Lab 3: Distributed scheduling

Solutions to the distributed scheduling problem can be achieved through load sharing and load balancing. Allowing processes to get computation service on idle nodes is called load sharing. Load balancing algorithms strive to equalize the system workload among all nodes of a distributed system.

In this lab, you will create an Elastic Load Balancer on Amazon AWS which will monitor the CPU utilisation of the EC2 instance where your website is hosted. Under high load, a new instance should be created to attend the demand whilst an existing instance is removed on low load. Finally, an alert should be sent to your email every time the CPU utilisation of your servers exceeds a certain threshold.

Exercise 1. Create an EC2 Instance

1. Create an EC2 instance. Recommended: Ubuntu
2. Make sure that your VPC has an Internet Gateway associated with it (Services > Virtual Private Cloud > Internet Gateways) and the option “Change DNS hostnames” is set to enable (Services > Virtual Private Cloud > your VPC > Actions > Edit DNS hostnames).
3. Select an existing security group that allows inbound HTTP and SSH traffic.
4. Connect to your instance and install Apache Http server on Ubuntu:
    a. `sudo su -`
    b. `apt-get update`
    c. `apt-get install apache2`
    d. Type the public DNS of your instance in your browser’s URL text box. The Apache2 Ubuntu Default Page should appear.
    e. `cd /var/www/html`
    f. Create an HTML with your name on the “`index.html`” page and refresh your browser.

Exercise 2. Create ELB

> Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.

1. Services > EC2 > Load Balancers > Create Load Balancer > Classic Load Balancer
2. Enter a Load Balancer name (must only contain alphanumeric characters or hyphens)
3. Mark the option “Enable advanced VPC configuration”, and select the subnet where your instance is located.
4. On your utilized Security Group, allow all traffic (0.0.0.0/0) to outbound network and HTTP and SSH for inbound traffic.
5. On Configure Health Check Step, change the “Ping Protocol” to TCP.

6. On Add EC2 Instances step, add your instance
7. Review and create
8. Enter the DNS of your Load Balancer in the address bar of your browser.

Exercise 3. Create AMI

> An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. You must specify a source AMI when you launch an instance. You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.

1. Services > EC2
2. Select the instance, Actions > Image > Create Image
3. Enter a name for your instance, description and create

Exercise 4. Create Launch Configuration

> A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

1. EC2 > Auto Scaling > Auto Scaling group > Create Auto Scaling group
2. Create a new launch configuration, Next
3. My AMIs > select your instance
4. On Step 3 (Create Launch Configuration), enter a name for your launch configuration
5. Next and Create a launch configuration

Exercise 5. Create AutoScaling Group

> What is AWS AutoScaling Group?
>
> - Maintain application availability by scaling up/down your Amazon EC2 automatically
> - Add more instances on high load
> - Removes instances on less load
> - Run desired number of instances

1. Enter a name for the new Auto Scaling group and the subnets of your VPC
2. Advanced Details > Mark the option “Receive traffic from one or more load balancers”.
3. In Classic Load Balancers, enter your Load Balancer, Next
4. Mark the option “Use scaling policies to adjust the capacity of this group” to scale Scale between 1 and 5 instances.
    - Click on “Scale the Auto Scaling group using a step or simple scaling policies”.

    - Add a new alarm for “Increase Group Size”.
        - Unselect the option “Send a notification”
        - Set the alarm to fire up when the CPU utilisation is above 10%
        - Create Alarm
        - Take the action: Add 1 instance
    - Add a new alarm for “Decrease Group Size”.
        - Unselect the option of sending notification
        - Whenever CPU utilisation <= 5 per cent
        - Take the action: Remove 1 instance
        - Create Alarm
5. Create Auto Scaling group
6. Test setup
    - Terminate your instance
    - After five minutes (suggestion: take a short break), visualise the Activity History of your AutoScaling group and the existing instances

Exercise 6. Increasing CPU utilisation

1. Create a CloudWatch alarm on your Load Balancer. This alarm should send an email to you everytime the average “Unhealthy Hosts” is above 10 percent for at least 5 minutes.
2. Run the following command to increase the CPU load to above 10%

   ```
   yes > /dev/null &
   ```

3. AutoScaling group will add the required instances and CloudWatch will send you the alarm.

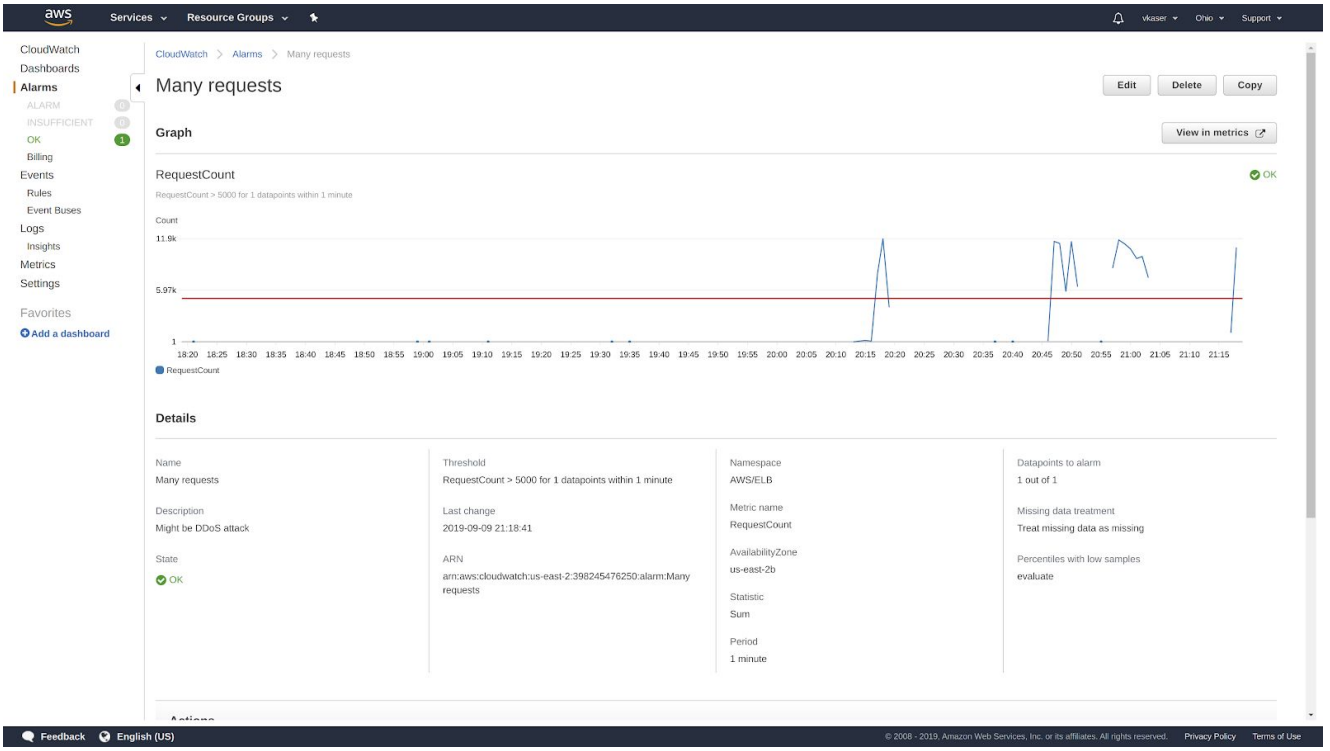
Assignment

Create protection against DDoS attacks for your infrastructure. To do this, you need to create a CloudWatch alarm and use the AutoScaling group on your Load Balancer. This alarm should send you an e-mail every time the sum of the number of requests to your web server exceeds 5000 per minute. The AutoScaling group should add a new instance when an alarm occurs and delete the instance when your state is OK. Use any load testing tool (for example, `yandex-tank`) to test your settings.

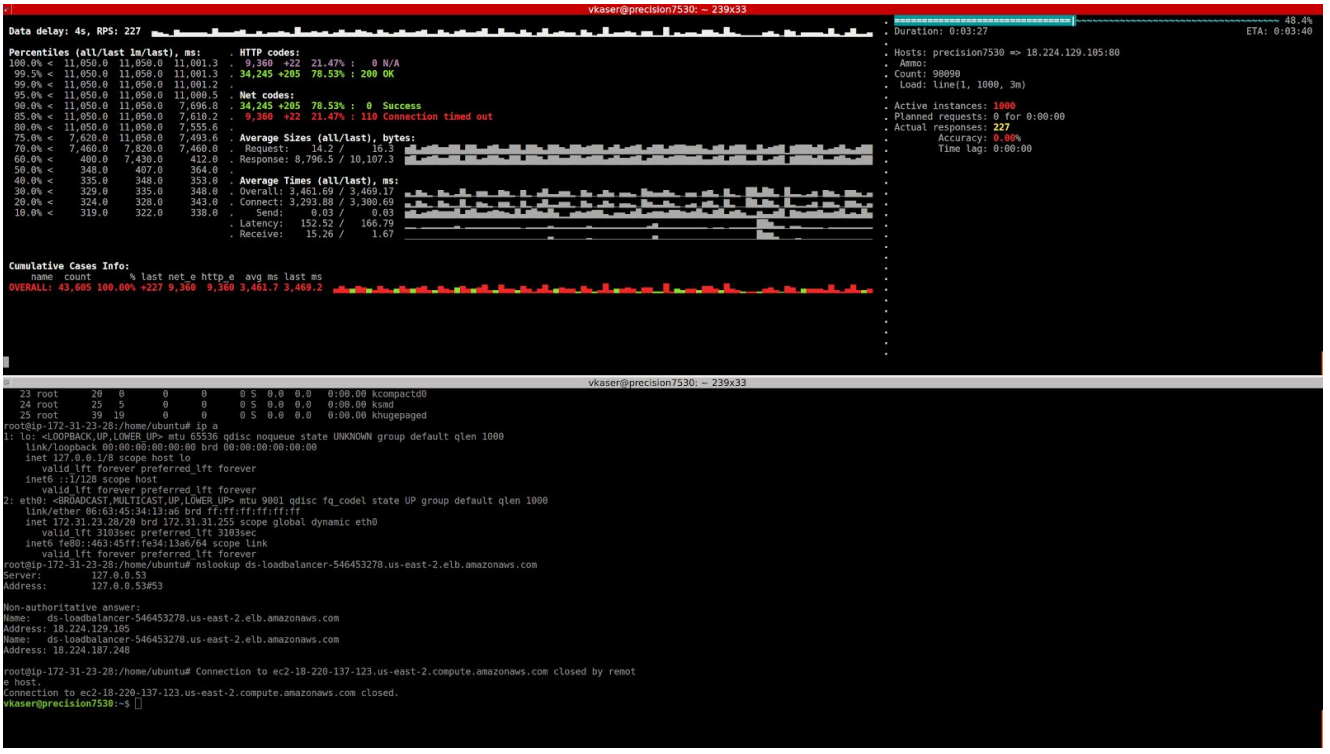
Test your load balancer public address, not the instance!

Submit the following screenshots with your results to the Moodle:

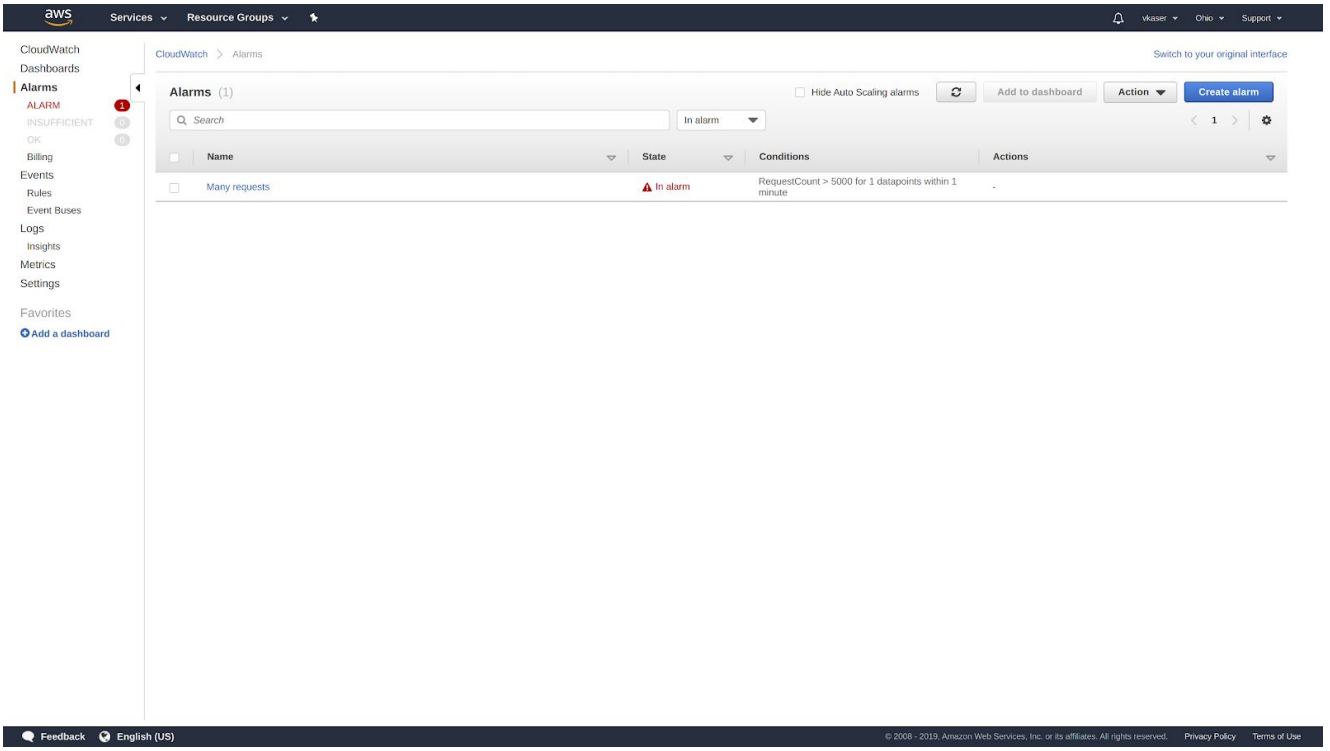
1. The CloudWatch alarm window:



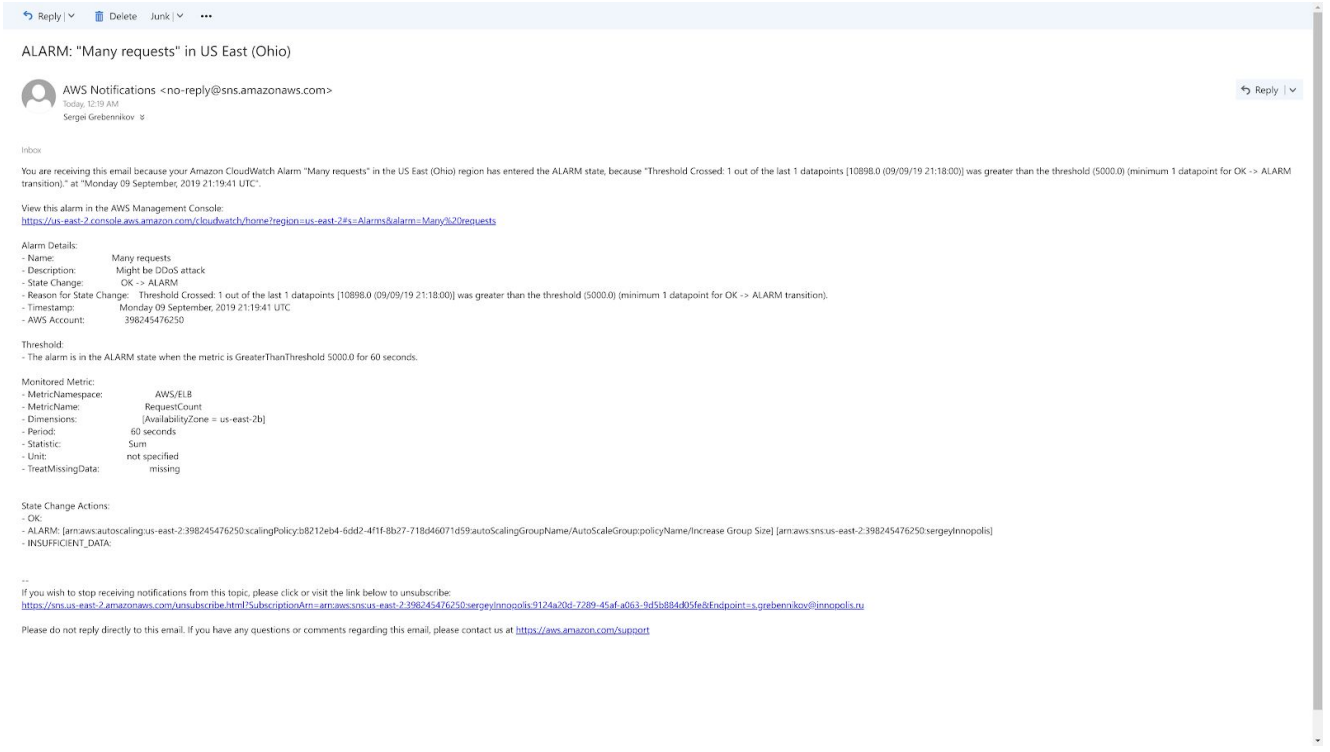
2. The work of your load testing tool:



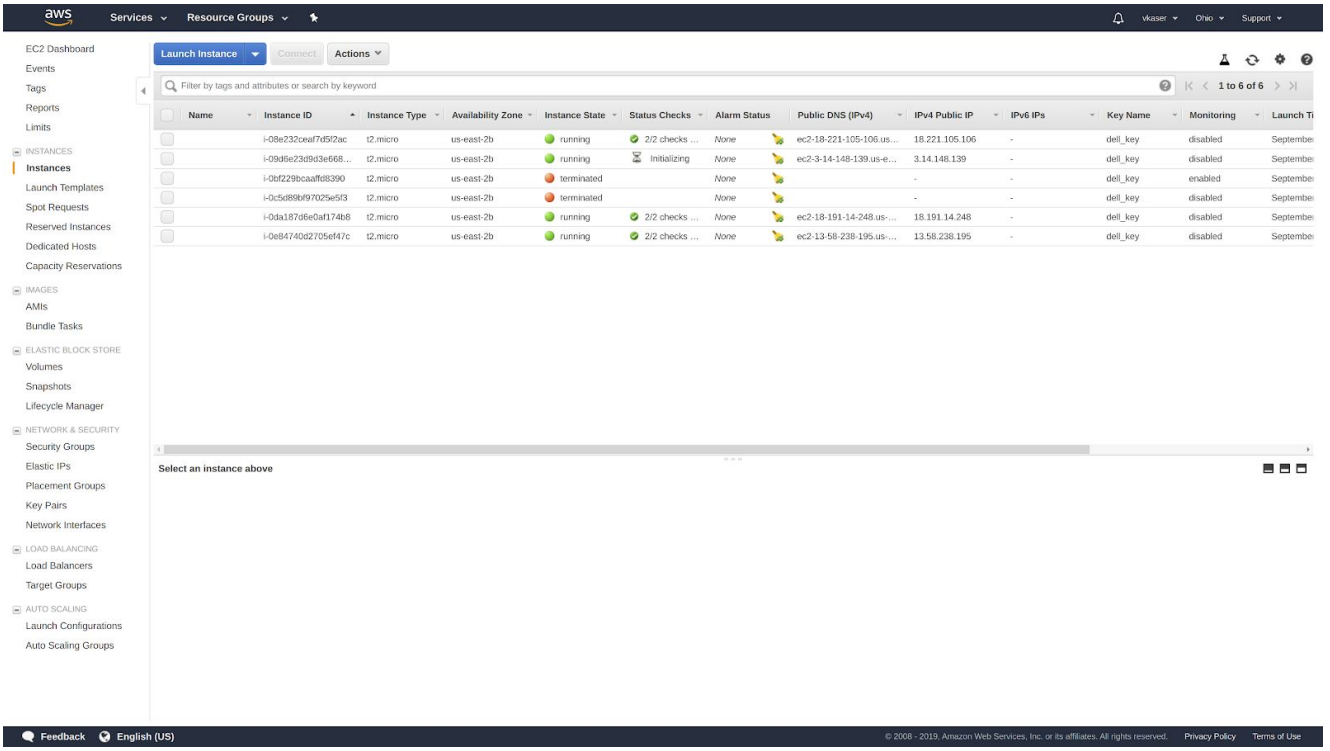
3. The alarm notification:



4. The email notification:



5. The created instances after testing:



All screenshots should be relevant to your AWS account.

Delete all the entities created in this lab (hint: delete them in reverse sequence)