Lab 3: Right size Amazon EC2 instances using Amazon CloudWatch metrics v1.0.5

Sunday, August 9, 2020 7:18 PM

Lab 3:

- Between Steps 46 and 47 you need to select the Web-Server
- Task 7.1 is still written for the old EC 2 console. Use the steps from Task 5 to resize the DB-Server to an Instance Type of r5a.2xlarge.

Lab 3: Cost optimization: Right size Amazon EC2 instances using Amazon CloudWatch metrics

2 hours Free

aws training and certification

In this lab, you will identify EC2 instances incorrectly sized for their compute capacity by observing Amazon CloudWatch custom metrics. You will resolve this by changing the instance type and configuring CloudWatch alarms to monitor such recurrences in the future.

Objectives

After completing this lab, you will be able to:

After completing this lab, you will be able to:

- Create a resource group based on resource tags.
- Install and configure CloudWatch agent on EC2 instances.
- Troubleshoot EC2 instances using CloudWatch metrics.
- Right-size EC2 instances based on CloudWatch metrics.
- Configure CloudWatch alarms.

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.
- An Internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported)

Duration

This lab will require **60** minutes to complete.

Introduction

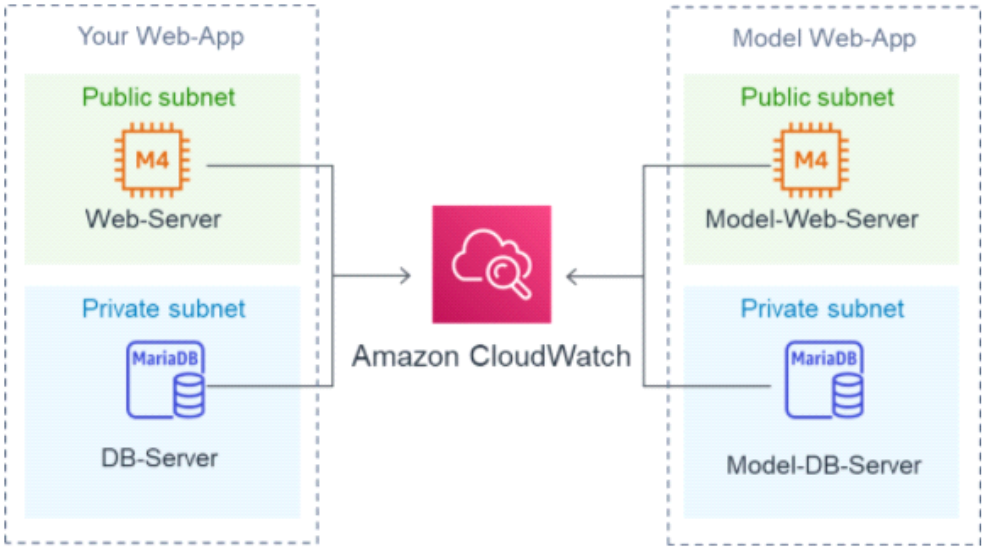
From your previous actions you are already beginning to find significant cost savings. You have now created an automated backup process for your Production environment that is leveraged to ensure that the test environments are in-sync with Production, ensuring operational efficiency. In addition to environmental consistently, you have also standardized instance types for applications that were part of your lift and shift.

Since the lift and shift, there has been a degradation in performance of one of your team's web applications. You were asked to look into the performance issues. Fortunately, another team instituted a set of best practices with a similar architecture for their web application. They offered their support and instance metrics to assist in your optimization efforts.

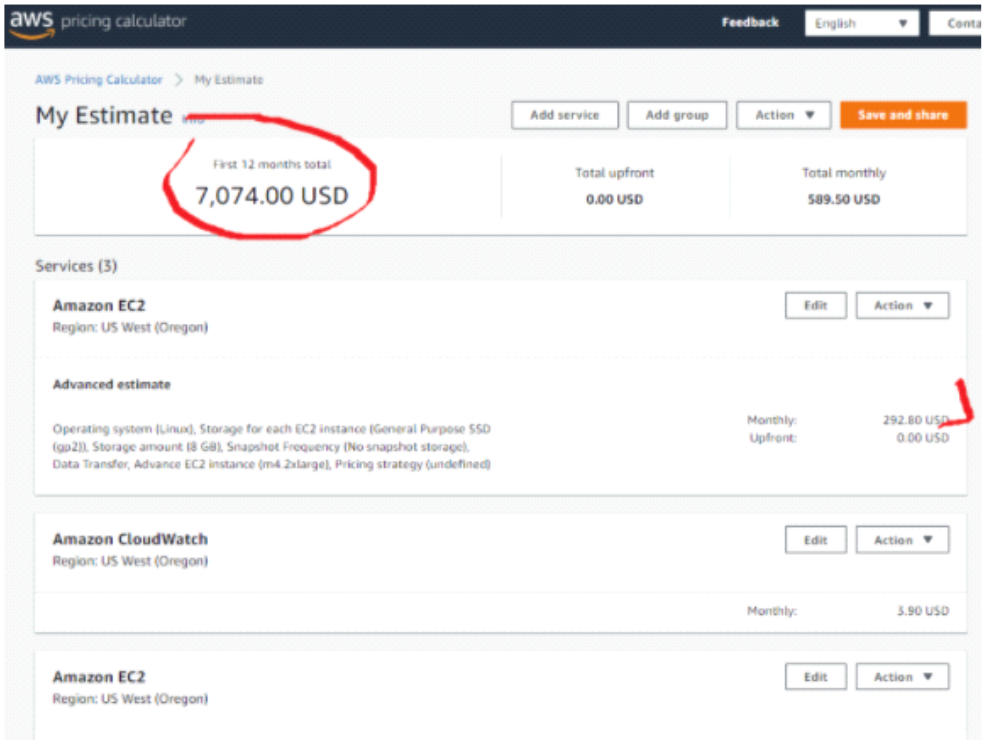
Through your initial discovery, it is clear that CPU and Memory utilization will be the key metrics that will drive your conclusions.

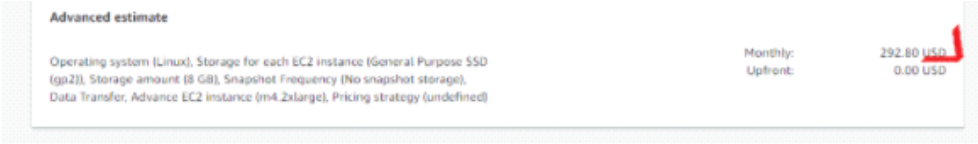
For the duration of this lab, you will be investigating and correcting the performance issues on your instances. You will also be using best practices from the model team.

At the start of this lab you will be provided with the following architecture:



Below is a screenshot of the AWS pricing calculator showing the cost of Your Web App environment at the start of the lab, not including Best Practice Web App. The prices are as of 5/28/2020. You can select the link below for an estimate online with current prices.





https://calculator.aws/#/estimate?id=651784e943c2de08f0f1b803f7020e885138147f

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

You must first log out before logging into a different AWS account.

To logout, click here

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

Task 1: Create your web application resource group

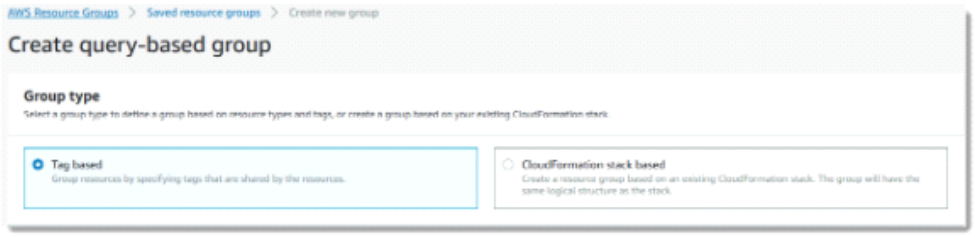
To begin, you will need to create a resource group so you can isolate your team's application instances. You will use this resource group to make concurrent changes to your instances as well as consolidate metrics.

Important: Make sure that you are using the Region suggested by your trainer for this lab.

4. On the navigation bar, choose Services, and choose **Resource Groups & Tag Editor**.

5. Select Create a resource group.

6. In **Group type**, choose the **Tag based** group type.



7. In **Grouping criteria**, choose the resource types that you want to be in your resource group. You can have a maximum of 20 resource types in a query. For this walkthrough, choose `AWS::EC2::Instance`

Task 2: Install and configure CloudWatch Agent on Web-Application-Instances resource group

Through discussions with the model team, you already know that memory utilization will be a key metric in diagnosing performance issues within the web application. However, memory utilization is not a metric that CloudWatch can gather without a CloudWatch Agent (CWA) installed directly on the instances. The CWA can be used to gather statistics and metrics for servers on-premises and in most clouds, making it a valuable tool for hybrid

10. Under **Group details**, enter `Web-Application-Instances` as the **Group Name**.

11. When you are finished, choose **Create group**

    You should find a success message across the top of your screen, indicating the group was created successfully:

`SystemsManager` and select **Systems Manager**.

13. In the navigation panel on the left, choose **Run Command**.

14. Choose **Run a Command**

15. In the Command document list, choose `AWS-ConfigureAWSPackage` radio button.

    Note: *If you navigate into the document, you will need to select* **Run Command**

16. Under Command parameters:

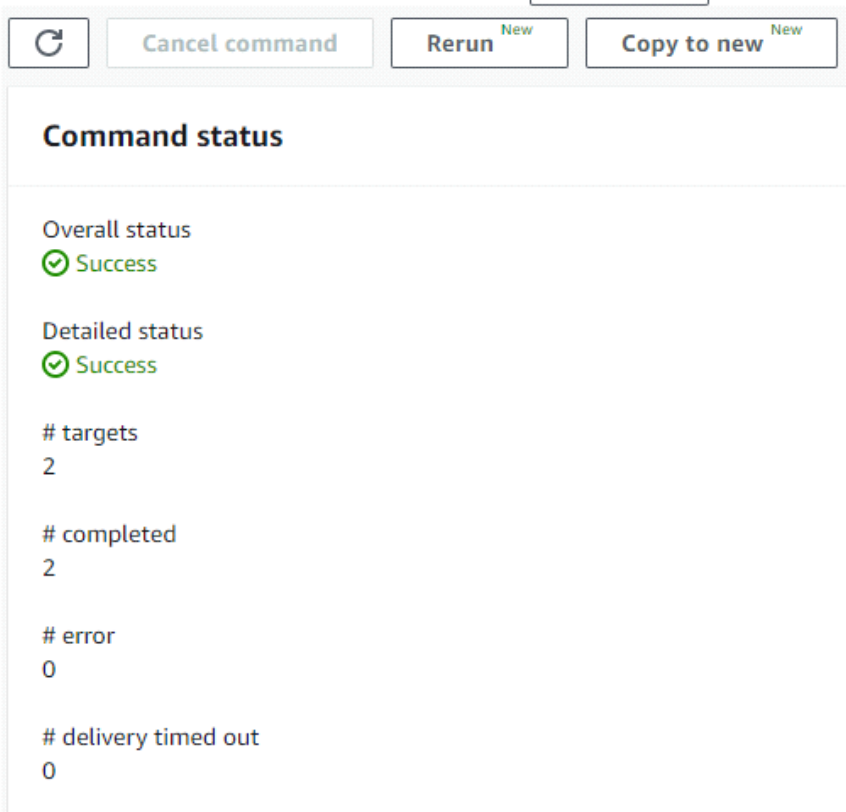
- For **Action** List, choose **Install**.
- For **Installation Type** list, choose **Uninstall and reinstall**.
- For **Name field**, enter `AmazonCloudWatchAgent`

17. In the Targets area, select the **Choose a resource group** radio button. **Web-Application-Instances** should autofill, as it is currently your only resource group. (If not, in the Resource Group list, choose Web-Application-Instances)

18. Within the **Output options** panel deselect the **Enable writing to an S3 bucket** check box.

19. Choose **Run** and verify an **Overall Status** of *Success* upon completion.

    Note: This may take up to a minute, refresh page for results. If there was one or more failures, you can view the details by selecting the instance from within **Targets and outputs** and selecting **View Output**



2.2: Configure and Start the CloudWatch Agents on Web-Application-Instances Resource Group

20. In the navigation panel on the left, once again choose **Run Command** to return to the Run Command menu.

21. Choose **Run Command**

22. In the Command document list, search for and choose the `AmazonCloudWatch-ManageAgent` radio button.

23. Under Command parameters:

24. For the **Action** list, choose **configure**.

25. For the **Optional Configuration Source** list, choose **ssm**.

26. For the **Optional Configuration Location** input box, enter `AmazonCloudWatch-AgentConfig`

Note: This configuration file (posted below) has been created on your behalf for this lab. This Json file specifies the metrics and logs that the agent is meant to collect, including custom metrics.

```
{
   "agent":{
      "metrics_collection_interval":15,
      "run_as_user":"root"
   },
   "metrics":{
      "namespace":"MemoryUsage",
      "append_dimensions":{
         "InstanceId":"${aws:InstanceId}"
      },
      "metrics_collected":{
         "mem":{
            "measurement":[
               "mem_used_percent"
            ],
```

```
            "metrics_collection_interval":15
        }
    }
  }
}
```

24. In the **Optional Restart** list, choose **yes**. This will start the agent after you have finished these steps.

25. In the Targets area, select the `Choose a resource group` radio button. **Web-Application-Instances** should autofill, as it is currently your only resource group.

26. Within the **Output options** panel deselect the **Enable writing to an S3 bucket** check box.

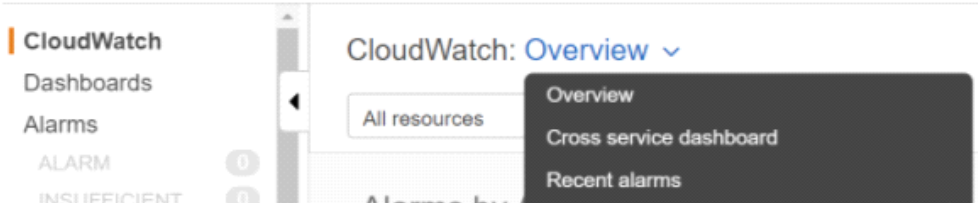
27. Choose **Run** and verify an **Overall Status** of *Success* upon completion.

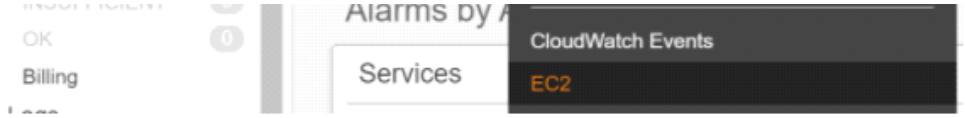
Task 3: Analyze CPU metrics

Now that you have installed and configured the CWA on your **Web-Application-Instances** resource group, it is time to examine the performance metrics. The memory metrics will take some time to populate, so you should start with CPU utilization.

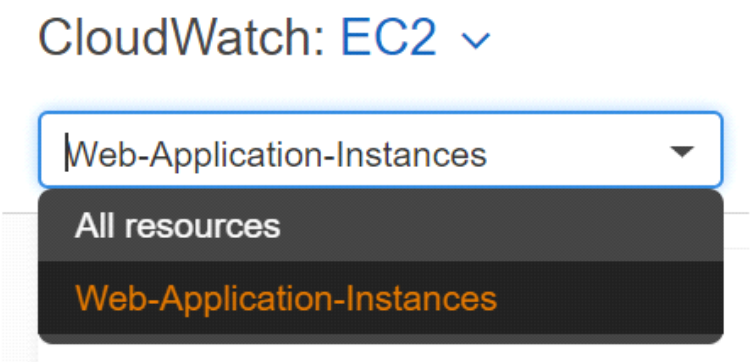
28. In the **AWS Console**, on the **Services** menu, enter `CloudWatch` and select **CloudWatch**.

29. Select **EC2** from the Overview list.





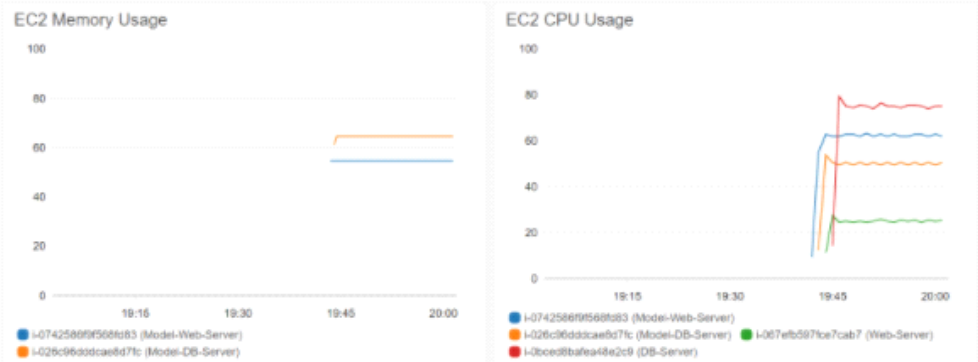
30. You will find various performance metrics for all EC2 instances within your region. In order to narrow the scope and focus to the Lift and Shift instances, from the **All resources** drop down menu select **Web-Application-Instances**.



31. Take a moment to explore the metrics available for your EC2 Instances.

32. In order to leverage the model team's help, you should go ahead an compare your instance's metrics to the model team's. This can be done by navigating to **Dashboards** in the navigation panel on the left.

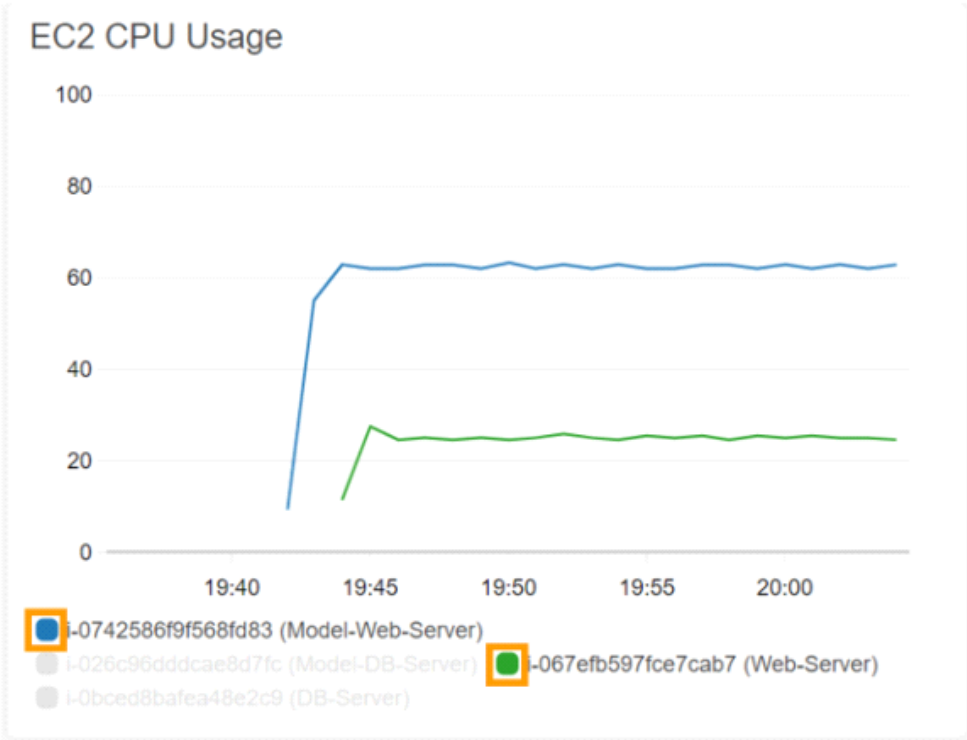
33. Select the **EC2_Metric_Comparison** dashboard from the Dashboard list. This dashboard was created ahead of time for this lab.



Note: You have 2 graphs, however the memory usage for your web application instances are missing. This is because when the Dashboard was created, the CloudWatch Agent was not installed, thus memory utilization metrics were not available. Now that you have installed the CloudWatch Agent, you will be able add your web application instances to this metric widget (You will do this in the next Task).

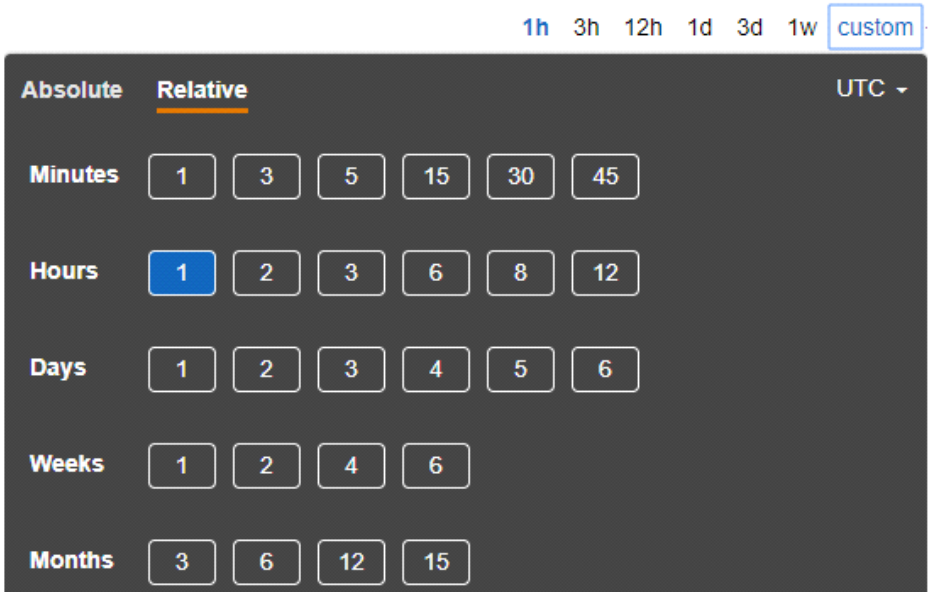
34. In order to help isolate web instances from database instances, consider selecting instances with the same function from both teams to isolate the information in the graph: **Model-Web-Server** and **Web-Server**.

35. Select the colored square icon next to the instances in which you would like to select or de-select



Likewise, you can compare the two DB instances **Model-DB-Server** and **DB-Server**.

Note: By adjusting the **Time Period** settings in the top right corner, you can zoom in on more granular data points.



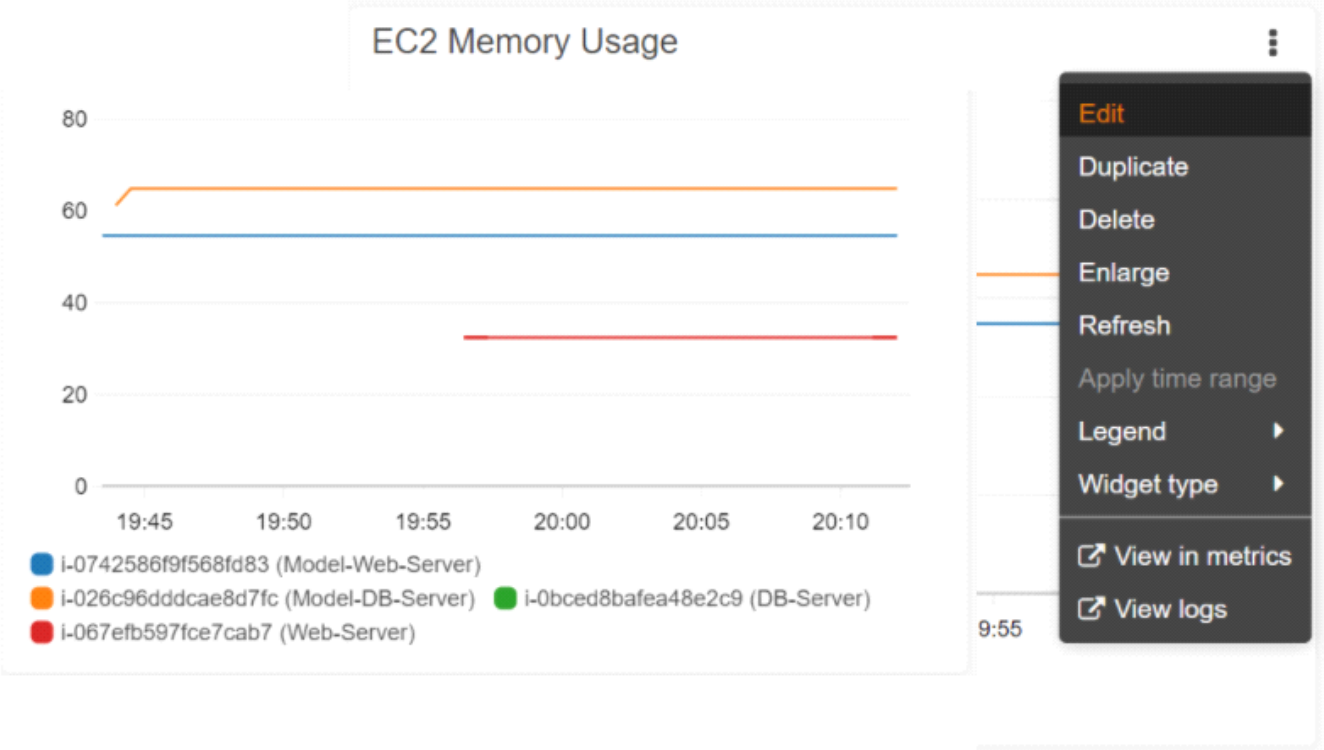
By comparing CPU utilization for the Model-Web-Server and the Web-Server you will notice that the Web-Server instance is over provisioned. The DB-Server, however, seems to be operating efficiently, with respect to CPU.

Task 4: Analyze memory metrics

Considering that neither the Web-Server nor DB-Server's CPU metrics were high enough to cause performance issues, you must now analyze memory utilization. This can be accomplished by adding their memory metrics to the dashboard.

36. Choose the ⋮ that appears in the top right corner when you hover your mouse over the EC2 Memory Usage Widget.

37. Select **Edit**



ıvailable.

lect **Memory Usage**.

Task 5: Resizing of the Web-Server

You should now resize your Web-Server instance to an **m4.xlarge**, as both CPU and memory usage were extremely low. While this will not necessarily fix the performance issues, it help drive down cost and optimize your instance.

Note: *In order to remedy the memory issues on the DB-Server instance, you will need to change the instance type to an r5a.2xlarge instance that is memory optimized. This may take an approval to add to the Approved Instance type list in Config (Lab 1), so in the mean time you will optimize the Web-Server instance. You will tackle the database instance in the optional task at the end of this lab.*

NOTE: Please use the **New EC2 Experience** on the left hand navigation pane.

44. In the **AWS Console**, on the **Services** menu, enter `EC2` and select **EC2**.
45. In the navigation panel on the left, choose **Instances**.
46. Choose **Instance state**
47. Select **Stop instance**
48. Confirm by choosing on **Stop**
49. Choose **Actions**
50. Choose **Instance Settings** in the dropdown.
51. Select **Change Instance Type**.
52. Set the Instance Type to **m4.xlarge**.
53. Choose **Apply**
54. Choose **Instance state**
55. Select **Start instance**

Task 6: Create CPU utilization alarms

Now that you have reduced the instance size of the Web-Server, it is important to monitor the CPU utilization metric to ensure that the instance does not become over or under utilized in the future. To facilitate this, you will now set up a CloudWatch Alarm to monitor CPU utilization .

6.1: Alarm for CPU over utilization

56. In the **AWS Console**, on the **Services** menu, enter `CloudWatch` and select **CloudWatch**.

57. In the navigation panel on the left, choose **Alarms**.

58. Select **Create alarm**

59. Choose **Select metric**

60. Within the **AWS Namespace**, select **EC2** .

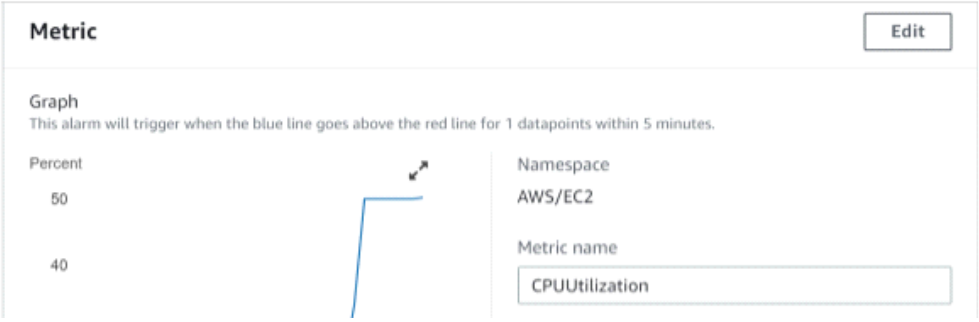
61. Select **Per-Instance Metrics**.

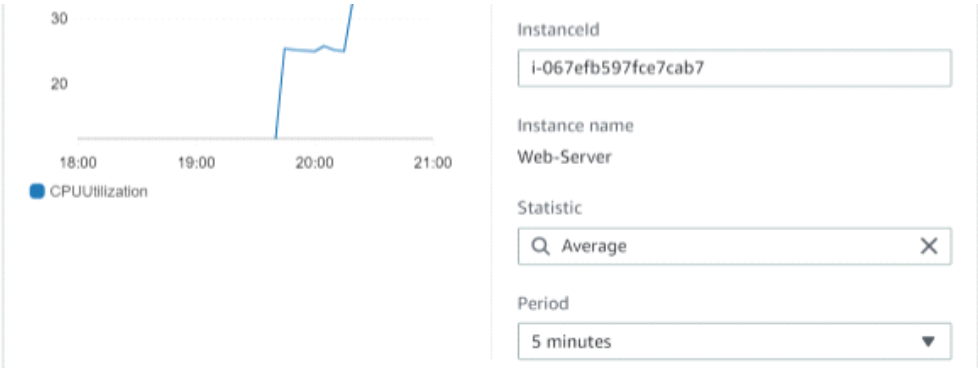
62. Search for `CPUUtilization`

63. Select ☑ **CPU Utilization** for the **Web-Server** instance.

64. Select **Select metric**

Note You will notice that the graph will now show CPU utilization comparable to Model-Web-Server. This can be confirmed by revisiting the dashboard from Task 3 - Step 5.





65. Within the **Conditions** menu, under **Whenever CPUUtilization is...** select the **Greater/Equal** radio button and set the **threshold value** to `75`

66. Select **Next**

67. Select **Remove** within the **Notification** menu.

    Note: This can be used to notify your team, but for this lab you will remove it.

68. Select **Next**

69. For **Alarm name** enter `Over Utilized CPU: Web-Server`

70. Select **Next**

71. Review the settings and select **Create alarm**

6.2: Alarm for CPU under utilization

Although CPU under utilization is not necessarily a concern regarding application performance, it is a good idea to create an alert in order to quickly identify over provisioned instances.

72. Select **Create alarm**

73. Choose **Select metric**

74. Within the **AWS Namespace**, select **EC2**.

75. Select **Per-Instance Metrics**.

76. Search for `CPUUtilization`

77. Select ☑ **CPU Utilization** for the **Web-Server** instance.

78. Select **Select metric**

79. Within the Conditions menu, under **Whenever CPUUtilization is...** select the **Lower/Equal** radio button and set the **threshold value** to `25`

80. Select **Next**

81. Select **Remove** within the Notification menu.

82. Select **Next**

83. For **Alarm name** enter `Under Utilized CPU: Web-Server`

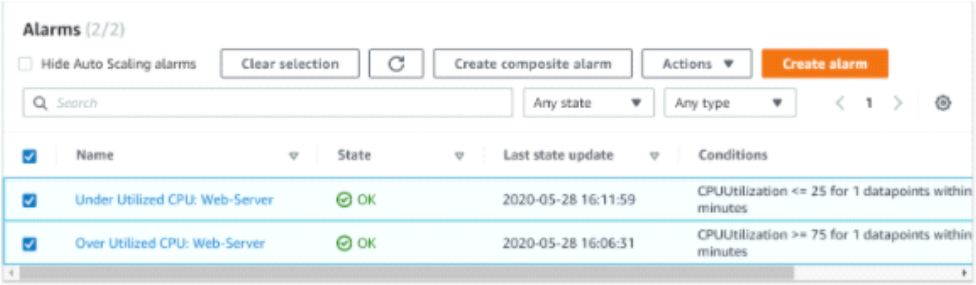
84. Select **Next**

85. Review the settings and select **Create alarm**

6.3: Add alarms to your dashboard

Now that you have created two alarms, you should now add them to your dashboard in order to begin developing a single view into your application's health.

86. Select the ☑ for both alarms you have just created.



87. In the **Actions** list, select **Add to dashboard**.

88. From the dashboard dropdown list, choose **EC2_Metric_Comparison**.

89. Select **Add to dashboard**

90. Navigate to your dashboard by selecting **View dashboard** located in the success notification bar at the top of the screen.

91. Select **Save dashboard** at the top of the dashboard.

    Note: In the future, consider adding an action to take place when an alarm is triggered (ie. Send a note to System Administrator or scale your application out with an AutoScaling group).

(Optional/Challenge) Task 7: Optimize DB-Server instance type and create a composite alarm

Assume you have received the proper approvals to resize the **DB-Server** instance to an *r5a.2xlarge* memory optimized instance. Once the instance has been updated, you will set up an alarm that will monitor memory usage in the database instance.

7.1: Remediation of the DB-Server

92. In the **AWS Console**, on the **Services** menu, enter `EC2` and select **EC2**.

93. In the navigation panel on the left, choose **Instances**.

94. Select the **DB-Server** instance and choose *Actions -> Instance State -> Stop*.

95. In the confirmation dialog box, choose **Yes, Stop** . It can take a few minutes for the instance to stop.

96. Once the instance has stopped choose *Actions -> Instance Settings -> Change Instance Type*.

    Note: This action is disabled if the instance state is not stopped.

97. For **Instance Type**, select the **r5a.2xlarge** instance type. Choose **Apply** to accept the new settings.

98. To restart the stopped instance, select the instance and choose *Actions -> Instance State -> Start*.

99. In the confirmation dialog box, choose **Yes, Start**. It can take a few minutes for the instance to enter the running state.

7.2: Memory alarm for DB-Server

100. In the **AWS Console**, on the **Services** menu, enter `CloudWatch` and select **CloudWatch**.

101. In the navigation panel on the left, choose **Alarms**.

102. Select **Create alarm**

103. Choose **Select metric**

104. Within the **Custom Namespace**, select **MemoryUsage**.

105. Select **InstanceId**.

106. Select ☑ **DB-Server**.

107. Select **Select metric**

108. Within the Conditions menu, under **Whenever mem_used_percent is...** select the **Greater/Equal** radio button and set the **threshold value** to `75`

109. Select **Next**

110. Select **Remove** within the Notification menu.

111. Select **Next**

112. Enter `Over Utilized Memory: DB-Server` as the **Alarm name**.

113. Select **Next**

114. Review the settings and select **Create alarm**

7.3: Create a composite alarm

In Task 6 you set up two alarms that monitor the Web-Server's CPU utilization. Now, you will create a composite alarm that combines the memory utilization alarm you just created for the DB-Server and the CPU over utilization alarm you previously created for the Web-Server. This will allow you to monitor performance issues across multiple instances with a single alarm.

115. Select the ☑ next to **Over Utilized Memory: DB-Server** and **Over Utilized CPU: Web-Server** alarms.

116. Select **Create composite alarm**

117. Within the **Conditions** menu you should have two statements pre-populated:

118. Select **Next**

119. Select **Remove** within the Notification menu.

120. Select **Next**

121. Enter `Web Application: Over Utilized` as the alarm name.

122. Select **Next**

123. Select **Create composite alarm**

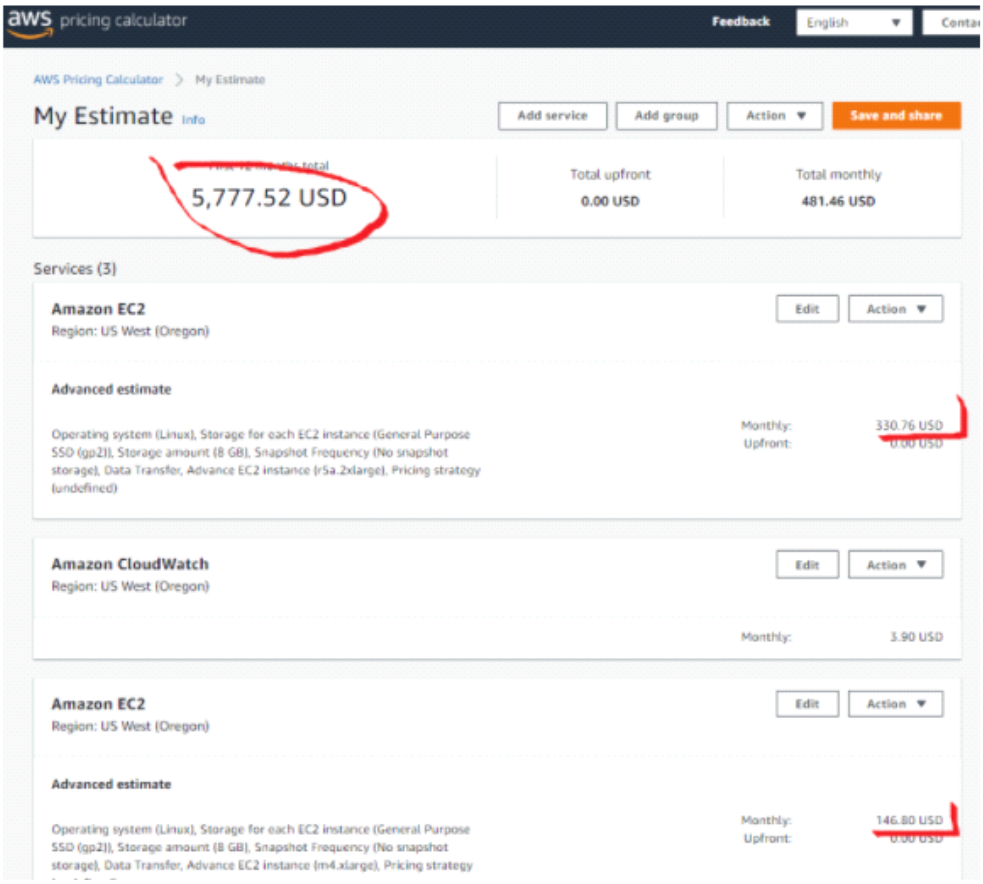
124. To view the new alarm, select the name of your newly created composite alarm **Web Application: Over Utilized**. Here you can find details on the composite alarm, including the individual alarms that were used to create it.

If you wish to take this exercise a step further, you could create a memory over utilization alarm for Web-Server and a CPU over utilization alarm for your DB-Server. Add them to your recently created composite alarm, ensuring you can monitor your web application's performance through a single composite alarm.

Summary

In this lab you have saved costs by right sizing your instances. In practice, you can take advantage of tools like AWS Compute Optimizer to help, however it will be an on-going process that is important to re-visit continuously.

Below is a screenshot of the AWS pricing calculator showing the cost of Your Web App environment after implementing the optimizations in this lab. The prices are as of 5/28/2020. You can select the link below for an estimate online with current prices.



(undefined)

https://calculator.aws/#/estimate?id=f2bd7203624356f04b852e73afb8998dd1288d71

End Lab

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

125. Return to the AWS Management Console.

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

127. Choose **End Lab**

128. Choose **OK**

129. (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*.