Accelerate Database Development and Testing with Amazon Aurora

Lab Guide

[PART 1. Setup Environment 2](#_Toc45737806)

[Task 1.1 – Sign in to Management Console, Select Region and Create Key Pair 2](#_Toc45737807)

[Task 1.2 - Creating a Stack using CloudFormation 4](#_Toc45737808)

[Task 1.3 - Connecting to the workstation EC2 instance 8](#_Toc45737809)

[Task 1.4 – Set up the AWS CLI and seed the DB Cluster 8](#_Toc45737810)

[PART 2. Cluster Endpoints and Auto Scaling 9](#_Toc45737811)

[Task 2.1 – Running a read-only workload 9](#_Toc45737812)

[PART 3. Cloning database 12](#_Toc45737813)

[Task 3.1 - Creating a Clone 12](#_Toc45737814)

[PART 4. Performance Insights 16](#_Toc45737815)

[Task 4.1 – Generate Load on Your Database Cluster 16](#_Toc45737816)

[PART 5. Delete Lab Env 17](#_Toc45737817)

[APPENDIX 18](#_Toc45737818)

[Appendix 1 – Setting up PuTTY and connecting via SSH 18](#_Toc45737819)

# PART 1. Setup Environment

Please log into the AWS Management Console using the credentials provided to you on the separate card.

You will be using the **Beijing (cn-north-1)** region.

In this part of the lab you will leverage AWS CloudFormation to provision an Aurora MySQL 5.7 compatible database cluster, along with a Linux EC2 instance to be used as a workstation. You will connect to the workstation using SSH.

The environment deployed using CloudFormation includes several components, as listed below. Please download the CloudFormation template (instructions below) and review it for more details.

1. Amazon VPC network configuration with public and private subnets
2. Database subnet group and relevant security groups for the cluster and workstation
3. Amazon EC2 instance configured with the software components needed for the lab
4. Roles with access permissions for the workstation and cluster permissions for enhanced monitoring, S3 access and logging
5. Custom cluster and DB instance parameter groups for the Amazon Aurora cluster, enabling logging and performance schema
6. Amazon Aurora DB cluster with 2 nodes: a writer and read replica
7. Read replica auto scaling configuration
8. AWS Systems Manager command document to execute a load test

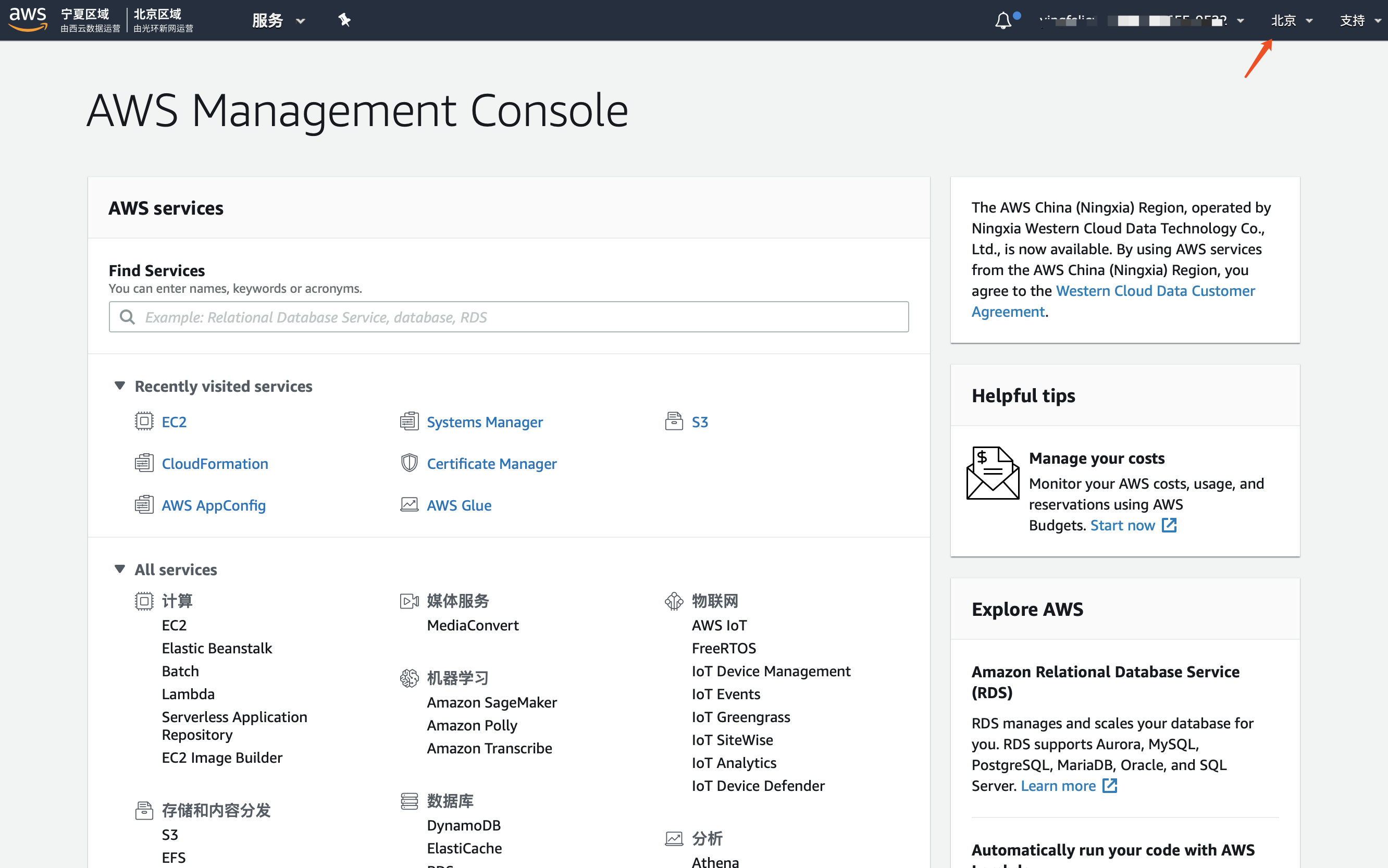
## Task 1.1 – Sign in to Management Console, Select Region and Create Key Pair

1. Type the **Console URL**, **Login in your account**
2. Enter the **Username** and **Password**, click **Sign In**.

图形用户界面, 应用程序

描述已自动生成

1. Ensure the **Beijing(cn-north-1)** region is selected in the top right corner, if not use that dropdown to choose the correct region



1. Open the **Key Pairs** section of the EC2 service console, using this link:

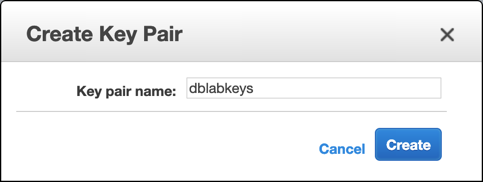
https://console.amazonaws.cn/ec2/v2/home?region=cn-north-1#KeyPairs:sort=keyName

1. Ensure you are still in the correct region, and click **Create Key Pair**.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Name the key pair “dblabkeys” and then click **Create** and download the file named **dblabkeys.pem** to your computer, save it in a memorable location like your desktop. You will need this file later in the lab.



## Task 1.2 - Creating a Stack using CloudFormation

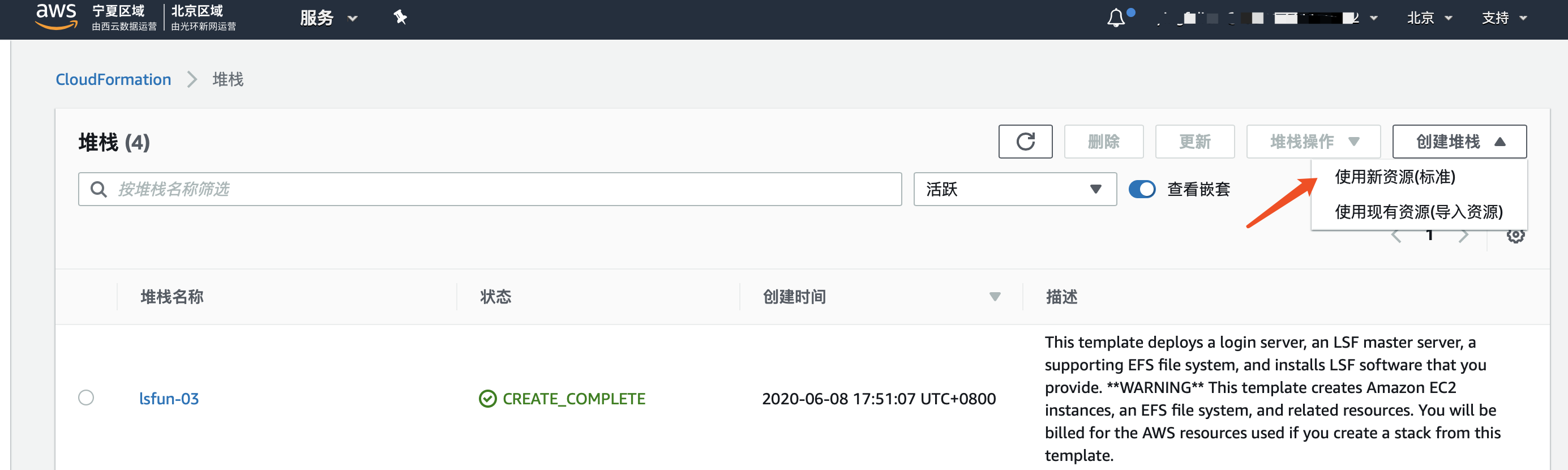
1. Open the **CloudFormation** service console located at

https://console.amazonaws.cn/cloudformation/home?region=cn-north-1#/

1. Click **Create Stack.**

**Notice:** The CloudFormation console has been upgraded recently. Depending on your previous usage of the CloudFormation console UI, you may see the old design or the new design, you may also be presented with a prompt to toggle between them. In this lab we are using the **new design** for reference, although the steps will work similarly in the old console design as well, if you are more familiar with it.

and then click **Next**.



Select the template file and next

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. In the field named **Stack Name**, enter the value “dblabstack”, select the **ec2KeyPair** value as “dblabkeys” (the key pair you have created previously) and then click **Next**.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

On the **Configure stack options** page, leave the defaults as they are, scroll to the bottom and click **Next**.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. On the **Review dblabstack** page, scroll to the bottom, check the box that reads: **I acknowledge that AWS CloudFormation might create IAM resources with custom names** and then click **Create**.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. The stack will take approximatively 20 minutes to provision, you can monitor the status on the **Stack detail** page. You can monitor the progress of the stack creation process by refreshing the **Events** tab. The latest event in the list will indicate **CREATE\_COMPLETE** for the **dblabstack** resource.  
     
   In the meantime we will discuss some important considerations when architecting and automating the deployment of Aurora clusters.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Once the status of the stack is **CREATE\_COMPLETE**, click on the **Outputs** tab. The values here will be critical to the completion of the remainder of the lab. Please take a moment to save these values somewhere that you will have easy access to them during the remainder of the lab. The names that appear in the **Key** column are referenced directly in the instructions in subsequent steps, using the parameter format: **[outputKey]**

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

## Task 1.3 - Connecting to the workstation EC2 instance

**For Windows users:** We will use PuTTY and PuTTY Key Generator to connect to the workstation using SSH. If you do not have these applications already installed please use the steps in [**Appendix 1 - Setting up PuTTY and connecting via SSH**](#_Appendix_1_–) below.

**For macOS or Linux users:** You can connect using the following command from a terminal, however you will need to change the permissions of the certificate file first:

**chmod 0600 [path to downloaded .pem file]**

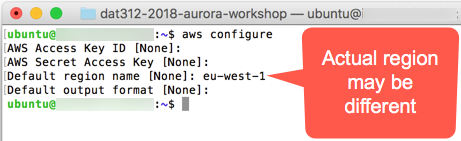
**ssh -i [path to downloaded .pem file] ubuntu@[bastionEndpoint]**

## Task 1.4 – Set up the AWS CLI and seed the DB Cluster

1. Enter the following command in the SSH console to configure the AWS CLI:

**aws configure**

Then select the defaults for everything except the default region name. For the default region name, enter “cn-north-1”.



1. Download 2 sql files

**aws s3 cp s3://aws-bjs-lab/create\_sbtest1.sql .**

**aws s3 cp s3://aws-bjs-lab/load\_data\_s3.sql .**

1. Connect to the Aurora database using the following command:

**mysql -h [clusterEndpoint] -u masteruser -p mylab**

Unless otherwise specified the cluster master username is **masteruser** and the password is **Password1**

1. Run the following queries on the database server, they will create a table, and load data from S3 into it:

**4a. Execute sql file1**

**source create\_sbtest1.sql**

**Create\_sbtest1.sql’s content as below:**

DROP TABLE IF EXISTS `sbtest1`;

CREATE TABLE `sbtest1` (  
 `id` int(10) unsigned NOT NULL AUTO\_INCREMENT,  
 `k` int(10) unsigned NOT NULL DEFAULT '0',  
 `c` char(120) NOT NULL DEFAULT '',  
 `pad` char(60) NOT NULL DEFAULT '',  
PRIMARY KEY (`id`),  
KEY `k\_1` (`k`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**4b. Execute sql file 2**

**source load\_data\_s3.sql**

**load\_data\_s3.sql’s content as below:**

LOAD DATA FROM S3 MANIFEST  
's3://aws-bjs-lab/sample.manifest'  
REPLACE  
INTO TABLE sbtest1  
CHARACTER SET 'latin1'  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\r\n';

# PART 2. Cluster Endpoints and Auto Scaling

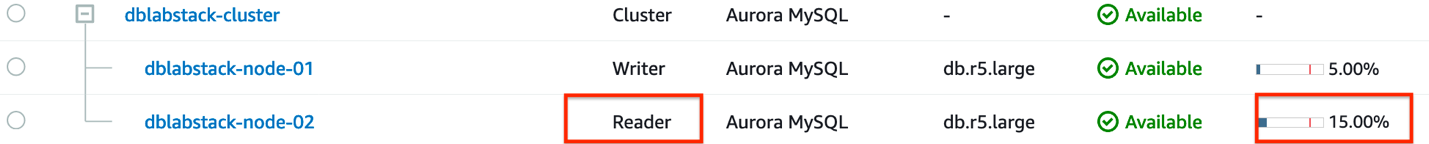
In this part we will explore the cluster endpoints and how auto scaling of read replicas operates.

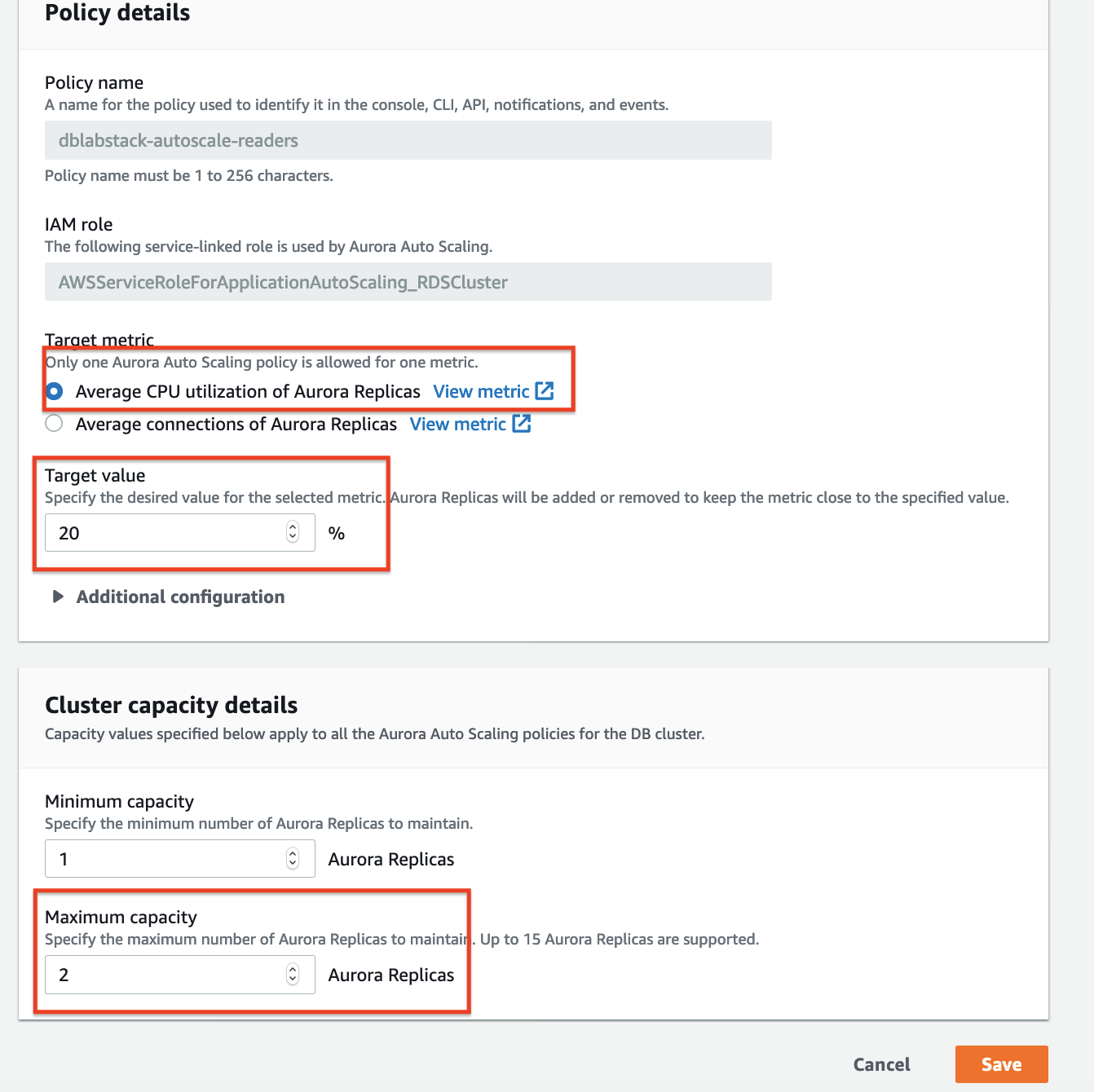
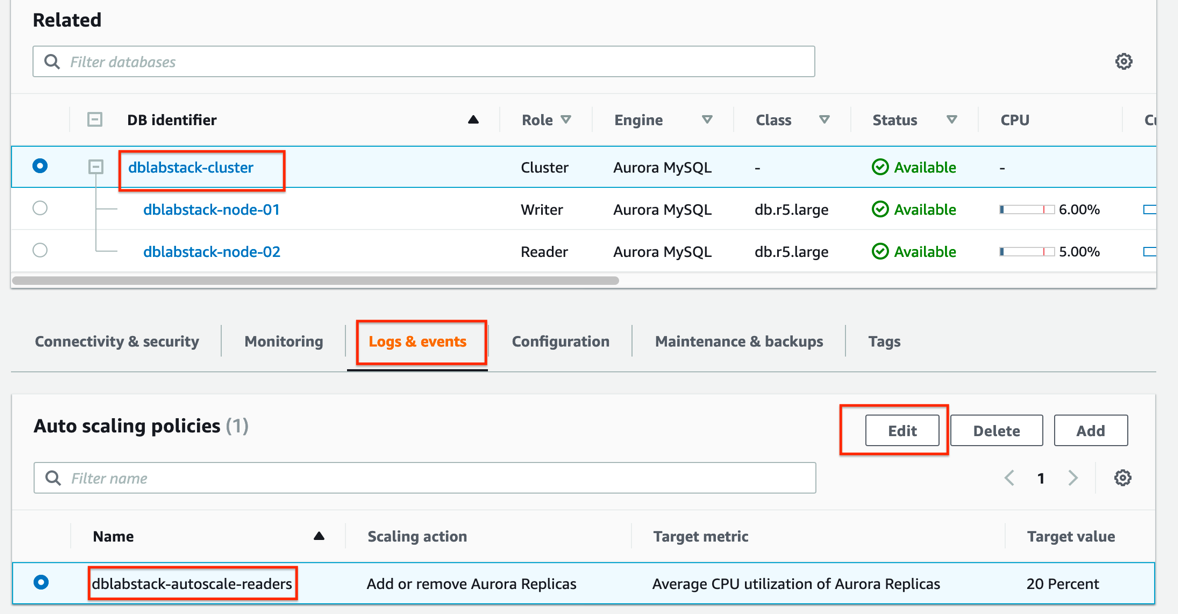
## Task 2.1 – Running a read-only workload

1. On the bastion host, execute the following statement:

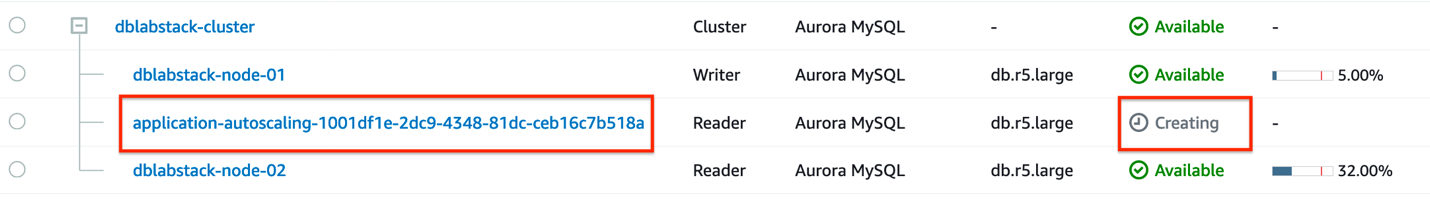
**python loadtest.py [readerEndpoint]**

1. Open the **Amazon RDS** service console located at: https://console.amazonaws.cn/rds/home?region=cn-north-1 .
2. Take note that the reader node is currently receiving load. It may take a minute or more for the metrics to fully reflect the incoming load.





1. After a few minutes return to the list of instances and notice that a new reader is being provisioned to your cluster.



1. Once the replicas are added, note that the they are starting to receive load.



1. You can now type CTRL+C on the bastion host to end the read load, if you wish to. After a while the additional readers will be removed automatically.

# PART 3. Cloning database

## Task 3.1 - Creating a Clone

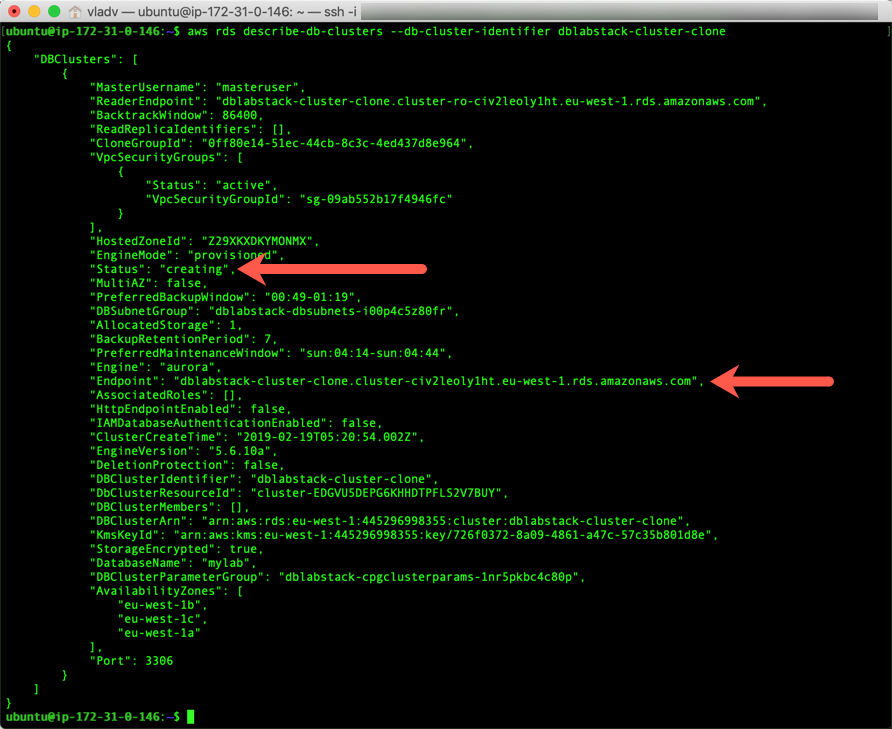
1. On the bastion host, enter:

**aws rds restore-db-cluster-to-point-in-time --restore-type copy-on-write --use-latest-restorable-time --source-db-cluster-identifier [clusterName] --db-cluster-identifier [clusterName]-clone --vpc-security-group-ids [dbSecurityGroup] --db-subnet-group-name [dbSubnetGroup]**

1. Next, to check the status of the creation of your clone, enter the following command on the bastion host. The cloning process can take several minutes to complete. See the example output below.

**Note:** This step will create the Aurora DB cluster itself, but without any compute nodes. You will will add a computer node in the next step.

**aws rds describe-db-clusters --db-cluster-identifier [clusterName]-clone**



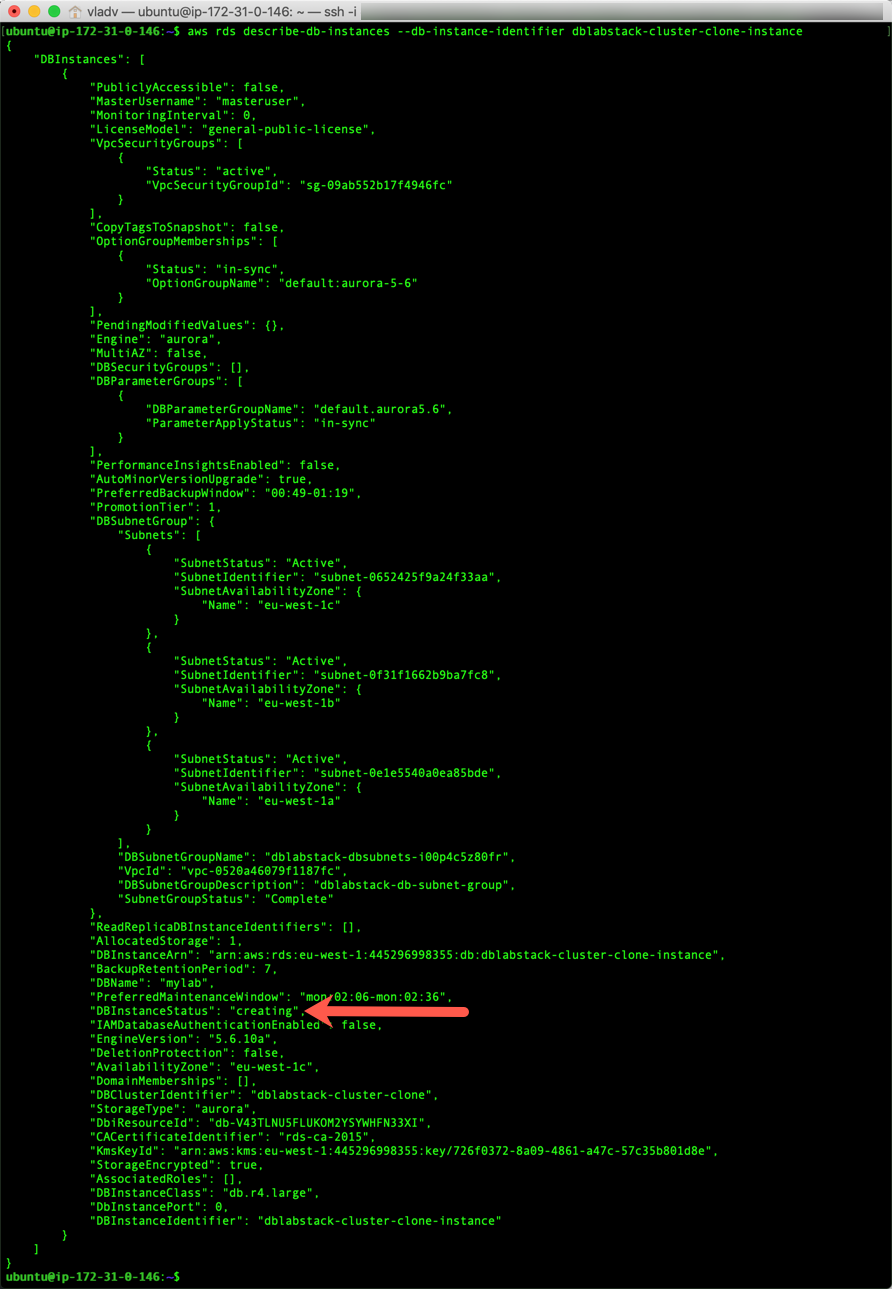
1. Take note of both the **“Status”** and the **“Endpoint.”** Once the **Status** becomes **available**, you can add an instance to the cluster and once the instance is added, you will want to connect to the cluster via the **Endpoint** value. To add an instance to the cluster once the status becomes **available**, enter the following:

**aws rds create-db-instance --db-instance-class db.r5.large --engine** **aurora-mysql --db-cluster-identifier [clusterName]-clone --db-instance-identifier [clusterName]-clone-instance**

1. To check the creation of the instance, enter the following at the command line:

**aws rds describe-db-instances --db-instance-identifier [clusterName]-clone-instance**

1. Once the **DBInstanceStatus** changes from **creating** to **available**, you have a functioning clone. Creating a node in a cluster also takes several minutes.



1. Once your instance is created, connect to the instance using the following command:

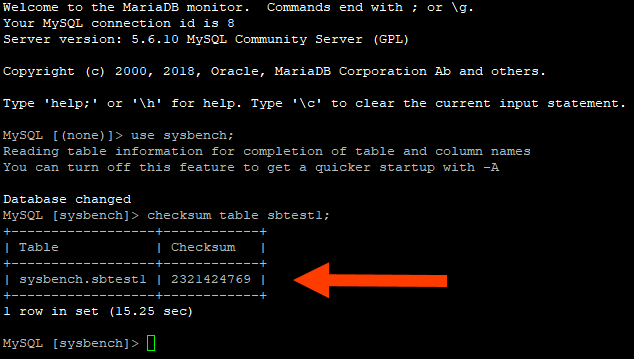
**mysql -h [cluster endpoint of clone cluster] -u masteruser -p mylab**

**Note:** the master user account credentials will be the same as with the source of the cloned cluster. If you customized the CloudFormation template and changed the values, use the customized username and password.

1. In order to verify that the clone is identical to the source, we will perform a checksum of the sbtest1 table using the following:

**checksum table sbtest1;**

1. The output of your commands should look similar to the example below:



1. Please take note of the value for your specific clone cluster.
2. Next, we will disconnect from the clone and connect to the original cluster with the following:

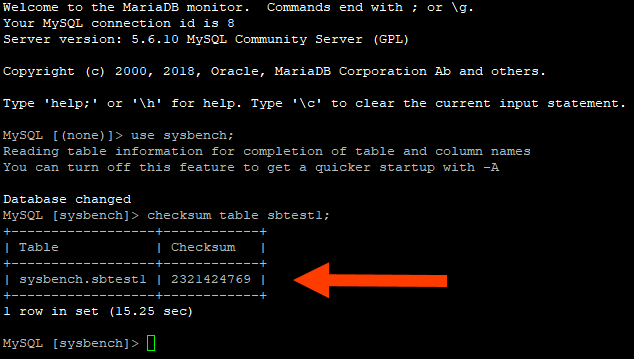
**quit;**

**mysql -h [clusterEndpoint] -u masteruser -p mylab**

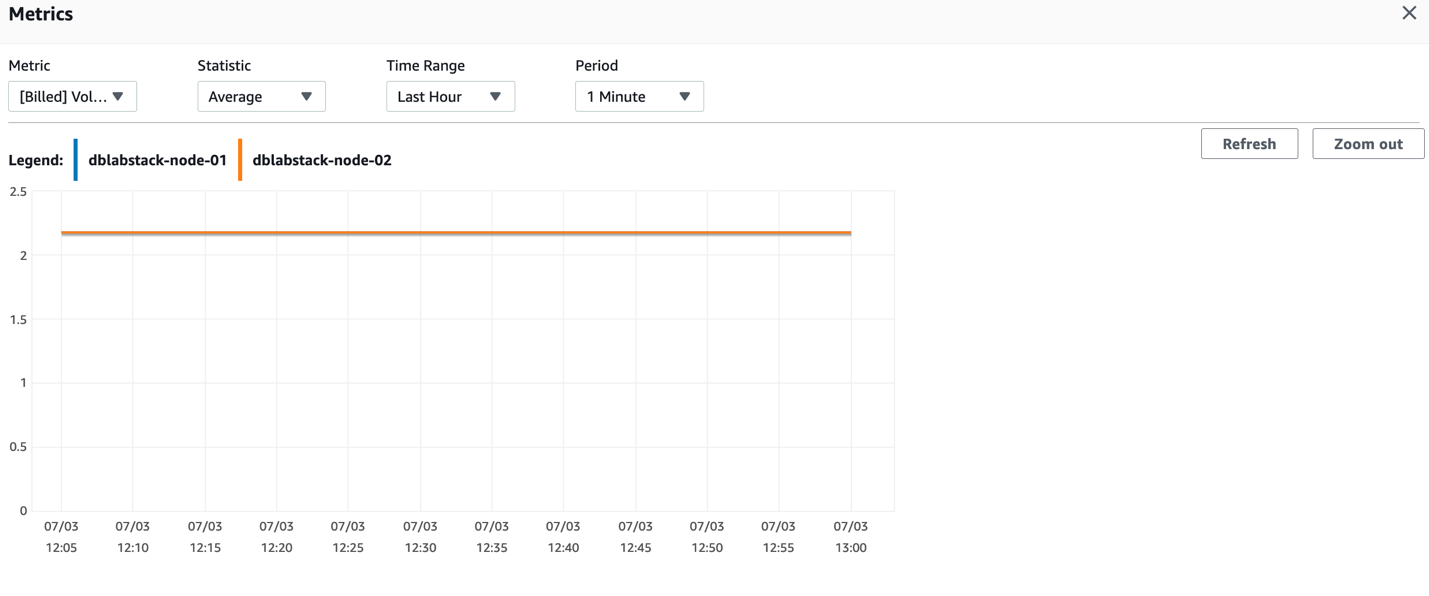
1. Next, we will execute the same commands that we executed on the clone:

**checksum table sbtest1;**

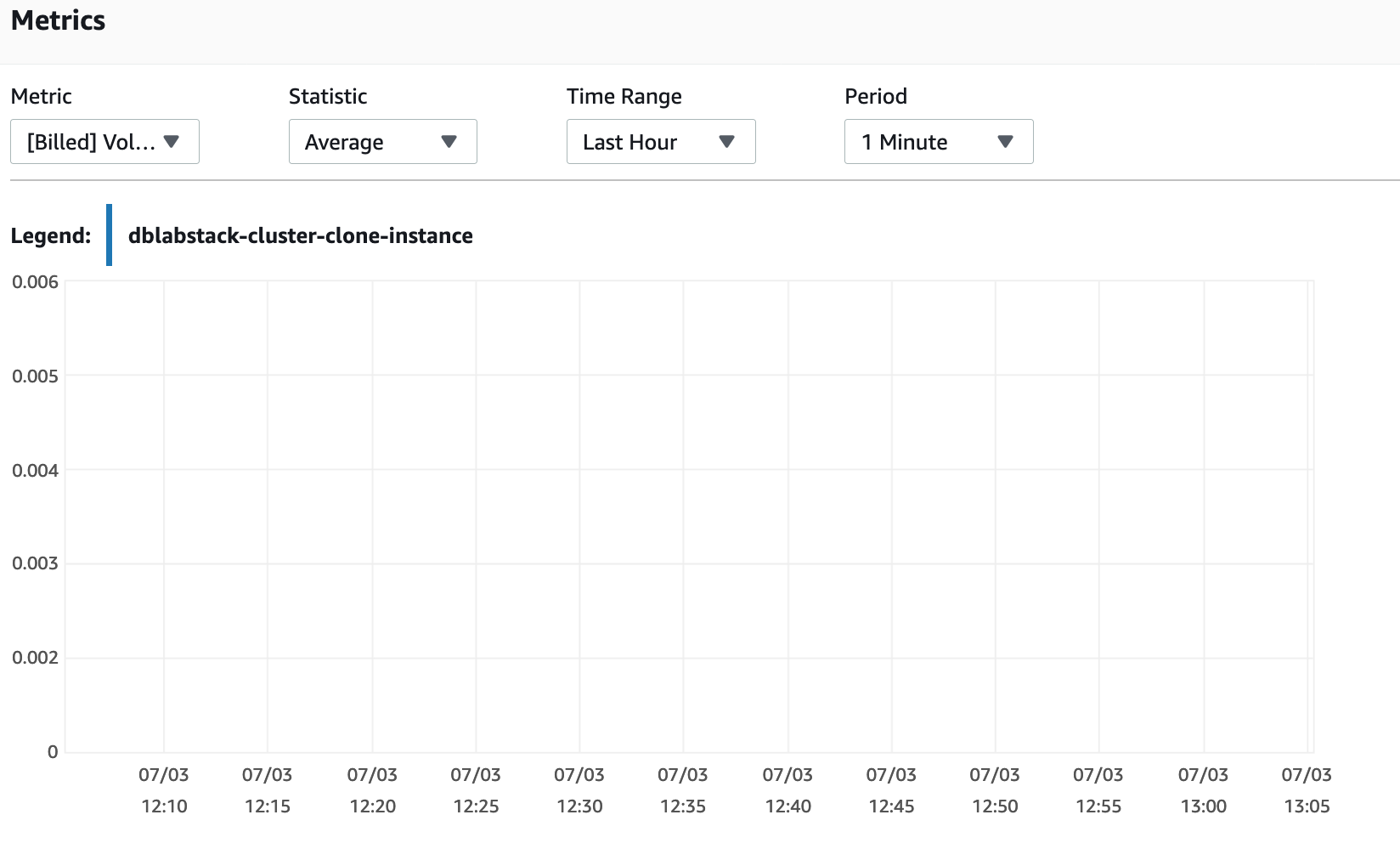
1. Please take note of the value for your specific source cluster. The checksum should be identical.



**Observe the metric “[Billed] Volume Bytes Used(GiB)”of dblabstack-cluster**



**Observe the metric “[Billed] Volume Bytes Used(GiB)”of dblabstack-cluster-clone**



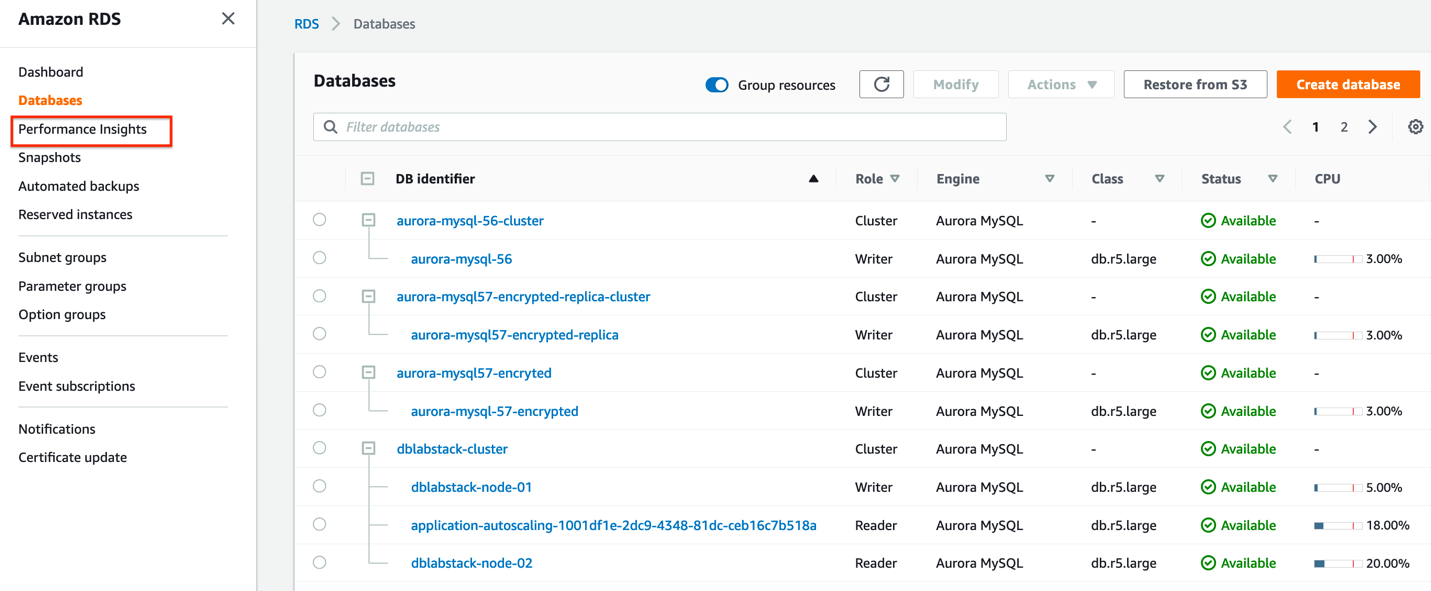
# PART 4. Performance Insights

## Task 4.1 – Generate Load on Your Database Cluster

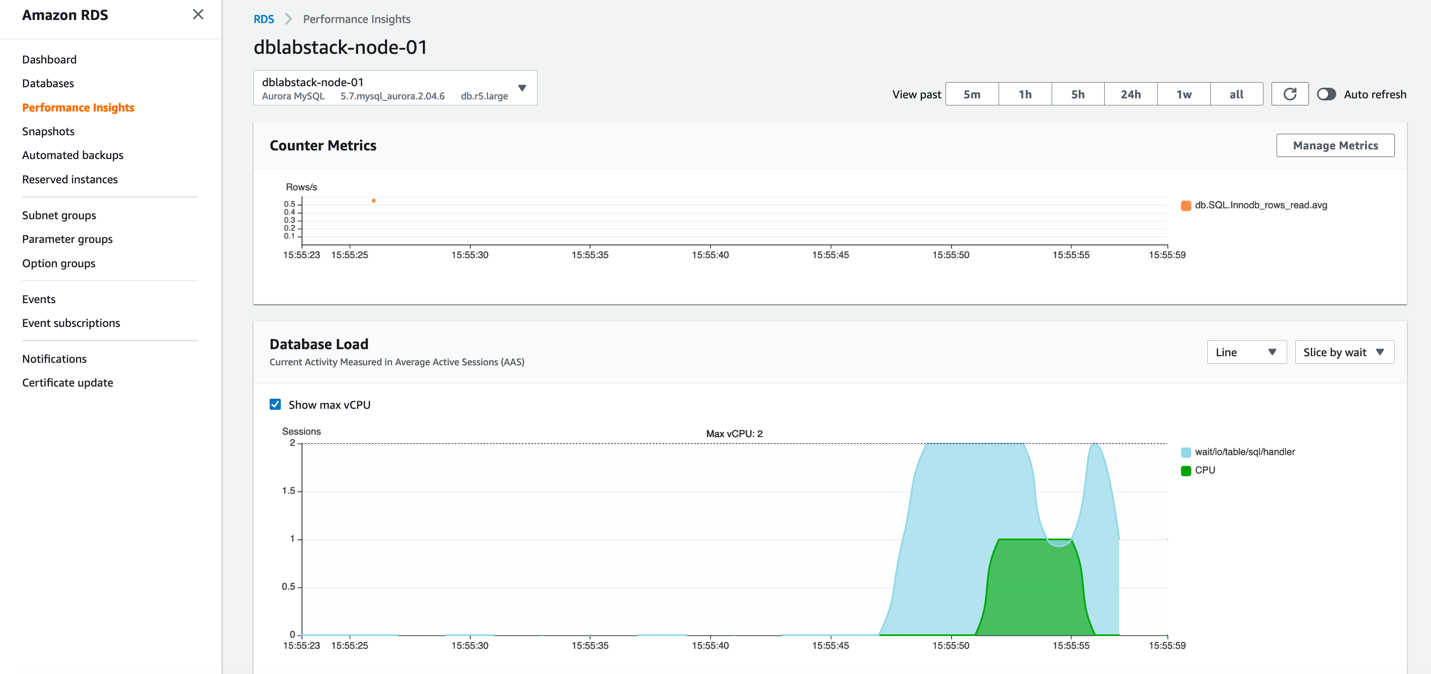
1. You will use [Percona’s TPCC-like benchmark script](https://github.com/Percona-Lab/sysbench-tpcc) based on sysbench to generate load. For simplicity we have packaged the correct set of commands in an AWS Systems Manager Command Document. You will use AWS Systems Manager Run Command to execute the test.
2. On the workstation host, execute the following statement:

**aws ssm send-command --document-name [loadTestRunDoc] --instance-ids [bastionInstance]**

1. The command will be sent to the workstation EC2 instance which will prepare the test data set and run the load test. It may take up to a minute for CloudWatch to reflect the additional load in the metrics.
2. Navigate to the RDS service console (https://console.amazonaws.cn/rds/home?region=cn-north-1) and click on **Performance Insights** in the left side navigation bar.



1. Examine the performance of your DB instance **dblabstack-node-01** using Performance Insights. What conclusions can you reach?



# PART 5. Delete Lab Env

1. Delete Clone Database Cluster

Navigate to Database Cluster **[clusterName]-clone, select [clusterName]-clone-**instance, then delete the instance.

1. Delete CloudFormation (Related resource would be deleted)

# APPENDIX

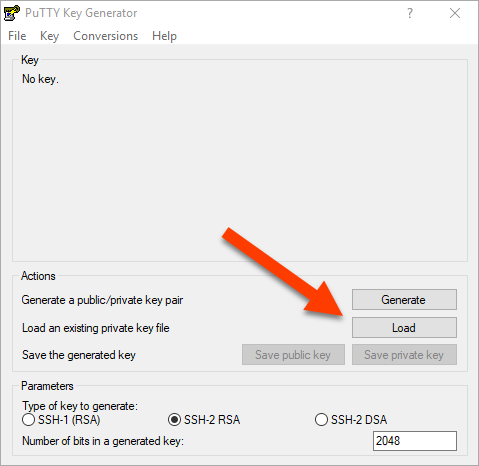
## Appendix 1 – Setting up PuTTY and connecting via SSH

For Windows users, please download **PuTTY** (putty) and the **PuTTY Key Generator** (puttygen) from the following links:

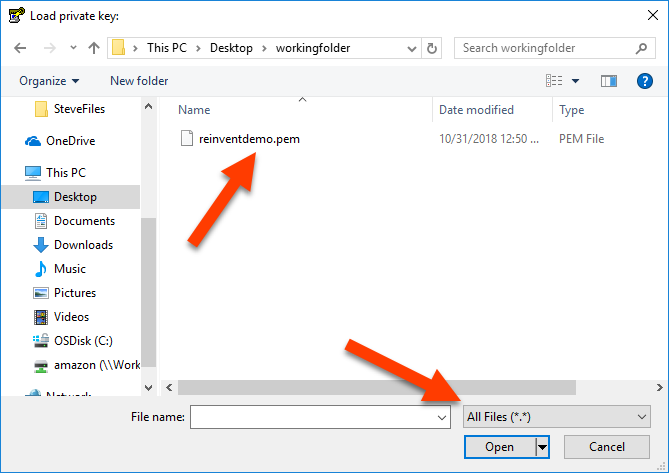
<https://aurora-lab.s3.cn-northwest-1.amazonaws.com.cn/putty.exe>

<https://aurora-lab.s3.cn-northwest-1.amazonaws.com.cn/puttygen.exe>

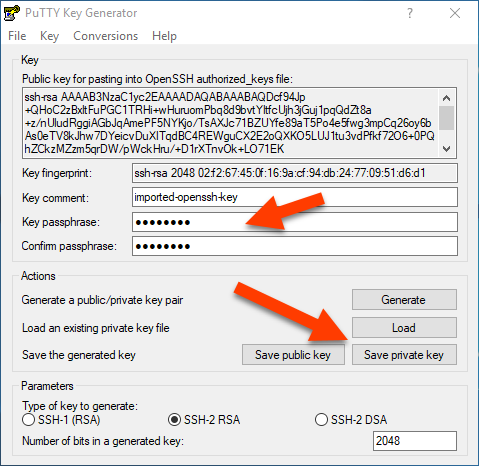
1. Once you have downloaded putty and puttygen, open **puttygen** and click on **Load**.



1. Please make sure that the file filter is set to “All Files (\*.\*) and then select **dblabkeys.pem**.

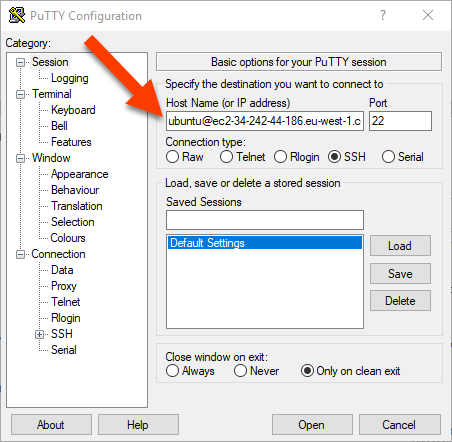


1. Fill in the **Key passphrase** and **Confirm passphrase** fields with a password of your choice that will be used to encrypt your private key and then click **Save private key**. Please use “dblabkeys.ppk” as your new key name.

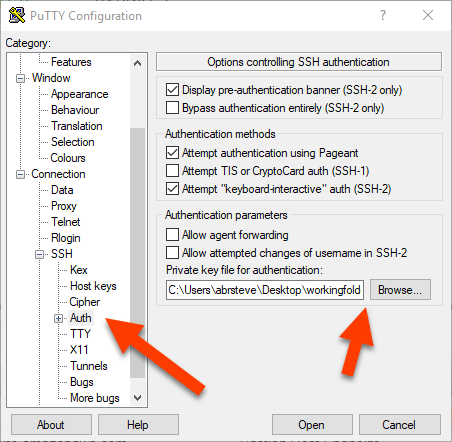


1. Next, open putty and enter into the **Host Name (or IP address)** field the following value:

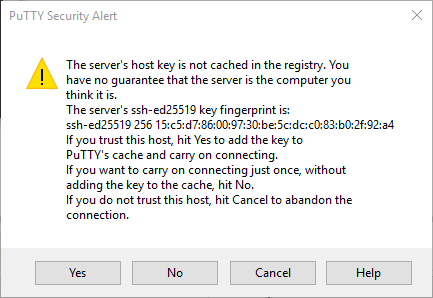
**ubuntu@[bastionEndpoint]**



1. Next, navigate within PuTTY to **Connection** → **SSH** → **Auth** and browse to thedblabkeys.ppk **ppk** file that you created with the PuTTY Key Generator previously, and then click **Open**.



1. When prompted by the PuTTY Security Alert, click **Yes**.



1. Next, enter the password that you configured when you created the **dblabkeys.ppk** private file previously.

