLE

Now How Do I Construct the Authentication URL for My Account? (2/2)

5. Note the REST Endpoint URL, which is displayed in the **REST Endpoint** field under the **Additional Information** section.
   - For example: `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
6. Delete the following portion of the REST Endpoint URL: `v1/Storage-myIdentity3`
   - Now, the edited URL should be: `https://foo.storage.oraclecloud.com/`
7. Append the following to the edited URL: `auth/v1.0`
8. Assuming that the REST endpoint URL for your account is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`, the equivalent authentication URL would be `https://foo.storage.oraclecloud.com/auth/v1.0`

Done. What's Next?

Constructed authentication URL

- When sending the `GET` request you must include user credentials in the following headers:
  - `X-Storage-User`

    Syntax for metered subscriptions:
    `X-Storage-User: Storage-identityDomainID:username`

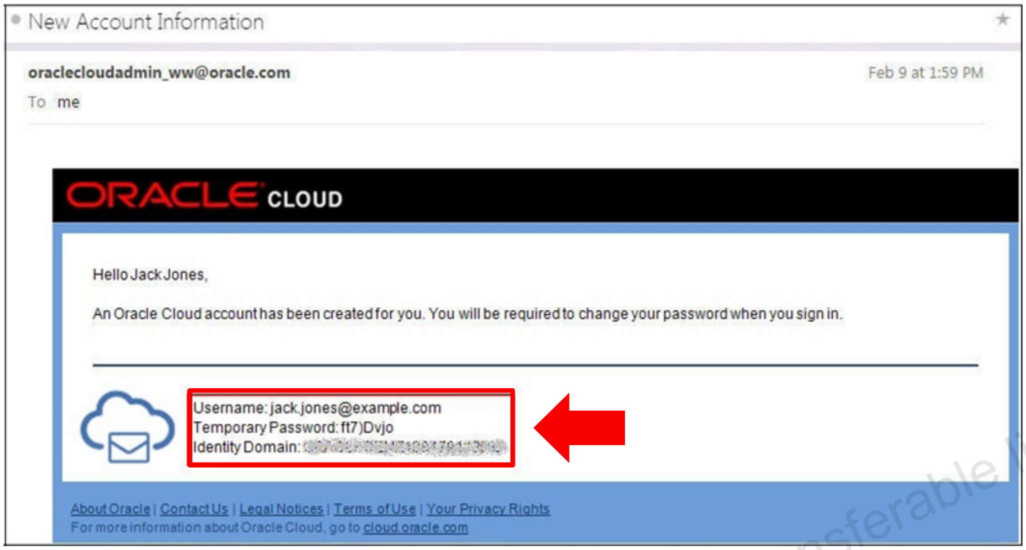
    Syntax for nonmetered subscriptions:
    `X-Storage-User: serviceInstanceName-identityDomainID:userName`

  - `X-Storage-Pass: password`

When you send the `GET` request to the authentication URL that you just constructed, include the user credentials in the following headers:

- `X-Storage-User` has two types of subscribers:
  - **Syntax for <u>metered</u> subscriptions:** `X-Storage-User: Storage-identityDomainID:username`
  - **Syntax for <u>nonmetered</u> subscriptions:** `X-Storage-User: serviceInstanceName-identityDomainID:username`
- `X-Storage-Pass: password`

Where Can I Find My User Information?



- You can find out your username, password, and identity domain from the "*New Account Information*" email that you received from Oracle Cloud when your account was set up, as shown in the slide.
- If you do not have your "*New Account Information*" email, ask your account administrator for your Oracle Cloud username, password, and identity domain.

How Do I Store an Object?

- Storing an object in an account using an authentication token
- Execute the cURL command:
  – PUT request

```
curl -v -X PUT \
    -H "X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6" \
    -d "Hello, World!" \
    https://foo.storage.oraclecloud.com/v1/myService-
 myIdentity3/myContainer/myObject
```

ORACLE

- The slide shows an example of a cURL command for storing an object in an account using an authentication token using PUT.
- The following slide shows the output.

Output

Output of the PUT request:

```
> PUT /v1/myService-myIdentity3/myContainer/myObject HTTP/1.1
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk209f7f2ea1265a0d3f29d28a2dc8ced6
> Content-Length: 13
> Content-Type: application/x-www-form-urlencoded

< HTTP/1.1 201 Created
< Content-Length: 0
< Etag: 65a8e27d8879283831b664bd8b7f0ad4
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx287a1a8e33cc45e5a1431817e3e87621
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
```


This shows an example of the output of this command.

Using Access Control Lists (ACLs) to Protect Data

- User privileges
  - Permissions based on user roles
    - `X-Container-Read`
    - `X-Container-Write`
- Data protection
  - Implementing Access Control Lists to containers
    - Access to containers and objects can be granted or denied
  - Permission to read and/or write
  - Unique user role: `Storage_Adiministrator`

- The ability to read and write objects in a container is governed by the Access Control Lists (ACLs) assigned to the container. These ACLs are written to two metadata fields: `X-Container-Read and X-Container-Write`.
- Users with roles assigned to these metadata fields can perform the following actions:
  - **`X-Container-Read:`** Users can read objects and associated metadata in the given container.
  - **`X-Container-Write:`** Users can create and delete objects and associated metadata in the given container.
- Data protection is guaranteed because service administrators can grant read or write access to users.
- The metadata field values are a comma-separated list of identity domain ID and role pairs. This allows service administrators to grant read or write access to users in other identity domains. Users with the *Storage_Administrator* role may define their own roles on the My Services Users page and assign them to the `X-Container-Read` and `X-Container-Write` headers on containers, as required.
- Users with the *Storage_Administrator* role will always have read and write access to all containers in their service instance.

So, What Are the Default Values when Creating a New Container?

- Default ACLs assigned to containers
  1. X-Container-Read: identity_domain_ID.storage_service.Storage_ReadOnlyGroup,identity_domain_ID.storage_service.Storage_ReadWriteGroup
  2. X-Container-Write: identity_domain_ID.storage_service.Storage_ReadWriteGroup
- All non-administrators are subject to the ACLs for a given container with the exception of the service instance root path.
  - However, only users with the Storage_Administrator role can create or delete containers.

- By default, when a container is created in the Oracle Storage Cloud Service, the following ACLs are assigned:
  - X-Container-Read: identity_domain_ID.storage_service.Storage_ReadOnlyGroup,identity_domain_ID.storage_service.Storage_ReadWriteGroup
  - X-Container-Write: identity_domain_ID.storage_service.Storage_ReadWriteGroup
- All non-administrator users are subject to the ACLs for a given container.
  - The service instance root path is an exception to this, because it does not have ACLs associated with it.
    - For this path, all users can obtain a list of containers; however, only users with the *Storage_Administrator* role can create or delete containers.

Example: Creating a Container with Default Values

The following are the newly created container ACL values for a service instance named `Storage` in an identity domain named `myIdentity3`:

- `X-Container-Read:`
  `myIdentityDomainID.Storage.Storage_ReadOnlyGroup,`
  `myIdentityDomainID.Storage.Storage_ReadWriteGroup`
- `X-Container-Write:`
  `myIdentityDomainID.Storage.Storage_ReadWriteGroup`

ORACLE

Quiz

What does ACLs stand for?

a. Access Control Lists
b. Access Computer Lists
c. Admin Control Lists
d. Admin Command Lists

Quiz

An authentication token must be requested in order to authenticate access to Storage Cloud Service.

a. True

b. False

Quiz

What request command do you use to store an object?

a. POST

b. PUT

c. GET

d. POST OR PUT

Quiz

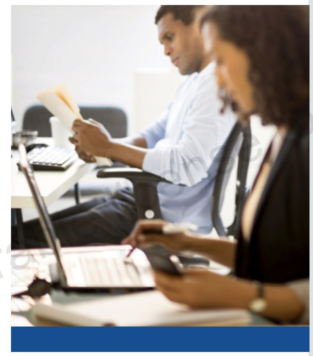
To better protect data from users, what do you implement to grant or deny access permissions?

a. Admin Control Lists
b. Access Control Levels
c. Access Control Lists
d. Admin Control Lists

Summary

In this lesson, you should have learned how to:

- Access object storage on Oracle Cloud
- Define Access Control Lists (ACLs)
- Protect data stored in object storage on Oracle Cloud using ACLs



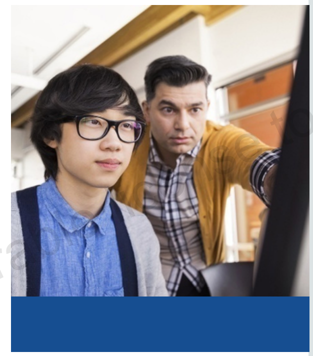
15

Oracle Storage Cloud Service Containers and Objects

Objectives

After completing this lesson, you should be able to:

- Create containers
- List containers
- Set container metadata
- Delete containers
- List objects in a container
- Create objects
- Download objects
- Delete objects
- Update object metadata



Two Sections

SECTION I: CONTAINERS
- Create containers
- List containers
- Set container metadata
- Delete containers

SECTION II: OBJECTS
- List objects in a container
- Create objects
- Download objects
- Delete objects
- Update object metadata

This lesson has two sections.

- **Section I:** Focuses on managing (object storage) *containers* in Oracle Storage Cloud Service
- **Section II:** Focuses on managing (object storage) *objects* in Oracle Storage Cloud Service

Containers

SECTION I: Creating Containers

- cURL command syntax:

```
curl -v -X PUT \
     -H "X-Auth-Token: token" \
     accountURL/containerName
```

- cURL command example:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer
```

ORACLE

You will need to substitute the parameters highlighted in the slide based on your account's information.

Where:

- `token` is the authentication token obtained earlier from Oracle Storage Cloud Service.
- `accountURL` is the URL that you see in My Services when signed in to the Oracle Cloud My Services application—This is the Global namespace URL for all customers.
  - Other type of URLs assigned to customers include: Data center–specific URL, URLs based on metered subscription and nonmetered subscription. (The latter two are URLs in the earlier release of Oracle Storage Cloud Service.)
- `containerName` is the name of the container to be created.
  - The container name can be from 1 to 256 characters long and must consist of alphanumeric characters.

HTTP Response Codes

- `Success: 201 Created`

cURL Command Example:

```
curl -v -X PUT \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

- Where:
  - `Token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> PUT /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 201 Created
< Date: Fri, 06 Mar 2015 10:34:20 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx23a1084b8c674fdeae8d4-0054f982ac
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Listing Containers

- All containers within an account can be listed.
- Any user in the identity domain can perform this task.
- Containers are sorted lexicographically using `memcmp()`.
- Query parameters include:
  - `token`
  - `marker`
  - `end_marker`
  - `format`
    - Return `Xml or json` format (REST API only)

ORACLE

Containers are sorted lexicographically using **`memcmp()`**. All containers, up to 10000 by default, will be returned in the list, unless you filter the list by using any of the following query parameters:

- **`limit:`** Limit the number of containers listed to the specified value. The default and maximum value is 10000.
- **`marker:`** Return containers with names greater than the specified string.
- end_marker: Return containers with names less than the specified string.
- **`format:`** Return extended information about each returned container in either *xml* or *json* format (REST API only).

SECTION I: Listing Containers

- cURL command syntax:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     accountURL[?query_parameter=value]
```

- cURL command example:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3?limit=15
```

ORACLE

- You will need to substitute the parameters highlighted in the slide based on your account's information.
- Parameter not yet introduced:
  - `query_parameter=value` is the optional filtering parameter

HTTP Response Codes

- `Success: 200 OK`

cURL Command Example:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3?limit=15
```

- Where:
  - `token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `query_parameter` **is** `limit=15`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust Network,O=Symantec Corporation,C=US
> GET /v1/Storage-myIdentity3 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk6403794c218a709d1c6c5a76444d01f6
>
< HTTP/1.1 200 OK
< Date: Fri, 06 Mar 2015 10:38:15 GMT
< Content-Length: 109
< X-Account-Container-Count: 3
< Accept-Ranges: bytes
< X-Account-Object-Count: 843
< X-Account-Bytes-Used: 10304761355
< X-Timestamp: 1412823447.62495
< X-Account-Meta-Test5: test1
< X-Account-Meta-Quota-Bytes: 107374182400
< Content-Type: text/plain; charset=utf-8
< X-Account-Meta-Test: test
< X-Account-Meta-Test1: test1
< X-Trans-Id: tx29052c64fe384fc690ccc-0054f98397
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
FirstContainer
hello
lab
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Setting Container Metadata

- Only the following metadata can be changed in a container:
  - Container ACLs: `X-Container-Read` and `X-Container-Write`
  - Quotas: `X-Container-Meta-Quota-Bytes` and `X-Container-Meta-Quota-Count`
  - Custom metadata: `X-Container-Meta-Name`

ORACLE

SECTION I: Setting Container Metadata: ACLs

- Governed by Access Control Lists (ACLs)
  - Ability to read and write objects in assigned container
  - User must have the Service Administrator role
- A container has two ACLs:
  - `X-Container-Read`
  - `X-Container-Write`
- Roles can be built-in roles or custom roles
- Referrer designation indicates the host (or hosts)
  - Read access to the container should be allowed or denied.
  - The syntax of the referrer designation is `.r:value`.

ORACLE

The ability to read and write objects in a container is governed by the Access Control Lists (ACLs) assigned to the container. Any user with the Service Administrator role can perform this task.

A container has two ACLs: `X-Container-Read` and `X-Container-Write`.

The `X-Container-Read` ACL consists of a comma-separated list of roles or referrer designations. The `X-Container-Write` ACL consists of a comma-separated list of roles.

- The roles can be built-in roles or custom roles. Custom roles are defined on the My Services Security page.
  - For a role that was provisioned as part of another service instance, the format is `domainName.serviceName.roleName`.
  - For a custom role, the format is `domainName.roleName`.
- A referrer designation indicates the host (or hosts) for which read access to the container should be allowed or denied. When the server receives a request for the container, it compares the referrer designations specified in the `X-Container-Read` ACL with the value of the `Referrer` header in the request, and determines whether access should be allowed or denied. The syntax of the referrer designation is `.r:value`.

SECTION I: Setting Container Metadata: ACLs

- cURL command syntax: Grant access `X-Container-Write`

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Read: item[,item...]" \
     -H "X-Container-Write: item[,item...]" accountURL/containerName
```

- cURL command example: Grant access `X-Container-Write`

```
curl -v -X POST \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        -H "X-Container-Write:
myDomain.Storage.Storage_ReadWriteGroup,myDomain.myCustomRole" \
        https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

ORACLE

Provide write access for any user with the predefined role, `Storage_ReadWriteGroup` or the custom role, `myCustomRole`.

HTTP Response Codes:

- `Success: 204 No Content`

cURL Command Example:

```
curl -v -X POST \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        -H "X-Container-Write:
myDomain.Storage.Storage_ReadWriteGroup,myDomain.myCustomRole" \
        https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `X-Container-Write` is `myDomain.Storage.Storage_ReadWriteGroup,myDomain.myCustomRole`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Write:Storage-myIdentity3.Storage.Storage_ReadWriteGroup,Storage-
myIdentity3.myCustomRole
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:19:21 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbf2c736d57494bf88e76a-0054f98d39
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Setting Container Metadata: ACLs

- cURL Command Syntax: Grant access `X-Container-Read`

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Read: item[,item...]" \
     -H "X-Container-Write: item[,item...]" accountURL/containerName
```

- cURL Command Example: Grant access `X-Container-Read`

```
curl -v -X POST \
       -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
       -H "X-Container-Read: .r:*,.rlistings" \
  https://foo.storage.oraclecloud.com/v1/Storage-myDomain/FirstContainer
```

ORACLE

- A referrer designation indicates the host (or hosts) for which read access to the container should be allowed or denied. When the server receives a request for the container, it compares the referrer designations specified in the `X-Container-Read` ACL with the value of the `Referrer` header in the request, and determines whether access should be allowed or denied. The syntax of the referrer designation is: `.r:value`
  - value indicates the host for which access to the container should be allowed. It can be a specific host name (example: `.r:www.example.com`), a domain (example: `.r:.example.com`), or an asterisk (`.r:*`) to indicate all hosts. Note that if `.r:*` is specified, objects in the container will be publicly readable without authentication.
  - A minus sign (–) before value (example: `.r:-temp.example.com`) indicates that the host specified in the value field must be denied access to the container.
  - By default, read access to a container does not include permission to list the objects in the container. To allow listing of objects as well, include the `.rlistings` directive in the ACL (example: `.r:*,.rlistings`).
- Parameter not yet introduced:
  - item can be either a role or a referrer designation.

cURL Command Example:

```
curl -v -X POST \
        -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
        -H "X-Container-Read: .r:*,.rlistings" \
        https://foo.storage.oraclecloud.com/v1/Storage-myDomain/FirstContainer
```

- Where:
  - `token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `X-Container-Read` **is** `.r:*,.rlistings`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myDomain`
  - `containerName` **is** `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Read: .r:*,.rlistings
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:23:16 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx9127a70f18144c17afce5-0054f98e24
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Setting Container Quotas

- Setting quotas to each container:
  - Maximum number of bytes
    - `X-Container-Meta-Quota-Bytes`
  - Maximum number of objects
    - `X-Container-Meta-Quota-Count`
  - User must have the Service Administrator role.
- Any user with the Service Administrator role can perform this task.

ORACLE

For each container, you can set quotas for the maximum number of bytes the container can contain (`X-Container-Meta-Quota-Bytes`) and the maximum number of objects the container can contain (`X-Container-Meta-Quota-Count`).

SECTION I: Setting Container Quotas

- cURL command syntax:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Quota-Bytes: maxBytes" \
     -H "X-Container-Meta-Quota-Count: maxObjects" accountURL/containerName
```

- cURL command example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Quota-Bytes: 10737418240" \
     -H "X-Container-Meta-Quota-Count: 100" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

ORACLE

- Parameters not yet introduced:
  - `maxBytes` is the maximum number of bytes of data that can be stored in the container.
  - `maxObjects` is the maximum number of objects that can be created in the container.

HTTP Response Codes

- `Success: 204 No Content`

cURL Command Example:

```
curl -v -X POST \
    -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
    -H "X-Container-Meta-Quota-Bytes: 10737418240" \
    -H "X-Container-Meta-Quota-Count: 100" \
    https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

- Where:
  - `token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `maxBytes` **is** `10737418240`
  - `maxObjects` **is** `100`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/`
  - `containerName` **is** `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Quota-Bytes: 10737418240
> X-Container-Meta-Quota-Count: 100
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:32:19 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txe8869b3edea348e5b49eb-0054f99043
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Setting Custom Metadata

- cURL command syntax:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Container-Meta-Name: value" \
     accountURL/containerName
```

- cURL command example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Container-Meta-Category: Books" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

ORACLE

- Custom metadata are arbitrary key-value pairs associated with a container. You may create any custom or arbitrary metadata you need.
- Any user with the Service Administrator role can perform this task.
- Parameters not yet introduced:
  - `Name` and `value` are the metadata key and value to be created.

HTTP Response Codes

- `Success: 204 No Content`

cURL Command Example:

```
curl -v -X POST \
    -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
    -H "X-Container-Meta-Category: Books" \
    https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `X-Container-Meta-Category` is `Books`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Container-Meta-Category: Books
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 11:35:35 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx3e77b77de39f4097a5a49-0054f99107
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION I: Deleting Containers

- To delete a container, all of its objects must be deleted first.
- Any user with the Service Administrator role can perform this task.

- **All** objects within a container **must** first be deleted before the container can be deleted. To find out whether a container contains any objects, send a HEAD request to the container URL.
- Any user with the Service Administrator role can perform this task.

HTTP Response Codes

- `Success: 204 Content`

SECTION I: Deleting Containers

- cURL command syntax:

```
curl -v -X DELETE \
     -H "X-Auth-Token: token" \
     accountURL/containerName
```

- cURL command example:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

ORACLE

cURL Command Example:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3/FirstContainer
```

- Where:
  - `token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` **is** `FirstContainer`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> DELETE /v1/Storage-myIdentity3/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Fri, 06 Mar 2015 10:43:38 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txc100a7408d564f82916fb-0054f984da
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

Objects

SECTION II: Listing Objects in a Container

- All objects in a container can be listed.
- Who can perform this task?
  - Any user with the Service Administrator role
  - Any user with a role specified in the `X-Container-Read` ACL of the container
- Objects are sorted by their names lexicographically, using `memcmp()`.
- Available parameters:
  - `limit`
  - `marker`
  - `end_marker`
  - `format`
  - `prefix`
  - `Delimiter`

ORACLE

Objects are sorted by their names lexicographically, using `memcmp()`.

All objects, up to 10000 by default, will be returned in the list, unless you filter the list by using any of the following parameters:

- **`limit:`** Limits the number of objects listed to the specified value. The default and maximum value is 10000.
- **`marker:`** Returns objects with names greater than the specified string
- **`end_marker:`** Returns objects with names less than the specified string
- format: Returns extended information about each returned object in either `xml` or `json` format (REST API only)
- **`prefix:`** Returns objects with names that start with the specified string
- **`delimiter:`** Returns objects with names that include the specified character. Only the substring of object names before the specified character are returned; only unique substrings are returned.
  - If the `prefix` parameter is also used, any matches of the specified delimiter character are ignored.
  - The prefix parameter is used to emulate directory structures in a container (that is, with a forward slash (/) as the delimiter)

SECTION II: Listing Objects in a Container

- cURL command syntax:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     accountURL/containerName[?query_parameter=value]
```

- cURL command example:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentityDomainID/myContainer?limit=15
```


- Parameters not yet introduced:
  - `containerName` is the name of the container for which objects should be listed.
  - `query_parameter=value` is the optional filtering parameter.

HTTP Response Codes

- `Success: 200 OK`
- *If there are no objects, the HTTP response code would be* `204 No Content.`

cURL Command Example:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentityDomainID/myContainer?limit=15
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentityDomainID`
  - `containerName` is `myContainer`
  - `limit is 15`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> GET /v1/Storage-myIdentityDomainID/FirstContainer HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tk4af5123f84d2e3ffb9e77ba657ac8edf
>
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:15:50 GMT
< Content-Length: 63
< X-Container-Object-Count: 4
< X-Container-Write: myIdentityDomainID.Storage.Storage_ReadWriteGroup
< Accept-Ranges: bytes
< X-Timestamp: 1425033529.95392
< X-Container-Read:
myIdentityDomainID.Storage.Storage_ReadOnlyGroup,myIdentityDomainID.Storage.Stora
ge_ReadWriteGroup
< X-Container-Bytes-Used: 92095
< Content-Type: text/plain; charset=utf-8
< X-Trans-Id: tx23ba568df8864b45bc443-0054fd80e6
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
Backup-2-0_24680
Backup-3-0_32872
MetadataLog-0_32872
test.key
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

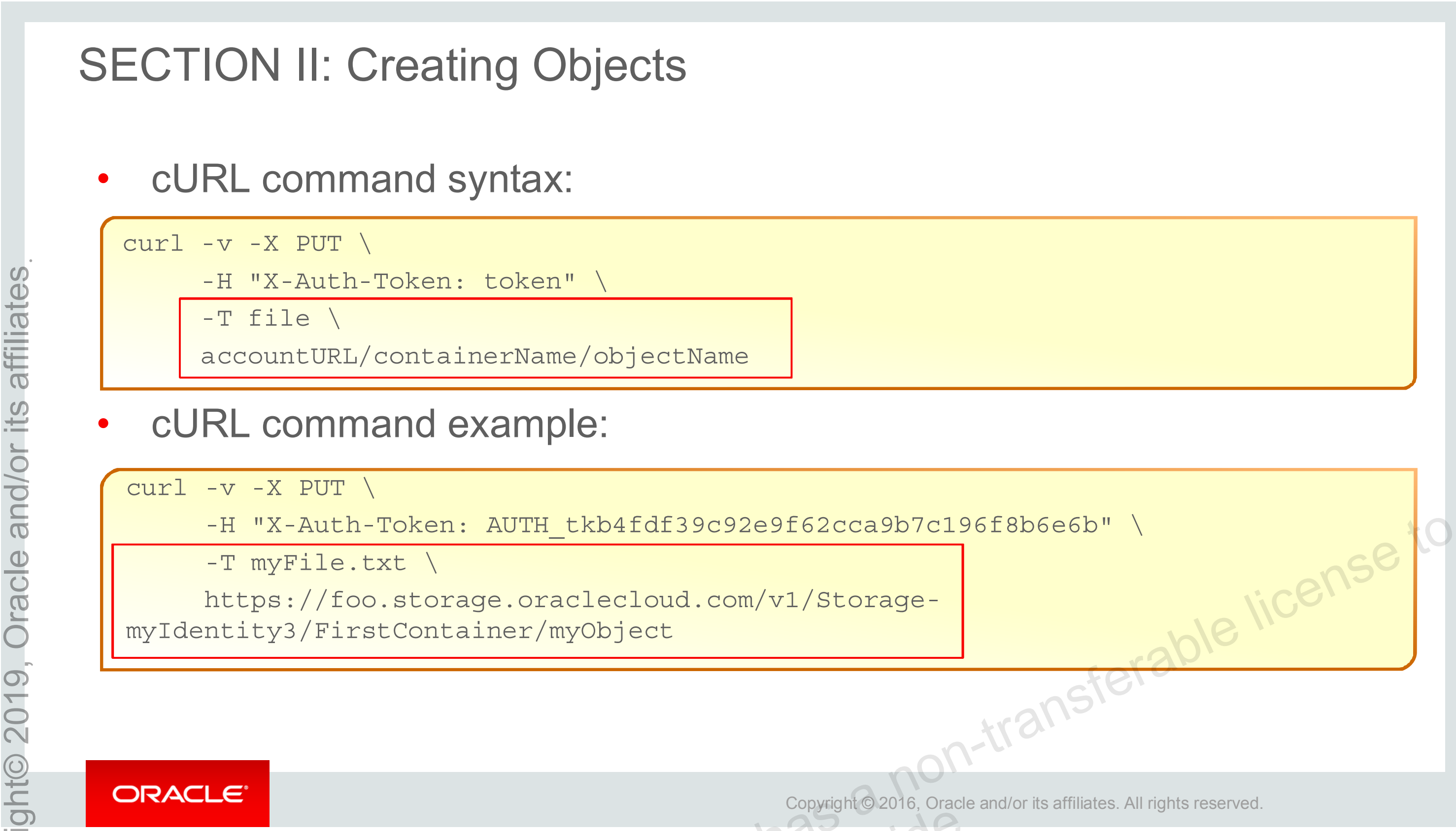
SECTION II: Creating Objects

- Objects must be created in a container.
- There are two way to create objects:
  - Uploading files
  - Specifying metadata
- Who can perform this task?
  - Any user with the Service Administrator role
  - Any user with a role that is specified in the `X-Container-Write` ACL of the container
- Objects can be created in a `standard` or an `archive` container
- The following are the object parameters:
  - `containerName, objectName, file`

Objects are sorted by their names lexicographically using `memcmp()`. All objects, up to 10000 by default, will be returned in the list, unless you filter the list by using any of the following parameters:

- `containerName` is the name of the container in which the object should be created.
- `objectName` is the name of the object to be created.
- `file` is the full path and name of the file to be uploaded.

Note: If you have a large file, greater than 5GB in size, split it into segments using the `split` command on Linux or a utility such as WinZip on a Windows computer.



- Parameters not yet introduced:
  - `containerName` is the name of the container in which the object should be created.

HTTP Response Codes

- `Success: 201 Created`

cURL Command Example:

```
curl -v -X PUT \
    -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
    -T myFile.txt \
    https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer/myObject
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`
  - `objectName` is `myObject`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust Network,O=Symantec Corporation,C=US
> PUT /v1/Storage-myIdentity3/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> Content-Length: 23
> Expect: 100-continue
>
* Done waiting for 100-continue
< HTTP/1.1 201 Created
< Date: Mon, 09 Mar 2015 11:26:57 GMT
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Content-Length: 0
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx2a97f34acb7048679ae3b-0054fd8381
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION II: Downloading Objects

- cURL command syntax:

```
curl -v -X GET \
     -H "X-Auth-Token: token" \
     -o file \
     accountURL/containerName/objectName
```

- cURL command example:

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -o myFile.txt \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```


When you download an object, the object's metadata and data are downloaded.

Any user with the Service Administrator role or a role that is specified in the X-Container-Read ACL of the container can perform this task.

Parameters:

- `file` is the full path and name of the file to which the object should be downloaded.
- `containerName` is the name of the container that contains the object to be downloaded.
- `objectName` is the name of the object to be downloaded.

HTTP Response Codes

- `Success: 200 OK`

cURL Command Example

```
curl -v -X GET \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -o myFile.txt \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `myContainer`
  - `objectName` is `myObject`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> GET /v1/Storage-myIdentity3/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
<
< HTTP/1.1 200 OK
< Date: Mon, 09 Mar 2015 11:34:33 GMT
< Content-Length: 23
< Accept-Ranges: bytes
< Last-Modified: Mon, 09 Mar 2015 11:26:58 GMT
< Etag: 846fa9d298be05e5f598703f0c3d6f51
< X-Timestamp: 1425900417.95553
< Content-Type: application/octet-stream
< X-Trans-Id: txf0b592c7e49b4475944f8-0054fd8549
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
{ [data not shown]
  0    23    0    23    0     0     53      0 --:--:-- --:--:-- --:--:--   234*
Connection #0 to host foo.storage.oraclecloud.com left intact

* Closing connection #0
```

SECTION II: Deleting Objects

- cURL command syntax:

```
curl -v -X DELETE \
     -H "X-Auth-Token: token" \
     accountURL/containerName/objectName
```

- cURL command example:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer/myObject2
```

ORACLE

Any user with the Service Administrator role or a role that is specified in the `X-Container-Read` ACL of the container can perform this task.

Parameters:

- `containerName` is the name of the container that contains the object to be deleted.
- `objectName` is the name of the object to be deleted.

HTTP Response Codes

- `Success: 204 Content`

cURL Command Example:

```
curl -v -X DELETE \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer/myObject2
```

- Where:
  - `token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`
  - `objectName` is `myObject2`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust Network,O=Symantec Corporation,C=US
> DELETE /v1/Storage-myIdentity3/FirstContainer/myObject2 HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
>
< HTTP/1.1 204 No Content
< Date: Mon, 09 Mar 2015 11:40:23 GMT
< Content-Length: 0
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: tx47aef42f16c44bd9a72cb-0054fd86a7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
```

SECTION II: Updating Object Metadata

- Updating Custom Metadata for Objects
  - Arbitrary key-value pairs
  - Metadata keys:
    - Underscore
    - First letter capitalized, the rest lowercase
- Schedule deletion
  - At a specified time in the future
  - After a specified period of time has elapsed
- Any user with the Service Administrator role can perform this task:
  - `X-Container-Write`
- Scheduling Automatic Deletion of Objects

ORACLE

- Custom metadata are arbitrary key-value pairs. You may define and update any custom or arbitrary metadata that you need.
- Any user with the Service Administrator role or a role that is specified in the `X-Container-Write` ACL of the container can perform this task.
- The service transforms custom metadata keys as follows:
  - Underscores are converted to hyphens.
  - The first character after a hyphen is capitalized. All other letters are converted to lowercase.
- You can schedule deletion of objects at a specified time in the future or after a specified period of time has elapsed, by using the `X-Delete-After` or `X-Delete-At header`, respectively.

SECTION II: Updating Object Metadata

- cURL command syntax:

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Object-Meta-Name: value" \
     accountURL/containerName/objectName
```

- cURL command example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Object-Meta-Language: english" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer/myObject
```

ORACLE

Remember: The service transforms custom metadata keys as follows:

- Underscores are converted to hyphens.
- The first character after a hyphen is capitalized. All other letters are converted to lowercase.

Parameters not yet introduced:

- `name` and `value` are the metadata key and value to be created (using `X-Object-Meta-Name`).
- `containerName` is the name of the container that contains the object for which custom metadata should be created
- `objectName` is the name of the object for which custom metadata should be created.

HTTP Response Codes

- `Success: 202 Accepted`

cURL Command Example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Object-Meta-Language: english" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/FirstContainer/myObject
```

- Where:
  - `Token` is `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` is `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` is `FirstContainer`
  - `objectName` is `myObject`
  - `X-Object-Meta-Name` is `X-Object-Meta-Language`
  - `value` is `english`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/FirstContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Object-Meta-Language: english
>
< HTTP/1.1 202 Accepted
< Date: Mon, 09 Mar 2015 11:46:34 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txd54813b92dcd46849b009-0054fd8819
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
Accepted. The request is accepted for processing.
```

SECTION II: Scheduling Automatic Deletion of Objects

- cURL command syntax: After elapsed time

```
curl -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Delete-After: period" \
     accountURL/containerName/objectName
```

- cURL command example: After elapsed time

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Delete-After: 86400" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```


You can schedule deletion of objects after a specified period of time has elapsed, by using the `X-Delete-After` or `X-Delete-At` header, respectively.

Note: You cannot schedule automatic deletion of objects for an `Archive` container by using the `X-Delete-After` and `X-Delete-At` headers.

The following command sets the object named `myObject` to be deleted automatically after <u>86400</u> seconds:

Parameters not yet introduced:

- `period` is the duration, in seconds, after which the object should be deleted (`X-Delete-After`).

HTTP Response Codes

- `Success: 202 Accepted`

cURL Command Example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Delete-After: 86400" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```

- Where:
  - `token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` **is** `myContainer`
  - `objectName` **is** `myObject`
  - `period` **is** `86400`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/myContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Delete-After: 86400
>
< HTTP/1.1 202 Accepted
< Date: Mon, 23 Mar 2015 12:32:39 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbb5a2f22164e47aa8116f-00551007e7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
The request is accepted for processing.
```

SECTION II: Scheduling Automatic Deletion of Objects

- cURL command syntax: Specified Time

```
url -v -X POST \
     -H "X-Auth-Token: token" \
     -H "X-Delete-At: time" \
     accountURL/containerName/objectName
```

- cURL command example: Specified Time

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Delete-At: 1417341600" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```


You can schedule deletion of objects at a specified time in the future, by using the `X-Delete-After` or `X-Delete-At` header, respectively.

Note: You cannot schedule automatic deletion of objects for an `Archive` container by using the `X-Delete-After` and `X-Delete-At` headers.

The following command sets the object named myObject to be deleted automatically on November 30, 2014 at 10:00:00 GMT, represented by the UNIX Epoch timestamp, 1417341600:

- Parameters not yet introduced:
  - `time` is the UNIX Epoch timestamp representing the date and time at which the object should be deleted. For example, 1416218400 represents November 17, 2014 10:00:00 GMT (`X-Delete-At`).

HTTP Response Codes

- `Success: 202 Accepted`

cURL Command Example:

```
curl -v -X POST \
     -H "X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b" \
     -H "X-Delete-At: 1417341600" \
     https://foo.storage.oraclecloud.com/v1/Storage-
myIdentity3/myContainer/myObject
```

- Where:
  - `Token` **is** `AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b`
  - `accountURL` **is** `https://foo.storage.oraclecloud.com/v1/Storage-myIdentity3`
  - `containerName` **is** `myContainer`
  - `objectName` **is** `myObject`
  - `time` **is** `1417341600`

The following is an example of the output of this command:

```
* About to connect() to foo.storage.oraclecloud.com port 443 (#0)
*   Trying 160.34.0.51... connected
* Connected to foo.storage.oraclecloud.com (160.34.0.51) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSL connection using TLS_RSA_WITH_AES_128_CBC_SHA
* Server certificate:
*       subject: CN=*.us2.oraclecloud.com,O=Oracle Corporation,L=Redwood
Shores,ST=California,C=US
*       start date: Oct 22 00:00:00 2014 GMT
*       expire date: Dec 21 23:59:59 2015 GMT
*       common name: *.us2.oraclecloud.com
*       issuer: CN=Symantec Class 3 Secure Server CA - G4,OU=Symantec Trust
Network,O=Symantec Corporation,C=US
> POST /v1/Storage-myIdentity3/myContainer/myObject HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.14.0.0
zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: foo.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkb4fdf39c92e9f62cca9b7c196f8b6e6b
> X-Delete-At: 1417341600
>
< HTTP/1.1 202 Accepted
< Date: Mon, 23 Mar 2015 12:32:39 GMT
< Content-Length: 76
< Content-Type: text/html; charset=UTF-8
< X-Trans-Id: txbb5a2f22164e47aa8116f-00551007e7
< Cache-Control: no-cache
< Pragma: no-cache
< Content-Language: en
<
* Connection #0 to host foo.storage.oraclecloud.com left intact
* Closing connection #0
The request is accepted for processing.
```

Quiz

Which of the following was NOT covered in this lesson?

a. Creating containers
b. Downloading containers
c. Updating object metadata
d. Listing objects in a container
e. Deleting containers

Quiz

What are the two ACLs for containers?

a. `X-Container-Read` and `X-Container-Write`
b. `X-Container-Read` and `Y-Container-Write`
c. `Y-Container-Read` and `Y-Container-Write`
d. `Y-Container-Read` and `X-Container-Write`

Quiz

When you successfully retrieve a list of objects in a container, what is the HTTP Response Code?

a. 200 OK

b. 201 Created

c. 202 Accepted

d. 204 No Content

Quiz

Which of the following is NOT a parameter when downloading an object?

a. `file`
b. `containerName`
c. `objectName`
d. `format`

Quiz

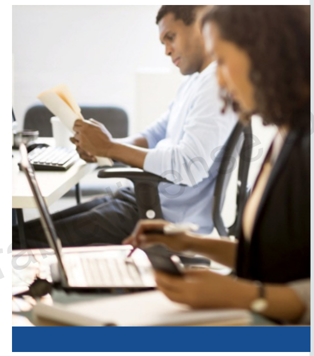
You can schedule automatic deletion of objects:

a. By specifying a time
b. After a specified period of time has elapsed
c. On a recurring basis
d. A and B
e. A and C

Summary

In this lesson, you should have learned how to:

- Create containers
- List containers
- Set container metadata
- Delete containers
- List objects in a container
- Create objects
- Download objects
- Delete objects
- Update object metadata



ORACLE