OCI Object Storage
Best Practices Guide

Best Practices for Oracle Cloud Infrastructure
Object Storage

April 2023, version 1.0
Copyright © 2023, Oracle and/or its affiliates
Public
Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document.

<table>
<thead>
<tr>
<th>DATE</th>
<th>REVISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2023</td>
<td>Revision and copyedit</td>
</tr>
<tr>
<td>April 2023</td>
<td>Initial publication</td>
</tr>
</tbody>
</table>
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>4</td>
</tr>
<tr>
<td><strong>Object Storage Limits</strong></td>
<td>4</td>
</tr>
<tr>
<td>Soft Limits</td>
<td>4</td>
</tr>
<tr>
<td>Hard Limits</td>
<td>4</td>
</tr>
<tr>
<td>Service Level Agreement</td>
<td>5</td>
</tr>
<tr>
<td><strong>Object Storage Best Practices</strong></td>
<td>5</td>
</tr>
<tr>
<td>Bucket Performance</td>
<td>5</td>
</tr>
<tr>
<td><strong>CPU Considerations</strong></td>
<td>6</td>
</tr>
<tr>
<td>Throughput</td>
<td>6</td>
</tr>
<tr>
<td>Latency</td>
<td>6</td>
</tr>
<tr>
<td>DNS Load Balancing</td>
<td>6</td>
</tr>
<tr>
<td><strong>Best Practices for Object Storage Use Cases</strong></td>
<td>6</td>
</tr>
<tr>
<td>Backup and Archive</td>
<td>6</td>
</tr>
<tr>
<td>Big Data, Data Lakes, and Hadoop</td>
<td>7</td>
</tr>
<tr>
<td>Media and Content</td>
<td>7</td>
</tr>
<tr>
<td>IoT and Security Cameras</td>
<td>7</td>
</tr>
<tr>
<td>Enterprise Workloads</td>
<td>8</td>
</tr>
<tr>
<td><strong>Conclusion</strong></td>
<td>8</td>
</tr>
</tbody>
</table>
Purpose
This document provides best practices for using the Oracle Cloud Infrastructure (OCI) Object Storage service. It explores service limits, design patterns for performance, and use cases. The performance targets mentioned in the document are true at the time of publication and constantly improving.

Object Storage Limits
OCI Object Storage has some soft and hard limits on scale and performance.

Soft Limits
OCI soft limits are set to protect against unintentional resource spikes, but they are configurable upon request based on available capacity within a region. Object Storage has the following soft limits:

- 10,000 buckets per tenancy
- 4 billion objects per bucket (higher when bucket is multisharded)
- 3,000 writes/second per bucket
- 10,000 reads/second per bucket
- 2,000 lists/second per bucket
- Bandwidth up to 1 Tbps
- Default throttling rate limits: 50,000 writes/second, 500,000 reads/second, and bandwidth up to 3 Tbps per tenancy

Hard Limits
Object Storage has the following hard limits:

- Number of Object Storage namespaces per root compartment: 1
- Number of compartments per tenancy: 1,000
- Unlimited number of buckets for a customer
- Maximum object size: 10 TiB
- Maximum number of parts in multipart upload: 10,000
- Maximum object part size in multipart upload: 50 GiB
Service Level Agreement

OCI Object Storage’s service level agreement (SLA) has 99.9% availability, but the service has no SLAs for manageability or performance.

Object Storage Best Practices

Use the following design patterns to get the best performance with OCI Object Storage storing and retrieving your data.

- **Content Delivery Network (CDN) service**: Customers can use a CDN service for caching frequently accessed objects that allows delivering of the content to end users from a nearby location in a geographically distributed network. Customers benefit from integrated APIs, the Oracle Cloud Console, Universal Credits billing, and stronger integrations between OCI Object Storage, Compute, and a CDN to reduce origin-server-to-CDN egress fees. For example, the OCI Media Streams service can create distribution channel for video content stored in Object Storage through OCI Edge or Akamai CDNs. You can also use Cloudflare for free egress through the Bandwidth Alliance.

- **Retries for timeouts**: For latency-sensitive applications, we recommend tracking the responses from Object Storage, such as HTTP 429 throttle 503 slowdown responses, and using exponential backoff and automatic retry logic. You can use OCI software developer kits (SDKs) that are configured by default to retry certain operations that fail.

- **Parallel requests for high throughput**: When possible, we recommend scaling horizontally and using requests in parallel for accessing objects. This approach distributes the network load across multiple paths in Object Storage distributed architecture. Using a HTTP connection pool and reusing the same connection for multiple requests can help achieve higher overall throughput.

- **Use in-region OCI Compute instances**: Using Compute instances in the same region as the Object Storage endpoints help with the performance by keeping the latency between the client and server at the minimum.

- **Use byte range operations**: For large objects, especially ones uploaded as multipart objects, we recommend using byte-range GET requests. You can use multiple concurrent connections to fetch different byte ranges of the same object in parallel.

- **Use storage tiers**: OCI Object Storage supports standard, infrequent access, and archive tiers for hot, warm, and cold data. We recommend understanding and using the storage tiers for storing data long-term in a cost-effective manner. You can also use automatic tiering for moving objects between the standard and infrequent access tiers.

Bucket Performance

OCI Object Storage scales performance for metadata operations sharded by bucket. We can support approximately 10,000 metadata requests per second in a bucket assigned to a single database shard. So, supporting about 160 Gbp/s of upload throughput requires one database shard to be 100% utilized (5,000 4-MB transactions per second, with 2 TPS per upload operation). To scale beyond this limit, create more buckets or contact Object Storage using a service request to enable your tenancy for multishard bucket operations for better performance.

**Note**: You can use My Oracle Support to file a service request for increasing or decreasing your service limits. The service limits are changed only after they’re approved. Approval is not immediate and can take a few days to become effective.
CPU Considerations

Object Storage recommends starting with a single request at a time. Measure the network bandwidth being achieved and the use of other resources that your application uses in processing the data. You can then identify the bottleneck resource (the resource with the highest usage), and the number of requests that are likely to be useful. For example, if processing one request at a time leads to a CPU usage of 25 percent, it suggests that up to four concurrent requests can be accommodated. Measurement is essential, and it's worth confirming resource use as the request rate is increased.

Object Storage has seen client libraries being throttled based on concurrency limitations (Python SDK) and cryptography limitations of Java virtual machines (JVMs).

Throughput

We recommend making concurrent requests for byte ranges or whole objects at the granularity of 16–32 MB. To saturate a 10-Gb/s network interface card (NIC), for example, you might use 16–32 concurrent requests over separate connections. You can scale up the concurrent requests over more connections to saturate faster NICs, such as 25-Gb/s NICs. As a best practice, use multipart uploads and downloads for large objects for better throughput. Also, as a best practice, use OCI SDKs because we continuously optimize them for performance.

Latency

For small object operations of sizes 512 KB–1 MB, we optimize for latency instead of throughput. Ensure that your application avoids per-request connection setup to remove the need to perform TCP slow-start and secure sockets layer (SSL) handshakes on each request. These applications can achieve consistent small object latencies (and time to first byte TTFB latencies for larger objects) of roughly 100–250 milliseconds. ListObject calls can return large amounts of data and take a longer time to complete.

DNS Load Balancing

Ensure that requests are spread over a wide pool of Object Storage IP addresses. We currently support one public IP address per availability domain. For three-availability domain regions, you need to spread requests among each of these endpoints. Caching resolvers or application code that reuses a single IP address don’t benefit from address diversity and the load balancing that follows from it. For Java, Sun’s JVM caches DNS lookups forever by default. For information on how to change this behavior, see the "InetAddress Caching" section of the InetAddress documentation.

Best Practices for Object Storage Use Cases

Use the following best practices for your Object Storage use cases.

Backup and Archive

OCI Object Storage is certified with Veritas, Commvault, and Veeam for storing backups. Other backup software vendors can also store backups to OCI Object Storage using the S3-compatible API as a S3 Cloud Storage tier. The backup software can run on-premises and store backups directly to OCI Object Storage, in which case a FastConnect connection to OCI is recommended. In case of backup software running on OCI services like Compute, Oracle Cloud VMware Solution, and Oracle Container Engine for Kubernetes, we recommend connecting to buckets in the same region for faster backups and restores.
You can store backups on the standard tier initially and then move them to the infrequent access tier when they become cool. You can also use the autotiering capability to move backups older than 31 days from standard to the infrequent access tier automatically. If the backup in the infrequent access tier is accessed, it’s immediately available for reading and moves back into standard tier if it’s accessed multiple times within a span of a month. OCI Archive Storage and Object Storage archive tier have a one-hour SLA for restoration, so they’re not currently supported with backup software automated restore workflows.

**Big Data, Data Lakes, and Hadoop**

The OCI Big Data service provides a Hadoop stack that includes Ambari, Hadoop, HBase, Hive, and Spark that you can use for such workloads and can store data into OCI Object Storage using the Hadoop Distributed File System (HDFS) Connector. Streaming and batch workloads can both use the data stored in Object Storage. As a general best practice, use bigger chunk sizes (approximately 640 MB) to prevent excessive range reads resulting from smaller chunk sizes.

We also recommend using DirectFileOutputCommitter in the Spark config and a smaller cluster size to resolve HTTP 429 (too many request) errors. The OCI Data Flow serverless platform can run Spark jobs to collect and analyze data in AWS S3 or Azure blob store and write the data and results in OCI Object Storage. This [reference architecture for a cloud data lakehouse](https://www.oracle.com/technology/products/oci-data-flow/) shows how to process enterprise and streaming data for analysis and machine learning.

**Media and Content**

You can use OCI Object Storage as a content repository for media content, which a CDN can distribute by configuring media streams to act as the origin service for video distribution through CDN, like OCI Edge or Akamai. The OCI Media Services includes OCI Media Flow and OCI Media Streams. You can use the services independently or together and operate on the content stored in OCI Object Storage.

For media content to be retained for a long duration, you can use OCI Archive Storage, which offers inexpensive long-term storage. Before you can read objects stored in archive storage, they must be restored to standard storage. The restoration can take up to one hour. If the use case requires instant reads, we recommend using the infrequent access tier for this content.

For example, media assets stored in OCI Object Storage can be processed transcoded, and the result can be again stored in Object Storage. By utilizing OCI’s serverless capabilities, including OCI Functions and OCI NoSQL, we can quickly create a management system for [processing media content](https://www.oracle.com/technology/products/oci-no-sql/) using ephemeral OCI Compute workers.

**IoT and Security Cameras**

Internet of things (IoT) and security camera devices are usually PUT-heavy workloads, uploading many small objects that are very infrequently accessed. They usually upload 10–60-second long video clips as objects into buckets to store for a few days. The buckets are configured to expire the objects using object lifecycle management (OLM) policies after 3, 7, or 30 days, based on the class of service opted by the consumer. Some of the video clip objects might be kept for longer time if the owners want to hold them for legal reasons.

The best practice for these workloads is to distribute the data across multiple buckets. The customers should raise a service request for multisharding their buckets by default to support up to 4 billion objects per bucket and up to 10,000 operations per second. We recommend that the data to be stored for less than 30 days be kept on standard buckets in standard tier, while data to be stored for 30 days or more be stored in standard buckets in the infrequent access tier, which offers cheaper storage with a small fee for access. Objects needed to be held forever can be uploaded or moved into Archive storage tier.
Sometimes short-lived preauthenticated requests (PARs) are used to upload objects directly from the cameras. In these cases, we recommend using S3-compatible APIs, which support presigned URLs for such uploads, or using a small number of PARs that are stored on the IoT devices and rotated every 30 days, for example.

**Enterprise Workloads**

As a best practice, subscribe to all the OCI regions needed by your enterprise and then create compartments and add users to your tenancies. This strategy ensures minimum propagation delays for your remote employees to access OCI resources. You can have a hierarchy between your tenancies by setting up parent-child tenancy relationships for ease of organization management. Child tenancies inherit the policies from the parent and consume from the primary funded OCI cloud subscription of the parent tenancy. You can create child tenancies or invite existing tenancies to join as child tenancies in a parent tenancy.

**Conclusion**

OCI Object Storage provides highly durable, secure, and scalable retention and management for your unstructured data. Following the best practice recommendations in this guide provide the best experience for your workloads that use OCI Object Storage.