MODULE 1

Cloud Computing Overview

OneCloud
Consulting

Module 1 Objectives

By the end of this module, you should be able to explain:

- What “Cloud” means – precisely.
- Purpose and History of Cloud
- How do we, as developers, interact with the following resource pools:
  - Compute
  - Storage
  - Network
- What is the ‘programmatic cloud’ we call OpenStack?

Clouds Defined

Cirrocumulus

Altocumulus

Cumulus

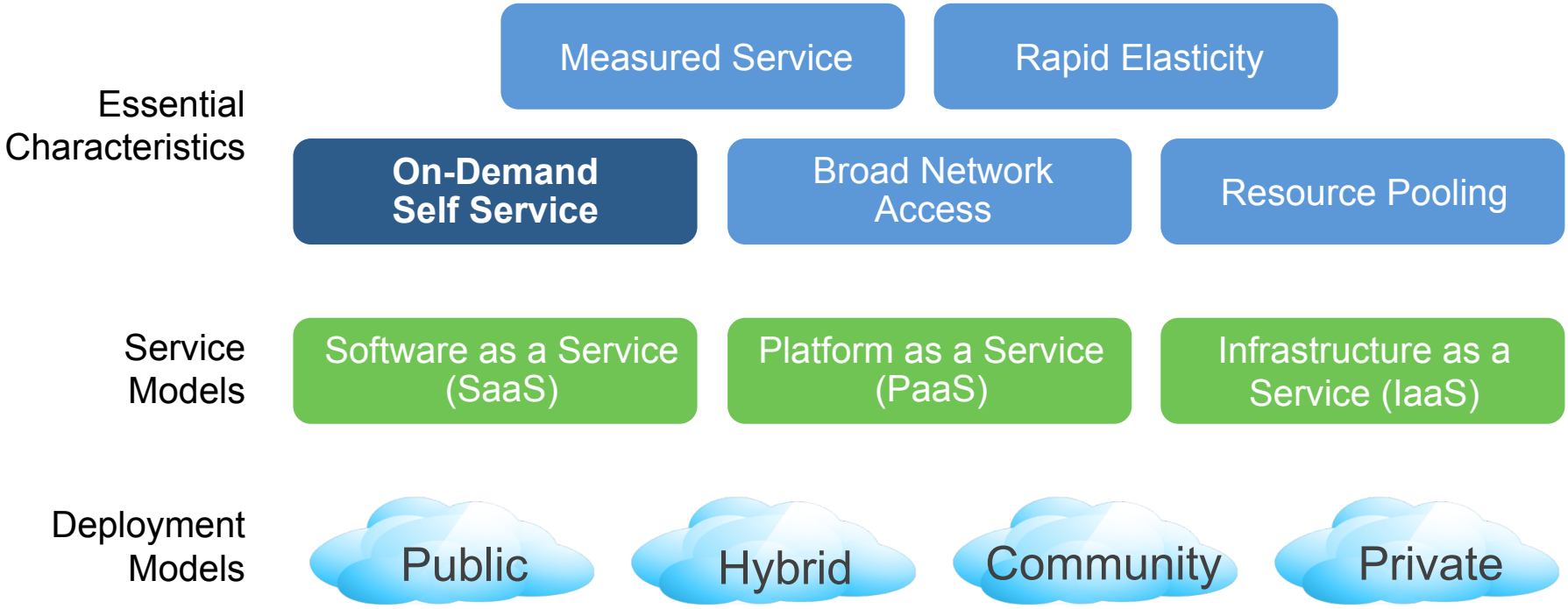
Lenticular

Cirrus

OneCloud Consulting


How would you
define Cloud?

Cloud Defined



US NIST Cloud Model

OneCloud Consulting

Application History

Monolithic (1940-1980)

- Mega scale “single” compute needed to deal with app scale
- Development as waterfall process, major projects for development, upgrades, etc.
- Security and scale through dedicated external appliances (firewall, SAN storage)

Distributed (1980-2010)

- Application broken into scalable units
- Development still waterfall, with complex management and staging required for component upgrades
- Security and scale rely more heavily on network, QoS, embedded services
- Internet bubble web sites, load balancing via appliances to monolithic web, app, db servers

WebScale (2010-Present)

- Applications broken into smaller easy to replace units with managed APIs for interaction
- “Agile” development with Continuous Integration and Deployment
- Distributed security, either in the app, or at the network edge
- Current super-scale web services and rapidly developed super-custom apps


Why do
you care?

Benefits of Cloud

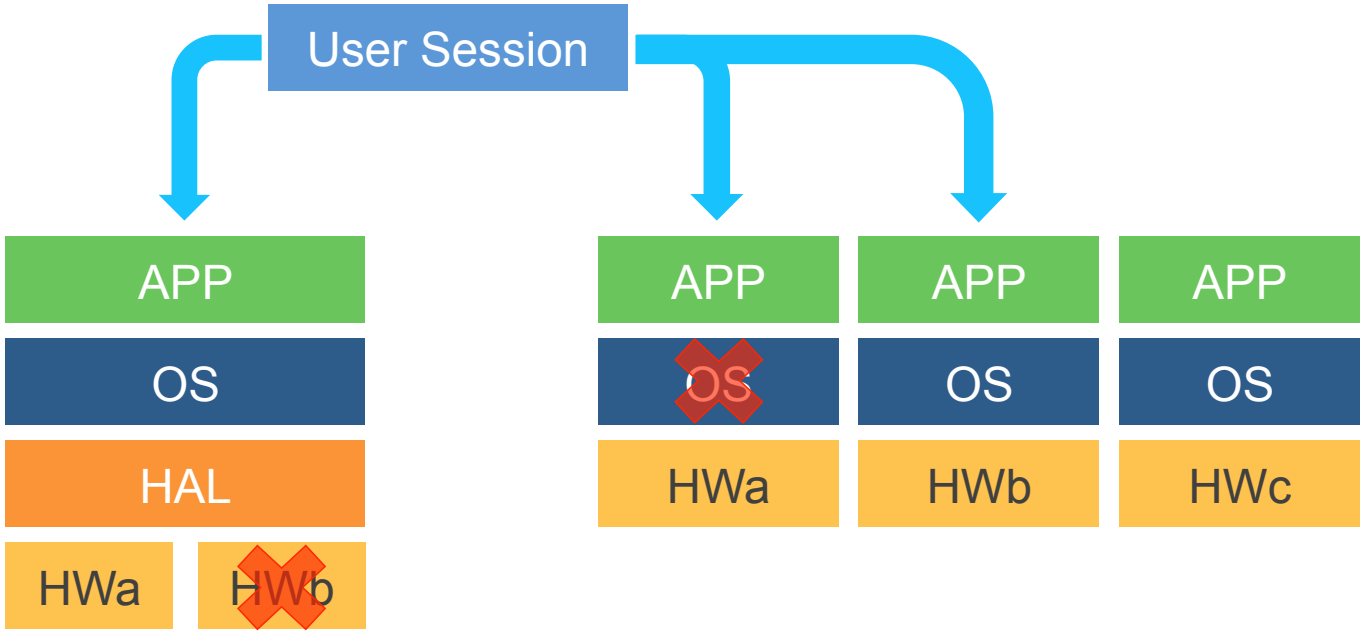
- **Reduced capital and operations costs**
  - No large up-front capital investment on datacenters
  - Eliminate the need to plan ahead for provisioning
  - pay-as-you-go pricing
- **Application development**
  - reduces overall development time
  - self-provision development and testing environments (aka devops)
  - Centralized workspace enhances ease of collaboration
- **Application deployment & management is simplified**
  - Common programming model across mobile, browser, client, server, cloud
  - Access to strong ecosystem of widely deployed applications
  - Integration with existing IT assets (Software + Services)

OneCloud Consulting

Resiliency models

Infrastructure Resiliency

Application Resiliency



OneCloud Consulting

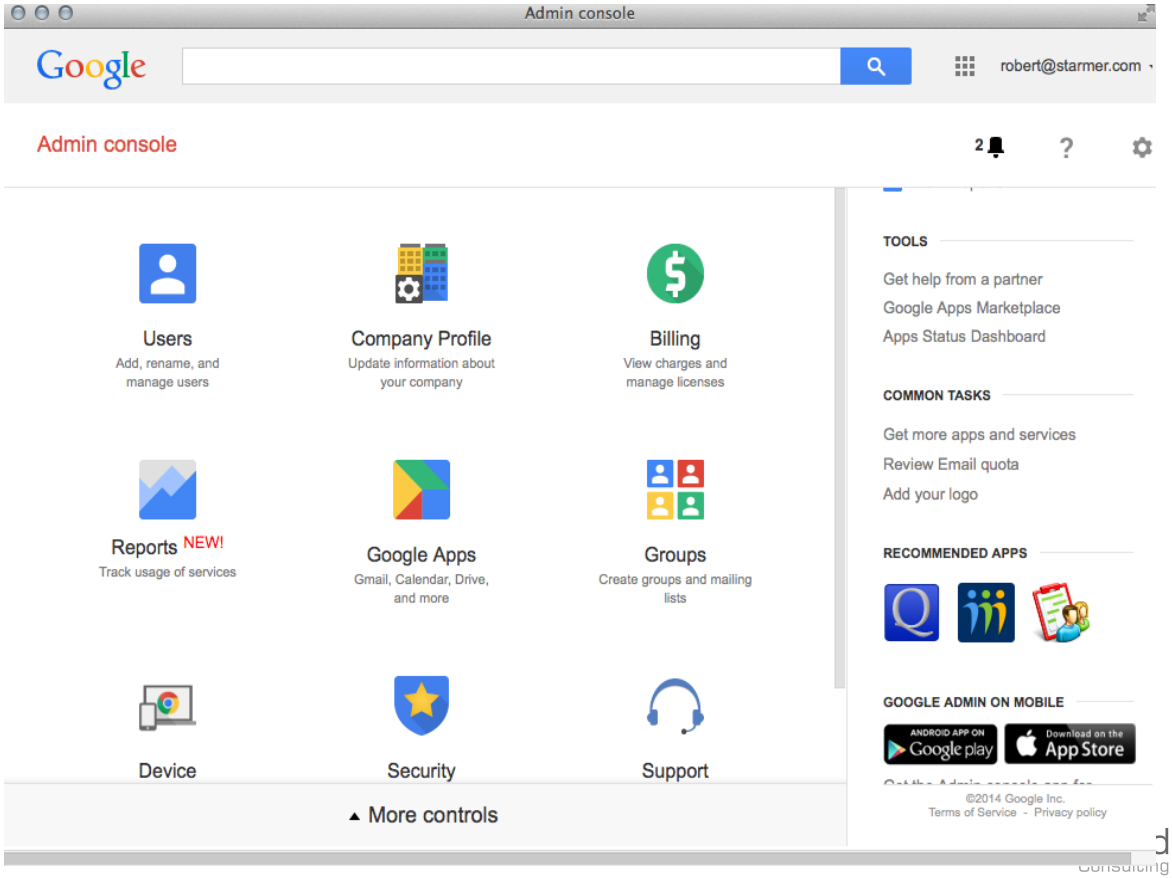
SERVICE MODELS

Saas, PaaS, IaaS

OneCloud
Consulting

Software as a Service - SaaS

- Company management as a service
- Can add additional software capabilities
- Central Dashboard/ Catalog

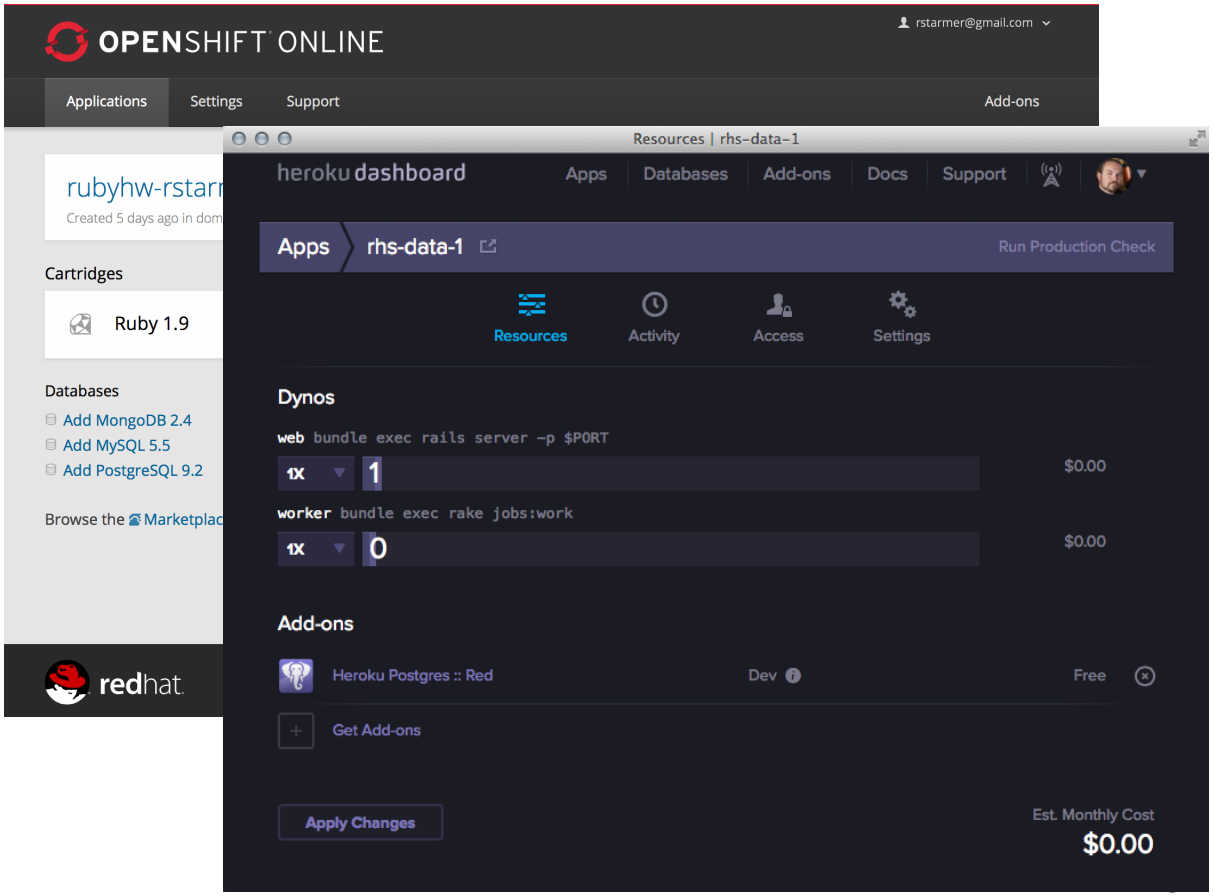


SaaS – For Dev

- Fully featured applications, but more and more are really services
  - Web-mail services (Google, Yahoo, Outlook)
  - Web based CRM services (SalesForce, Oracle, etc.)
- SaaS Apps can accelerate “solution” development
  - Integrate Google Docs into your apps
  - Leverage Salesforce data as an input into an analytics engine
- One downside is that there are no standards for interaction, beyond the limited scope of older SOAP/WS based models, or newer REST interactions, where an API document is needed.

OpenShift/Heroku– PaaS

- Applications as “work elements” instead of infrastructure components
- Measured and consumed on demand



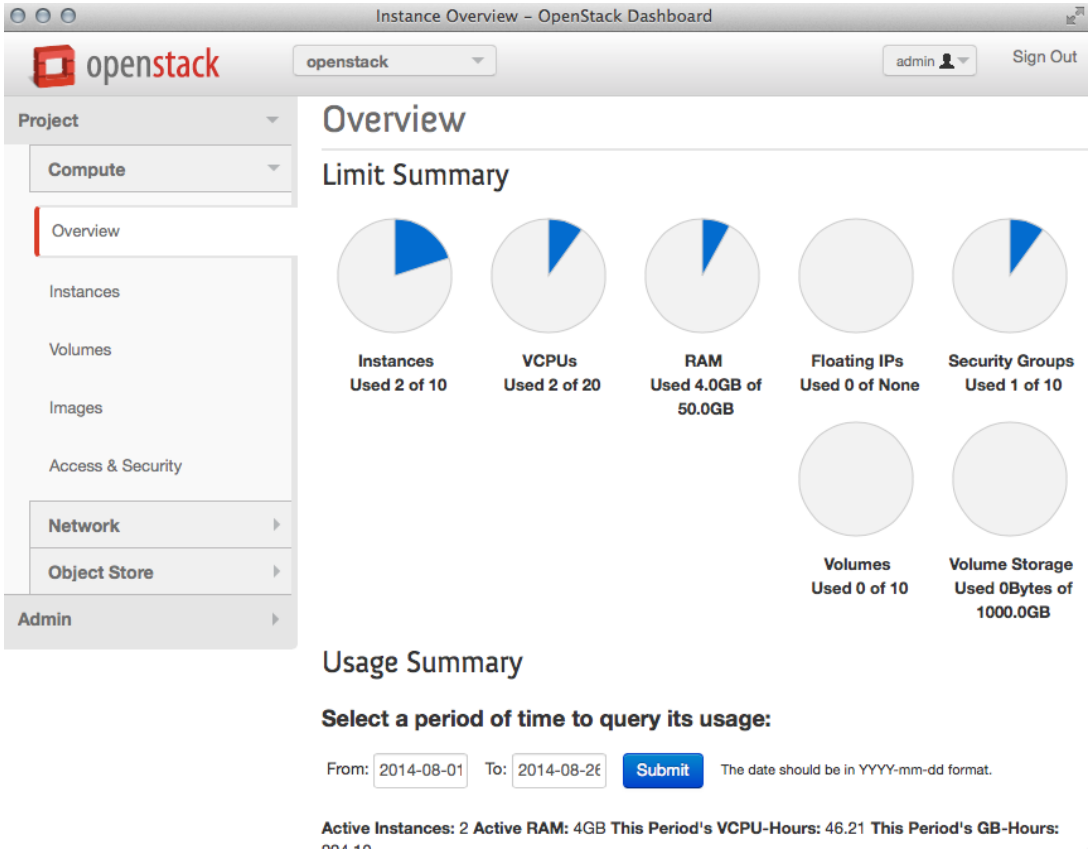
PaaS for Developers

- Platform intended to provide core “services” behind deployments, often leveraging MVC systems models as application services tools.
- Often provide model data storage (databases), Runtime engine (against specific app models), and the ability to scale (via embedded SLB frontends)
- Example: Heroku (https://devcenter.heroku.com/articles/architecting-apps)

Development and Configuration	Strict separation of code and configuration, explicit dependency declaration, tight development iterations and parity between environments.
Runtime	Applications are run as independent, lightweight, and stateless processes with quick startup and shutdown.
Management and Visibility	Execute auxiliary tasks in one-off processes and view application output via collated log-stream.

OpenStack – IaaS Middleware

- Infrastructure components
- Available on demand (Quota or cost limits consumption)
- Deploy without “approval”



VMware vs. OpenStack – Platform Benefits

VMware Model

- Clustered, centrally managed storage and compute resources
- Workload automated management (DRS, Vmotion, Storage Vmotion)
- Capacity rescheduling

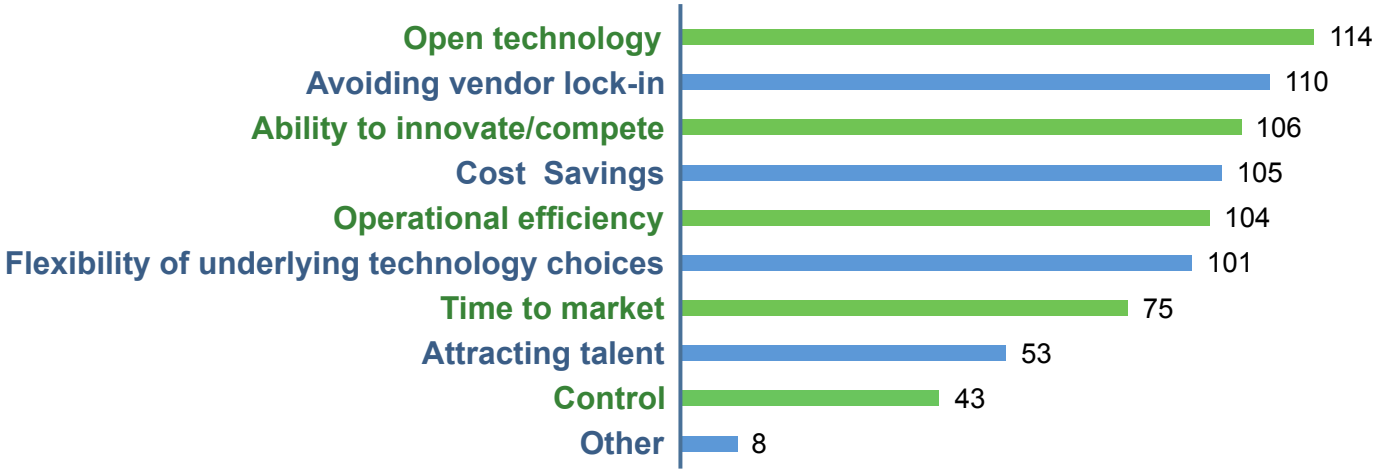
OpenStack Model

- Open Source Foundation managed project
- RESTful APIs for easy automation
- System Abstraction (no vendor lock-in)
- Well known dev model (AWS equivalent)

OneCloud Consulting

Why OpenStack?

1. **Open**: No vendor lock-in
2. **Platform**: Solution for private and public clouds
3. **Cost**: Low software costs, automation reduces opex
4. **Storage**: Low-cost storage solutions – Ceph, Swift, Cinder
5. **Flexibility**: Modular software architecture



Data Source: OpenStack Users Summit Survey, Atlanta 2014

OneCloud Consulting

IaaS - Use Cases

- **Development, Stage and Test**
  - A highly programmable environment supporting approaches to continual delivery
- **Application Deployment Scenarios**
  - Hybrid approaches to deployment, supporting on-premise and off-premise scenarios
- **Web Hosting, Viral Marketing**
  - Auto-scaling and elastic services supporting web volume and viral marketing campaigns
- **High Performance Compute**
  - Short terms needs such as analytics or batch can pay as you go for HPC
- **Temporary Capacity or Bursting**
  - Allows enterprise to extend their environment into cloud capacity on demand
- **Backup and Disaster Recovery**
  - Provides offsite backup and recovery services for storage or machine state

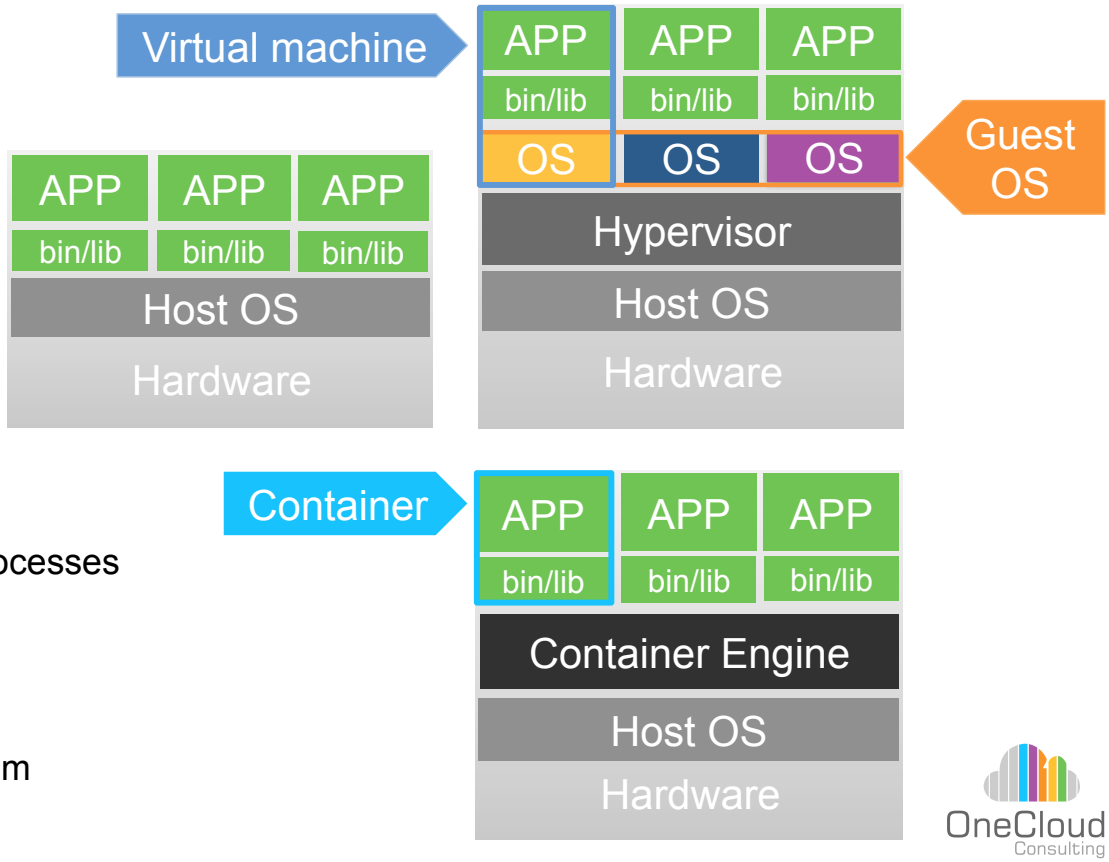
OneCloud Consulting

Cloud Compute

OneCloud
Consulting

Compute - Hosts vs. VMs

- **Bare Metal**
  - x86, ARM, other processor
  - Memory
  - Local “block” storage subsystem
- **Operating System and Process**
  - Linux - Apache
  - Windows – IIS
- **Hypervisor or Container**
  - Hypervisor - Hardware access management and segregation
    - ESX, KVM, Hyper-V, Xen, LPAR
  - Container – OS level segregation of processes
    - Docker/LXC, Solaris containers
- **Virtual Machine**
  - Virtual “Bare Metal”
  - Runs a full copy of the Operating System
  - Runs on Hypervisor



OneCloud Consulting

Compute for Developers

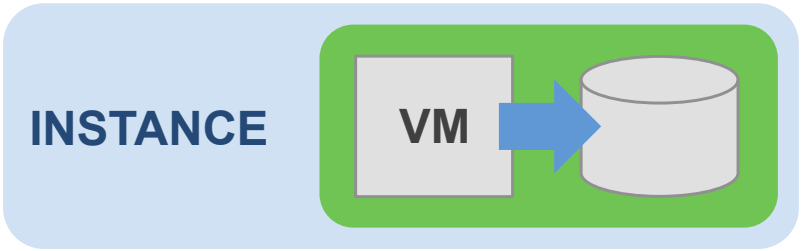
- IaaS/OpenStack provides a Virtual Machine model for compute
- Other options may come along in the future:
  - Ironic project – Managing bare metal compute
  - Docker project – Leveraging Containers rather than full VMs
- All models require that you operate and manage the OS, at least from a “Golden Image” starting point
- Systems maintenance, security patching, backups, all still need to be considered

Cloud Storage

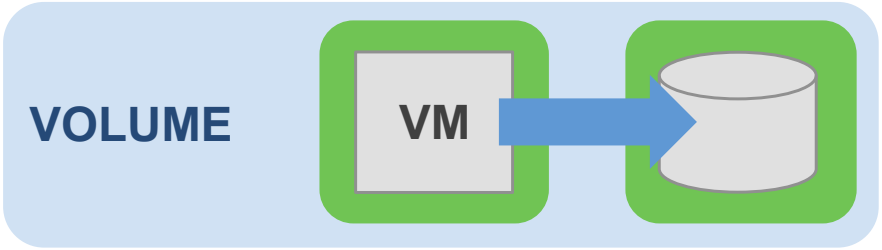
OneCloud
Consulting

Storage Models in Cloud

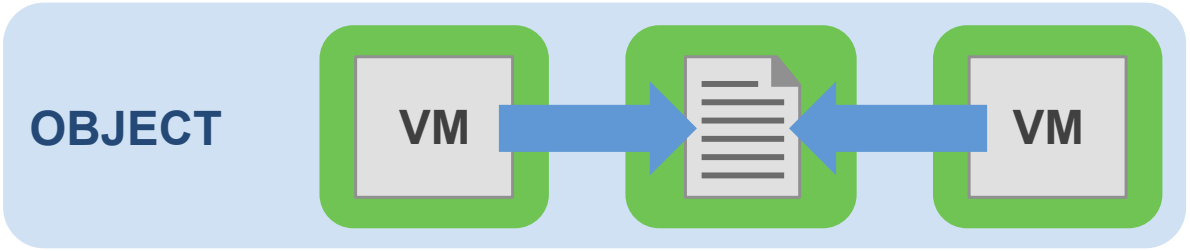
‘Ephemeral’
Block Storage



Persistent
Block Storage



Persistent
Object Storage



OneCloud
Consulting

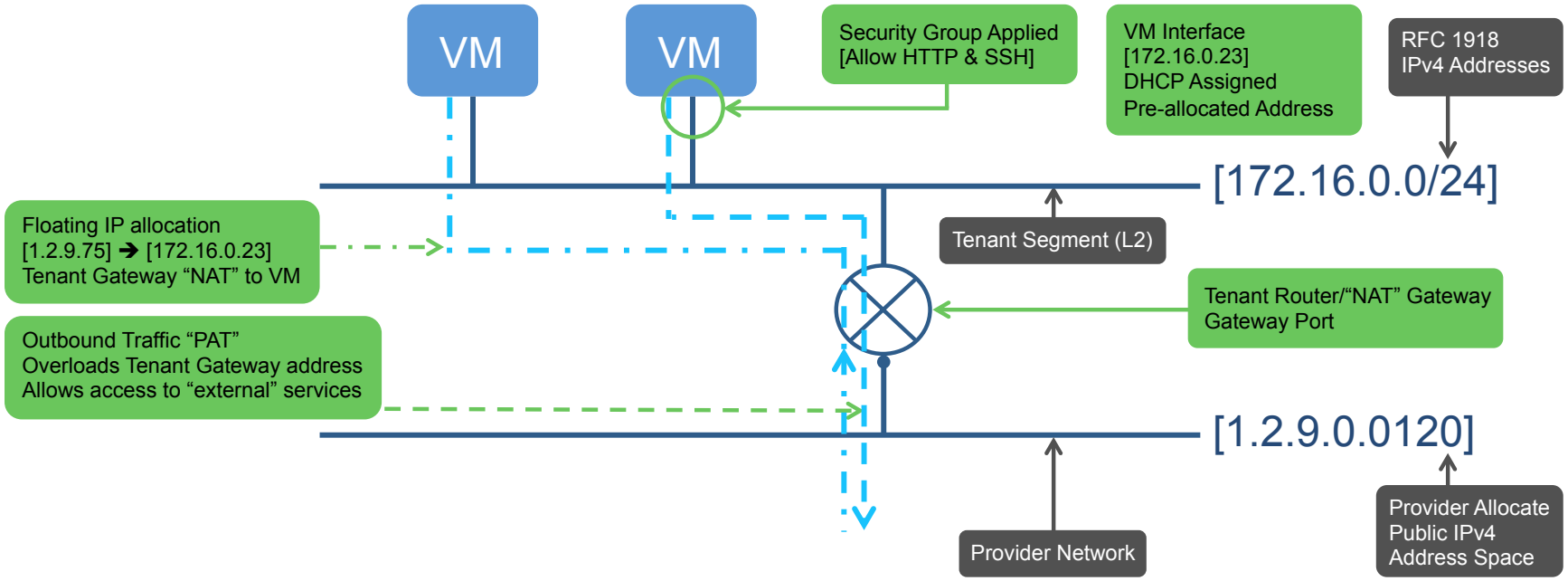
Storage for Developers

- Object Storage may change how an application stores certain aspects of your application (e.g. do you need to put your images into a database, or just a pointer to the object…)
- Decisions on local storage (e.g. for a database) need to consider not only performance, but also resiliency, and recovery options
- Alternate models include deploying applications against re-buildable storage models (e.g. deploy across a distributed storage platform like HDFS)

Cloud Networks

Cloud Network Model

OpenStack Generic Network



OneCloud Consulting

Network for Developers

- Classic models are changing with access to dynamic provisioning and VM level ACLs
  - Other models from the Software Defined Networks environments, like Policy may further abstract the need for complex network systems
- Public IPv4 scarcity is addressed at least in the near term (5-10 years) with Floating IPs, either registered to a Server Load Balancer, or to the application endpoints (e.g. VMs) that need to be accessed from “outside”
- Other systems can still access the public (or “outside”) network through the now generic Cloud network service model

Programmatic Cloud

Programmatic Access

RESTful API Overview

- Method for interaction for “Web Services” (HTTP/HTTPS protocol)
- Stateless
  - Messages only provide the current data, and the protocol itself has no transactional knowledge.
  - Every message needs to contain any data needed to provide continuity of the conversation (token, etc.)
- Uses HTTP methods explicitly
  - POST/PUT/GET/DELETE
- Directory Structure URIs
- Transfers data via XML or JSON
- Data is transferred as an object, the URI specifies the domain of the request

JSON vs. XML

- JSON (JavaScript Object Notation) - Method for serializing data from JavaScript environments
- XML – Xtensible Markup Language, can “describe” objects
- Heavily used as an easier to manipulate (programmatically)
- Schema is implied
  - Key:Value model rather than wrapped SGML style markup

```
{
   "network": {
      "cidr": "192.168.0.0/24",
      "id": "f212726e-6321-4210-9bae-a13f5a33f83f",
      "label": "superprivate_xml"
   }
}
```

```
<network>
   <cidr>192.168.0.0/24</cidr>
   <id>f212726e-6321-4210-9bae-a13f5a33f83f</id>
   <label>superprivate_xml</label>
</network>
```

```
network:
  cidr: "192.168.0.0/24”
  id: "f212726e-6321-4210-9bae-a13f5a33f83f”
  label: "superprivate_xml”
```

YAML is a JSON variant that is considered easier for Humans to manipulate

OneCloud Consulting

OpenStack APIs Simplified

- OpenStack Identity Service API v2

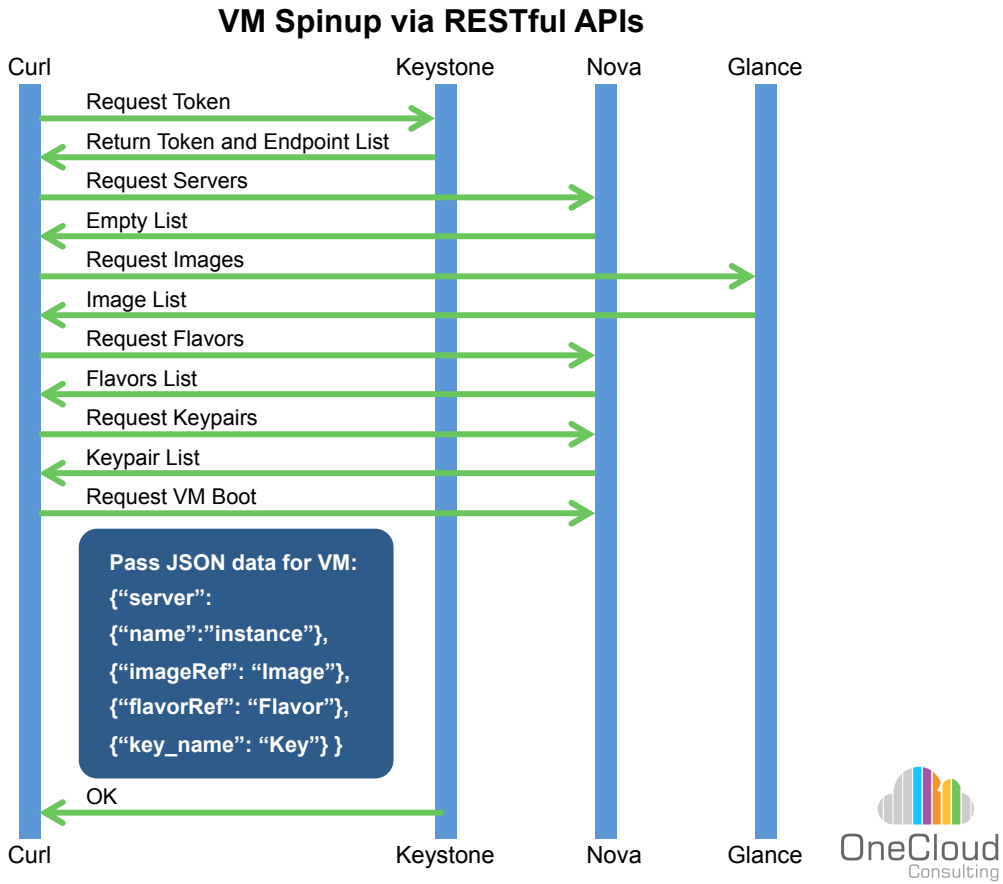
Category	Sub Category	Verb	URI	Description
Client API	Authenticate	POST	v2.0/tokens	Authenticates and generates a token.
Admin API	Token	GET	v2.0/tokens/{tokenId}/endpoints	Lists the endpoints associated with a specified token.
Admin API	User	GET	v2.0/users/{?name}	Gets detailed information about a specified user by user name.

- OpenStack Networking API v2

Category	Verb	URI	Description
Networks	GET	v2.0/networks	Lists networks to which the specified tenant has access.
Networks	PUT	v2.0/networks/{network_id}	Updates a specified network.
Subnets	DELETE	v2.0/subnets/{subnet_id}	Deletes a specified subnet.

API Demo

- We can control all of OpenStack via RESTful API Calls
- Let's use this model and the 'curl' application to do this, from the request for authentication to the deployment of a virtual machine.



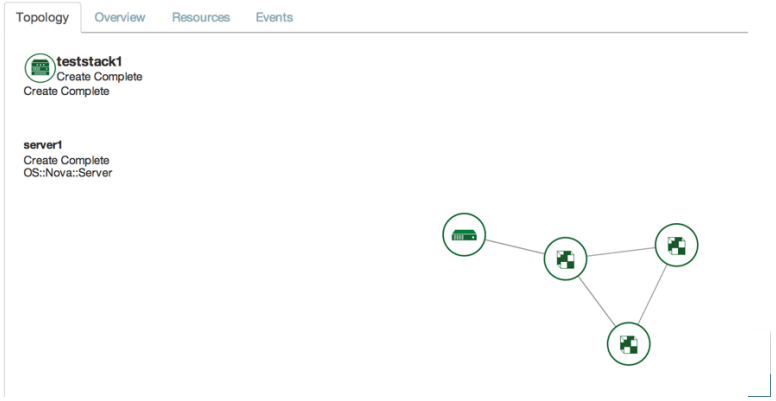
OneCloud Consulting

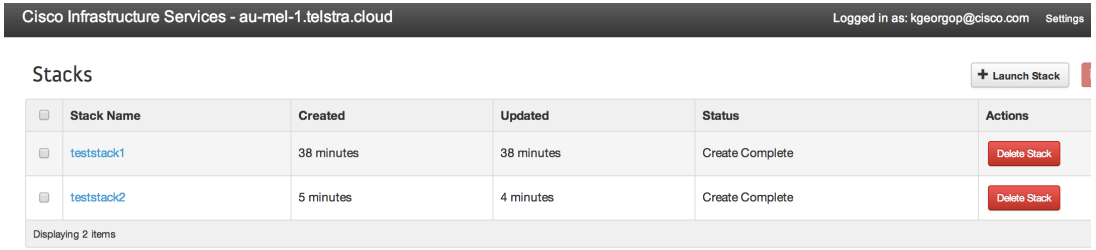
Cloud Application Templates

- OpenStack Heat Orchestration Templates
  - *Template based orchestration for describing a cloud application by executing appropriate OpenStack API calls to generate running cloud applications <…> Allows creation of most OpenStack resource types (such as instances, floating ips, volumes, security groups, users, etc)*
- Templates can be provided by Cisco, Partner, and/or uploaded by the end tenant

```
heat_template_version: 2013-05-23
description: Heat template to create 2 instances and 1 network
resources:
  private_net:
    type: OS::Neutron::Net
    properties:
      name: ext-network-by-heat
  private_subnet:
    type: OS::Neutron::Subnet
    properties:
      name: ext-subnet-by-heat
      network_id: { get_resource: private_net }
      cidr: 192.168.3.0/24
      gateway_ip: 192.168.3.1
  server1:
    type: OS::Nova::Server
    properties:
      name: "Server1"
      image: rhel-6.5_x86_64-2014-05-23d-telstra-v3
      flavor: "GP-Small"
      key_name: kgeorgop
```

Stack Detail: teststack1





Example HOT template

```
heat_template_version: 2013-05-23

description: Test Template
```

```
parameters:
  ImageName:
    type: string
    description: Image use to boot a server
  NetName:
    type: string
    description: Network ID for the server
```

```
resources:
  server1:
    type: OS::Nova::Server
    properties:
      name: "Test server"
      image: { get_param: ImageName }
      flavor: "m1.tiny"
      networks:
        - port: { get_resource: server1_port }
```

```
server1_port:
    type: OS::Neutron::Port
    properties:
      network: { get_param: NetName }
```

```
outputs:
  server1_private_ip:
    description: IP address of the server in the private network
    value: { get_attr: [ server1, first_address ] }
```

OneCloud Consulting

HEAT Application Deployment

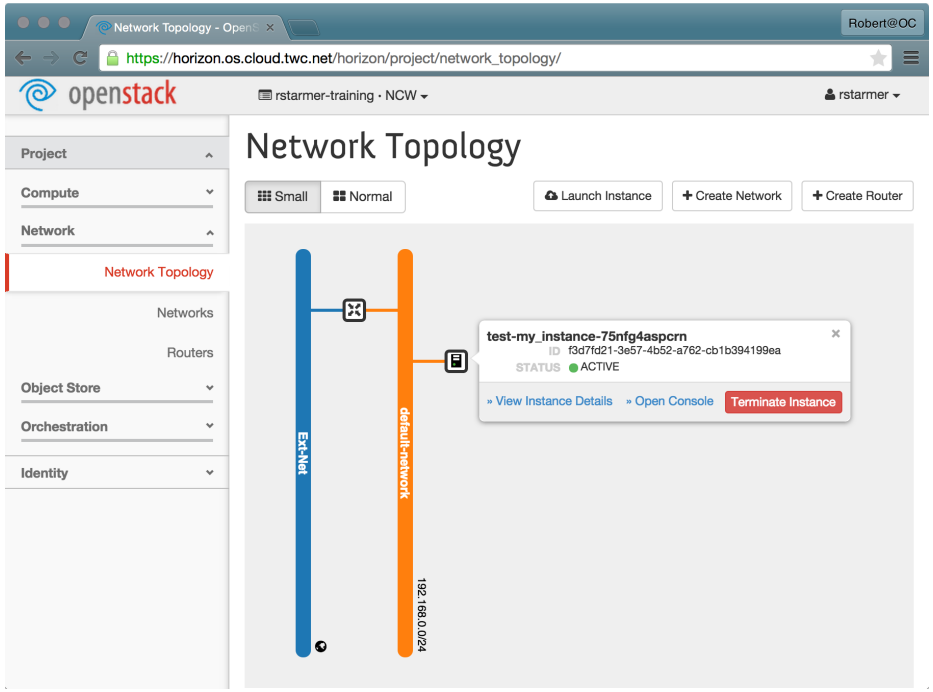
- http://50.250.253.88:8080/v1/AUTH_e71d06beb59a40d1a9e29df6b014444e/osbootcamp/Simple_HOT.yaml
- **http://tinyurl.com/simple-hot**

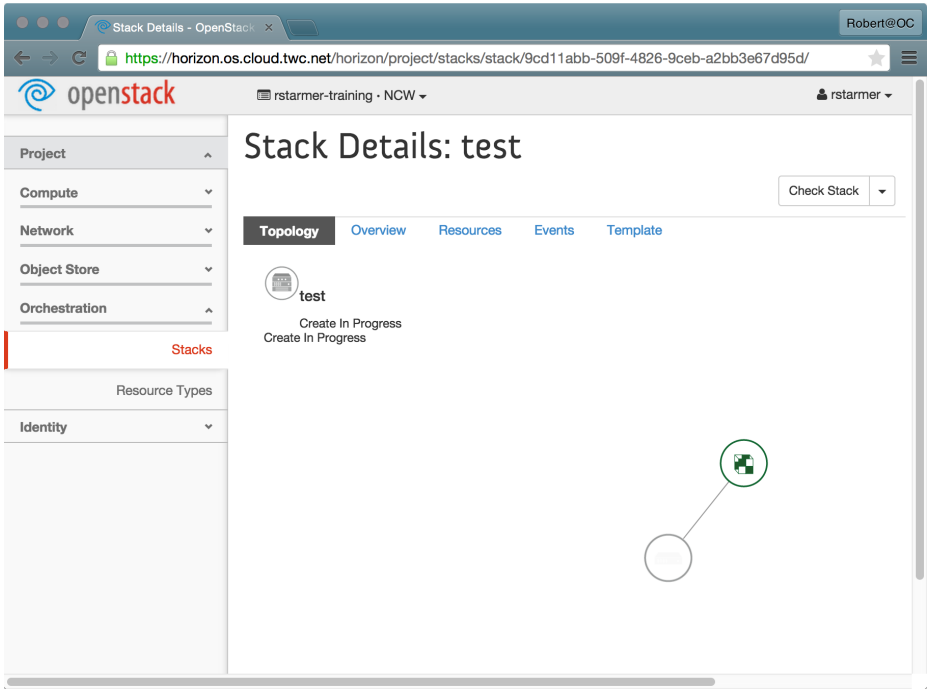
- Deploy using Horizon
- Deploy using the CLI, and pass the ImageName and NetName parameters…

- heat stack-create --url **http://tinyurl.com/simple-hot** --parameters "{ImageName=Ubuntu-Server-14.04,NetName=default-network}" --name test

OneCloud Consulting

5 Second Orchestration Topology





OneCloud Consulting