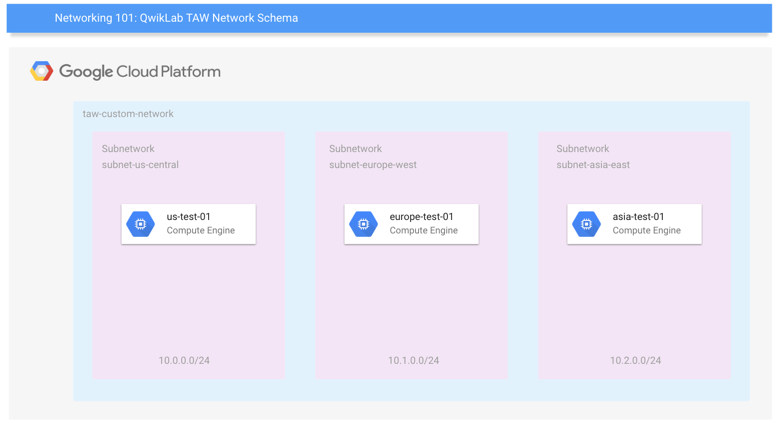
**Networking 101**

**Overview**

In this lab you will learn how to perform basic networking tasks on Google Cloud (including Compute Engine instances) and how Google Cloud might differ from an on-premises setup. You'll develop a network and 3 subnetworks, resulting in this end-state environment:



Finally, you'll learn how to create firewall rules and use instance tags to apply the firewall rules.

What you'll learn

* Basics concepts and constructs of Google Cloud networking
* How default and user-created networks are configured.
* How to create firewall rules, and use instance tags to apply firewall rules

Understanding Regions and Zones

Certain Compute Engine resources live in regions or zones. A region is a specific geographical location where you can run your resources. Each region has one or more zones. For example, the us-central1 region denotes a region in the Central United States that has zones us-central1-a, us-central1-b, us-central1-c, and us-central1-f.

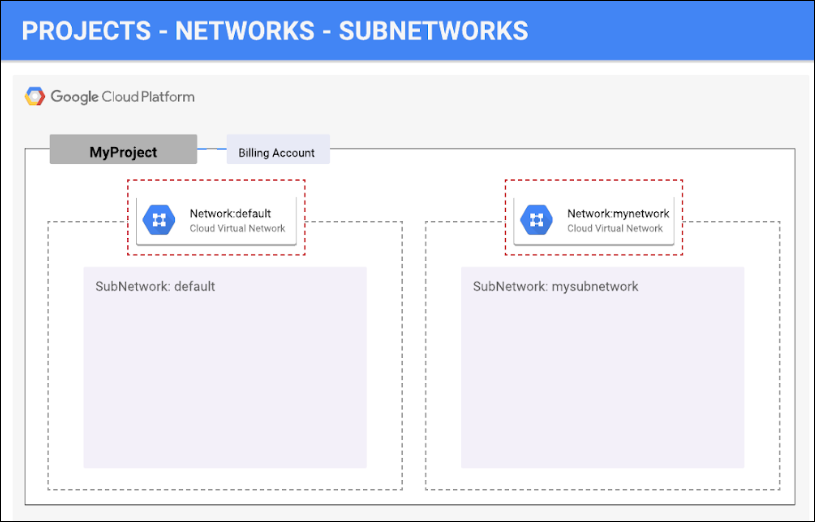
|  |  |
| --- | --- |
| Regions | Zones |
| Western US | us-west1-a, us-west1-b |
| Central US | us-central1-a, us-central1-b, us-central1-d, us-central1-f |
| Eastern US | us-east1-b, us-east1-c, us-east1-d |
| Western Europe | europe-west1-b, europe-west1-c, europe-west1-d |
| Eastern Asia | asia-east1-a, asia-east1-b, asia-east1-c |

Resources that live in a zone are referred to as zonal resources. Virtual machine Instances and persistent disks live in a zone. To attach a persistent disk to a virtual machine instance, both resources must be in the same zone. Similarly, if you want to assign a static IP address to an instance, the instance must be in the same region as the static IP.

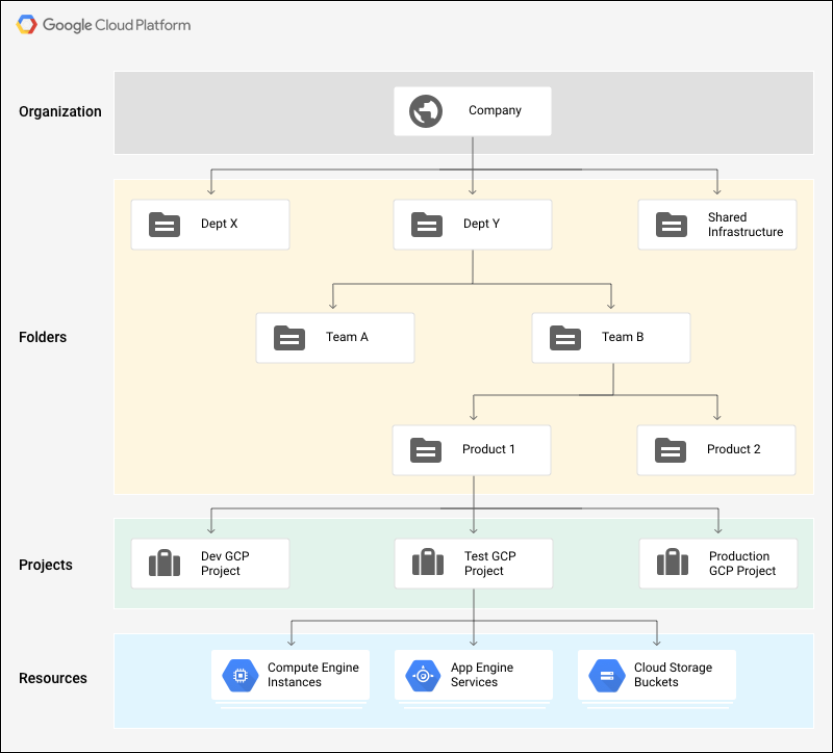
**Google Cloud Network Concepts**

In Google Cloud Platform, networks provide data connections into and out of your cloud resources (mostly Compute Engine instances). Securing your Networks is critical to securing your data and controlling access to your resources.

Google Cloud Platform supports Projects, Networks, and Subnetworks to provide flexible, logical isolation of unrelated resources.



***Projects*** are the outermost container and are used to group resources that share the same trust boundary. Many developers map Projects to teams since each Project has its own access policy (IAM) and member list. Projects also serve as a collector of billing and quota details reflecting resource consumption. Projects contain Networks which contain Subnetworks, Firewall rules, and Routes (see below architecture diagrams for illustration).



***Networks*** directly connect your resources to each other and to the outside world. Networks, using Firewalls, also house the access policies for incoming and outgoing connections. Networks can be Global (offering horizontal scalability across multiple Regions) or Regional (offering low-latency within a single Region).

***Subnetworks*** allow you to group related resources (Compute Engine instances) into RFC1918 private address spaces. Subnetworks can only be Regional. A subnetwork can be in auto mode or custom mode.

* An auto mode network has one subnet per region, each with a predetermined IP range and gateway. These subnets are created automatically when you create the auto mode network, and each subnet has the same name as the overall network.
* A custom mode network has no subnets at creation. In order to create an instance in a custom mode network, you must first create a subnetwork in that region and specify its IP range. A custom mode network can have zero, one, or many subnets per region.

Firewalls

for more information on how you can use firewall rules to isolate subnetworks, refer to subnetworks and firewall rules.

Each network has a default firewall that blocks all inbound traffic to instances. To allow traffic to come into an instance, you must create "allow" rules for the firewall. Additionally, the default firewall allows traffic from instances unless you configure it to block outbound connections using an "egress" firewall configuration. Therefore, by default you can create "allow" rules for traffic you wish to pass ingress, and "deny" rules for traffic you wish to restrict egress. You may also create a default-deny policy for egress and prohibit external connections entirely.

In general, it is recommended to configure the least permissive firewall rule that will support the kind of traffic you are trying to pass. For example, if you need to allow traffic to reach some instances, but restrict traffic from reaching others, create rules that allow traffic to the intended instances only. This more restrictive configuration is more predictable than a large firewall rule that allows traffic to all of the instances. If you want to have "deny" rules to override certain "allow" rules, you can set priority levels on each rule and the rule with the lowest numbered priority will be evaluated first. Creating large and complex sets of override rules can lead to allowing or blocking traffic that is not intended.

The default network has automatically created firewall rules, which are shown below. No manually created network of any type has automatically created firewall rules. For all networks except the default network, you must create any firewall rules you need.

The ingress firewall rules automatically created for the default network are as follows:

|  |  |
| --- | --- |
| default-allow-internal | Allows network connections of any protocol and port between instances on the network. |
| default-allow-ssh | Allows SSH connections from any source to any instance on the network over TCP port 22. |
| default-allow-rdp | Allows RDP connections from any source to any instance on the network over TCP port 3389. |
| default-allow-icmp | Allows ICMP traffic from any source to any instance on the network. |

Network route

All networks have routes created automatically to the Internet (default route) and to the IP ranges in the network. The route names are automatically generated and will look different for each project.

When manually assigning subnetwork ranges, you first create a custom network, then create the subnets you want within a region. You do not have to specify subnetworks for all regions right away, or even at all, but you cannot create instances in regions that have no subnetwork defined.

When you create a new subnetwork, its name must be unique in that project for that region, even across networks. The same name can appear twice in a project as long as each one is in a different region. Because this is a subnetwork, there is no network-level IPv4 range or gateway IP, so none will be displayed.

You can either create your custom network with the console or with Cloud Shell. You'll be shown you both, but you have to decide which method to use while taking the lab. For example, you cannot go through a section using the instructions for the console, then go through the same section using gcloud command line.

To allow access to VM instances, you must apply firewall rules. For this lab, you will use an instance tag to apply the firewall rule to your VM instances. The firewall rule will apply to any VM using the same instance tag.

**Note:**Instance Tags are used by networks and firewalls to apply certain firewall rules to tagged VM instances. For example, if there are several instances that perform the same task, such as serving a large website, you can tag these instances with a shared word or term and then use that tag to allow HTTP access to those instances with a firewall rule.

Tags are also reflected in the metadata server, so you can use them for applications running on your instances.