EDx CC-615 Cloud Infrastructure

Week 7

Cloud Infrastructure – Lab Exercise: Monitoring IaaS cloud deployment

Objectives
In week 7, we will learn more about the operation of our of cloud IaaS deployment created in Week 6 by monitoring it using AWS CloudWatch. We will:

- Create the AWS CloudWatch dashboard
- Set up the dashboard for BallotOnline web server deployment services by adding widgets to it to watch load balancer and auto scale group metrics
- Create the AWS CloudWatch billing dashboard for BallotOnline deployment
- Test the dashboard by causing errors and outages and monitoring them
- Stop and terminate your AWS deployment infrastructure

All students are encouraged to complete the labs. At the end of the week, Audit students will take the weekly exam assessing their understanding of the labs. Verified students will submit Lab Report including the material requested in the lab steps.

General Lab Instructions

- You have to complete Week 6 lab before proceeding with Week 7 lab because the latter uses the deployment from the former.
- Note that you are responsible for any charges incurred using AWS for this lab. You were provided a credit in edX CC 607, which you may use for this course as well. If you have depleted the Amazon credits, then you will need to provide appropriate payment details.
- Each time that you connect to the AWS console, you need to ensure that you are connecting to the region where your resources are created. If you do not see your resources, then you are in the wrong region and need to go to the Global menu at the top right of your screen to change the region. Example: If you created your instances in US-West, then you need to ensure that US-West is selected after you log into the console.

Lab Report Requirements

You are required to submit a lab report at the end of the lab. Throughout the steps, you will see indicators labeled with **LAB REPORT:** in which you should incorporate an item from that step into your report. Examples include screenshots of your progress, URLs for running AWS instances, and summaries of your progress.

1. **Learning Objective: Create the AWS CloudWatch Service Dashboard**

First, you will create the CloudWatch dashboard in your AWS account. It will allow you to monitor your deployments.

1. In a web browser, open the AWS CloudWatch Console: https://console.aws.amazon.com/cloudwatch/

2. If you are in the incorrect region, use the navigation bar to change the region to the region where your AWS resources were located.

3. Enter your AWS login credentials into the boxes provided and click "Sign In."

4. Click the "Dashboards" menu on the left.

5. Click "Create Dashboard."

6. For the new dashboard, enter the following as the dashboard name:

BallotOnline-Dashboard

7. Click "Create Dashboard."

8. Select "Cancel" on the "Add to this dashboard" screen.

LAB REPORT: Write a one-paragraph summary of your lessons learned while completing this portion of the lab.

Congratulations! You have completed the steps for the first part of this lab.

2. **Learning Objective: Set up the dashboard for BallotOnline web server deployment**

Next, you will customize your dashboard to monitor the load balancer and auto scale group metrics in your deployment.

1. *Learning Topic: Add Load Balancer numeric widgets to the CloudWatch dashboard*

    1. In a web browser, open the AWS CloudWatch Console: https://console.aws.amazon.com/cloudwatch/

2. If you are in the incorrect region, use the navigation bar to change the region to the region where your AWS resources were located.

3. Enter your AWS login credentials into the boxes provided and click "Sign In."

4. Click the “Dashboards” menu on the left.

5. Click on the “BallotOnline-Dashboard” link.

6. Click on “Add widget.”

7. On the “Add to this dashboard” screen, select “Number” and then click “Configure.”

8. At the top of the screen, select “1d”

9. At the top left of the screen, click on the “pencil icon” next to “Untitled graph” and type the following:

BallotOnline Load Balancer Stats

   Click the “checkmark icon” to save the change.

10. On the “Add metric graph” screen, you will be adding several metrics. The first metric that you will add is the Elastic Load Balancer.

11. Click on the “ELB” link T the bottom of the screen.

12. Click on “Per-LB Metrics.”

13. You will need to select the checkbox for the following metrics:

Metric	**Purpose**
HealthyHostCount	Displays the number of healthy EC2 instances in the load balancer pool
EstimatedALBActiveConnectionCount	Displays the number of active connections to the load balancer
UnHealthyHostCount	Displays the number of unhealthy EC2 instances in the load balancer pool
HTTPCode_ELB_5XX	Displays the number of HTTP 5xx codes returned by the load balancer
HTTPCode_Backend_4XX	Displays the number of HTTP 4xx codes returned by the load balancer

14. Click “Create Widget.”

2. *Learning Topic: Add Load Balancer graph widgets to the CloudWatch dashboard*

15. Click on the **“**BallotOnline-Dashboard**”** link.

16. Click on “Add Widget.”

17. On the “Add to this dashboard**”** screen, select “Line” and then click “Configure”

18. At the top of the screen, select “1d.”

19. At the top left of the screen, click on the “pencil icon” next to “Untitled graph**”** and type the following:

BallotOnline Load Balancer Graph

    Click the “checkmark icon**”** to save the change.

20. On the “Add metric graph” screen, you will be adding several metrics. The next metric that you will add is the Elastic Load Balancer.

21. Click on the “ELB**”** link.

22. Click on “Per-LB Metrics.”

23. You will need to select the checkbox for the following metrics:

Metric	**Purpose**
HealthyHostCount	Displays the number of healthy EC2 instances in the load balancer pool
EstimatedALBActiveConnectionCount	Displays the number of active connections to the load balancer
UnHealthyHostCount	Displays the number of unhealthy EC2 instances in the load balancer pool
HTTPCode_ELB_5XX	Displays the number of HTTP 5xx codes returned by the load balancer
HTTPCode_Backend_4XX	Displays the number of HTTP 4xx codes returned by the load balancer

24. Click “Create Widget.”

25. On the "Dashboard" screen, select the bottom right corner of that graph that you just created and drag to enlarge until the graph is the same size as the stats window above.

3. *Learning Topic: Add Auto Scale Group numeric widgets to the CloudWatch dashboard*

26. Click "Add Widget."

27. On the "Add to this dashboard" screen, select "Number" and then click "Configure."

28. At the top of the screen, select "1d."

29. At the top left of the screen, click on the "pencil icon" next to "Untitled graph" and type the following:

BallotOnline Auto-Scaling Group Metrics

Click the "checkmark icon" to save the change.

30. On the "Add metric graph" screen, you will be adding several metrics. The next metric that you will add is the Auto Scaling Group.

31. Click on the "EC2" link.

32. Click on "By Auto Scaling Group."

33. You will need to select the checkbox for the following metrics:

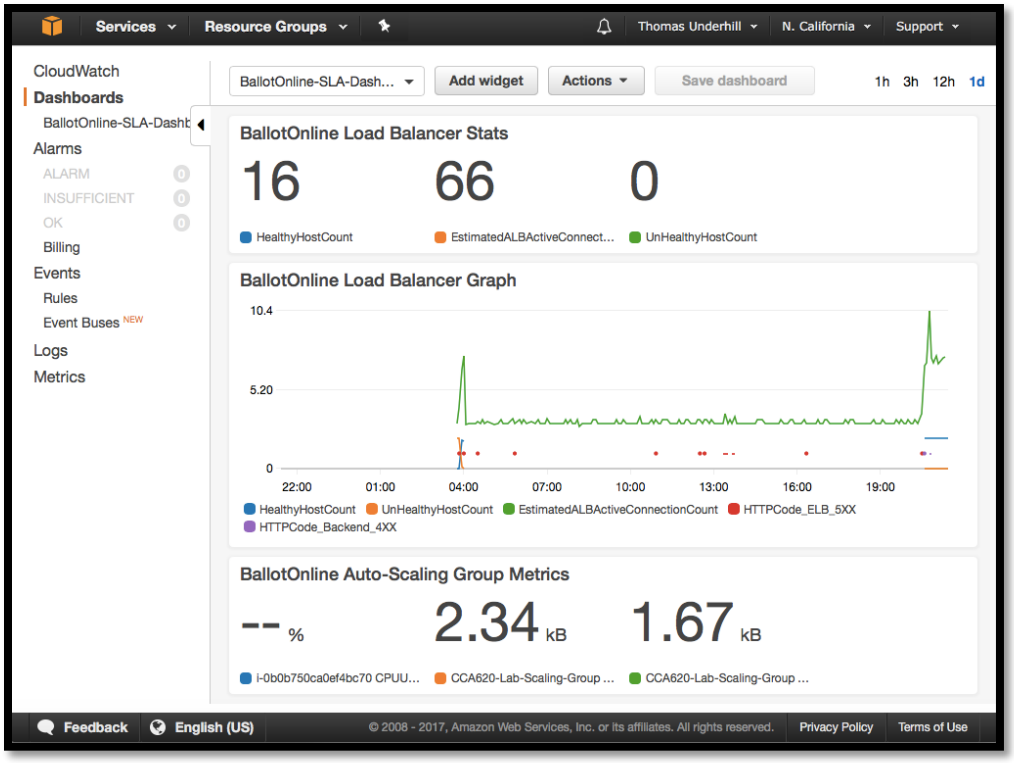
Metric	**Purpose**
NetworkOut	Displays the outbound network traffic
NetworkIn	Displays the inbound network traffic
CPU Utilization	Displays the CPU utilization for the entire auto-scaling group

34. Click "Create Widget."

35. Click on "BallotOnline-Dashboard."

36. Click "Save Dashboard."

37. Click "1d" on the upper right.



LAB REPORT: Capture screenshots of your dashboard, showing all of the metrics that you are capturing. Insert these screenshots into your lab report, and write a two- to three-paragraph summary of your activities during this portion of the lab, any lessons learned, and any areas for future experimentation.

Congratulations! You have successfully created a service dashboard for BallotOnline. Now, you will create a billing dashboard.

3. Learning Objective: Create the AWS CloudWatch Billing Dashboard

In this part of the lab, you will set up the CloudWatch Billing dashboard for BallotOnline to enable management to monitor costs.

1. In a web browser, go to the AWS Billing Management Console: https://console.aws.amazon.com/billing/home#/

2. Click on “Preferences” on the left menu.

3. Check the box for “Receive Billing Alerts.”

4. Click “Save Preferences.’

5. In a web browser, open the AWS CloudWatch Console: https://console.aws.amazon.com/cloudwatch/

6. Enter your AWS login credentials into the boxes provided and click "Sign In."

7. Select the US-East (N. Virginia) region. **NOTE: This is the only region where billing dashboards are available.**

8. Click on “Dashboards” on the left menu.

9. Click “Create Dashboard.”

10. For dashboard name, enter the following:

BallotOnline-Billing-Dashboard

    Click on the “Create Dashboard” link.

11. Under the “Add to this dashboard” screen, click “Cancel.”

12. On the left menu, select “Metrics.”

13. In the right pane, in the All Metrics tab, click the “Billing” link.

14. At the top of the screen, select “1d.”

15. At the top left of the screen, click on the “pencil icon” next to “Untitled graph” and type the following:

BallotOnline Total Estimated Charges

Click the “checkmark icon” to save the change.

16. At the top right of the screen, select “Number.”

17. Select “Total Estimated Charge.”

18. Check the box for “Estimated Charges.”

19. Select the “All Metrics” tab.

20. Click on the “Billing” link.

21. Click on the “By Service” metrics.

22. Select “AmazonEC2, Amazon CloudWatch, and AWS Data Transfer.”

23. Click the “Actions” menu, and then select “Add to Dashboard.”

24. On the “Add to dashboard” screen, select the “BallotOnline-Billing-Dashboard” and then select the “Number” widget type and click “Add to dashboard.’

25. On the *Dashboards screen*, click “Save Dashboard.”

LAB REPORT: Capture screenshots of your billing dashboard, showing all of the metrics that you are capturing. Insert these screenshots into your lab report, and write a two- to three-paragraph summary of your activities during this portion of the lab, any lessons learned, and explain how this dashboard can be used to help management control cloud hosting costs.

Congratulations! You have successfully created a billing dashboard for BallotOnline.

4. Learning Objective: Test the dashboard by causing errors and outages and monitoring them

In this part of the lab, you will create site outages and errors and monitor using the dashboard that you created.

1. Learning Topic: Monitor Web Server errors

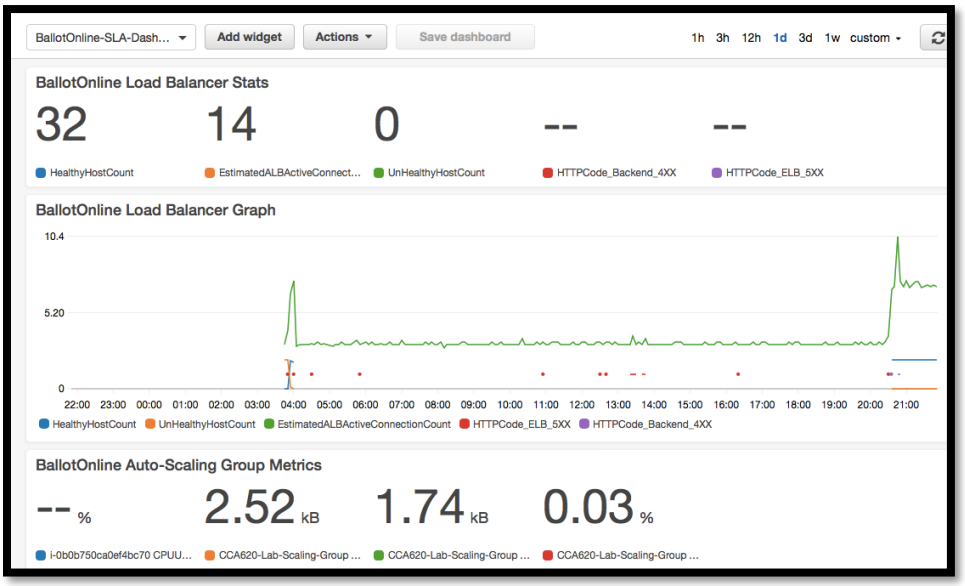
1. In a web browser, open the AWS CloudWatch Console: https://console.aws.amazon.com/cloudwatch/

2. If you are in the incorrect region, use the navigation bar to change the region to the region where your AWS resources were located.

3. Enter your AWS login credentials into the boxes provided and click "Sign In."

4. Click the “Dashboards” menu on the left.

5. Click on the “BallotOnline-Dashboard” link, and you should see a screen similar to the one below.



LAB REPORT: Take a screenshot of your dashboard before proceeding further, and insert it into your lab report. You should also write a paragraph explaining how you expect your dashboard to be affected by the activities in this lab.

6. Go to the BallotOnline New Web Service URL from your Project 3 Lab Report.

Example: http://CC615-LB-1234567890.us-wes-1.elb.amazonaws.com

7. Now, you are going to append a nonexistent web page to the end of the BallotOnline New Web Service URL from the previous step to generate traffic and errors.

Example: http://CC615-LB-1234567890.us-wes-1.elb.amazonaws.com/this-page-does-not-exist.html

LAB REPORT: Take a screenshot of the results that you receive in your browser and insert it into your lab report.

8. Monitor your dashboard, while you refresh your browser window for the above nonexistent webpage *20* times.
9. On your dashboard window, click on the *refresh* icon. 

You should see the following types of changes in your dashboard:

- EstimatedALBActiveConnectionCount should substantially increase.
- HTTPCode_Backend_4XX count should increase.

You may have to wait a few minutes before you see all changes in the dashboard. You may also experiment with the time range and refresh intervals (in the upper right screen corner), or edit your dashboard (by selecting Edit on the three-dot dropdown list in the upper right corner on the dashboard) and change the Statistics column of the metrics from Average to Sum, to see accumulated changes.

LAB REPORT: Take a screenshot of the dashboard and insert it into your lab report. Write a paragraph explaining why the dashboard changed and provide some examples of other activities that could affect the dashboard.

2. *Learning Topic: Monitor Web Server Outages*

10. Run this command to check that *nginx* is running on each instance:

```
netstat -an | grep 80 | grep LISTEN
```

You should see the following output:

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
```

11. Stop the **nginx** server on one of your instances with:

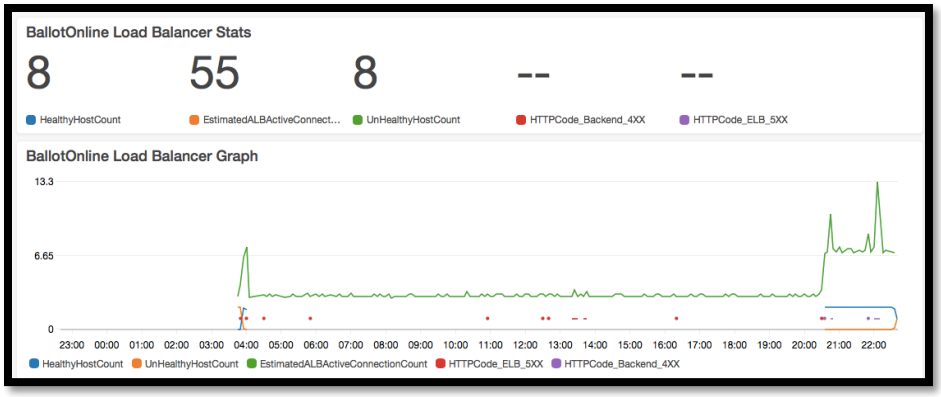
```
sudo nginx -s stop
```

12. Run this command to check that *nginx* is down and is no longer listening:

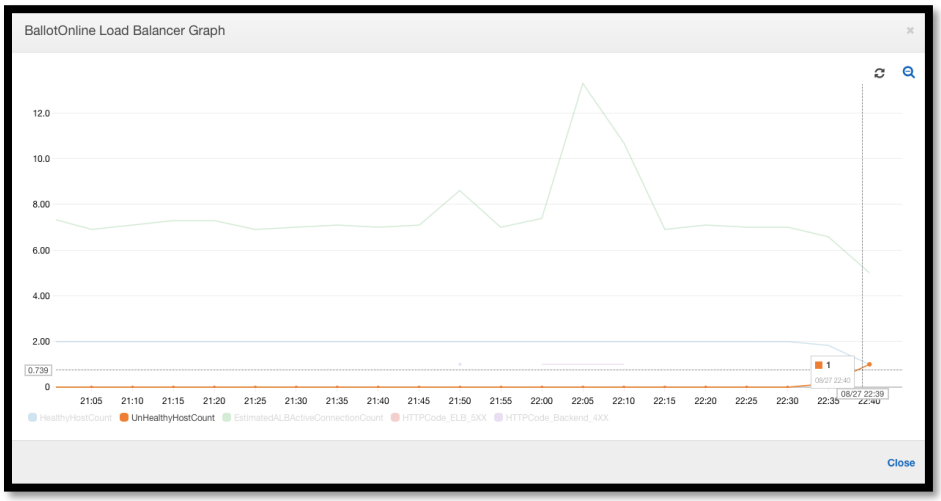
```
netstat -an | grep 80 | grep LISTEN
```

You should not see any output from the above command.

13. Go to your Dashboard, and you should see "UnHealthyHostCount" increasing, as in the screenshot below:



Use the mouse to zoom in on the graph to look at it in more detail:



14. Now, you are going to go to the BallotOnline New Web Service URL from the previous step to confirm that the site is still available with one instance down.

Example: http://CC615-LB-1234567890.us-wes-1.elb.amazonaws.com/index.html

If you can successfully connect to the BallotOnline website, then your load balancer is working properly.

LAB REPORT: Take a screenshot of your dashboard and the BallotOnline website (including the URL), and insert it into your report along with your observations during this portion of the lab.

15. Now, you will take down your other instance to generate a complete failure scenario.

    16. Run this command to check that *nginx is* running on each instance:

```
netstat -an | grep 80 | grep LISTEN
```

You should see the following output:

```
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
```

17. Stop the **nginx** server on one of your instances with:

```
sudo nginx -s stop
```

18. Run this command to check that **nginx** is down and is no longer listening:

```
netstat -an | grep 80 | grep LISTEN
```

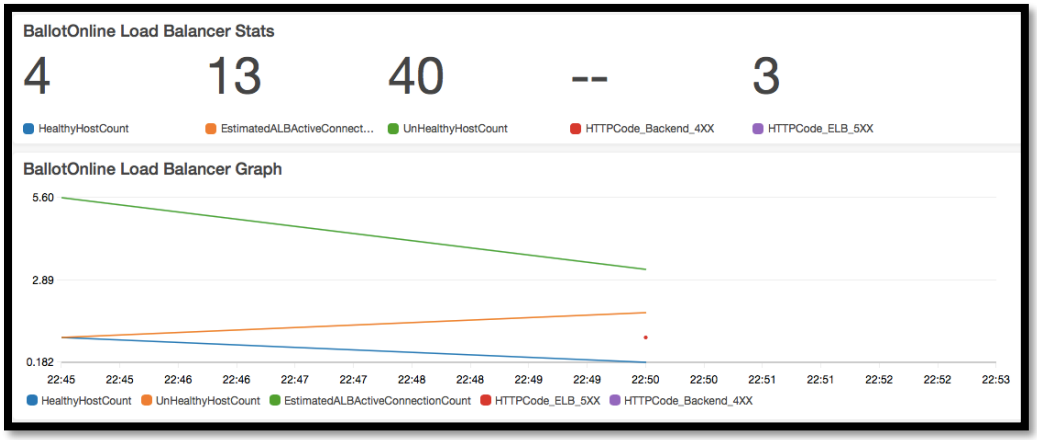
You should not see any output from the above command.

19. Now, you are going to go to the BallotOnline New Web Service URL from the previous step to confirm that the site is no longer available with both instances down.

Example: http://CC615-LB-1234567890.us-wes-1.elb.amazonaws.com/index.html

If you can no longer connect to the BallotOnline website, then you have successfully caused a complete outage of the site by shutting down the service on both instances.

20. Go to your Dashboard and select "custom" from the menu at the upper right. Select "1 Minutes."



You should see the *UnhealthyHostCoun***t** equal to the number of instances that you have, and the *HealthyHostCount as 0.*

It is important that you always have instances available behind your load balancer. For this lab, we only had two instances, so you can lose one of them and still maintain availability of the service.

> **LAB REPORT:** Take a screenshot of your dashboard, and analyze what actions you could take to minimize the likelihood of outages and ensure compliance with your business requirements. Your analysis should be at least two paragraphs in length.

Now that you've successfully caused a service outage, and seen what it can do to your site, you can **stop and terminate your AWS infrastructure. Remember: Billing continues until you terminate your infrastructure.**

5. Learning Topic: Stop and terminate your AWS deployment infrastructure

WARNING: Do not perform the steps in this section until you are finished with both Project 3 and Project 4 labs and have submitted all of your lab reports.

1. Go to the CloudWatch Console: https://console.aws.amazon.com/cloudwatch/
2. Click on "Dashboards."
3. Click on each of your dashboards, then click "Actions: Delete Dashboard", and then confirm with "Delete Dashboard". Repeat for both of your dashboards.
4. Go to the "EC2" console by selecting "EC2" from the "Services" menu at the top of your window.
5. Click on "Auto Scaling Groups" on the left.
6. Select your "CC615-Lab-Scaling-Group" auto scaling group.
7. Click on the "Actions:Delete" menu.
8. Confirm that you wish to delete your auto-scaling group with "Yes, Delete."
9. Click on "Instances" on the left.
10. Click on each of your instances and then go to "Actions: Instance State: Terminate." When prompted, click "Yes, Terminate". Repeat for both instances.
11. Click on "Load Balancers" on the left.

12. Click on your load balancer "(CC615-LB)", and then go to "Actions: Delete."

13. It is recommended that you check your AWS Billing dashboard periodically to ensure that you are no longer being billed for anything.

    https://console.aws.amazon.com/billing/

Week 7 Lab Report

Please include information requested under the LAB REPORT: labels in the lab steps into Week 7 Lab Report.

SLIDES:

In addition to the overview slide, just a couple of slides with this content:

In Week 7 labs you will accomplish the following:

- Create an AWS dashboard to monitor the BallotOnline deployment in AWS
- Configure the dashboard to show the load balancer numeric metrics widgets for healthy/unhealthy instance numbers, estimated connection number, and HTTP errors returned by the load balancer and the instances
- Add graphing widgets to the dashboard for the same metrics
- Add numeric widgets for the auto scale group, for network bandwidth and CPU utilization
- Create a billing dashboard for BallotOnline and add widgets showing estimated amounts for selected services
- Cause HTTP "Page not found" errors and monitor them in the dashboard
- Bring down one instance and monitor instance data in the dashboard
- Bring down both instances and monitor instance data in the dashboard
- After we are done with all the labs, terminate the AWS IaaS BallotOnline infrastructure deployment.