Practical Exercise: Auto-scaling a simple website

In this lab we will install a web server on a "master" instance and then see how to run replicas of this server and automatically control the number of replicas based on events that occur.

A. "Master" instance and AMI (same as Part A of the load balancing exercise)

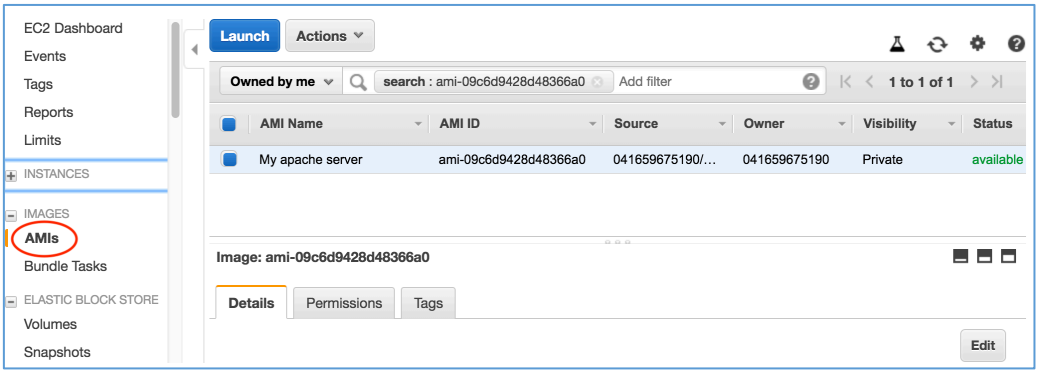
1. Launch a standard Amazon Linux 2 instance.
   - Choose t2.micro in step 1
   - Tag the instance with a suitable name in step 3 – e.g. "Master web server"
   - Ensure the security group chosen in step 6 allows inbound SSH and HTTP.
2. Connect to your EC2 instance with SSH/PuTTY.
3. Install the apache web server (httpd), enable at startup, and start service.

   ```
   sudo yum update -y                  # ensures OS patches are applied
   sudo yum install httpd -y           # install server
   sudo systemctl enable httpd         # start server at boot
   sudo service httpd start            # start server now
   ```

   You should now be able to browse to the public IP address or the public DNS name of your instance and see the apache default page.
4. Create a new default page:

   ```
   sudo nano /var/www/html/index.html      # editor to add custom content (CTRL-x to quit)
   ```

   Any content is fine here, even "Hello World". Refresh the page in your browser to check that it worked.
5. Stop the instance. When it has stopped, select **Actions** -> **Image** -> **Create Image**
6. Give the image (AMI) a name and description. It takes a couple of minutes for the image to become available. You can check the status by clicking on AMIs on the left side menu in the EC2 console.

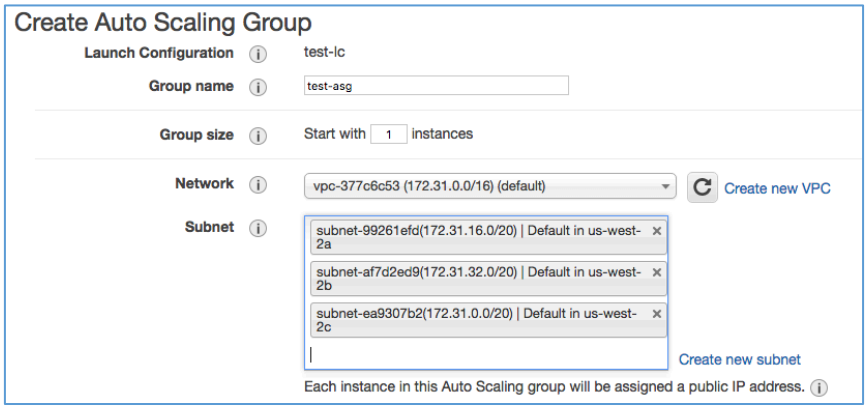


B. Auto scaling based on target tracking

The steps below are similar to those detailed at:
http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.html

7. Create a **launch configuration** based on your AMI.
   - On the EC2 console, under Auto Scaling (left menu), choose Launch Configurations.
   - Choose Create launch configuration.
   - On the Choose AMI page, select "My AMIs" and choose ***your Apache server AMI***
   - Choose t2.micro. On the next page give your launch configuration a name such as test-lc.
   - On the next page (storage), leave the storage settings at defaults.
   - On the next page (security groups), ensure the security group chosen allows inbound SSH and HTTP
8. Create an **auto scaling group** based on this launch configuration. On the Create Auto Scaling Group page, do the following:
   - For Group name, type a name for your Auto Scaling group such as test-asg.

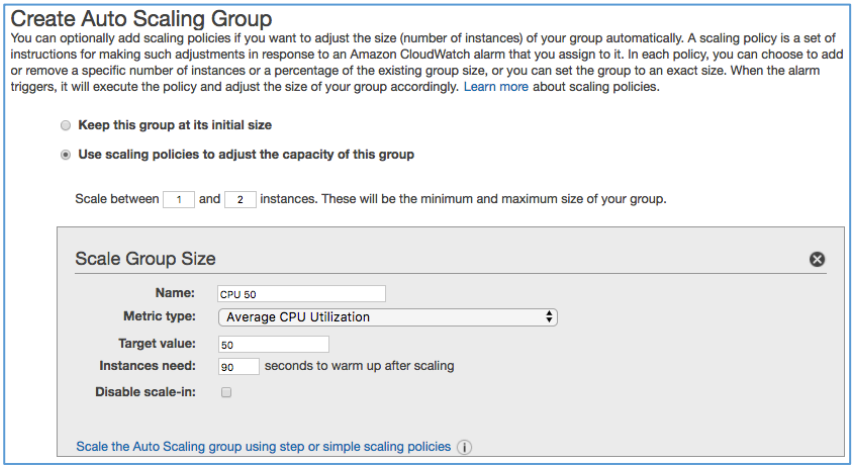
- Keep Group size set to the default value of 1 instance for this tutorial.
- Select a subnet in each Availability Zone (AZ) to add them to the Subnet list. For example, in the Ireland region, make sure you have included a subnet in each of eu-west-1a, eu-west-1b, eu-west-1c.



9. On the next page select **Use scaling policies to adjust the capacity of this group** and set it up to scale between 1 and 2 instances. Scaling policies allow you to specify how instances are started up and shut down according to changing circumstances.

10. There are a number of ways to specify scaling policies in AWS, as follows:
    - Target tracking scaling
    - Simple scaling
    - Step scaling

11. Target tracking is the default option show, so we'll do this first.

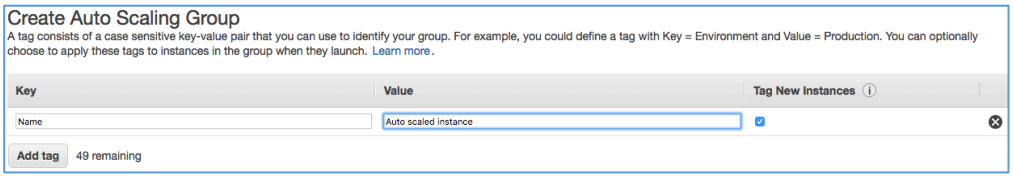


Target tracking allows you to increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that a thermostat maintains the temperature of a room – you select a temperature and the thermostat does the rest.

Choose a metric such as Average CPU Utilisation and a Target value (e.g. 50%)

12. On the next step add a notification to send yourself an email whenever a new instance is automatically launched or terminated. To do this, select "create topic". Enter any string in the "Send a notification to:" box and add your email address in the box below.

13. On the next page set up "Name" tags for your auto-scaled instances.



14. Click Review and then Create Auto Scaling Group. You should see a first auto-scaled instance starting up in the "Instances" view of EC2.

15. What happens if you manually terminate this instance?

C. Triggering an increase in group size

16. To force CPU utilisation up on your instance, log into your instance and create a simple bash script (you could call it *cpu100*) with the following or similar:
    while true; do x=0; done

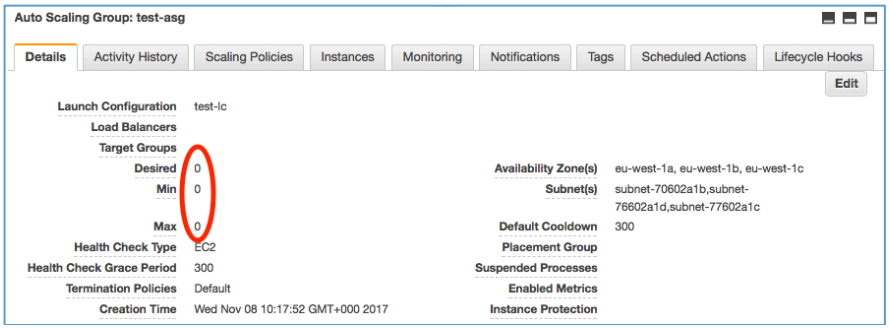
17. Make the script executable and run as a background process i.e. **./cpu100 &**

18. Use the **top** command to view process CPU utilisation. After you have received your notification email kill the cpu100 process.

19. Observe the instances in the management console. Again, try to terminate one.

D. Clean up

20. Either delete your auto scaling group or edit it to set the Desired/Min/Max number of instances to zero. This will ensure your instances are terminated and you do not use up credits unnecessarily.

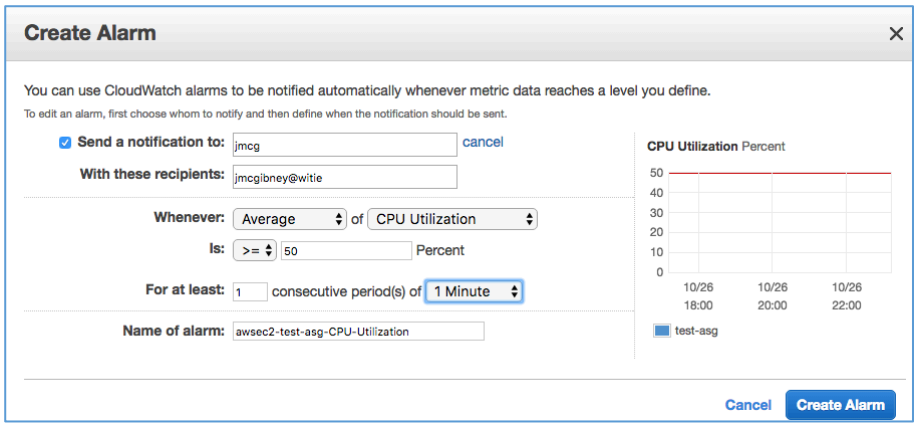


21. Note there is no need to keep the master instance that you created in Part A. This can be re-created any time from the AMI

E. Auto scaling based on CloudWatch alarms (simple scaling)

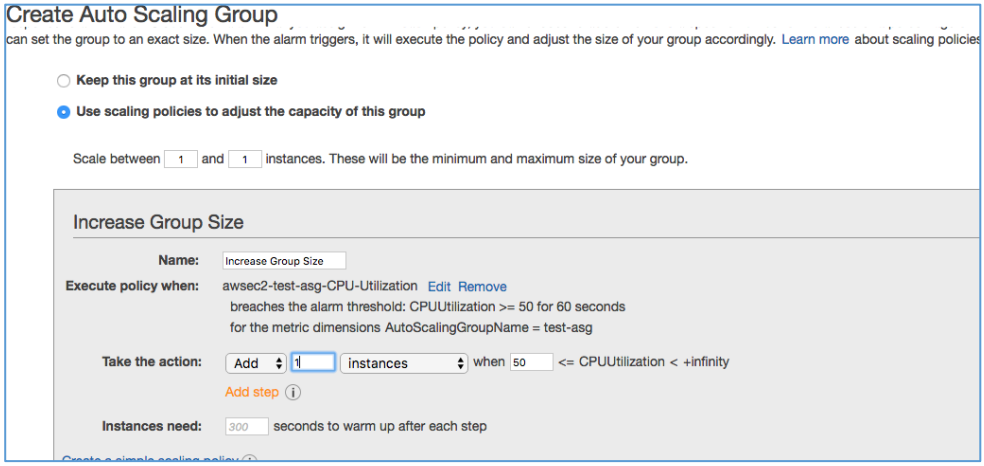
22. Create another auto scaling group, but this time when selecting **Use scaling policies to adjust the capacity of this group**, click on **Scale the Auto Scaling group using step or simple scaling policies** at the bottom.

23. For scaling-up (Increase Group Size) policy, click on **Add new alarm** and set one up to trigger on CPU Utilization exceeding 50% for 1 period of 1 minute:



24. Click "create topic" beside the "Send a notification to" box to have it send you an email when the alarm is triggered.

25. Set it up to add one additional instance when this alarm is triggered.



26. You would normally specify Decrease conditions as well, but skip this for now (click the X in the top right of the *Decrease Group Size* section). Also skip the next page (notifications).
27. Again you should see a first auto-scaled instance starting up.

F. Triggering an increase in group size

28. Enable Group Metrics collection. To do this, go to Auto Scaling Groups on the EC2 console, select your group and the **Monitoring** tab. Click on **Enable Group Metrics Collection**
29. Repeat the steps in Part C above for your new auto scaling group.

G. Clean up

30. Again. either delete your auto scaling group or edit it to set the Desired/Min/Max number of instances to zero. This will ensure your instances are terminated and you do not use up credits unnecessarily.