Lab 8  
Load Balancing / Autoscaling

CSCI 4417/5417  
Spring 2016  
Jack Ramsey, Lecturer

Purpose

Create a machine image simulating a web server, a load balancer, and an autoscaling group. Test the system on AWS

# Create Image

* 1. **tasksel**

1. Launch a new free tier Ubuntu instance. Name the instance *lastname*UbuntuASInstance. In the configuration, this time, don’t specify a private IP address. Use the security group we’ve been using this semester (SSH and HTTP have to be open)
2. When the image has launched, log in to it using PuTTY
   1. Run **sudo apt-get** update
3. Install the LAMP stack (**sudo** **tasksel install lamp-server**). Use **Passw0rd!** for the MySQL root password
4. Install siege (**sudo apt-get install siege**). This application will be used to simulate a server load
   1. VPC Add a cron job (cron is Ubuntu’s task scheduler. What we want to do is configure the server to start siege shortly after (re)booting

**sudo crontab -e**

The system will ask you which editor you prefer. As it advises you, nano is the easiest to use. Use vi or vim, if you’re brave.

1. Using the arrow key, position the cursor at the bottom of the page (the first uncommented line). Enter the following:

**@reboot sleep 60 && siege -c 250 127.0.0.1**

This will cause the siege application to run, simulating 250 simultaneous users, one minute after the instance boots (127.0.0.1, obviously, makes it hit itself)

1. Type Ctrl-x to exit, ‘y’ to save, and Enter
2. Our instance is now ready to save
3. Exit out of PuTTY
4. Return to the EC2 dashboard, right click on the instance, select Image -> Create Image
5. Save your image as *lastname*UbuntuASInstance

# Create a new subnet

1. While the above image is being created, navigate to the VPC dashboard
2. Select your public subnet and make note of which availability zone it is in (i.e., us-east-1e)

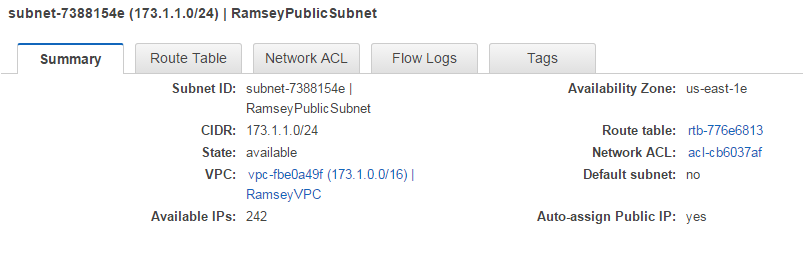
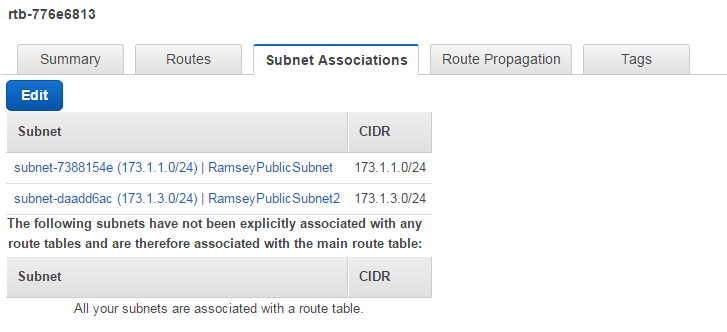


Figure 1: Checking Public Subnet AZ

1. Create a new public subnet in your VPC
2. Name your new subnet *lastname*PublicSubnet2. Make sure it is in a different availability zone than the other public subnet. Give it a CIDR block of 173.1.3.0/24
3. Once the second subnet is available, be sure to click on Subnet Actions -> Modify Auto-Assign Public IP and Enable auto-assign Public IP
4. Click on Route Tables on the left menu and select your VPC’s Route table
5. Click on Subnet Associations -> Edit, and add the new subnet to the list. Click Save



* 1. Figure 2: Adding New Subnet to Route Table

# Create a Load Balancer

1. Return to the EC2 dashboard
2. Click on Load Balancers (under Load Balancing) on the left
3. Click on Create Load Balancer
4. Name your load balancer *lastname*LB
5. In the ‘Create LB Inside:’ field, select your VPC
6. Under ‘Available Subnets,’ click on the plus signs to the left of your public subnets

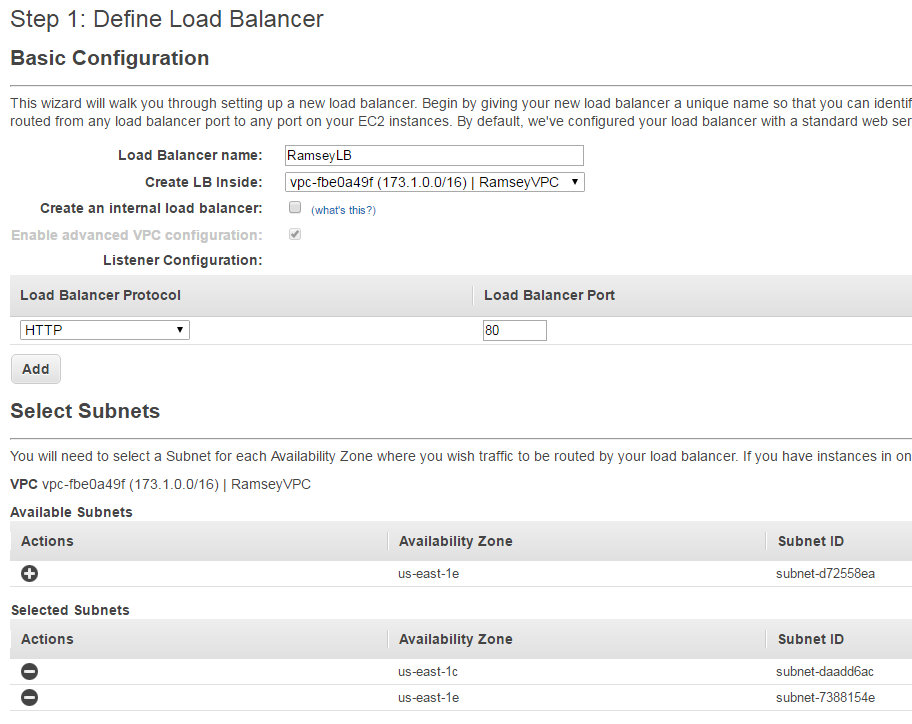


Figure 3: Configuring Load Balancer

* 1. Click ‘Next: Assign Security Groups’ in the lower right of the display

1. Select the security group we’ve been using this semester

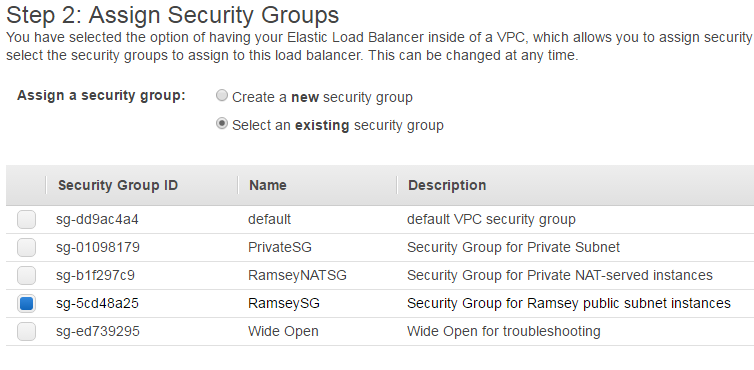


Figure 4: LB Security Group

1. Click ‘Next: Configure Security Settings’
2. Click ‘Next: Configure Health Check’
3. Click ‘Next: Add EC2 Instances’
4. Click ‘Next: Add Tags’
5. In this window (Step 6) add ‘Name’ to the Key value and ‘*lastname*LB’ to the Value field
6. Click ‘Review and Create’
7. Click ‘Create’ and ‘Close’
   1. Create Launch Configuration & Auto Scaling Group
8. NOTE: If, for some reason you can’t complete the lab today, stop here ([see the red note at the end of this document](#_Finishing_Up))
9. From the EC2 dashboard, click the Launch Configurations link under Auto Scaling on the left
10. Click ‘Create Launch Configuration’ (if ‘Create Launch Configuration’ isn’t visible, then click on ‘Create Auto Scaling Group.’ ‘Create Launch Configuration’ will then be available in the lower right corner of the display.)
11. Click ‘My AMIs’ on the left
12. Select *lastname*UbuntuASInstance
13. Ensure that t2.micro is selected and click ‘Next: Configure details’
14. Name the configuration *lastname*ASInstance
15. Click ‘Advanced Details’
16. Select ‘Assign a public IP address to every instance.’
17. Click ‘Next: Add Storage’
18. Click ‘Next: Configure Security Group’
19. Select your existing security group

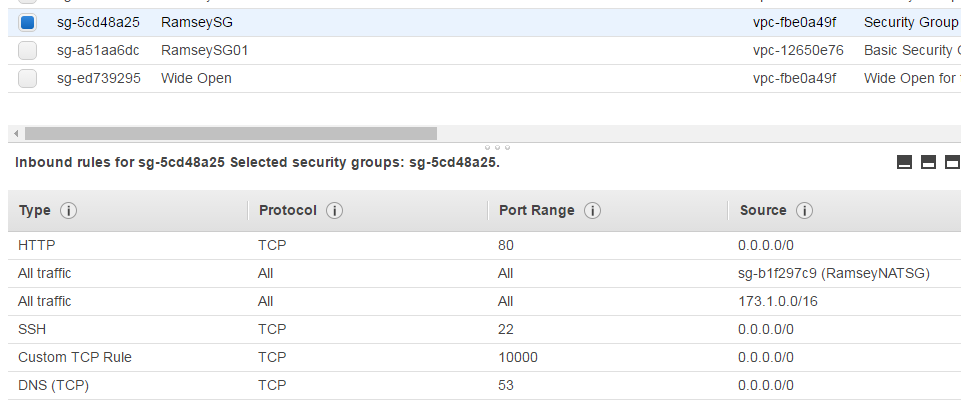


Figure 5: SG for Launch Configuration

1. Click ‘Review’
2. Click ‘Create launch configuration’
3. Select your key pair, click the acknowledgement, ‘Create launch configuration’ and ‘Close’
4. Click ‘Create Auto Scaling group’ at the top of the display
5. Make the Group name *lastname*ASInstance (also)
6. In the ‘Group size’ field, change 1 to 2
7. Select your VPC in the ‘Network’ field
8. Click on the (empty) Subnet field, select one of your public subnets
9. Repeat step 19 and add the other public subnet
10. Click ‘Advanced Details’
11. Select ‘Receive traffic from Elastic Load Balancer(s)’
12. Click in the empty field that appears, and select your *lastname*LB
13. In ‘Health Check Type,’ select ‘ELB’
14. Click ‘Next: Configure scaling policies’
15. Select ‘Use scaling policies to adjust the capacity of this group’
16. Change ‘Scale between 2 and 2 instances’ to ‘Scale between 2 and 4 instances’
17. For the following steps, refer to the screen shot below:

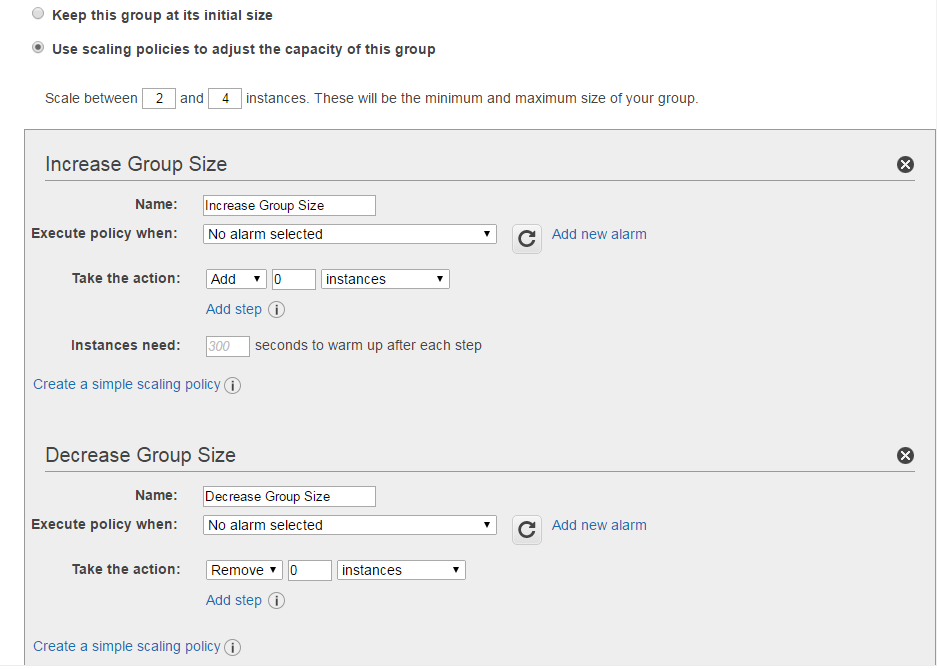


Figure 6: Configuring Conditions for Scaling

1. In the ‘Increase Group Size’ group, click on ‘Add new alarm’
2. Deselect ‘Send a notification to:’
3. In the ‘Percent’ field, enter 20 (this will invoke this rule whenever the average CPU utilization for the instance exceeds 20%)
4. Click ‘Create Alarm’
5. In the ‘Take the action:’ field, change the ‘0’ to ‘1’ (i.e., Add 1 instances when 20 <= CPUUtilization < +infinity)
6. In the ‘Decrease Group Size’ group, click ‘Add new alarm’
7. Again, deselect ‘Send a notification to:’
8. Change ‘>=’ to ‘<=’ and enter 19 in the ‘Percent’ field
9. Click ‘Create Alarm’
10. In the ‘Take the action:’ field (in the ‘Decrease Group Size’ group), change the ‘0’ to ‘1’, (i.e., Remove 1 instances when 19 >= CPUUtilization > -infinity
11. Click ‘Next: Configure Notifications’
12. Click ‘next: Configure Tags’
13. Enter ‘Name’ in the Key field and *lastname*ASGroup in the Value field
14. Click ‘Review,’ ‘Create Auto Scaling group,’ and ‘Close’

# Auto Scaling in Action

1. This is where things slow down a bit. By default, AWS updates instance statistics every five minutes. What’s happening is that the instances, one minute after they launch, will begin to experience a spike (because of the **siege** application). When the system detects CPU Utilization in each is greater than 20% (after about 5 five minutes) the Auto Scaling group will begin to launch additional instances
2. Return to the Instance listing in the EC2 dashboard
3. Your new instances should be launching (though you may need to click the Refresh button in the upper right of the display a couple of times)
4. Click on one of the two
5. Select the ‘Monitoring’ tab in the bottom pane
6. Click on the CPU Utilization chart
7. Change ‘Statistic: Average’ to ‘Maximum’
8. Since there’s a five minute delay, the chart will probably be empty
9. After about 5 minutes (and refreshing the display), you’ll see a single data point:

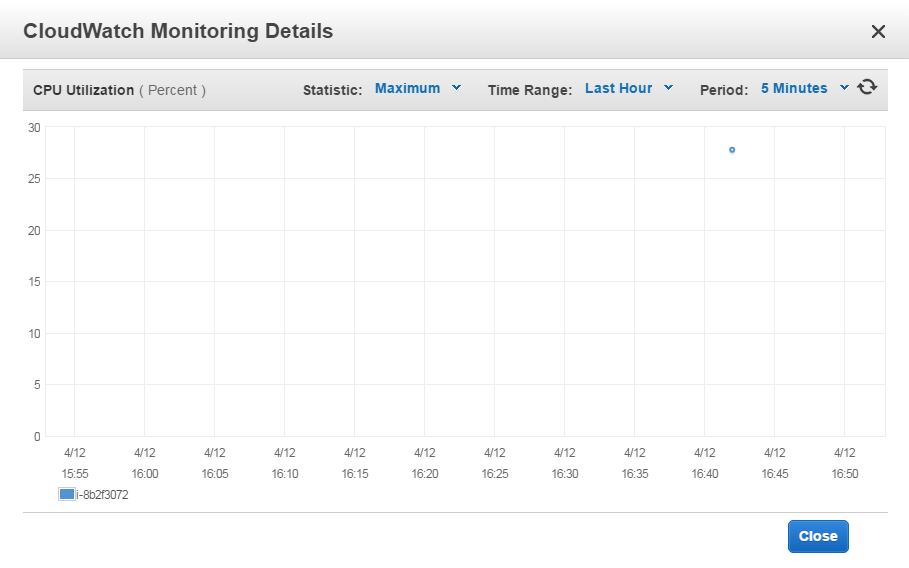


Figure 7: AWS CloudWatch Monitoring

1. Notice in the above image, the load is already in excess of 20%
2. Close the window and click on ‘Load Balancers’ on the left
3. Select your load balancer, if it isn’t already selected
4. On the Description tab, copy the DNS Name (don’t copy the ‘(A Record)’ part)
5. Open a new tab in your browser and paste the DNS name into the address bar. Hit Enter
6. You should see the default Apache server page, indicating that the load balancer is working
7. Click on the ‘Instances’ tab in the lower pane
8. Your new instances should appear (you may need to click on the ‘Refresh’ button in the upper right of the display a couple of times)
9. Using PuTTY, log in to your two instances (You can quickly launch additional PuTTY windows by right-clicking on the menu bar and selecting ‘New Session…’)
10. Return to the EC2 dashboard. You may need to refresh the display a couple of times, but soon you should see a new instance launching
11. Once the new instance is up and running, log in to it using PuTTY
12. You can wait for the fourth instance to launch, or complete the final steps with just the three
13. Select one of your active PuTTY windows (remember, if PuTTY locks up, you can right-click on the menu bar and select ‘Duplicate Session’ to log back in)
14. Enter **ps -aux | grep siege**
15. Note the PID for the siege application:

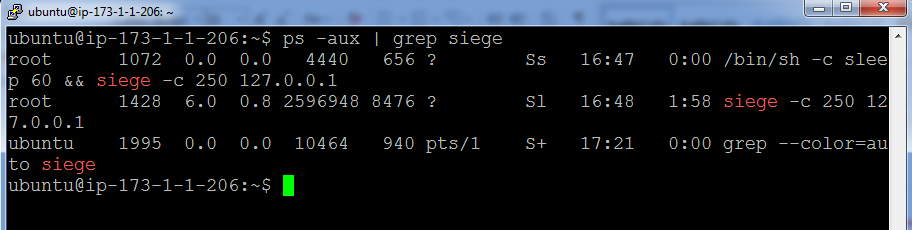


Figure 8: PID for siege

1. The first PID (above) actually indicates the cron job that launched siege. Killing it won’t kill the application, but killing the application will kill both
2. Enter **sudo kill -15 PID** (where ‘**PID’** is the process ID of siege) -- remember what the ‘-15’ means?
3. Repeat steps 20-26 for the other two (or three) instances
4. Return to your EC2 dashboard
5. Again, it will take a little while, but in five minutes or so, one of the instances will be terminated (hit the Refresh button every so often)
6. If you leave the PuTTY windows open, each time one of the instances is taken off line, it’ll ding at you
7. You can also monitor the progress by looking in the ‘Instances’ tab of your load balancer

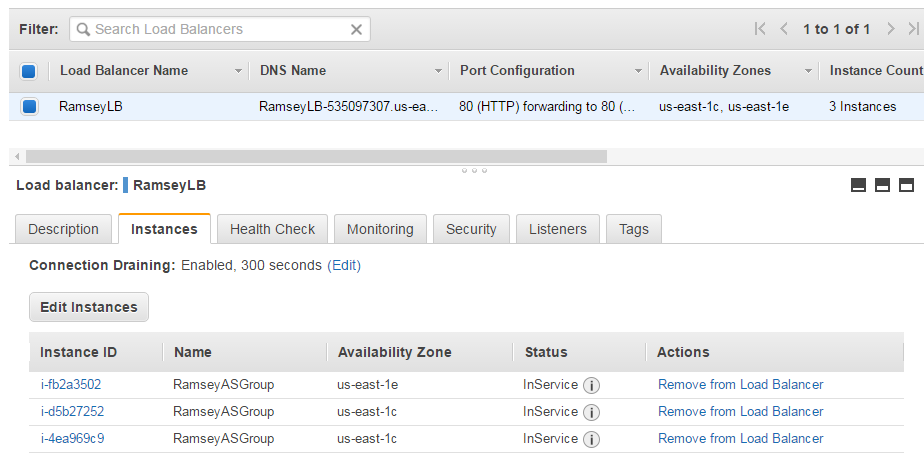


Figure 9: Load Balancer Dashboard

1. Once you’re back down to two running instances, terminate one of them and watch the Auto Scaling group replace it

# Finishing Up

1. **VERY IMPORTANT!** When you have completed the lab, be sure to delete the Load Balancer, the Auto Scaling Group, and the Launch Configurations you created. If the Auto Scaling Group isn’t deleted, it will keep launching instances even after you log off
2. Terminate any remaining running instances from the EC2 dashboard
3. Complete your report in accordance with the Lab Manual. It is due by class time next Tuesday