Using OpenStack in a demanding production environment

Bruno Almeida
F-Secure
OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds.”

– opensource.com

- Project started in 2010

- Rackspace and NASA are the key initial contributors.
Components

- Nova (Compute Infrastructure)
- Cinder (Block Storage)
- Swift (Object Storage)
- Neutron (Networking Service)
- Glance (Imaging Service)
- Keystone (Identity Service)
Cloud

Expectation

Reality
Paradigm Shift

- Single server with multiple services
- Multiple servers with multiple services
- Multiple servers (Virtualized) with 1 service per virtual server
Paradigm Shift

- Multiple servers (Virtualized and in a Cloud Infrastructure) with 1 service per virtual server

Differences:

- No more 99.999% uptime
- 1 service = 1 server
- I don’t know (and I don’t care) in which physical server my service is running
Paradigm Shift

- Multiple servers doing the same functions according to workload (Stateless Services)

- Service has more control over the server itself, i.e., scale vertically or horizontally according with needs (Auto Scaling)

- Virtual server is becoming more “independent” from the physical server (e.g., Docker)

- Server is a disposable element that can and should be re-deployed if necessary
Example Cloud Infrastructure

- Controller: 10.11.0.1
- Storage: 10.11.2.X
- Virtual Instances: 10.11.3.X
- Internet: 10.11.1.X
- node0X: 10.11.1.X
Constrain #1:
Isolation within different security domains

**Mitigation techniques:**

- Different tenants/projects for different use cases
- Virtual Switching
- Egress and Ingress firewall
- Different Availability Zones
Constrain #2: Malware Handling

- Separate Cloud Infrastructure
- Different Identity, Compute and Storage Services

**Mitigate Possible Threats:**
- Hypervisor exploits
- Malware behaviour due to virtualization recognition
- Network infection
Constrain #3: Performance issues in fast testing scenarios

Spawn – Run – Terminate

- Residual data is left behind
- Slow compared with the traditional single server virtualization approach

Possible solution:
- Combine both approaches (Cloud + Traditional Virtualization) by using nested virtualization
Questions?

Bruno Almeida
F-Secure
bruno.almeida@f-secure.com