OCI Load Balancing Service (LBaaS)

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. 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 for Oracle's products remains at the sole discretion of Oracle.

Basic Concepts

A load balancer sits between the clients and the backends performs tasks such as:

- **Service Discovery**: What backends are available in the system? How should the load balancer talk to them?

- **Health Check**: What backends are currently healthy and available to accept requests?

- **Algorithm**: What algorithm should be used to balance individual requests across the healthy backends?

Load Balancer benefits

- **Fault tolerance and HA**: using health check + LB algorithms, a LB can effectively route around a bad or overloaded backend

- **Scale**: LB maximizes throughput, minimizes response time, and avoids overload of any single resource

- **Naming abstraction**: name resolution can be delegated to the LB
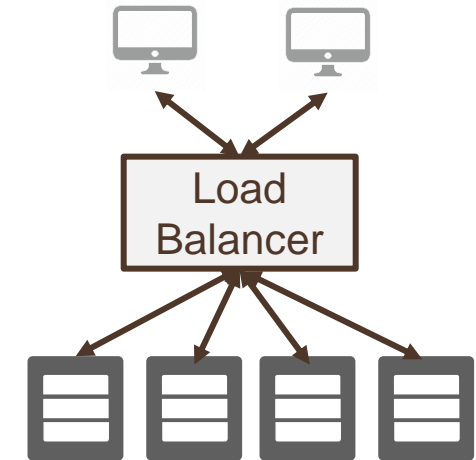
OCI Load Balancing Service

- Load Balancer as-a-service, provides scale and HA

- Public and Private Load Balancer options

- Supported Protocols – TCP, HTTP/1.0, HTTP/1.1, HTTP/2, WebSocket

- Supports *frontend SSL backend SSL point-to-point SSL*

- Supports advanced features such as session persistence and policy routing

- Flexible Load Balancing

- Flexible Network Load Balancer

- Key differentiators

  - Private or Public Load Balancer (with Public IP address)

  - Dynamic bandwidth and/or 100 Mbps, 400 Mbps, 8 Gbps size

  - Flexible shaping, OCI Load Balancer automatically adjust bandwidth

Public Load Balancer

- Accepts traffic from the internet using a public IP address that serves as the entry point for incoming traffic

- Public Load Balancer is a **regional service**

- If your region includes multiple availability domains, a public load balancer requires either a **regional subnet (recommended) or two availability domain-specific (AD-specific) subnets**, each in a separate availability domain.

- Load Balancing service creates a primary load balancer and a standby load balancer, each in a different availability domain

- Supports AD failover in the event of an AD outage in an Oracle Cloud Infrastructure multi-AD region

- Floating Public IP is attached to the primary load balancer, and in the event of an AD outage Floating Public IP is attached to the standby load balancer

- Service treats the two load balancers as equally you cannot denote one as "primary"

Public Load Balancer (Regional Subnets - recommended)
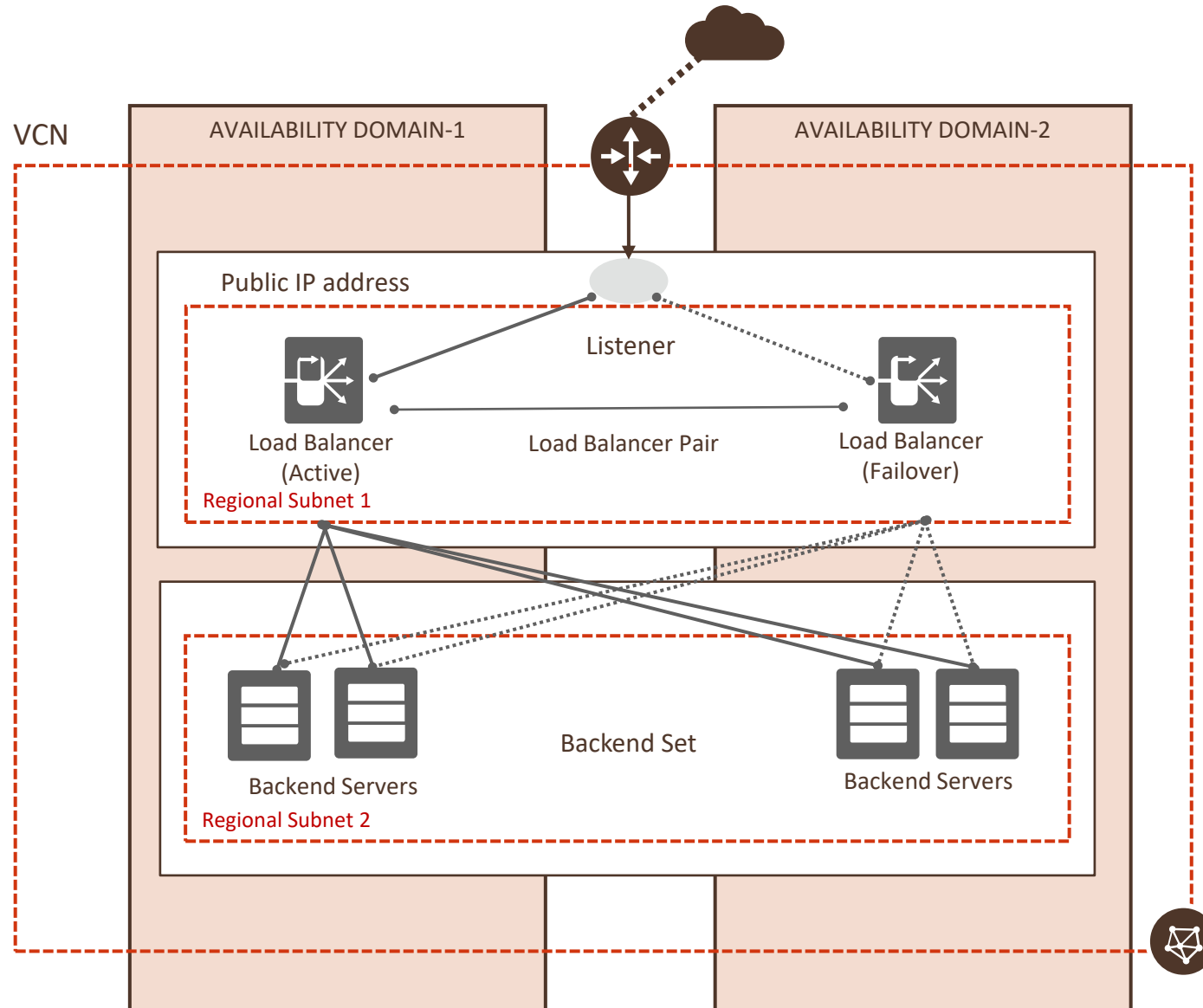
Public Load Balancer (AD Specific Subnets)
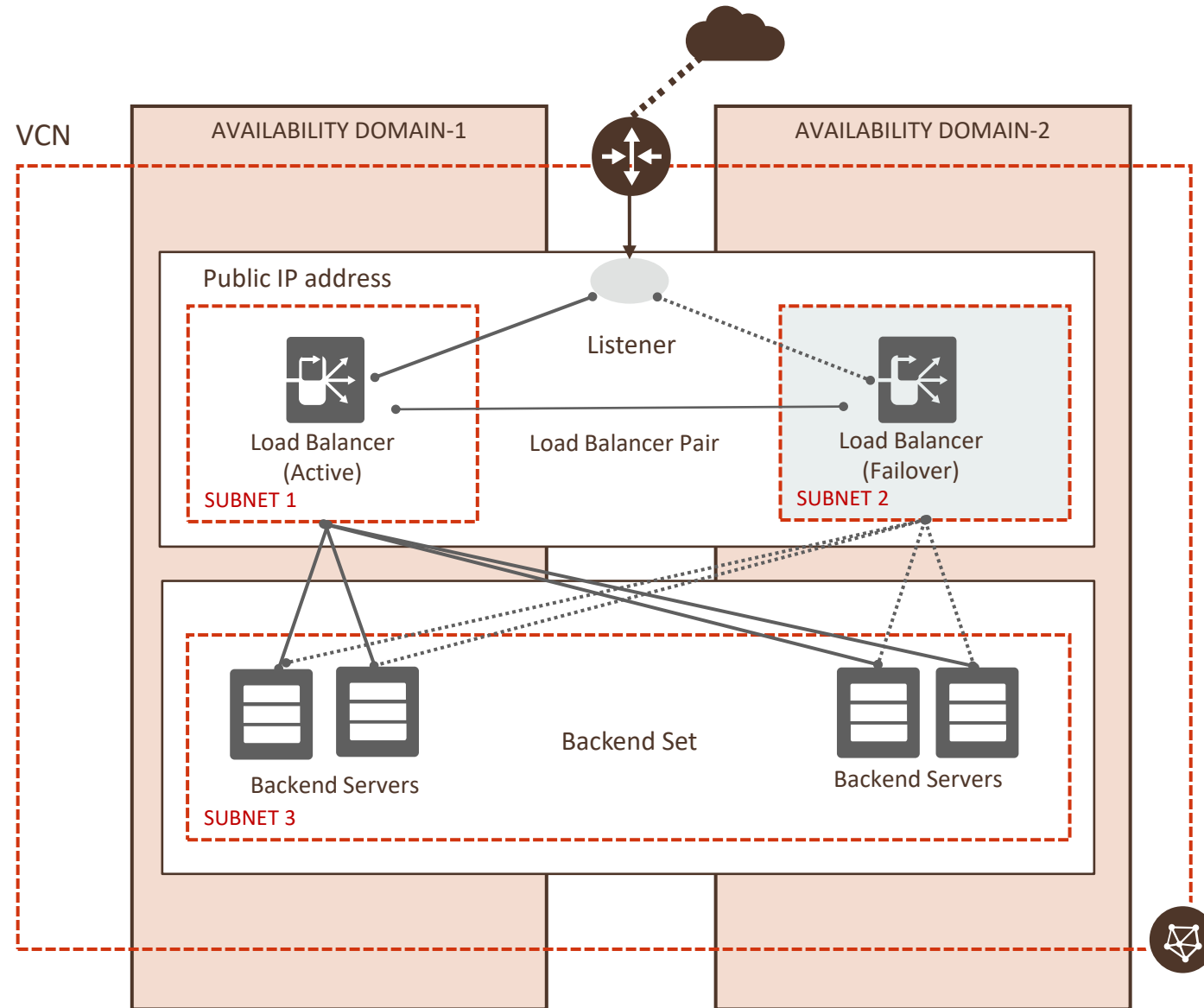
Concepts - Public Load Balancer
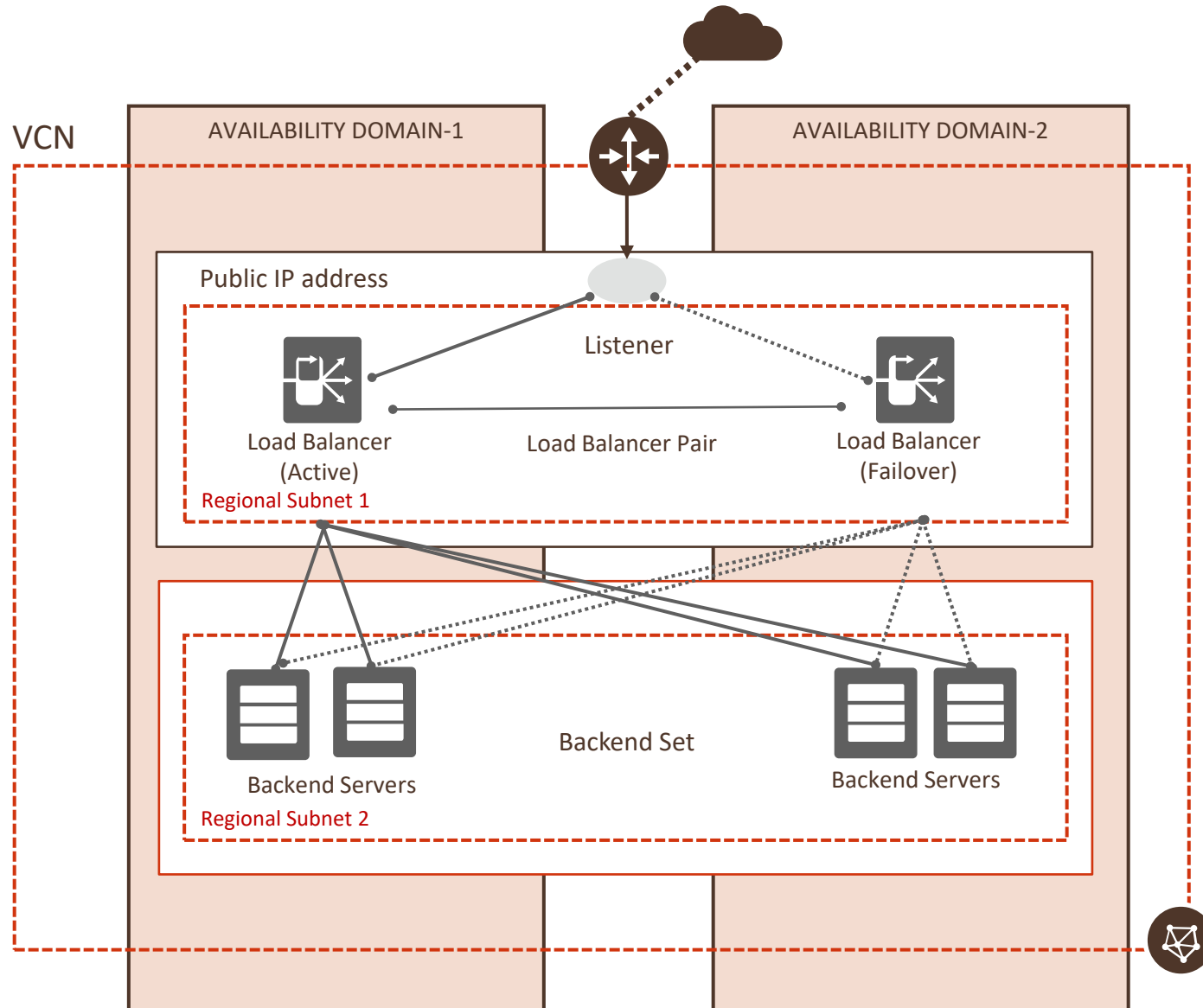
- **Load Balancing Policy** – tells the load balancer how to distribute incoming traffic to the backend servers
  - round-robin
  - IP hash
  - least connection

- **Backend Server** – application server responsible for generating content in reply to the incoming TCP or HTTP traffic

- **Health Checks** – a test to confirm the availability of backend servers; supports
  - TCP-level
  - UDP-level
  - HTTP-level health checks

- **Backend Set** – logical entity defined by a list of backend servers, a load balancing policy, and a health check policy

- **Listener** – entity that checks for incoming traffic on the load balancer's IP address

Private Load Balancer

- Assigned a private IP address from the subnet hosting the load balancer

- The load balancer can be regional or AD-specific, depending on the scope of the host subnet; highly-available within an AD with AD specific subnets or Highly available with regional subnets

- The primary and standby load balancer each require a private IP address from that subnet

- The load balancer is accessible only from within the VCN that contains the associated subnet, or as further restricted by your security list rules

Dynamic Change of LB Shapes

OCI Standard Load Balancer

OCI launched Dynamic Change of Shapes, where customers can switch between shape options on their LB instance at runtime

Customer can switch between bandwidth shapes without having to create a new LB as a workaround

10 Mbps 100 Mbps 400 Mbps 8,000 Mbps shapes

Flexible Load Balancer

Customer just defines the minimum and maximum bandwidth

• Minimum bandwidth provides instant readiness for load

• Maximum bandwidth allows control of maximum cost

• Customer pays a minimal base cost for the load balancer and then pays a simple single rate for the larger of the reserved bandwidth or the maximum bandwidth actually used each minute

Load Balancing Policies

- **Round Robin**: default policy, distributes incoming traffic sequentially to each server in a backend set. After each server has received a connection, the load balancer repeats the list in the same order.

- **IP Hash**: uses an incoming request's source IP address as a hashing key to route non-sticky traffic to the same backend server

- **Least Connection**: routes incoming non-sticky request traffic to the backend server with the fewest active connections

- Load balancer policy decisions apply differently to TCP load balancer, cookie-based session persistent HTTP requests (sticky requests), and non-sticky HTTP requests
  - A TCP load balancer considers policy and weight criteria
  - An HTTP load balancer w/ cookie-based session persistence forwards requests using cookie's session info
  - For non-sticky HTTP requests, the load balancer applies policy and weight criteria

Health Check

- Health check is a test to confirm the availability of backend servers. Health Check is activated for
    - Backends
    - Backend set
    - Overall Load Balancer

- A load balancer IP can have up to 16 listeners (port numbers). Each listener has a backend set that can have 1 to N backend servers

Load Balancer
IP

Listener

Listener

Server 1

Server 2

Server 3

Backend set
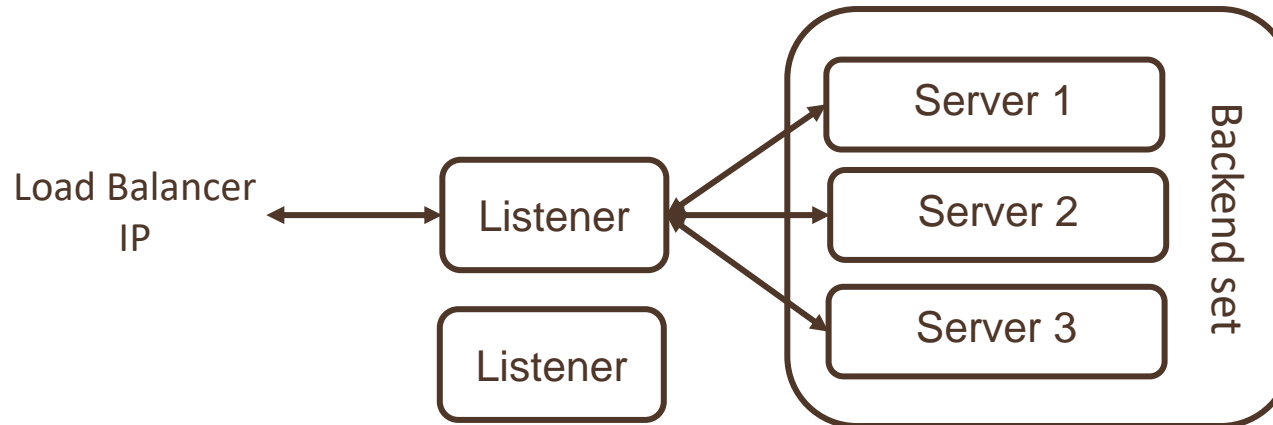
- Health API provides a 4-state health status (ok, warning, critical, unknown)

- Health status is updated every three minutes

Health Check

OCI Standard Load Balancer

Request Routing

- Each HTTP request is evaluated against the rules.
- The rules are run in the order that is defined in the policy.
- Each rule has at least one condition and a backend set.
- If the HTTP request condition matches a rule, the request is forwarded to the backend set defined for the rule. The other rules in the policy are skipped and the request is not evaluated against them

Rule Set

A rule set is a named set of rules associated with a load balancer and applied to one or more listeners on that load balancer

•Access control rules restrict access to application resources based on the source of the request.

•Access method rules specify the permitted HTTP methods.

•URL redirect rules route incoming HTTP requests to a different destination URL.

•Request and response header rules, which add, alter, or remove HTTP request or response headers.

•HTTP header rules, which specify the size of the HTTP header, enlarge header and whether period and underscore characters are permitted within the headers.

SSL Handling

OCI Standard Load Balancer

Terminate SSL at the load balancer. This configuration is *frontend SSL*. Your load balancer can accept encrypted traffic from a client. No encryption of traffic exists between the load balancer and the backend servers.

Implement SSL between the load balancer and your backend servers. This configuration is *backend SSL*. Your load balancer does not accept encrypted traffic from client servers. Traffic between the load balancer and the backend servers is encrypted.

Implement point-to-point SSL. Your load balancer can accept SSL encrypted traffic from clients and encrypts traffic to the backend servers.

To use SSL with your load balancer, you must add one or more certificate bundles to your system.

Oracle Cloud Infrastructure accepts x.509 type certificates in PEM format only.

Session Persistence

Session persistence is a feature that can be configured and enabled on the level of a backend set. For each backend set, two parameters are set to control it:

- **Cookie name**: the name of a cookie, or a match-all pattern, that will be set in the response from the backend server to request session persistence, application or lb generated

- **Fallback**: a boolean value that controls how OCI LB handles session persisted requests in case the original backing server isn't available false: the load balancer fails the request and returns an HTTP 502 code

Logging

https://docs.oracle.com/en-us/iaas/Content/Balance/Reference/troubleshooting_load_balancer.htm

General Metrics for LB

Metrics for Backend Set

OCI Standard Load Balancer
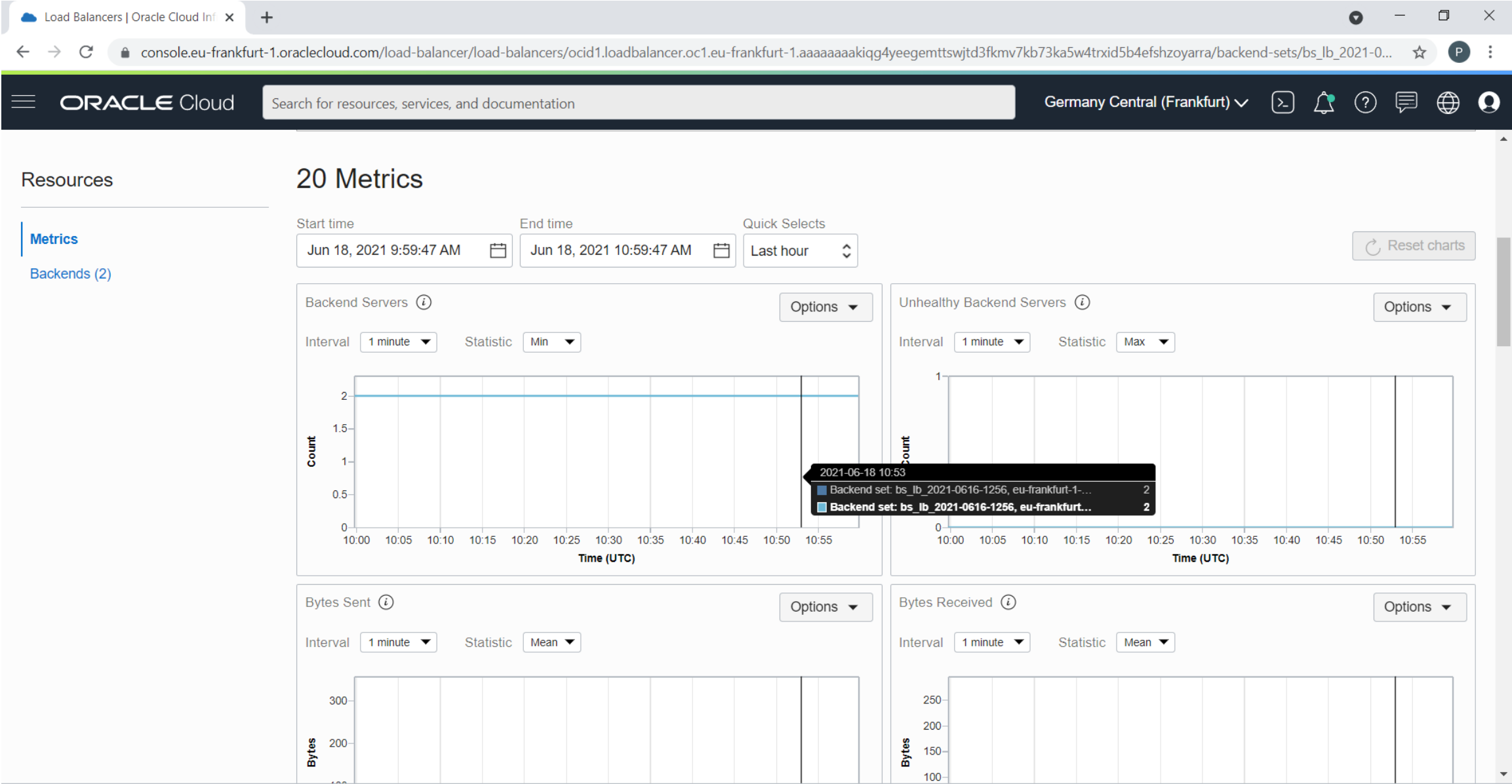
Network Load Balancer

The Network Load Balancer load balance layer 3 and layer 4 (TCP/UDP/ICMP) workloads

It's designed to handle volatile traffic patterns and millions of flows, offering high throughput while maintaining ultra-low latency.

Ideal load balancing solution for latency-sensitive workloads includes real-time streaming, VoIP, Internet of Things, and trading platforms

The Network Load Balancer is optimized for long-running connections in the order of days or months, which makes it best suited for your database or WebSocket type applications

The Network Load Balancer operates at the connection level and balances incoming client connections to healthy backend with session affinity for udp/tcp a given flow is always forwarded to the same backend for the lifetime of the connection

OCI Flexible Network Load Balancer can be public or private

Provides the benefits of flow high availability, source and destination IP addresses, and port preservation
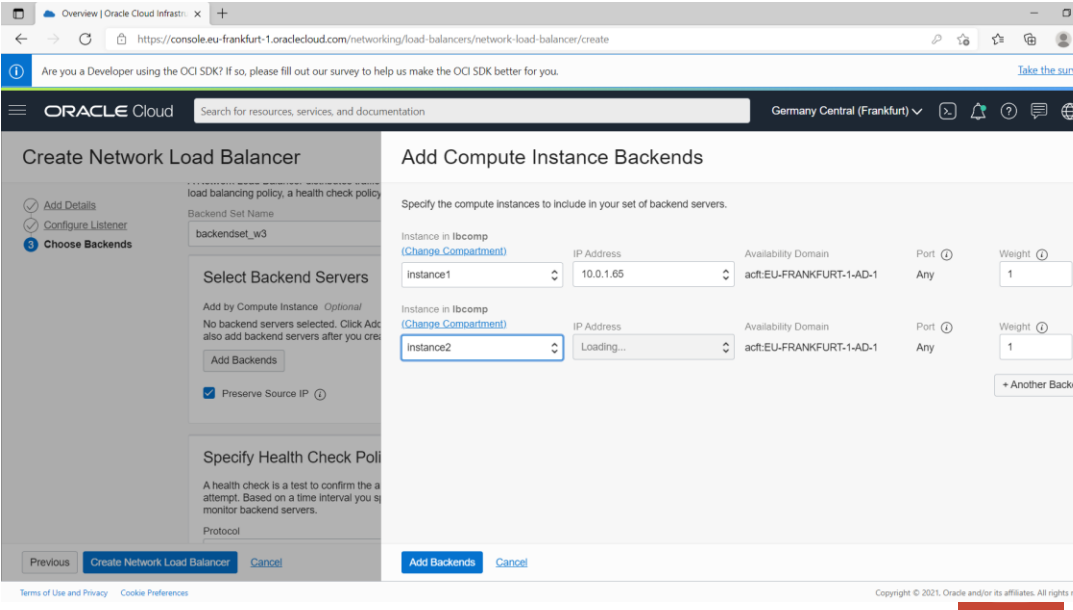
Network Load Balancer Creation Steps

nlb
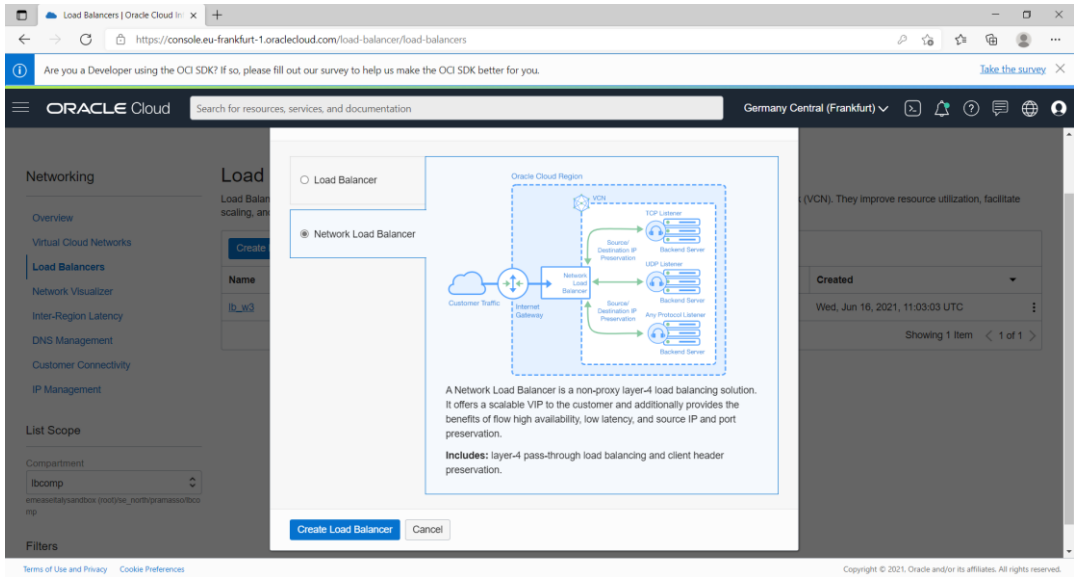
Listener
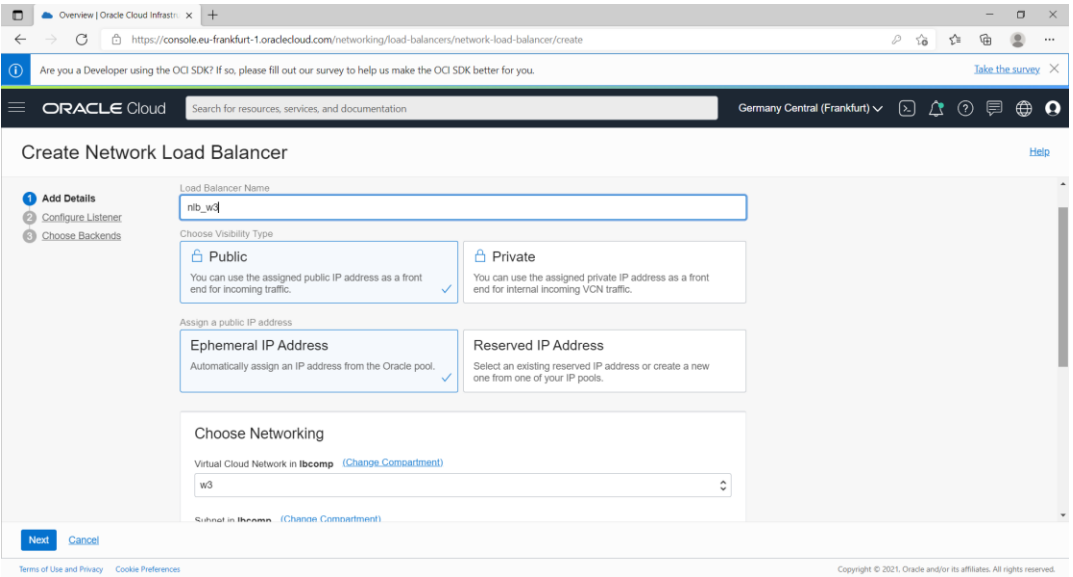
Portdef.
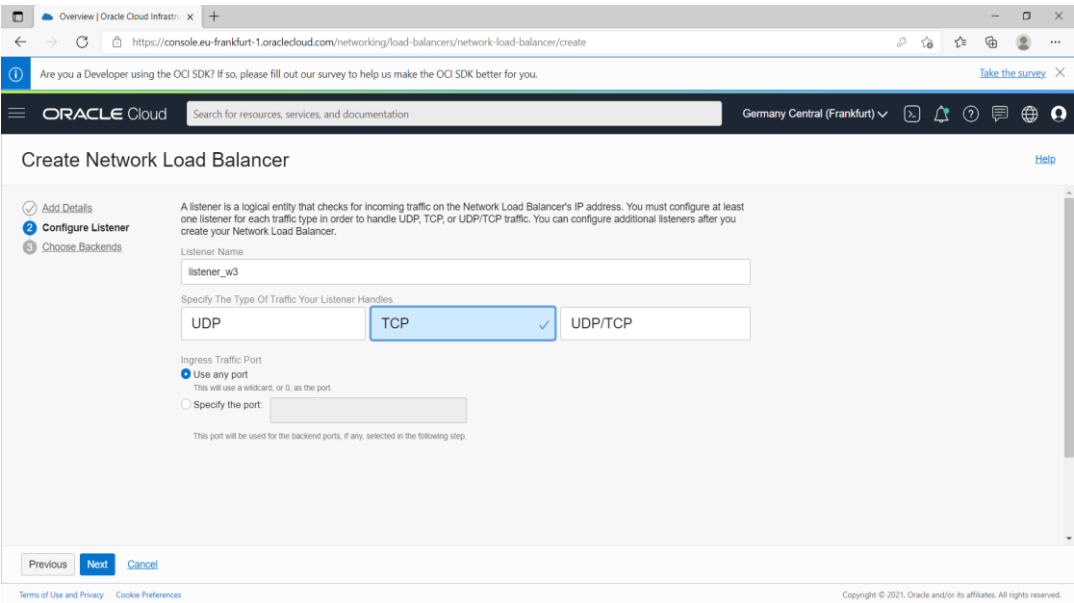
backends

Network Load Balancer (cont.)

OCI Network Load Balancer

•**Protocol:** Required.
Select one of the
following options:
- **UDP**
- **TCP**
- **UDP/TCP**

Network Load Balancer (cont.)

- **5-Tuple Hash**: Routs incoming traffic based on 5-Tuple (source IP and port, destination IP and port, protocol) Hash. This is the default network load balancer policy.
- **3-Tuple Hash**: Routs incoming traffic based on 3-Tuple (source IP, destination IP, protocol) Hash.
- **2-Tuple Hash**: Routs incoming traffic based on 2-Tuple (source IP Destination, destination IP) Hash.

The 5-Tuple Hash policy provides session affinity within a given TCP or UDP session, where packets in the same session are directed to the same backend server behind the flexible network load balancer. Use a 3-Tuple or 2-Tuple network load balancing policy to provide session affinity beyond the lifetime of a given session

Network Load Balancer (cont.)

OCI Network Load Balancer

Check **Preserve Source IP** to preserve the original source and destination header (IP addresses and ports) of each incoming packet all the way to the backend server

Network Load Balancer (cont.)

OCI Network Load Balancer

Weighting affects the proportion of requests directed to each server. For example, a server weighted as 3 receives three times the number of connections as a server weighted as 1

Network Load Balancer (cont.)

OCI Network Load Balancer

https://docs.oracle.com/en-us/iaas/Content/NetworkLoadBalancer/WorkRequests/metrics.htm

https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/vcn_flow_logs.htm#vcn_flow_logs

Both LB and NLB Summary

Oracle Cloud Infrastructure **Load Balancing service** provides automated traffic distribution from one entry point to multiple servers reachable from your virtual cloud network (VCN) to load balance transport Layer 4 and Layer 7 (TCP and HTTP) traffic

Oracle Cloud Infrastructure Flexible **Network Load Balancing service** (Network Load Balancer) provides automated traffic distribution from one entry point to multiple backend servers in your virtual cloud network (VCN). It operates at the connection level and load balances incoming client connections to healthy backend servers based on Layer 3/Layer 4 (IP protocol) data.

Q/A

Thank you