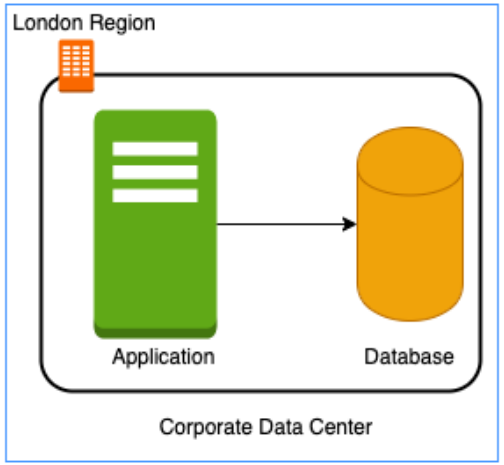
Networking

Need for Google Cloud VPC

- In a corporate network or an on-premises data center:
  - Can anyone on the internet **see the data exchange** between the application and the database?
    - **No**
  - Can anyone from internet **directly connect to your database**?
    - Typically **NO**.
    - You need to connect to your corporate network and then access your applications or databases.
- Corporate network provides a **secure internal network** protecting your resources, data and communication from external users
- How do you do create **your own private network** in the cloud?
  - Enter **Virtual Private Cloud (VPC)**



Google Cloud VPC (Virtual Private Cloud)

In 28 Minutes

- Your **own isolated network** in GCP cloud
  - Network traffic within a VPC is isolated (not visible) from all other Google Cloud VPCs
- You **control all the traffic** coming in and going outside a VPC
- (**Best Practice**) Create all your GCP resources (compute, storage, databases etc) **within a VPC**
  - Secure resources from unauthorized access AND
  - Enable secure communication between your cloud resources
- VPC is a **global resource** & contains subnets in one or more region
  - (REMEMBER) NOT tied to a region or a zone. VPC resources can be in any region or zone!

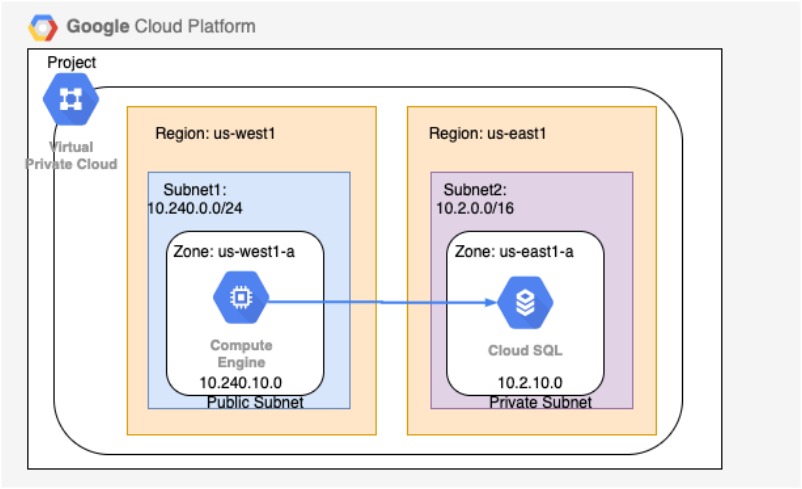
Virtual Private Cloud

Need for VPC Subnets

- Different types of resources are created on cloud - databases, compute etc
  - Each type of resource has **its own access needs**
  - Load Balancers are accessible from internet (**public** resources)
  - Databases or VM instances should NOT be accessible from internet
    - ONLY applications within your network (VPC) should be able to access them(**private** resources)
- How do you **separate public resources from private resources** inside a VPC?
  - Create separate Subnets!
- (Additional Reason) You want to distribute resources across multiple regions for high availability

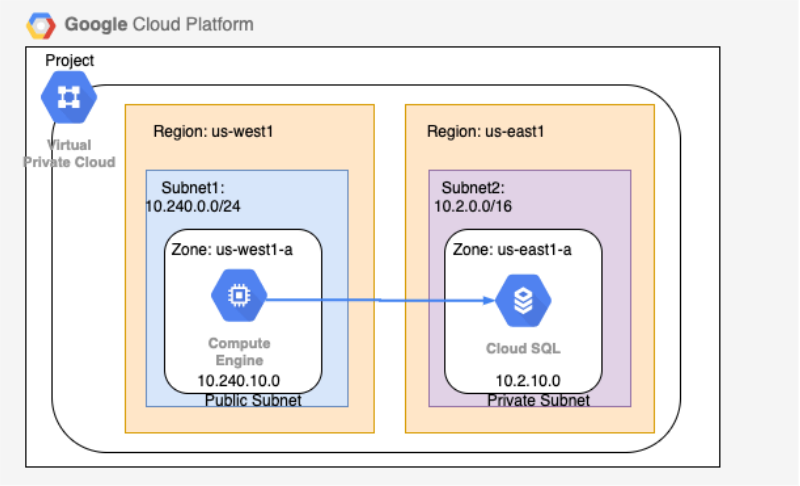
VPC Subnets

- (Solution) **Create different subnets** for public and private resources
  - Resources in a public subnet **CAN** be accessed from internet
  - Resources in a private subnet **CANNOT** be accessed from internet
  - BUT resources in public subnet can talk to resources in private subnet
- Each Subnet is created in a region
- **Example** : VPC - demo-vpc => Subnets - region us-central1, europe-west1 or us-west1 or ..



Creating VPCs and Subnets

- By default, every project has a default VPC
- You can create YOUR own VPCs:
  - **OPTION 1**: Auto mode VPC network:
    - Subnets are automatically created in each region
    - Default VPC created automatically in the project uses auto mode!
  - **OPTION 2**: Custom mode VPC network:
    - No subnets are automatically created
    - You have complete control over subnets and their IP ranges
    - Recommended for Production
- Options when you create a subnet:
  - Enable **Private Google Access** - Allows VM's to connect to Google API's using private IP's
  - Enable **FlowLogs** - To troubleshoot any VPC related network issues



CIDR (Classless Inter-Domain Routing) Blocks

In28 Minutes

- Resources in a network use continuous IP addresses to make routing easy:
  - Example: Resources inside a specific network can use IP addresses from 69.208.0.0 to 69.208.0.15
- How do you express a **range of addresses** that resources in a network can have?
  - **CIDR block**
- A **CIDR block** consists of a **starting IP address(69.208.0.0)** and a **range(/28)**
  - Example: CIDR block 69.208.0.0/28 represents addresses from 69.208.0.0 to 69.208.0.15 - a total of 16 addresses
- **Quick Tip**: 69.208.0.0/28 indicates that the first 28 bits (out of 32) are fixed.
  - Last 4 bits can change => 2 to the power 4 = 16 addresses

CIDR Exercises

In28 Minutes

CIDR	Start Range	End Range	Total addresses	Bits selected in IP address
69.208.0.0/24	69.208.0.0	69.208.0.255	256	01000101.11010000.00000000.********
69.208.0.0/25	69.208.0.0	69.208.0.127	128	01000101.11010000.00000000.0*******
69.208.0.0/26	69.208.0.0	69.208.0.63	64	01000101.11010000.00000000.00******
69.208.0.0/27	69.208.0.0	69.208.0.31	32	01000101.11010000.00000000.000*****
69.208.0.0/28	69.208.0.0	69.208.0.15	16	01000101.11010000.00000000.0000****
69.208.0.0/29	69.208.0.0	69.208.0.7	8	01000101.11010000.00000000.00000***
69.208.0.0/30	69.208.0.0	69.208.0.3	4	01000101.11010000.00000000.000000**
69.208.0.0/31	69.208.0.0	69.208.0.1	2	01000101.11010000.00000000.0000000*
69.208.0.0/32	69.208.0.0	69.208.0.0	1	01000101.11010000.00000000.00000000

- Exercise : How many addresses does **69.208.0.0/26** represent?
  - 2 to the power (32-26 = 6) = 64 addresses from 69.208.0.0 to 69.208.0.63
- Exercise : How many addresses does **69.208.0.0/30** represent?
  - 2 to the power (32-30 = 2) = 4 addresses from 69.208.0.0 to 69.208.0.3
- Exercise : What is the difference between **0.0.0.0/0** and **0.0.0.0/32**?
  - 0.0.0.0/0 represent all IP addresses. 0.0.0.0/32 represents just one IP address 0.0.0.0.

Examples of Recommended CIDR Blocks - VPC Subnets

In 28 Minutes

- **Recommended CIDR Blocks**
  - Private IP addresses RFC 1918: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - Shared address space RFC 6598: 100.64.0.0/10
  - IETF protocol assignments RFC 6890: 192.0.0.0/24
- **Restricted Range Examples**
  - You CANNOT use these as CIDR for VPC Subnets
    - Private Google Access-specific virtual IP addresses: 199.36.153.4/30, 199.36.153.8/30
    - Current (local) network RFC 1122: 0.0.0.0/8
    - Local host RFC 1122: 127.0.0.0/8
- (REMEMBER) You **CAN EXTEND** the CIDR Block Range of a Subnet (Secondary CIDR Block)

Virtual Private Cloud

Firewall Rules

In 28 Minutes

- Configure Firewall Rules to control traffic going in or out of the network:
  - Stateful
  - Each firewall rule has priority (0-65535) assigned to it
  - 0 has highest priority. 65535 has least priority
  - Default implied rule with lowest priority (65535)
    - Allow all egress
    - Deny all ingress
    - Default rules can't be deleted
    - You can override default rules by defining new rules with priority 0-65534
  - Default VPC has 4 additional rules with priority 65534
    - Allow incoming traffic from VM instances in same network (**default-allow-internal**)
    - Allow Incoming TCP traffic on port 22 (SSH) **default-allow-ssh**
    - Allow Incoming TCP traffic on port 3389 (RDP) **default-allow-rdp**
    - Allow Incoming ICMP from any source on the network **default-allow-icmp**

Virtual Private Cloud

Firewall Rules - Ingress and Egress Rules

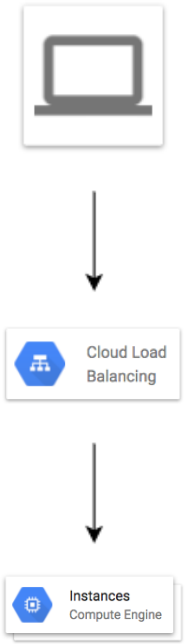
In 28 Minutes

- **Ingress Rules**: Incoming traffic from outside to GCP targets
  - **Target (defines the destination)**: All instances or instances with TAG/SA
  - **Source (defines where the traffic is coming from)**: CIDR or All instances or instances with TAG/SA
- **Egress Rules**: Outgoing traffic to destination from GCP targets
  - **Target (defines the source)**: All instances or instances with TAG/SA
  - **Destination**: CIDR Block
- **Along with each rule**, you can also define:
  - **Priority** - Lower the number, higher the priority
  - **Action on match** - Allow or Deny traffic
  - **Protocol** - ex. TCP or UDP or ICMP
  - **Port** - Which port?
  - **Enforcement status** - Enable or Disable the rule

Virtual Private Cloud

Firewall Rules - Best Practices

- Use **network tags** and control allowed traffic into a VM using firewall rules
- Ensure that firewall rule allow the right kind of traffic:
  - Only allow traffic from load balancing into VM instances
  - Remove `0.0.0.0/0` from Source IP ranges
    - Add `130.211.0.0/22` and `35.191.0.0/16`
    - Allows health checks from load balancing to VM instances
- (REMEMBER) All egress from an VM instance is allowed by default:
  - To allow Specific EGRESS ONLY
    - 1: Create an egress rule with low priority to deny all traffic
    - 2: Create egress rule with high priority to allow traffic on specific port



Shared VPC

In 28 Minutes

- Scenario: Your organization has multiple projects. You want resources in different projects to talk to each other?
  - How to allow resources in different projects to talk with internal IPs securely and efficiently?
- Enter **Shared VPC**
  - Created at organization or shared folder level (Access Needed: Shared VPC Admin)
  - Allows VPC network to be shared between projects in same organization
  - Shared VPC contains one host project and multiple service projects:
    - **Host Project** - Contains shared VPC network
    - **Service Projects** - Attached to host projects
- Helps you achieve **separation of concerns**:
  - Network administrators responsible for Host projects and Resource users use Service Project

VPC Peering

In 28 Minutes

- Scenario: How to connect VPC networks across different organizations?
- Enter **VPC Peering**
  - Networks in same project, different projects and across projects in different organizations can be peered
  - All communication happens using internal IP addresses
    - Highly efficient because all communication happens **inside Google network**
    - Highly secure because **not accessible from Internet**
    - **No data transfer charges** for data transfer between services
  - (REMEMBER) Network administration is NOT changed:
    - Admin of one VPC do not get the role automatically in a peered network

Virtual Private Cloud