Lecture 7

OpenStack Nova

# OpenStack Internals

The primary sub-parts of the OpenStack Cloud software:

	Service	Comments
nova	Compute	Manages pools of computer resources
glance	Image	Disk and server images
keystone	Identity	Common authentication system
horizon	Dashboard	GUI
cinder	Block Storage	Persistent block storage devices
swift	Object Storage	Object and file management
neutron	Networking	IP addresses, VLANS, etc.
trove	Database	Provisions relational and non-relational DBs
heat	Orchestration	Launches composite cloud applications based on templates, specify relationships between resources
ironic	Bare metal provisioning	Provisions and turns on and off bare metal machines instead of virtual machines
ceilometer	telemetry	Billing system
sahara	Elastic Map Reduce	Handles Hadoop clusters
zaqar	Multiple Tenant Cloud Messaging	Can be used in SaaS to send messages between multiple components of the SaaS

OpenStack Cloud (cont'd): Nova
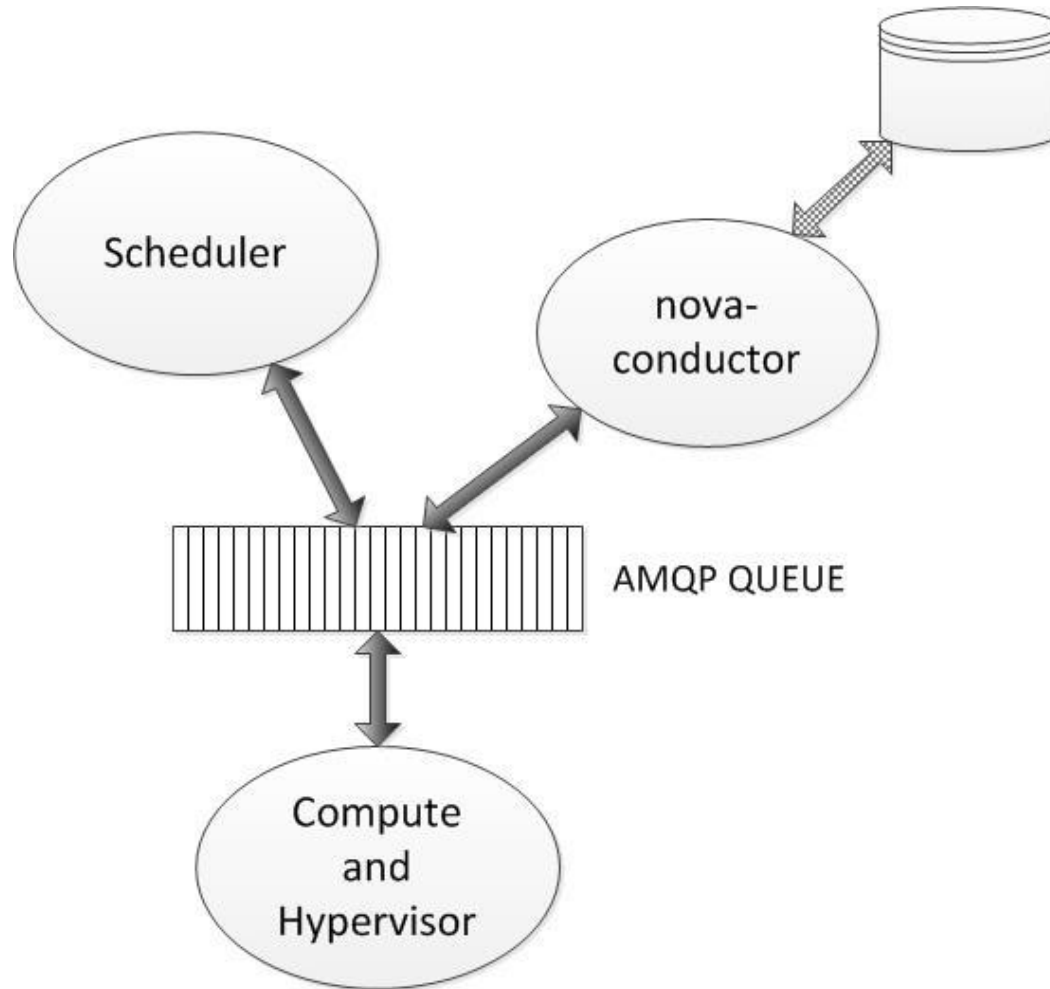
OpenStack Cloud (cont'd): Nova

Nova consists of several components, which can typically run on different servers:

- Message Queue—central hub for passing messages between components. Uses Advanced Message Queuing Protocol (AMQP). By default uses RabbitMQ, can use Apache Qpid

- Nova API: handles OpenStack RESTful API, EC2, and admin interfaces. Provides an endpoint, initiates running and instances, enforces some policies

- Authorization manager—Provides APIs for users, projects, roles, communicates with Keystone for details. Actual user store can be a database or LDAP back end.

- nova-objectstore—replicates S3 API to do image storage and retrieval, facilitates interaction with euca2ools. Can be useful to test features that will eventually run on S3. Normally replaced with OpenStack Glance.

OpenStack Cloud (cont'd): Nova

Nova components (cont'd):

- nova-network (deprecated in favor of Neutron): accepts network tasks from the queue, responsible for IP forwarding, VLANs, and bridges

- nova-database: SQL database (SQLAlchemy compatible) that stores build time and run time state. Includes instances in use, instance types available for use, projects, networks.

- nova-conductor: a stateless RPC server. Acts as a proxy to the database. For example, inside nova-compute, when making a VM update in the database, do an RPC to nova-conductor.

OpenStack Cloud (cont'd): Nova

Nova components (cont'd):

- nova-scheduler: allocates VMs to hosts. Takes a VM instance requests from the queue and determines which compute host it should run on. Allows plugins for different scheduling algorithms.

- nova-compute: handles interface between VMs and hypervisor. Creates and terminates VM instances— accepts actions from the queue then performs system commands to implement these actions while updating state in the database

OpenStack Cloud (cont'd): Nova

○ With Nova, an AMQP message broker handles communication between any two Nova components and messages are queued

○ This means that the Nova components are only loosely coupled, which is a good thing in this situation

● synchronous communication and the associated blocking of one component to wait on another component could be very bad from a performance standpoint

○ Nova implements both request/response and one-way communication between Nova components.

OpenStack Cloud (cont'd): Nova

- it is possible to configure Nova to work without nova-conductor, which means that each compute node would access the database directly

- However, without nova-conductor, if any nova-compute node is successfully attacked, then the attacker also has full database access.

  - However, even with nova-conductor running, an attacker with control of a compute node could still list all instances in the database and destroy them or update their state, see Mao (2013).

OpenStack Cloud (cont'd): Nova

- According to Mao (2013) using nova-conductor allows the database schema to be easily upgraded, and improves performance because by default a database call in Nova is blocking.

- Also Mao (2013) says that you can start as many instances of nova-conductor as needed.

OpenStack Cloud (cont'd): Nova

○ The nova-scheduler handles scheduling for virtual machine instances based on filtering and weighing

○ The filter scheduler (nova.scheduler.filter_scheduler.FilterScheduler) is the default scheduler for scheduling virtual machine instances.

- It supports filtering and weighting to make informed decisions on where a new instance should be created
- When it receives a resource request, it first applies filters to determine which host is eligible—the host is either accepted or rejected by a filter, there is no "maybe" result

OpenStack Cloud (cont'd): Nova

○ Example filters include (among others):

- AvailabilityZoneFilter—are in the requested availability zone
- ComputeFilter—passes all hosts that are running and enabled
- DiskFilter—is there sufficient disk space
- CoreFilter—are there sufficient CPU cores available
- RamFilter—does the host have sufficient RAM available

OpenStack Cloud (cont'd): Nova

- The filter scheduler weights each host in the list of available hosts

- By default, it uses the RAM Weigher to determine which host to place the instance on.

- Using the RAM Weigher, the hosts with large quantities of RAM will be selected first, until you exceed a maximum number of VMs per node.

- If you disable the RAM Weigher, then VMs will be randomly distributed among available hosts
  - note that each host already was checked by a filter to determine that it had sufficient RAM to run the VM

OpenStack Cloud (cont'd): Nova

○ By default, virtual machine instances are spread evenly across all available hosts, and hosts for new instances are chosen randomly from a set of the N best available hosts.

○ Various options can change how this works:

  ● for example, one option in the nova.conf (configuration) file allows virtual machine instances to be stacked on one host until that host's resources are used up

    • set ram_weight_multiplier in the config file to a negative number.

  ● Another option selects the value N used to select the N best available hosts

    • scheduler_host_subset_size

UAH Computer Science Department

OpenStack Cloud (cont'd): Nova

○ nova-network has been deprecated in favor of Neutron.

○ Remember that with networking in the cloud, we are concerned not only with physical networks between devices but also with virtual networks that connect virtual machines

OpenStack Cloud (cont'd): Nova

- Nova-network does not support as many network topologies as Neutron.
- The three network topologies supported by nova-network are:
  - Flat—all VMs share the same network (same subnet and bridge). The subnet and bridge are specified beforehand by the network administrator.
    - Each VM receives a fixed IP address from a pool of available addresses.
  - Flat DHCP—a DHCP server allocates IP addresses to VM instances on the specified subnet.
    - These IP addresses are not assigned to host network interfaces but only to VM interfaces.
  - VLAN—a VLAN and bridge are created for each project (tenant). Each project has a range of private IPs that can only be accessed from inside the VLAN.
    - Each project has its own VLAN, Linux networking bridge, and subnet.

UAH Computer Science Department