Lecture15-16.2 – OpenStack & Comparing and Contrasting AWS with NeCTAR Cloud

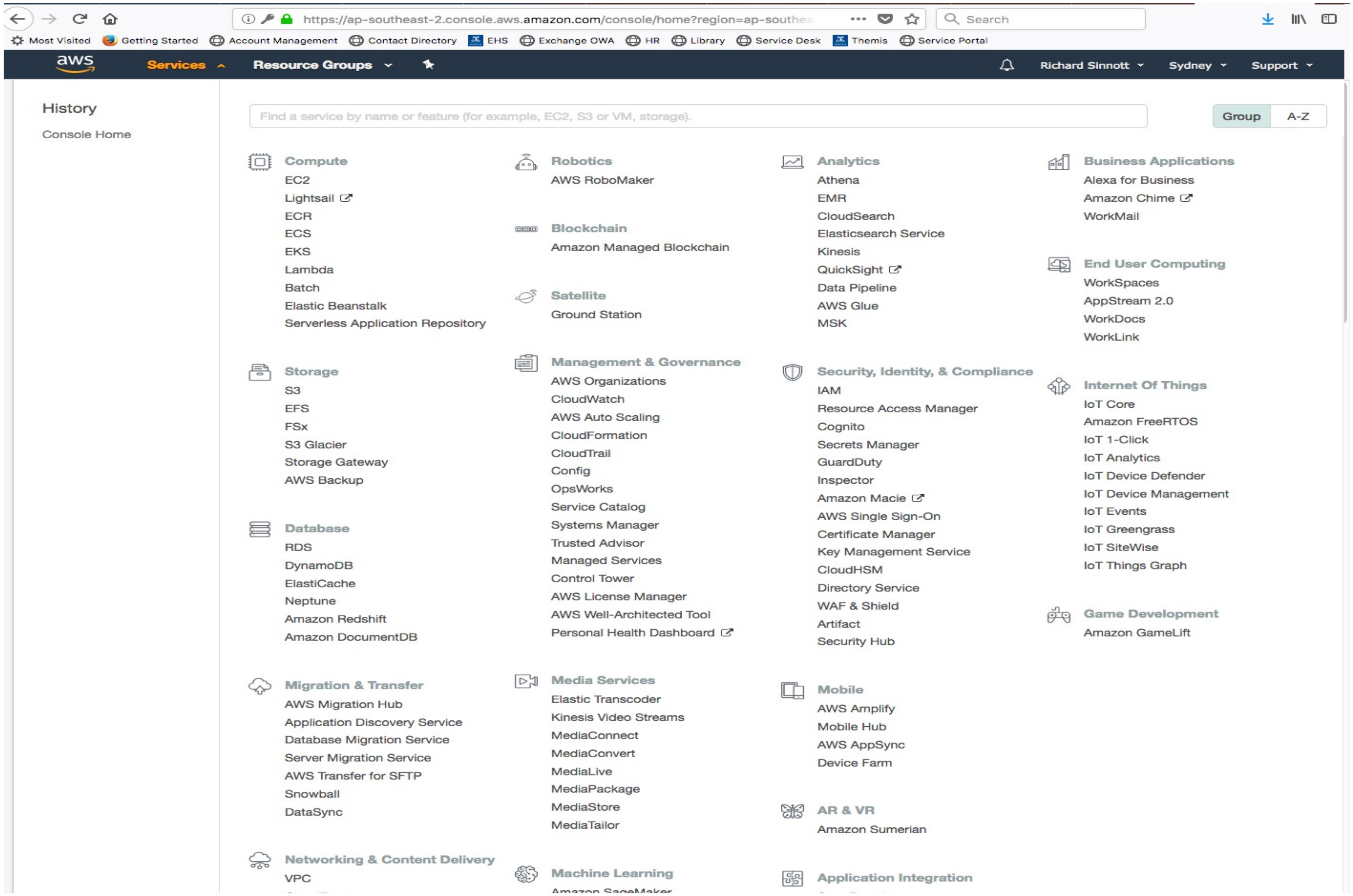
Professor Richard O. Sinnott & Yao Pan
University of Melbourne
rsinnott@unimelb.edu.au

nectar & amazon web services

- UniMelb/NeCTAR Research Cloud
  - open source Cloud platform
    - openStack
      - overview of the major services*
- AWS (http://aws.amazon.com)
  - mainstream Cloud platform
    - Examples of the kinds of services that are available

*note that not all openstack services are available (yet!?) on the MRC/NeCTAR Research Cloud

THE UNIVERSITY OF MELBOURNE nectar & amazon web services



History

Console Home

Compute
EC2
Lightsail
ECR
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup

Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon Redshift
Amazon DocumentDB

Migration & Transfer
AWS Migration Hub
Application Discovery Service
Database Migration Service
Server Migration Service
AWS Transfer for SFTP
Snowball
DataSync

Networking & Content Delivery
VPC

Robotics
AWS RoboMaker

Blockchain
Amazon Managed Blockchain

Satellite
Ground Station

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashboard

Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConnect
MediaConvert
MediaLive
MediaPackage
MediaStore
MediaTailor

Machine Learning

Analytics
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
QuickSight
Data Pipeline
AWS Glue
MSK

Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
Artifact
Security Hub

Mobile
AWS Amplify
Mobile Hub
AWS AppSync
Device Farm

AR & VR
Amazon Sumerian

Application Integration

Business Applications
Alexa for Business
Amazon Chime
WorkMail

End User Computing
WorkSpaces
AppStream 2.0
WorkDocs
WorkLink

Internet Of Things
IoT Core
Amazon FreeRTOS
IoT 1-Click
IoT Analytics
IoT Device Defender
IoT Device Management
IoT Events
IoT Greengrass
IoT SiteWise
IoT Things Graph

Game Development
Amazon GameLift

OpenStack

- Began in 2010 as a joint project between Rackspace and NASA
- Offers free and open-source software platform for cloud computing for (mostly) IaaS
- Consists of interrelated components (services) that control / support compute, storage, and networking resources
- Often used through web-based dashboards, through command-line tools, or programmatically through ReSTful APIs
- Released under the terms of the Apache License
- Managed/coordinated by the OpenStack Foundation
  - non-profit corporate entity established in 2012 to promote OpenStack software and its community
  - Over 500 companies have since joined the project

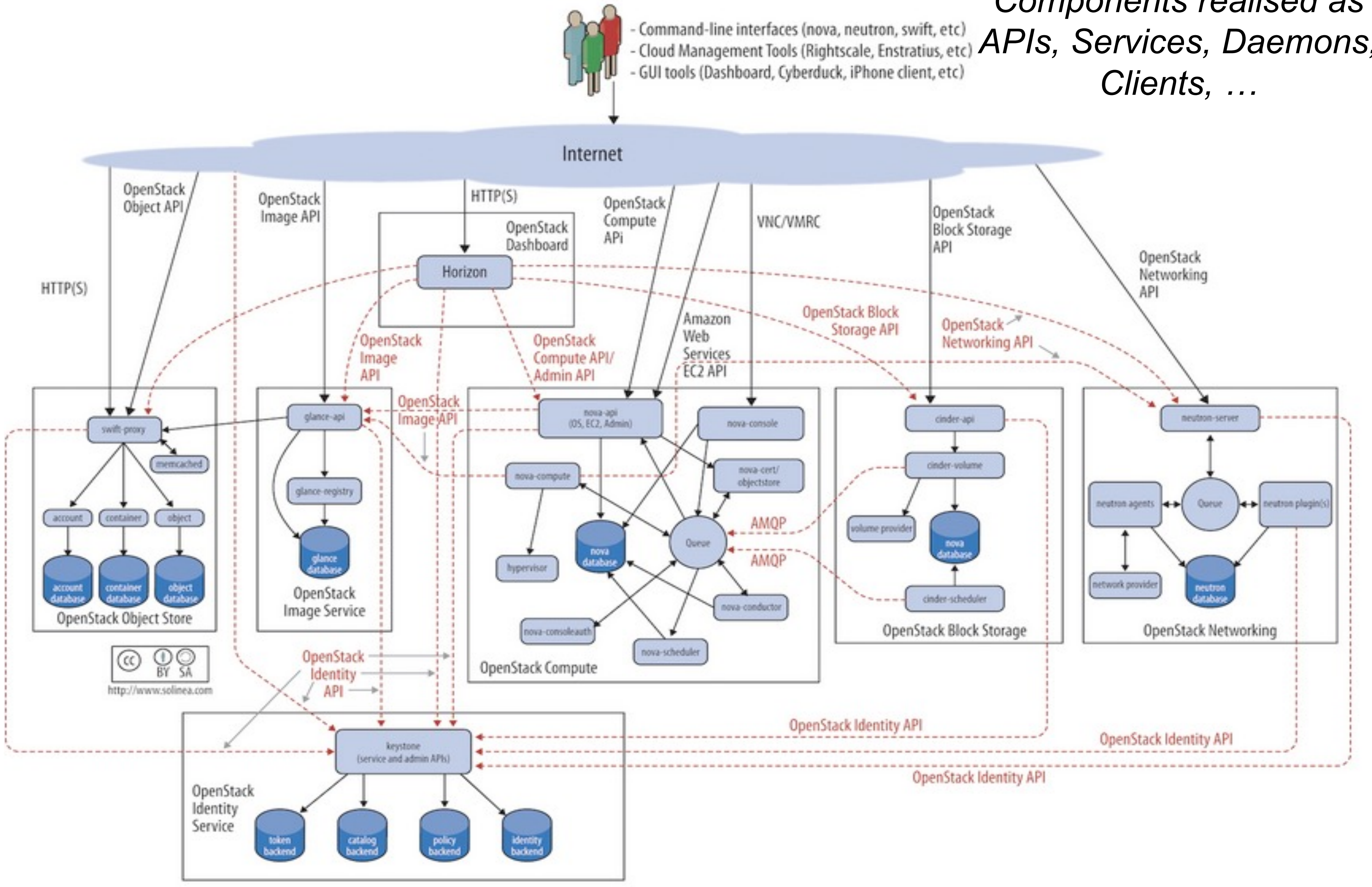
OpenStack Components

- Many associated/underpinning services
  - Compute Service (code-named Nova)
  - Image Service (code-named Glance)
  - Block Storage Service (code named Cinder)
  - Object Storage Service (code-named Swift)
  - Security Management (code-named Keystone)
  - Orchestration Service (code-named Heat)
  - Network Service (code-named Neutron)
  - Container Service (code-named Zun)
  - Database service (code-named Trove)
  - Dashboard service (code-named Horizon)
  - Search service (code-named Searchlight)
  - …

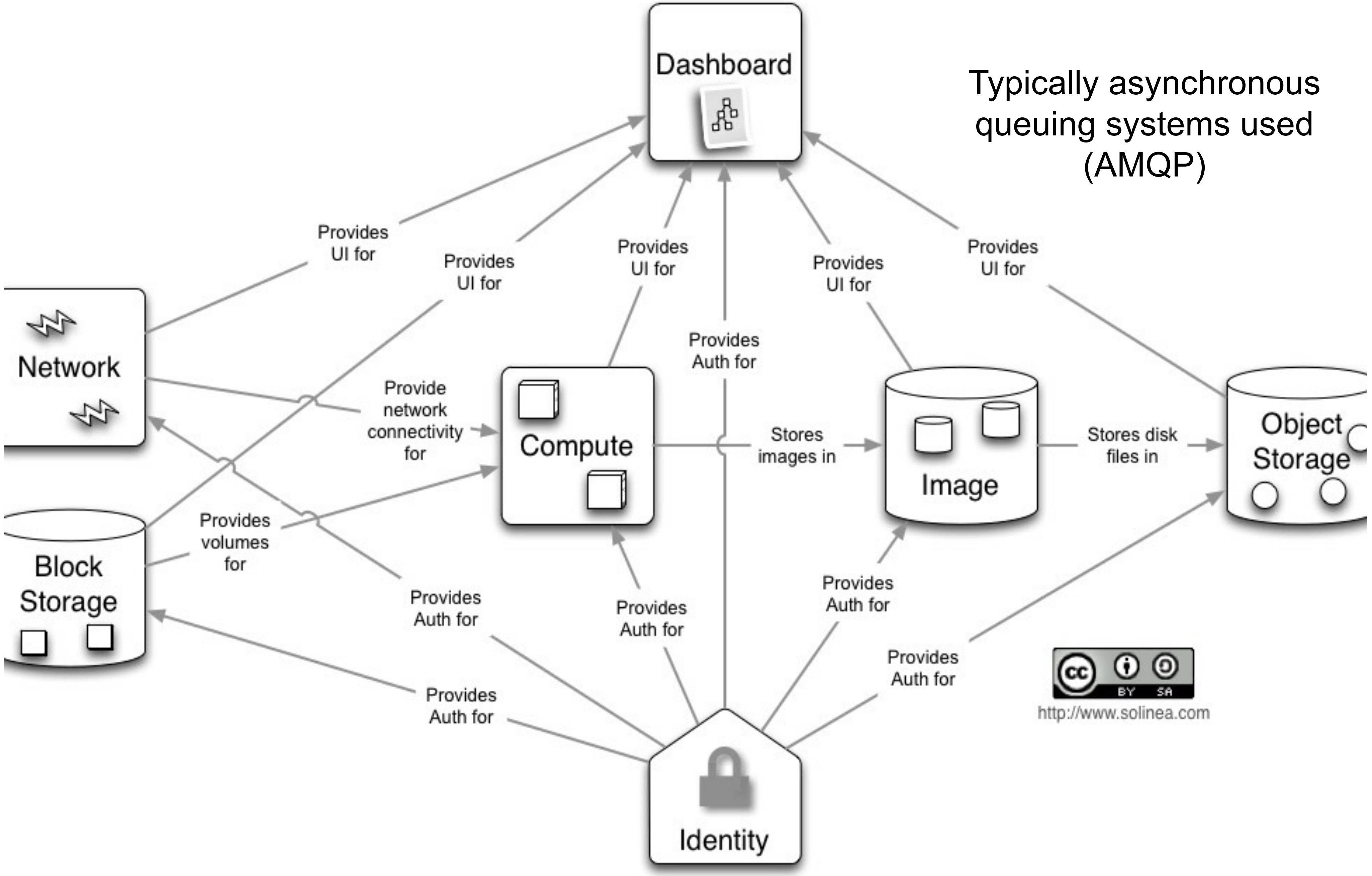
https://www.openstack.org/software/project-navigator/openstack-components#openstack-services

(Simplified) OpenStack Architecture

Components realised as APIs, Services, Daemons, Clients, …



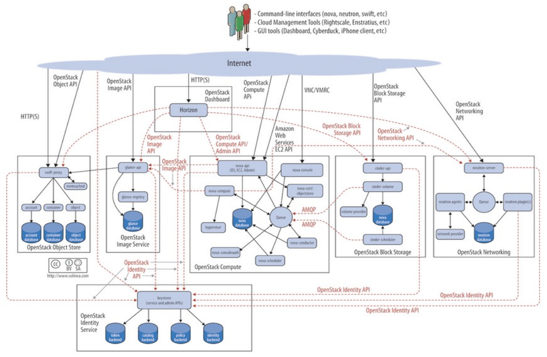
(Simplified) User Perspective



Key Services::Identity Service

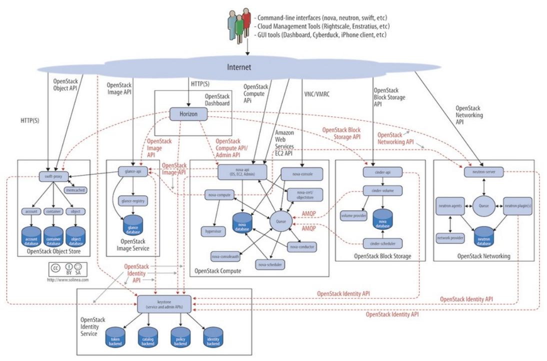
- Keystone
  - Provides an authentication and authorization* service for OpenStack services
    - Tracks users/permissions
  - Provides a catalog of endpoints for all OpenStack services
    - Each service registered during install
      - Know where they are and who can do what with them
    - Project membership; firewall rules; image mgt; …

  - *Generic authorization system for openStack…
    - more in security lecture



Key Services::Compute

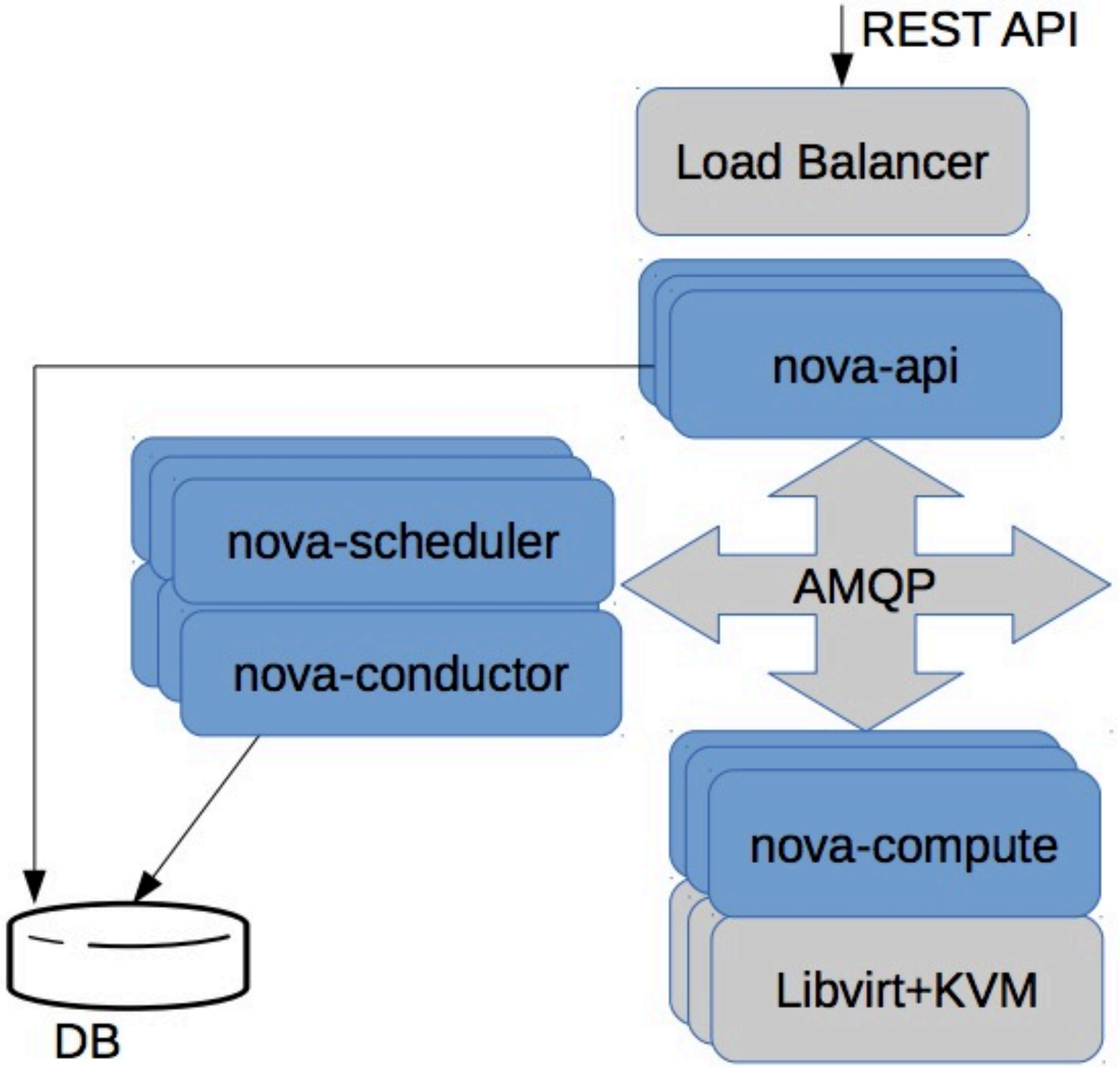
- Nova
  - Manages the lifecycle of compute instances in an OpenStack environment
  - Responsibilities include spawning, scheduling and decommissioning of virtual machines on demand
  - Virtualisation agnostic
    - Libvirt
      - open source API, daemon and tools for managing platform virtualisation including support for Kernel based virtual machine (KVM), Quick Emulator (QEMU), Xen, Lightweight Linux Container System (LXC)
    - XenAPI, Hyper-V, VMWare ESX,
    - Docker (more later from Luca)
    - …


Command-line interfaces (nova, neutron, swift, etc)
Cloud Management Tools (Rightscale, Enstratius, etc)
GUI tools (Dashboard, Cyberduck, iPhone client, etc)
Internet
Horizon
OpenStack Object Store
OpenStack Image Service
OpenStack Compute
OpenStack Block Storage
OpenStack Networking
OpenStack Identity Service

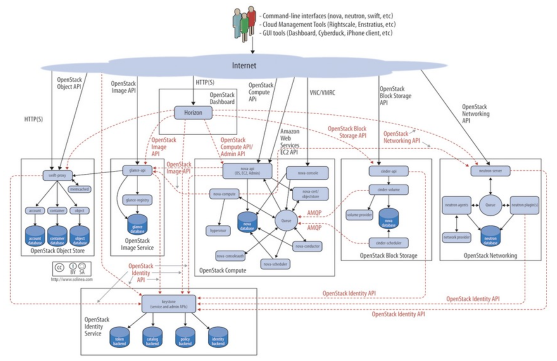
Key Services::Compute

- Nova
  - API
    - Nova-api - accepts/responds to end user API calls; supports openStack Compute & EC2 & admin APIs
  - Compute Core
    - Nova-compute - Daemon that creates/terminates VMs through hypervisor APIs
    - Nova-scheduler - schedules VM instance requests from queue and determines which server host to run
    - Nova-conductor - Mediates interactions between compute services and other components, e.g. image database
  - Networking
    - Nova-network - Accepts network tasks from queue and manipulates network, e.g. changing IPtable rules
  - Image Mgt, Client Tools, …

Simplified (Scalable) Nova Architecture

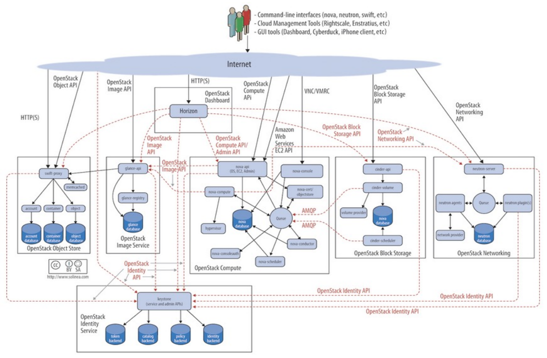


I need a VM with:
•64Gb memory,
•8vCPUs,
•in Melbourne,
•running Ubuntu 12.04,
•…



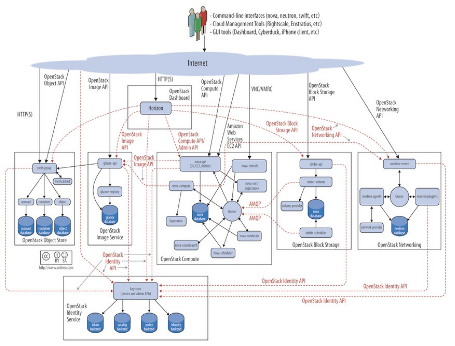
Key Services::Object Storage

- Swift
  - Stores and retrieves arbitrary unstructured data objects via RESTful API, e.g. VM images and data
    - Not POSIX (atomic operations); eventual consistency
  - Fault tolerant with data replication and scale-out architecture.
    - Available from anywhere; persists until deleted
    - Allows to write objects and files to multiple drives, ensuring the data is replicated across a server cluster
  - Can be used with/without Nova/compute
  - Client; admin support
    - e.g. Swift client – allows users to submit commands to ReST API through command line clients to configure/ connect object storage to VMs



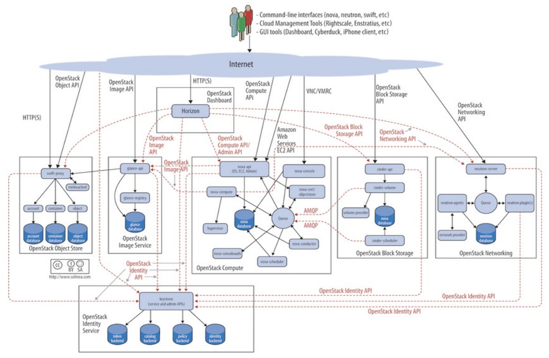
Key Services::Block Storage

- Cinder
  - Provides persistent block storage to virtual machines (instances) and supports creation and management of block storage devices
  - Cinder access associated with a VM
    - Cinder-api – routes requests to cinder-volume
    - Cinder-volume – interacts with block storage service and scheduler to read/write requests; can interact with multiple flavours of storage (flexible driver architecture)
    - Cinder-scheduler – selects optimal storage provider node to create volumes (ala nova-scheduler)
    - Cinder-backup – provides backup to any types of volume to backup storage provider
      - Can interact with variety of storage solutions



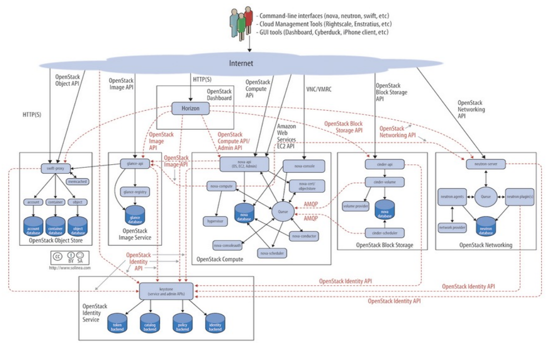
Key Services::Image Service

- Glance
  - Accepts requests for disk or server images and their associated metadata (from Swift) and retrieves / installs (through Nova)
    - Glance-api – image discovery, retrieval and storage requests
    - Glance-registry – stores, processes and retrieves metadata about images, e.g. size and type
      - Ubuntu 14.04…?
      - My last good snapshot…?



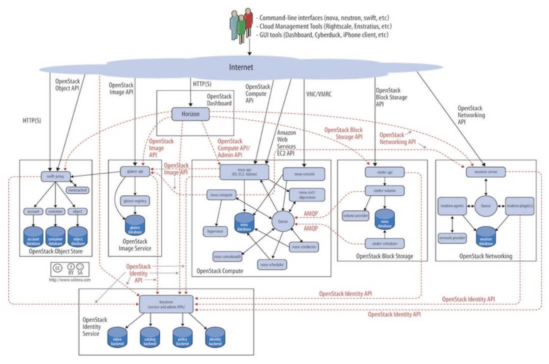
Key Services::Networking

- Neutron
  - Supports networking of OpenStack services
  - Offers an API for users to define networks and the attachments into them, e.g. switches, routers
  - Pluggable architecture that supports multiple networking vendors and technologies
  - Neutron-server – accepts and routes API requests to appropriate plug-ins for action
    - Port management, e.g. default SSH, VM-specific rules, …
    - More broadly configuration of availability zone networking, e.g. subnets, DHCP, …



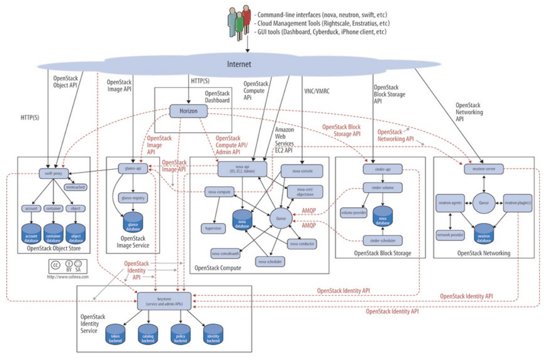
Key Services::Dashboard

- Horizon
  - Provides a web-based self-service portal to interact with underlying OpenStack services, such as launching an instance, assigning IP addresses and configuring access controls.
  - Based on Python/Django web application
  - Mod_wsgi
    - Apache plug realising web service gateway interface
  - Requires Nova, Keystone, Glance, Neutron
  - Other services optional…



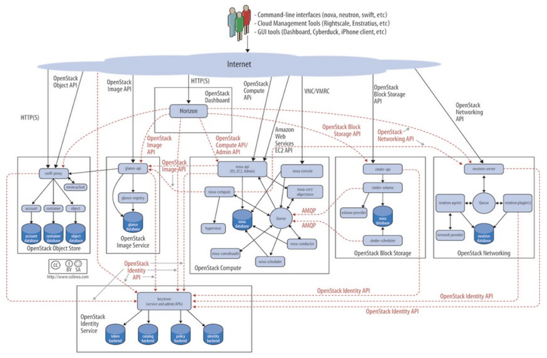
Key Services::Database Service

- Trove
  - Provides scalable and reliable Cloud database (DBaaS) functionality for both relational and non-relational database engines (for the masses!)
    - Resource isolation, high performance, automates deployment, config, patching, backups, restores, monitoring…
      - e.g. Set up 3 VMs with mySQL, CouchDB, MongoDB
    - Use image service for each DB type and trove-manage to offer them to tenants/user communities



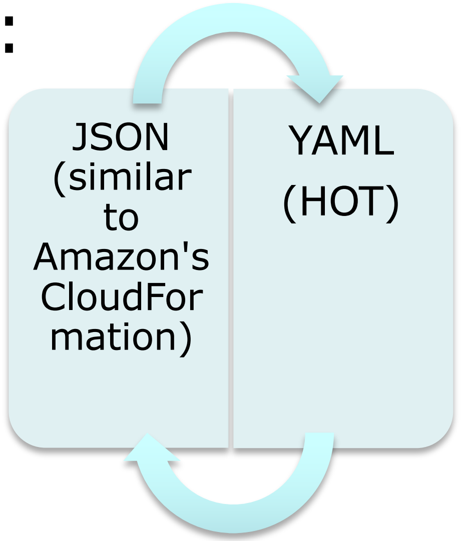
Key Services::Data Processing Service

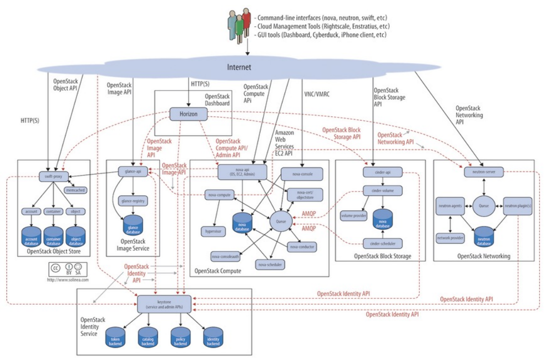
- Sahara
  - Provides capabilities to provision and scale Hadoop clusters in OpenStack by specifying parameters such as Hadoop version, cluster topology and node hardware details
    - User fills in details and Sahara supports the automated deployment of infrastructure with support for addition/removal of worker nodes on demand



Key Services::Orchestration Service

- Heat
  - Template-driven service to manage lifecycle of applications deployed on Openstack
  - *Stack*: Another name for the template and procedure behind creating infrastructure and the required resources from the template file
  - Can be integrated with automation tools such as Chef, Puppet, Ansible, etc.
  - Template format:





Key Services::Orchestration Service

- Heat details
  - *heat_template_version*: allows to specify which version of Heat, the template was written for (*optional*)
  - *Description*: describes the intent of the template to a human audience (*optional*)
  - *Parameters*: the arguments that the user might be required to provide (*optional*)
  - *Resources*: the specifications of resources that are to be created (*mandatory*)
  - *Outputs*: any expected values that are to be returned once the template has been processed (*optional*)

Creating Stacks in MRC/NeCTAR

1) Create the template file according to your requirements
2) Provide environment details (name of key file, image id, etc)
3) Select a name for your stack and confirm the parameters
4) Make sure rollback checkbox is marked, so if anything goes wrong, all partially created resources get dumped too
5) Wait for the magic to happen!

Demonstration of HEAT

- Creating a Wordpress website MRC/NeCTAR-style
- Creating a Wordpress website AWS-style

References

1) NeCTAR sample template repository
(https://github.com/NeCTAR-RC/heat-templates)