



re:Invent 2019

ARC406

Building Multi-Region Microservices

Copyright © 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved.

This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited.

All trademarks are the property of their owners.

Table of Contents

Prerequisites	4
Problem Statement	6
Solution Overview	7
Task 1 – Build the regional solution stacks using CloudFormation	8
Task 2 – Architect and design	8
Task 3 – Review the CloudFormation-built solution	9
Task 4 – Configure the leaderboard website and deploy to S3.....	9
Task 5 – Configure DynamoDB multi-region replication with Global Tables.....	12
Task 6 – Import leaderboard data.....	13
Task 7 – Test & validate	13
Task 8 – Cleanup	14
Task 9 – Optional (further reading).....	14
Appendix A: Troubleshooting common issues	15
Appendix B: Prerequisites for running on local machine.....	15

Prerequisites

This exercise requires access to Amazon Web Services console and a functional AWS CLI installation. The console is found here: <https://console.aws.amazon.com>. As this is a 400-level workshop, it is assumed that you already have the knowledge required to satisfy the prerequisites for the workshop, as follows:

Prerequisite 1: AWS account and console access

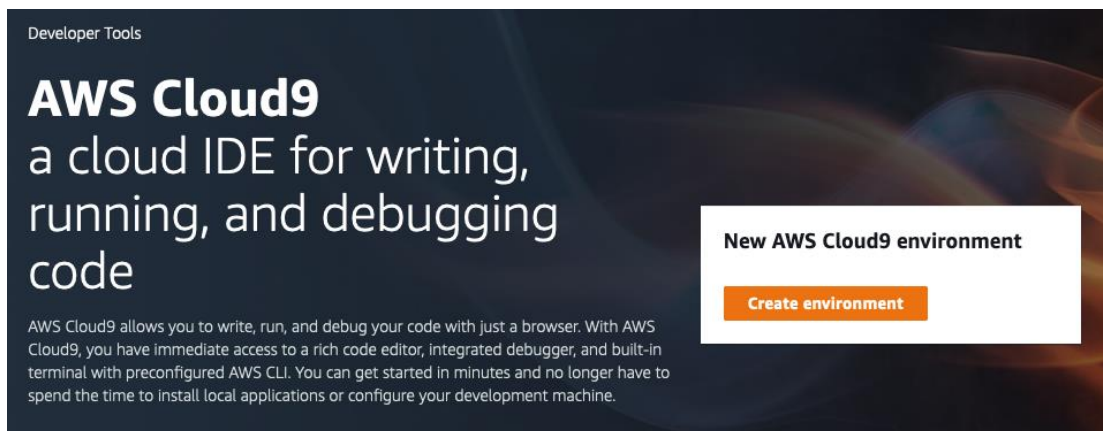
1. If you are doing this with a pre-provisioned account, your lab facilitator will provide you a URL and credentials
2. <https://dashboard.eventengine.run/login>
3. Use that information and log into your account

Alternatively, If doing this on your own machine

1. Setup an AWS account,
2. Open your browser, go to <https://console.aws.amazon.com>, and log in.
3. Go to [Appendix B](#) for additional steps

Prerequisite 2: Create AWS Cloud9 Environment

1. Ensure your region is - us-west-2 (Oregon) (option in top right dropdown of console)
2. Go to the AWS Cloud9 Service (Cloud 9 is in services under 'Developer Tools')
3. On the welcome screen click 'Create Environment'



4. On the next screen provide a Name for your environment (Arc406 should be fine); description is optional
5. On the 'Configure Settings' page, leave the defaults (as follows)

Configure settings

Environment settings

Environment type [Info](#)

Choose between creating a new EC2 instance for your new environment or connecting directly to your server over SSH.

☒ **Create a new instance for environment (EC2)**

Launch a new instance in this region to run your new environment.

☐ **Connect and run in remote server (SSH)**

Display instructions to connect remotely over SSH and run your new environment.

Instance type

☒ **t2.micro (1 GiB RAM + 1 vCPU)**

Free-tier eligible. Ideal for educational users and exploration.

☐ **t2.small (2 GiB RAM + 1 vCPU)**

Recommended for small-sized web projects.

☐ **m4.large (8 GiB RAM + 2 vCPU)**

Recommended for production and general-purpose development.

☐ **Other instance type**

Select an instance type.

t3.nano

Platform

☒ **Amazon Linux**

☐ **Ubuntu Server 18.04 LTS**

Cost-saving setting

Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

IAM role

AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

► Network settings (advanced)

6. Click 'Next step'
7. On the next screen click 'Create environment'

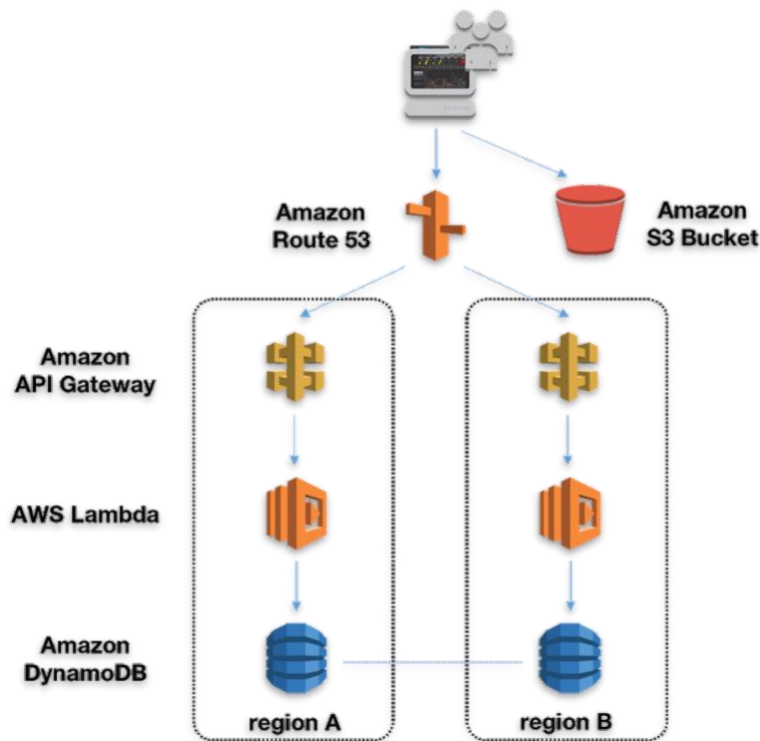
***Note – it can take a few minutes for the Cloud9 environment to provision

Problem Statement

You are a game developer who plans to release a global game. You want to deploy the game to multiple AWS regions for latency reasons. You want to share a leaderboard. In addition, you want the ability to route players' requests to different regions.

This workshop focuses on the leaderboard aspect of the game. The leaderboard is a microservice, built using [Amazon S3](#), [Amazon API Gateway](#), [AWS Lambda](#), and [Amazon DynamoDB](#). For data replication it uses the [multi-master replication capability of DynamoDB \(Global Tables\)](#).

Solution Overview



The solution, shown above, is deployed in two AWS regions – region A: us-west-2 (Oregon), and region B: eu-west-1 (Dublin). Note that the Route 53 setup is outside the scope of this workshop. It can be performed as an optional exercise.

The leaderboard microservice publishes an API interface using API Gateway, processes requests using AWS Lambda, and stores its data in DynamoDB. A simple web UI for the leaderboard service is hosted from S3. The UI uses JavaScript to query the leaderboard table. Given a domain, Route 53 can be configured to route those requests to one of the two regional deployments based on latency and availability, in case of a failure. This requires provisioning your own domain name for use with this application.

In the workshop, you deploy the solution using AWS CloudFormation, configure the leaderboard UI, and DynamoDB replication using the DynamoDB Global Tables features, and then exercise the multi-master data replication. In addition, you can optionally configure Route 53, create and test a failure scenario.

Task 1 – Build the regional solution stacks using CloudFormation

Launch the AWS CloudFormation stack for the region A, us-west-2 (Oregon):



The link opens the CloudFormation console in your account in us-west-2 region.

1. In “Select Template”, click Next.
2. Under “Specify Details”, enter the following:
 - a. For Stack name, you can use the default name or enter your own name of the CloudFormation stack.
 - b. Under Parameters, enter the prefix string, `ServiceNamePrefix`, that will be used in the names of all the resources created, such as the S3 bucket and DynamoDB table. Use lower case and numbers only. Make note of the prefix you used here, as you will need it again below. *****Use lowercase characters and numbers only and keep it short. We recommend using your initials.**
3. Click Next.
4. In the Options section, scroll down to the bottom and click Next.
5. In the Review section, scroll down to the bottom, check the “I acknowledge...” checkbox and click Create. You may have to click the refresh button in CloudFormation console for the stack to appear; it should show `CREATE_IN_PROGRESS` under Status.

While the stack for the region A is being built, launch the AWS CloudFormation stack for the region B, eu-west-1 (Ireland):



Repeat the CloudFormation stack configuration steps above using the same settings you used for the previous stack. Use the same value for `ServiceNamePrefix`.

Task 2 – Architect and design

While the stacks are being built, we will discuss the challenges of multi-master data replication across global regions, and our solution.

Task 3 – Review the CloudFormation-built solution

When the CloudFormation builds are completed:

- Review the Outputs tab. This tab lists the key resources built for you. Copy the key-values to a text file. You need them in the steps that follow.

Below is the output from a sample run (your values will be different).

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets	Rollback Triggers
Key		Value			Description				
ApiGatewayUrl		https://j48z1qfukd.execute-api.us-east-1.amazonaws.com			Root URL of the API gateway				
S3BucketSecureURL		https://reinvent-buildersession2018-bucket.s3.amazonaws.com			Name of S3 bucket to hold website content				
HostedZoneName		reinvent.builder-reinvent2018.com			The fully-qualified domain name.				
NameServerNames		ns-1036.awsdns-01.org,ns-547.awsdns-04.net,ns-371.awsdns-46.co.uk,ns-1909.awsdns-46.co.uk			The nameservers for the hosted zone				
OutGSIName		Country-Score-Index			The regionScore GSI Name.				
WebsiteURL		http://reinvent-buildersession2018-bucket.s3-website-us-east-1.amazonaws.com			URL for website hosted on S3				
HostedZoneID		Z1WN2ORVJZWFPG			The ID of the Hosted Zone.				
OutTableName		reinvent			DynamoDB table name.				

Task 4 – Configure the leaderboard website and deploy to S3

The following cli commands should be run in your Cloud9 environment. Please ensure your AWS cli and boto3 are updated by running

```
$ sudo pip-3.6 install boto3 awscli
```

Copy the leaderboard website files from <https://s3-us-west-2.amazonaws.com/reinvent2019-arc406/www/> to your S3 buckets created by the CloudFormation builds in both regions. You can do this by using 's3 sync' command of AWS CLI in your Cloud9 environment;

```
$ aws s3 sync s3://reinvent2019-arc406/www/ s3://<my bucket in region us-west-2>
```

```
$ aws s3 sync s3://reinvent2019-arc406/www/ s3://<my bucket in region eu-west-1>
```

Next, for each bucket, perform the steps below.

Update the API Gateway endpoint in your index.html

- Download index.html from your S3 bucket, and open it in a text editor.

- Search for “var apiEndpoint” and update the value to depict your ApiGatewayUrl value that you noted from the output of the CloudFormation build in this region:

```

49
50 // Update endpoint below (ApiGatewayUrl from CloudFormation Stack) >>>>>
51 var apiEndpoint = "https://ivbxivy8gi.execute-api.us-east-1.amazonaws.com/LATEST/builder";
52 // <<<<<<< End Update Endpoints
53
54 var resultSet;
55

```

- Upload the updated index.html back to your S3 bucket.
- Repeat this step for both regions.

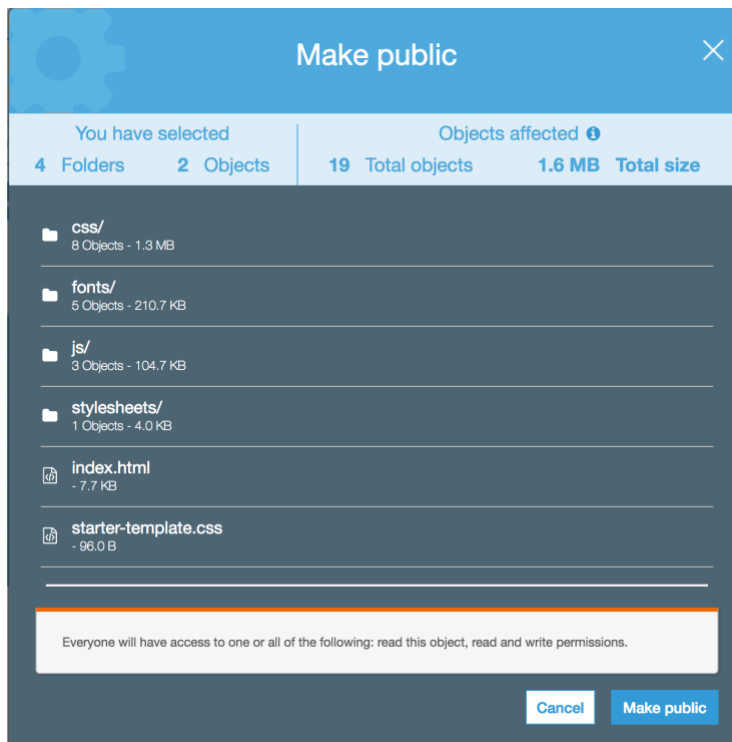
Grant public read access to the contents of the bucket

- In the S3 console click on the bucket. Select all the objects by clicking the checkbox next to Name, click on Actions, and select Make public:

The screenshot shows the AWS S3 console interface. At the top, there are tabs for Overview, Properties, Permissions (labeled 'Public'), and Management. Below the tabs is a search bar. The main area shows a bucket named 'US West (Oregon)' with a refresh icon. On the left, there are buttons for 'Upload' and 'Create folder'. Below these, there is a list of objects with checkboxes for selection. The objects are: css, fonts, js, stylesheets, index.html, and starter-template.css. The 'Actions' menu is open, showing options like Open, Get size, Download as, Select from, Rename, Delete, Undo delete, Cut, Copy, Paste, Change storage class, Initiate restore, Change encryption, Change metadata, and Make public. The 'Make public' option is highlighted at the bottom of the menu. On the right, there is a table showing the details of the selected objects, including Last modified, Size, and Storage class.

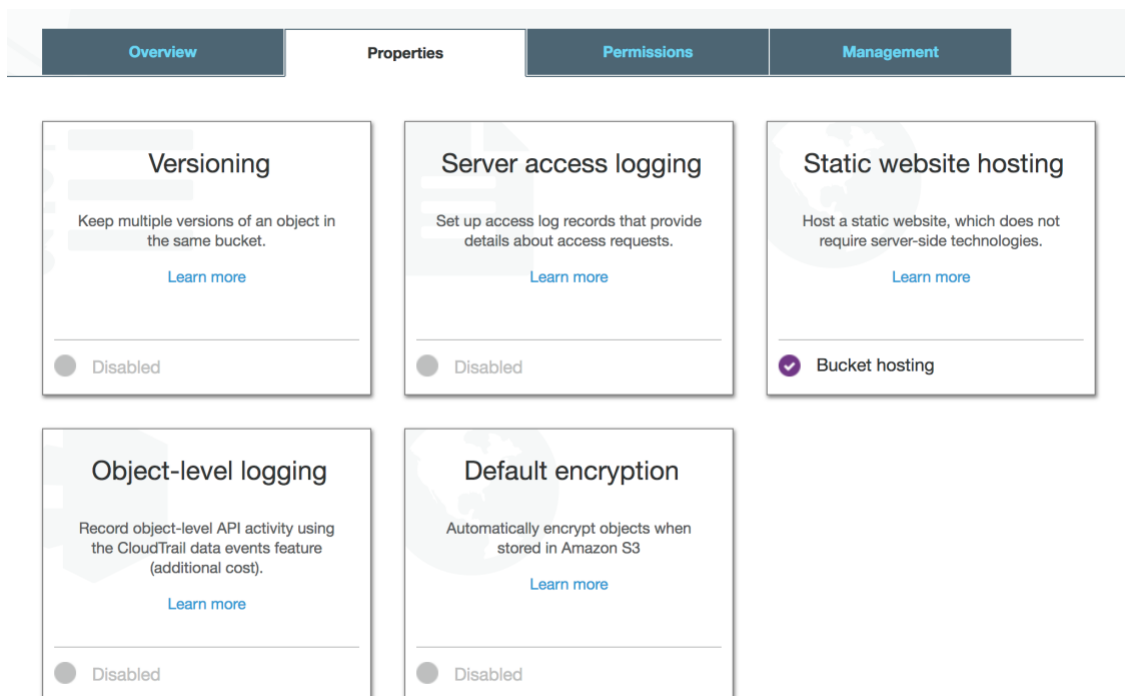
Name	Last modified	Size	Storage class
css	--	--	--
fonts	--	--	--
js	--	--	--
stylesheets	--	--	--
index.html	Nov 7, 2018 11:16:48 AM GMT-0800	7.7 KB	Standard
starter-template.css	Nov 7, 2018 11:16:48 AM GMT-0800	96.0 B	Standard

- In the dialog that opens up, click the ‘Make public’ button (see the image below).



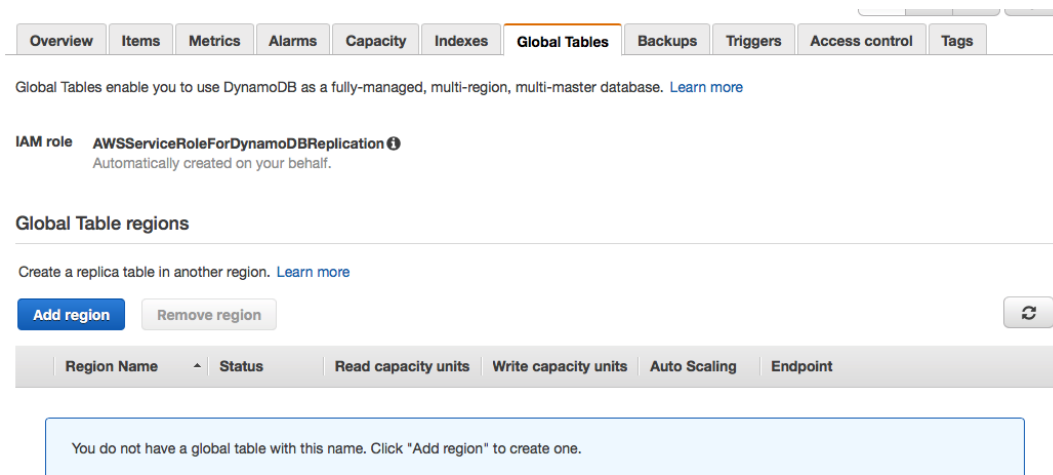
Verify that static web hosting is enabled

- Click on Properties in the menu navigation bar. In the “Static website hosting” section verify that Bucket hosting is checked, as shown below.

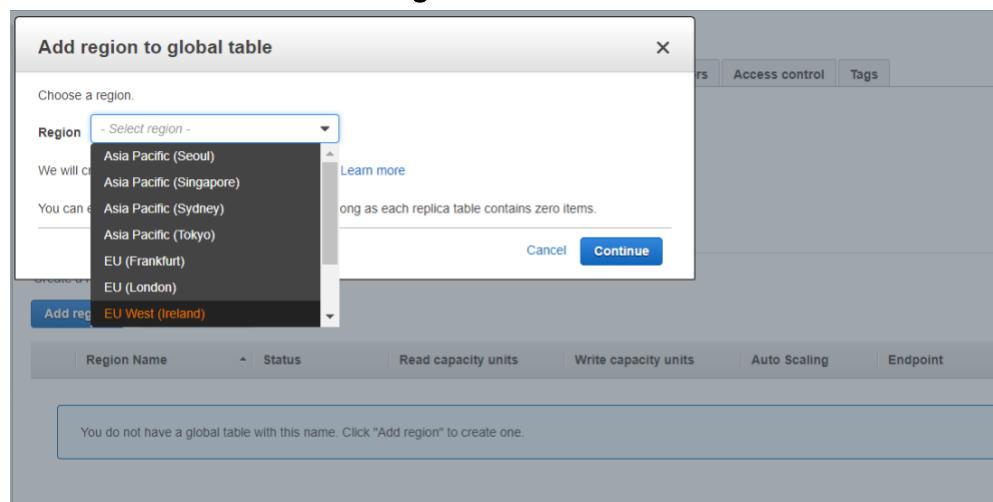


Task 5 – Configure DynamoDB multi-region replication with Global Tables

- Create the global replica for your DynamoDB table
 - Ensure that you are working in 'us-west-2'.
 - Look in the Outputs tab of your CloudFormation script. Note the name of the DynamoDB table created for you (OutTableName).
 - Go to the DynamoDB console. Choose **Tables**, then choose the table created for you.
 - Choose the tab **Global Tables**.



- Now choose the button **Add Region**.



- Choose **EU West (Ireland)**.
- Choose **Continue**.

- Wait while the table is being created.

Task 6 – Import leaderboard data

- Download the [DDBImport.py](#) script and the leaderboard data file, [import-data.csv](#), to the same directory. The script imports the data from the csv file into the DynamoDB table in us-west-2.

****if you are having difficulty downloading the files try to use a Chrome or Firefox browser*

- First download these files to your local machine
- Then, within your Cloud9 environment click on 'File' | 'Upload local files' and select these two files
- Once uploaded right click on the 'DDBImport.py' in the left pane and select run
****If your DDBImport.py script fails to run, try re-installing boto3 in the terminal window with this command 'sudo pip-3.6 install boto3 awscli' and rerun the DDBImport.py script*

If you are running this lab on your local machine (versus Cloud9) run the following commands

- Run the DDBImport.py script: `python3 DDBImport.py`. The output should be similar to the following sample output:

```
python3 DDBImport.py
row: 100 in 3.164921998977661
row: 200 in 2.9544801712036133
row: 300 in 3.071554183959961
row: 400 in 3.1441802978515625
row: 500 in 3.1302428245544434
RowCount: 500, Total seconds: 15.538883209228516
```

Task 7 – Test & validate

- In the DynamoDB console for us-west-2 region, take a look at the DynamoDB table and verify the data is there.
- Change the region to eu-west-1 and look at the DynamoDB replica table in that region. The data should be replicated as we are using a Global Table.
- Go to the leaderboard web page in us-west-2 and check that it displays the top scores.
- Click the "Add score" button and add a high score.
- Refresh the leaderboard page; it should show the added score.
- Now check the leaderboard web page in eu-west-1 and verify you see the score.

Task 8 – Cleanup

If running in a provided account you do not need to worry about cleanup as the accounts will be decommissioned once the session is over.

Alternatively, if you are running this in your own account and to avoid incurring additional charges, remember to delete the CloudFormation templates in both region A and region B. You will need to delete all items from your bucket as well as any hosted zone resource sets before deleting the template.

Task 9 – Optional (further reading)

Networking

When building multi-region microservices failover and latency-based routing should be considered for the solution.

In order to create a custom domain for the website it is important that the [bucket name be the same as the DNS name](#).

For the API gateway you can set up [automatic failover](#) using latency based routing. In order to do create the routing you have to own the DNS name and load or request an SSL certificate from ACM as API Gateway uses https only. The cloud formation template does provide examples for creating the bucket and hosted zone to support this.

Monitoring and logging

Identify, for each component, the metrics to monitor that will tell you when your system is not functioning correctly.

Create CloudWatch alarms for chosen metrics. Consider (at least) the following components:

- DynamoDB
- API Gateway
- Lambda

For DynamoDB, use the following information to brainstorm the metrics that you'd like to monitor for your DynamoDB table:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/monitoring-cloudwatch.html>

Set up CloudWatch alarms for your selected metrics:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/creating-alarms.html>

Failover/recovery

It is possible that the application deployment in a region may fail.

Think through the various components of your system, and build a list of failure scenarios.

For each scenario, consider:

- How would you notice that this component has failed?
- How does this failure affect the functioning of the application?
- What recovery actions could be taken, for this failure scenario?
- Could this scenario be avoided through changes to your code or configuration?

[Amazon Route 53](#) can be used not only to configure a custom domain for your application, but also to configure application health checks and routing to healthy endpoints, as well as monitor the health of your application and its endpoints. In addition, Route 53 supports routing based on latency, Geo DNS, Geoproximity as well as weighted round robin routing. In combination with DNS Failover, these can be used to enable a low latency, fault-tolerant solution that best suits the needs of your application.

Appendix A: Troubleshooting common issues

Problem: I don't have Boto3 installed, and when I try to install it, I get the error "pip not found"

Fix: Install pip by following the instructions at: <https://pip.pypa.io/en/latest/installing/>

Problem: Website doesn't display

Fix: Verify bucket and object permissions, and make sure they are set to public

Problem: The data import script, DDBImport.py fails

Fix: Verify that import-data.csv is in the same directory as the script

Problem: The leaderboard website doesn't list scores, and/or adding a score fails

Fix: Troubleshoot with the following steps:

1. Test the Lambda functions
2. Verify and test the API in the API Gateway console
3. Verify the API Gateway endpoint in index.html

Appendix B: Prerequisites for running on local machine

If you choose to leverage the AWS CLI and issue commands locally on your machine, complete the following steps

Prerequisite 1: Create IAM Access Key

Create access keys for your IAM user, and store in your credentials on your laptop ([instructions here](#)). These access keys will be used by the AWS CLI and the load generator script.

Prerequisite 2: Install Python and AWS SDK for Python

Make sure you have Python 2 version 2.6.5+ or Python 3 version 3.3+ installed. Also, make sure you have [Boto 3 – The AWS SDK for Python](#) installed.

Prerequisite 3: Configure your AWS CLI

1. Install the AWS CLI following the instructions in [Installing the AWS Command Line Interface](#).
2. Using the access key from #2 above, configure the AWS CLI following the instructions in [Configuring the AWS CLI](#).