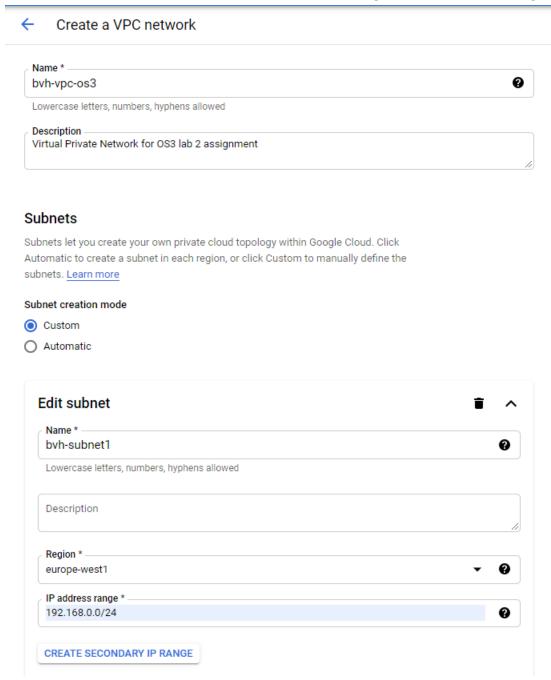
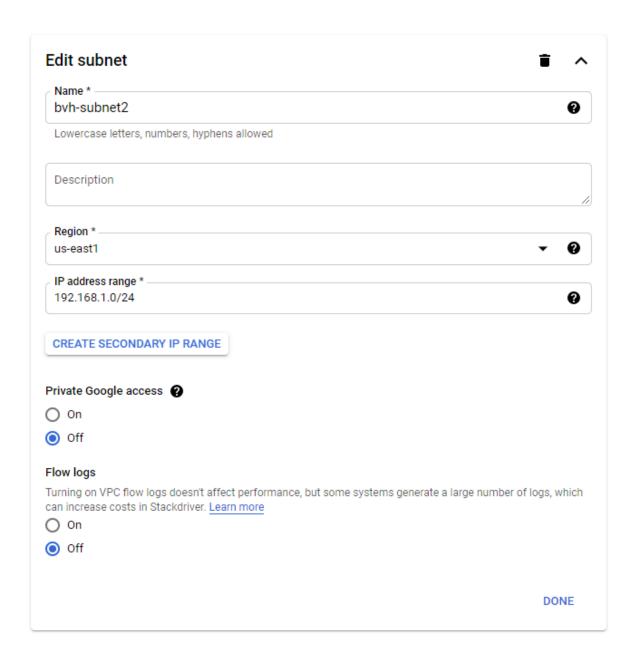
Lab 2: Cloud Project

Step 1: Create the VPC network

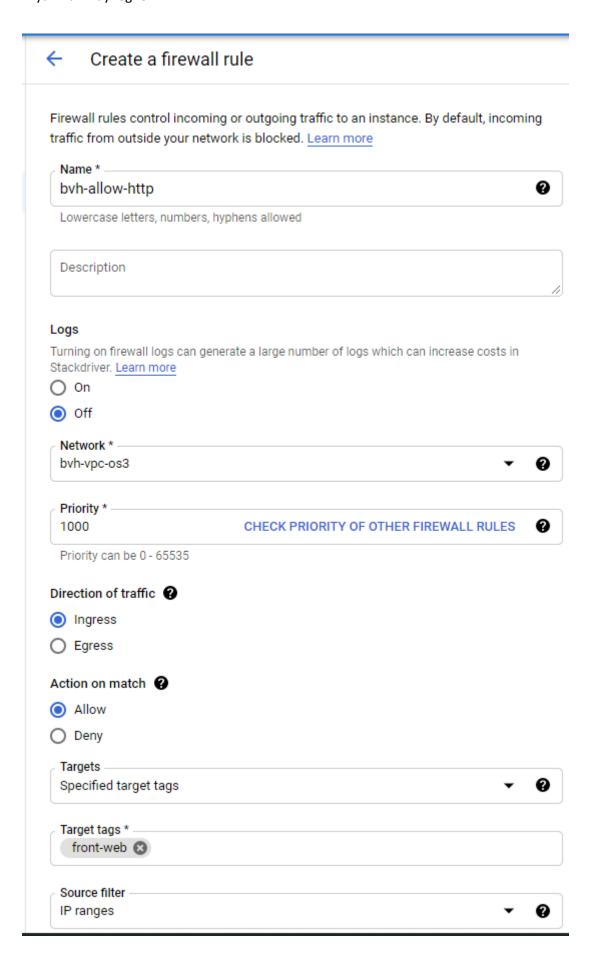
We create a VPC network with two subnets, one for the US region and one for the EU region.

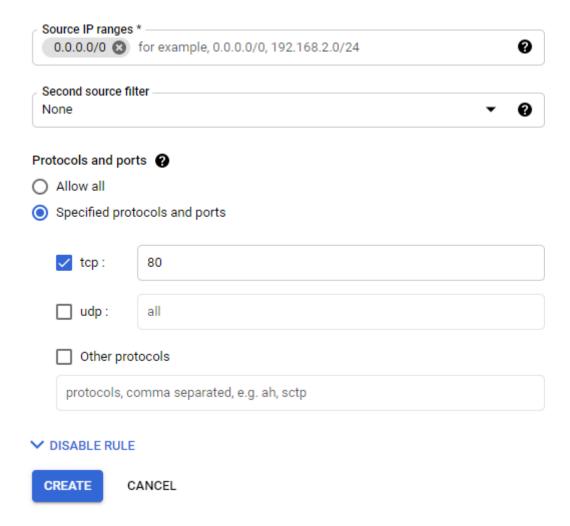


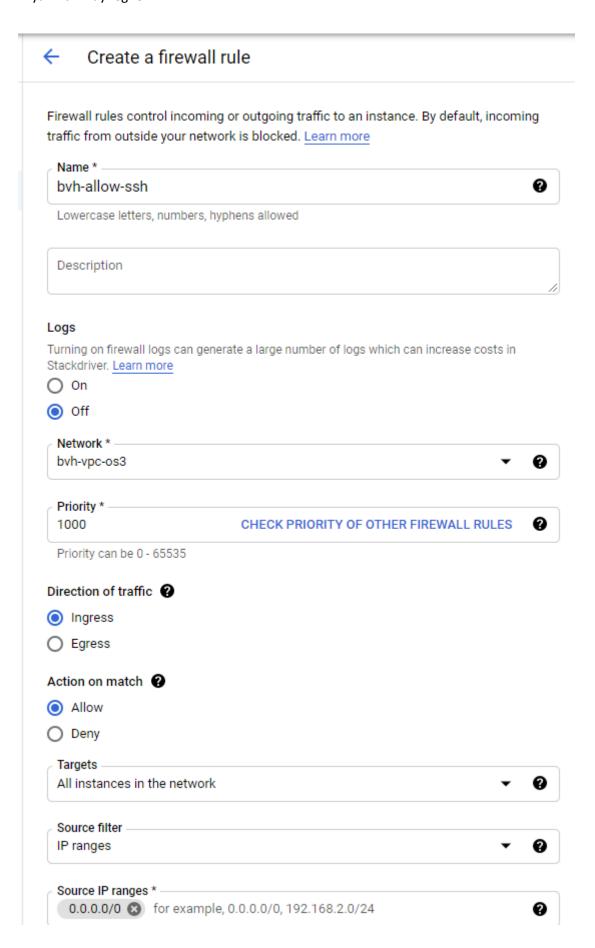


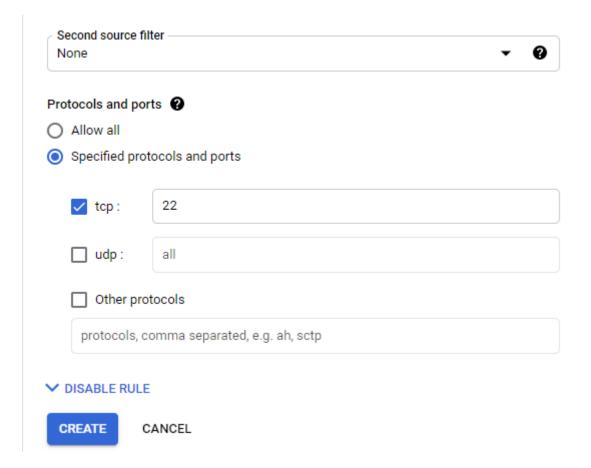
Step 2: Create the Firewall rules

We will need two firewall rules: one that allows HTTP traffic from port 80 and one that allows SSH from port 22 so we can connect to our VMs.



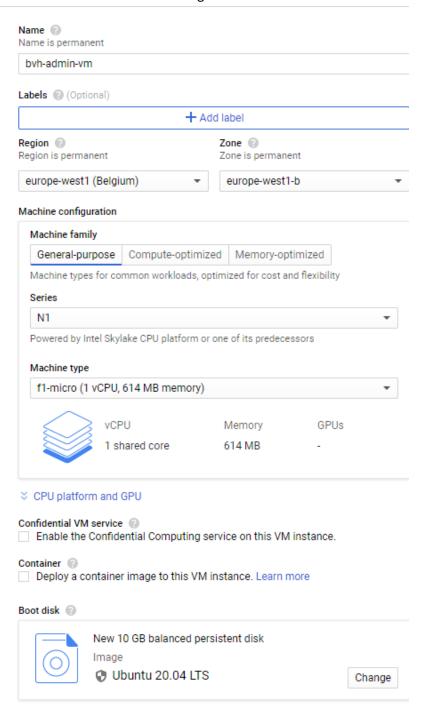


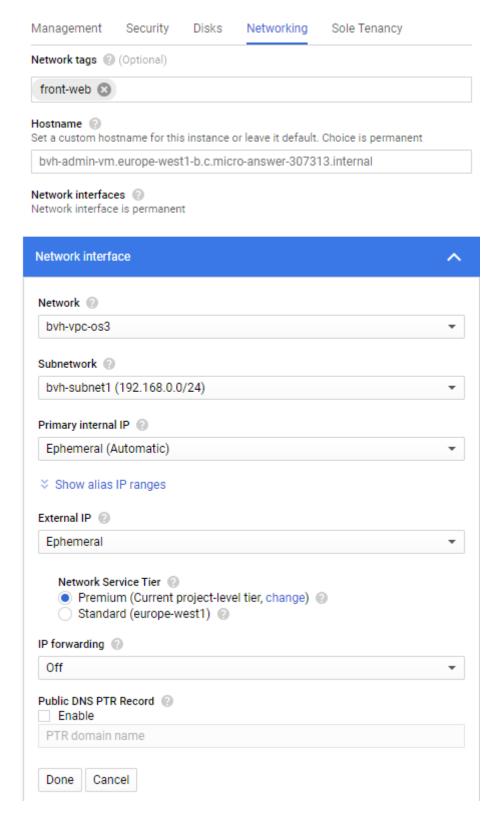




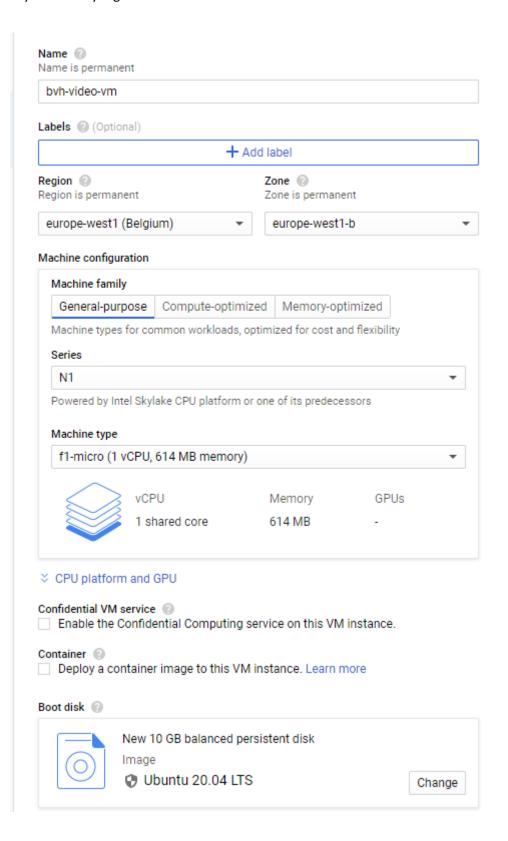
Step 3: Provision VMs

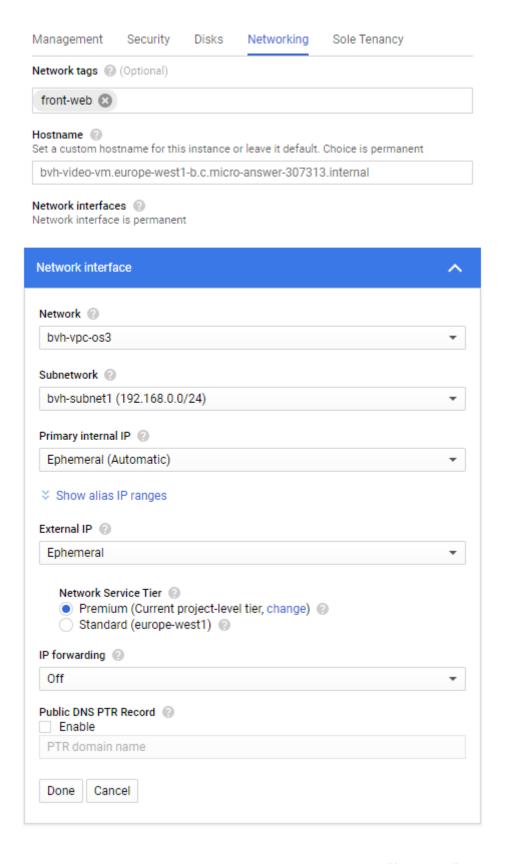
We will need one VM for the admin service and one VM for the video service to ensure that updates can be rolled out without affecting one another.





This VM will be deployed in the Europe-west1 region using the "front-web" network-tag and a custom VPC "bvh-vpc-os3".





This VM will be deployed in the Europe-west1 region using the "front-web" network-tag and a custom VPC "bvh-vpc-os3".

This results in the following:

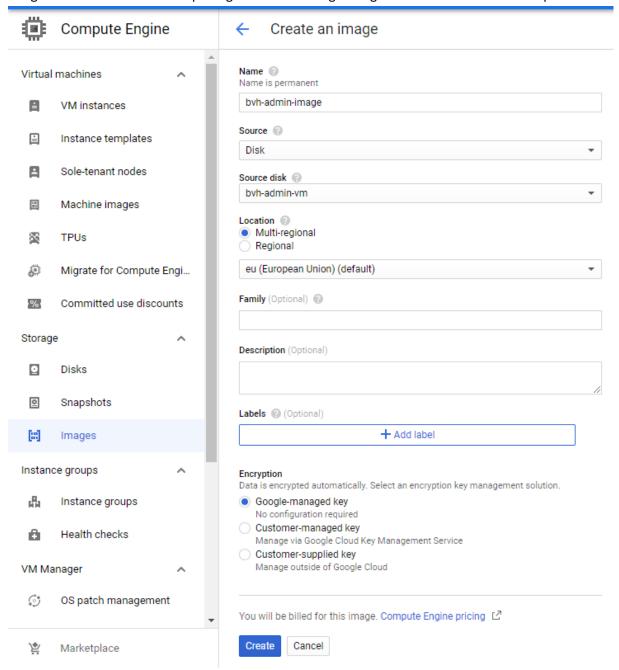


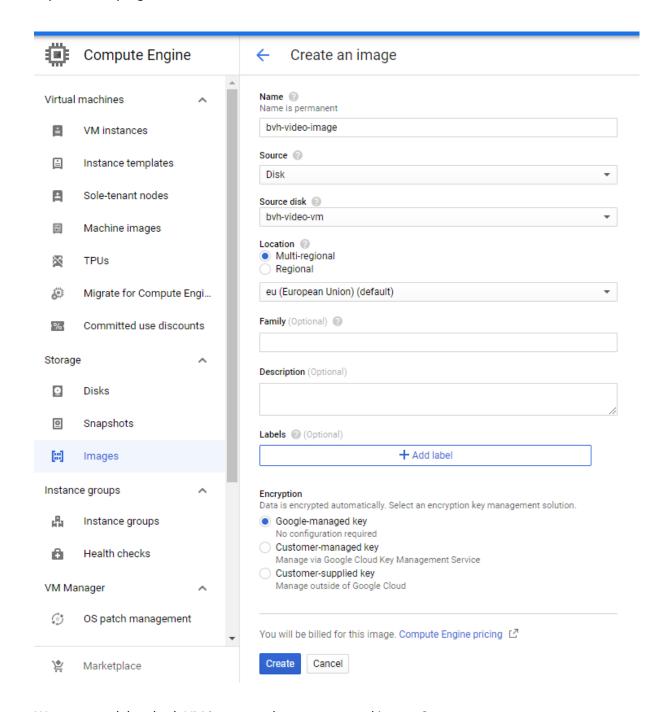
We will now connect to our VMs using SSH and execute the following script on both VMs:

```
root@bvh-admin-vm:~# systemctl status apache2
apache2.service - The Apache HTTP Server
    Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset; enabled)
    Active: active (running) since Tue 2021-03-16 12:46:53 UTC; 17s ago
      Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2286 (apache2)
     Tasks: 55 (limit: 682)
    Memory: 5.4M
    CGroup: /system.slice/apache2.service
             —2286 /usr/sbin/apache2 -k start
             —2288 /usr/sbin/apache2 -k start
             L2289 /usr/sbin/apache2 -k start
Mar 16 12:46:53 bvh-admin-vm systemd[1]: Starting The Apache HTTP Server...
Mar 16 12:46:53 bvh-admin-vm systemd[1]: Started The Apache HTTP Server.
root@bvh-video-vm:~# systemctl status apache2
 apache2.service - The Apache HTTP Server
    Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
    Active: active (running) since Tue 2021-03-16 12:50:15 UTC; 28s ago
      Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2247 (apache2)
     Tasks: 55 (limit: 682)
    Memory: 5.4M
    CGroup: /system.slice/apache2.service
              -2247 /usr/sbin/apache2 -k start
               -2249 /usr/sbin/apache2 -k start
             Mar 16 12:50:15 bvh-video-vm systemd[1]: Starting The Apache HTTP Server...
Mar 16 12:50:15 bvh-video-vm systemd[1]: Started The Apache HTTP Server.
```

Step 4: Create disk image

We create the disk images for video and admin from the VMs created in the previous step. This image will be available in multiple regions because of growing demand in the US and Europe.



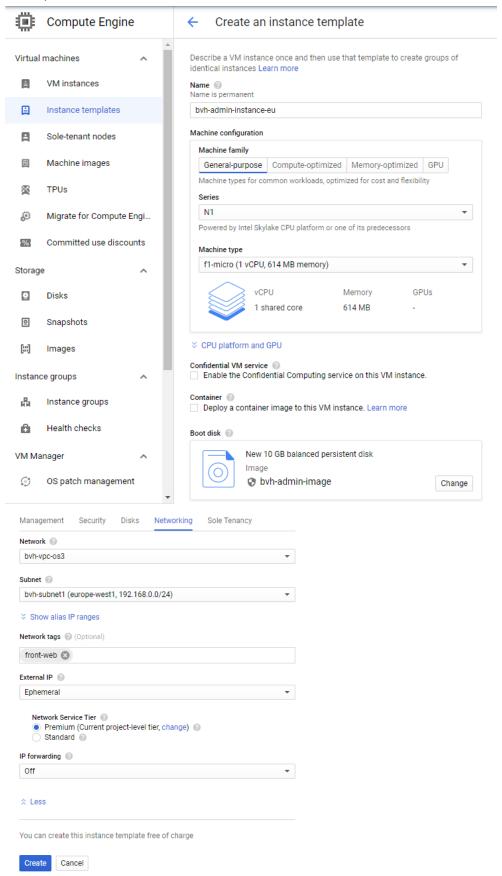


We can now delete both VM instances that were created in step 3.

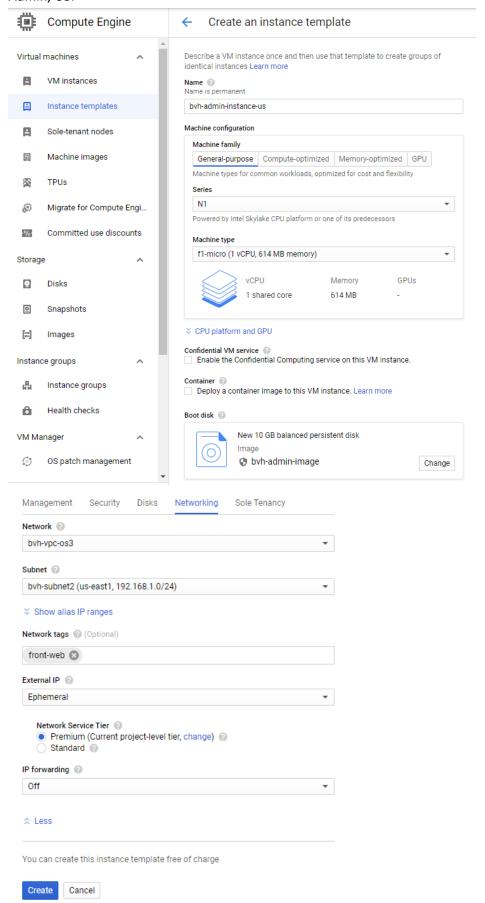
Step 5: Create the instance template

We will now create two instance templates (one for each region) for both our video and admin services using the disk images created in the previous step.

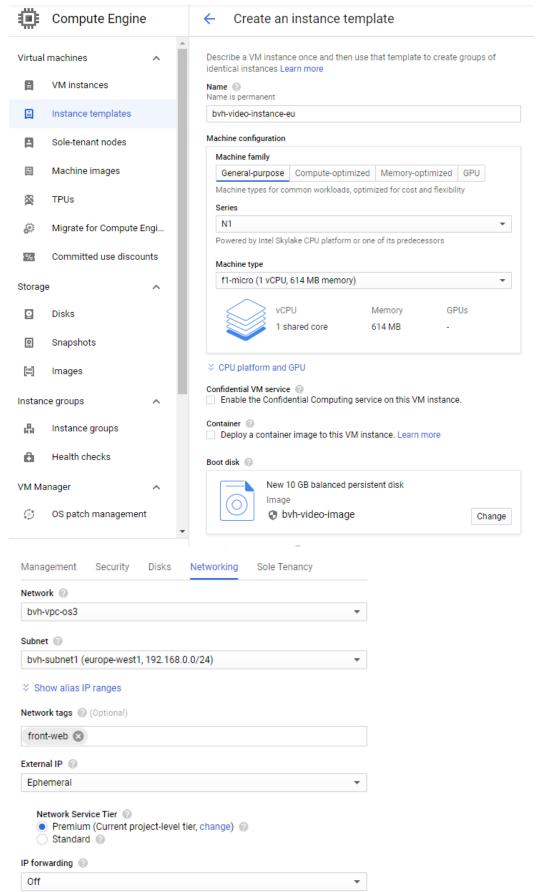
Admin, EU:



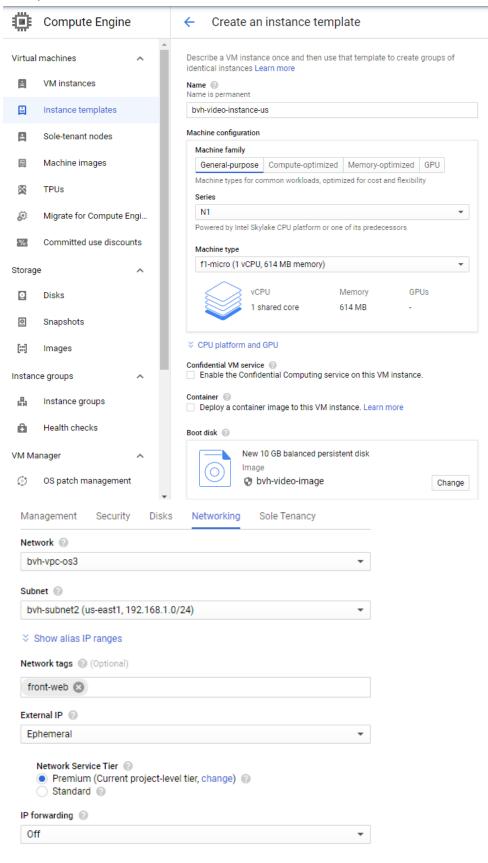
Admin, US:



Video, EU:

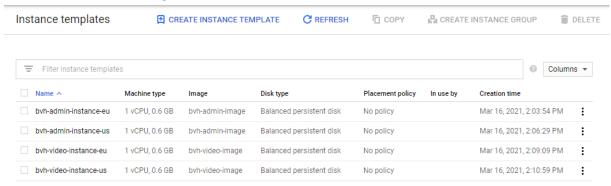


Video, US:



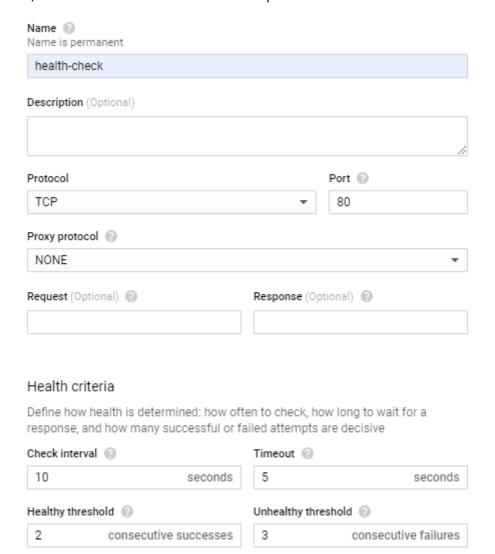
☆ Less

This results in the following:

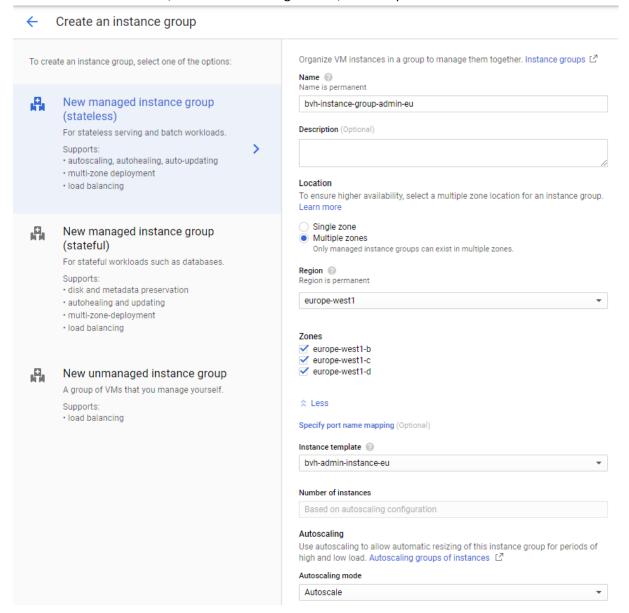


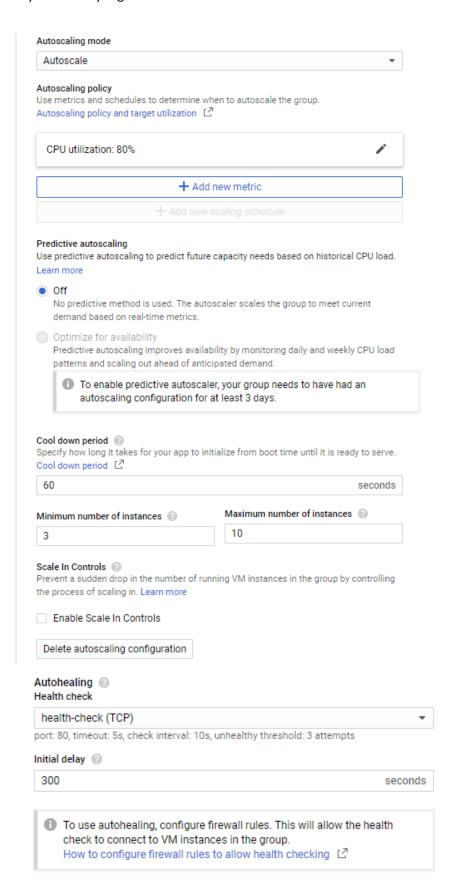
Step 6: Create the instance groups

First, we create a TCP health check to see if port 80 is available.



This is the instance group for admin in EU. It will have at minimum 3 instances spread over multiple zones (3) as suggested by GCP. It uses auto scaling to scale up or down. GCP suggests having the minimum number of instances be equal to the number of zones, in our case 3, to ensure high availability, as per the requirement of the assignment. Scaling occurs when the CPU load is over 80%. We use TCP health checks, created in the image above, to see if port 80 is available.



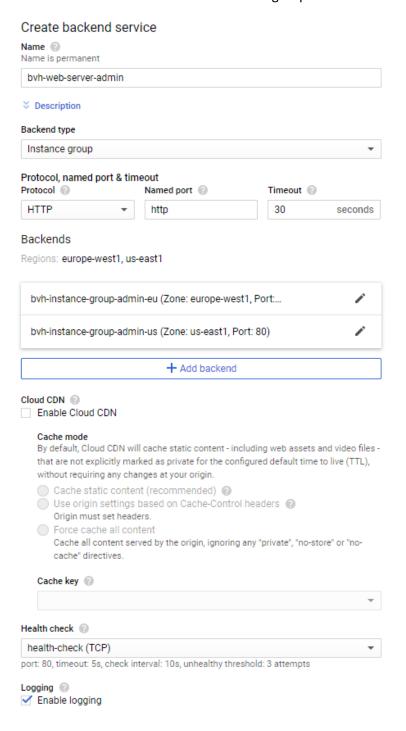


We have now created the instance group for admin in the EU.

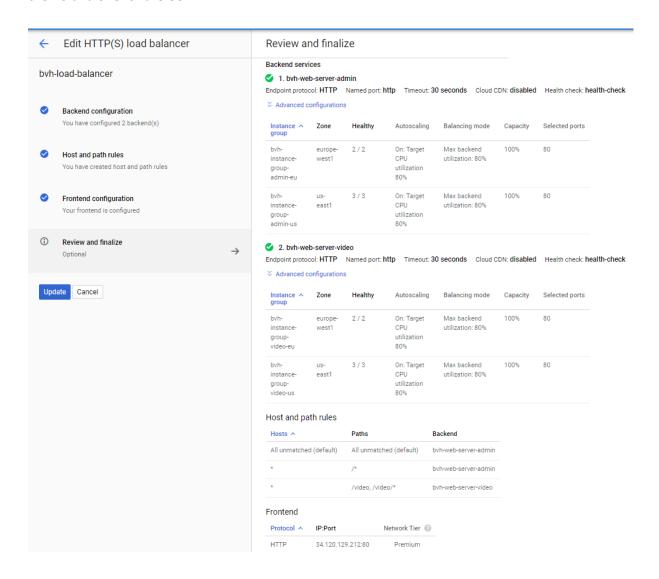
We create 3 more of these instance groups: one instance group for admin in the US, one instance group for video in the EU and one instance group for admin the US. The creation of these groups is similar to the previous and is therefore not shown.

Step 7: Create the load balancer

We create the load balancer for the instance groups for admin and video, adding these as backends.



This results in the following, where we have specified the host and path rules such that all default traffic is routed to our admin service, and traffic starting with video is routed to our video service. Notice both backends, one for each service, with each backend containing 2 instance groups, one for the EU and one for the US.



The final overview of the load balancer can be seen in the image below. It has appropriate host and path rules to deal with /video requests. The backend services are configured correctly, linking to our instance groups for admin and video for both the US and EU.

