**Manual**

**How to run the code:**

**Pre-requisite:**

Run *bash install.sh* to your machine.

1. **Local Queue Steps:**
2. Go to Script folder and run “gen\_worload.sh”. which can generate folder for different folder to make workload files.

*$ bash gen\_worload.sh && bash gen\_sleep0.sh*

2. Move to the “Local” directory by issuing the following command.

*$ cd Local*

3. Now execute the “Client.py” file using the following command line arguments.

*$ python Client.py -t 8 -w <workload\_file> -s LOCAL -t 4*

-s LOCAL or —schedule = LOCAL Specifies the SQS queue name.

-t 4 or —threads=4 Specifies the number of threads.

-w or —workload= Specifies the workload file name.

**2. Remote Queue:**

**On Client Node**

1. Go to Script folder and run “gen\_worload.sh”. which can generate folder for different folder to make workload files.

*$ bash gen\_worload.sh && bash gen\_sleep0.sh*

2. Run “Launch\_Instances.py” to instantiate the Instances.

*$Python Launch\_Instances.py*

3. Move to the “Remote” directory by issuing the following command.

*$ cd Remote*

3. Now execute the “client.py” file using the following command line arguments.

*$ python client.py -s QNAME -w <workload\_file> -t 4*

-s QNAME or —queue=QNAME Specifies the SQS queue name.

-w or —workload= Specifies the workload file name.

-t or –thread = Specify the number of remote instances to Launch.

**On Worker Node**:

1. Move to the “remote” directory by issuing the following command.

*$ cd remote*

2. Now execute the “Worker.py” file using the following command line arguments.

*$ python Worker.py -s QNAME -w <workload\_file> -t 4*

-s QNAME or —queue=QNAME Specifies the SQS queue name.

**For Animoto:**

**Client Side:**

*$ python client.py -s QNAME -w <workload\_file> -t 1*

**Remoteworker:**

*Cd animoto/*

*python Worker.py -s QNAME -w <workload\_file> -t 1 -a animoto*