Lab 4: Reducing data transfer costs using CloudFront and VPC endpoints v1.0.5

Sunday, August 9, 2020 7:22 PM

Lab 4:

- Step 62 is no longer needed with the new EC2 console.
- WAIT for the AMI to be created before completing Task 3.2 or you might get an error.

Lab 4: Cost Optimization: Reducing data transfer costs using CloudFront and VPC endpoints

2 hours Free

In this lab, you will deploy an Amazon CloudFront distribution and an Auto Scaling group in order to improve end-user network latency and optimize network and compute costs.

Scaling group in order to improve end-user network latency and optimize network and compute costs.

Objectives

After completing this lab, you will be able to:

- Configure static and dynamic origins in CloudFront.
- Manage CloudFront caching for your static data.
- Use Auto Scaling to replace large EC2 instances with smaller and more cost effective instances.
- Create VPC endpoints for DynamoDB and Session Manager.

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat).
- The qwikLABS lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the student guide.
- For Microsoft Windows users: Administrator access to the computer.
- An Internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported).
- An SSH client such as PuTTY.

Duration

This lab will require **90** minutes to complete.

Start Lab

Start Lab

1. At the top of your screen, launch your lab by choosing **Start Lab**

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

   ℹ If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing **Open Console**

   This opens an AWS Management Console sign-in page.

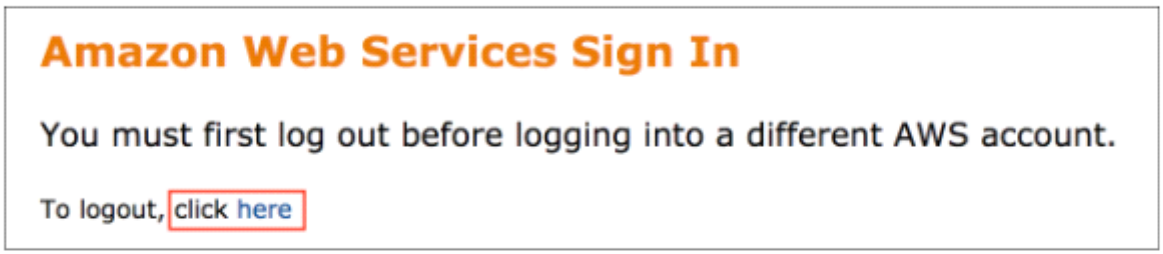
3. On the sign-in page, configure:

   - **IAM user name:** `awsstudent`
   - **Password:** Paste the value of **Password** from the left side of the lab page
   - Choose **Sign In**

   ⚠ **Do not change the Region unless instructed.**

Common Login Errors

Error: You must first log out



If you see the message, **You must first log out before logging into a different AWS account:**

If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **click here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Introduction

Imagine your company is currently hosting their Accounts Payable web application on two large EC2 instances in private subnets. The team is using an application load balancer to split traffic across two availability zones. The team is using Amazon DynamoDB to store the relevant accounts payable data, leveraging two NAT gateways that communicate between DynamoDB and Amazon EC2 (diagram below).

Currently the application hosts a significant amount of static web content, as well as some dynamic data served from DynamoDB. Customers have reported that there is a significant latency request depending on the size of the request and the location from which the request originates. Additionally, your financial team has indicated the web application is over budget. Thus, you are tasked with optimizing both cost and performance for the web application.

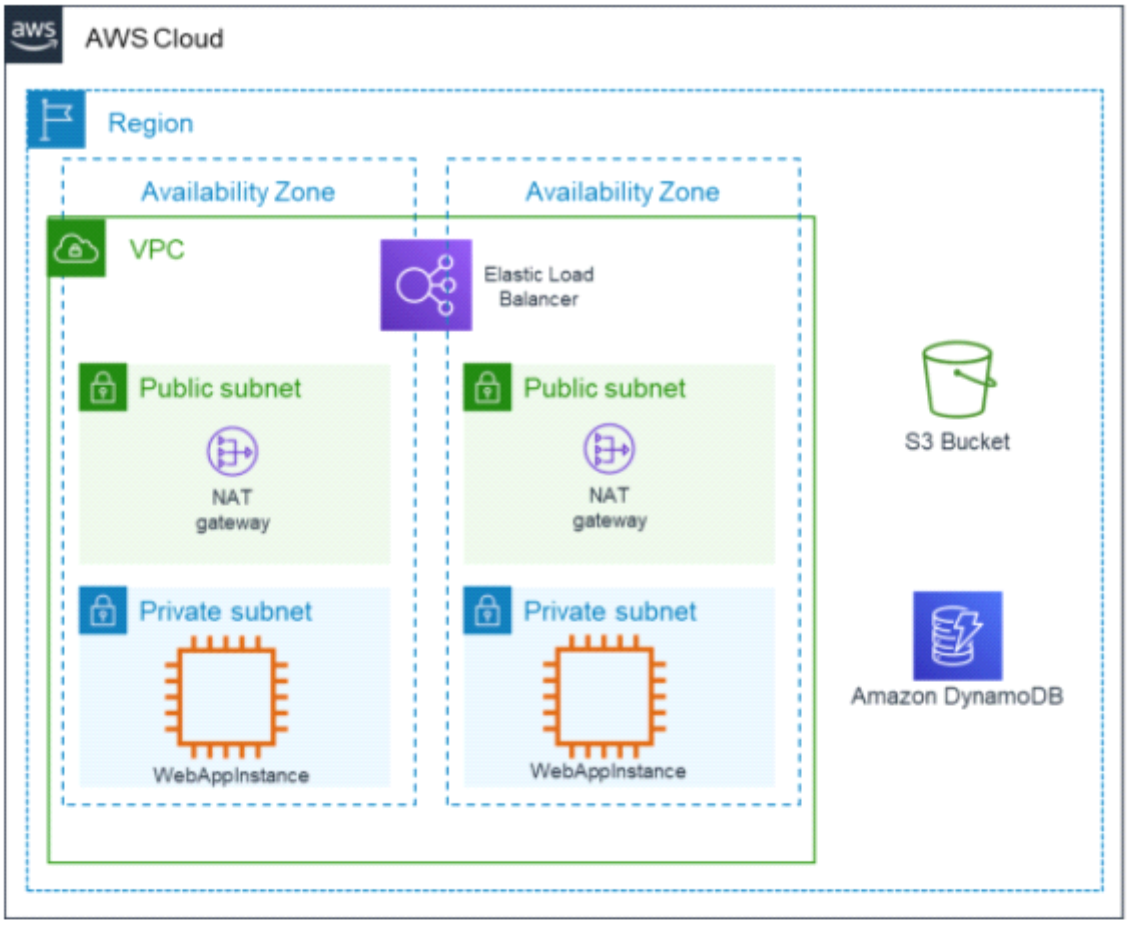
CloudFront is a global Content Delivery Network (CDN) service. Data transfers from EC2 and S3 to CloudFront cost nothing. If you move high volumes of data to your users, such as videos, images and audio, then CloudFront can help you keep your data transfer costs down. In this lab, you will build a CloudFront distribution to leverage S3 for your static data and EC2 instances for dynamic data.

EC2 instances for dynamic data.

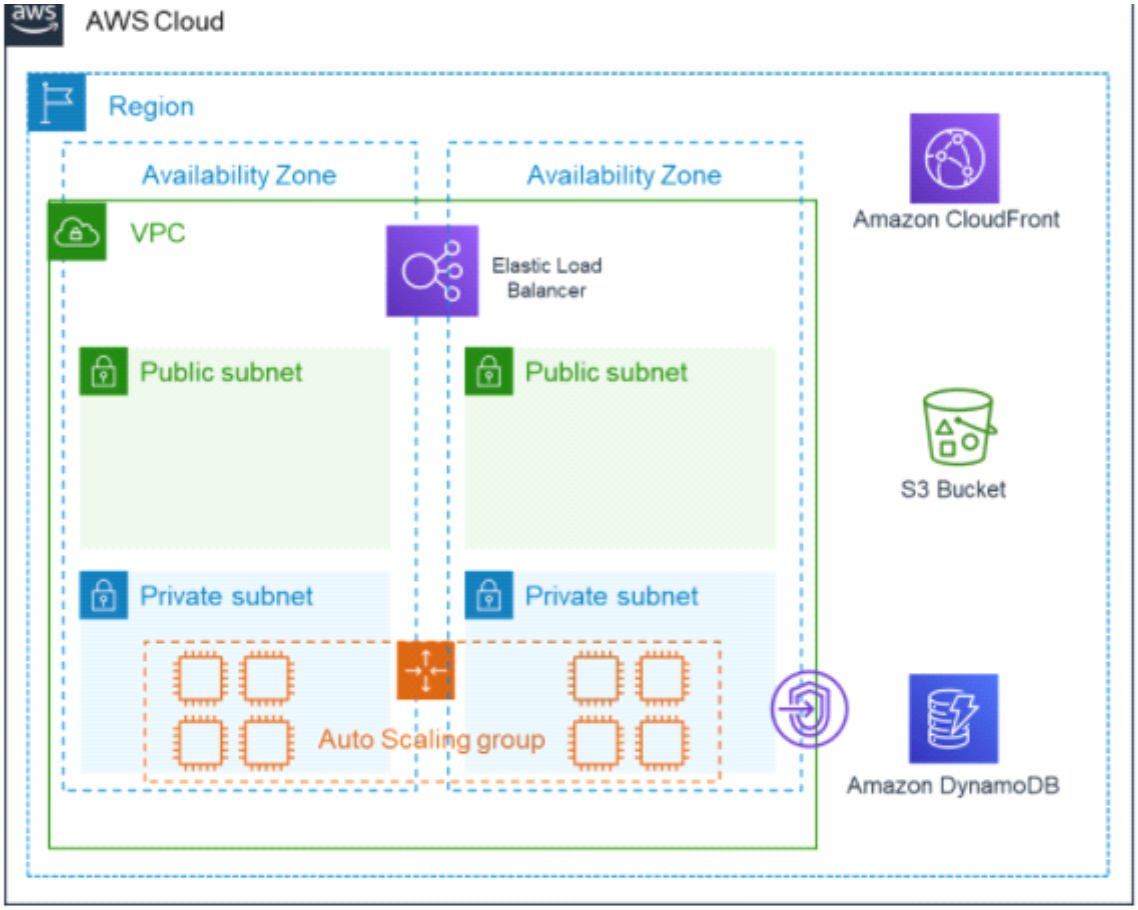
Once you have your CloudFront distribution built, you will create an Auto Scaling group for the Web Application. This will allow you to approach a true pay-for-what-you-use model by creating smaller and less expensive instances that can scale to meet demand that is more granular without over-provisioning.

Next, you will then remove the NAT Gateways used to communicate with DynamoDB and Systems Manager, and replace them with VPC endpoints. While NAT Gateways will get the job done, VPC endpoints will decrease the total cost of data transfer as well as increase the overall security of the architecture.

Starting Architecture:



Final Architecture:



Below is a screenshot of the AWS Simple Monthly Calculator showing the cost of Your Web App environment at the start of the lab. The prices are as of 5/28/2020. You can select the link below to find an estimate online with current prices.

Services | Estimate of your Monthly Bill ($ 1349.93)

Estimate of Your Monthly Bill

Show First Month's Bill (include all one-time fees, if any)

Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.

Export to CSV | Save and Share

Amazon EC2 Service (US East (N. Virginia))	$ 997.12
Amazon S3 Service (US East (N. Virginia))	$ 0.26
Amazon Elastic Load Balancing (US East (N. Virginia))	$ 17.06
Amazon DynamoDB Service (US East (N. Virginia))	$ 211.20
Amazon VPC Service (US East (N. Virginia))	$ 79.38
AWS Data Transfer Out	$ 44.91
AWS Support (Basic)	$ 0.00
Total Monthly Payment:	**$ 1349.93**

https://calculator.s3.amazonaws.com/index.html#key=files/calc-

https://calculator.s3.amazonaws.com/index.html#key=files/calc-ca04245a424bba0efe7e915b8b31ec7428da4ef5&r=IAD&v=ver20200527oH

Task 1: Setup and configure the CloudFront Distribution

In order to leverage CloudFront to decrease data transfer costs and improve latency you will need to configure from where the data originates, as well as the desired caching behavior.

In this task, you will create two CloudFront distribution origins - one to deliver the static web content in an S3 bucket and another to deliver dynamic content from application servers routed through the load balancer. Then, you will configure the distribution behaviors to handle static and dynamic traffic.

Task 1.1: Configure CloudFront distribution origins for dynamic content

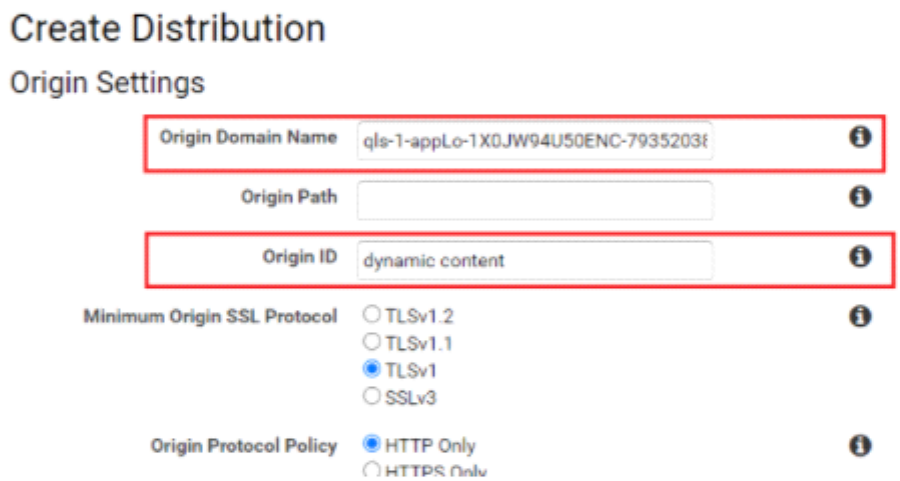
To begin, you will need to create a CloudFront distribution for the application servers to be able to deliver dynamic data.

4. On the navigation bar, choose **Services**, and choose **CloudFront**.
5. Select **Create Distribution**
6. Within the **Web** section, select **Get Started**
7. In the **Origin Settings** menu, select into the **Origin Domain Name** field to

7. In the **Origin Settings** menu, select into the **Origin Domain Name** field to populate a dropdown list of options.

8. Navigate to the **Elastic Load Balancers** section and choose the only option.

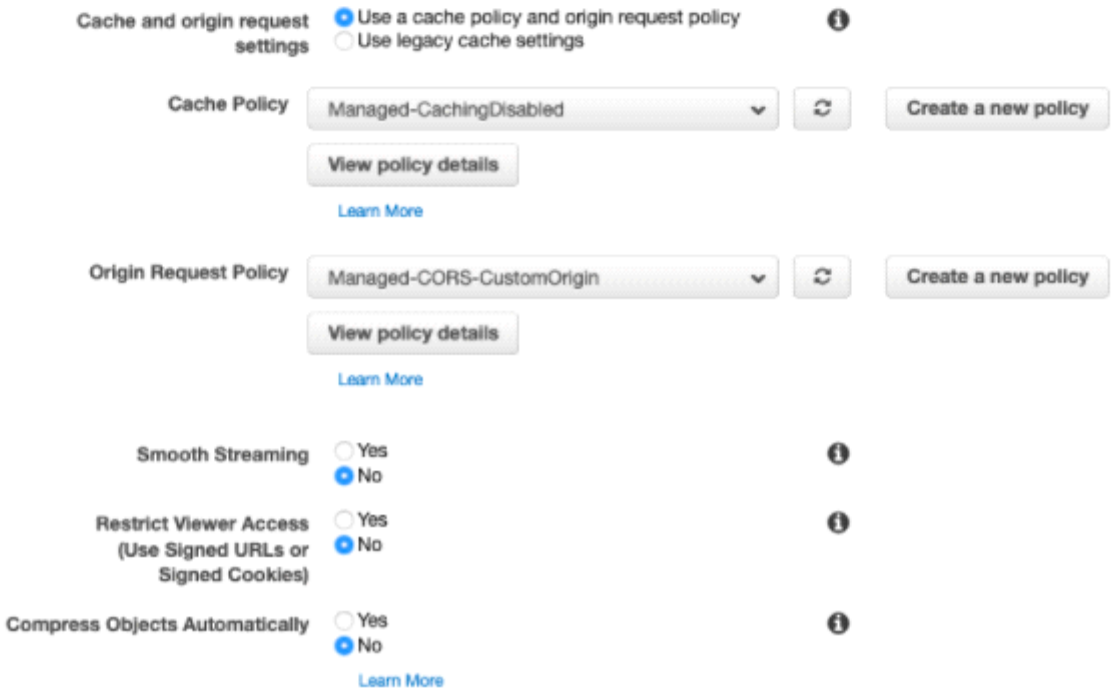
9. For **Origin ID** replace the default value with `dynamic content`



10. Configure **Default Cache Behavior Settings**:

    - For **Cache Policy**, select **Managed-CachingDisabled**.
    - For **Origin Request Policy**, select **Managed-CORS-CustomOrigin**.

    With these policy selections, CloudFront will not cache dynamic data and forward the correct information to the origin about the request.



11. At the bottom of the page, choose **Create Distribution**

    The **Status** column shows **In Progress** for your distribution. After CloudFront has created your distribution, the value of the **Status** column for your distribution will change to **Deployed**. At this point, it will be ready to process requests. This should take around 5-10 minutes, there is no need to wait for the distribution to be complete. Please continue with the next steps.

Task 1.2: Configure CloudFront distribution origin for static content

CloudFront now knows where your Elastic Load Balancer is, and you know the domain name associated with the distribution. At this point in time, the CloudFront domain name will route users directly to your Elastic Load Balancer only. You must now add another origin for the static content in S3 that can be cached.

12. Select the newly created Distribution **ID**.

13. Choose the **Origins and Origin Groups** tab.

13. Choose the **Origins and Origin Groups** tab.

14. Choose **Create Origin**

15. In the **Origin Domain Name** field input in the **regionSpecificS3bucketlocation** from the navigation panel to the left of these instructions.

    Note: In most cases, you would choose the bucket name directly from the drop down, similarly to what you did in 1.1. However, because of the nature of new buckets, you must specify the region of the bucket to ensure that the cache behavior is effective immediately.

16. For **Origin ID** replace the default value with `static content`

17. For **Restrict Bucket Access** choose *Yes*.

18. For **Origin Access Identity** choose *Create a New Identity*.

19. In the **Comment** entry field, enter `static content for web application`

20. For **Grant Read Permissions on Bucket** choose *Yes, Update Bucket Policy*

21. Choose **Create**

Task 1.3: Configure CloudFront distribution behaviors for your static origin

CloudFront now has two origins, Amazon S3 and the Elastic Load Balancer used to distribute traffic to the application servers. The only thing left to do, is to configure the behavior for the static data in S3. You have already set the behavior for dynamic data when you created the distribution in task 1.1 and you can find it listed under Behaviors as Default(*).

22. Choose the **Behaviors** tab.

22. Choose the **Behaviors** tab.

23. Choose **Create Behavior**

24. For the **Path Pattern** enter `static/*`

25. For the **Origin or Origin Group** dropdown list, choose *static content*.

26. For **Origin Request Policy**, select **Managed-CORS-S3Origin**.

27. At the end of the configuration settings, choose **Create**

Task 2: Migrate static assets to Amazon S3 and verify CloudFront caching

Now that you have created the caching and routing behavior for the static objects, you will need to move the static objects to the S3 bucket you specified when creating the behavior. In order to accomplish this, you will connect to one of your EC2 instances via Systems Manager and copy the files to S3.

Note: *This task could be accomplished using traditional SSH with an SSH Key, however Session Manager, which does not require an SSH Key, allows system administrators to control access through IAM Roles and Policies.*

Task 2.1: Migrate Static Assets to S3

28. On the navigation bar, choose **Services**, and choose **Systems Manager**.

Important: Make sure that you are using the Region listed in the

28. On the navigation bar, choose **Services**, and choose **Systems Manager**.

    Important: Make sure that you are using the Region listed in the navigation panel to the left of these instructions.

29. In the navigation panel on the left, choose **Session Manager**.

30. Choose **Start session**

31. Choose **WebApp-Server-1** from the list of **Target instances**.

    Note: *As WebApp-Server-1 and WebApp-Server-2 are identical, you could accomplish the same task by choosing WebApp-Server-2, however for lab purposes please select WebApp-Server-1.*

32. Choose **Start session**

33. From the terminal, run the following command. Replace <InsertYourBucketNameHere> with your value of the **S3bucketname** from the navigation panel to the left of these instructions:

```
aws s3 cp --recursive /var/www/html/static
s3://<InsertYourBucketNameHere>/static/
```

    Example:

```
aws s3 cp --recursive /var/www/html/static s3://ql-
155751-9c54b7d99407914f-generatedbucket-
15v20u4pmixl1/static/
```

34. Confirm that the command has run successfully with no errors:

35. Close this browser window

35. Close this browser window.

36. In *AWS Systems Manager* console browser tab, in the **Sessions** section, select the radio button to choose the active session.

    Note: Refresh the page if you do not find the session listed already.

37. Choose **Terminate**

38. Choose **Terminate** again to end the session.

Task 2.2: Verify CloudFront caching

In this task, you will verify if the static content renders from CloudFront edge location closer to you than from the actual S3 bucket. To verify that, you will use the web browser's developer tools and pick a sample static file. The static file names used in your application end with file types: *.png .svg, .js,* or *.css*. The instructions in this task use the static file named ***APLogo.png***, but you can choose any static file.

Before proceeding, confirm that your CloudFront Distribution's status is in the **Deployed** state.

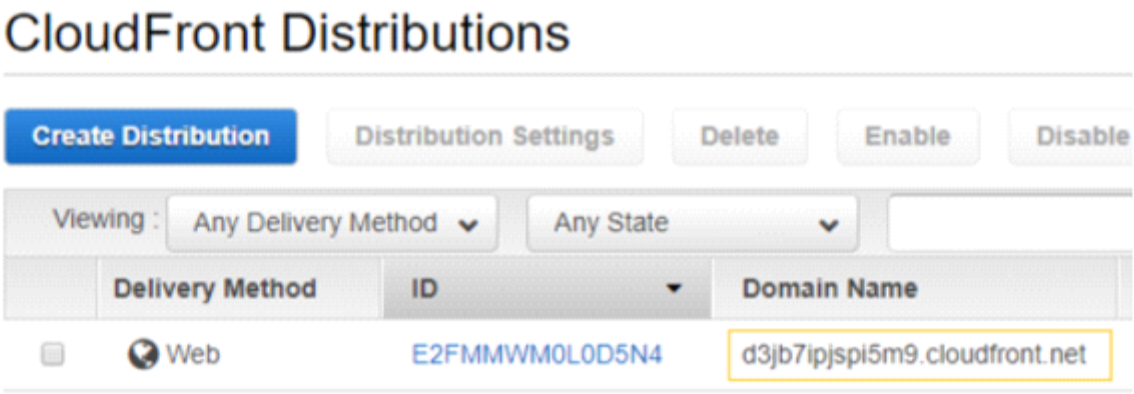
39. On the navigation bar, choose **Services**, and choose **CloudFront**.

    The domain name that CloudFront assigns to your distribution appears in the list of distributions.

40. Make sure that your CloudFront Distribution's status is in the **Deployed** state. If not then wait till the status changes to *Deployed*. Refresh the page as needed.

41. Copy the **Domain Name** for your CloudFront Distribution. Your Domain Name will be listed within the **CloudFront Distributions** table. It will look similar to

41. Copy the **Domain Name** for your CloudFront Distribution. Your Domain Name will be listed within the **CloudFront Distributions** table. It will look similar to *dm2afjy05tegj.cloudfront.net* :

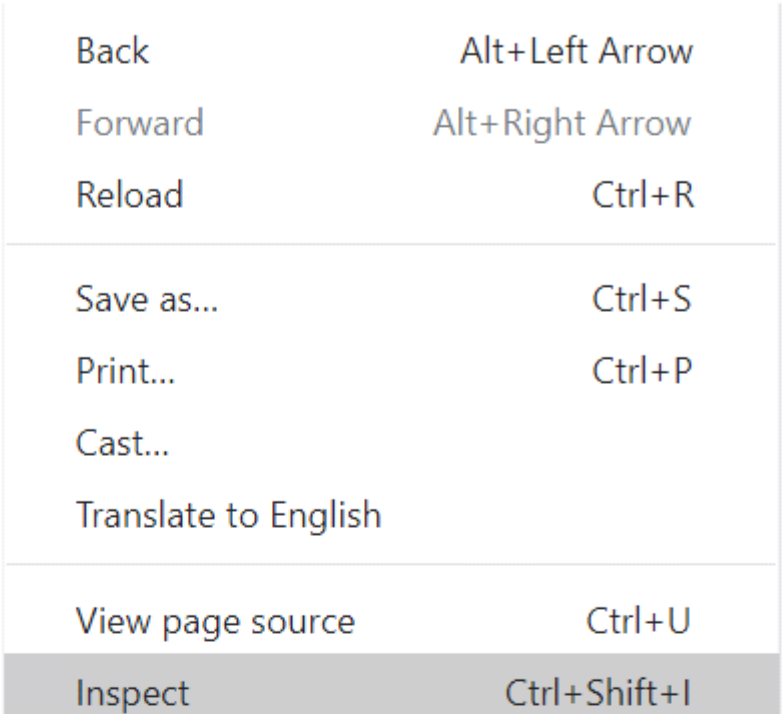


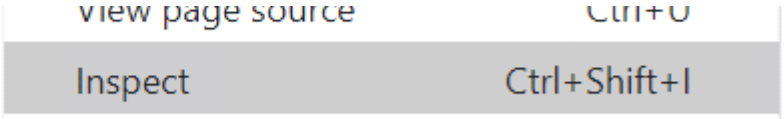
You will use the domain name to test the CloudFront distribution during a later step. Do not navigate to the site yet.

Note: *The images and steps for this are for Chrome Browser, Firefox instructions can be found here*

42. Open a new tab in your browser.

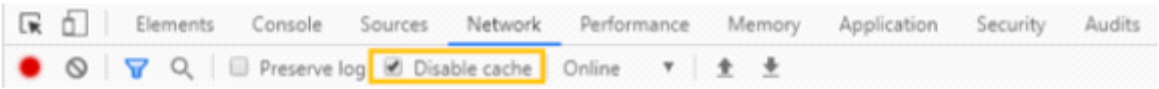
43. Before entering the CloudFront domain name, *Right Click* on the page and choose **Inspect**.





44. Choose the **Network** Tab.

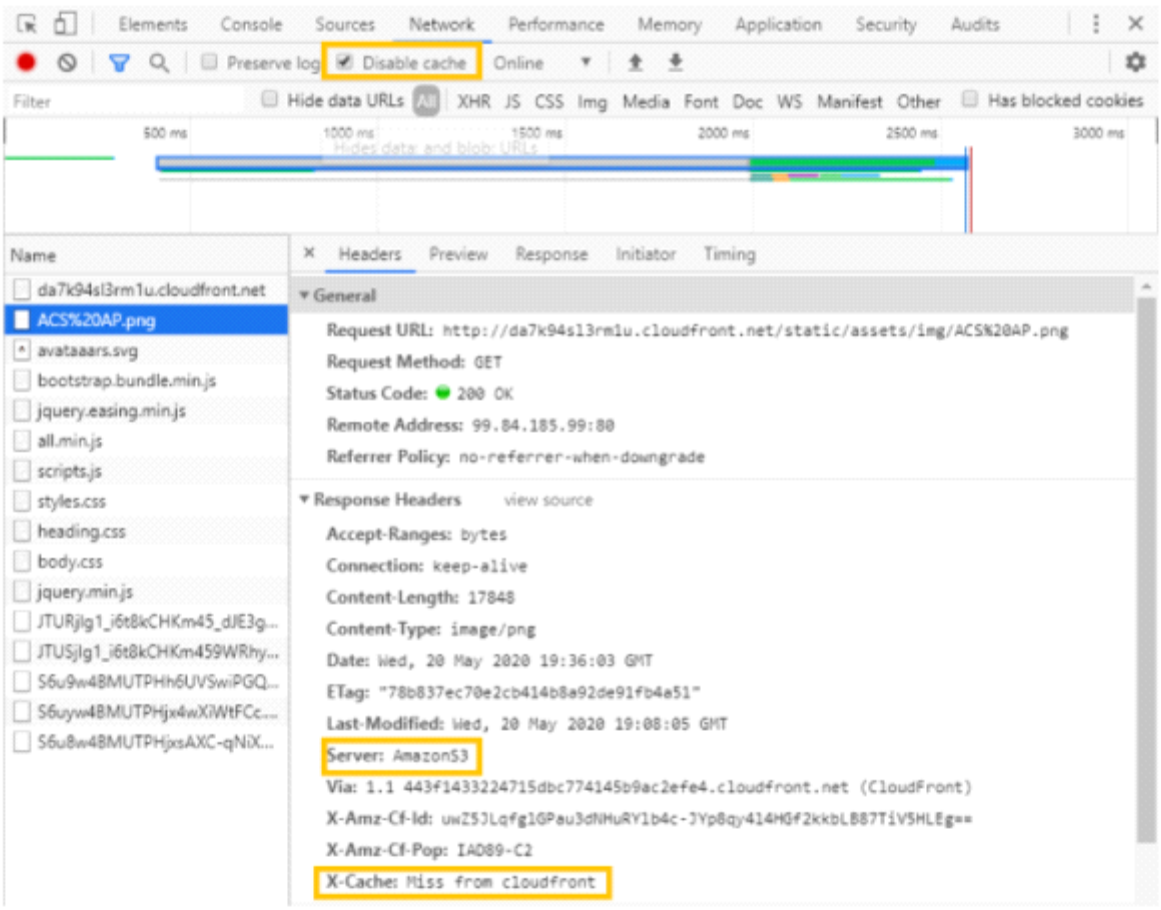
45. Select the **Disable cache** option, to prevent your browser from caching the static objects.



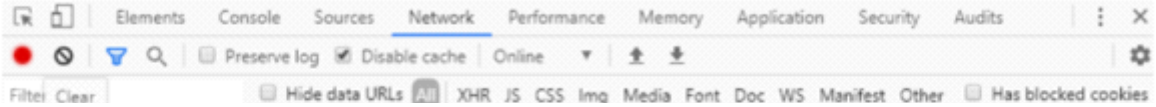
46. Enter your CloudFront Distribution domain name, noted down in a previous step, in the browser.

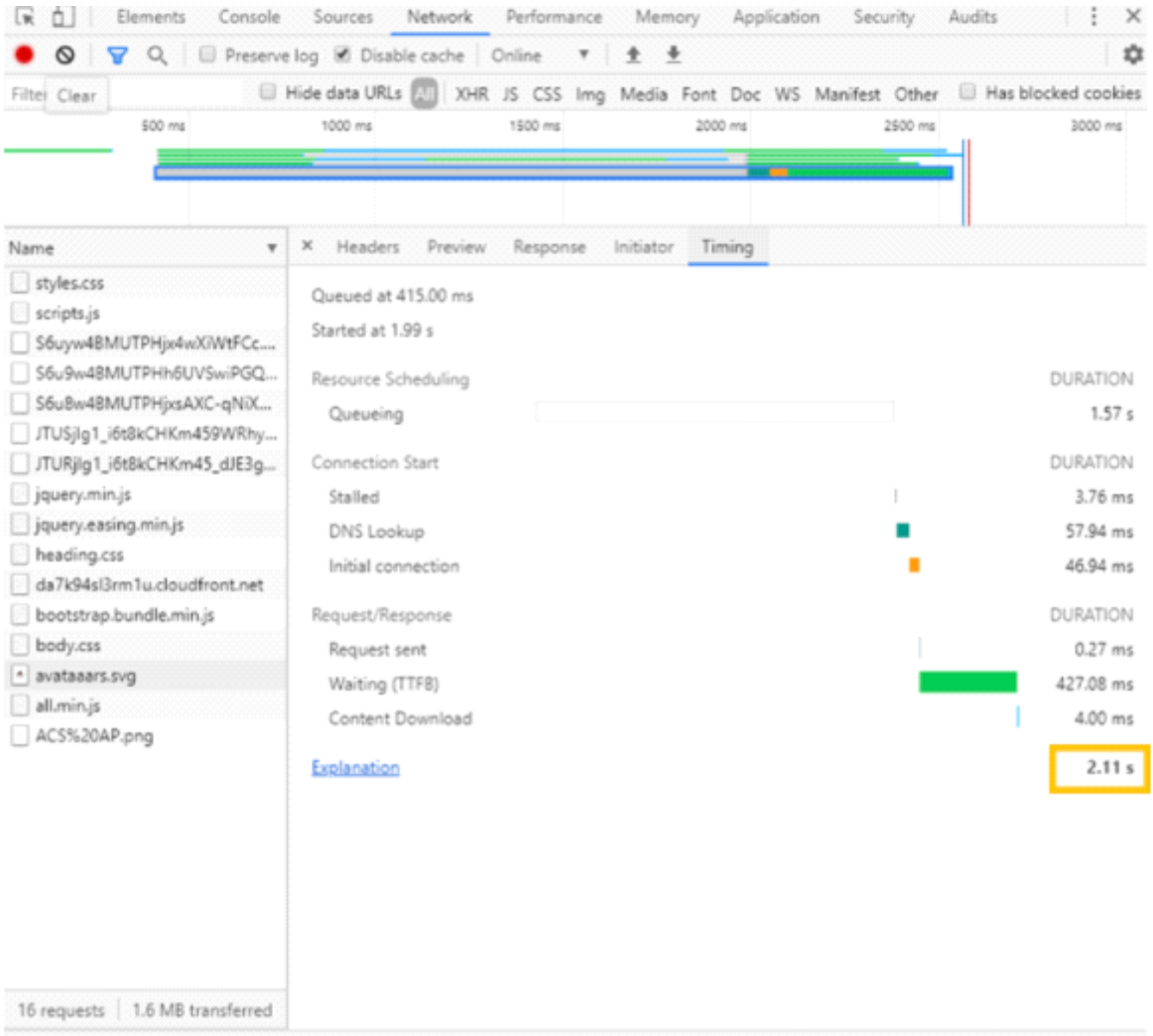
47. Select **APLogo.png** and select the **Header** tab.

48. In the **Response Headers** menu, you should notice:



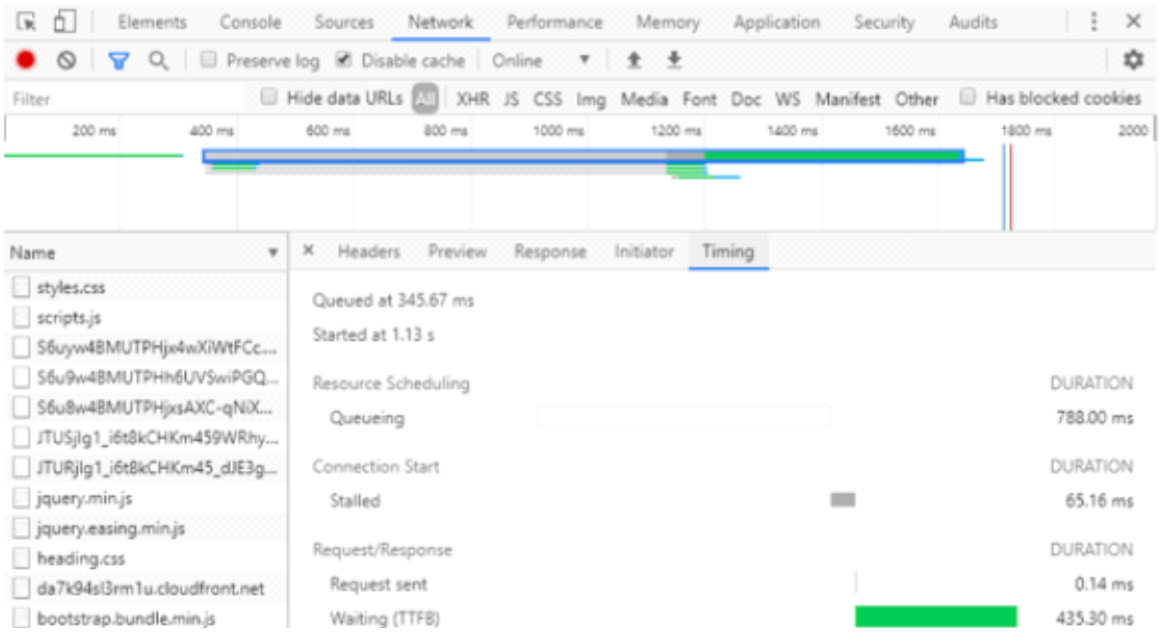
49. Select the **Timing** tab and note the similar timing details:

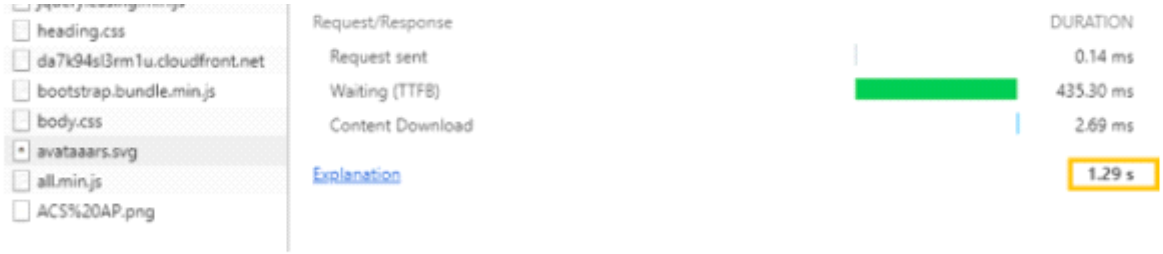




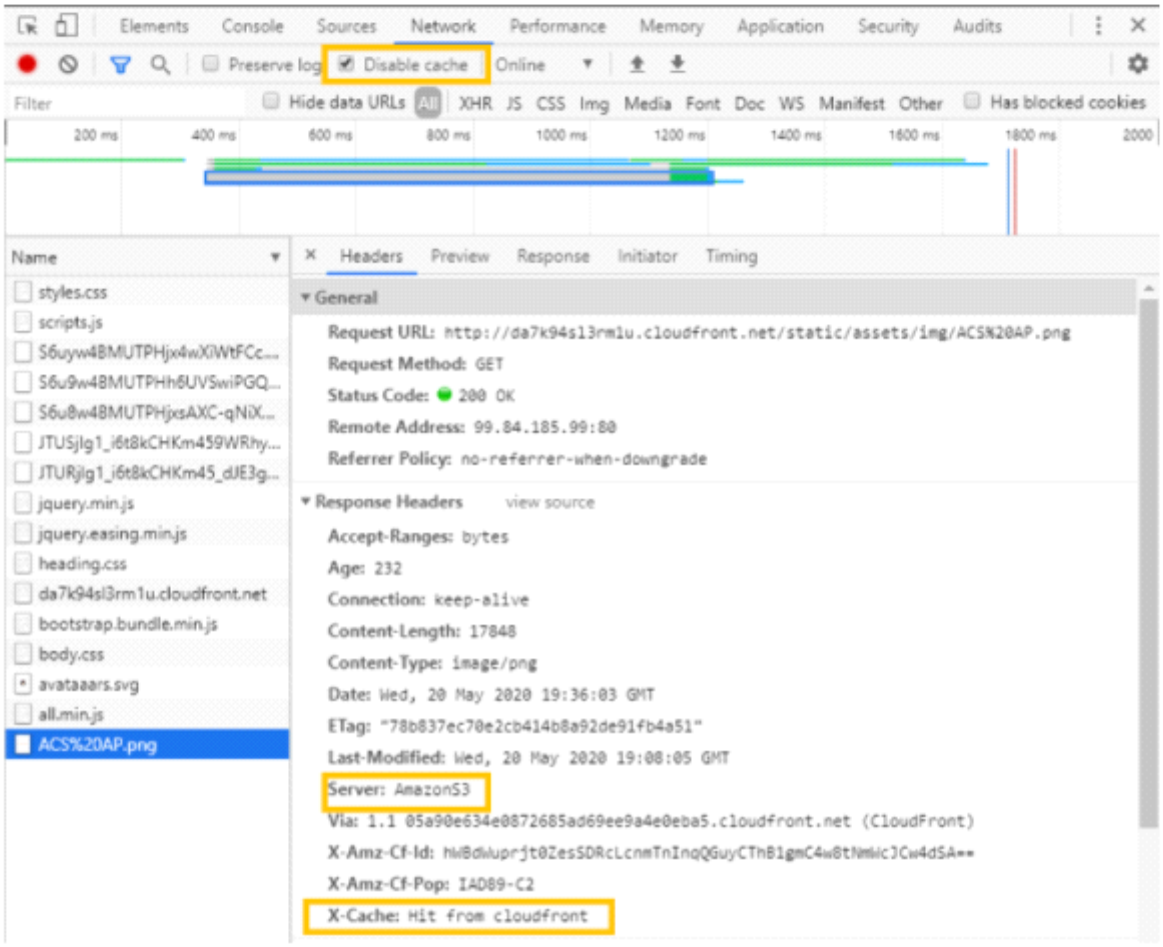
50. Reload the page

51. You should notice less latency, as the image is now cached in CloudFront. Note the timing details similar to the following:





52. Confirm that it was a Cache hit by returning to the **Header** tab.



Note: If you do not receive a cache hit in from CloudFront, reload the page again until you notice a cache hit.

Task 3: Leverage Auto Scaling for your Web Application

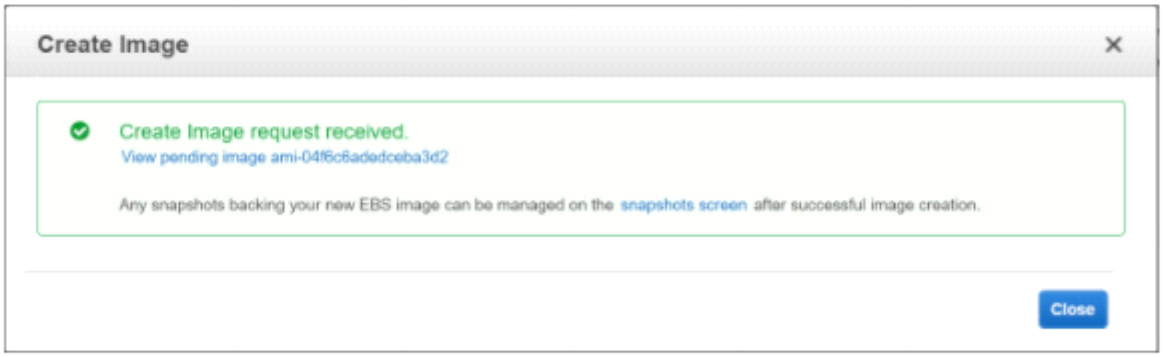
Web Application

Currently, your web application is hosted on a large *c5.4xlarge* instance. By leveraging an Amazon Machine Image that contains your Web Application code, you can use an Auto Scaling Group to host on smaller instances, at a significant decrease in cost, and automate scaling the Application up as demand dictates. Thus paying only for the capacity required.

Task 3.1: Leverage Auto Scaling for your Web Application

First, you need to create an image that will be used when the application needs to provision more instances to meet demand. For this you will leverage a Web Application instance currently in use.

53. On the navigation bar, choose **Services**, and choose **EC2**.
54. On the left navigation pane, choose **Instances**.
55. Select the check box next to **WebApp-Server-1**.
56. Choose the **Actions**
57. Choose **Image and templates**.
58. Choose **Create Image**.
59. In the **Image name** field enter `APWebServer`
60. For the **Image Description**, enter `Production AP Web Application`
61. Choose **Create Image**



62. Choose **Close**

Task 3.2: Create Launch Configuration

Now that you have established the Image of your Web Application, you need to specify a Launch Configuration that your Auto Scaling Group will leverage when there is an increase in demand and your application requires more resources.

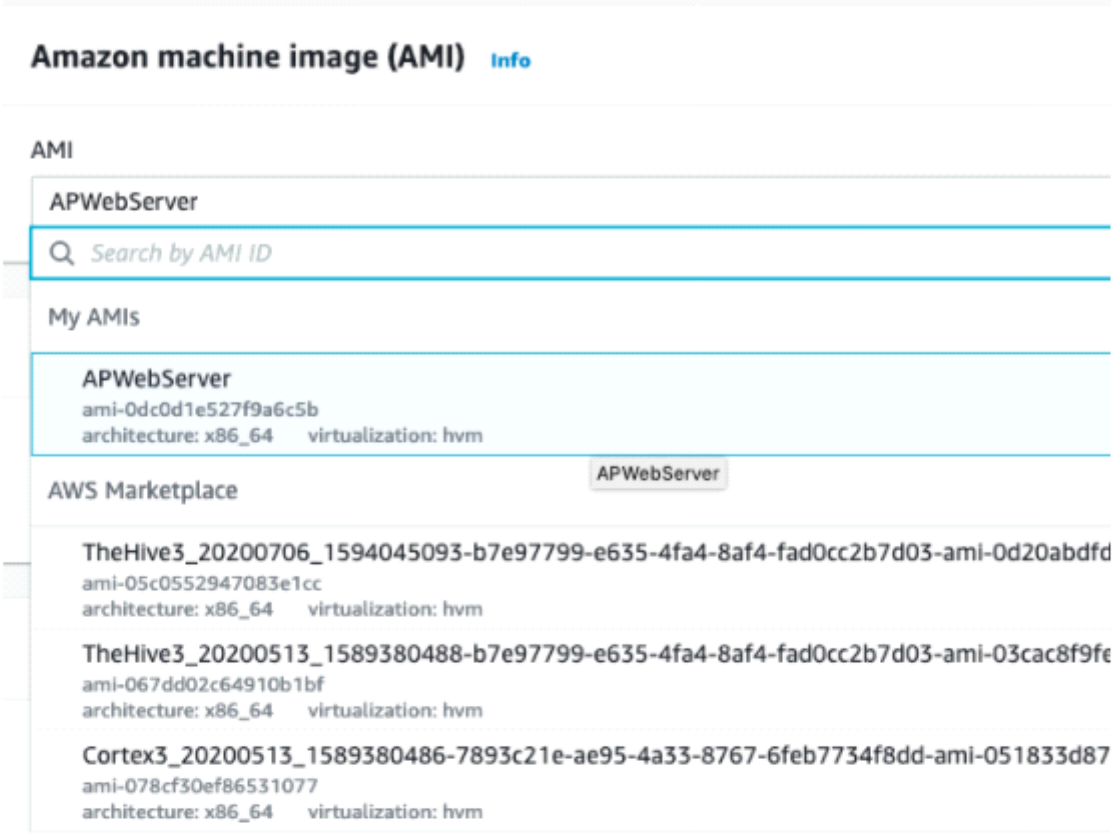
63. On the left navigation panel, choose **Launch Configurations** under ▾ **Auto Scaling**.

64. Choose **Create launch configuration**

65. Configure the following settings:

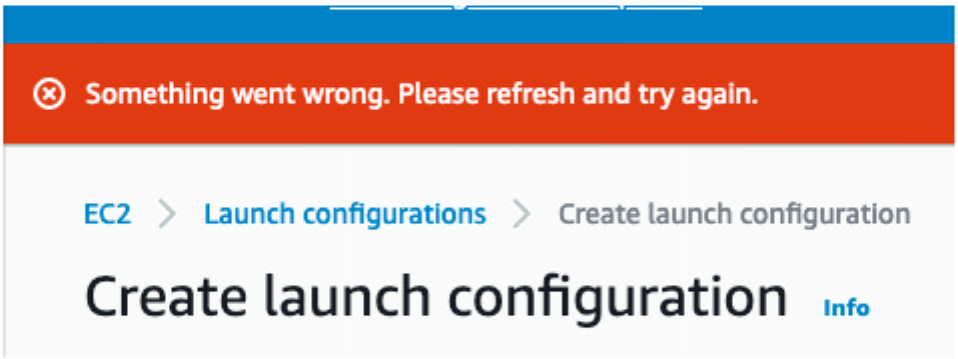
    ⚠ Leave the other settings as is.

    - **Launch configuration name**:
        - For **Name:** `WebServer`
    - **Amazon machine image (AMI)**:
        - For **AMI**, press *Choose an AMI*.
        - Under **My AMIs**, select **APWebServer**.



- **Instance type**:
    - For **Instance Type**, press Choose Instance Type
    - Select `t3.micro`, and press **Choose**

Important: You can ignore the Something went wrong error. This error is shown because you do not have permissions to view spot pricing in this lab.



- **Additional configuration** - ***optional***

- **Additional configuration** - *optional*
    - For **IAM instance profile**, select the role that contains **AppServerPermissions**
- **Security Groups**:
    - Choose **Select an existing security group**.
    - Select the security group with the description `Security Group for AP App Server`
- **Key pair (login)**:
    - For **Key pair options**, select **Proceed without a key pair**.
    - check the box for **I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI**.

66. Choose **Create launch configuration**

Task 3.3: Create Auto-Scaling Group

Now that you have defined the configuration, you will need to configure the Auto Scaling group and specify the metrics or events that will trigger scaling activities. In this task, you will use CPU usage as the scaling metric.

67. Select the launch configuration named **WebServer**.

68. Choose **Actions**

69. Choose **Create Auto Scaling group**.

70. For **Auto Scaling Group Name**, enter `web-server-ASG`

71. Choose **Next**

71. Choose **Next**

72. For **VPC**, select the *Lab-VPC*.

73. For **Subnet**, select both private subnets (*Lab-private-a*, *Lab-private-b*).

74. Choose **Next**

75. Select **Attach to an existing load balancer**.

76. Choose **Choose from Application or Network Load Balancer target groups**, select **ALBTargetGroup | HTTP** from the target group dropdown.

77. For **Health check type**, select **ELB**.

78. For **Health check grace period**, enter `240`

79. Choose **Next**

80. For **Group size - *optional*** configure:

    - **Desired capacity**: `2`
    - **Minimum capacity**: `2`
    - **Maximum capacity**: `4`

81. For **Scaling policies - *optional***, choose **Target tracking scaling policy**, and configure the following only:

    - **Target value**: `60`
    - **Instances need**: `240`

82. Choose **Skip to review**

83. Choose **Create Auto Scaling group**

Task 3.4: Stop the legacy EC2 instances

Now that you have implemented an Auto Scaling group to handle the web application traffic, you will now need to stop the legacy instances to ensure that they will not receive any traffic and billing will stop.

84. On the left navigation panel, choose **Instances** from within the ▾ **Instances** section

85. Select the check box next to the **WebApp-Server-2** and **WebApp-Server-1** instances.

86. Choose **Instance state**

87. Select **Stop instance**

88. Confirm by choosing **Stop**

Note: You should now notice two instances in the 🔴 **stopped** (or stopping) state, and two instances in the 🟢 **running** state. This may take 1-2 minutes to kick-in. Refresh the page until you notice the new instances.

Task 3.5: Verify Auto Scaling group

Before moving on, you need to confirm that the Auto Scaling group is configured correctly to serve your web application.

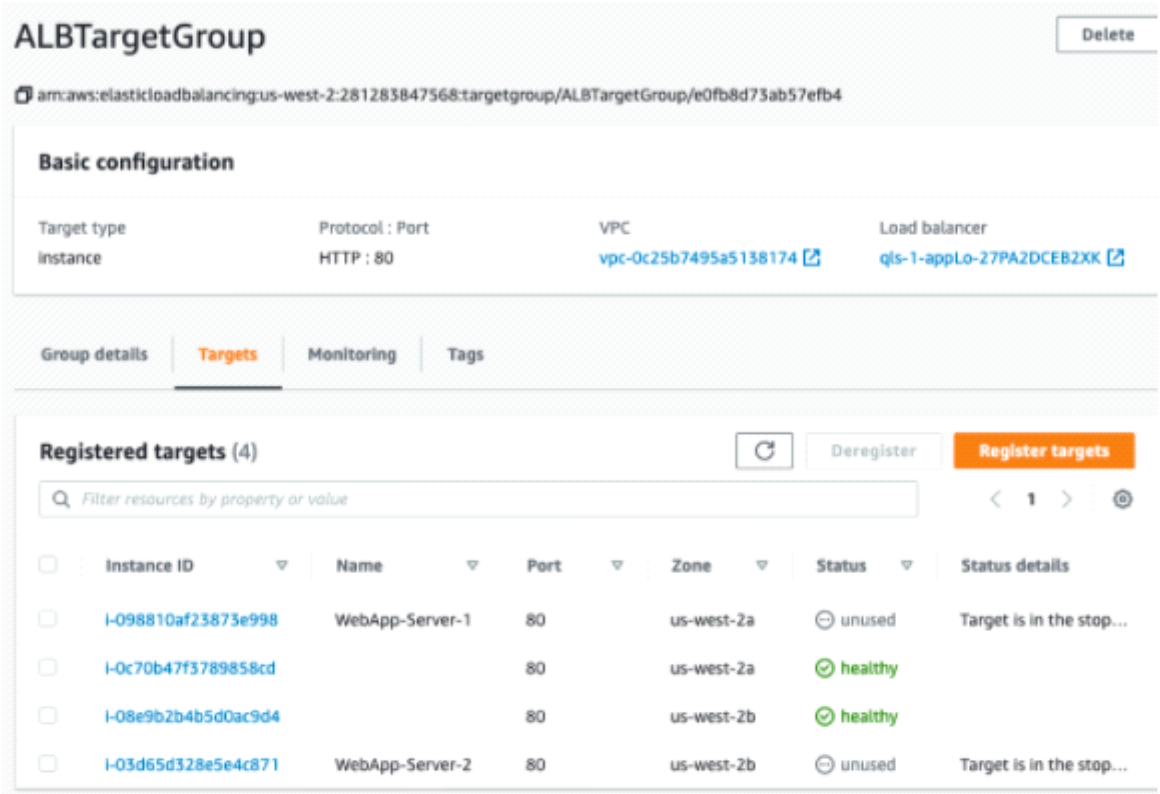
89. On the left navigation panel, choose **Target Groups** under ▾ **Load Balancing**.

90. Select the **ALBTargetGroup** name.

91. Select the **Targets** tab.

91. Select the **Targets** tab.

92. You will notice 4 total instances, two of which have the **Status** of *unused* and two of which have the **Status** of *healthy*. This confirms the Auto Scaling group has been added to the target group correctly.

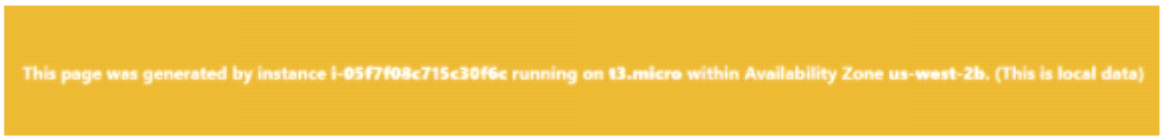


93. Select the check box next to **WebApp-Server-2** and **WebApp-Server-1** and select Deregister

    You have now removed the legacy large instances from the target group.

94. Return to the web application that is accessed with your CloudFront domain name.

95. In the footer of the web page, you should now notice that the site is hosted on a *t3.micro* instance type, confirming the web application is now being served by your autoscaling group.

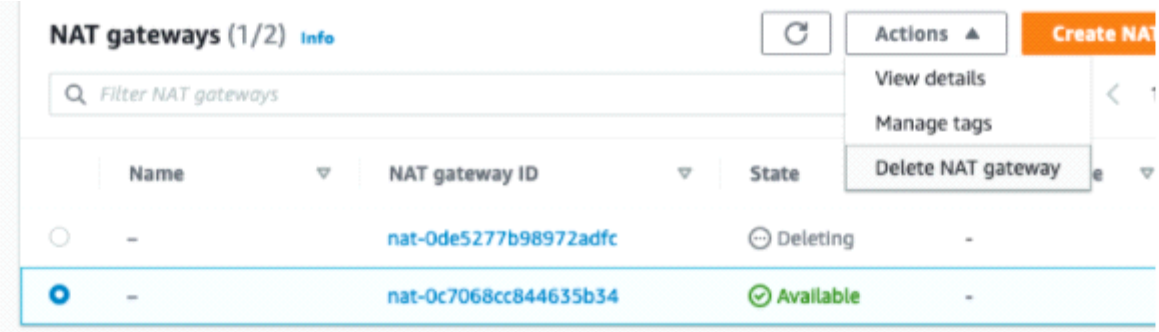


Task 4: Replace NAT Gateways with VPC endpoints

At the start of this lab, the NAT gateways allowed traffic out from the EC2 instances, primarily to access DynamoDB and Session Manager. While this does work, it is not the most cost efficient way to create these network connections. In this task, you will create VPC endpoints for DynamoDB and remove the NAT Gateways. This will result in a significant decrease in the per GB data processing charge as well as the hourly cost of using a NAT gateway.

Task 4.1: Remove NAT Gateways

96. On the navigation bar, choose **Services**, and choose **VPC**.
97. In the navigation panel on the left, choose **NAT Gateways**.
98. Choose the selection box for first of the two NAT Gateways listed.
99. Choose **Actions**
100. Choose **Delete NAT gateway**.



101. Enter `delete` into the confirmation box and choose **Delete**

102. Follow the same steps and delete the remaining NAT Gateway.

103. Refresh your console every so often, until you notice the status for both Gateway's as **Deleted**

     Note: This may take up to 5 minutes.

104. Return to the web application that is accessed with your CloudFront domain name to observe the impact of removing the NAT Gateways.

     After a 5 second timeout, the static data will load but there will be an error message for the dynamic data. This is because the EC2 instances no longer have a network connection to DynamoDB. You will fix that next.

Task 4.2: Create DynamoDB VPC Gateway Endpoint

Without a NAT gateway, you need to create an endpoint in the VPC to connect to DynamoDB.

105. In the navigation panel on the left, choose **Endpoints**.

106. Select **Create Endpoint**

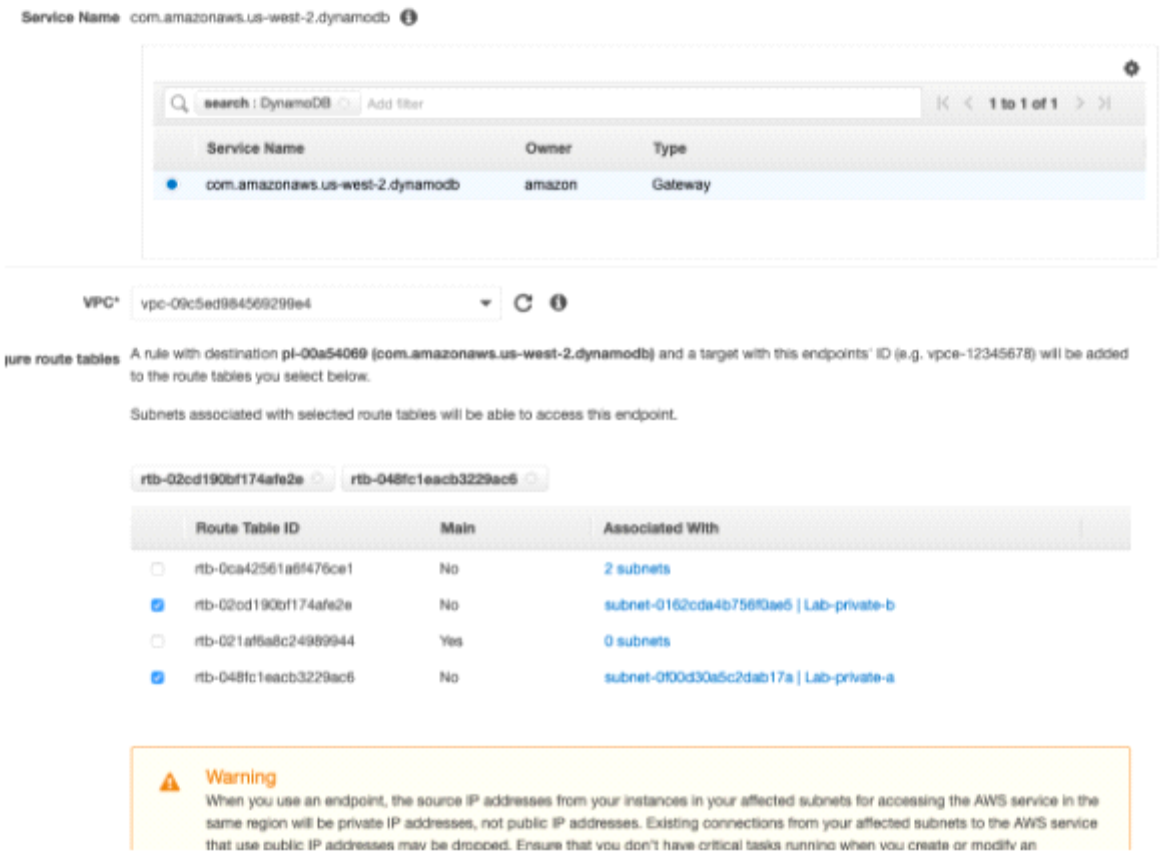
107. In the **Service Name** search bar input `DynamoDB` and press **ENTER**.

108. Select the only entry that appears.

108. Select the only entry that appears.

109. In the **VPC** dropdown list, select `Lab-VPC`

110. For **Configure route tables**, select the two **Route Table IDs** that are **Associated With** `Lab-private-a` and `Lab-private-b`



111. Choose `Full Access` for the **Policy**.

112. Choose **Create endpoint**

113. Choose **Close**

Note: Although the status of the endpoint may reflect available, it may take up to a minute for the route table to update. You can test your configuration by reloading your web application in your browser. If everything was successful, the application will load fully.

(Optional/Challenge) Task 5: Create

120. Copy and paste `Security Group for Session Manager Endpoint in Challenge` into the search bar and press **ENTER**.

121. Select the security group that is returned from the search.

132. Choose **Create endpoint**

133. Choose **Close**

Task 5.3: Interface Endpoint - ssm

134. Select **Create Endpoint**

135. In the **Service Name** search bar input `ssm` and press **ENTER**.

136. Select the option that matches *com.amazonaws.<LabRegion>.ssm*

137. In the **VPC** dropdown list, select `Lab-VPC`

138. For **Subnets**, select the two private **Subnet IDs** (`Lab-private-a` and `Lab-private-b`)

129. For **Security group**, remove the preselected group.

130. Copy and paste `Security Group for Session Manager Endpoint in Challenge` into the search bar and press **ENTER**.

the search bar and press **ENTER**.

131. Select the security group that is returned from the search.

132. Choose **Create endpoint**

Once all three of these endpoints have the status of available, you can test the access by connecting to an instance via Session Manager:.

143. Open the **Systems Manager Console** by selecting **Services** and typing `Systems Manager` in the filter box, and select **Systems Manager**.

144. In the navigation panel on the left, choose **Session Manager**.

145. Choose **Start session**

146. Choose any instance from the list of **Target instances**.

Note: *If there are no instances listed, wait 5 minutes and refresh the console.*

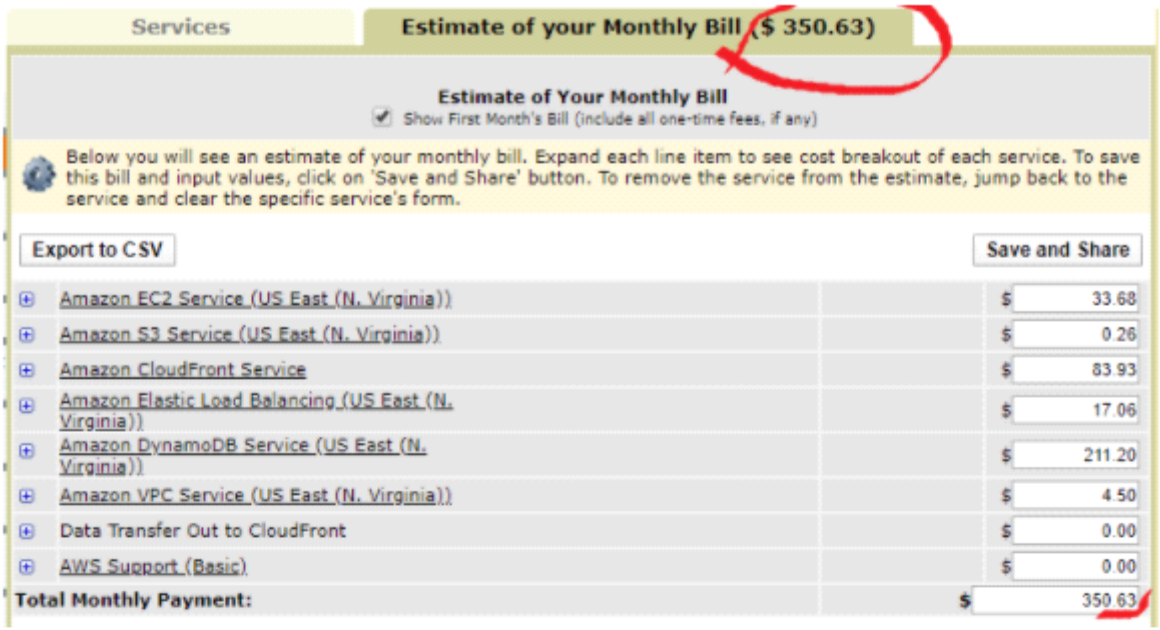
147. Choose **Start session**

148. If the terminal loads, then you have successfully configured the Interface endpoints!

Summary

Below is a screenshot of the AWS Simple Monthly Calculator showing the cost of Your Web App environment after implementing the optimizations in this lab. The prices are as of 5/28/2020. Note that it is in the legacy pricing

cost of Your Web App environment after implementing the optimizations in this lab. The prices are as of 5/28/2020. Note that it is in the legacy pricing application to ensure all services are accounted for. You can select the link below find an estimate online with current prices.



Services | Estimate of your Monthly Bill ($ 350.63)

Estimate of Your Monthly Bill

Show First Month's Bill (include all one-time fees, if any)

Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.

Export to CSV | Save and Share

Amazon EC2 Service (US East (N. Virginia))	$ 33.68
Amazon S3 Service (US East (N. Virginia))	$ 0.26
Amazon CloudFront Service	$ 83.93
Amazon Elastic Load Balancing (US East (N. Virginia))	$ 17.06
Amazon DynamoDB Service (US East (N. Virginia))	$ 211.20
Amazon VPC Service (US East (N. Virginia))	$ 4.50
Data Transfer Out to CloudFront	$ 0.00
AWS Support (Basic)	$ 0.00
Total Monthly Payment:	**$** 350.63

https://calculator.s3.amazonaws.com/index.html#key=files/calc-396042750e8d3656e38ba90f1776832b4f483ca1&r=IAD&v=ver20200527oH

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

149. Return to the AWS Management Console.

150. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.

151. Choose End Lab

151. Choose **End Lab**

152. Choose **OK**

153. (Optional):

- Select the applicable number of stars ☆
- Type a comment
- Choose **Submit**

    - 1 star = Very dissatisfied
    - 2 stars = Dissatisfied
    - 3 stars = Neutral
    - 4 stars = Satisfied
    - 5 stars = Very satisfied

You may close the window if you don't want to provide feedback.

For more information about AWS Training and Certification, see *http://aws.amazon.com/training/*.

Your feedback is welcome and appreciated.

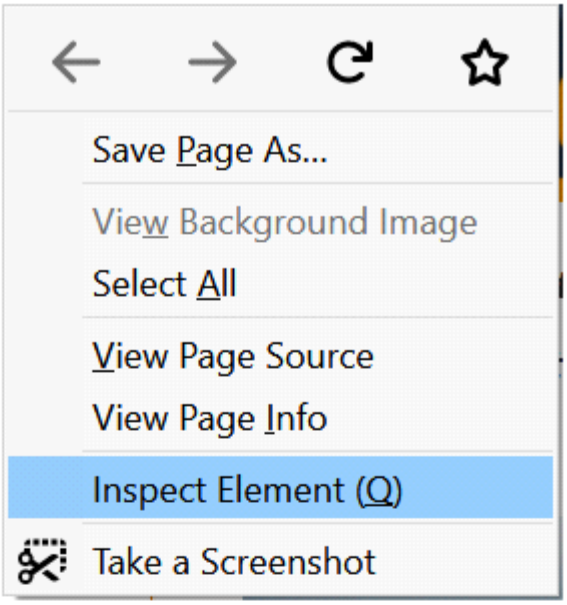
If you would like to share any feedback, suggestions, or corrections, please provide the details in our *AWS Training and Certification Contact Form*.

Appendix: Firefox cache validation

154. Open a new tab in your browser.

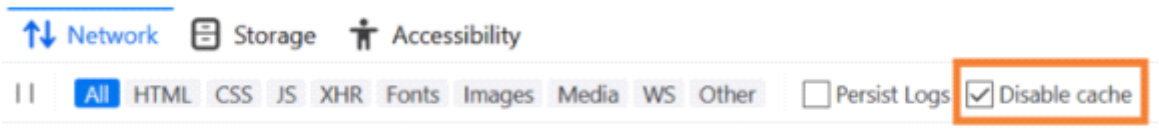
154. Open a new tab in your browser.

155. Before entering the CloudFront domain name, *Right Click* on the page and choose **Inspect**.



156. Choose the **Network** Tab.

157. Select the **Disable cache** option, to prevent your browser from caching the static objects.

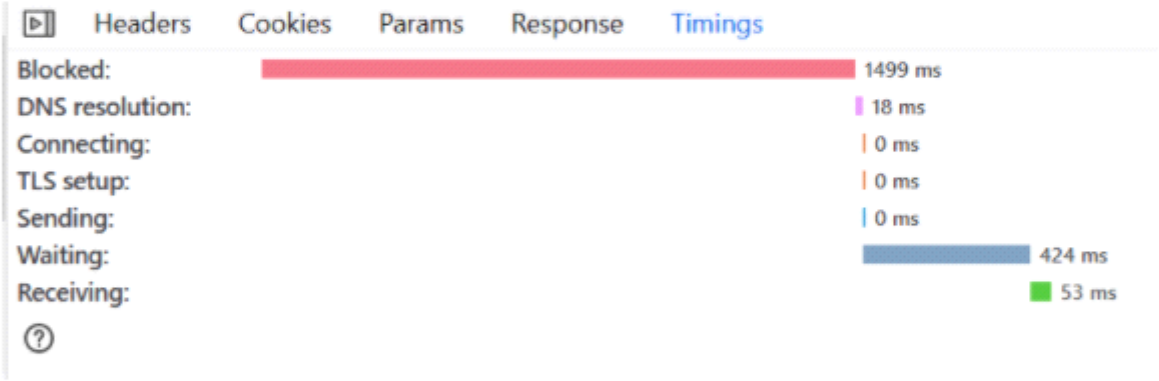


158. Enter your CloudFront Distribution domain name, noted down in a previous step, in the browser.

159. Select **APLogo.png** and select the **Header** tab.

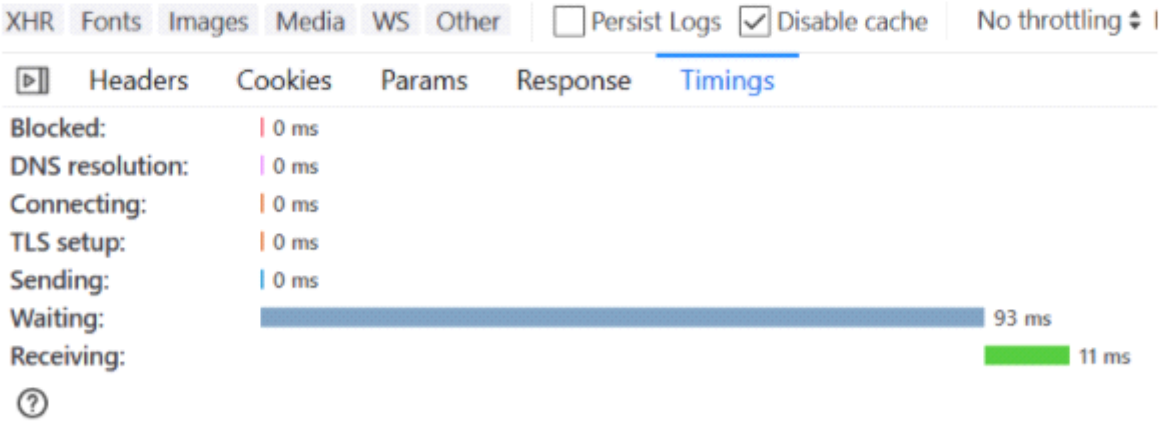
160. In the **Headers** menu, you should notice:

161. Select the **Timing** tab and note the timing details.



162. Reload the page.

163. Note the timing details. You should notice less latency, as the image is now cached in CloudFront.



164. Confirm that it was a Cache hit by returning to the **Header** tab.

To continue this lab, move on to Task 3:

https://aws.amazon.com/about-aws/whats-new/2021/05/amazon-vpc-announces-pricing-change-for-vpc-peering/