**Auto-Scaling LEMP Application**

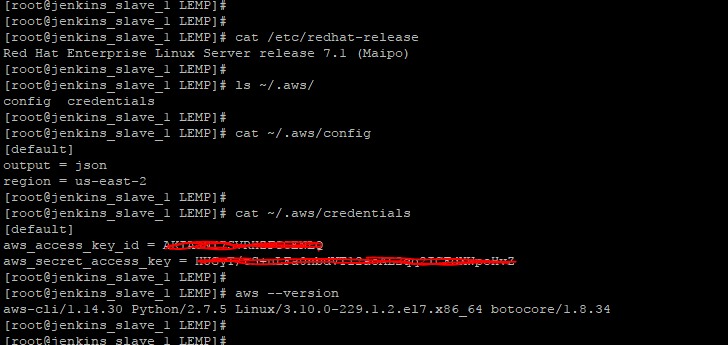
To use the project :

1) First configure a Linux machine with AWS CLI tool (Preferably Rhel 7).

a) Install AWS CLI. <https://docs.aws.amazon.com/cli/latest/userguide/installing.html>

b) Configure the AWS secret credentials, region and default output.

<https://docs.aws.amazon.com/cli/latest/userguide/cli-config-files.html> (Make sure to keep the default output as 'json').

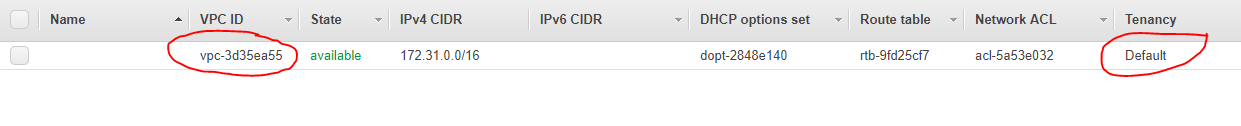


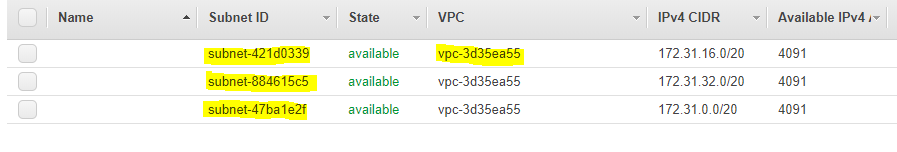
2) AWS Account Configuration

a) Make sure that your AWS Account has one **default VPC** with atleast **2 Subnets** in it.

(This is very much needed coz the script will use your default VPC to launch the instances in and AWS Elastic Load Balancer requires minimum 2 Subnets to be specified.)

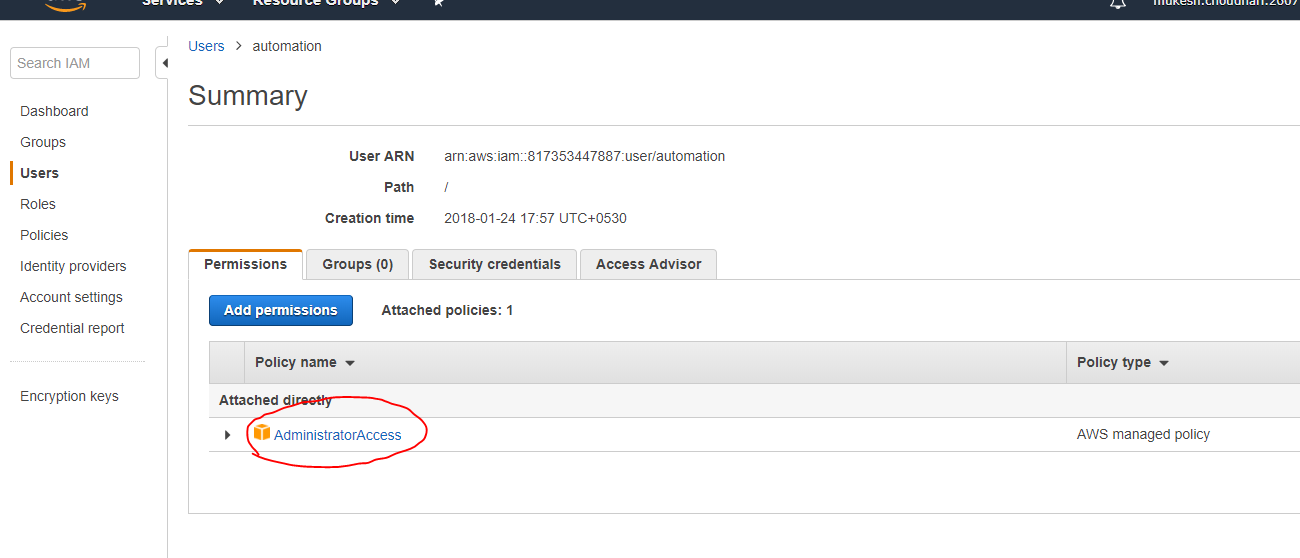
My Default VPC has 3 Subnets.



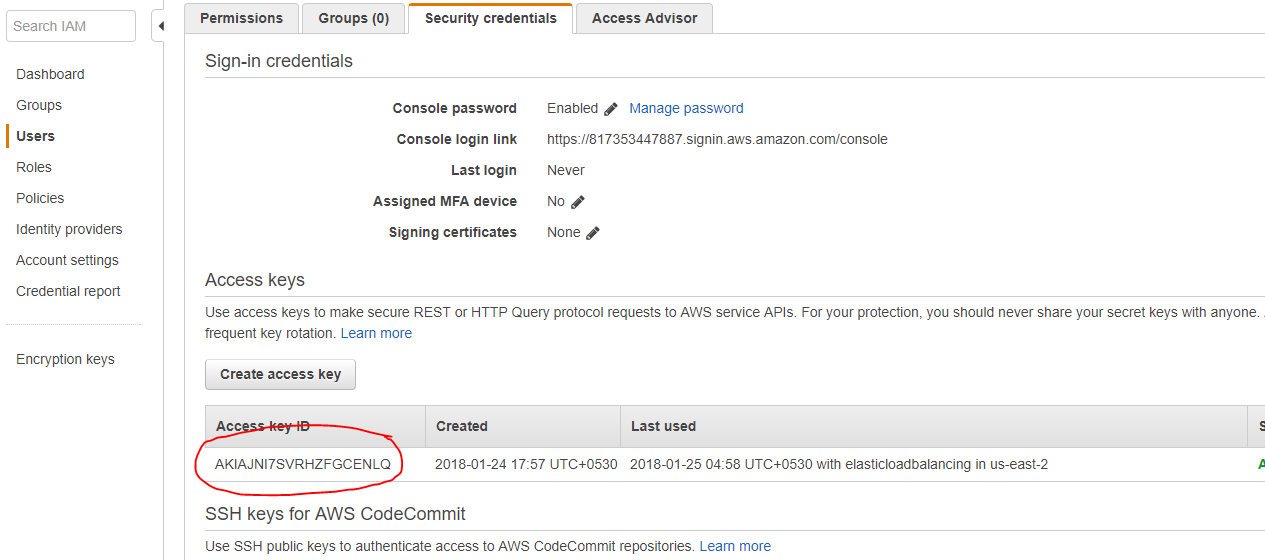


b) Provide Administrator Access to the IAM User that you will be using to configure AWS.

I created an IAM user “automation” and provided it Administrator Access and used the Access Keys to configure AWS CLI.

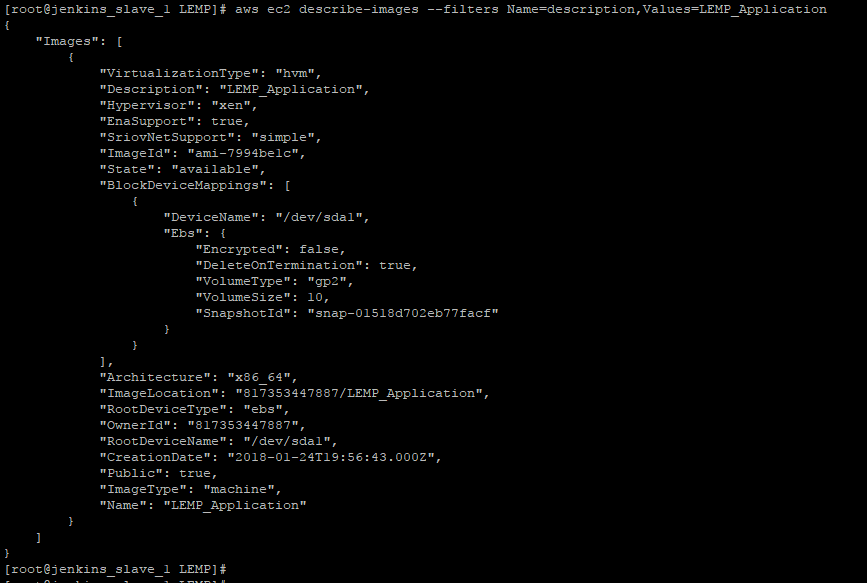


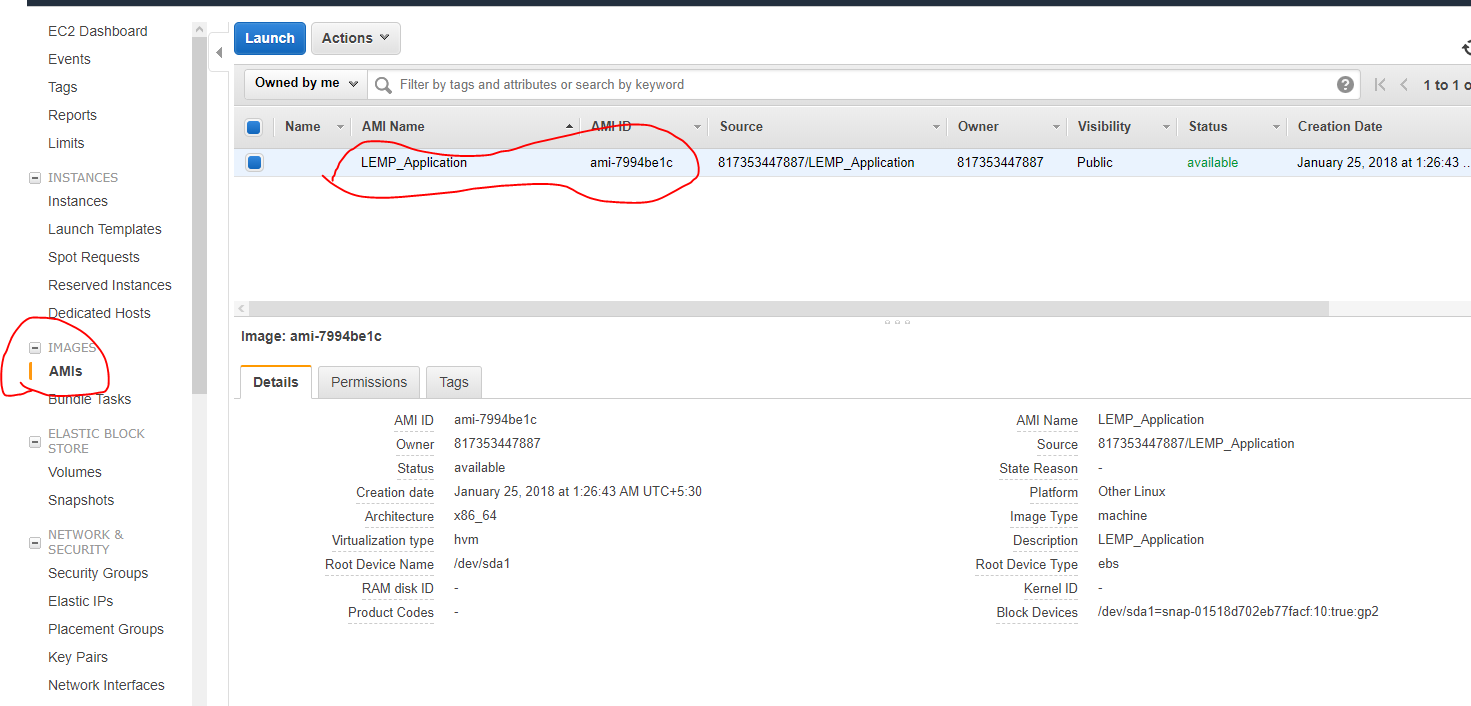
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



c) Make sure my Public AMI is accessible.  
 Fire the below command to verify that my Public Shared AMI is accessible.

**aws ec2 describe-images --filters Name=description,Values=LEMP\_Application**



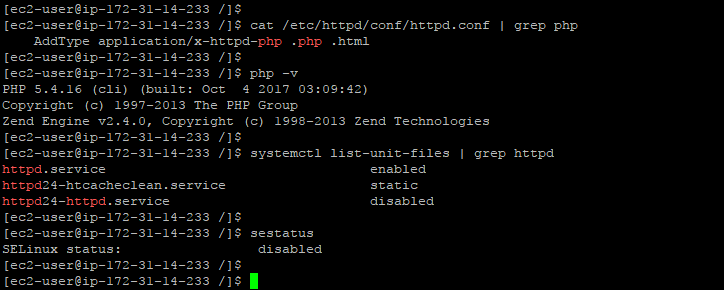


If the AMI is not visible and you don’t get an output similar to above example image then switch the region in the ~/.aws/config file to “us-east-2”.  
  
AMI Details :   
 I) Rhel 7.3

II) Installed tools httpd and php5

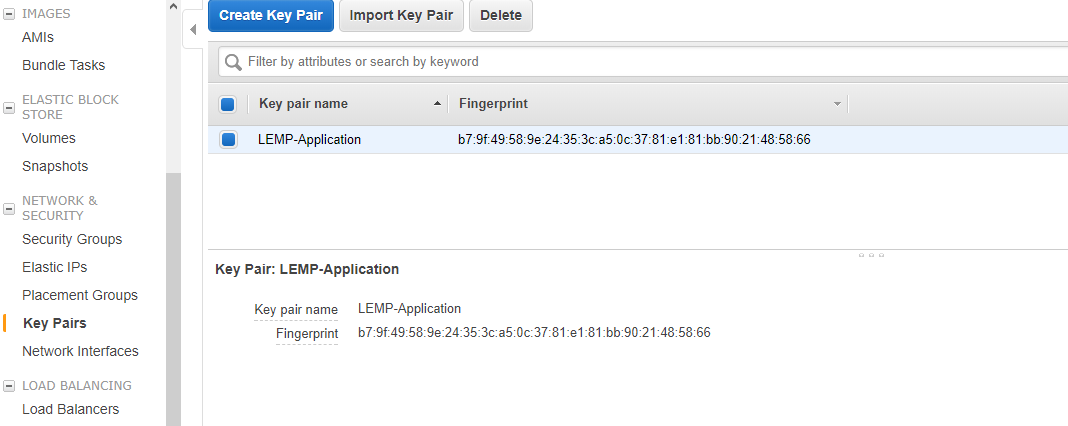
III) httpd service configured to work with php and also start on boot.

IV) SELinux is disabled.



d) Generate a KeyPair for connecting to the Instances.

Generate a KeyPair using the AWS Management Console.



3) Checkout the project from github on the machine you just configured.

GitHub Link : <https://github.com/mukesh-choudhari/LEMP>

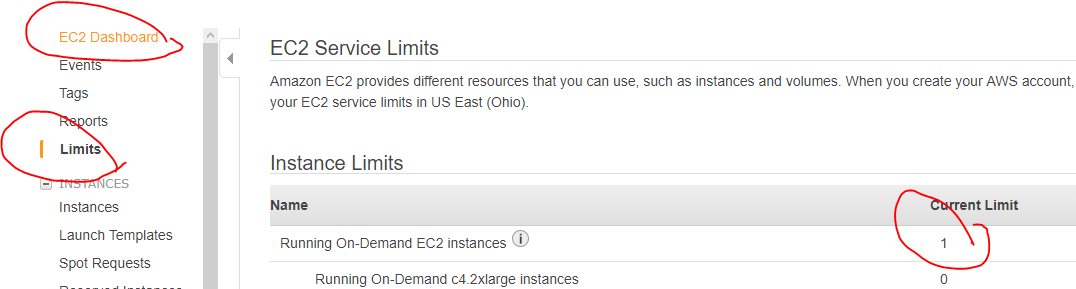
4) Place the downloaded KeyPair in the checkout location LEMP/pem/

5) After verifying all of the above thoroughly, now trigger the script LEMP.sh  
  
 bash LEMP.sh

**Note : If the script fails then all the resources created in AWS will have to be manually deleted before triggering the script again. Rollback steps in case of failure have not been included in the script.**

**Testing the Auto-Scaling Feature**

Since my AWS free account did not allow more than 1 EC2 instance to be in running state. So I could not come up with scripts to load test the Application.  
However there is still a way using which we can Test that AutoScaling works and i.e. by deleting the EC2 instance created by the Auto-Scaling group.  
Since the Auto-Scale policy is to keep atleast 1 machine always running. So post deletion it should automatically create an EC2 machine from the same AMI and Load Balancer should be able to communicate to the newly created EC2 instance.

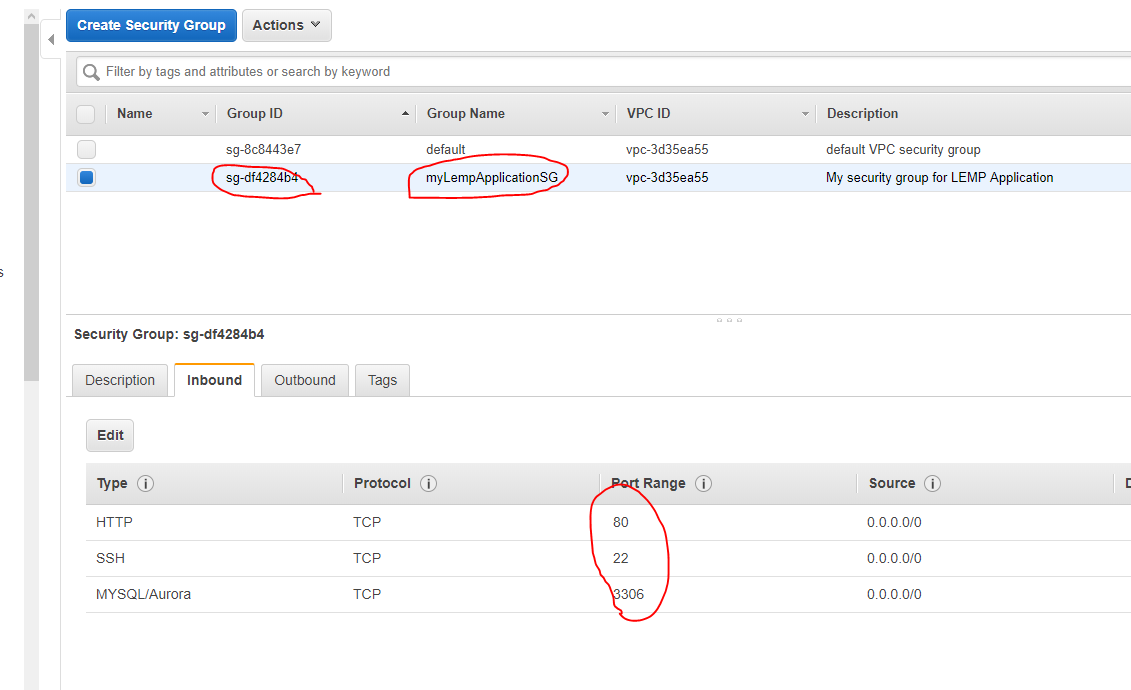


**Screenshots of the successful deployment of Application and Auto-Scaling**

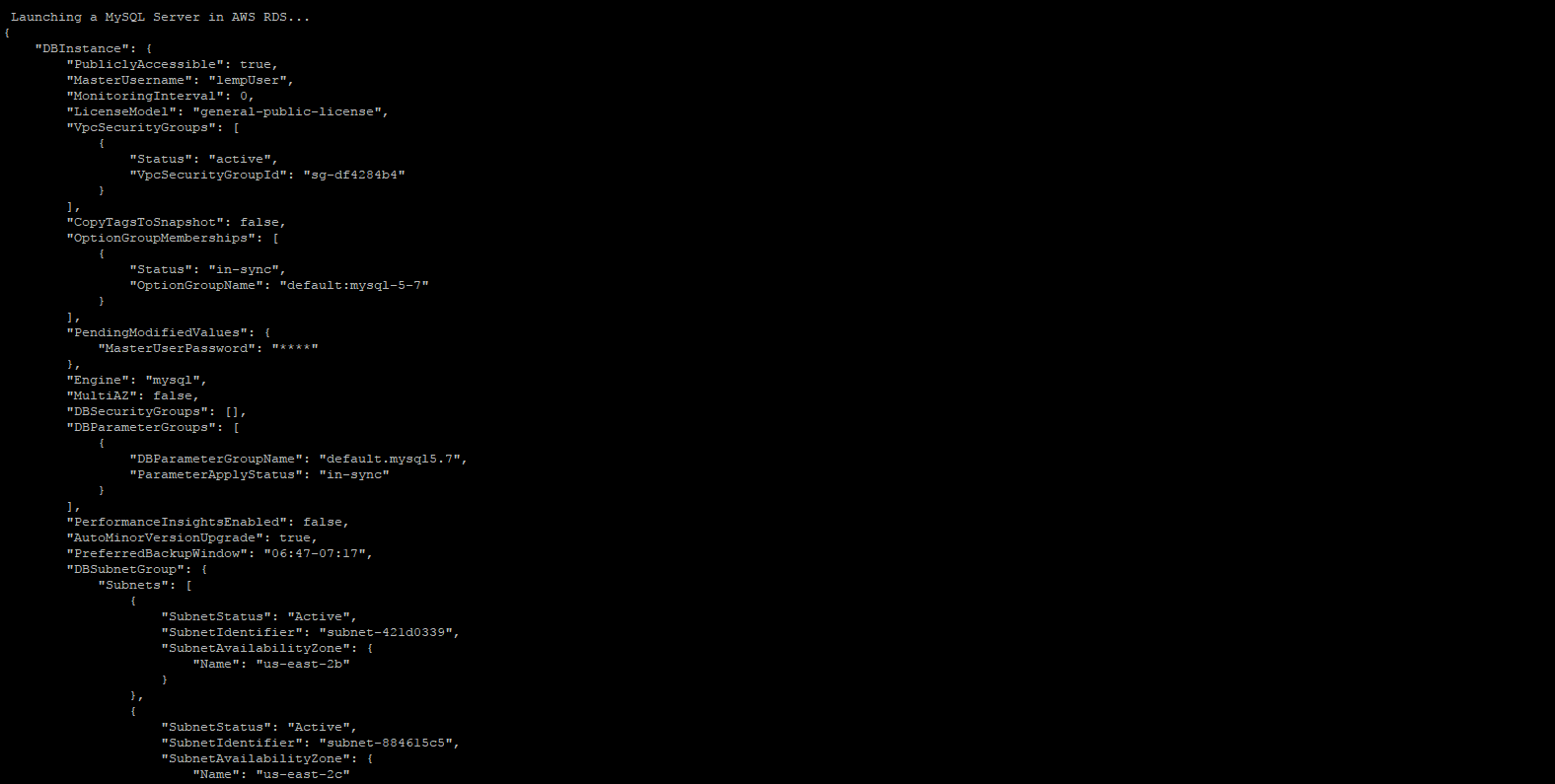
1. Creation of new security group.
2. Script Logs:

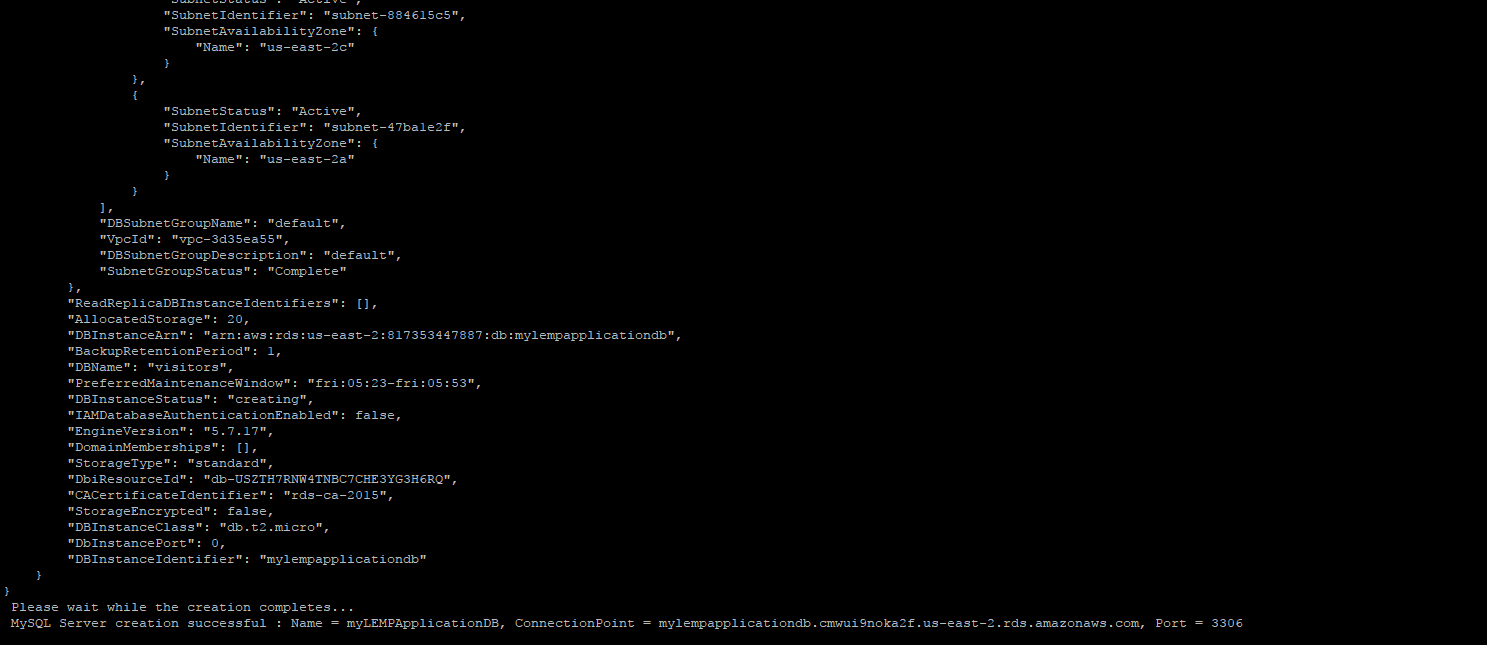


1. UI Screenshot:

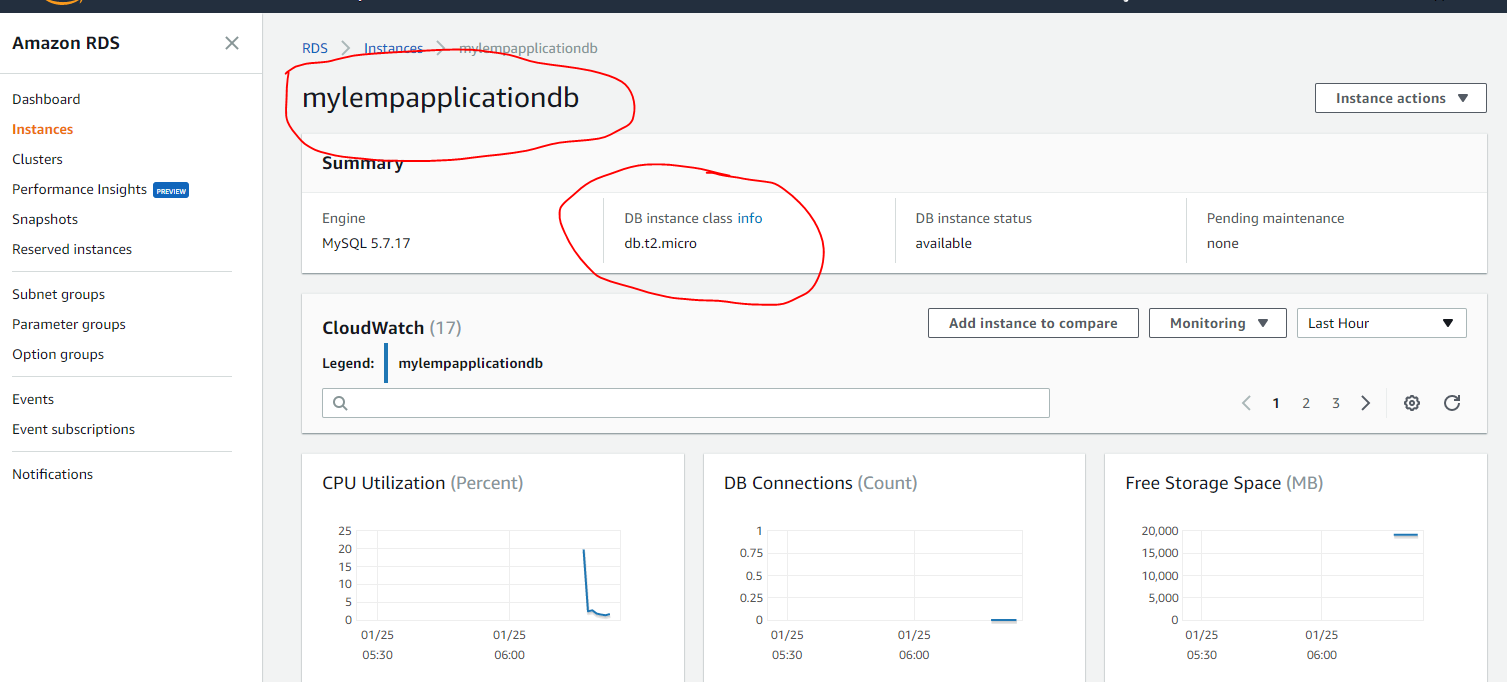


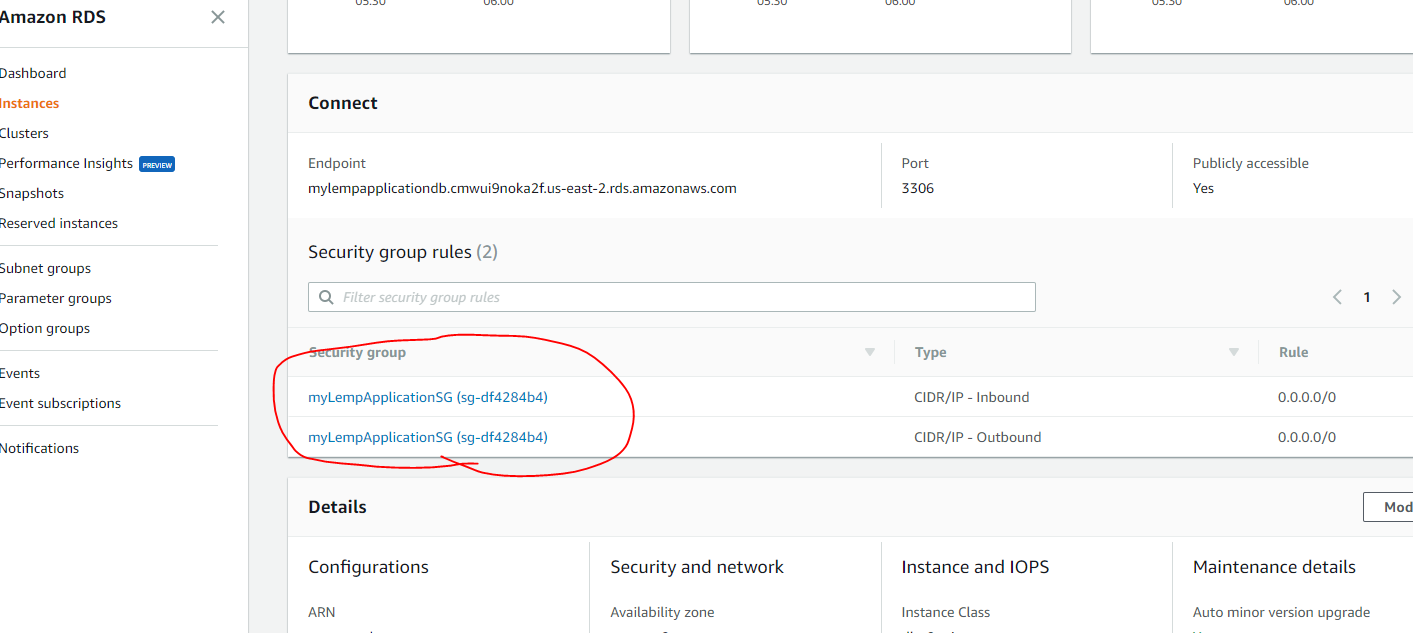
1. Creation of MySQL Server in AWS RDS
2. Script Logs:

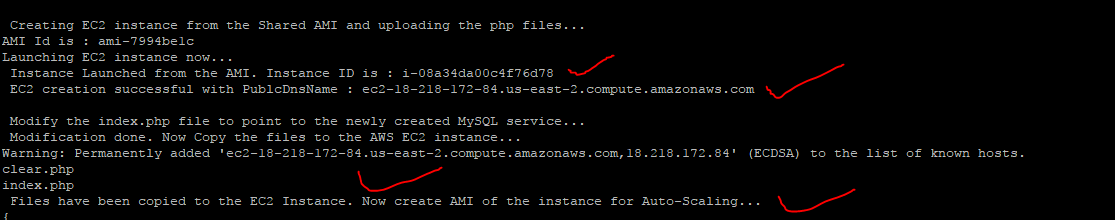


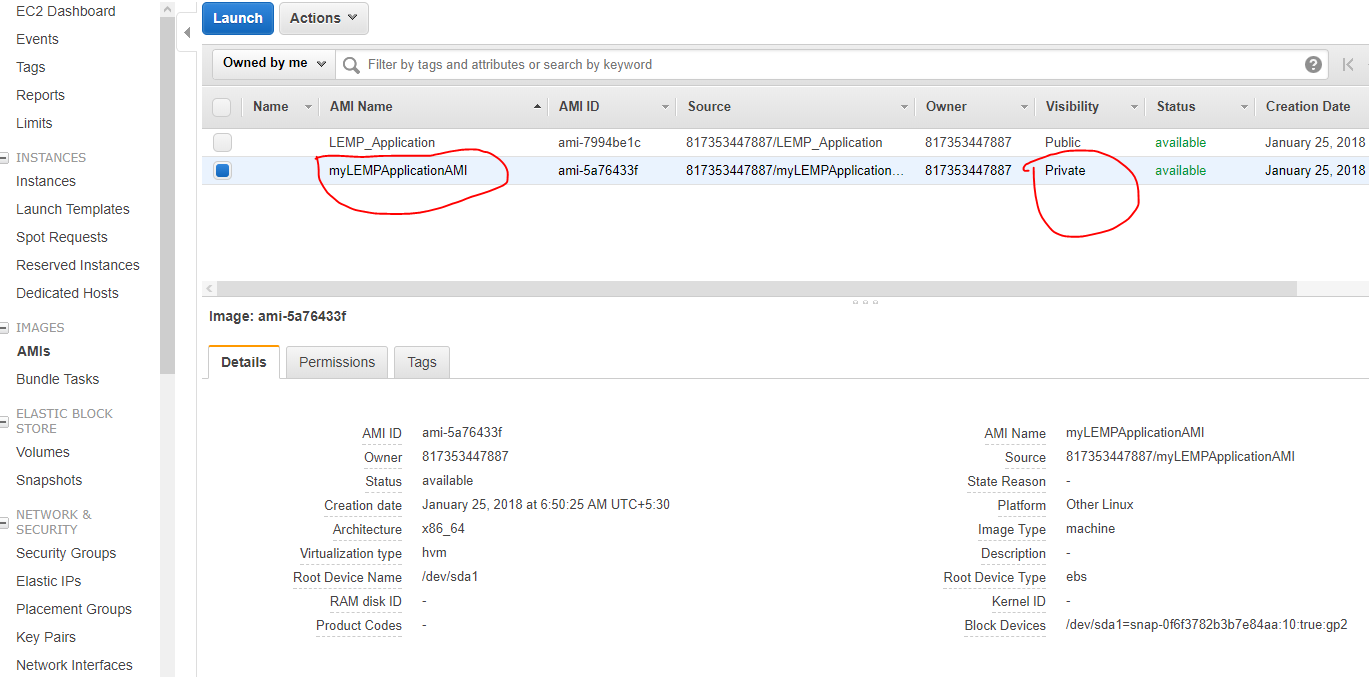


1. UI Screenshot:

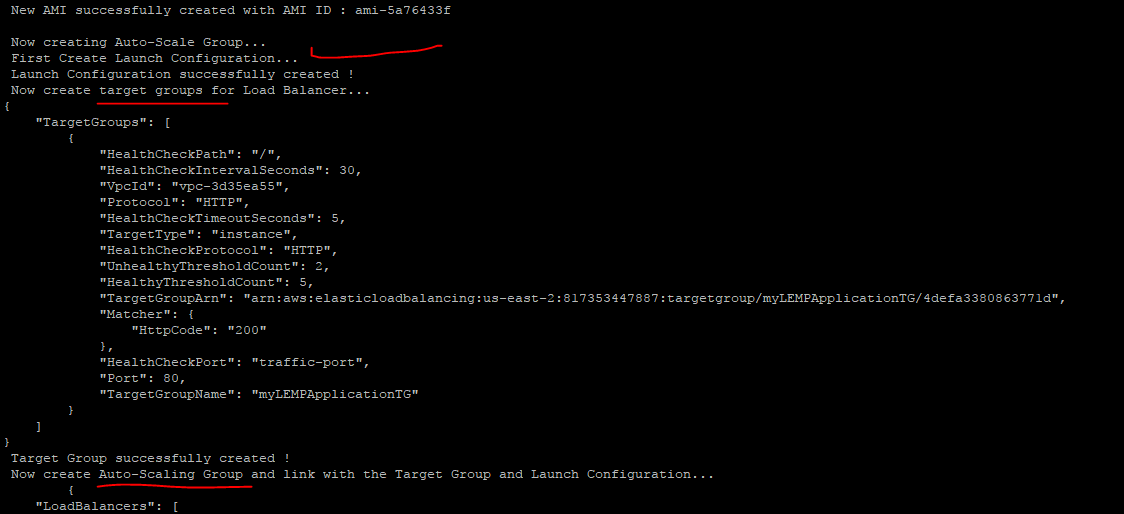




1. Creation of AWS EC2 Instance from the Public Shared AMI, where the modified php files (php files are modified to point to the newly created MySQL database) are placed and then a new AMI is created which is eventually used in the Auto-Scaling Group.
2. Script Logs:
3. UI Screenshot:

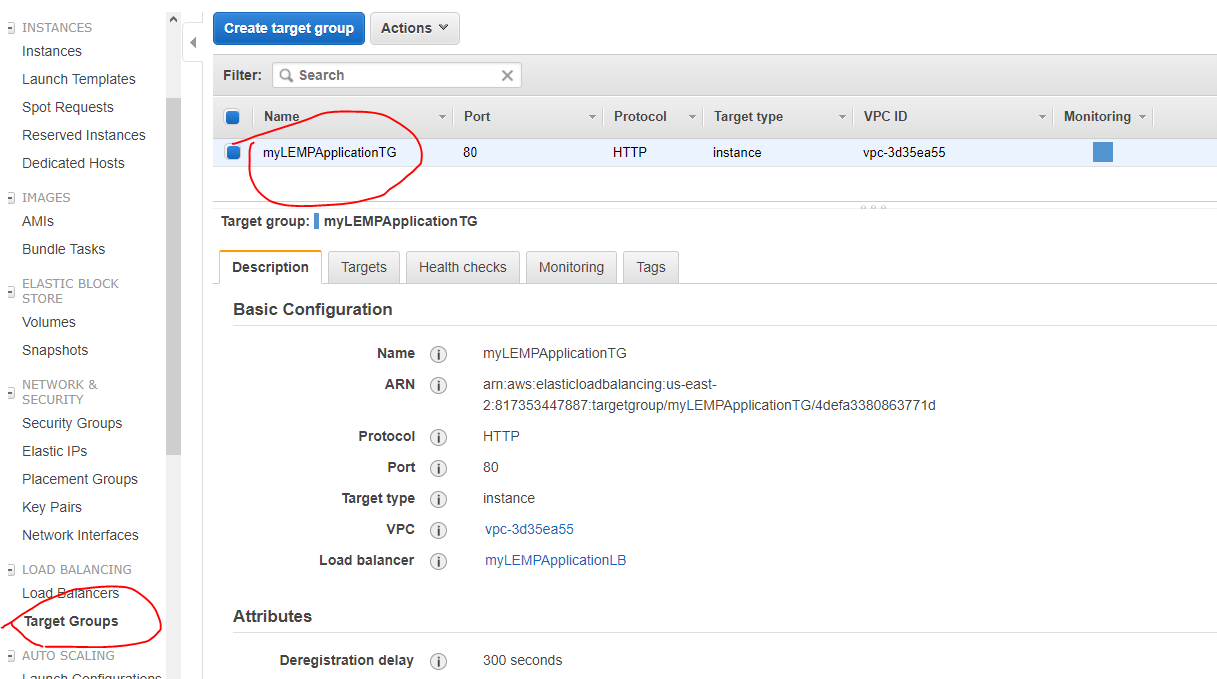


1. Creation of Launch Configurations, Auto-Scale Group, Target Groups.
2. Script Logs:

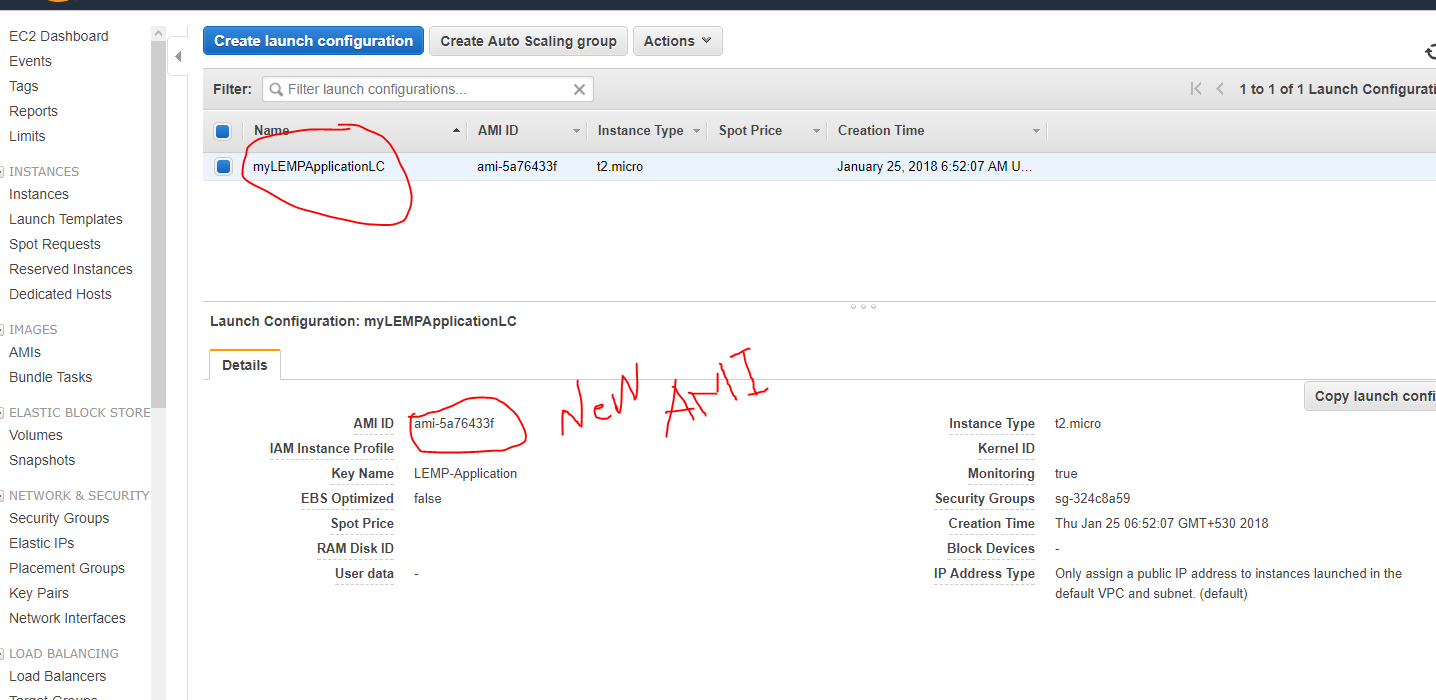


1. UI Screenshot:

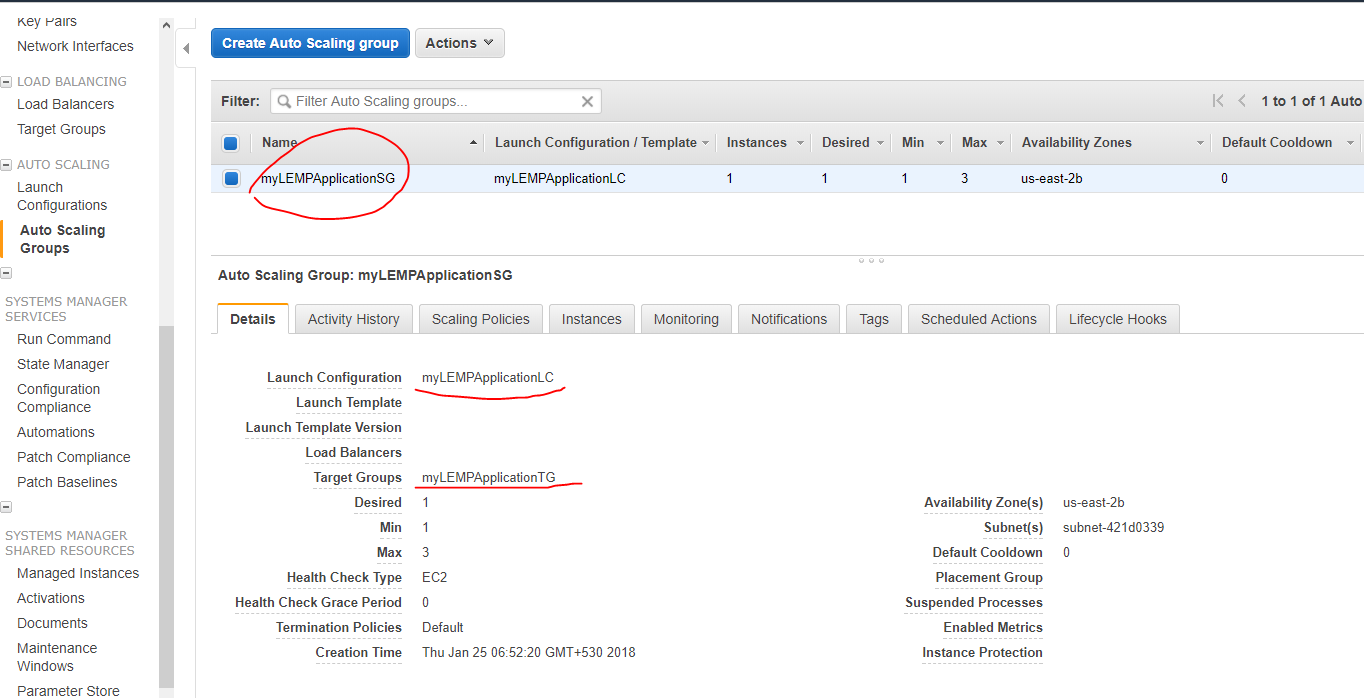
Target Group



Launch Configuration



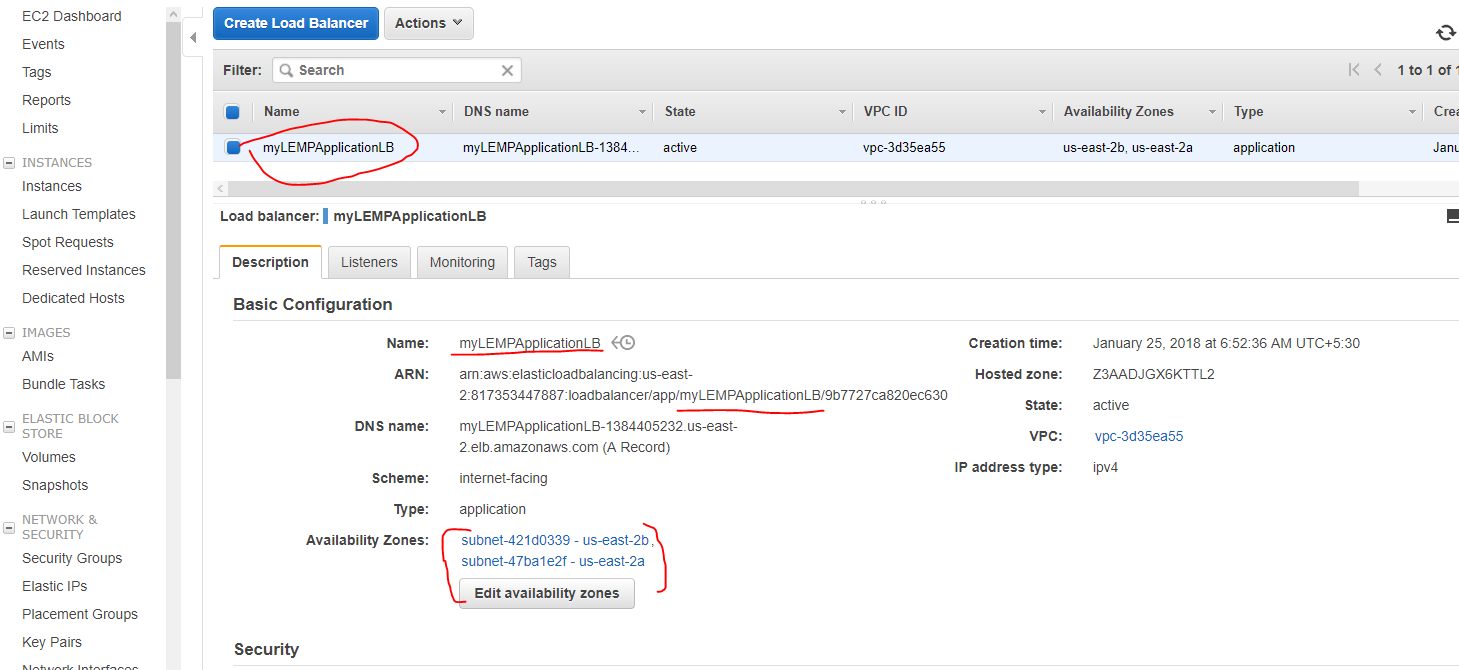
Auto-Scale Group

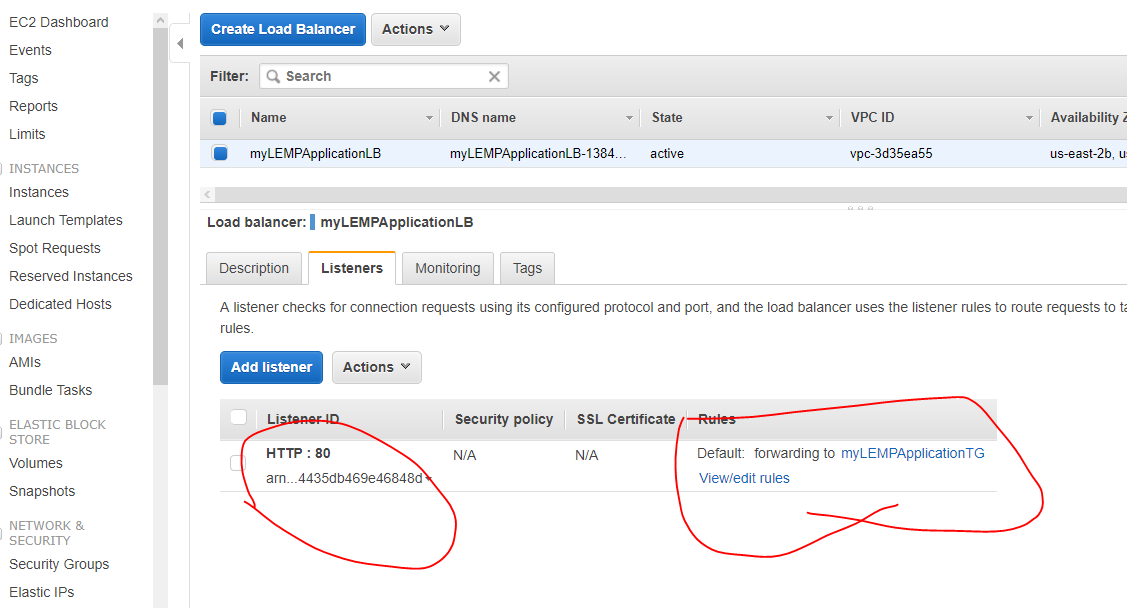


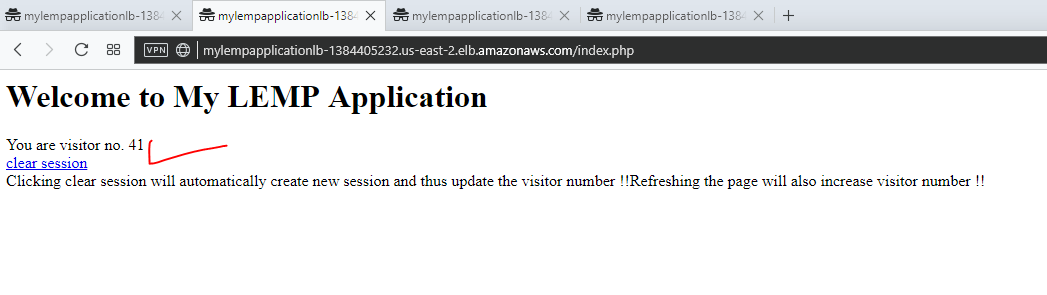
1. Load Balancer Creation
2. Script Logs:

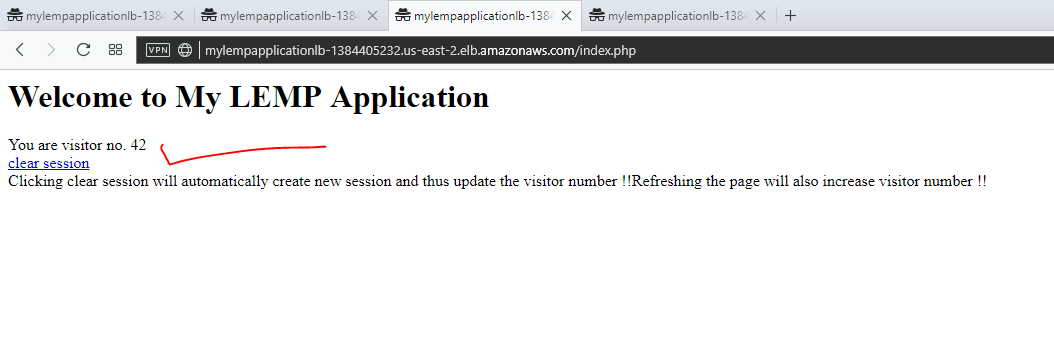


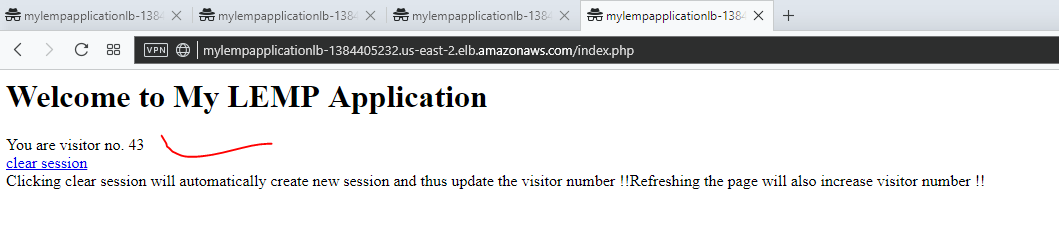
1. UI Screenshot:

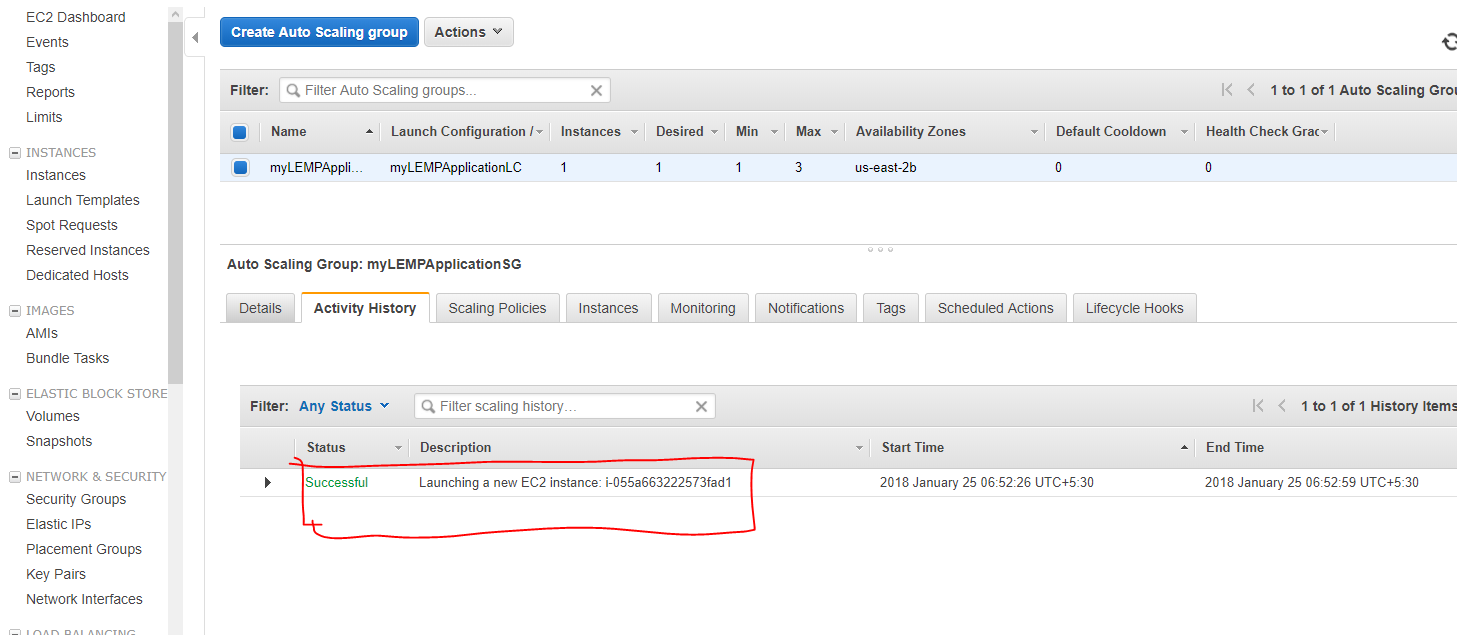


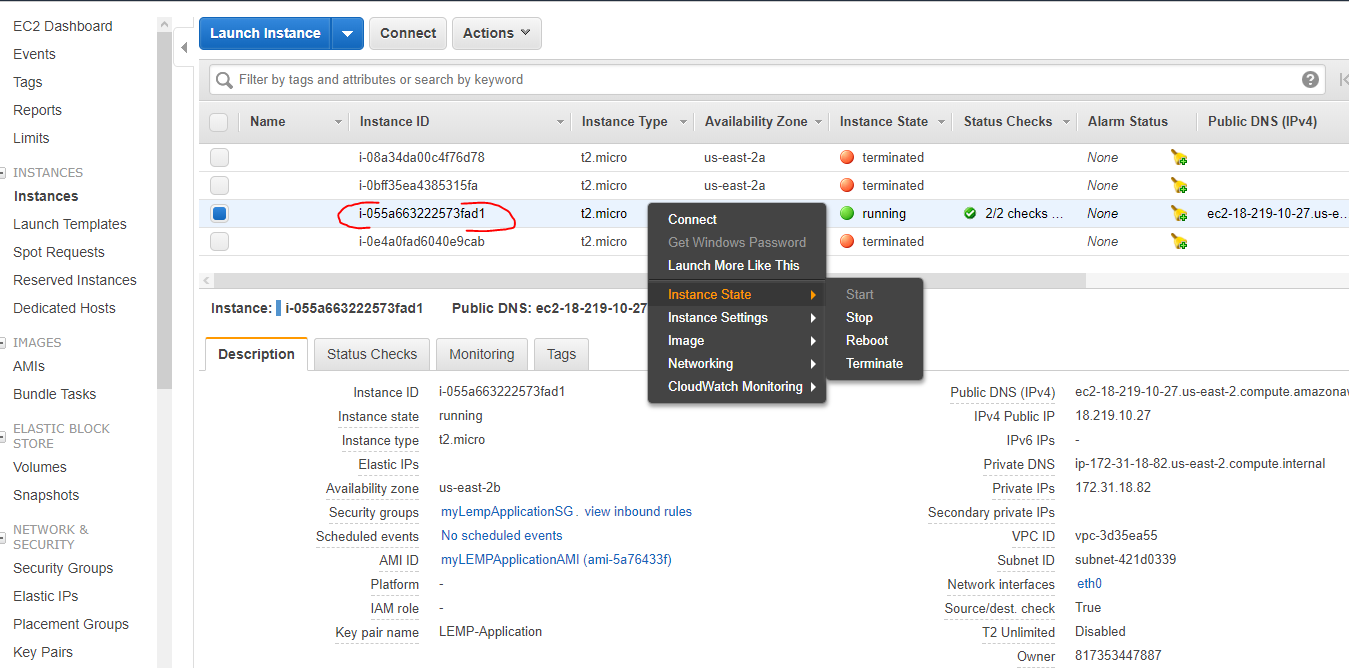


1. UI of the Application accessed through the Load Balancer  
   

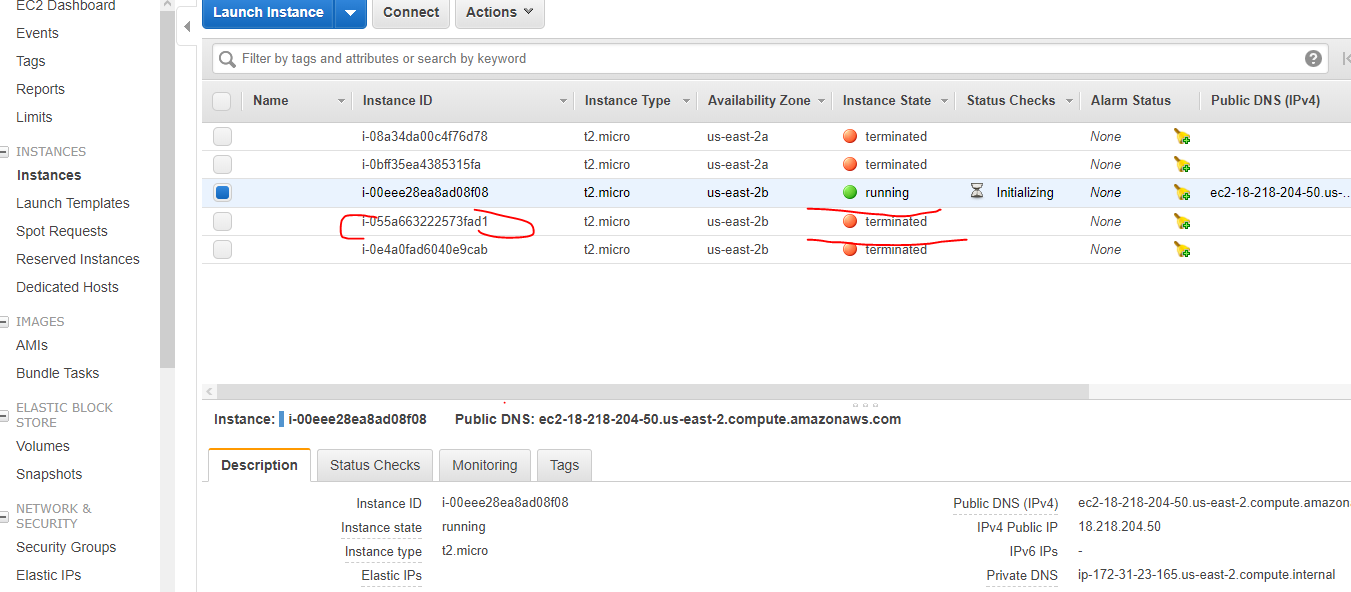


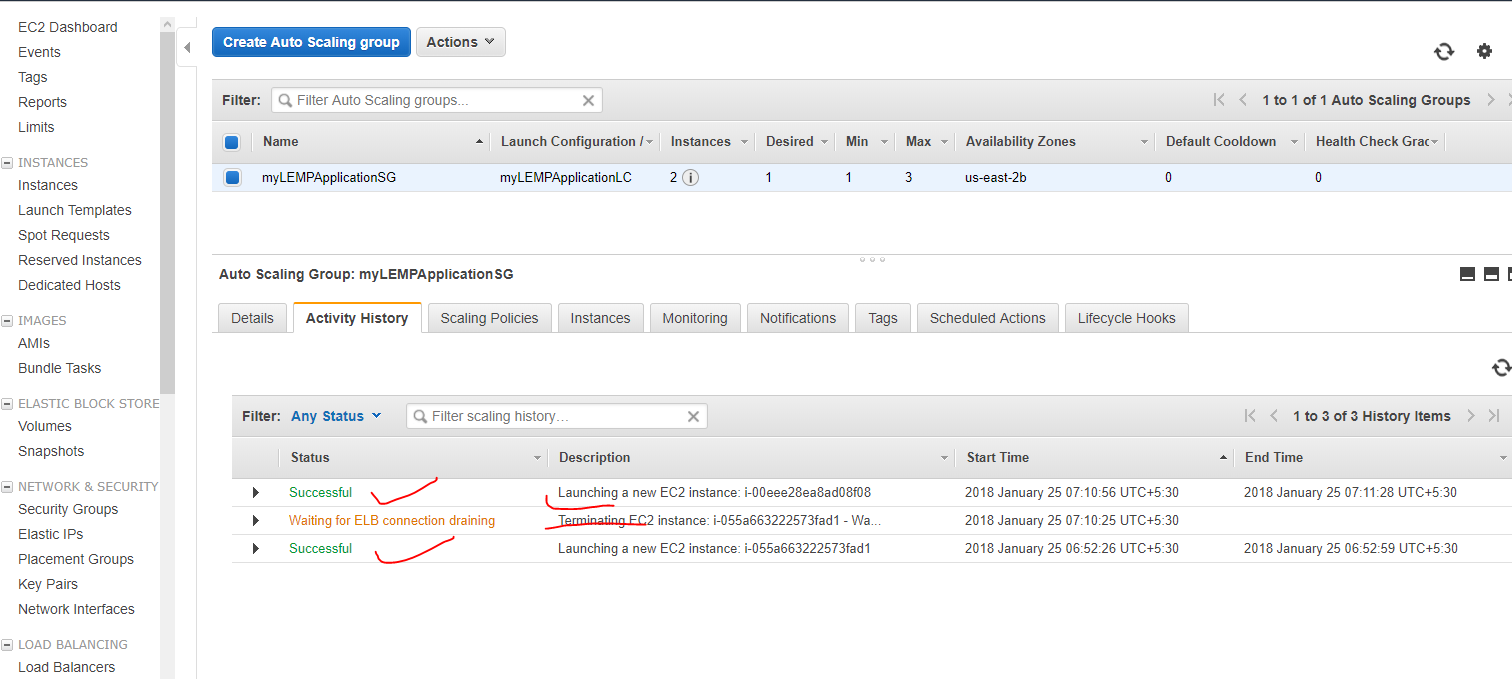


1. Auto-Scaling Feature  
   Couldn’t test using scripts since more than 2 machines creation was prohibited by AWS Free Tier.  
   Thus deleted the instance created by Auto-Scale Group and then it got automatically created by the Auto Scale group  
     
   



The 1 st instance deleted manually, the Auto-Scale group immediately launched another instance





END.