Architecting Highly Available Applications on AWS

Lab 1

Table of Contents

Introduction

This lab is the first in a series that will transform a fragile two-tier web app into a resilient, scalable application that takes advantage of the core availability features AWS provides.

The basis for the labs will be the MediaWiki application, which we call CloudWiki. AWS CloudFormation will launch stacks with progressively better availability over the course of the class. During these labs, you can examine the architecture and experiment with the configuration. The goal is expose and address the weak spots CloudWiki's architecture.

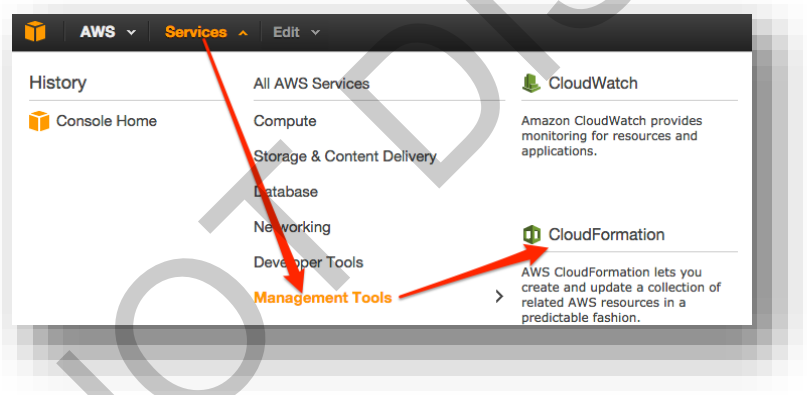
Connect to CloudWiki

Using the instructions provided, you should have launched the lab using qwikLABS.

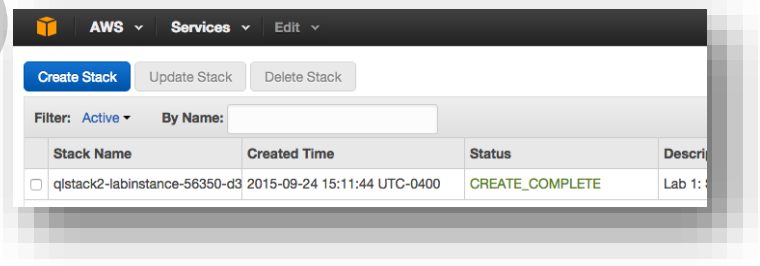
Note **Certain details such as instance names, database names, EIP addresses, security groups, CloudFormation stack names, etc., will differ from what's captured in the screen shots.**

1. Log into the **AWS Management Console** with the **User Name** and **Password** that qwikLABS provided for Lab 1.

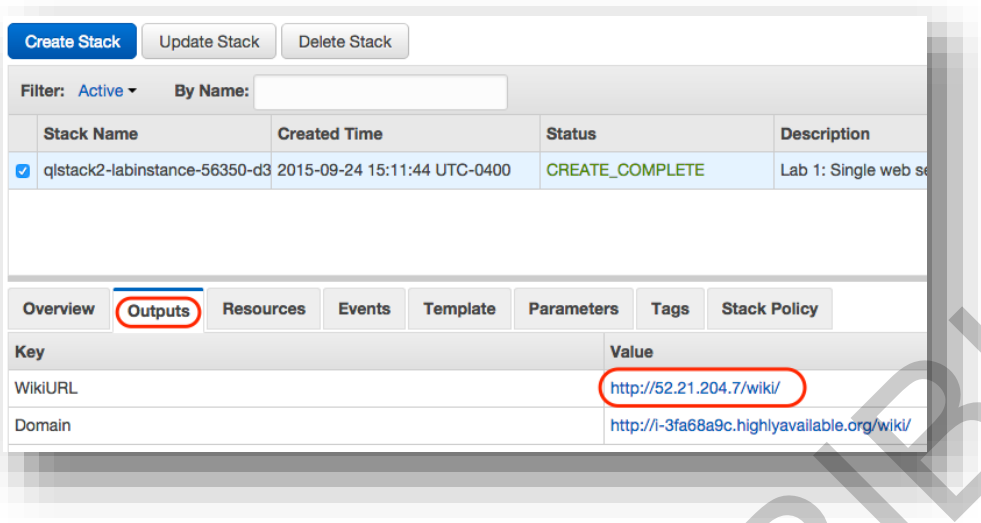
2. In the **AWS Management Console**, on the **Services** menu, click **Management Tools**, and then click **CloudFormation**.



3. On the CloudFormation page, you will see a stack for Lab 1. The **Status** will be either *CREATE_IN_PROGRESS* or *CREATE_COMPLETE*. If stack creation is in progress, select the stack. You will see eight tabs in the bottom part of the page. Click the **Events** tab to monitor progress.



4. Once the status is CREATE_COMPLETE, click the **Outputs** tab. Copy the values for WikiURL and Domain to a text file for later use. Click on the link for the **WikiURL** key.



5. Clicking on the WikiURL link will take you to the CloudWiki homepage.

Exploring the Architecture

The current CloudWiki architecture is very simple and brittle. It consists of a web server that relies on a database server. If either of these instances fails, CloudWiki will be unavailable. It's also not scalable. If the site were to experience a surge in traffic, performance would degrade. As it stands, it would be difficult to horizontally scale the web tier by adding more web servers. It is straightforward to scale both instances vertically to more powerful EC2 instances. But to do so, the instances will have to be stopped first. Obviously, stopping the instances will impact availability.

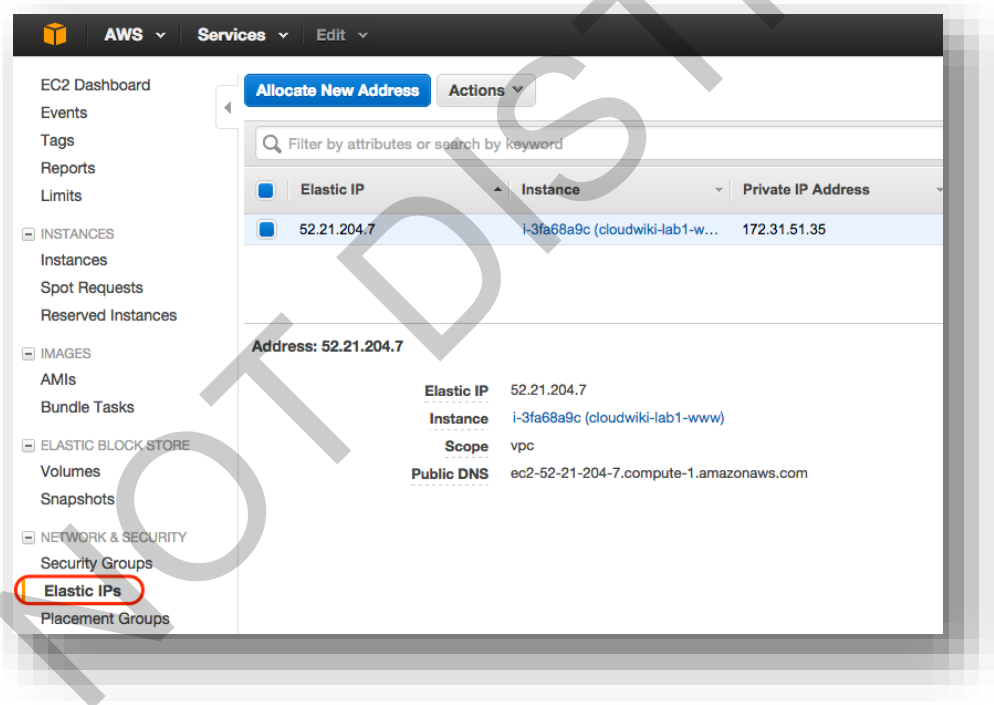
Elastic IPs

The web server instance has an Elastic IP (EIP) address attached to it. This enables the instance to be reached via a static address. A DNS A record is typically mapped to an EIP so that you can access a server using a friendly host name such as www.example.com.

Now we'll view the EIP.

6. In the **AWS Management Console**, on the **Services** menu, click **Compute** and then click **EC2**.

7. In the left navigation column, click **Elastic IPs**.
   You will see an EIP address listed and mapped to the web server instance.



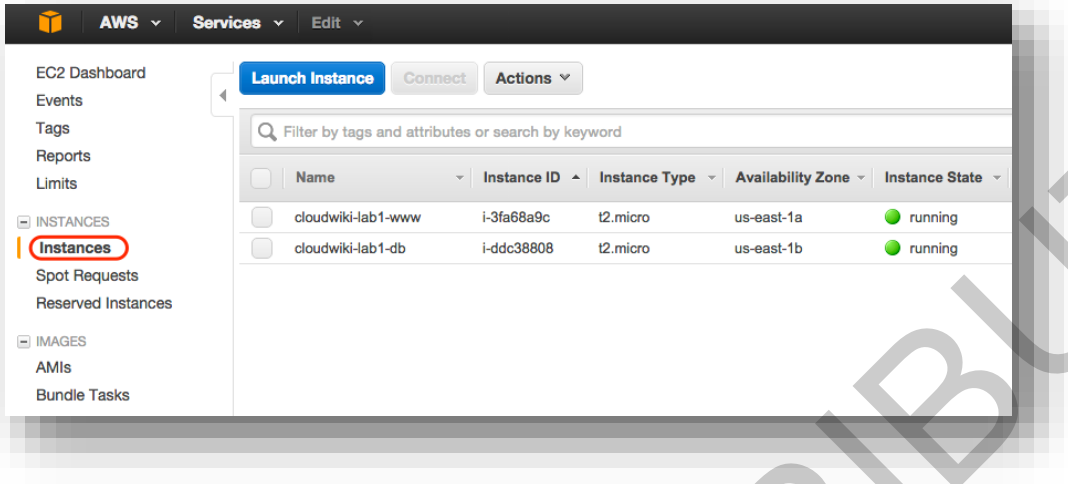
If the instance needs to be replaced, the same EIP can be mapped to the replacement instance so that no DNS changes are required.

Instances

If one of the instances in this two-tier web application fails, then the entire application will fail. In the next few steps, we will perform a manual recovery of an EC2 instance after a simulated failure. As we will see later, AWS has options to automatically mitigate these types of faults.

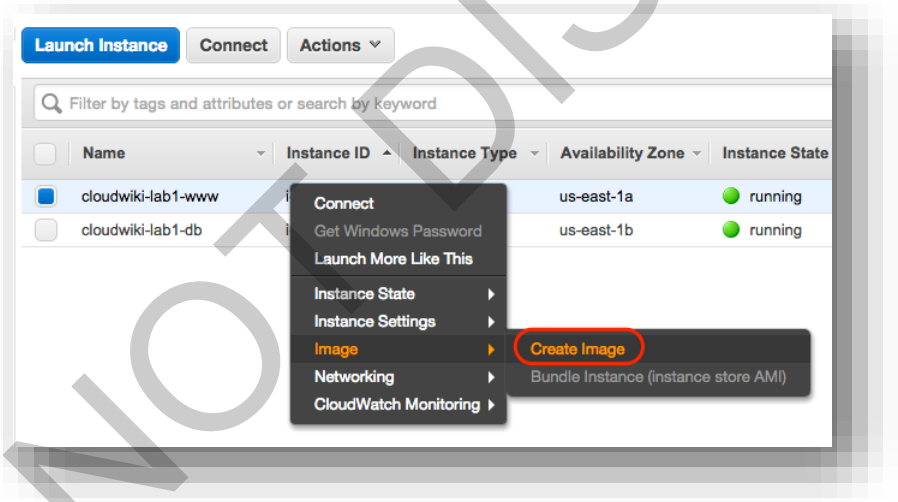
Let's launch a replacement web server and kill the existing web server.

9. In the navigation pane, click **Instances** to see the two EC2 instances that represent the web and database tier for CloudWiki.



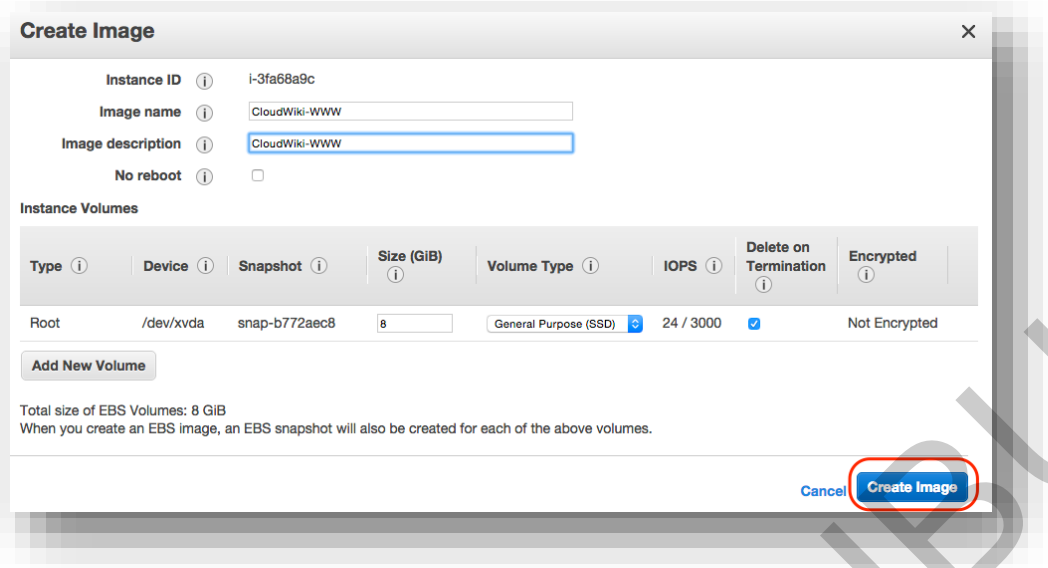
A fault can be simulated by stopping (or optionally terminating) the web server instance. But before doing that, let's create an Amazon Machine Image (AMI, a template to launch instances) based on this running instance. This is necessary in order to launch a replacement instance in the same exact state as the current running instance.

10. Right-click on the **cloudwiki-lab1-www** web instance and click **Image** and then click **Create Image**.

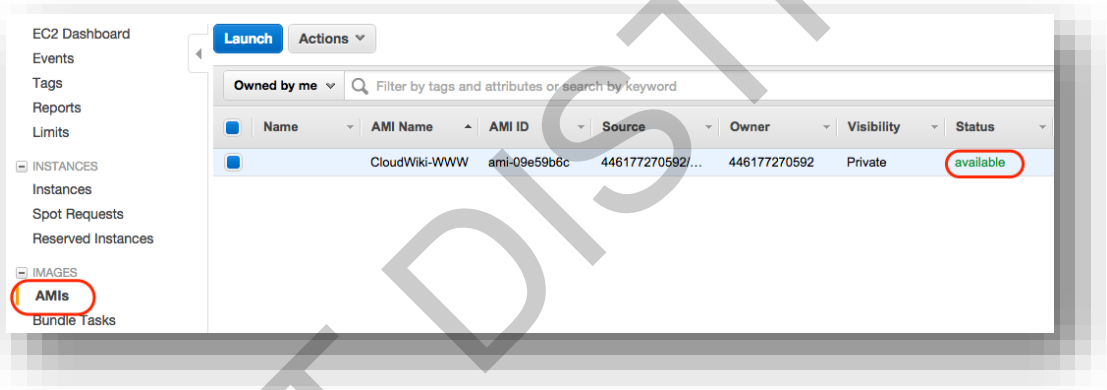


11. In the **Create Image** dialog box, in the **Image name** and **Image description** box, type **CloudWiki-WWW**

12. Click **Create Image**. The instance will reboot (unless you checked the **No Reboot** option).

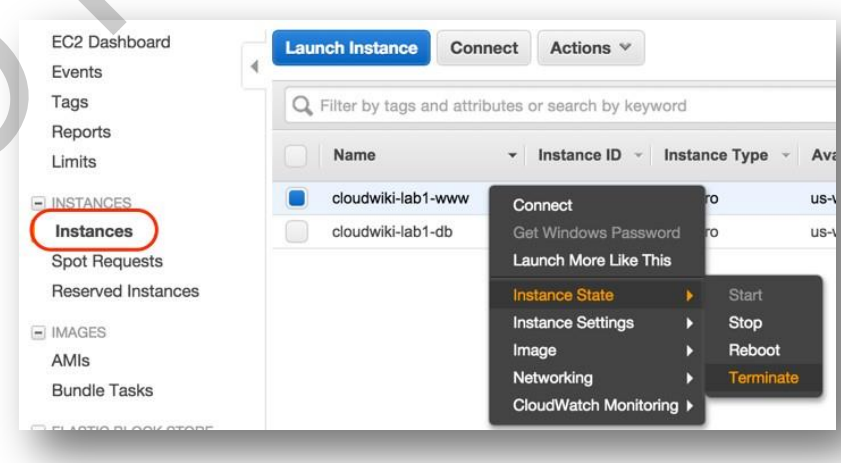


13. In the confirmation box, click the green **View pending image ami** link.
14. The newly created AMI should be listed with the **Status** of *available* (this will take a few minutes).



With our AMI ready, if a fault occurs with our web server, we can simply launch a replacement instance from this AMI. Let's simulate a fault with the web server.

15. In the navigation pane, click **Instances**.
16. Right-click on the **cloudwiki-lab1-www** web server instance (**not the database instance**) and click **Instance State** and then click **Terminate**.



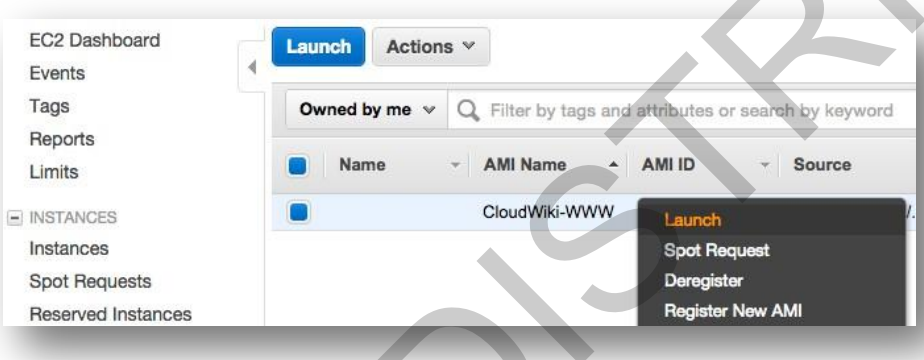
17. In the **Terminate Instances** box, click "**Yes, Terminate"**, which will shut down the web server and remove it from your AWS account. At this point, the web server will not be available. You can try browsing the CloudWiki home page to confirm the site is down.

Replacing a faulty instance

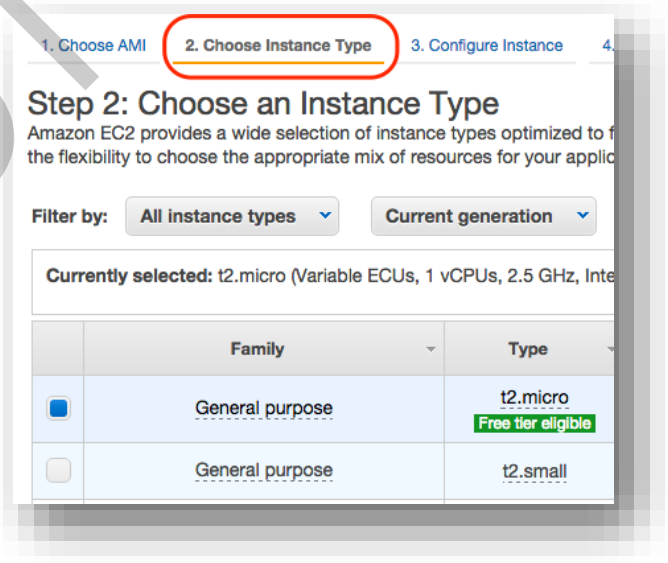
AWS provides CloudWatch monitoring and you can set up alarms when various conditions occur on your AWS resources. Imagine you received a CloudWatch alarm in the form of a text message and now you have to scramble to replace the web server.

18. In the navigation pane, click **AMIs**.

19. Right-click the AMI we created and click **Launch**.

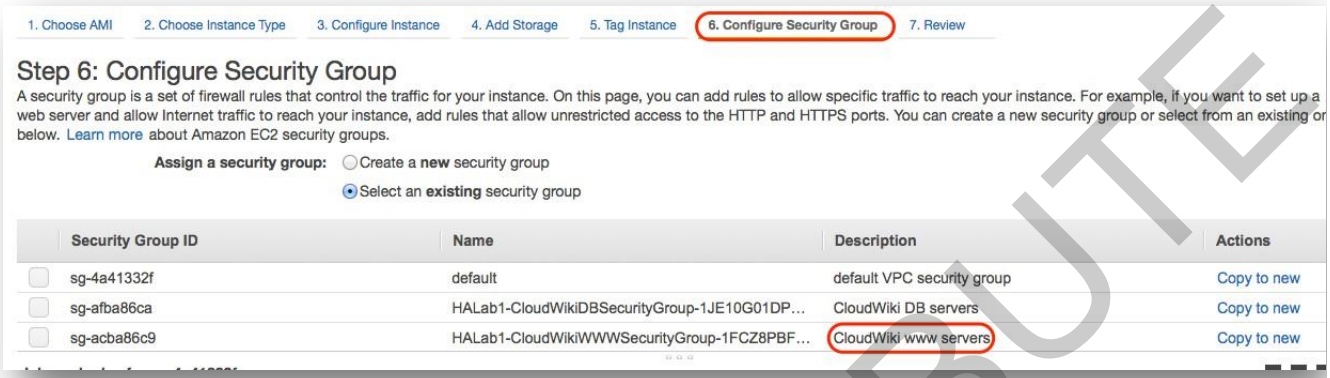


20. On the **Choose an Instance Type** page, keep the default **t2.micro** instance size.



21. Near the top-right, click **Configure Security Group**. Click **Select an existing security group**. Select the check box for the security group that has **CloudWiki www servers** in the **Description**

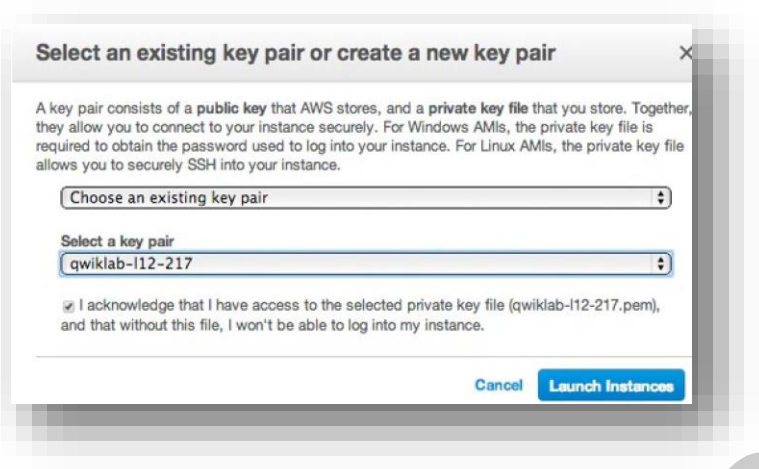
column.



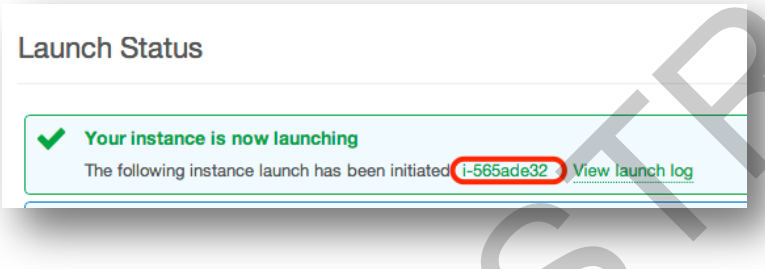
NOTE If you do not see the **CloudWiki www servers** in the table of existing security groups, use the table adjuster (see the three circles in the figure below) to drag the table to show all the security groups.

22. Click **Review and Launch**.
23. On the **Review Instance Launch** page, click **Launch** (for the purpose of this lab you can ignore the following warning on this page).

24. Click the **Launch**.
25. In the **Select a key pair** drop-down list, click the key pair that starts with *qwikLABS* key pair.
26. Select the acknowledgement check box.
27. Click on **Launch Instances**.



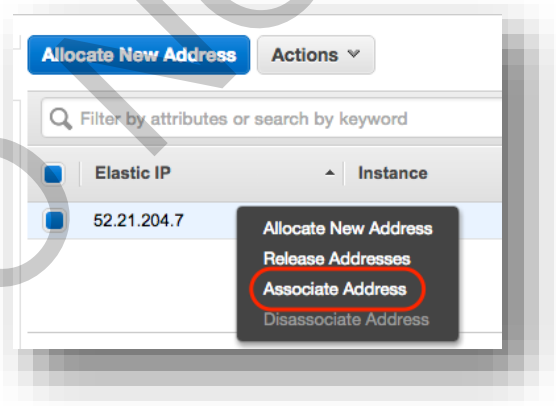
28. On the **Launch Status** page, click on the instance id for the newly launched web server.



29. In the navigation pane, click **Instances**. The instance just launched should be in the *pending* state. In the next minute or so, click on the refresh icon (the two arrow in a circle) near the top-right corner to see if the instance has completed the launch process and is in the *running* state. The instance will be ready to accept requests when the **Status Checks** column shows ***2/2 checks passed***. This may take a few minutes.

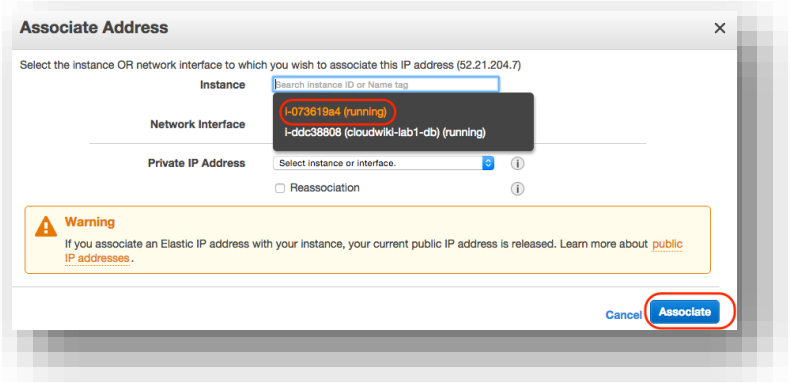
30. In the navigation pane, click **Elastic IPs** and you should see that the EIP is no longer associated with an instance.

31. Right-click the EIP and click **Associate Address**.



32. Click in the text box next to the **Instance** label and select the web server instance and then click

Associate.



We replaced the faulty web server and reattached the EIP address to it. We are now back to where we started. In a minute or so, you should now be able to view the CloudWiki site as before.

Clearly trying to provide HA manually is a tedious process and even when done right, it can be time consuming and error prone. But as you've seen, executing these activities in AWS is a lot simpler than working with traditional IT components. In some cases, this process can be automated using the EC2 Auto Recovery feature.

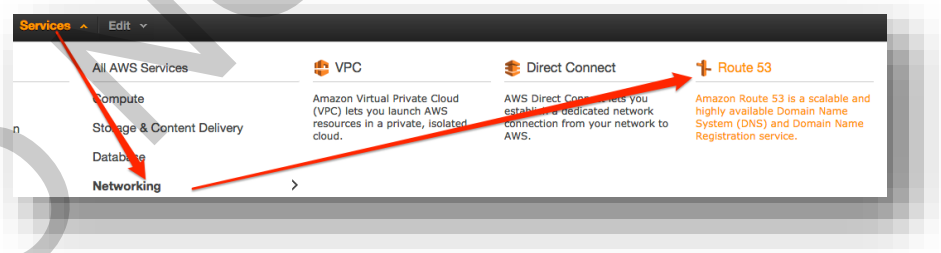
Even with such a brittle configuration, AWS offers services that can help increase availability. Let's look at one of them.

Route 53 DNS Failover

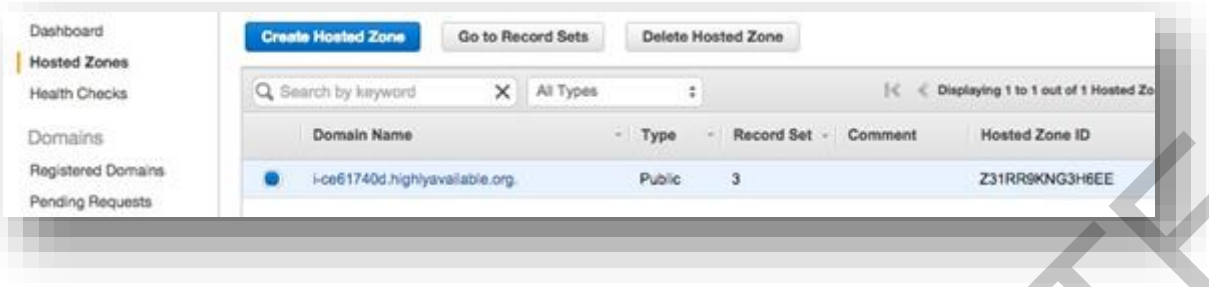
The highlyavailable.org Domain Name

AWS provides ways to failover to a secondary site if your primary site is unavailable. The following exercise will demonstrate how to configure a simple backup website on Amazon S3 and use Amazon Route 53 DNS Failover to automatically route traffic to the backup site when the primary site is unavailable.

33. In the **AWS Management Console**, on the **Services** menu, click **Networking** and then click **Route 53**.

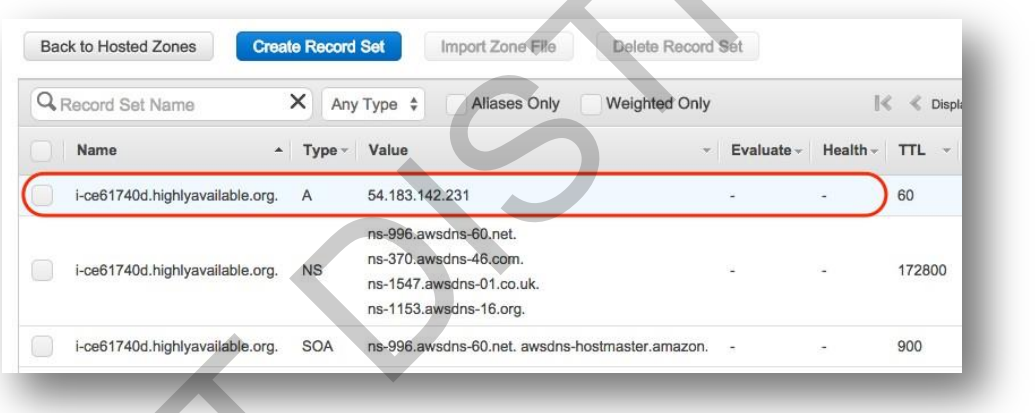


34. The Route 53 home page will look like this. If you see an error such as "Oops, Sorry, you do not have sufficient permissions to execute this action", you can safely ignore it since it's a side effect of the lab environment. In the navigation pane, click **Hosted Zones**.



35. You will see a domain (or "Hosted Zone") has already been created for you to use. In a real-world scenario, you would create a hosted zone for your own domain name; however, this step would require you to modify the DNS settings for a domain that you own and these changes can require multiple hours or days to take effect, so instead we have preconfigured a domain name and hosted zone for you to use in this lab.

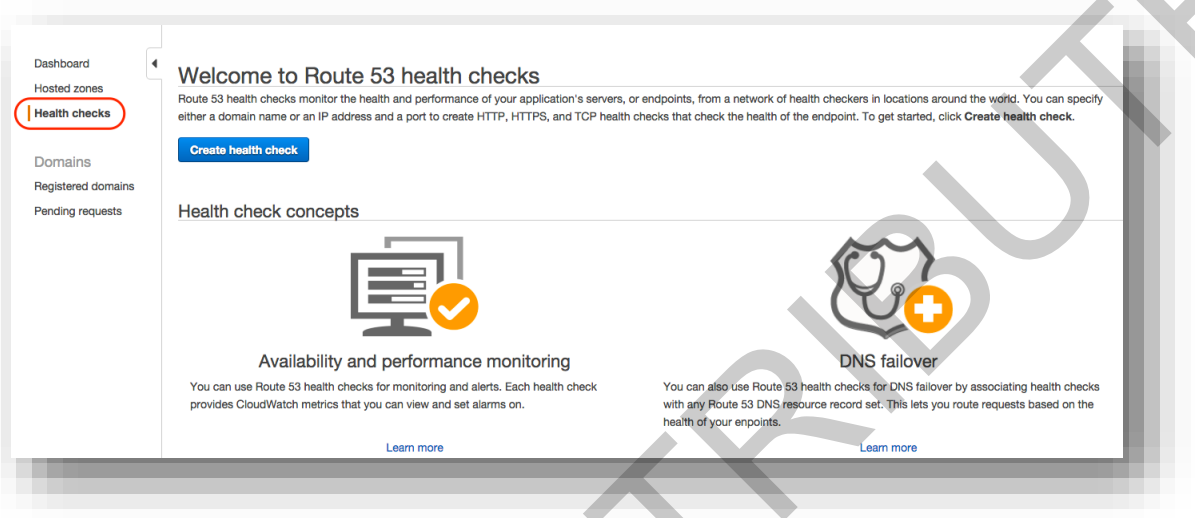
36. Select the hosted zone and click **Go to Record Sets**.
37. You will see three DNS records: two default records that come with your hosted zone, plus an A record pointing to the Elastic IP address of your web server.



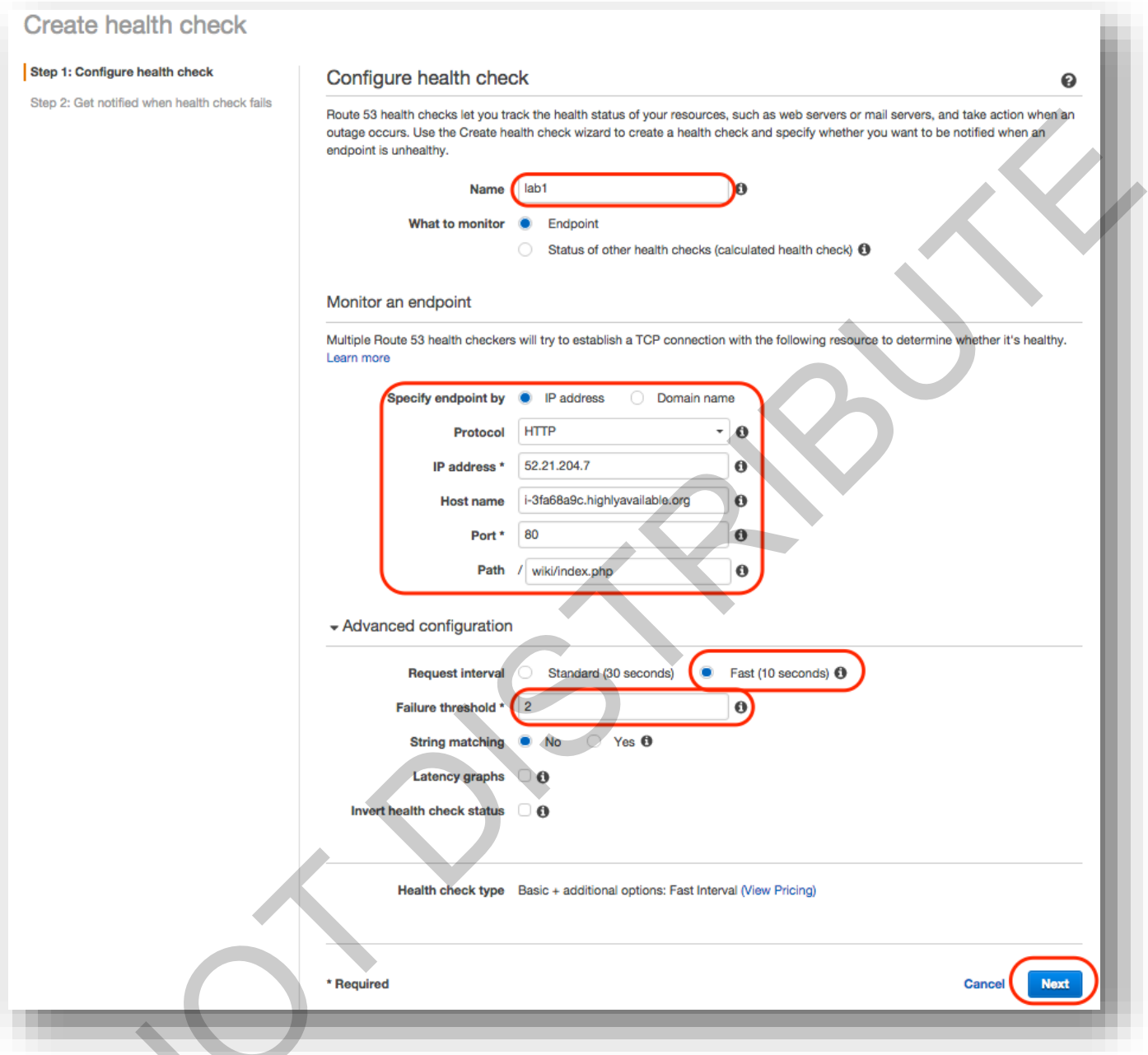
38. To confirm that your domain name points to the application you have just launched, enter this domain name into a new web browser window and you will see the CloudWiki homepage. In your browser's address bar, add a **/wiki** after the domain name, otherwise you'll see the Apache test page.

Creating a Route 53 Health Check

39. The first step in setting up DNS Failover is to create a Route 53 health check, which will monitor your web server availability. From the Route 53 console, in the navigation pane, click **Health Checks**.
40. Click **Create Health Check**.

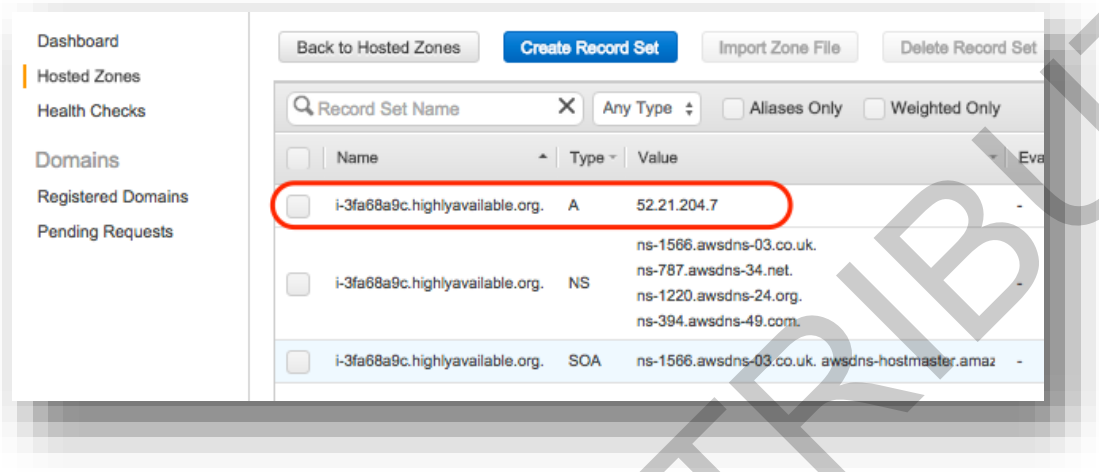


41. On the **Create Health Check** page, in the **Name** box, type **lab1**
42. In the **IP Address** box, type the Elastic IP address of your web server instance.
43. In the **Host Name** box, type your site's domain name (the same domain name that you were given for your hosted zone). Make sure only the domain name is entered (not "http://" or trailing slash, etc.)
44. In the **Port** box, type **80**
45. In the **Path** box, type **wiki/index.php**
46. For **Request Interval**, click the **Fast (10 seconds)** option
47. Set the **Failure Threshold** to **2**. The resulting form should look something like this



48. Click **Next**.
49. On the next screen, keep the default value of **No** for SNS notifications/alarms and click "**Create health check**".

50. In the navigation pane, click **Hosted Zones** to go back to our hosted zone, double click on your hosted zone and click on the **A record** pointing to the Elastic IP address of your web server.



51. In the **Edit Record Set** panel on the right side of the page, do the following:

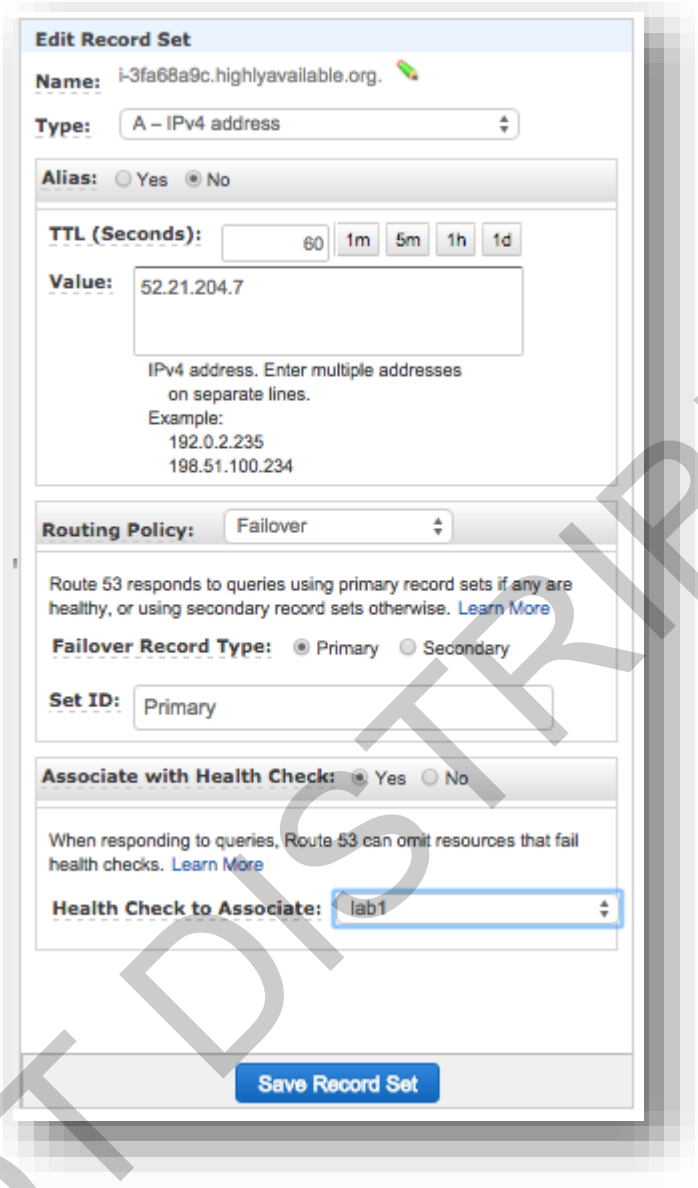
    a. Verify the **TTL (Seconds)** is set to **60** seconds. This limits the amount of time this DNS record will be cached within the Internet's DNS system, which means that there will be a shorter delay between the time failover occurs and the time that end users begin to be routed to your backup site.

    b. Set the **Routing Policy** to **Failover**.

    c. For **Failover Record Type**, click the **Primary** option.

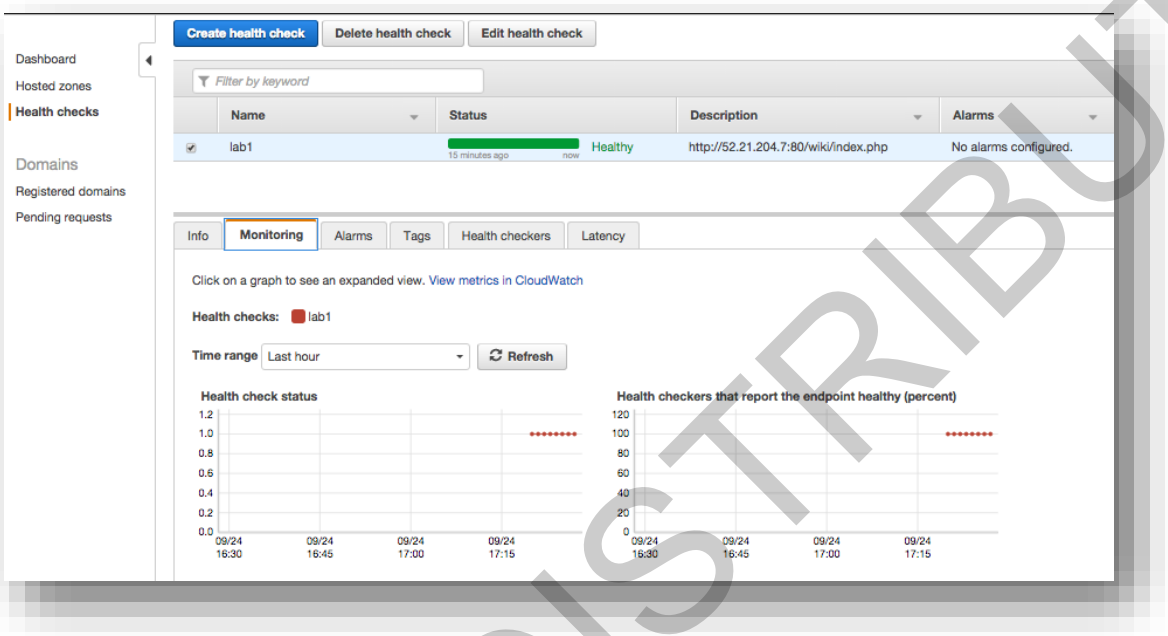
    d. For **Associate with Health Check**, click the **Yes** option.

    e. In the **Health Check to Associate** drop-down list, click **lab1**. You should see the health check we just created and it should be **lab1** as specified is Step 8. Select this health check.



f. Click "**Save Record Set"**.

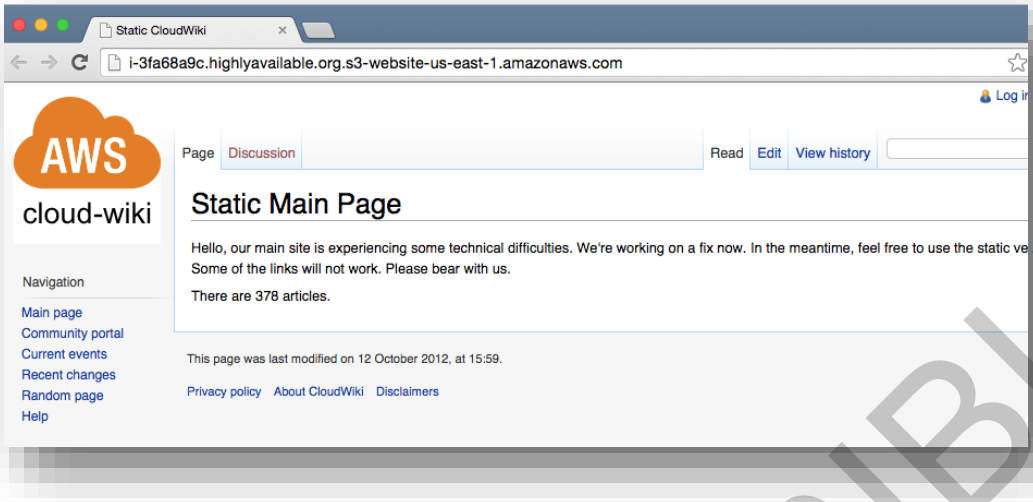
We've now set up a health check and associated with the CloudWiki website. Route 53 will now check the health of your site by periodically requesting your homepage and verifying that it returns a successful response (to be more specific, it's checking independently from multiple locations around the world, with each location requesting the page every 10 seconds). You can view the CloudWatch metrics for the lab1 health check on the **Monitoring** tab on the **Health Checks** page.



S3 Backup Website

52. Let's configure a backup site on Amazon S3. A S3 bucket containing a static version of CloudWiki has been created for you. This bucket has been configured as a S3 website, which can host a static website without any infrastructure. To view the bucket, navigate to S3.
53. In the **AWS Management Console**, on the **Services** menu, click **Storage & Content Delivery** and then click **S3**.
54. You will see a bucket that represents your static website. The bucket's name is the same as your domain.

55. Right-click on the bucket and click **Properties** and on the right hand side, expand the **Static Website Hosting** section. Click on the **Endpoint** link and you will see the static website.



56. The static website contents can be viewed by clicking on the bucket in AWS Web Console.

57. Return to the **Route 53** console page. In the navigation pane, click **Hosted Zones**.

58. Click on the domain name.

59. Click **Create Record Set**.

60. Leave the **Name** box blank.

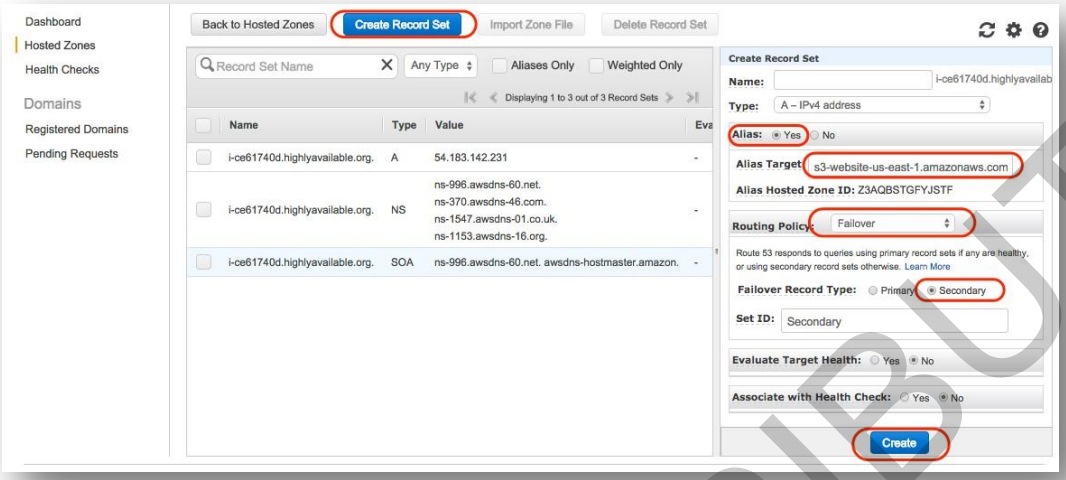
61. For **Alias**, click the **Yes** option.

62. In the **Alias Target** drop-down list, click your S3 website endpoint**.** You may need to refresh the page if you don't see the S3 website listed.

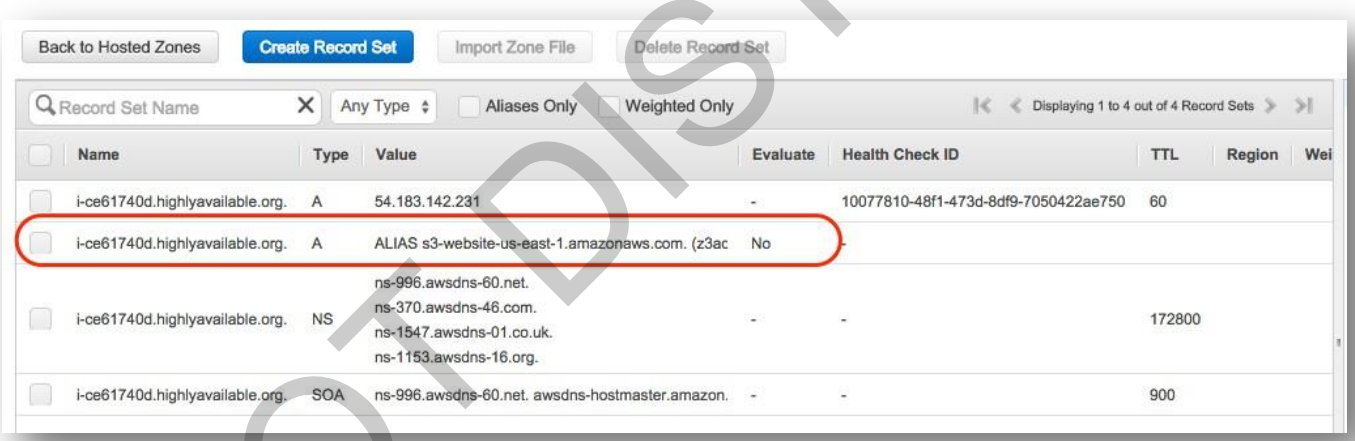
63. Set the **Routing Policy** to **Failover**.

64. For **Failover Record Type**, click the **Secondary** option.

65. Leave the remaining check boxes (Evaluate Target Health and associate with Health Check) at their default settings of **No**, and then click **Create**.



Here's what your Route 53 hosted zone looks like after you're finished.



Now we've configured the S3 bucket as the failover site in case Route 53's health checks detect the primary website is unavailable.

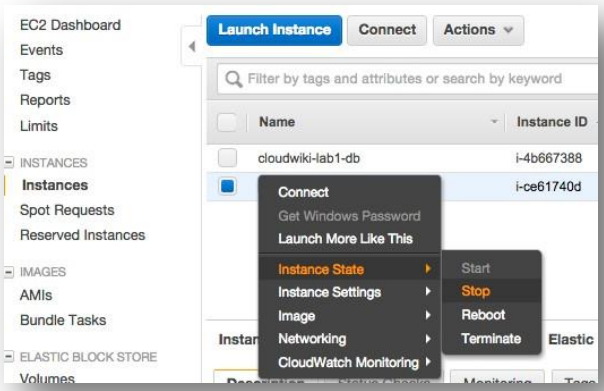
Failover to the Backup site

Now, we'll simulate failure of your web application and demonstrate how Route 53 automatically detects and responds to the failure with no input required from you.

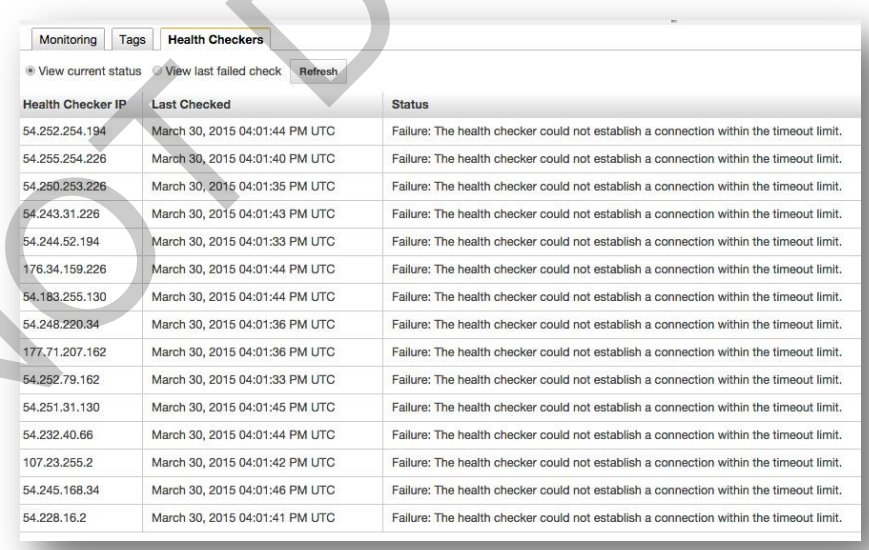
66. First, visit the CloudWiki site using the domain name in the web browser.

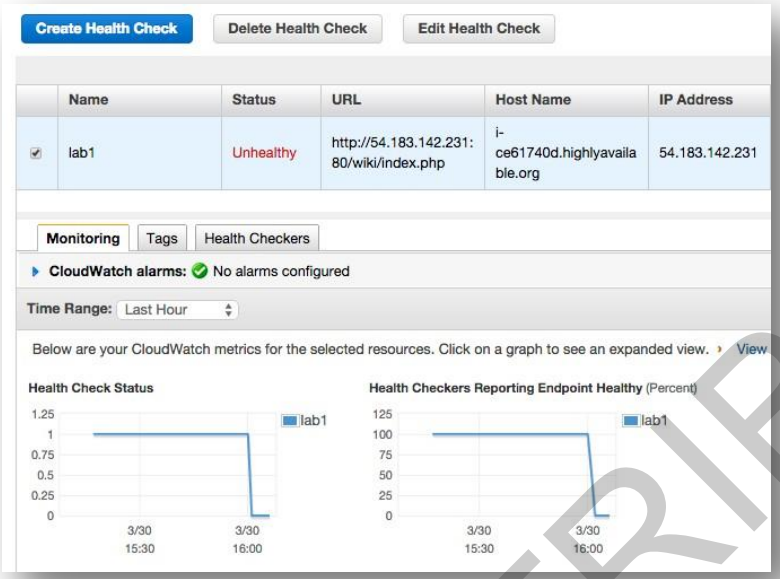
67. Navigate to the EC2 console and right-click on your web server instance, click **Instance State** and then click **Stop**.

68. Click **Yes, Stop**.



69. Go back to your browser and reload the page for CloudWiki using its domain name. You should see an error, which will last approximately 90-120 seconds, which is the amount of time required for Route 53 to detect and confirm failure of your web server.
70. You can view the health check failing via CloudWatch. In the navigation pane, click **Health Checks**, select the check box for **lab1**. In the lower pane, click the **Health Checkers** tab for the **lab1**, you should see something like this:





71. After about two minutes have elapsed, reload the CloudWiki page in your browser (the address shouldn't contain /wiki or /wiki/index.php, it should just be the domain name). You should now see a static version of CloudWiki (not all the links will work). Route 53 has detected the failure of the primary and is now sending traffic to the backup site.

72. **Bonus** If time permits, you can see what happens when the primary site recovers.

    g. Start the previously stopped EC2 web server instance.

    h. Within a few minutes, refresh the site (add /wiki to the end of the address) and now instead of the static site, you should see the primary site.

With a few simple steps, we have configured a backup website and demonstrated how Route 53 will automatically fail over to the backup site when your primary website goes down. If your site were to experience downtime, you can leverage this feature to provide a standby version of your site until your primary site is restored. While we've demonstrated failover to a static site, Route 53 can be used to failover to another active site as well, perhaps in a different region or even outside of AWS.

Summary

In this lab, we saw a simple web architecture. It works, but it's not designed to be highly available. There's no fault tolerance built into the web or data tiers. If anything goes wrong, the entire application will become unavailable. But by using Route 53 DNS failover, in case something goes wrong with our application, we can provide a better user experience by falling back to a static site.

In subsequent labs, you will see how this application adds AWS HA features to build in more resiliency and scalability.

