Per-requisites for the practical

- Sign up for a new AWS account (credit card required)
- Manual configuration steps
    - Set an alias for your AWS account
    - Enable billing alerts
    - Create an EC2 Key Pair
- What you'll achieve:
    - Deploy a VPC
    - Deploy an EC2 with CentOS Linux
    - Install a simple Apache Web server
    - Create an S3 bucket, place files in it, and set permissions
    - Create a simple Web page to link to your files' URL

LAB ONE.

Task 1: Create a VPC

Before you can deploy any AWS services, you must have a VPC to deploy them into. VPCs are
like VLANs – virtual networks for virtual devices, not unlike how a physical network would
connect and regulate access between physical devices.

1. Log into the AWS Management Console.
2. Click the "Services" menu at the top.
3. Scroll a little more than halfway down and see the "Networking & Content Delivery" section. Click "VPC."
4. On the left side navigation of the VPC Management Console, click "Your VPCs." Observe that your AWS account already includes a default VPC.
5. Go back to the "VPC Dashboard" in the top left navigation pane and Click "Launch VPC Wizard."

6. Click the second tab for “VPC with a Single Public Subnet,” then click “Select” to confirm.
7. In the VPC name field, enter “it321-lab1”.
8. Leave all other fields as they are, and click “Create VPC” in the bottom right-hand corner.
9. Click “Your VPCs” on the left navigation pane. Observe that you now have an it321-lab1 VPC in addition to the default VPC observed in Step 4.
10. Click “Subnets” on the left navigation pane. Observe that a public subnet was created and attached to the it321-lab1 VPC.
11. Click “Elastic IPs” on the left navigation pane. This is where you can assign IP addresses, accessible from the public Internet, to your VPCs.
12. Click “Allocate Elastic IP address” at the top right of the window. “Amazon’s pool of IPv4 addresses” should be pre-selected. Click “Allocate” at the bottom right of this box.
13. You will be returned to the Elastic IP details page with your new IP address already checked. Make a note of your Elastic IP’s Public IPv4 address.

Task 2: Deploy an EC2 with CentOS

In this task, we will deploy an EC2 virtual machine instance into our new VPC. This will lay the
groundwork for us to later install a Web server onto the instance, allowing us to create a simple
Web page with links to our files in S3.

1. Click the “Services” menu at the top of the AWS Management Console.
2. One of the first sections is “Compute.” Under this heading, click “EC2.”
3. Ensure that you are in the Northern Virginia Region by looking at the top right of your browser window. If the Region to the right of your name does not say “N. Virginia,” click to change it to N. Virginia. (*This should already be the case, but it is important to verify as only Northern Virginia AWS resources qualify for Free Tier usage.)
4. On the left navigation pane, click “Instances.”

5. At the top of the instance list, click “Launch Instance.”
6. The first step to launching an EC2 instance is selecting an Amazon Machine Image, or AMI. While these are often used as templates for automating the deployment of other instances, AWS also provides blank templates with the most common operating systems pre-installed, expediting your instance’s deployment.
7. In the search box labeled “Search for an AMI…,” type in “CentOS,” and hit Enter.
8. It is normal to get a “no results” message, since CentOS is not in the quick start catalog. That’s okay, Observe that the “AWS Marketplace” tab to the left shows 300+ results, Click this tab.
9. The first result should be “CentOS 7 (x86_64) – with Updates HVM.” Click the “Select” button to the right of it.
10. A pricing window will appear, but these costs can be ignored – observe the “free tier eligible” line that appears just below the CentOS logo. Click Continue in the bottom right of the window.
11. Make sure the box to the left of “General purpose – t2.micro (free tier eligible) is ticked.
12. This is the only type of instance that AWS will allow us to deploy for free with our Free Tier account.
13. Click “Next: Configure Instance Details” at the bottom right of your screen. Leave this page as it is.
14. Click “Next: Add Storage” at the bottom right of your screen. Observe that an 8 GB EBS volume will be automatically provisioned and attached to your EC2 instance so that CentOS can be installed and run properly. Leave this page as it is.
15. Click “Review and Launch” at the bottom right of your screen. Review the information on this final page and ensure that it is accurate, then click “Launch” at the bottom right of your screen.
16. A window will appear talking about key pairs. For security reasons, AWS does not deploy instances with root passwords; you must generate a pair of public and private keys. The server will hold the public key, and you will hold the private key as a means of authentication, instead of a root password.

17. In the first dropdown, select “Create a new key pair.”
18. In the “Key pair name” text box, enter “it321-lab1-key”.
19. Click “Download Key Pair.” The private key will be downloaded to your computer as a .pem file.
20. Click “Launch Instances” and wait for the successful notification.
21. Scroll to the bottom of the following screen and click “View Instances” to be returned to the instances list.
22. Wait for your new instance to enter a “running” state with a green light. Click the refresh arrow icon at the top right of the instances list to re-check status.
23. Navigate back to the VPC Console by clicking the “Services” menu at the top left of your screen, and clicking “VPC.” (It should now be found under both the “Networking & Content Delivery” menus as well as the History sidebar on the left.)
24. Click “Elastic IPs” on the left navigation pane. Check the box to the left of your Elastic IP address, click the “Actions” menu at the top right, and select “Associate Elastic IP address.”
25. In the “Instance” field on the next form, click inside the box. A dropdown list of your running EC2 instances should appear, and I assume you only have one. Select it, and click the orange “Associate” button at the bottom right.
26. You will be returned to the Elastic IP details page, with your Elastic IP already selected.
27. In the details pane at the bottom, make a note of your “Public DNS” domain name for this Elastic IP.

Task 3: Create an S3 Bucket and Upload Files

In this task, we will create an Amazon S3 storage bucket that will contain some files that we will
link to from our new EC2 web server

1. Click the “Services” menu at the top of the AWS Management Console.
2. Under the “Storage” section, click “S3.”
3. Click the blue “Create bucket” button.

4. In the “Bucket name” field, type “it321-lab1-storage-YOUR-NAME,” without the quotes or comma, and replace YOUR-NAME with your first and last name in lower case. For example, it321-lab1-storage-jay-staks.
5. Make sure the “Region” dropdown is set to “US East (N. Virginia).” At the bottom right of this form, click the “Next” button.
6. Leave the next page (Configure Options) as it is, and click “Next” again.
7. On the Set Permissions page, uncheck “Block all public access.” This will cause an alert box to appear, asking you to acknowledge that this will make your S3 files public. Check the box next to “I acknowledge that the current settings may result in this bucket and the objects within becoming public,” and click “Next” at the bottom right.
8. Validate that your settings are correct, and then click the blue “Create bucket” button in the bottom right of the form and observe that your new S3 bucket has been created.
9. Click on your bucket name (it321-lab1-storage). Click the “Upload” button at the top of this page.
10. In the file uploader, click “Add files.” Find at least three (3) files on your computer that you don’t mind being publicly shared – these can be absolutely anything, a picture of a family pet, a funny meme, the PDF from this class’s syllabus – anything, just choose 3 files that do not contain your personal information. Click the “Next” button at the bottom right of this form.
11. Under “Manage public permissions,” change this dropdown to “Grant public read access to this object(s).” Click “Next.”
12. Leave the storage class as “Standard” and click the “Next” button again.
13. Click the “Upload” button in the bottom right.
14. For each of your three (3) uploaded files, click the file name in the S3 console, and make a note of the “Object URL.” Copy each of these URLs down here – you will need them later. This is also a good opportunity to load the URLs in your browser and ensure that the file is accessible from S3, and that there are no permissions issues.

Task 4: Install Apache Web Server and Make It Accessible

In these steps, we will tie together everything you've just configured to make your first functional Web site in the cloud, which will serve up the files you've just uploaded to S3! For this task, you will need an application that can use SSH to connect to your EC2 instance and configure it. This looks a bit different on Windows and Mac, so we will briefly outline the steps on how to do this on either operating system

1. Install or launch the appropriate tool to SSH into your EC2 instance.

   On Windows, you can use PuTTY:

   https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

   (You will most likely want the 64-bit MSI installer)

   On Mac or Linux systems, you can simply open a Terminal window (on Mac, the Terminal can be found by opening Finder, going to "Applications," and then the "Utilities" folder).
2. Connect to your EC2 instance.
   - Windows
     a. Take your Elastic IP address from earlier, and place it in the "Host name (or IP address)" field. Leave the port as "22."
     b. You will also need to open PuTTYgen, which was installed on your system at the same time as PuTTY.
     c. Follow these instructions in AWS's documentation to convert the .pem file you saved when creating your key pair into a .ppk file.
     d. On PuTTY's left navigation pane, under "Connection," expand "SSH," and then click "Auth." Click "Browse," select the .ppk file that was generated from PuTTYgen, and click "Open."
     e. Click "Open" again on the main PuTTY window to connect. You will be asked to confirm if you want to connect to this new system – select "Yes.
     f. You will be asked for a username. Type in "centos" and hit enter.
   - Mac
     a. Your Terminal screen should have you placed in your Home directory, with a prompt such as "your-computer-name:~

yourname$.” First, we must change to the directory where you saved your .pem file – most likely your Downloads folder, but replace that path below if you saved it elsewhere: cd /Downloads/

b. Type the below command to connect to your instance, replacing 8.8.8.8 with the Elastic IP address you created earlier.
c. ssh centos@8.8.8.8 -i it321-lab1-key.pem
d. You will be asked to confirm if you want to connect to this new system – type “yes” and hit your Enter/Return key.
e. Depending on your computer, you may get an error that says your private
   key file has “bad permissions.” This illustrates how critically important security is to AWS. We can fix this by running the following command to make our private key inaccessible to other users on your own computer:
   - chmod 0600 it321-lab1-key.pem
f. Once that has been fixed, start from step “ii” again.
g. You should now be logged into your system with a default command line prompt. Type “whoami” and hit Enter/Return. You should see a result that your own session is logged in (remember, your username is “centos.” This is because AWS does not allow “root” accounts.)
h. Run the below command to become a superuser on the system.
   - sudo su
i. Run the below command to install Apache, one of the world’s most common, free and open source Web servers:
   - yum -y install httpd
j. Note: “httpd” simply means “HTTP Daemon.” You may recall that all Website URLs that begin with http://, the Hypertext Transfer Protocol that Apache serves.

k. Once this process completes, run the two below commands to start Apache and enable it to be started automatically whenever your EC2 is rebooted:
    - systemctl start httpd
    - systemctl enable httpd
l. Back in the AWS Console, go to “Services” at the top left, and navigate to “EC2.”
m. Click on the “Instances” link on the left navigation pane.
n. Click on the row containing your EC2 instance. In the details pane at the bottom half of the window, scroll down and look at the assigned “Security groups” in the right-hand column. It may be named something like “CentOS 7 x86_64…” – click this name.
o. You will be taken to the Security Groups section of the EC2 Dashboard, with the relevant Security Group ID already checked. In the details pane at the bottom half of the window, click on the “Inbound rules” tab.
p. Observe that there are no inbound rules, which means no outside traffic is allowed to access our EC2 instance right now. Click the “Edit inbound rules” button at the top right of the Inbound Rules table.
q. For the first rule, select the “Custom TCP” dropdown under “Type” and type “HTTP” in next to the magnifying glass icon. Click “HTTP.” The wizard will automatically populate TCP port 80. For “Source,” click in the dropdown and select “Anywhere.”
r. Click “Add rule” to add a second rule. AWS allows our SSH connection just long enough to get things set up, but now that it knows that we are configuring our Security Group, our SSH connection will be terminated if we do not also allow traffic on port 22.

s. Select the “Custom TCP” dropdown and, this time, search for “SSH.” Click “SSH” when it appears. Observe that TCP port 22 is populated. Again, change the “Source” dropdown to “Anywhere.”
t. Click the “Save rules” button at the bottom right of this window.
u. Open a new tab in your browser, and enter in the Public DNS domain name that you captured after associating your Elastic IP address to your EC2 instance. Hit your Enter key to load this domain name as if it were any other Web site

Task 5: Making Your Web Page With Links

In these last steps, you should have created a functioning Web server using EC2, VPC and the

relevant Security Groups. Now it’s time to replace that default Web page with one of our own,

which will link to our S3 Objects.

1. Go back to your command-line interface that is connected to your EC2 instance. If you were disconnected or took a break, refer back to step # 2 above to re-connect. Make sure you are still a superuser (root); if not, check step # h above.
2. Change to the /var/www/html/ directory by running:
    - cd /var/www/html/
3. We will use the “vi” text editor tool to create an “index.html” file, the root of any Website. Run the below command:
    - nano index.html
4. Copy and paste the below HTML code into your terminal. Replace items in {{ curly
5. brackets }} with values that relate to you. (It may be helpful to make those changes here first, or in a separate text editor, before making the final paste into your terminal window.)
6. Note: If you have background in HTML/Web design, feel free to make your own Web page, as long as it links to your three (3) S3 objects.

HTML CodeSnippet

```
<html>
<head>
    <title>{{ Your Name }}'s First AWS Web Site</title>
</head>
<body>
    <h1>{{ Your Name }}'s First AWS Web Site</h1>
    <p>Here are the links to my S3 Objects:</p>
    <ul>
        <li><a href="{{ First S3 Object URL }}">{{ Description of first file }}</a></li>
        <li><a href="{{ Second S3 Object URL }}">{{ Description of second file }}</a></li>
        <li><a href="{{ Third S3 Object URL }}">{{ Description of third file }}</a></li>
    </ul>
</body>
</html>
```

- You will be returned back to your command line prompt. Type exit and hit Enter, and repeat this step again, to both exit your superuser role, and exit the terminal altogether, disconnecting from your EC2 instance.
- Refresh the test page in your browser or re-enter your Public DNS URL again. You should now see your Web page in your browser!
- Confirm that all links to your S3 Objects are functional.

CONGRATULATIONS YOU DID IT!

LAB TWO.

What you'll achieve:

- Create an Amazon Machine Image (AMI) from the EC2 instance you deployed in Lab #1
- Create a Launch Configuration from this AMI

- Configure an Auto Scaling Group from your Launch Configuration, to automatically deploy additional instances when the CloudWatch Alarm is triggered
- Create said CloudWatch Alarms that monitor instance CPU utilization
- Use a stress-testing tool to create fake load and force an Auto Scaling event
- Terminate resources from both Lab #1 and #2 to prevent your credit card from being charged

Task 1: Create an AMI and Launch Configuration

Auto Scaling Groups require Launch Configurations to know what type of EC2 instances to
automatically deploy in response to a CloudWatch alarm. A Launch Configuration is based on an Amazon Machine Image, or AMI, which is effectively a Snapshot of a pre-configured instance that can be repeatedly deployed for consistency.
In this task, we will create an AMI based on your currently running EC2 instance deployed
during Lab #1

1. Click the “Services” menu at the top of the AWS Management Console.
2. Immediately under the “Compute” section, select “EC2.”
3. On the left sidebar, under the “Instances” heading, click “Instances.”
4. Right-click on the row containing details about your EC2 instance from Lab #1. NOTE: If you deployed multiple EC2s during Lab #1 while attempting to get the Lab to work, make sure you are selecting the instance that is fully completed and that was submitted for Lab #1. You can validate this by checking that its Public DNS address matches the one you noted down.
5. In the menu, navigate to Image > “Create Image.”
6. In the Image name field, type in “My First AMI” (without quotes).
7. Leave all other options as they are, and click the blue “Create Image” button in the bottom-right of the pop-up.
8. A confirmation window will appear to inform you that your “Create Image request was received.” Click the blue Close button at the bottom-right.

9. On the left sidebar, under the “Images” heading, click “AMIs.”
10. You should see a row entitled “My First AMI.” It’s status will likely be orange and say “pending.” Periodically click the refresh icon at the top right of the Dashboard until the status turns green and says “available.” This may take anywhere from 5 to 10 minutes.
11. On the left sidebar, scroll all the way to the bottom. Under the “Auto Scaling” heading, click “Launch Configurations.”
12. At the top of the Dashboard, click the blue “Create launch configuration” button.
13. On the left side of the wizard, click on the “My AMIs” tab.
14. You should see your “My First AMI.” Click the blue “Select” button on the right side of your screen.
15. On the next page, make sure that the row for a General purpose – t2.micro instance is selected. A green notification below t2.micro should inform you that this instance type is “Free tier eligible.” Click on the gray “Next: Configure details” button at the bottom right of your screen.
16. In the Name field, type “My First Launch Configuration.” Click the gray “Next: Add Storage” button at the bottom right of your screen.
17. Do not change anything on this page. Click the gray “Next: Configure Security Group” button at the bottom right of your screen.
18. AMIs do not store Security Group settings, so we must re-add the HTTP rules we added from Lab #1 to the Launch Configuration Security Group. Ensure that the button to the left of “Create a new security group” is selected.
19. Change the Security group name field to “My Auto Scaling Security Group.”
20. Click the gray “Add Rule” button along the left side of this screen.
21. A new row will appear. On this row, change the dropdown from “Custom TCP Rule” to
22. “HTTP.” Verify that the “Source” dropdown is set to “Anywhere.”
23. Ignore the warning that is displayed. Click the blue “Review” button at the bottom right of your screen.
24. Click the blue “Create launch configuration” button at the bottom right of your screen.

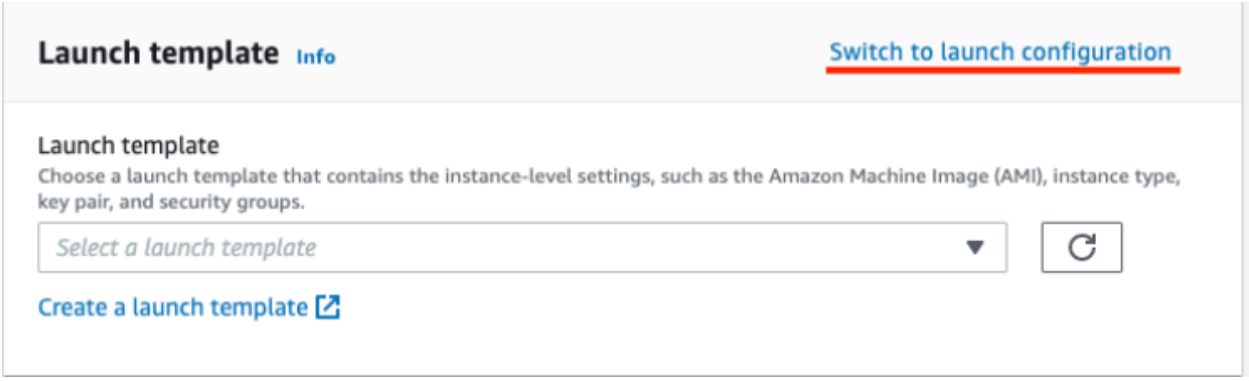
25. You will be prompted to select a key pair for the instances that are deployed as part of this Launch Configuration. Verify that the first dropdown is set to “Choose an existing key pair,” and under “Select a key pair,” make sure you select the “it321-lab1-key,” or whichever key you created that helped you successfully complete Lab #1.
    NOTE: You may have multiple key pairs if you made several attempts at completing Lab #1. You can use any one of these, as long as you have the .pem (Mac) or .ppk (Windows/PuTTY) file for this key still saved on your system! If you no longer have your .pem/.ppk file for a key pair, you must change the first dropdown box to “Create a new key pair,” name it something like “it321-lab2-key,” and “Download Key Pair.” If using Windows/PuTTY, refer back to the AWS instructions on creating a .ppk file from the downloaded .pem file so that you will be able to access your new instances.
26. Check the box associated with the “I acknowledge…” statement, and click the blue “Create launch configuration” button at the bottom-right of the pop-up.
27. After the Launch Configuration has finished creating, click the blue “Close” button on the right-hand side of your screen.

Task 2: Create an Auto Scaling Group

Now that we have a Launch Configuration, it’s time to configure an Auto Scaling Group with
defined metrics for when the configuration should be utilized.

- While still in the EC2 area of the AWS Management Console, on the left sidebar, scroll all the way to the bottom, and under the “Auto Scaling” heading, click “Auto Scaling Groups.”
- You may be taken to the old Auto Scaling Dashboard by default. If you have a blue bubble at the top of the page that says “Try the new design for Amazon EC2 Auto Scaling,” click the link in it for “Go to the new console.”
- Once in the new Auto Scaling console, click the orange “Create Auto Scaling group” button on the landing page.
- In the “Auto Scaling group name” field, type “My First Auto Scaling Group.”

- The box below this may say “Launch template.” If so, make sure to click the “Switch to launch configuration” link in the top right of the box.



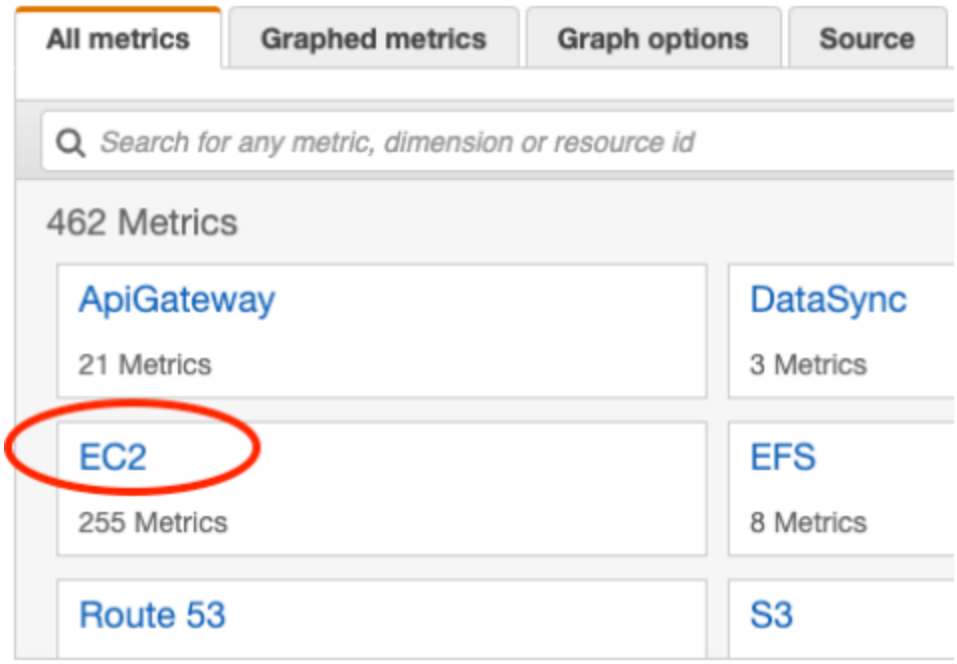
- Click inside the “Select a launch configuration” dropdown, and click on the row containing your “My First Launch Configuration” item.
- Click the orange “Next” button at the bottom of the form.
- Ensure that the VPC that is selected says “Default” underneath
- In the “Subnets” field, click in the “Select subnets” dropdown and select the subnet
- associated with “us-east-1a.”
- Click the orange “Next” button at the bottom right of the form.
- On the next page, focus on the middle of the three boxes entitled “Health checks.” Below “Health check grace period,” change the field to 120 seconds.
- Click the orange “Next” button at the bottom right of the form.
- On the next page, under the “Group size” box, change the fields to reflect the following settings:
    - Desired capacity: 1
    - Minimum capacity: 1
    - Maximum capacity: 3
- This tells AWS that we want our Auto Scaling Group to start with a preferred number of just one (1) EC2 instance, and that this is also the minimally acceptable number of instances. However, if scaling is needed, we are capping the maximum number of instances to three (3).
- Click the white “Skip to review” button at the bottom of the form.

- Scroll through the summary of reviewed content, ensuring that everything was configured properly, and then click the orange “Create Auto Scaling group” button at the bottom of the form.

Task 3: Create an Auto Scaling Policy with CloudWatch Alarm

In this task, we will create an Auto Scaling Policy, driven by a CloudWatch Alarm that is monitoring the CPU utilization of our instances. CPU utilization is a great metric to monitor on an EC2 instance to determine whether or not the instance is capable of serving the end users. High utilization means that the instance is reaching its capacity limits, and we want to configure our Auto Scaling Policy to respond to such an event by adding more instances (scaling out/horizontal scaling).

- While still in the EC2 area of the AWS Management Console, on the left sidebar, scroll all the way to the bottom, and under the “Auto Scaling” heading, click “Auto Scaling Groups.” Verify that you are in the new Auto Scaling console by clicking the link in the blue bubble, as you did previously in step 29, if it appears.
- Click on your “My First Auto Scaling Group” item.
- Click the “Automatic scaling” tab.
- The first heading that appears under this section is labeled “Scaling policies (0).” This means we currently have no scaling policies applied, so our Auto Scaling Group isn’t doing much of anything right now. Let’s change that.
- Click on the white “Add policy” button in the right side of this section.
- Change the “Policy type” dropdown to “Simple scaling.”
- Change the “Scaling policy name” field to “My Scale-Out Policy.”
- Under the dropdown box for “CloudWatch alarm,” click on the blue link that says “Create a CloudWatch alarm.”
- A new window/tab will open and take you to the CloudWatch alarm creation wizard. Click on the white “Select metric” button.
- A pop-up will appear. Towards the bottom left of this pop-up, click on the box that says “EC2” to look at the types of EC2-related metrics CloudWatch can monitor.



- When the next set of boxes appears, click “By Auto Scaling Group.”
- Scroll through this list to find a metric for your “My First Auto Scaling Group” name that is associated with the “CPUUtilization” metric.
- Check the box for this row, then click the blue “Select metric” button at the bottom right of the pop-up.
- Back in the CloudWatch wizard, scroll down to the box labeled “Conditions.” The threshold type should already be set to “Static” and the “Greater >” threshold should be selected. In the field below “than…,” type 50, and click the orange Next button at the bottom right of the form. (This configures our CloudWatch Alarm to trigger whenever the average CPU utilization of instances in our Auto Scaling Group exceeds 50%.)
- On the next page, in the “Notification” box, verify that “Alarm state trigger” is set to “In alarm.” Under the “Select an SNS topic” section, click the button to the left of “Create new topic.”
- In the “Create a new topic…” field that appears, type in “MySNSTopic” (without quotes or spaces).
- In the field beneath “Email endpoints that will receive the notification…,” type in your e-mail address. Click the white “Create topic” button.

- Skip past the “Auto Scaling action” section, as this only works with certain types of Auto Scaling Policies that we won’t cover in this course. Click the orange “Next” button at the bottom right of the form.
- In the “Alarm name” field, enter “CPU Utilization Exceeds 50%” (without quotes).
- Click the orange “Next” button at the bottom right of the form.
- Scroll through the review page, making sure that your configurations were accepted correctly, and then click the orange “Create alarm” button at the bottom of the form.
- Keep this window/tab open, and open another to check your email. You will have an email from “AWS Notifications" with a link asking you to “Confirm subscription.” Click this link, and then close the window/tab that was opened to confirm your subscription.
- Go back to the first window/tab that still has our “Create scaling policy” form, and click the small refresh button next to the CloudWatch alarm dropdown.
- Click inside the CloudWatch alarm dropdown again, and you should have an option for the “CPU Utilization Exceeds 50%” alarm that you just created. Select it.
- Under the “Take the action” section, change the middle field to 1, so that the full statement reflects “Add 1 capacity units.”
- Under “And then wait,” set this to 120 seconds before allowing another scaling activity.
- Click the orange “Create” button at the bottom right of the form.
- Repeat steps 47 through 70, but with the following modifications:
    - This scaling policy’s name will be “My Scale-In Policy” (without quotes).
    - When creating a new CloudWatch alarm, use the same CPUUtilization metric for the same Auto Scaling Group name, but in the “Conditions” box, change the threshold to “Lower <” and set the number to 40.
    - In the “Notification” box, do not create a new topic again. Select the option for “Select an existing SNS topic,” click in the field below “Send a notification to…” and select the “MySNSTopic” that you’ve already created.

- This alarm's name should be "CPU Utilization is Below 40%" (without quotes).
- Back in the scaling policy configuration, "Take the action" should be set to "Remove 1 capacity units." Make sure to change all fields to reflect this statement.

Task 4: Stress Your EC2 Instance

Now that we've done all of the work associated with setting up a Launch Configuration, Auto
Scaling Group, and Auto Scaling Policies/CloudWatch Alarms for scale-out and scale-in, it's time to test our Policies by stress-testing the default EC2 instance in the Group, and seeing if the Policies respond appropriately by scaling our infrastructure.

- While still in the EC2 area of the AWS Management Console, on the left sidebar, under the "Instances" heading, click "Instances."
- Click on the row that contains the default EC2 instance that was created by the Auto Scaling Group. (Hint: It won't be the EC2 instance you used from Lab #1, and it probably has a blank space instead of a name).
- Make a note of the "IPv4 Public IP" that appears in the Description section at the bottom of the EC2 Dashboard.
- Connect to this EC2 instance by following the same steps you did in step 62 of Lab #1, but using the IP address from step 74 of this Lab instead.
  NOTE: Mac users can skip the step to "chmod" their .pem file, and Windows users can skip the step of using PuTTYgen to make a .ppk file, as these are one-time tasks that you've already done during the first Lab.
- Once presented with your prompt that begins with "centos," run the following command to become the "root" user:
  - sudo su
- Run the following command to install the epel-release repository, which is required so that we can locate the stress-testing tool.
  - yum -y install epel-release
- Now, run the following command to install our stress-testing tool:

        - yum -y install stress
- Once presented with the “Complete!” message and returned to a prompt, run the following command to begin a stress test of this instance’s CPU:
        - stress --cpu 12 --timeout 660
- You will receive a message about “stress: info: Dispatching hogs” and be unable to use your terminal for 11 minutes (660 seconds). Remember that our CloudWatch alarms were configured by default to only check the status of our instances every five (5) minutes, which is why we’ve set our stress test to at least cover two (2) status checks.
- Navigate back to your EC2 Instances list, and keep clicking the refresh button. As long as our stress test continues, you should see at least one more instance appear. If you do not see this third instance appear, and your stress test has completed with a message in your terminal about “successful run completed,” run the command from 2-steps above again.

Task 5: Clean-Up of Both Labs

AWS Free Tier generously gives us a lot of free credit to use for these Labs, but we need to
make sure that we terminate any resources that we aren’t using so that we can conserve our
credits for other projects (and make sure that we aren’t charged real money when our year of
Free Tier expires!)

1. On the left sidebar of the EC2 section of the AWS Management Console, scroll all the way to the bottom, and under the “Auto Scaling” heading, click “Auto Scaling Groups. If you are prompted again about using the new console, ensure that you follow the link to use that new console.
2. Verify that “My First Auto Scaling Group” is checked, and click the white “Delete” button at the top right of the table.
3. You will be prompted to confirm deletion by typing “delete” (without quotes) in the field. Do this, and then click the orange “Delete” button.

4. On the left sidebar, scroll back to the bottom. Under the “Auto Scaling” heading, click “Launch Configurations.”
5. Right-click on the row that contains “My First Launch Configuration” and select “Delete launch configuration.” When prompted, confirm the deletion by clicking the blue “Yes, Delete” button in the pop-up.
6. On the left sidebar, under the “Images” heading, click “AMIs.”
7. Right-click on the row that contains “My First AMI” and select “Deregister.” When prompted, confirm the deregistration of your AMI with the blue “Continue” button.

The following steps will terminate your resources from Lab #1. If you intend to build upon your infrastructure from Lab #1 for your Projects, or want to keep those resources for any reason, DO NOT perform these steps.

1. On the left sidebar, under the “Instances” heading, click “Instances.”
2. Check the boxes next to any remaining instances that you no longer intend to use. Click the gray “Actions” dropdown button at the top of the table, and select “Instance State > Terminate.” When prompted to confirm, click the blue “Yes, Terminate” button at the bottom-right of the pop-up. This cannot be undone.
3. Click the “Services” menu at the top of the AWS Management Console.
4. Under the “Storage” section, click “S3.”
5. Check the box next to your S3 bucket name, and click the white-and-blue “Delete” button at the top of the table. You will be prompted to enter the name of your bucket perfectly to confirm deletion.

That’s it! You have successfully created all of the necessary configurations to make an AMI
based on your EC2 instance from Lab #1, create a Launch Configuration with a custom Security
Group (even though we didn’t use it, this is an important step if you were intending to scale-out
your Web site), and configure the necessary Auto Scaling Policies and CloudWatch Alarms to

respond to your instance’s CPU utilization. Large businesses all over the globe have performed
the same steps you did to ensure their systems are always highly-available. It wasn’t that hard,
was it? Just one of the many cool, powerful features of cloud computing and AWS!