Static Load Benchmarking

148

Configure and Deploy an Elastic Load Blancer on AWS

In the previous part you have setup a load balancer using the web interface manually. In this part we will work on programmatically scaling the ELB using the ELB and EC2 APIs. ELB can be programmatically accessed through the ELB command line APIs, or through the AWS SDKs or boto for python. You can choose any API to complete the following tasks, but you will have to ensure that all of the required API calls and functionalities exist within the APIs that you plan to use:

You will configure a pool of servers that will grow to satisfy a particular condition. In this scenario, your program/script will automatically benchmark the ELB and continue to add servers to it until you achieve at least 3000 requests/second.

Note: For this checkpoint, assign the tag with Key: **`Project`** and Value: **`2.3`** for all resources

Provision an **`m1.medium`** instance to serve as your launchpad for load testing instance and run your script/program from it. Setup the required credentials in **`$HOME/.bashrc`**. If you are using the Cloudwatch CLI tools, additionally enter your credentials in **`$AWS_CLOUDWATCH_HOME/credential-file-path.template`** Your program/script should perform the following steps within the code:

1. Create an ELB and configures the appropriate port forwarding and health checks programmatically.
2. Launch an m1.small EC2 instance with AMI ID **`ami-69e3d500`** in the same availability zone as the ELB
3. Attach the launched instance to the ELB.
4. Run the apache_benchmark from the load testing instance to the ELB. The benchmark should be run using 100,000 requests with 100 running concurrently using the sample.jpg image in the same directory.
5. Retrieve and parse the result of the Apache Benchmark and retrieve the Requests/second metric. Save this value in a file for future reference.
6. If the number of requests per second is less than 2000, add another instance and benchmark it by repeating steps 2-6.

When you are ready, complete the following checkpoint quiz:

checkpoint

Static Load Benchmarking — Assess your progress.