Basic Intro

- Systems are connected to each other via a communication link, could be coaxial copper or fiber cables or radio waves
- **Data transmission rate**s are measures in bytes per second (**1 byte is 8 bits**)
- Band- **width** is the maximum possible data transmission rate across a given network

Network Protocols

- 1. Define the **format and the order of messages** exchanged between two entities
- 2. and also the **action taken on transmission or receipt** of the message

The tale of 2 Network Models

This was not used much and eventual went invisible

OSI

The different functions / steps that make up these protocols were divided into layers

HTTP/FTP

7 Application

6 Presentation — Handles rendering of data, audio video codec, PNG, JPEG

5 Session

4 Transport — Transports app layer messages between client and server, TCP/UDP

3 Network — Moves packets of data from one node to another

2 Data Link

1 Physical — The physical device that moves data

- Moves transport layer protocol from one host to another
- Defines fields in the data packet (datagram) and also how end systems + routers act on the field.

(IP - the most popular layer 3 protocol)

merged together into....

Open Systems Inter connect

This became more widespread

TCP / IP

7 Application

4 Transport

3 Network — IP addr & routes

2 Data Link — Mac addresses

1 Physical — Ethernet cables

One weird thing with network engineers is that, while they are using the TCP/IP model, they still use the numbering system in the OSI model to refer to the layers.

1. Physical
2. Data Link
3. Network
4. Transport

Anddd...

7. Application

A way to remember it

7 Application = All
6 Presentation = People
5 Session. = Seem
4 Transport. = To
3 Network. = Need
2 Data Link. = Data
1 Physical. = Processing

Layer 4 - Transport Protocol

Transmission Control Protocol (TCP)

- Reliable - ensures data is sent **correctly** and in **order**
- Connection is kept **alive** till data transfer is successful
- **Erroneous** packets are **retransmitted**
- Has **congestion control**, will not send next chunk until it has received OK (ACK) signal from receiver
- Application layer protocols that use TCP are HTTP (web application servers), FTP, SMTP (email server)...

Layer 4 - Transport Protocol

Basically if it dosen't do anything that TCP does, it is UDP

User Datagram Protocol (UDP)

Designed to do as little work as possible to send data:

- No congestion control - data is transmitted as soon as it can
- Doesn't care if message was successfully received
- No over head of connection establishment
- No tracking of connection state and order of data received
- Use cases: video streaming, Domain Name Systems (DNS) ????

Internet Protocol (IP)

It's job is to relay datagrams **across network boundaries.**
Like UDP it is a best-effort delivery service, makes **no guarantees** about delivering correct data, the order of data and not providing duplicate data
First major version is IPv4, and it's successor is IPv6

IPv4 Addressing System

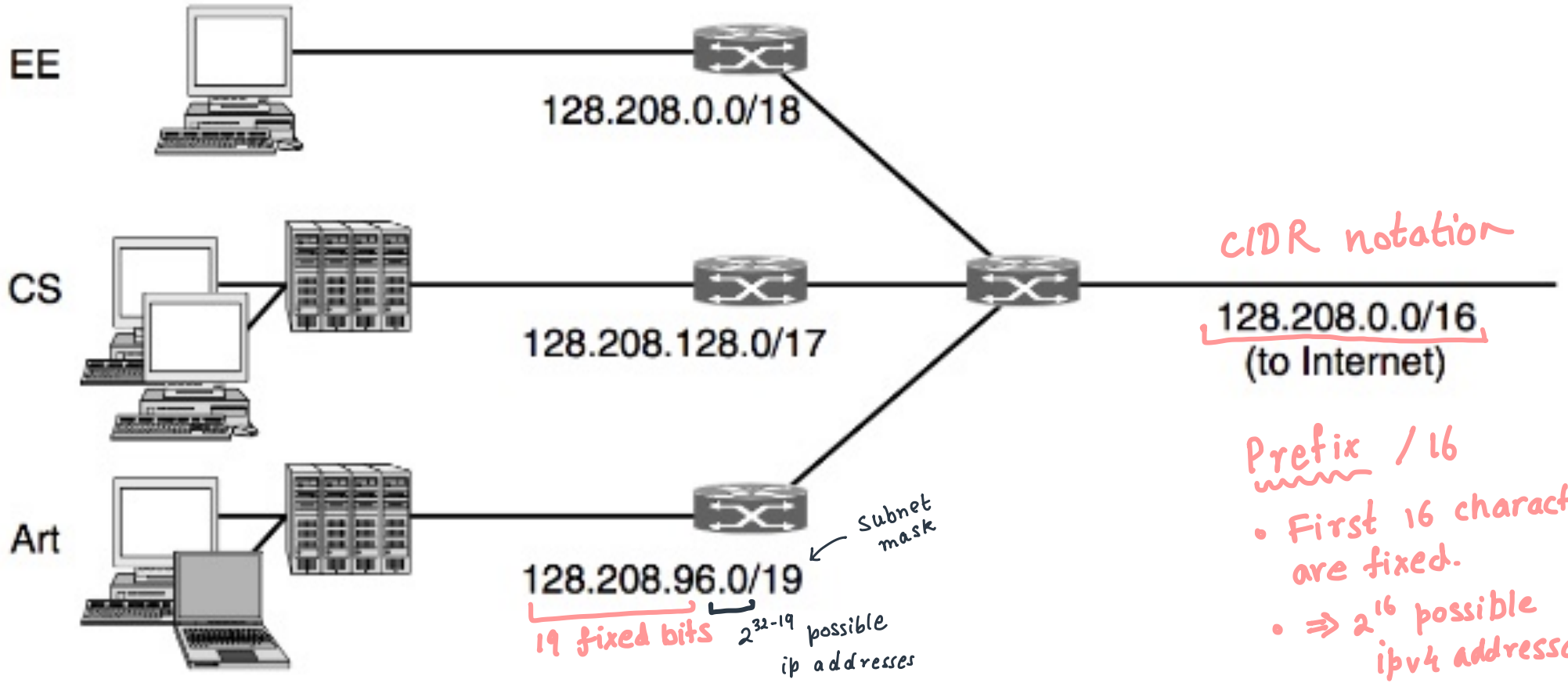
- 2^32 addresses
- 32 bit addressing system
- Each 8 bits are grouped to one number from 0 to 255
- 4 billion IP addresses in total
- 0.0.0.0 or 255.255.255.255
- Exhausted since 2011

IPv6 Addressing System

- Uses 2^128 bit addressing system
- But infra worldwide have not adapted to it
- Uses hexadecimal notation - 8 groups of 4 hexadecimal digits
- 2001:0db8:0000:0042:0000:8a2e:0370:7334
- Each 16 bits are grouped to one hexadecimal number 0 to FFFE (65534)

Subnet:

- An Isolated network to which IP addresses are assigned along with a subnet mask



NAT - Network Address Translation

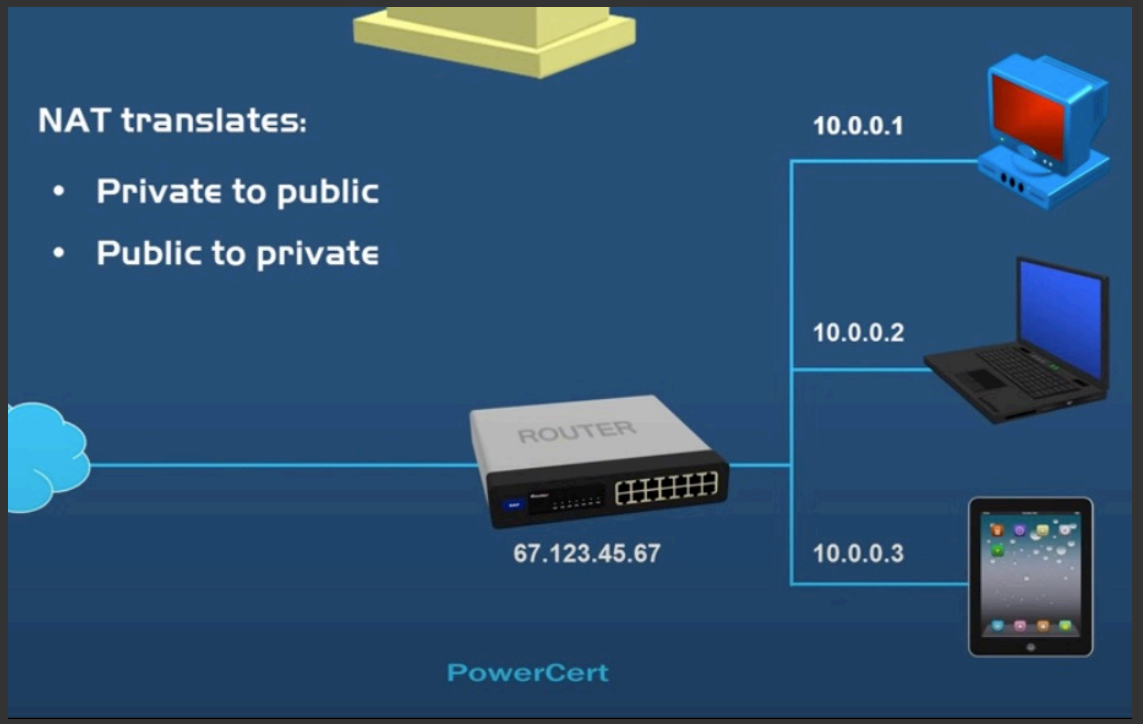
This is used in routers:

- Translates a set of IP addresses into another set of IP addresses
- This is to help preserve the limited number of public IPv4 addresses that we have around the world - the advent of IoT has strengthened the requirement of NAT and also public IPv4 addresses have expired as of 2011

NAT does the translation both ways

- Public to Private IP
- Private to Public IP

This allows for both parties to communicate with each other over the internet

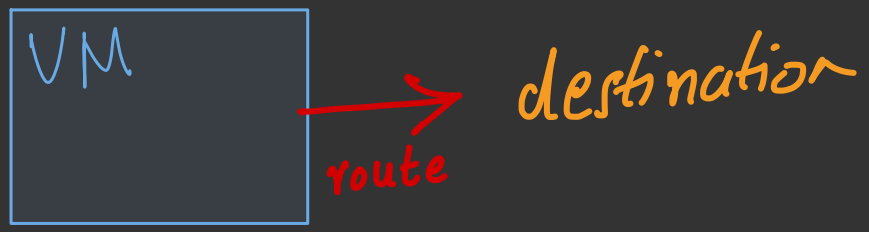


Virtual Private Cloud (VPC)

Is a virtual network that belongs to the user and is logically isolated from other virtual networks / the internet

- The default VPCs on cloud providers comes along with an internet gateway which let's us connect to the internet - https://stackoverflow.com/questions/71133605/what-is-the-meaning-of-gateway-in-the-gcp-vpc-table
- Helps us configure firewall
- Routing table ?
- Private / public subnets
- Network gateways
- Routes ? - https://cloud.google.com/vpc/docs/routes

Routes:



- Google Cloud **routes define the paths that network traffic takes from a virtual machine (VM) instance to other destinations**. These destinations can be inside your Google Cloud Virtual Private Cloud (VPC) network (for example, in another VM) or outside it.
- In a VPC network, a route consists of a single destination prefix in CIDR format and a single next hop. When an instance in a VPC network sends a packet, Google Cloud delivers the packet to the route's next hop if the packet's destination address is within the route's destination range.

Example:

```
resource "google_compute_route"
"zero_for_webapp" {
  name             = var.route_name
  dest_range       = var.destination_range
  network          = your.vpc
  priority         = var.route_priority
  next_hop_gateway = var.next_hop_gateway
}
```

ENI - Elastic Network Interfaces (AWS) / Network Interface (GCP)

VPC Networks are by default isolated private networking domains.

- Every instance in VPC network has a default network interface. When we configure a network interface to an instance, we select a VPC network and a subnet within the VPC network to connect the interface to.
- We **can create multiple network interfaces attached to VMs** but each interface must attach to a **different VPC network**, this let's us create configs in which an instance connects directly to several VPC networks.

Example :

```
network_interface {
  network    = google_compute_network.vpc.self_link
  subnetwork = google_compute_subnetwork.webapp.self_link
  access_config {
    // Ephemeral public IP
  }
}
```

More about Network Interfaces : https://cloud.google.com/vpc/docs/multiple-interfaces-concepts

Every **VM can have up to 8 interfaces**, depending on the instance type.

- Network interfaces can be moved around to various VM instances , however the **primary NI attached to a VM cannot be removed** as it serves as the only way for the VM to be attached to the network.
- Suppose a VM is serving requests in that network, and it goes down, the **interface can immediately be moved to another VM instance** and the service can be continued to be served - this is because the IP address is attached to the NI and not the VM

Each interface can have the following configured:

- Internal IPv4 address (required)
- External IPv4 address
- IPv6 - either internal or external (not both)

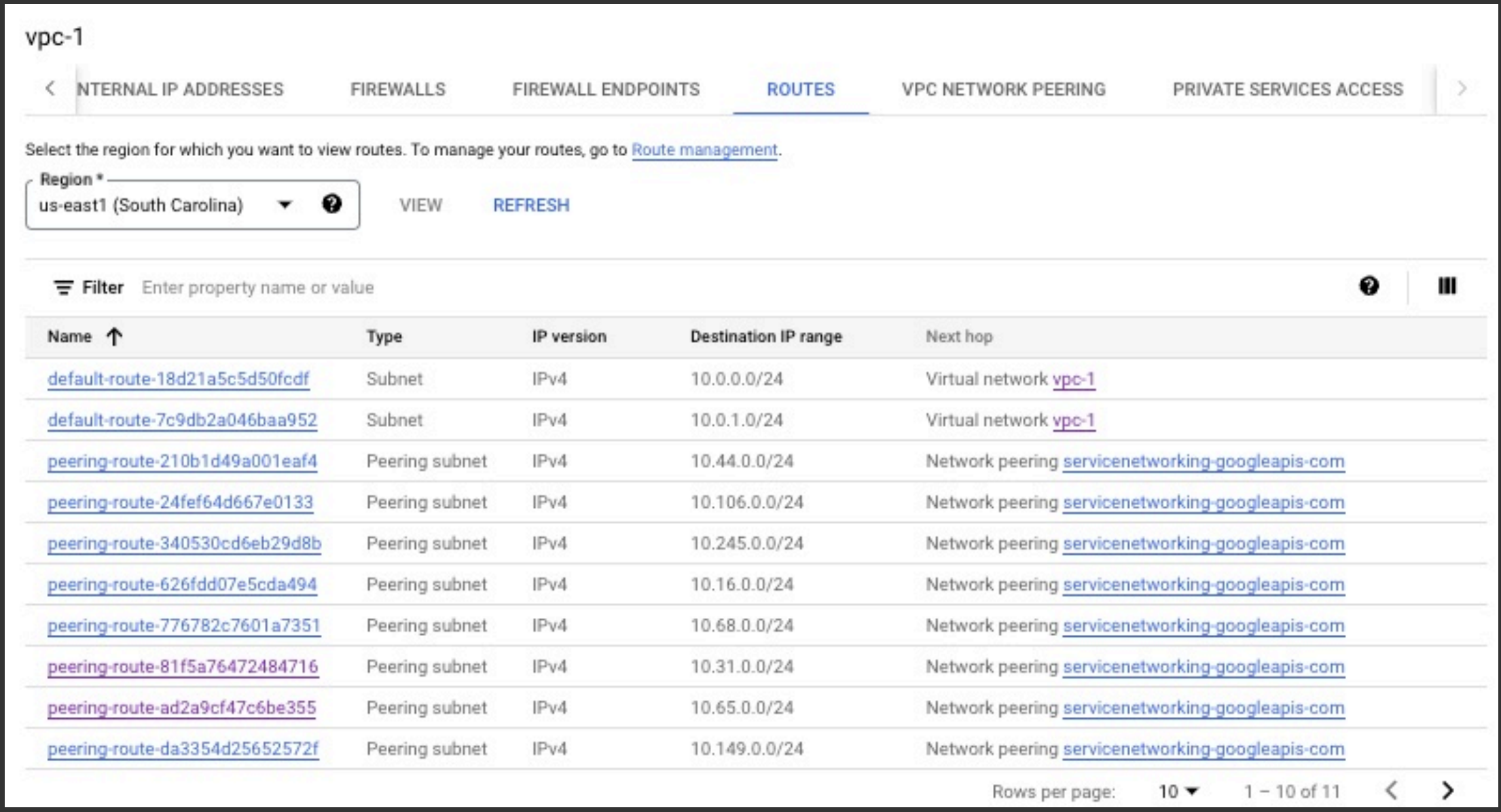
Route Tables

A route table consists a set of routes that are used to determine where traffic is directed

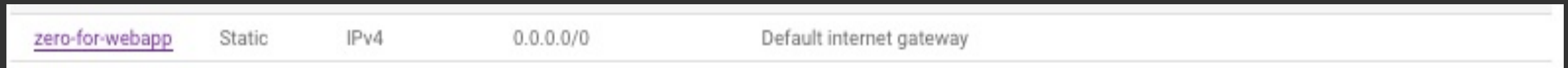
- Each subnet in the VPC must be associated with a route table
- Each subnet can have only one route table, but a single route table can be attached to multiple subnets

In the next page is a routing table for the VPC we created as part of our assignments. We can notice that there are 2 rules mentioned,

1. If a request is directed towards either the 10.0.0.0/24 or the 10.0.1.0/24 range, it must go via the VPC network - as these are related to the subnets we created
2. The remaining rules have been created because we created the private service connection and it has allocated some of the available ranges , so in the next hop we notice the service.networking.api mentioned
3. Scrolling down further we also see a route that directs traffic to the default internet gateway



Route to the internet gateway if it is directed towards 0.0.0.0/0

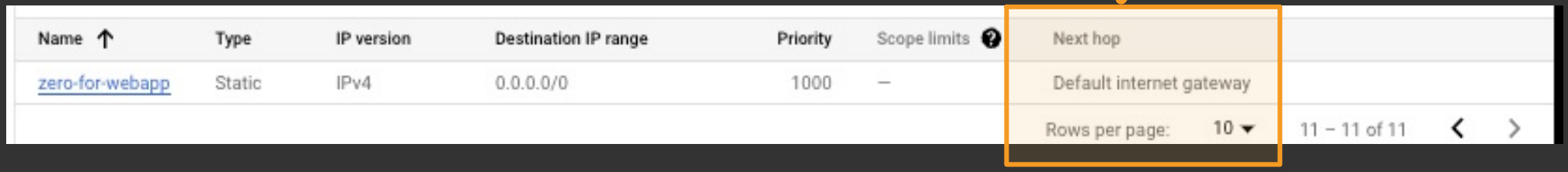


Internet Gateways

Horizontally scaled, **redundant** and a **highly available** VPC **component** that allows for communication between instances in the VPC and the internet

- Imposes no bandwidth constraints on network traffic
- Provides a target in the VPC route table for internet routable traffic
- Performs NAT translation for instances that have been assigned a public IPv4

target / next hop



Name	Type	IP version	Destination IP range	Priority	Scope limits	Next hop
zero-for-webapp	Static	IPv4	0.0.0.0/0	1000	—	Default internet gateway

Rows per page: 10 11 – 11 of 11

Security Groups

Security groups are present at an instance level and not the subnet level

- We add rules to **control inbound and outbound traffic** to the instance
- Each instance can have up to 5 security groups
- It is **easier to manage firewall rules** using SGs than applying it on individual instances.

Inbound			
Source	**Protocol**	**Port Range**	**Comments**
0.0.0.0/0	TCP	80	Allow inbound HTTP access from all IPv4 addresses
::/0	TCP	80	Allow inbound HTTP access from all IPv6 addresses
0.0.0.0/0	TCP	443	Allow inbound HTTPS access from all IPv4 addresses
::/0	TCP	443	Allow inbound HTTPS access from all IPv6 addresses
Your network's public IPv4 address range	TCP	22	Allow inbound SSH access to Linux instances from IPv4 IP addresses in your network (over the Internet gateway)
Your network's public IPv4 address range	TCP	3389	Allow inbound RDP access to Windows instances from IPv4 IP addresses in your network (over the Internet gateway)
Outbound			
Destination	**Protocol**	**Port Range**	**Comments**
The ID of the security group for your database servers	TCP	1433	Allow outbound Microsoft SQL Server access to instances in the specified security group
The ID of the security group for your MySQL database servers	TCP	3306	Allow outbound MySQL access to instances in the specified security group