AWS Lab – Comparing ElastiCache with MySQL using LandSat data

Table of Contents

Tasks

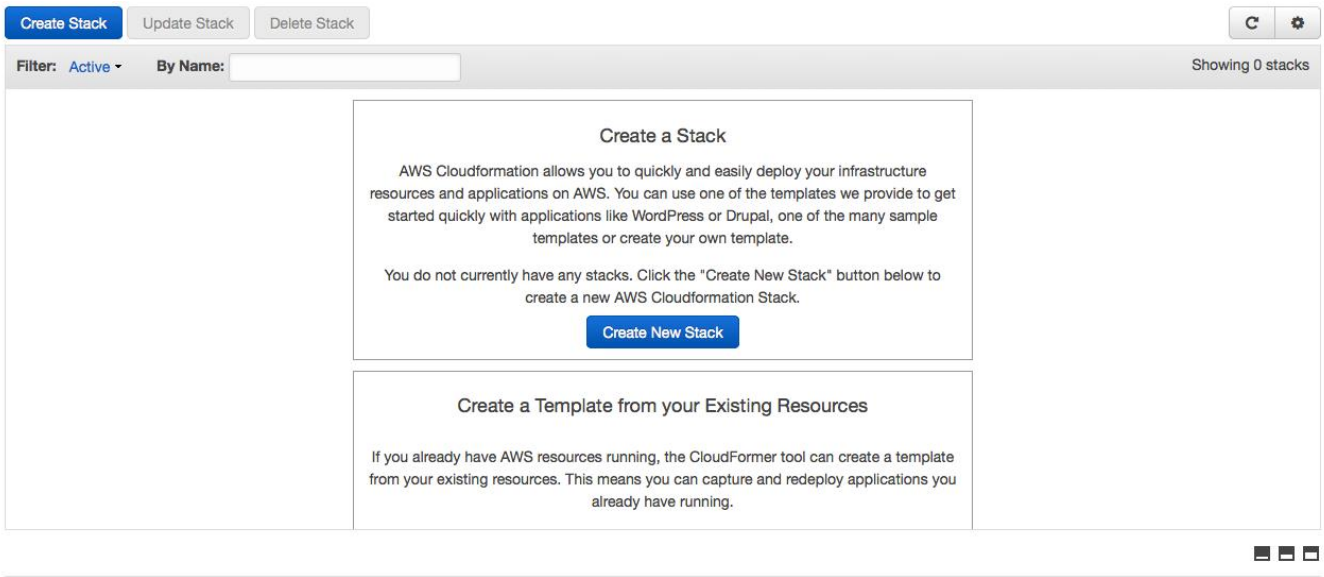
Task 1 – Build your Infrastructure

For this step, we are going to launch an AWS CloudFormation script which will build our reference infrastructure.

1.1 Launch AWS CloudFormation template

Let's walk through launching this first basic stack.

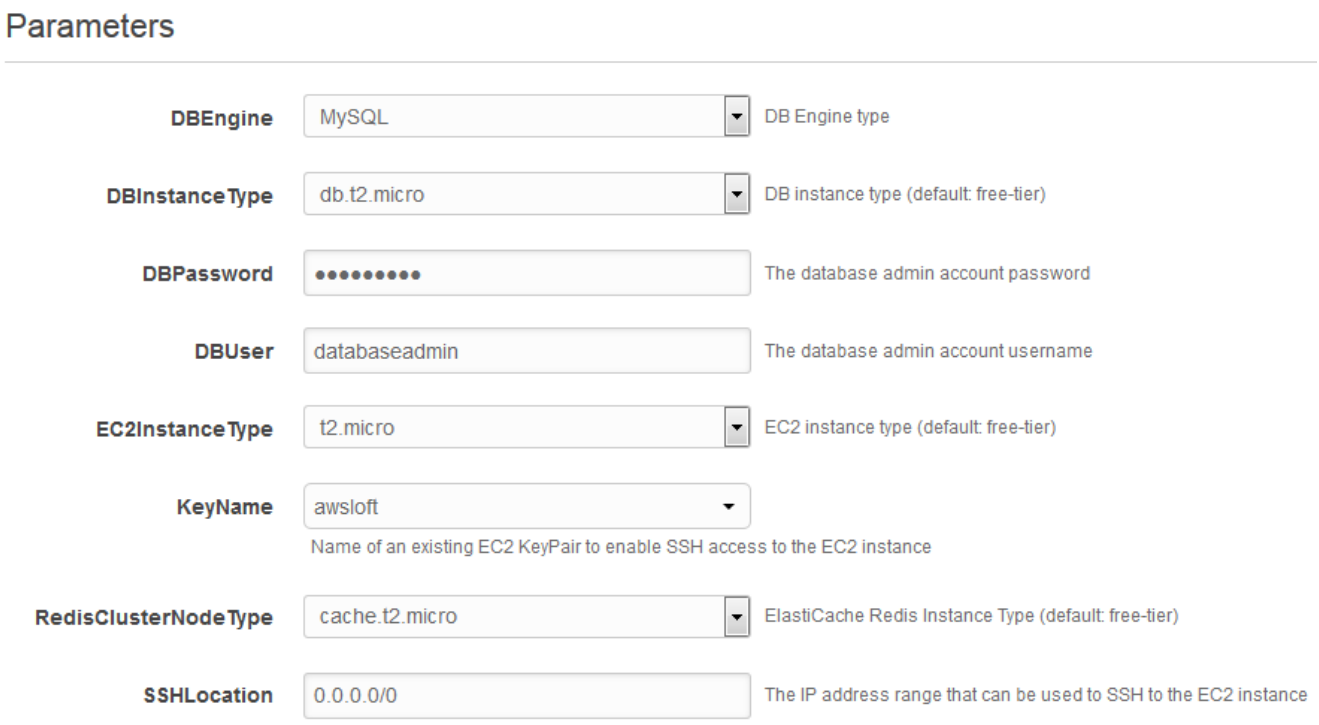
1. Navigate to the AWS CloudFormation console: https://console.aws.amazon.com/cloudformation/home?region=us-east-1



2. You should now see the "**Create Stack**" wizard in front of you. Next select "**Specify an Amazon S3 template URL**" and enter the following template location, then **click Next**

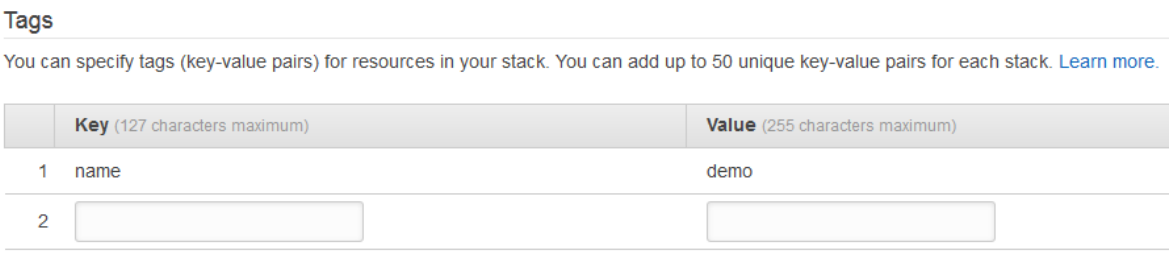
   https://s3-us-west-2.amazonaws.com/elasticache-labs/LandSat/cfn-landsat-lab.json

3. Chose a meaningful name for your stack such as "AWSLoft" and spend some time reviewing the other parameters in this script. Most importantly, you will need to **choose the SSH IP range that will allow you to connect to your EC2 instance** (this is open by default), **the new key you created for this lab,** and a **database username and password for your Amazon RDS instance**. If you don't update these, CloudFormation will use the default values provided, this is acceptable for your demo environment.



Note: Make sure to select your key

4. We now see we have the option of adding "Tags" to our environment. Tags provide a way to simplify management of the resources in your infrastructure by letting you specify key/value pairs containing information you consider important about this stack. Once you have added any tags you wish to, click Next



5. On this next screen we'll see a summary of what we are creating including the "Stack Description" that is taken from the template. The "Template" section contains the stack Name, S3 bucket location that this AWS CloudFormation template file will be uploaded to as part of creating this and the description of this stack. Clicking on the word "Cost" next to "Estimated Cost" would take us to the AWS Monthly Web Calculator and could give us a break down of any costs caused by the creation of this stack's resources. This first stack actually would cost us nothing to turn up right now, so the calculator won't show any monthly cost.

   This should be all good to go, so let's click on "Create" once more to start creating this stack.

6. The stack will begin to run. This may take 15-20 minutes, so rather than waiting, let's skip to Task 2 and check on this every few minutes to see updates.

	Stack Name	Created Time	Status
☑	AWSLoft	2018-04-17 08:11:38 UTC-0400	CREATE_IN_PROGRESS

Task 2 – Review & Configure Environment

1.1 Review AWS CloudFormation

For convenience, needed configuration information on your newly built environment can be viewed by navigating to the Outputs tab, such as the below example. Take a few minutes to review this information. You will be referencing it throughout this lab.

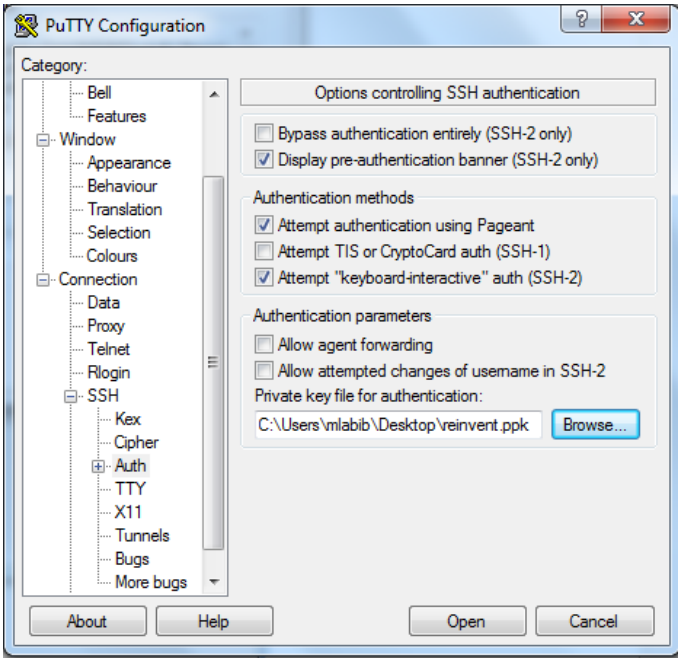
Overview | **Outputs** | Resources | Events | Template | Parameters | Tags | Stack Policy | Change Sets

Key	Value	Description
EC2IP	52.206.142.166	Application URL
ElastiCacheRedisEndpoint	redistest.foio87.0001.use1.cache.amazonaws.com	ElastiCache Endpoint
RDSEndpoint	ardszfsdo28l5r.cig3pkfdvcjv.us-east-1.rds.amazonaws.com	RDS Endpoint
RDSPort	3306	RDS Port
VPCId	vpc-930871e8	Lab VPCId
RDSSG	sg-78ed6a31	RDS SecurityGroup Id
ElastiCacheSG	sg-6cf47325	ElastiCache SecurityGroup Id
ElastiCachePort	6379	ElastiCache Port
AppSG	sg-6bf47322	Application SecurityGroup Id

1.2 SSH into your EC2 instance

Windows Users

Connect to the EC2 public IP address, adding your key to Putty:



Mac Users

Connect to the EC2 public IP address via SSH, including your key:

ssh –i /path/to/my-key-pair.pem ec2-user@<IP_ADDRESS>

1.3 Review Environment

Note: A command reference file can be found here for the below commands:

https://s3-us-west-2.amazonaws.com/elasticache-labs/LandSat/lab-commands.txt

1. Let's ensure your Redis connection is working:

   Type these commands on the command line

   - $ redis-cli -h [ElastiCacheRedisEndpoint] (see cloudformation outputs)
     (Example of successful connection)
     redistest.foio87.0001.use1.cache.amazonaws.com:6379>
     (CTRL+D to disconnect)

2. Let's ensure your MySQL connection is working: (below example)

   Type these commands on the command line

   - $ mysql -h [RDSEndpoint] -u databaseadmin –p (see cloudformation outputs)
   - Enter password: (default: AWSNYLoft)
     (Example of successful connection)
     Welcome to the MySQL monitor. Commands end with ; or \g.
     Your MySQL connection id is 9999
     Server version: 5.6.27-log MySQL Community Server (GPL)
     Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

- mysql> (CTRL+D to disconnect)

3. Download and prepare LandSat scenes

   See documentation at https://aws.amazon.com/public-data-sets/landsat/

   - From your EC2 instance, download the Landsat scenes
     $ wget http://landsat-pds.s3.amazonaws.com/scene_list.gz
   - Unzip the scene list
     $ gunzip scene_list.gz
   - Trim the list to the last 250,000 scenes
     $ cp scene_list scene_list.orig
     $ tail -n 250000 scene_list.orig > scene_list

4. Load to MySQL

   - Log into the mysql> console (step 2 example)
   - Create a landsat database
     mysql> CREATE DATABASE landsat;
     mysql> USE landsat;
   - Create the scene_list table
     mysql> CREATE TABLE scene_list (entityId VARCHAR(64), acquisitionDate DATETIME,cloudCover DECIMAL(5,2),processingLevel VARCHAR(8),path INT,row INT,min_lat DECIMAL(8,5),min_lon DECIMAL(8,5),max_lat DECIMAL(8,5),max_lon DECIMAL(8,5),download_url VARCHAR(128));
   - Load the landsat data
     mysql> LOAD DATA LOCAL INFILE 'scene_list' INTO TABLE scene_list FIELDS TERMINATED BY ',';
     Query OK, 250000 rows affected (2.84 sec)
     Records: 250000 Deleted: 0 Skipped: 0 Warnings: 0

5. Load data into Redis

   - Download file sql2redis.py
     $ wget https://s3-us-west-2.amazonaws.com/elasticache-labs/LandSat/sql2redis.py
   - Update (sql2redis.py) with your ElastiCache and RDS endpoints and credentials.
     (e.g. vi sql2redis.py)
   - After updating the file, run the python script to cache data in Redis
     $ python sql2redis.py

6. Ensure data was loaded into Redis

   - $ redis-cli -h [ElastiCacheRedisEndpoint] (see cloudformation outputs)
     (Check size of Redis db)

- redistest.foio87.0001.use1.cache.amazonaws.com:6379> DBSIZE
  (integer) 246473

You are now ready to start executing our tests!

Task 5 – Run Queries

1. Run SQL query
   - Log in to your MySQL node and run a query
     $ mysql -h [RDSEndpoint] -u databaseadmin –p (see cloudformation outputs)
     Enter password: (default: AWSNYLoft)

```
mysql> USE landsat;

mysql> SELECT DISTINCT(a.entityId) AS Id, a.cloudCover
 FROM scene_list a
 INNER JOIN (
 SELECT entityId, acquisitionDate
 FROM scene_list
 WHERE acquisitionDate > (
        SELECT MAX(acquisitionDate)
        FROM scene_list
        WHERE acquisitionDate < CURDATE() - INTERVAL 1 YEAR
 ) ) b ON a.entityId = b.entityId AND a.acquisitionDate = b.acquisitionDate
 WHERE cloudCover < 50
 ORDER BY Id;
```

mysql> (CTRL+D to disconnect)

   - This generates a list of all the satellite images during the last year which have less than 50% cloud cover
   - Note how long it takes to get an answer (e.g. 6311 rows in set (0.45 sec))

2. Run Redis query
   - Log in to your Redis (e.g redis-cli -h [ElastiCacheRedisEndpoint])

```
redis> zunionstore temp:cCov 1 cCov
```

```
redis> zremrangebyscore temp:cCov 50 inf

redis> zunionstore temp:acqDate 1 acqDate

redis> zremrangebyscore temp:acqDate 0 1523923200 (use date epoch)

redis> zinterstore out 2 temp:cCov temp:acqDate WEIGHTS 1 0
```

(CTRL+D to disconnect)

- Again, this generates a list of all the satellite images during the last year which have less than 50% cloud cover
- Note how long it takes to get an answer

Pre-lab

We will be doing most of our work today in the Amazon Web Services console and via SSH. The console is found here: https://console.aws.amazon.com.

Pre-lab step #1 Log into the Console

1. Setup an AWS account
2. Install PuTTY/PuTTYgen [Windows]
   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Pre-lab step #2 Create Key Pair and set up SSH client

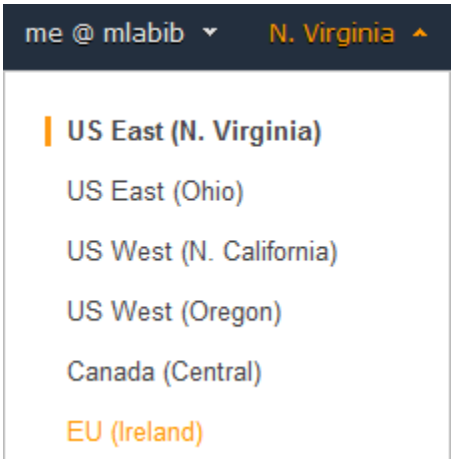
Open your browser and go to https://console.aws.amazon.com

It is very important to note, that you need to keep track of where you download this file! Do not delete it unless you have completed the lab and no longer plan to use the key.

It is also very important that you make sure you are in the correct region when you create your Key Pair.

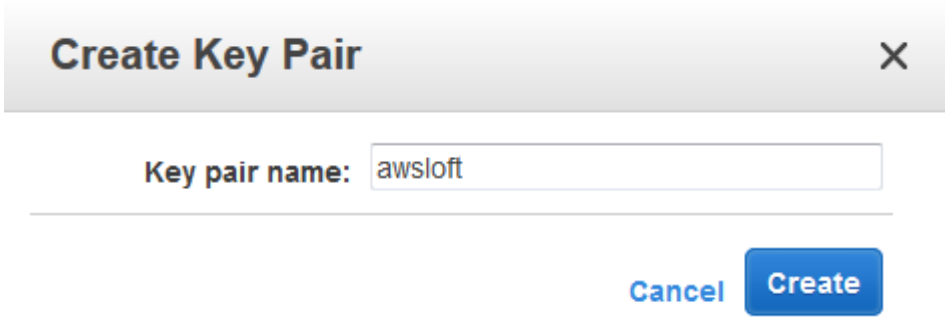
1. Now, from the main dashboard in the Console, find the EC2 link. From the main services lists page , you will see it listed under Compute, click EC2

2. Let's confirm we are in the right region. Everyone should be launching in the us-east-1 Virginia region today. If you are not, go to the top right corner of the site, and click the region drop down and select US East (N.Virginia):



3. Once you are finally at the Amazon EC2 Console Dashboard, look on the left panel for the Key Pairs link and **click it**.

4. Click on "Create Key Pair". Name it **"awsloft"** so you can easily remember it and click **Create**



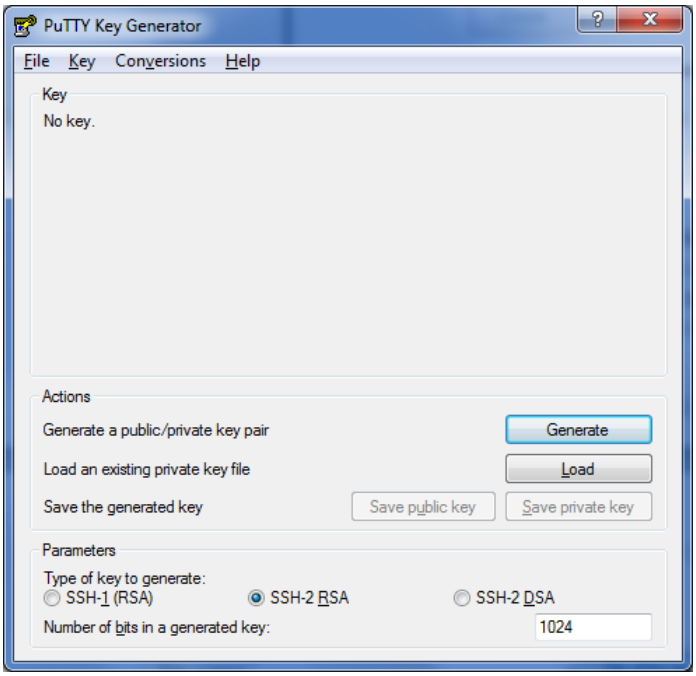
5. Download the file to your desktop, or some other location you will remember. You will need this file again later in the lab.

Windows Users

Note: Windows Users will need to follow the next steps to make the Key Pair we just created usable with PuTTY. If PuTTY is not your SSH client, you can ignore these steps and use the .pem file as it is. You are all done with the pre-lab setup if you don't need to follow these steps.

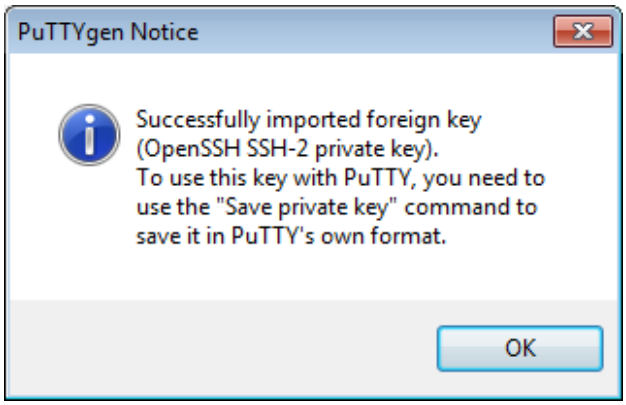
1. Now we need to convert the .PEM file we just downloaded into a format that can be used by our SSH tool PuTTY.
2.
    1) Start **PuTTYgen** (for example, from the Start menu, click All Programs > PuTTY > PuTTYgen).

2) Click Load and browse to the location of the private key file that you want to convert (e.g., awsloft.pem).



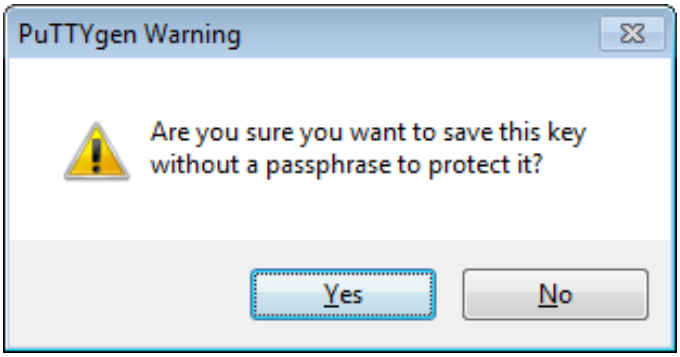
3) By default, PuTTYgen displays only files with extension .ppk; you'll need to change that to display files of all types in order to see your .pem key file. The private key file must end with a newline character or PuTTYgen cannot load it correctly.

4) Select your .pem key file and click Open. PuTTYgen displays the following message.



When you click OK, PuTTYgen displays a dialog box with information about the key you loaded, such as the public key and the fingerprint. The keys that Amazon EC2 generates are 1024-bit SSH-2 RSA keys.

5) Click **Save private key** to save the key in PuTTY's format. PuTTYgen asks if you want to save the key without a passphrase. Click "**Yes**". Normally you would want to make sure that the keys used to access your instances were as secure as possible, but for today's lab we'll skip the extra complication.



6) Use the same name for the key that you used for the Key Pair (for example, "**awsloft**"). PuTTY automatically adds the .ppk file extension. Close PuTTYgen.

Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client. Later today when you start firing up instances, you'll use PuTTY to connect to the ones that are Internet accessible. We will look at how to do this in PuTTY a little later. That's it! We should now be all set up and ready to go on with the rest of the lab.

Mac Users

On macOS, you can use the SSH client that is included by default. You will need to change permissions on the downloaded key pair such that it is not publicly viewable:

```
chmod 400 /path/to/my-key-pair.pem
```