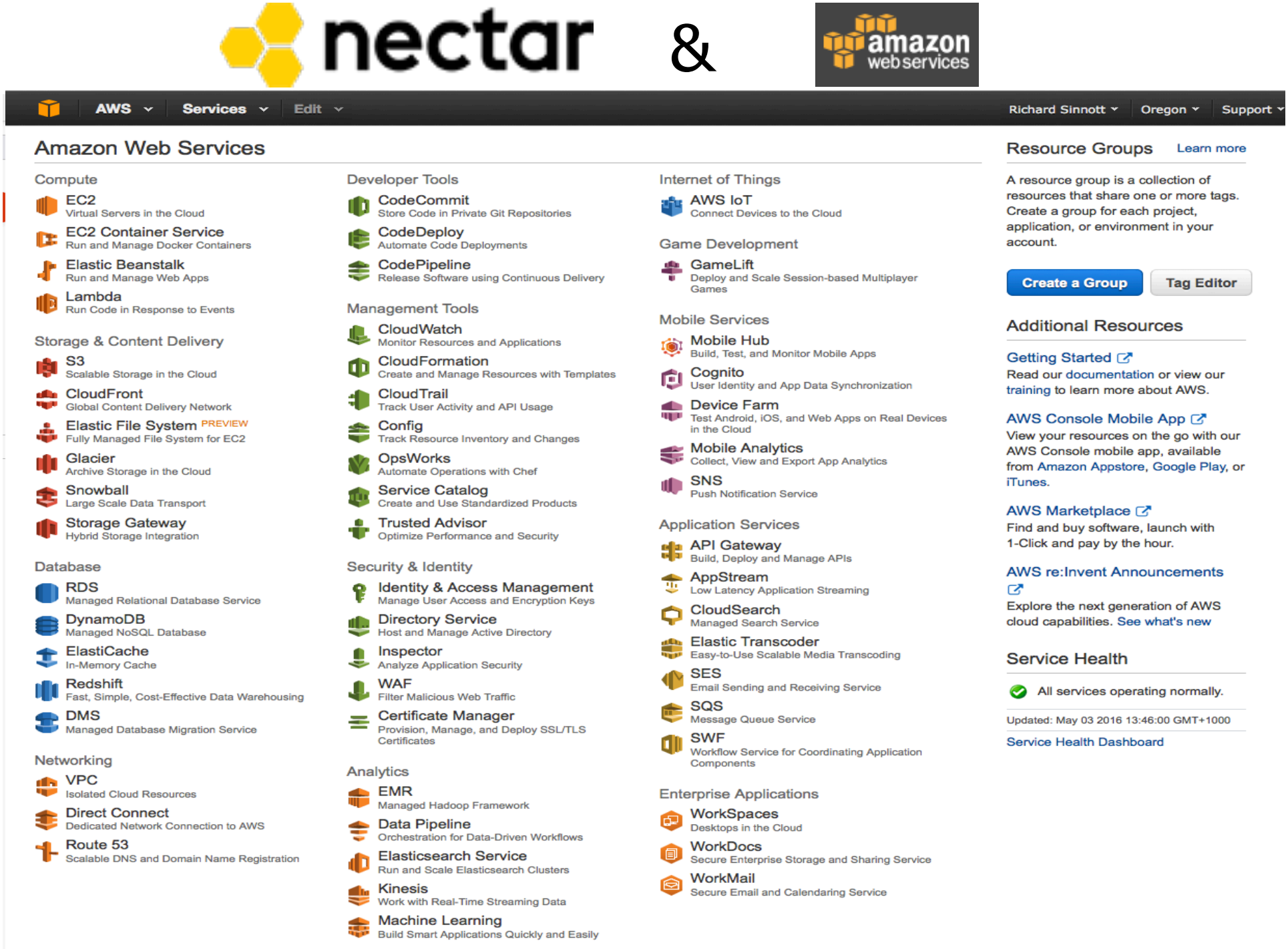
Lecture 9.2 - Compare and Contrast Amazon EC2 with NeCTAR Research Cloud

Professor Richard O. Sinnott & Farzad Khodadadi
University of Melbourne
rsinnott@unimelb.edu.au

&

- 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 NeCTAR Research Cloud


nectar
&
amazon web services
AWS
Services
Edit
Richard Sinnott
Oregon
Support
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
EC2 Container Service
Run and Manage Docker Containers
Elastic Beanstalk
Run and Manage Web Apps
Lambda
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
CloudFront
Global Content Delivery Network
Elastic File System PREVIEW
Fully Managed File System for EC2
Glacier
Archive Storage in the Cloud
Snowball
Large Scale Data Transport
Storage Gateway
Hybrid Storage Integration
Database
RDS
Managed Relational Database Service
DynamoDB
Managed NoSQL Database
ElastiCache
In-Memory Cache
Redshift
Fast, Simple, Cost-Effective Data Warehousing
DMS
Managed Database Migration Service
Networking
VPC
Isolated Cloud Resources
Direct Connect
Dedicated Network Connection to AWS
Route 53
Scalable DNS and Domain Name Registration
Developer Tools
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
Management Tools
CloudWatch
Monitor Resources and Applications
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
Config
Track Resource Inventory and Changes
OpsWorks
Automate Operations with Chef
Service Catalog
Create and Use Standardized Products
Trusted Advisor
Optimize Performance and Security
Security & Identity
Identity & Access Management
Manage User Access and Encryption Keys
Directory Service
Host and Manage Active Directory
Inspector
Analyze Application Security
WAF
Filter Malicious Web Traffic
Certificate Manager
Provision, Manage, and Deploy SSL/TLS Certificates
Analytics
EMR
Managed Hadoop Framework
Data Pipeline
Orchestration for Data-Driven Workflows
Elasticsearch Service
Run and Scale Elasticsearch Clusters
Kinesis
Work with Real-Time Streaming Data
Machine Learning
Build Smart Applications Quickly and Easily
Internet of Things
AWS IoT
Connect Devices to the Cloud
Game Development
GameLift
Deploy and Scale Session-based Multiplayer Games
Mobile Services
Mobile Hub
Build, Test, and Monitor Mobile Apps
Cognito
User Identity and App Data Synchronization
Device Farm
Test Android, iOS, and Web Apps on Real Devices in the Cloud
Mobile Analytics
Collect, View and Export App Analytics
SNS
Push Notification Service
Application Services
API Gateway
Build, Deploy and Manage APIs
AppStream
Low Latency Application Streaming
CloudSearch
Managed Search Service
Elastic Transcoder
Easy-to-Use Scalable Media Transcoding
SES
Email Sending and Receiving Service
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing Service
WorkMail
Secure Email and Calendaring Service
Resource Groups
Learn more
A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.
Create a Group
Tag Editor
Additional Resources
Getting Started
Read our documentation or view our training to learn more about AWS.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
AWS re:Invent Announcements
Explore the next generation of AWS cloud capabilities. See what's new
Service Health
All services operating normally.
Updated: May 03 2016 13:46:00 GMT+1000
Service Health Dashboard

OpenStack

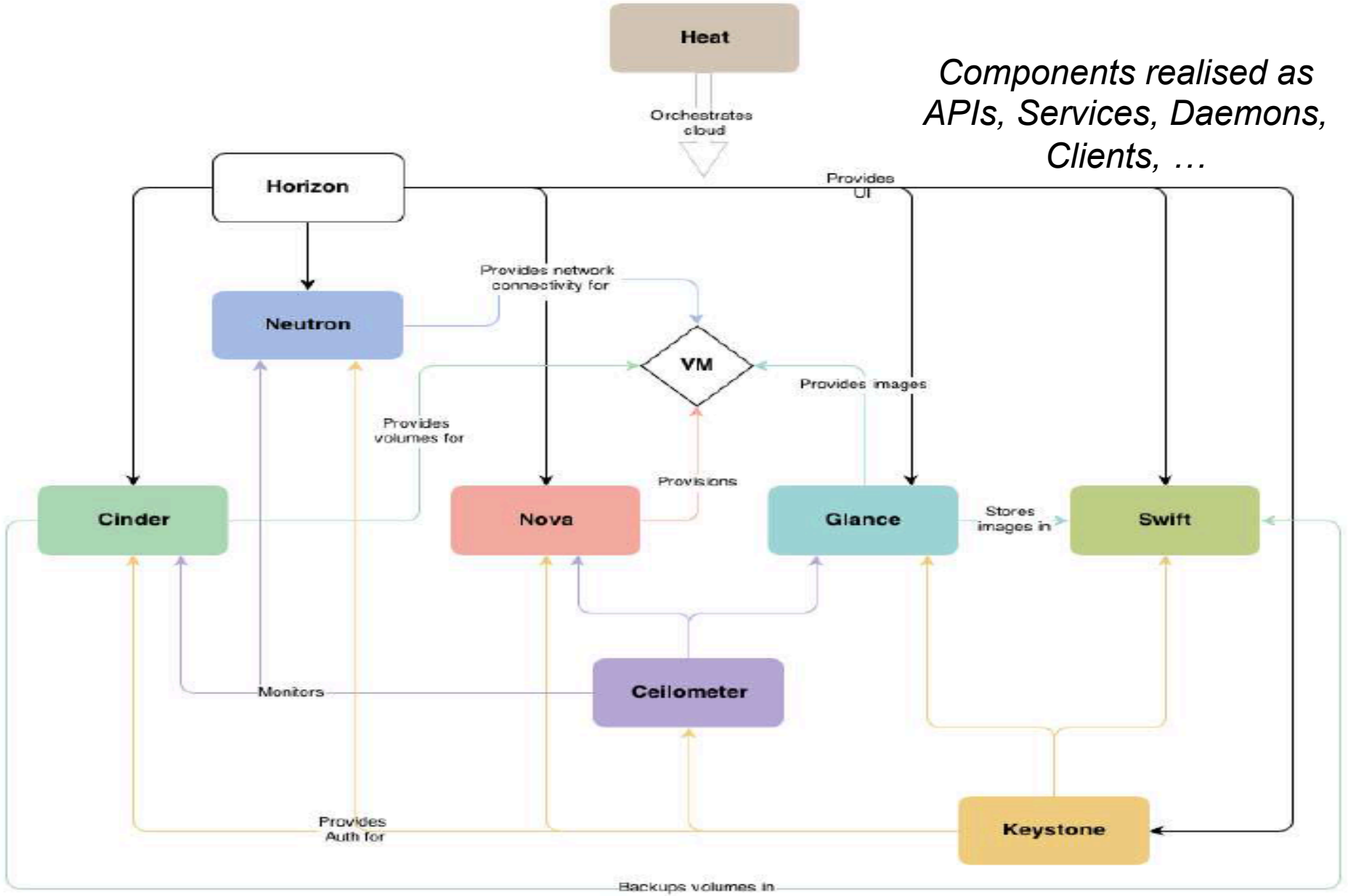
- Began in 2010 as a joint project between Rackspace and NASA
- Offers free and open-source software platform for cloud computing for (mostly) IaaS
- Consists of interrelated components (services) that control / support compute, storage, and networking resources
- Often used through web-based dashboards, 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)
  - Metering Service (code-named Ceilometer)
  - Database service (code-named Trove)
  - Dashboard service (code-named Horizon)
  - Search service (code-named Searchlight)
  - Security API (code named Barbican)…

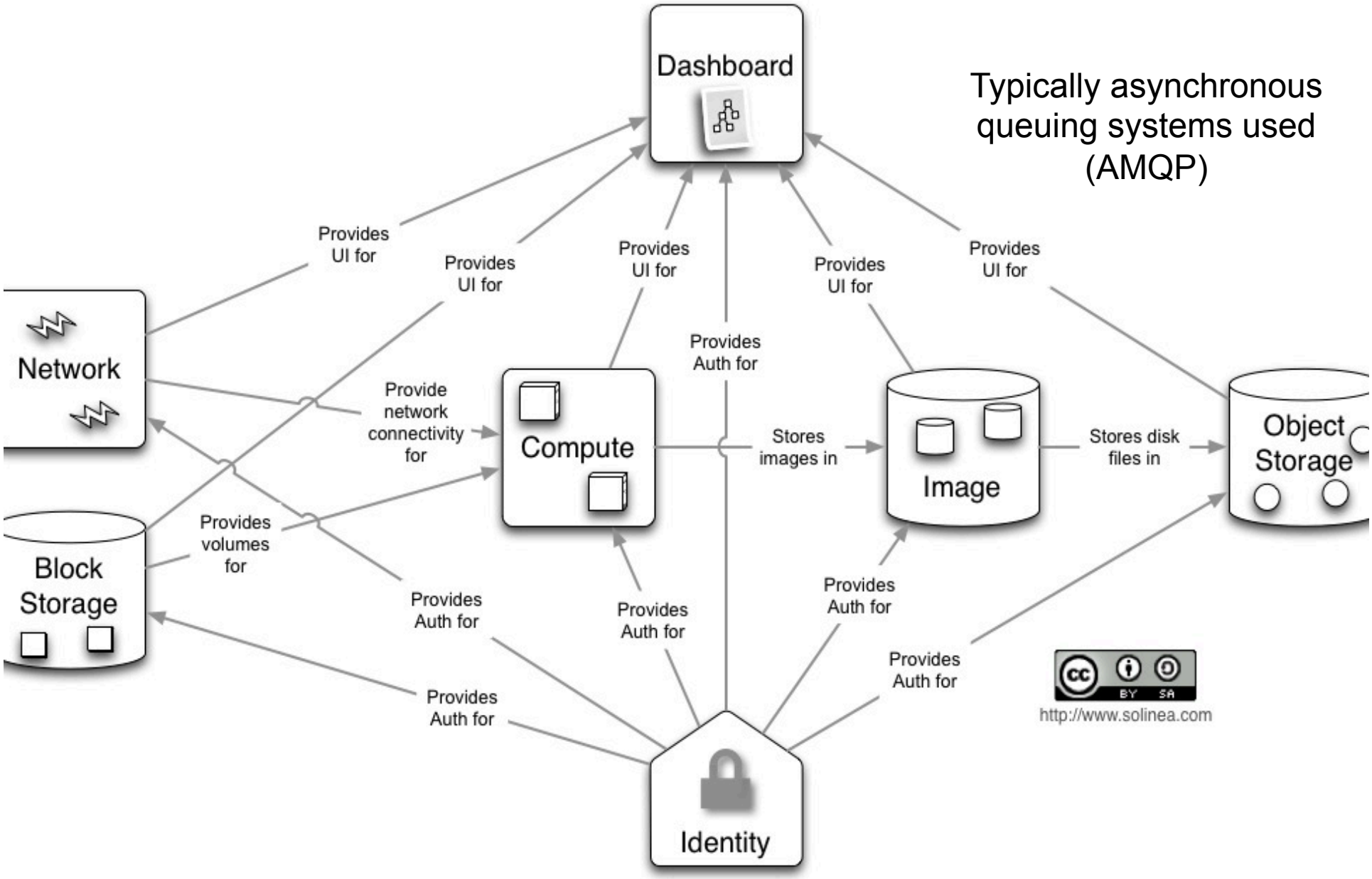
http://docs.openstack.org/

(V. simplified) OpenStack Architecture



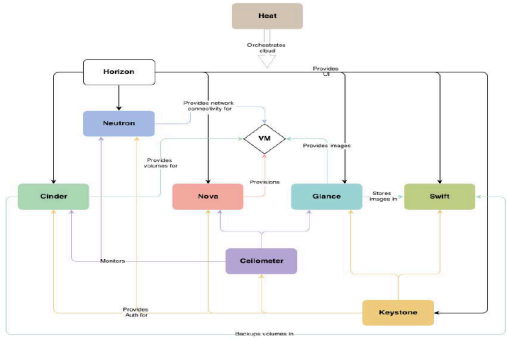
Components realised as APIs, Services, Daemons, Clients, …

(Simplified) User Perspective



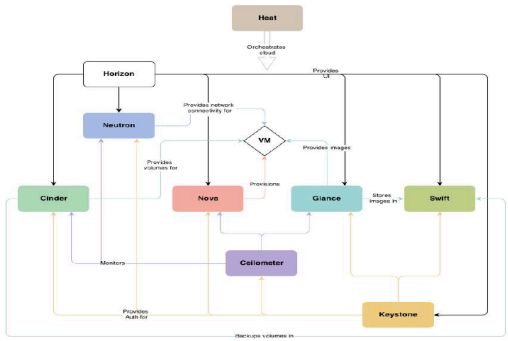
Key Services::Identity Service

- 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 next lecture



Key Services::Compute

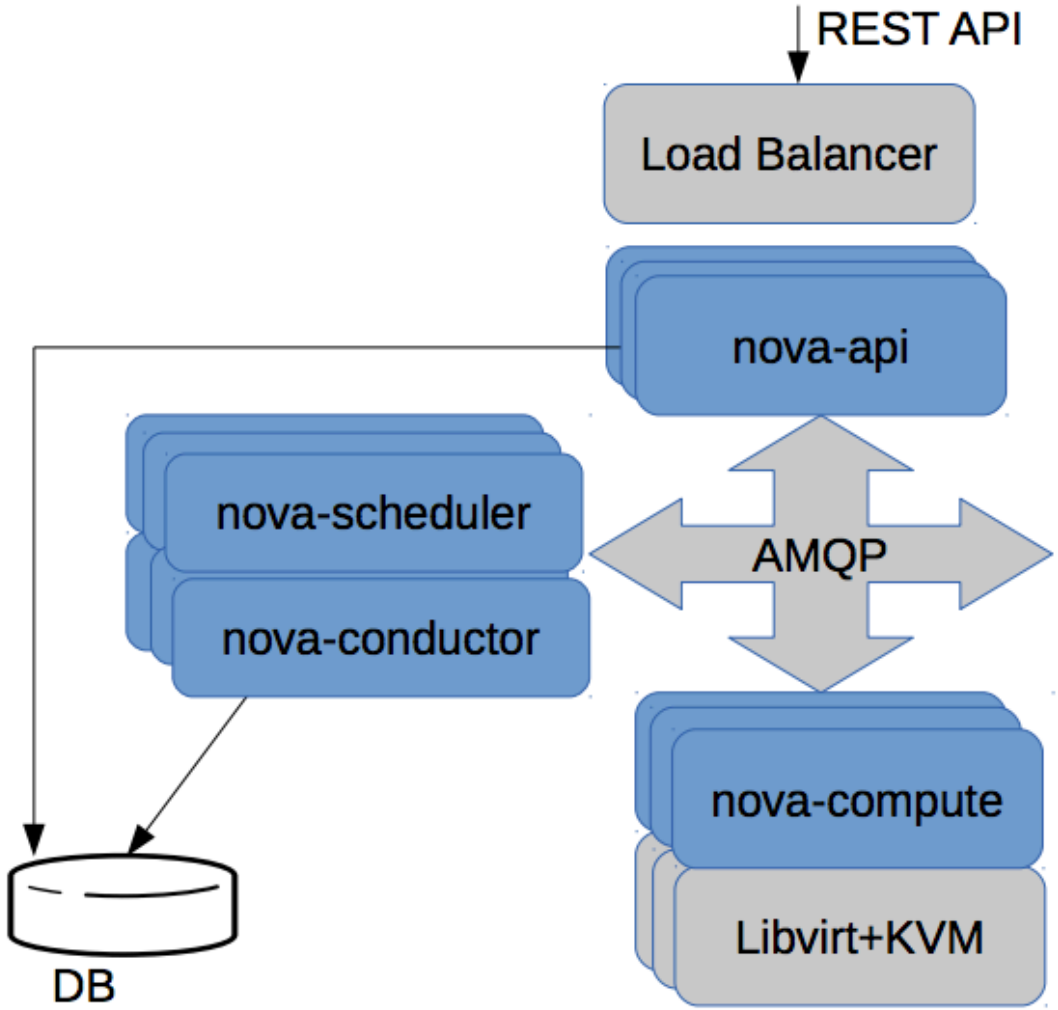
- Nova
  - Manages the lifecycle of compute instances in an OpenStack environment
  - Responsibilities include spawning, scheduling and decommissioning of virtual machines on demand
  - Virtualisation agnostic
    - Libvirt
      - open source API, daemon and tools for managing platform virtualisation 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)
    - …



Key Services::Compute

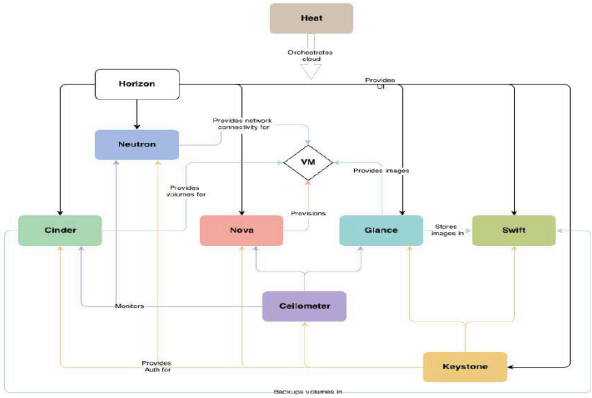
- Nova
  - API
    - Nova-api - accepts/responds to end user API calls; supports openStack Compute & EC2 & admin APIs
  - Compute Core
    - Nova-compute - Daemon that creates/terminates VMs through hypervisor APIs
    - Nova-scheduler - schedules VM instance requests from queue and determines which 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



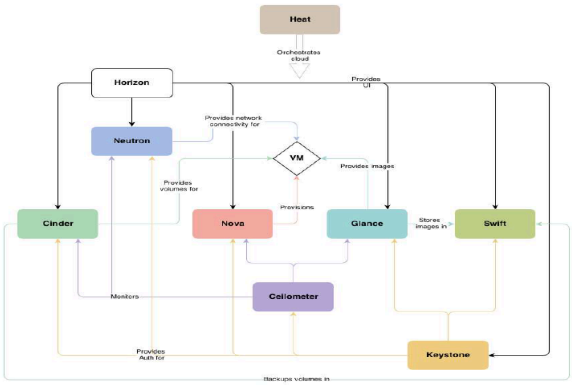
I need a VM with:

- 64Gb memory,
- 8vCPUs,
- in Melbourne,
- running Ubuntu 12.04,
- …



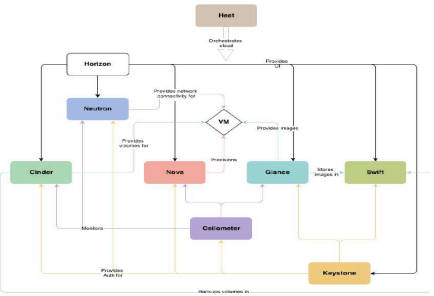
Key Services::Object Storage

- 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



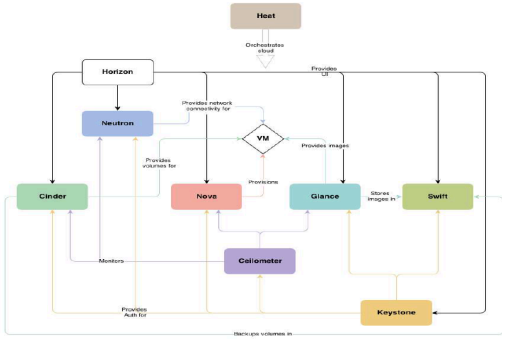
Key Services::Block Storage

- 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



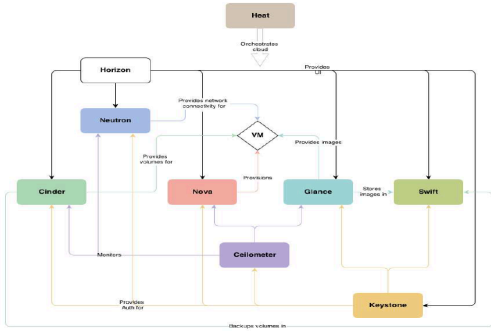
Key Services::Image Service

- 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 12.04…?
      - My last good snapshot…?



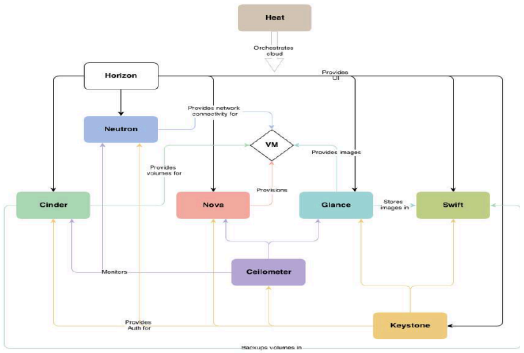
Key Services::Networking

- Neutron
  - Supports networking of OpenStack services (NaaS?)
  - 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, …



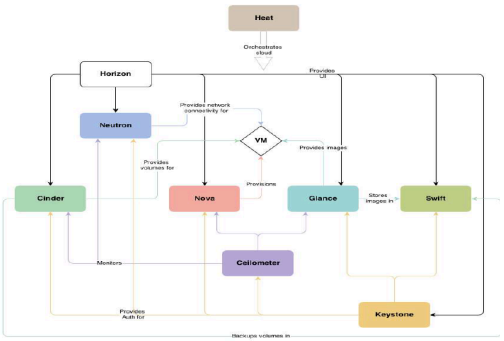
Key Services::Dashboard

- 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…



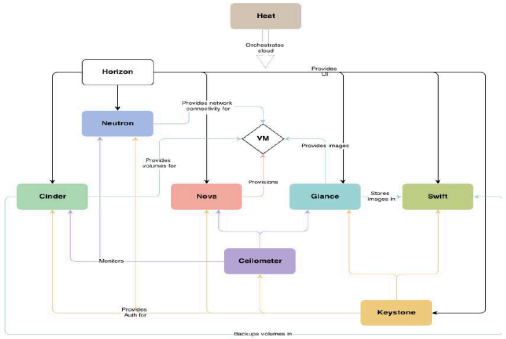
Key Services::Telemetry Service

- Ceilometer
  - Monitors and meters the OpenStack services for billing, benchmarking, scalability, and statistical purposes
    - Often referred to as telemetry information
  - Numerous services/APIs
    - Ceilometer-agent-compute – resource utilisation on each compute node
    - Ceilometer-agent-compute-central – aggregator and analyser capabilities
    - Ceilometer-agent-notification …
    - Ceilometer-collector …
    - Ceilometer-alarm-evaluator …
    - Ceilometer-alarm-notifier …



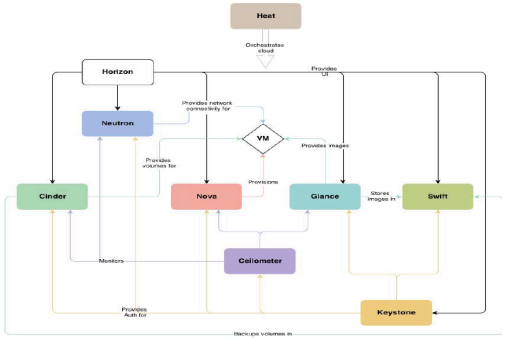
Key Services::Database Service

- 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



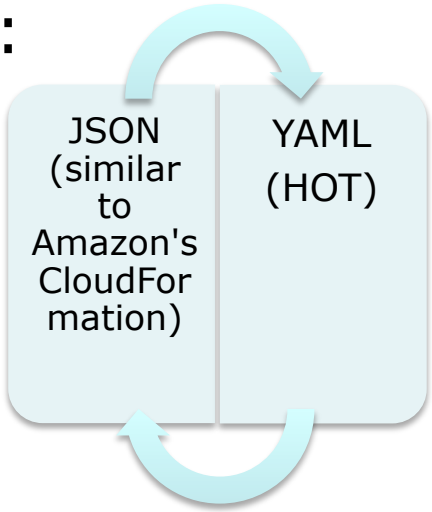
Key Services::Data Processing Service

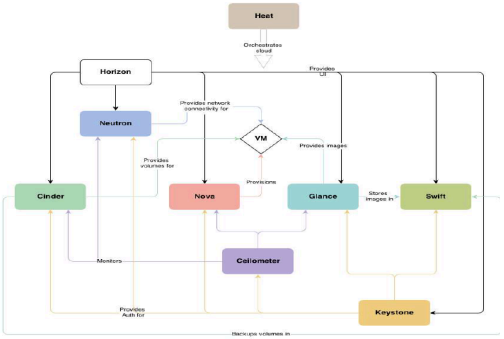
- Sahara
  - Provides capabilities to provision and scale Hadoop clusters in OpenStack by specifying parameters such as the Hadoop version, the cluster topology and the node hardware details
    - User fills in details and Sahara supports the automated deployment of infrastructure with support for addition/ 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 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 NeCTAR-style
- Creating a Wordpress website AWS-style

References

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