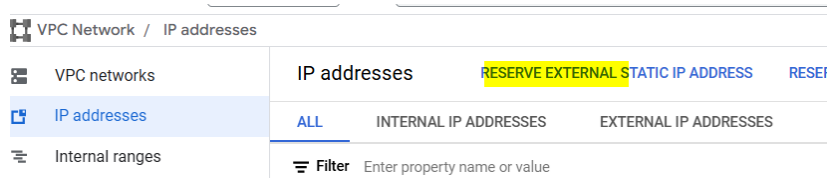


In this lab, we will create a load balancer and connect it to our inventory app

- 1) Now, before starting, we need a reserved external IP address that can be used for the external load balancer to access it from the internet
- 2) Go to VPC in GCP console, select the IP address, and then click Reserve External static IP



- 3) Give it a Name, make Premium service tier, and Global as a type

← Reserve a static address


Name *
inventory-lb-ip
Lowercase letters, numbers, hyphens allowed

Description

Network Service Tier ?
☒ Premium ?
Current project-level tier, [change](#)
☐ Standard ?

IP version
☒ IPv4
☐ IPv6

Type
☐ Regional
☒ Global
To be used with Global forwarding rules. [Learn more](#)

 Static IP addresses not attached to an instance or load balancer are billed at a higher hourly rate. [Pricing details](#)

RESERVE CANCEL

EQUIVALENT COMMAND LINE ▾

- 4) Search for load balancer and go to load balancing, then click on create load balancer

Load balancing

[+ Create load balancer](#)[Refresh](#)[Delete](#)[Learn](#)

Load balancers

Backends

Frontends

Service LB policies

Filter

Enter property name or value

?

⋮

<input type="checkbox"/>	Name	Load balancer type	Access type	Protocols	Region	Bac	Actions
--------------------------	------	--------------------	-------------	-----------	--------	-----	---------

5) Select Application Load Balancer and click Next

Internet

VPC network

Application Load Balancer

HTTP

HTTPS

Workloads

Workloads

Internet

VPC network

Network Load Balancer

TCP

UDP

TLS

Other L4 protocols

Workloads

Workloads

1

Type of load balancer

Application Load Balancer (HTTP/HTTPS)

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic.

2

Public facing or internal

Public facing (external)

Network Load Balancer (TCP/UDP/SSL)

Choose a Network Load Balancer when you need offloading at scale, support for UDP, and other protocols to your applications.

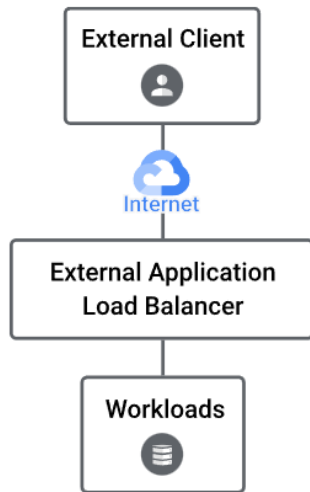
NEXT

6) Keep the Public facing option selected, click Next

2 Public facing or internal

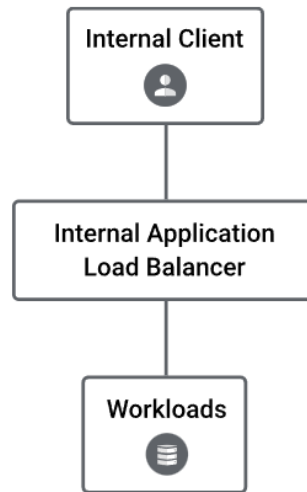
☒ Public facing (external)

An internet-facing load balancer routes requests from clients over the internet to targets.



☐ Internal

An internal load balancer routes requests from clients to backends using private IP addresses.



Next

7) Keep the global workload option selected for multiregional

Public facing (external)

3 Global or single region deployment

☒ Best for global workloads

Multiple regions. Use this for better performance if you have clients distributed globally (with a global anycast IP) or if you want to deploy backends in multiple regions.



☐ Best for regional workloads

Single region. Use this if you want traffic to be routed to a single region. For example, for workloads with compliance requirements.



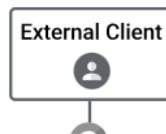
8) Keep the Global ALB option selected

Global workloads

4 Load balancer generation

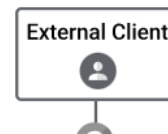
☒ Global external Application Load Balancer

Load balancer with EXTERNAL_MANAGED load balancing scheme (Recommended)



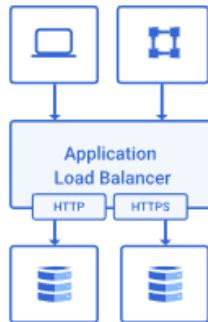
☐ Classic Application Load Balancer

Previous generation load balancer with EXTERNAL load balancing scheme



9) Click Configure

5 Create load balancer



You are about to create an Application Load Balancer

with following features:

- Public facing (external)
- Global

Configure

Cancel

10) Fill in the details for the front-end and click Done

← Create global external Application Load Balancer

Load Balancer name *
readit-inv-lb
Lowercase, no spaces.

Frontend configuration

Backend configuration

Routing rules

Review and finalize (optional)

New Frontend IP and port

Name
inventory-fe
Lowercase, no spaces.

Description

Protocol
HTTP
Select HTTPS to support clients that support HTTP/2. The load balancer automatically offers HTTP/2 as part of the TLS handshake.

Network Service Tier
Premium
Global HTTP(S) load balancing only supports the Premium Network Service Tier. [Learn more](#)

IP version
IPv4

IP address
inventory-lb-ip

Port *
80
Application load balancing supports all TCP ports. [Learn more](#)

SHOW ADVANCED FEATURES

DONE

11) Select the Backend configuration and select Create a backend service option

Load Balancer name *
readit-inv-lb
Lowercase, no spaces.

Frontend configuration

Backend configuration

Routing rules

Review and finalize (optional)

Backend configuration

Create or select a backend service for incoming traffic. You can add multiple backend servi

Only backend services created for HTTP(S) Load Balancer with Advanced Traffic Load Balancer cannot be used.
[DISMISS](#)

Backend services & backend buckets

Filter Type to filter

No matches for "

[CREATE A BACKEND SERVICE](#) [CREATE A BACKEND BUCKET](#)

12) Give it a name and select "serverless network endpoint group" on backend type. Note that this type is the same, which includes App Engine, Cloud Run, and

Functions.

Create backend service

Name *

inventory-be

Lowercase, no spaces.

Description

Backend type

Instance group

Network endpoint group

Zonal network endpoint group

GCE & GKE backends

Internet network endpoint group

External backends

Serverless network endpoint group

App Engine, Cloud Run, Cloud Functions

Private Service Connect network endpoint group

13) Select the Create serverless network endpoint group option under New backend

** *here, network endpoint group basically means a collection of IP addresses that the load balancer will call. So, our app engine or cloud run and cloud function expose IP address for communication with them and this IP address is going to be contained inside a network endpoint group. So the network endpoint group is basically some kind of mediator between the load balancer and the actual serverless resource.***

Backend type

Serverless network endpoint group

Protocol

HTTPS

Backends

New backend

Serverless network endpoint groups *

Filter | type to filter

No matches for ""

CREATE SERVERLESS NETWORK ENDPOINT GROUP

14) Fill in the details and select the inventory app engine in the service that we created previously

Create Serverless network endpoint group

Name *
inventory-neg ?
Lowercase, no spaces.

Region *
us-central1 (Iowa) ?

Serverless network endpoint group type

☐ Cloud Run ?

☐ Cloud Functions ?

☒ App Engine ?

☐ Default routing ?

☒ Select service name

Service *
inventory

☐ Use URL mask ?

CREATE CANCEL

15) Deselect Cloud CDN as it will cost more and also not needed

CDN is nothing but content delivery network which helps in providing data faster to users and we do not need that in lab

Cloud CDN ?

☐ Enable Cloud CDN

i Cloud CDN is a paid product that accelerates content delivery for web applications. [Learn more about features](#)

16) Go to advanced settings and check for the below configurations

Advanced configurations ^

Session affinity
None ?

Locality load balancing policy *
Round robin ?
Each healthy backend is selected in a round robin order.

Outlier detection ?

☐ Enable

17) Click Create

18) Go to routing rules, and keep all the default

19) Click review and finalize, go through all the configurations we have made, and then click Create

20) Once available, open the load balancer, go through the options shown, and also check the backend details

21) Copy the IP address and open it in the browser. (Initially it may give an error as it may take some time for LB to connect with backend services for first time, try again in a few minutes (Also make sure you have allowed all traffic at App engine firewall which we blocked in last lab)

← Load balancer details [EDIT](#) [DELETE](#)

readit-inv-lb
Global external Application Load Balancer

💡 Faster web performance and improved web protection with Cloud CDN and Cloud Armor

[DETAILS](#) [MONITORING](#) [CACHING](#)

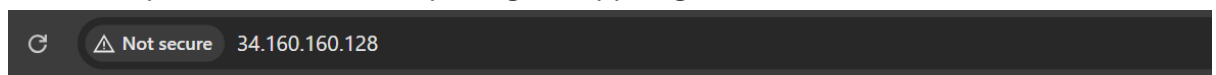
Frontend

Protocol ↑	IP:Port	Certificate	Certificate Map	SSL Policy
HTTP	34.160.160.128:80	-		

Routing rules

Hosts ↑	Paths	Backend
All unmatched (default)	All unmatched (default)	inventory-be

22) This should open our inventory app which we deployed at App Engine using Load Balancer ip address without exposing the app engine details to internet.



ReadIt!

Manage Inventory v2

Nothing to see here folks :-)