A simple fat tree network showcase

Course

Fog and Cloud Computing 2018/2019

Students

- Brugnera, Lorenzo <lorenzo.brugnera@studenti.unitn.it>, S197054
- Zaupa, Eros <eros.zaupa@studenti.unitn.it>, S208272

Proposal

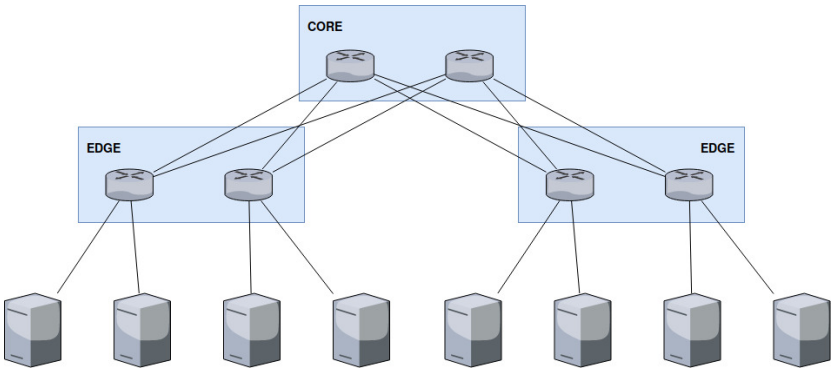
Our project goal is to reproduce a model of a fat tree network using the OpenStack service. Fat tree networks are usually deployed in environments with *high* computational power and *heavy* bandwidth consumption, such us *data centers* and *cluster supercomputers*. The resources made available by the OpenStack service can't compare to the requirements of real use cases, but a *simpler* and *lighter* implementation of this topology is still feasible. The model will cover the main property of the topology: top branches are "fatter" (thicker) than lower branches. This means that for each *edge* switch, the number of links that go to its siblings is equal to the number of links that go to its parents.

Configuration

We will use *Ansible* as an automation tool for two main reasons

1. The available OpenStack service comes with *no guarantees*, so preventing any data loss is up to us.
2. We want to define the project configuration in a *structured* way.

Network



- $L = 2$ level fat tree.
- $K = 4$ ports per switch.
- $(2L - 1)(K/2)^{L-1} = 3(K/2) = 6$ switches.
  - $(K/2)^{L-1} = K/2 = 2$ *core* switches.
  - $2(K/2)^{L-1} = K = 2$ *edge* switches.
- $N = 2 * (P/2)^{L} = 8$ hosts.

Requirements

Instances To keep the resources consumption at minimum, our goal is to use a *m1.tiny* flavor for each instance running an OS with low requirements (e.g. CirrOS) using 8 of the 10 instances slots available. This should be enough to execute simple connection tests (e.g. ping, ssh).

Routers We will use 6 of the 10 routers slots available.

References

- `https://clusterdesign.org/fat-trees/`
- `https://www.cs.cornell.edu/courses/cs5413/2014fa/lectures/08-fattree.pdf`
- `https://packetpushers.net/demystifying-dcn-topologies-clos-fat-trees-part2/`
- `https://www.ansible.com/`