**Load and Performance Testing using Apache JMeter**

Requirements to run JMeter for windows:

1. Install JDK any version greater than 8.
2. Download the zip file from the Source section using link: <https://jmeter.apache.org/download_jmeter.cgi>
3. Open the batch file for JMeter and it will start it:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Performing Load and Stress Testing:**

**Note: Testing** is done in the local environment as performing these tests in cloud can be expensive.

1. Open the JMX file which consists of already created tests.

A screenshot of a computer

Description automatically generated

1. In the thread group we can mention number of users performing the requests, ramp-up period (interval between the requests) and Loop Count (how many times these requests are performed).

A screenshot of a computer program

Description automatically generated

1. Keeping all these parameters to one and you can see the response results in View Results Tree

A screenshot of a computer

Description automatically generated

For Performance metrics select View Results in Table

A screenshot of a computer

Description automatically generated

1. Keeping users to 10, ramp-up period =2 secs and loop count to 2

A screenshot of a computer

Description automatically generated

After request from 7th user the pipeline couldn’t take any load and the nlp service in docker crashed.

A screenshot of a computer

Description automatically generated

**Conclusion:**

All these tests were conducted in the local machine (currently best available windows laptop) using docker and the whole project can tolerate 6 users using the whole project in the interval of 2 seconds. This performance has direct relation with the configuration of the system and depending upon that this number can be higher or lower.

**Possible Solution:** We can introduce a load balancer to balance the load between multiple EC2 instances if our userbase increases. To solve scalability issue, we can introduce ASG (Auto scaling Groups) which can adjust the number of EC2 instances behind the load balancer as per user traffic. This will make our system resilient, fault tolerant and scalable.