4.1.x Cloud Partitioning

The following subsections explain the concepts of cloud partitioning:

- **Partitions**
- **Network Partitions**
- **Network Partition Algorithms**
- **Network Partition Definition**

**Partitions**

A partition depicts the division in an IaaS and defines an area of an IaaS cloud used by a service subscription. A partition can be made at one of the following levels: Provider level, Region level, Zone level or Rack level. A partition should at least have a provider defined. If required, partition groups can be defined in the partition definition. DevOps, for the purpose of high availability, can define multiple partitions in order to instruct Apache Stratos to spawn instances in multiple areas (e.g., region, zone or rack). For example, if you spawn instances in multiple availability zones in EC2, your system will be still functional even if one availability zone is down.

**Network Partitions**

A network partition, which is also referred to as a partition group, is an area of an IaaS that is bound by one network of an IaaS (e.g., region). Each region will demarcate a separate geographical area. Each region can have multiple, isolated locations that are referred to as zones. Thereby, several partitions can be included inside a network partition. If needed, DevOps can define multiple partition groups that have multiple partitions for the purpose of high availability. The partition algorithm that is applied between partitions within a network partition is defined in the deployment policy. While, the partition algorithm that is applied between network partitions are defined in the application policy.

**Network Partition Algorithms**

When scaling takes place among several partitions, the partition algorithm that needs to be used has to be defined. Partition algorithms define the scaling mechanism that needs to be used to spawn instances between network partitions. The partition algorithm that is applied within a network partition, may vary from the partition algorithm that is applied between different network partition. For example, if there are two network partitions named
NP1 and NP2, and if there are two separate partitions named P1 and P2 within PG1, then the partition algorithm used between P1 and P2 within PG1 may vary from the partition algorithm used between NP1 and NP2.

The available partition algorithms are as follows:

<table>
<thead>
<tr>
<th>Partition Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One after another</td>
<td>When using this algorithm, instances will be spawned in the same partition until the maximum number of instances for the respective partition is reached, before spawning instances in the next partition. For example, if there are two partitions named P1 and P2, instances are spawned in the same partition until the maximum number of instances are reached, before spawning instances in P2. This partition algorithm can be applied between partitions and network partitions.</td>
</tr>
<tr>
<td>Round robin</td>
<td>When using this algorithm, instances will be spawned in each of the partitions sequentially. For example, if there are two partitions named P1 and P2, after an instance is spawned in P1 it will spawn the next instance in P2. This partition algorithm can only be applied between partitions.</td>
</tr>
<tr>
<td>All at once</td>
<td>The instance of the application is added to all the network partitions used by the application. This partition algorithm can be applied between partitions and network partitions.</td>
</tr>
</tbody>
</table>

Example Scenarios

In a typical cloud bursting scenario, you would use **one after the other**. For example, if an enterprise has its private cloud, all instances would be allocated there, and when you run out of capacity, you would go to the next partition on the public cloud. The **round robin** partition algorithm can be used for very high availability, where instances are spawned across multiple IaaS providers.
Auto-scaler makes decisions based on the defined partitions that are defined in the network partitions. Network partitions are defined in Apache Stratos to manage the instance count. The network partitions are defined in the network partition JSON file. The deployment policy definition can be defined at the group level or at the cartridge level, and will define the maximum number of instances (max) that will be allowed in a group or cartridge.