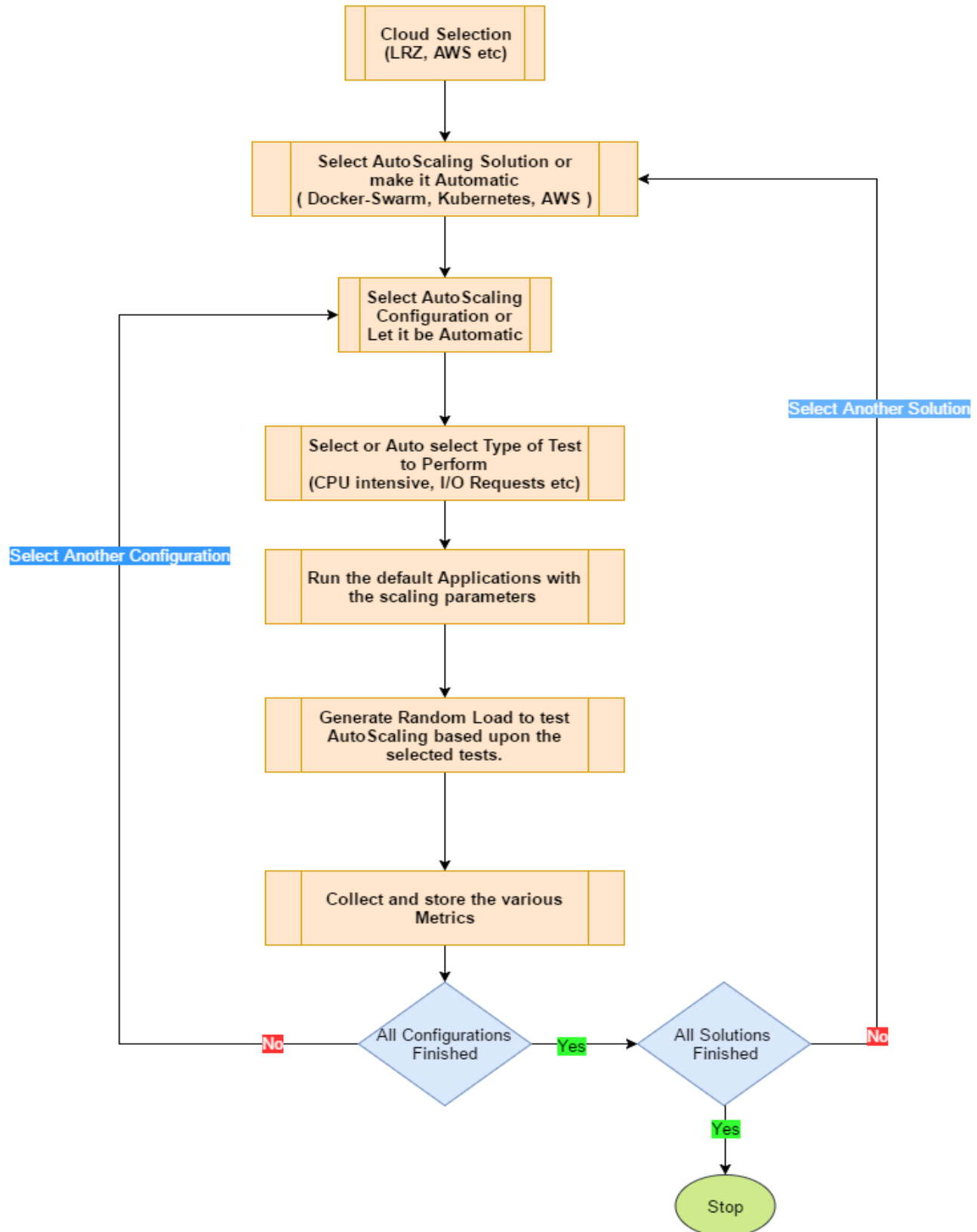


MULTI-LAYERED-CLOUD-APPLICATIONS-AUTO-SCALING-PERFORMANCE-ANALYSIS

Flow Diagram



VARIOUS PHASES OR PARTS OF THE SYSTEM ARE AS FOLLOWS:

1. **Cloud Provider Selection:** This part let the user choose the cloud provider on which he/she want to test the Autoscaling solution. For the starting point, we are going with the LRZ and then once the system is stable other public cloud providers like AWS, Azure will be added to it.
2. **Auto-Scaling Solution Selection:** This section describe the which Autoscaling solution is to be used to perform the test. Here the user has the following options to choose from (For detail about individual, refer to AutoScale.pdf):
 - a. **Docker-Swarm:** Docker Engine 1.12 includes swarm mode for natively managing a cluster of Docker Engines called a swarm. For each service, you can declare the number of tasks you want to run. When you scale up or down, the swarm manager automatically adapts by adding or removing tasks to maintain the desired state.
 - b. **Kubernetes:** Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.
 - c. **AWS Autoscale:** AWS Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define.
 - d. **Automatic Selection:** Here the platform selects all the solutions one by one and perform the test.
3. **Auto-Scaling Configuration Selection:** Here the user can choose different configurations to test the auto-scaling solutions or it can be selected as automatic where system selects all the configuration one by one and perform the test. Various configuration parameters are:
 - a. Type of Instance
 - b. Minimum and Maximum No of Instances to scale in and out.
 - c. Number of Containers/Pods in case of Kubernetes
 - d. Scaling Decision Metric (More detail can be found in Autoscaling.pdf):
 - i. CPU Utilization
 - ii. Disk Reads
 - iii. Disk Read Operations
 - iv. Disk Writes
 - v. Disk Write Operations
 - vi. Network In
 - vii. Network Out
 - e. Auto-Scaling Policy
 - i. Simple Scaling
 - ii. Step Scaling
4. **Applications to test the Configuration:** This part contains the various open-source applications selected to test the auto-scaling configurations. The applications will be divided into following categories:
 - a. CPU-Intensive Application
 - b. User Request Application
 - c. High Memory Usage Application
5. **Load Test for the Applications:** This section will contain the default random-auto-load-generator based upon the type of application. The load will vary from time to time to test the auto-scaling solution.

6. **Results Collection:** This part will collect and store the results into a csv file per configuration per solution.

Following metrics will be saved:

- a. CPU usage per VM instance and per docker container (for a regular interval of time).
- b. Memory usage per instance and per docker container.
- c. Start and Stop time duration for the VM instance and docker container.
- d. Requests served in the set interval (IOPS).
- e. Network in and Out from each VM Instance.
- f. Success, Failure, and Restart rate of each VM and Docker containers.
- g. CPU and memory consumption when the VM Instance or container is idle.
- h. Auto-Scaling graph (when and how many does the number of VM instances or docker containers are increased or decreased)