Metropolitan State University
ICS 432 - 01: Distributed and Cloud Computing
Fall 2021

Assignment 2: Load Balancing and Auto Scaling
Total points: 60
Out: Saturday, October 2, 2021
Due: 11:59 PM on Friday, October 22, 2021

The objective of this homework is to practice using cloud load balancing and auto scaling services by implement a scalable cloud infrastructure to serve the ICS 432 simple web site. We will implement the infrastructure two times, on AWS in Exercise 1 and on GCP in Exercise 2.

What to submit?

At various parts in this assignment, you are asked to take screenshots of your work. Open a word document and paste the screenshots in that document in the same order as mentioned in assignment and make sure to write the screenshot number. The assignment may also include some questions that requires written response so make sure to include answers to these questions in your submitted document.

In addition to taking screenshots, while working on the two exercises, fill in the following table where each row includes the following about a cloud service: (1) the purpose of the service, (2) name of that service in AWS, and (3) name of that service in GCP. Consider only services used in this assignment and include all services even if they are supported by only one cloud service provider (CSP).

Purpose	AWS	GCP
Running compute resources on the cloud	Compute →Amazon EC2	Compute→ Compute Engine → VM instances

Exercise 1: Implementing a scalable AWS infrastructure to host a web site

In this exercise, you will implement a scalable AWS infrastructure to host a web site for ICS 432. The web site html pages are given to you in the file ICS432Fall2021.zip. Make sure to use he ICS432Fall2021 folder from week 6 contents because it includes more file than what you used before in Lab 3.

You will build a scalable infrastructure to host the ICS432 web site such that the number of instances serving the web site automatically scales up and down depending on the web site traffic (i.e., when the load is high, more instances are automatically launched and when the load is light, instances are automatically shut down to save cost).

This exercise includes the following 9 steps:

1- Step 1: create a Security Group.
2- Step 2: create a web server EC2 Instance to host ICS 432 web site.
3- Step 3: create an Amazon Machine Image (AMI) using the ICS432 web server instance.
4- Step 4: create a Launch Configuration using the ICS432 AMI.
5- Step 5: create an Auto Scaling group.
6- Step 6: create a Load Balancer.
7- Step 7: Monitoring through CloudWatch.
8- Step 8: Cleaning up.
9- Step 9: Summarize your learning.

Reference

This assignment is based on materials that was covered in a Free course titled 'Cloud Computing Infrastructure' that is offered by edx.com:
https://www.edx.org/course/cloud-computing-infrastructure

Step 1: Create a Security Group

In this step, you will create a security group to be used by the instances that host the web site.

1- Create a security group with the name <your-last-name>-ics432website-securitygroup. For example, my group is called ghanem-ics432website-securitygroup.
2- Enter the following description for the group: "This security group is used by the instance group that hosts ICS 432 web site"
3- Add inbound rules to the group to allow:
   a. SSH connection.
   b. HTTP connections to allow internet traffic.

Homework report screenshot #1: take a screenshot of the security group dashboard. Make sure the group's name and inbound rules are clear in the screenshot.

Step 2: Create a web server to host ICS 432 web site

In this step, you will create an EC2 instance running a web server and copy the ICS 432 web site files to the instance. This instance will be the basic machine image that will be used by the auto scaling service to automatically launch new instances.

1- Launch an EC2 instance by following the same steps that you followed in Exercise 1 in Homework 1. Make sure of the following:
    a. Instance name: <your-last-name>-ics432website-instance.
    b. Security group: select the security group created in Step 1.
    c. Create a new key pair and download to your computer.
2- Once launched, open an SSH connection to the instance.
3- Copy the file `ICS432Fall2021.zip` to the instance.
4- In Assignment 1, Exercise 1, there are detailed steps on how to set up the ngnix web server and to serve the BallotOnline web site. In this Assignment, you are asked to follow the exact same steps to host ICS 432 website.

Homework report screenshot #2: take a screenshot of the command window after you run the final `curl` command.

Homework report screenshot #3: connect to the instance from the browser using the instance's Public IPv4. Take a screenshot of the ICS 432 web site's main page as displayed on the browser. Make sure the URL bar appears clearly in the screenshot.

Step 3: create an Amazon Machine Image (AMI).

In this step, you create an Amazon Machine Image (AMI) from the ICS 432 web server instance created in Step 2.

The steps to create an AMI are in Lab 5, Exercise 3, Task 1. Append your last name to the image name. For example, my AMI is called Ghanem-ICS432-Image.

Homework report screenshot #4: take a screenshot of the machine image dashboard that shows the list of Images including your recently created image.

Step 4: create a Launch Configuration

In this section, you will create a launch template using the AMI created in Step 3. The launch template is to be used by the auto scaling service to launch new instances as needed.

The steps to create a Launch Configuration are in Lab 5, Exercise 3, Task 3. Follow the same steps to create the launch configuration and make sure of the following:

1- Add your last name to the launch configuration name. For example, my launch configuration name is Ghanem-ics432website-launch-configuration.
2- Use the AMI created in Step 3 of this exercise.
3- Use the keypair created in Step 2 of this exercise.
4- Use the security group created in Step 1 of this exercise.
5- IMPORTANT: you will add `'user data'` to your launch template to add the instance `IP address` to an html page on the ICS 432 web site. This is useful so to check which instance is used to serve the request when a load balancer is used. This is the same as was done in Lab 5, Exercise 3, Task 4. However, in the AWS Academy lab, this code was

already included in the Web Server Instance that was automatically created by the lab environment. To add the code to the launch configuration, scroll down to the `Advanced Details` section and enter the following code the `User Data` section.

```
#! /bin/bash
vm_hostname="$(curl --write-out "\n" --request GET
"http://169.254.169.254/latest/meta-data/public-ipv4")"

echo "Page served from: $vm_hostname" | \
sudo tee /data/www/ICS432Fall2021/server-ip.html

sudo nginx
```

This code snippet has three lines of code to do the following: (1) retrieve `public-ipv4` from instance meta-data and save this to the `vm_hostname` variable, (2) write a sentence to the `server-ip.html` page with the retrieved IP, and (3) start the `nginx` server.

Homework report screenshot #5: take a screenshot to show your User Data section after you enter the code.

6- Test the launch configuration by launching TWO instances using the configuration. Connect to the instances using a browser and make sure that the ICS 432 web site is displayed.

Homework report screenshots #6(a) and #6(b): take screenshots of the browser windows connecting to the instances and make sure the URL bar appears in the screenshot.

Homework report screenshots #7(a) and #7(b): click on the server-ip link from the left menu of the main web site page and take screenshots of the browser pages that display in the IP addresses.

7- Delete the two test instances.

Step 5: create an auto Scaling group.

In this step, you will create an auto scaling group to launch and stop instances according to the web site traffic load.

The steps to create an auto scaling group are listed in Lab 5, Exercise 3, Task 3. Follow the same steps and make sure of the following:

1- Add your last name to the group name. For example, my group will be named ghanem-ics432website-auto-scaling-group.
2- Use the launch configuration created in Step 4 of this assignment.
3- Choose two subnets: us-east-1a and us-east-1c.
4- Do not add a load balancer. This will be done in the next step.
5- In Scaling policies:
    a. Desired capacity: 2

        b. Minimum capacity: 2
        c. Maximum capacity: 3 (this is the maximum capacity allowed for AWS educate accounts).
        d. Scaling policies: Target Tracking Scaling policy
            1. Average CPU utilization
            2. Targe value: 50
6- Add notification:
    a. Create Topic
    b. Send notification to: ics432website auto scaling.
    c. With these recipients: enter your email address to be notified with the group actions (e.g., scaling up with one instance).

Homework report screenshot #8: take a screenshot of EC2 dashboard and it should show two EC2 instances that are created by the auto scaling group.

Homework report screenshots #9(a) and #9(b): connect to the two instances using the browser and make sure the ICS 432 web site to displayed correctly. Take screenshots of the Server-IP pages of the two instances.

Step 6: create a load balancer

In this step, you will create a load balancer and attach it to the instance group so that the load balancer will distribute the requests to the group instances.

1- Go to Services→EC2→Load Balancing→ Load Balancer
2- Create load balancer
3- From the Classic Load Balancer section, click on Create.
    a. Name: <your-last-name>-ics432website-load-balancer
    b. Listeners: Load Balancer Protocol: HTTP, Load Balancer Port: 80
4- Select an existing security group and choose the security group created in this assignment.
5- Configure Health Check:
    a. Ping protocol: HTTP
    b. Ping port: 80
    c. Ping path: /
    d. Keep all the other values as default.
6- Do not add any instances for now because we are going to assign this load balancer to an instance group.
7- Do not add tags.
8- Review and Create.
9- Click close.
10- Go back to the auto scaling group page.
11- Scroll down to Load balancing Section and click on Edit.
12- Check the box next to Classic Load Balancer and choose your load balancer.
13- Click Update.

14- Go back to the load balancer screen in the EC2 Dashboard and look for the DNS name of the load balancer. Copy the DNS name and paste it in the browser. It may take few minutes for the load balancer to be ready to serve requests.

Homework report screenshot #10: click on instances tab and you should see two instances created and attached to the load balancer. Take a screenshot to show your load balancer dashboard with the list of instances.

15- Once ready, you can see the ICS 432 web page. Check the Server IP to know which instance is used to server your request. Keep on refreshing the browser and checking Server IP and you should see the Server IP changing. You may open the DNS name from a different browser and check the Server IP.

Homework report screenshots #11(a) and #11(b): take two screenshots to show Server IP page with two different values for IP. Make sure the URL bar appears in the screenshot. Note that, the same URL (or DNS) leads to requests being served by different servers.

16- Go to the auto scaling group and change the minimum and desired capacity to be 3.

17- Go back to the load balancer page, click on the instances tab and you should then see a third instance is added to the load balancer. This may take few minutes so keep on refreshing the page until the third instance appears.

Homework report screenshot #12: take a screenshot to show your load balancer page with three instances.

Step 7: Monitoring through CloudWatch

Create a CloudWatch dashboard to monitor your deployment:

1- Go to Services→CloudWatch

2- From the left menu, click on Add a dashboard, then click on Create dashboard.
   a. Name: <your-last-name>-ics432website-group-dashboard
   b. Click cancel on the Add to this dashboard screen.

Add Load Balancer Numeric Widgets:

1- From the dashboard, click on Add Widget.

2- Select Number then click on Next.

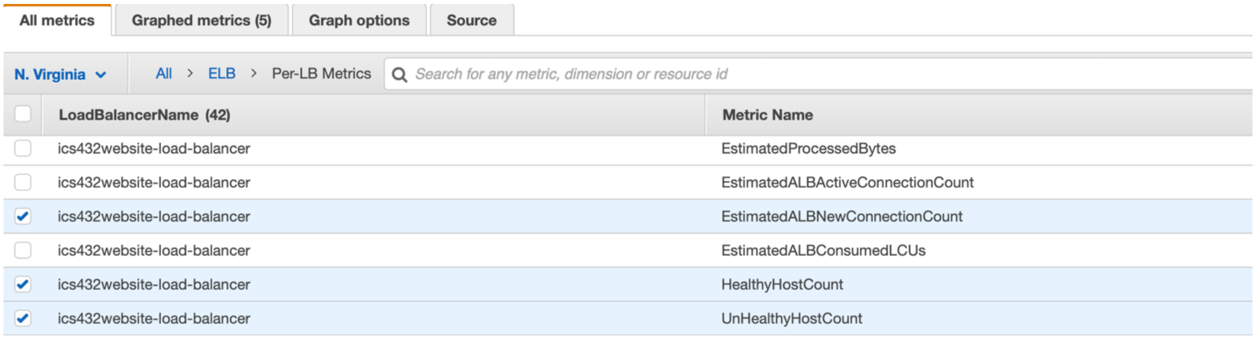
3- At the top of the screen, select 1d.

4- At the top left of the screen, click on the pencil icon next to Untitled graph and type the following: ICS431 Web Site Load Balancer Stats

5- On the Add metric graph screen, you will be adding several metrics. The first metric that you will add is for the Elastic Load Balancer.

6- Click on ELB. Click on Per-LB Metrics. Then select the check box for the following metrics. Then click on Create Widget.

Metric	Purpose
HealthyHostCount	Displays the number of healthy EC2 instances in the load balancer pool
EstimatedALBActiveConnectionCount	Displays the number of active connections to the load balancer
UnHealthyHostCount	Displays the number of unhealthy EC2 instances in the load balancer pool
HTTPCode_ELB_5XX	Displays the number of HTTP 5xx codes returned by the load balancer
HTTPCode_Backend_4XX	Displays the number of HTTP 4xx codes returned by the load balancer


All metrics
Graphed metrics (5)
Graph options
Source
N. Virginia
All > ELB > Per-LB Metrics
Search for any metric, dimension or resource id

	LoadBalancerName (42)	Metric Name
□	ics432website-load-balancer	EstimatedProcessedBytes
□	ics432website-load-balancer	EstimatedALBActiveConnectionCount
☑	ics432website-load-balancer	EstimatedALBNewConnectionCount
□	ics432website-load-balancer	EstimatedALBConsumedLCUs
☑	ics432website-load-balancer	HealthyHostCount
☑	ics432website-load-balancer	UnHealthyHostCount

Add Load Balancer Graph Widgets:

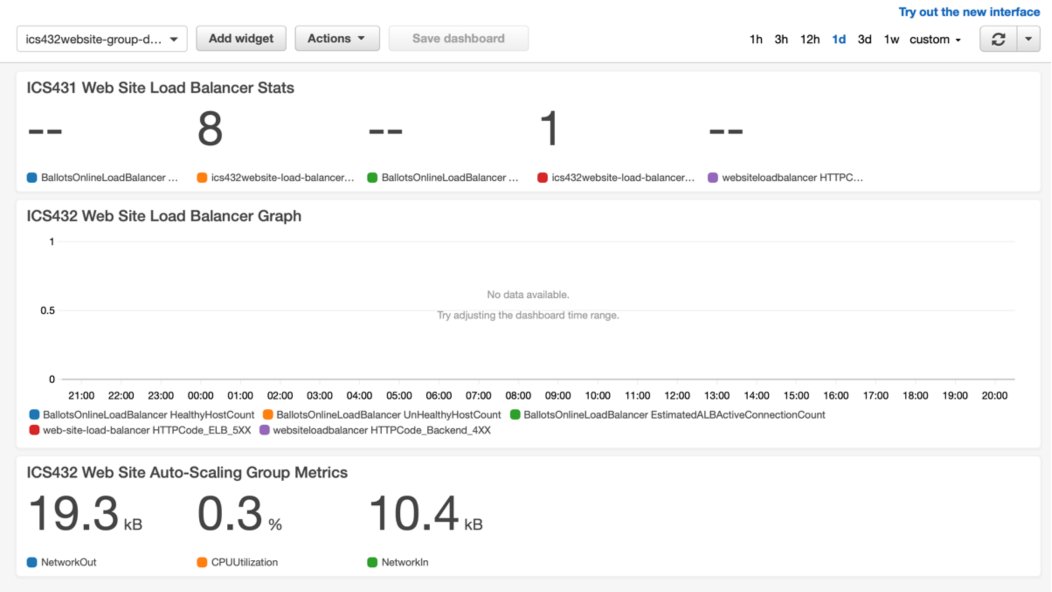
1- Click on dashboard link.
2- Click on **Add Widget**.

Metric	Purpose
NetworkOut	Displays the outbound network traffic
NetworkIn	Displays the inbound network traffic
CPUUtilization	Displays the CPU utilization for the entire auto-scaling group

3- On the **Add to this dashboard** screen, select **Line** and then click **Next**.
4- Click on Metrics then click Configure.
5- At the top of the screen, select **1d**.
6- At the top left of the screen, click on the **pencil icon** next to **Untitled graph** and type the following: <Your-last-name> ICS432 Web Site Load Balancer Graph
7- Click the **checkmark icon** to save the change.
8- On the **Add metric graph** screen, you will be adding several metrics. The next metric that you will add is for the Elastic Load Balancer.
9- Click on the **ELB** link.
10- Click on **Per-LB Metrics**.
11- select the checkbox for the following metrics:

12- Click **Create Widget**.
13- On the **Dashboard** screen, select the bottom right corner of that graph that you just created and drag to enlarge until the graph is the same size as the stats window above.

The Dashboard should look like as follows


Try out the new interface
ics432website-group-d...
Add widget
Actions
Save dashboard
1h 3h 12h 1d 3d 1w custom
ICS431 Web Site Load Balancer Stats
--
8
--
1
--
BallotsOnlineLoadBalancer ...
ics432website-load-balancer...
BallotsOnlineLoadBalancer ...
ics432website-load-balancer...
websiteloadbalancer HTTPC...
ICS432 Web Site Load Balancer Graph
1
0.5
0
21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00
BallotsOnlineLoadBalancer HealthyHostCount
BallotsOnlineLoadBalancer UnHealthyHostCount
BallotsOnlineLoadBalancer EstimatedALBActiveConnectionCount
web-site-load-balancer HTTPCode_ELB_5XX
websiteloadbalancer HTTPCode_Backend_4XX
ICS432 Web Site Auto-Scaling Group Metrics
19.3 kB
0.3 %
10.4 kB
NetworkOut
CPUUtilization
NetworkIn

No data available.
Try adjusting the dashboard time range.

Homework report screenshot #13: take a screenshot of your CloudWatch dashboard.

Add Auto Scaling Group Widgets:

Follow the same previous steps to add the following Numeric and Graph metrics for the Auto Scaling Group.

1- Numeric Widgets
2- Graph Name: <Your last name > ICS432 Web Site Auto Scaling Group Metric.
3- Click on EC2.
4- Click on By Auto Scaling Group.
5- Check the boxes next the following metrics then click Create Widget.

6- Click on Save dashboard.
7- Click on 1d on the upper right.

Homework report screenshot #14: take a screenshot of your CloudWatch dashboard.

Test scenarios:

After you created the dashboard, you will create site outages and errors and monitor the dashboard that you created.

1- Go to the load balancer link from the browser.
2- Add a non-existing page to the URL to create an error. For example: http://ics432website-load-balancer-828423979.us-east-1.elb.amazonaws.com/this-page-does-not-exist.html
3- Monitor your dashboard while you refresh your browser window for the above nonexistent webpage for 20 times.
4- On your dashboard window, click on the refresh icon. ⟳
5- You should see the following types of changes in your dashboard:
    a. EstimatedALBActiveConnectionCount should substantially increase.
    b. HTTPCode_Backend_4XX count should increase.
6- You may have to wait a few minutes before you see all changes in the dashboard.
7- You may also experiment with the time range and refresh intervals (in the upper right screen corner), or edit your dashboard (by selecting Edit on the three-dot dropdown list in the upper right corner on the dashboard) and change the Statistics column of the metrics from Average to Sum, to see accumulated changes.

Homework report screenshot #15: take a screenshot of your CloudWatch dashboard with the changed metric values.

Monitor Web Server Outages

1- Go to EC2 dashboard and stop one of the instances then go to your Dashboard, and you should see **UnHealthyHostCount** increasing.

Homework report screenshot #16: take a screenshot of your CloudWatch dashboard to show the number of unhealthy instances.

2- After few minutes, if you check the EC2 instances dashboard you will find a new instance is being created by the auto scaling group.

Homework report screenshot #17: take a screenshot of EC2 dashboard to see the list of instances.

3- Establish an SSH to each one of the instances and stop the web server using the following command to stop the web server:

```
sudo nginx -s stop
```

Homework report screenshot #18: take a screenshot of your CloudWatch dashboard to show the changes in metrics.

Step 8: Cleaning up

After I grade this assignment, you need to delete all the resources to avoid wasting your AWS Educate credits.

Go to the CloudWatch Console of your AWS Educate Starter Account.

1- Click on **Dashboards**.
2- Click on each of your <your-last-name>-ics432website-group-dashboard and click **Actions: Delete Dashboard** and confirm with **Delete Dashboard.** Repeat for both of your dashboards.
3- Go to the EC2 console by selecting **EC2** from the **Services** menu at the top of your window. Click on **Auto Scaling Groups** on the left. Select your auto scaling group and then click on **Delete.** Confirm that you wish to delete your auto-scaling group with **Yes, Delete**.
4- Click on **Instances** on the left. Click on each of your instances and then go to **Actions: Instance State: Terminate.** Click on **Load Balancers** on the left. Click on your load balancer and then go to **Actions: Delete**.

Step 9: Summarize your learning

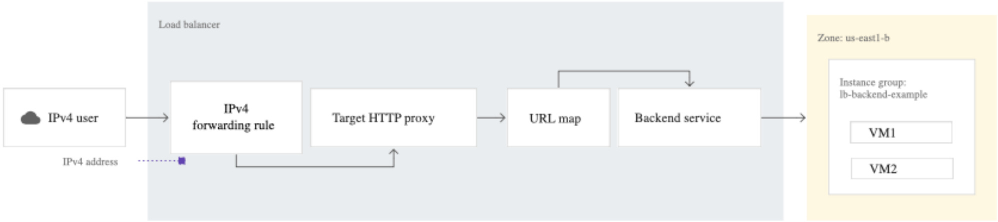
Write ~100-200 words to summarize what you learned from this exercise. You can write about what parts of the exercise were clear and what challenges did you face. You may also write about the most important things learned from this exercise.

Exercise 2: Implementing a scalable GCP infrastructure to host a web site

In the exercise, you will implement a scalable infrastructure on GCP that is similar to the AWS infrastructure created in Exercise 1.

The following diagram illustrates the sequence of steps to be followed. Basically, the explanation of the steps is below with the name of the cloud services in red. Reference: This exercise is based on Google's tutorial titled Setting up an external HTTP(S) load balancer with a Compute Engine backend that can be accessed via the following link:

```
https://cloud.google.com/load-balancing/docs/network/setting-up-network-backend-service
```



1. A client sends a content request to the external IPv4 address defined in a forwarding rule.

2. The forwarding rule directs the request to the target HTTP proxy.
3. The target proxy uses the rule in the URL map to determine that the single backend service receives all requests.
4. The load balancer determines that the backend service has only one instance group and directs the request to a virtual machine (VM) instance in that group.
5. The VM serves the content requested by the user.

You will complete this exercise in 10 steps:

1- Step 1: create a web server instance to host ICS 432 web site.
2- Step 2: create a disk image of the web server instance.
3- Step 3: create an instance template.
4- Step 4: create a managed instance group.
5- Step 5: add a named port to the instance group.
6- Step 6: creating a firewall rule.
7- Step 7: Reserving an external IP address for the load balancer
8- Step 8: Setting up a load balancer.
9- Step 9: cleaning up.
10- Step 10: Summarize your learning

Step 1: Create a web server instance

In the first step, we will create a web server instance to host the ICS 432 web site.

1- Login to your GCP Console account and make sure you are using your Metro State email address.
2- Create a new project called: <your-last-name>-homework2-loadbalancing.
3- Create a VM instance, called <your-last-name>-ics432website-instance:
   a. Choose CentOS as the operating system.
   b. Allow HTTP and HTTPs traffic.
4- Open an SSH connection to the VM.
5- Upload the ICS432Fall2021.zip file to the VM.
6- Execute the following commands (on the VM's SSH window) to set up ICS 432 web site on your VM. Enter Y when prompted.

```
sudo yum update
sudo yum install nginx
sudo mkdir -p /data/www
sudo yum install unzip
sudo unzip ICS432Fall2021.zip -d /data/www
sudo chmod -R 777 /data
sudo chmod 755 /data/www/ICS432Fall2021/index.html
sudo mv /etc/nginx/nginx.conf /etc/nginx/nginx.conf.old
sudo cp /data/www/ICS432Fall2021/nginx.conf /etc/nginx
sudo systemctl start nginx
sudo systemctl enable nginx
sudo chcon -Rt httpd_sys_content_t /data/www
```

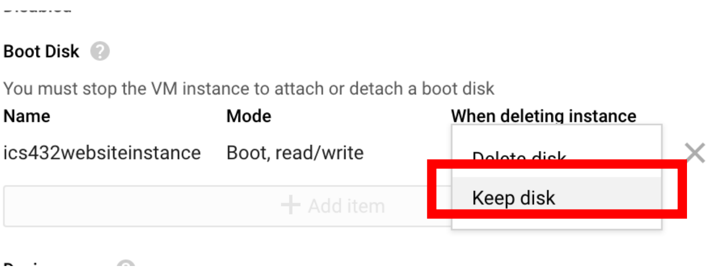
7- Test the web site is running by opening a browser window and typing the VM's public IP address. You should see the main ICS 432 web site page. If you get the error 'This site is not found', make sure that the URL is written with `http` not `https`. Test the web site by browsing the different links and making sure you can see both the server and client IP addresses when you click on Server IP link.

Homework report screenshot #19(a) and #19(b): take a screenshot of the browser window showing the main ICS 432 web site page. Make sure the URL tab appears in your screenshot. Take a screenshot to show the Server IP page.

Step 2: Create a Disk Image of the web server instance

In this step, you create a disk image of the web site instance and that image is to be used in launching additional instances as needed in the scalable infrastructure.

1- From the VM page dashboard, go to Instance Setting by clicking on the instance name then click on Edit from the top menu. Scroll down to the Boot Disk section and choose Keep Disk. We are going to use the disk of this instance to create an instance template.



Boot Disk

You must stop the VM instance to attach or detach a boot disk

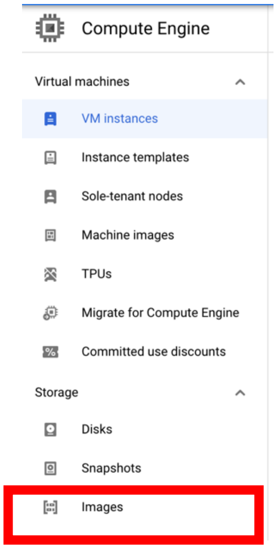
Name	Mode	When deleting instance
ics432websiteinstance	Boot, read/write	Keep disk

2- Scroll down and click on Save.

3- Stop the running instance because you are not going to use it for a while.

4- Delete the instance. Make sure to uncheck the box next to 'Delete boot disk'. Click on DELTE.

5- From the left navigation menu, choose Storage → Images. Important: This is different than Machine Image.

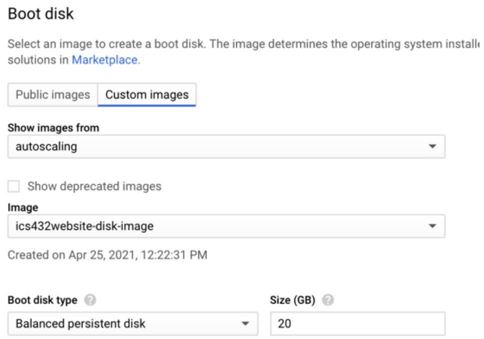


6- then click on CREATE IMAGE. Enter the following in the form:
   a. Name: <your-last-name>-ics432website-disk-image
   b. Source: Disk
   c. Source disk: <your-last-name>-ics432website-instance
   d. Location: Regional
7- Click on Create.

Homework report screenshot #20: take a screenshot of the Images dashboard showing your created image.

Step 3: Create an instance template

1- From the left navigation menu, choose Instance templates.
2- Click on CREATE INSTANCE TEMPLATE. Fill the form with the following values:
   a. Name: <your-last-name>-ics432website-instance-template.
   b. Boot disk: click on Change, then go to Custom images tab, then choose the disl image you created in the previous step (i.e., <your-last-name>-ics432website-disk-image). Click on Select.



Boot disk

Select an image to create a boot disk. The image determines the operating system installe
solutions in Marketplace.

Show images from
autoscaling

□ Show deprecated images

Image
ics432website-disk-image

Created on Apr 25, 2021, 12:22:31 PM

Boot disk type: Balanced persistent disk

Size (GB): 20

c. Check the boxes next to Allow HTTP traffic and Allow HTTPS traffic.
d. Click on Management, security, disks, networking, sole tenancy.
e. In the Networking tab, add the network tag: `allow-health-check`
f. In the Management tab, write the following code in Startup script section. This code retrieves the machine name and writes to a file in the web site (same as we did in Step 4 of Exercise 1).

```
#! /bin/bash
vm_hostname="$(curl -H "Metadata-Flavor:Google" \
http://169.254.169.254/computeMetadata/v1/instance/name)"
echo "Page served from: $vm_hostname" | \
sudo tee /data/www/ICS432Fall2021/server-ip.html
```

3- Click on Create.
4- Test the instance template by creating TWO instances based on the template and make sure both instances are successfully serving the ICS 432 web site. To create an instance based on a template, go to VM instances, click on CREATE INSTANCE, then from the left navigation menu, choose New VM instance from template.
5- When the instances are up and running, click on the External IP address to connect to the instance from the browser and make sure that the ICS432 web site is displayed. Check on Server IP and make sure it reflects the instance's External IP address. If you get the error 'This site can't be reached', make sure `http` is used in the URL and not `https`.

Homework report screenshot #21(a) and #21(b): take screenshots of the browser windows showing the main ICS 432 web site page as served by both instances. Make sure the URL tab appears in your screenshot.

Homework report screenshot #22(a) and #22(b): take a screenshot to show the Server IP pages from both instances.

6- After you make sure the instance template is working, go ahead and delete that two test instances.

Step 4: Create a managed instance group

In this step, you will use the instance template as the basis for a managed instance group. The managed instance group will automatically scale up and down based on the web site load.

1- From the left navigation menu, choose Compute Engine → Instance groups.
2- Click on CREATE INSTANCE GROUP. Set the form inputs as follows:
  a. Name: <your-last-name>-ics432website-instnace-group
  b. Location: Single zone.
  c. Region: us-sentral1 (Iowa)
  d. Zone: use-central1-a
  e. Instance template: the template you created in the previous step (i.e., <your-last-name>-ics432website-instance-template).
  f. Autoscaling mode: Autoscale

g. Minimum number of instances: 2
h. Maximum number of instances: 4
i. Cool down period: 60
j. Autoscaling policy: keep the default CPU utilization: 60%

3- Click Create.
4- Wait until the group is created, it may take few minutes.
5- Go to the VM instances dashboard and wait until the two group instances are created.

Homework report screenshot #23: take a screenshot of the VM dashboard showing the two instances.

6- Use the browser to make sure the two instances are serving the ICS 432 web site.
7- Go the Instance Group page, click Edit, and change the number of minimum instances to 3. Check that the third instance is created and is working successfully.

Homework report screenshot #24: take a screenshot of the VM dashboard showing the three instances.

Step 5: Adding a named port to the instance group

1- In the Google Cloud Console, from the left navigation menu, go to the **Instance groups** page.
2- Click the name of your instance group (i.e., <your-last-name>-ics432website-instnace-group) and click Edit from the top menu.
3- Click Specify port name mapping.
4- Click Add item.
5- For the port name, enter `http`. For the port number, enter and `80`.
6- Scroll down and Click Save.

Step 6: Configuring a firewall rule

In this step, you will create a firewall rule called `fw-allow-health-check` which is an ingress rule that allows traffic from the Google Cloud health checking systems (130.211.0.0/22 and 35.191.0.0/16). You will use the target tag allow-health-check to identify the VMs.

1- In the Google Cloud Console, from the left navigation menu, go to Networking→VPC Network→Firewall.
2- Click Create firewall rule to create the second firewall rule.
3- For Name, enter fw-allow-health-check.
4- Under Network, select Default.
5- Under Targets, select Specified target tags. Populate the Target tags field with allow-health-check.

6- Set Source filter to IP ranges.
7- Set Source IP ranges to 130.211.0.0/22 and 35.191.0.0/16.
8- Under Protocols and ports, select Specified protocols and ports.
9- Select the tcp checkbox, and then type 80 for the port numbers.
10- Click Create.

Homework report screenshot #25: take a screenshot of the firewalls rules dashboard showing your created rule.

Step 7: Reserving an external IP address for the load balancer

1- In the Google Cloud Console, from the left navigation menu, choose Networking→VPC Networks → External IP addresses page.

2- To reserve an IPv4 address, click Reserve static address.

3- For Name, enter lb-ipv4-ics432website.

4- Set Network Service Tier to Premium.

5- Set IP version to IPv4.

6- Set Type to Global.

7- Click Reserve.

Homework report screenshot #26: take a screenshot of the External IP addressed dashboard dashboard showing your created address.

Step 8: Setting up the load balancer

1- In the Google Cloud Console, from the left navigation menu, go to Networking→Network Services→Load balancing page.
2- Click Create load balancer. Use the following settings:
   a. Under HTTP(S) Load Balancing, click Start configuration.
   b. Select From Internet to my VMs, and then click Continue.
   c. For the load balancer Name, enter <your-last-name>-ics432website-load-balancer.
   d. Click Backend configuration.
   e. Under Create or select backend services & backend buckets, select Backend services > Create a backend service.
   f. Add a name for your backend: <your-last-name>-ics432website-backend-service.
   g. Under Protocol, select HTTP.
   h. For the Named Port, enter `http`.
   i. In Backends > New backend > Instance group, select the instance group you created (i.e., <your-last-name>-ics432website-instance-group).
   j. For the Port numbers, enter 80.

k. Retain the other default settings.
l. Under Health check, select Create a health check, and then add a name for your health: http-basic-check.
m. Set the protocol to HTTP.
n. Request path /, and then click Save and continue.
o. Retain the other default settings.

3- Click Create.
4- In Host and path rules, retain the default settings.
5- In Frontend configuration, use the following values:
   a. Set Protocol to HTTP.
   b. Set IP address to the address you created in the previous step (i.e., lb-ipv4-ics432website).
   c. Ensure that Port is set to 80 to allow HTTP traffic.
6- Click Done.
7- Click Review and finalize.
8- When you finish configuring the load balancer, click Create.
9- Wait for the load balancer to be created.
10- Once created, click the name of the load balancer name and note the IP:Port values listed on the load balancer details screen.
11- Wait until Health shows 3/3 (because your instance group has 3 instances) to indicate that all instances are healthy. This can take few minutes. Keep on refreshing the browser page until both instances are shown as healthy.

Homework report screenshot #27: take a screenshot of the load balancer main page that shows the three instances with healthy status.

12- Copy the load balancer IP:Port and paste it the browser to open the web site. If the web site is not opening, keep on refreshing the browser as it may take few minutes for the load balancer to start working.

Homework report screenshot #28: take a screenshot of the browser showing the main page of the ICS 432 web site. Make sure the URL bar is shown in the screenshot.

13- Click on Server IP to see the name of the instance that is used to server your request. Keep on refreshing the browser and checking the Server IP page to see how the load of serving requests is distributed among the three group instances. If you keep on getting the same machine name, try to access the web site from a different browser.

Homework report screenshot #29(a), #29(b), and #29(c): take three screenshot of the Server IP page showing three different IP values that correspond to the three instances. Make sure the URL bar is shown in your screenshots.

Step 9: Cleaning up

1- After you submit this homework, make sure to stop all the instances to avoid wasting your credits.

2- After the homework is graded, you can delete the GCP project.

Step 10: Summarize your learning

Write ~100-200 words to summarize what you learned from this exercise. You can write about what parts of the exercise were clear and what challenges did you face. You may also write about the most important things you learned from this exercise.