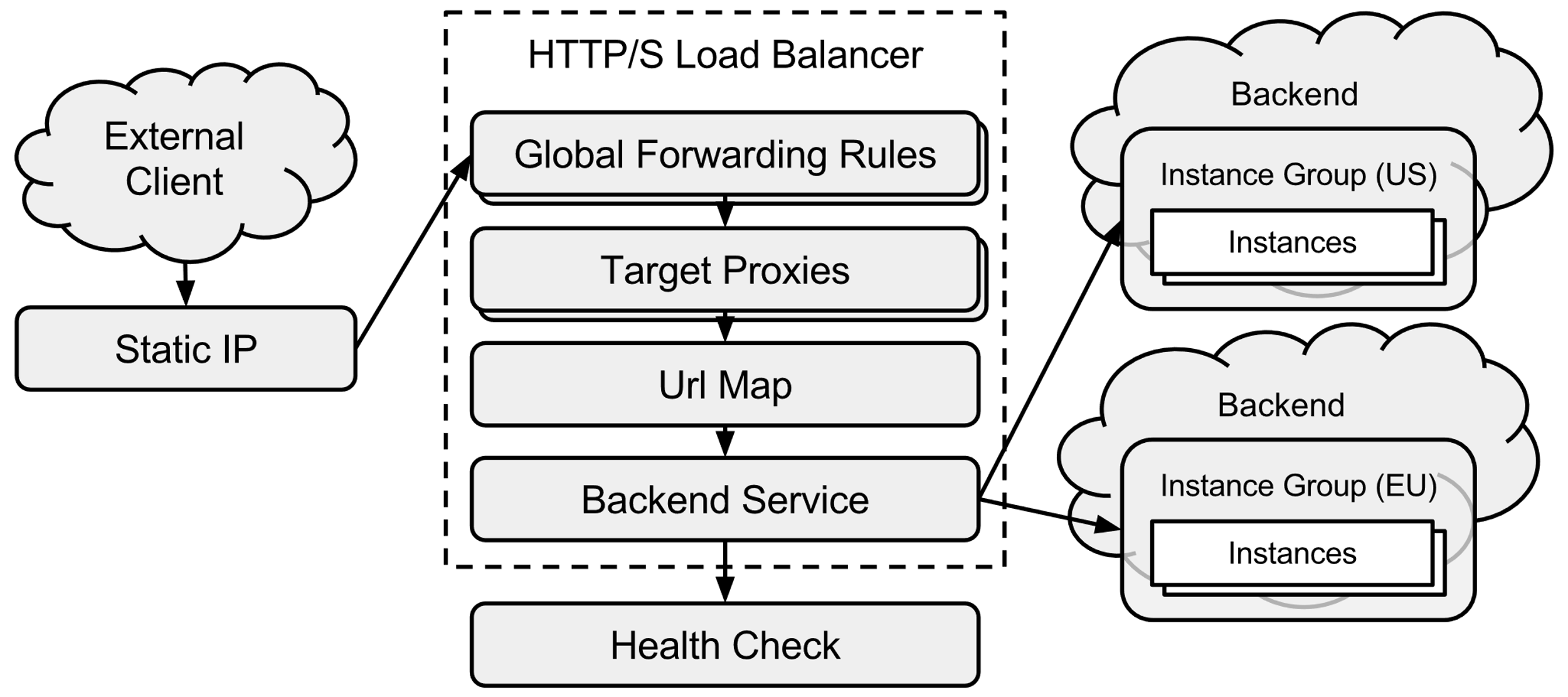
**GCP – HOL -Session 5**

**Configuring a Load Balancer for Internet Information Services**

## Scenario

You need to load balance traffic for the site www.revature.com. You want to ensure that incoming requests are routed to the closest region; however, you also want to ensure that in the event of a failure, or of instances in a region reaching capacity, the requests can fail over to a healthy instance in the next closest region.

When you finish configuring this scenario, you will have an HTTP(S) load balancer that takes requests through a single global IP address. This IP address will be able to route each incoming request by connection type—that is, HTTP or HTTPS. For HTTPS requests, the load balancer will implement SSL/TLS encryption between the client sending the request and the load balancer.

**Assumptions**

* You're using a Windows machine.
* You've created a Google Cloud Platform Console project.
* You've installed the Cloud SDK, which includes the gcloud command-line tool. You will use this tool to interact with Google Cloud Platform.
* You've run gcloud auth login to authenticate with Google Cloud Platform.
* You've installed the Chrome RDP for Google Cloud Platform
* You've set your project to be the default project that the gcloud tool will interact with. If you haven't, run the following command to do so:
* PS C:\> gcloud config set --project <project\_name>

## Step1 : Set up your backend instances

In this section, you will create two backend services in different regions. Each backend service will include two backend instances, each running a Microsoft IIS web server on Windows Server 2012. To avoid laborious manual configuration of each server, you will create a disk image from one server instance, and then use this image to create your other server instances.

### **1.1 Create your source image instance**

To create the instance that you'll use as a source image:

1. On your local Windows machine, open PowerShell.
2. Create a new Windows Server 2012 instance in the us-central1 region and add rdp-tag and www-tagtags to the instance. Later, you'll enable external access to your instance by creating firewall rules that target these tags:

gcloud compute instances create src-img ^

--zone us-central1-f --image windows-2012-r2 ^

--tags rdp-tag,www-tag

After you create your source image instance, set up firewall rules to allow external access to the instance:

1. Create a firewall rule to permit external access to port 3389 on all instances tagged rdp-tag. This rule will allow your source image instance, and any subsequent instances using the rdp-tag tag, to be accessible using RDP:
2. PS C:\> gcloud compute firewall-rules create rdp-rule ^
3. --allow tcp:3389 --source-ranges 0.0.0.0/0 ^

--target-tags rdp-tag

1. Create another firewall rule to permit external access to port 80 on all instances tagged www-tag. This rule will allow your source image instance, and any subsequent instances using the www-tag tag, to send and receive HTTP traffic:
2. PS C:> gcloud compute firewall-rules create www-rule ^
3. --allow tcp:80 --source-ranges 0.0.0.0/0 ^

--target-tags www-tag

### **1.2 Configure your source image instance**

Next, create a new Windows user on the source image instance and establish an RDP connection:

1. In your web browser, visit the [VM instances page](https://console.cloud.google.com/compute/instances) in the [GCP Console](https://console.cloud.google.com/) and click the name of your source image instance (src-img).
2. Click the **Create or reset Windows password** button.
3. In the **Set new Windows password** dialog, add your username and click **Set** to create the user account on your instance.
4. Copy the provided password and close the dialog box.
5. On the instance's console page, click **RDP**.
   * If you installed the Chrome RDP extension, the extension window will open. Confirm that you want to connect to the instance, enter your username and password, and then click **OK** to connect to your instance.
   * If you chose not to install the Chrome RDP extension, you'll be given the option to download the RDP file for your instance. Use this file to connect to the instance using Windows Remote Desktop Connection or your preferred third-party client.

After you establish an RDP connection with your source image instance, install IIS and add a default home page:

### **1.3 Verify that your source image instance is able to serve content**

### Step 2 : Create a reusable Windows Server 2012 image from your source image instance

After verifying that your source image instance is properly configured and able to serve content, create a reusable disk image from the instance's root persistent disk:

1. On your source image instance, open PowerShell as an administrator.
2. Run the following command to prepare your system for cloning:

PS C:> GCESysprep

When the GCESysprep operation completes, you will be disconnected from your RDP session automatically.

1. On your local machine, run the following to delete your source instance while retaining its root persistent disk:

PS C:> gcloud compute instances delete src-img --keep-disks boot

1. After the instance is deleted, create a new image from the root persistent disk you retained:

PS C:> gcloud compute images create win-be-img --source-disk src-img --source-disk-zone

### **2.1 Create an instance template using your source image**

Now that you've created a disk image from your configured Windows server, you can use the image as the source image for an instance template. Later, you'll configure two managed instance groups that will use this template to create new instances.

On your local machine, run the following to create an instance template, using win-be-img as your source image and rdp-tag and www-tag as your instance tags:

PS C:\> gcloud compute instance-templates create win-be-tmpl ^

--tags rdp-tag,www-tag ^

--image win-be-img

### **2.2 Create a managed instance group for each region**

Next, create managed instance groups in each region. After you create each instance group, the group will populate itself with two identical instances based on the instance template you defined earlier. Later, you will configure your load balancer to treat these instance groups as backend targets.

To create your managed instance groups:

1. On your local machine, run the following command to create a new managed instance group in the zone us-central1-f, and to populate it with two identical instances:
2. PS C:> gcloud compute instance-groups managed create us-be-group ^
3. --base-instance-name us ^
4. --size 2 ^
5. --zone us-central1-f ^

--template win-be-tmpl

1. Do the same in the zone europe-west1-d:
2. PS C:\> gcloud compute instance-groups managed create eu-be-group ^
3. --base-instance-name eu ^
4. --size 2 ^
5. --zone europe-west1-d ^

--template win-be-tmpl

### Step 3 : Verify that your backend instances are running

## 3.1 Create and configure your load balancing service

The Compute Engine load balancing service comprises several components. In this section, you will create these components and connect them together.

1. On your local machine, run the following to create a new health check. Your load balancer will use this health check to check the health of your backend instances:

PS C:\> gcloud compute http-health-checks create basic-check

1. Create a backend service:
2. PS C:\> gcloud compute backend-services create be-srv ^

--protocol HTTP --http-health-check basic-check

1. Add your instance groups as backend targets for your backend service:
2. PS C:\> gcloud beta compute backend-services add-backend be-srv ^
3. --instance-group us-be-group --zone us-central1-f
4. PS C:\> gcloud beta compute backend-services add-backend be-srv ^

--instance-group eu-be-group --zone europe-west1-d

1. Create a default URL map that directs all incoming requests to all your instances:

PS C:\> gcloud compute url-maps create lb-map --default-service be-srv

1. Create an SSL certificate resource. Your load balancer will use this resource to encrypt and decrypt traffic.

If you already have a private key and an SSL certificate from a certificate authority, you can use them to create a new SSLCertificate resource by running the command below. If not, you can create and use a self-signed certificate for testing.

Run the following command to create your SSL certificate resource. Replace <crt\_file\_path> with your certificate's local file path and <key\_file\_path> with your private key's file path.

PS C:\> gcloud beta compute ssl-certificates create www-cert ^

--certificate --private-key

1. Create target HTTP and HTTPS proxies to route requests to your URL map. The proxy is the portion of the load balancer that holds the SSL certificate for HTTPS load balancing, so you also associate your certificate with the proxy in this step:
2. PS C:> gcloud compute target-http-proxies create http-lb-proxy ^
3. --url-map lb-map
4. PS C:> gcloud beta compute target-https-proxies create https-lb-proxy ^

--url-map lb-map --ssl-certificate www-cert

1. For your load balancer to reliably receive traffic, you need to assign a global static IP address to the load balancer's global forwarding rule. To create a global static IP resource, run the following command:

PS C:> gcloud compute addresses create lb-ip --global

Take note of the IP address.

1. Create two global forwarding rules to handle incoming HTTP and HTTPS requests. Each forwarding rule will send traffic to one of the target proxies you created, depending on the IP address, IP protocol, and port specified.

Replace <lb\_ip\_addr> in the following commands with the static IP address you created in the previous step:

PS C:\> gcloud compute forwarding-rules create http-fwd-rule ^

--address <lb\_ip\_addr> --global ^

--target-http-proxy http-lb-proxy --port-range 80

PS C:\> gcloud beta compute forwarding-rules create https-fwd-rule ^

--address <lb\_ip\_addr> --global ^

--target-https-proxy https-lb-proxy --port-range 443

After you create the global forwarding rules, it can take several minutes for your configuration to propagate. To check the progress of the propagation, you can either monitor your configuration in the Google Cloud Platform Console or run the following command on your local machine:

PS C:\> gcloud compute backend-services get-health be-srv

## Step 4: Send traffic to your backends

Now that you've configured your load balancing service, you can start sending traffic to the forwarding rule and watch the traffic be dispersed to different instances.

To send traffic to your backends:

1. Open the [Load balancing page](https://console.cloud.google.com/net-services/loadbalancing/loadBalancers/list) in the GCP Console.
2. Click the name of your load balancer.
3. In the **Backend** section of the page, confirm that instances are healthy by checking the **Healthy** column. It can take a few moments for the display to indicate that the instances are healthy.
4. Once the display shows that the instances are healthy, copy the **IP:Port** from the **Frontend** section and paste that into your browser.
5. In your browser you should see your default content page displayed.

## Step 5: Restrict access to your backends

After you have verified that everything is working as intended, modify your firewall rules so HTTP(S) traffic can only come from your load balancing service:

1. On your local machine, run the following command to update your www-rule firewall rule. This restricts traffic from all IP ranges except the ranges 130.211.0.0/22 and 35.191.0.0/16, which are the HTTP(S) load balancing proxy and health check IP ranges:
2. PS C:\> gcloud compute firewall-rules update www-rule ^
3. --source-ranges 130.211.0.0/22,35.191.0.0/16 ^

--target-tags www-tag

1. In your web browser, navigate to the [VM instances page](https://console.cloud.google.com/compute/instances).
2. Click each instance to verify that the instance is now inaccessible.

## Step 6: Simulate an outage

You can simulate an outage for one or more instances in a region so that you can observe how the load will be balanced among the remaining healthy instances.

To stop an instance from receiving additional requests:

1. Establish an RDP connection to the instance.
2. On the instance, open PowerShell as an administrator.
3. Run the following command to create a new firewall rule on the instance. This command blocks the health check traffic from the health checker and prevents all new HTTP connections from the load balancer to the instance:

PS C:\> netsh advfirewall firewall add rule name="Outage Test" protocol=tcp dir=in localport=80 action=block remoteip=130.211.0.0/22,35.191.0.0/16

1. On your local machine, run the following command to verify that the instance now reports an UNHEALTHYstatus:

PS C:\> gcloud compute backend-services get-health be-srv

1. After the instance starts reporting an UNHEALTHY status, send a request to your load balancer. Only the healthy instances should respond.
2. After you've finished simulating an outage, you can restore your instance's connectivity by deleting the firewall rule. After opening PowerShell as an administrator on the unhealthy instance, run the following command to delete the rule:

PS C:\> netsh advfirewall firewall delete rule name="Outage Test"

**Congratulations you have successfully configured Load Balancing for IIS**