Introduction to OpenStack

Running a Cloud Computing Infrastructure with OpenStack

6th International Conference on Autonomous Infrastructure, Management and Security04 June 2012, University of Luxembourg

Anuj Sehgal
s.anuj@jacobs-university.de • http://cnds.jacobs-university.de/

Introduction to OpenStack

The OpenStack project was created with the goal of developing reliable, scalable and easily deployable cloud infrastructure software. This chapter provides information on cloud computing and OpenStack.

1. Cloud Computing

Cloud computing is a modern computing paradigm that delivers resources such as processing, storage, network and software are abstractions that are provided as services over the Internet in a remotely accessible fashion.

Users can access cloud applications using web-browsers, thin client machines or mobile devices, while all the data and software is stored on servers at a remote location, which are also used to perform all the heavy-duty processing. It is believed that cloud computing allows businesses and enterprises to function more efficiently since it offloads management and maintenance tasks, thereby making computing a service rather than a product. The sharing of resources, i.e. computing hardware, is also expected to reduce idle-time of machines and increase their productivity.

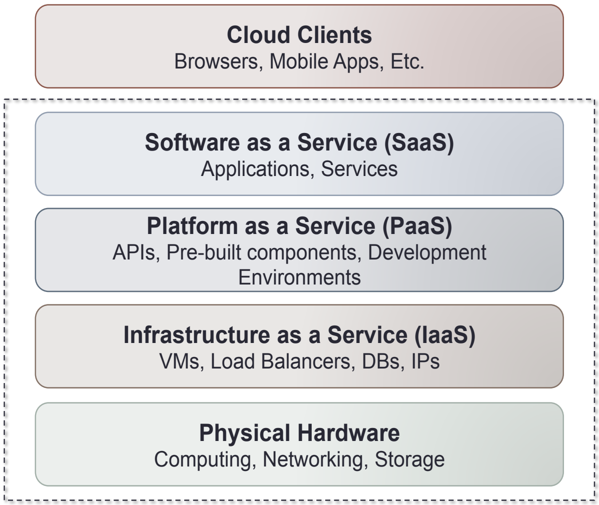
Cloud Clients
Browsers, Mobile Apps, Etc.
Software as a Service (SaaS)
Applications, Services
Platform as a Service (PaaS)
APIs, Pre-built components, Development Environments
Infrastructure as a Service (IaaS)
VMs, Load Balancers, DBs, IPs
Physical Hardware
Computing, Networking, Storage

Figure 1 - The cloud-computing layered model.

Due to the shift from computing being a resource to a utility, cloud computing introduces billing models that are based on time and utilities. On-demand availability, ease of provisioning, dynamic and virtually infinite scalability are some of the key attributes of cloud computing.

Services within the cloud are typically provided under the following three catagories:

a. Infrastructure as a Service (IaaS).
b. Platform as a Service (PaaS).
c. Software as a Service (Saas).

IaaS is the most basic cloud service model under which virtual machines, raw block storage, firewalls, load balancers and networking services are provided. In the PaaS model, a computing platform or solution stack including operating system, programming language execution environment, database, and web server are typically provided. Within the SaaS model, cloud providers install and operate application software in the cloud, which users access using cloud clients on computers, mobile devices, browsers and etc.

2. OpenStack

OpenStack is a collection of open source software projects that can be collectively utilized to operate a cloud network infrastructure in order to provide IaaS. The OpenStack project began as a collaboration of Rackspace Hosting and NASA as an open source project.

NASA was a user of the Eucalyptus open source cloud project before scalability issues in their Nebula project meant that they needed to develop their own technologies in this space. A contribution of their Cloud Files platform by Rackspace, combined with the Nebula computing software from NASA led to the initial birth of OpenStack. In the time since it's inception, the OpenStack consortium has managed to bring in over 100 members, including high profile industry names such as Citrix, Canonical and Dell.

Since Amazon's AWS was the first majorly used cloud service, OpenStack also makes its services available via Amazon EC2 and S3 compatible APIs. This ensures that all the existing tools that work with Amazon's cloud offerings, can work with deployments of OpenStack as well.

The OpenStack project is combination of three main components:

a. *OpenStack Compute (Nova)* – is used to orchestrate, manage and offer virtual machines upon many hypervisors, including QEMU and KVM. This is analogous to the Amazon Elastic Compute Cloud (EC2).
b. *OpenStack Object Store (Swift)* – provides redundant storage for static objects. This service is scalable to massive data sizes and theoretically can provide infinite storage. It is analogous to the Amazon Simple Storage Service (S3).
c. *OpenStack Image Service (Glance)* – provides storage for virtual disk, kernel and images. Glance is also used to provide image registration and querying services. It is able to accept images in many formats, including the popular Amazon Machine Image (AMI), Amazon Kernel Image (AKI) and Amazon Ramdisk Image (ARI).

Installing, configuring and working with Nova and Glance are covered as part of this tutorial. However, Swift is beyond the scope. Nonetheless, setting up Swift should not be a hard task for anyone who completes this tutorial.

It should also be noted that this tutorial is based upon the OpenStack Diablo release and uses an Ubuntu 11.10 Oneiric Ocelot server environment to setup all necessary packages. However, this is only because Ubuntu provides prebuilt packages for OpenStack Diablo. As such, the configuration of OpenStack can be performed on any flavor of Linux by using this tutorial as a basic guide.

2.1 OpenStack Compute (Nova)

Nova takes up the role of providing computing services within the OpenStack cloud. As such, any activity needed to support the life cycle of a virtual machine instance within the cloud is handled by

Nova. This includes things like managing block storage, networking, scheduling, computing resources, authorization and hypervisors.

However, Nova does not provide any virtualization capabilities by itself. It is designed to use `libvirt` APIs to interact with any supported hypervisors.

This means that Nova is hypervisor agnostic and provides support for Xen, XenServer/XCP, KVM, UML, VMware vSphere and Hyper-V amongst others.

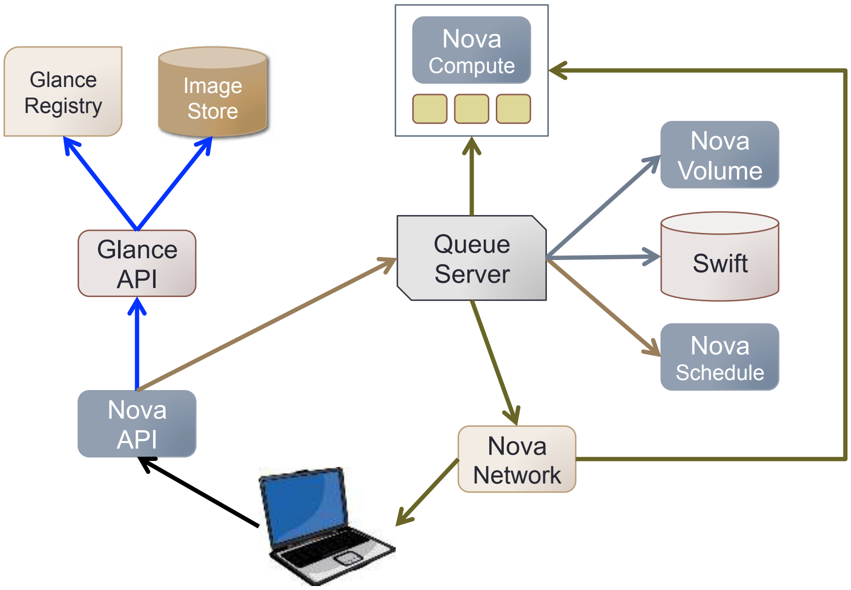

Figure 2 - An overview of the OpenStack architecture. The client interacts with Nova API services. This uses Glance API for image registration and retrieval. All other requests are sent to the queue server, which passes them off to the Compute, Volume, Schedule or Network nodes, as the need be.

All services provided by Nova are accessible via an API that is compatible with the AWS EC2. The main components of Nova are `nova-api`, `rabbitmq-server`, `nova-compute`, `nova-network`, `nova-volume` and `nova-scheduler`.

An overview of the OpenStack architecture can be seen in Figure 2. As can be seen, the client interacts with the Nova API server. In case requests dealing with registration or querying of images are sent, the API forwards these requests to the Glance API, which can perform queries within the Glance registry (stored in a SQL database).

However, if the request deals with managing an instance, then this is forwarded to the queue server, which in turn distributes the requests to appropriate components. Network address allocation, association and deallocation requests are handled by `nova-network`. On the other hand, block storage creation, deletion and association requests are handled by `nova-volume`. Similarly, virtual machine instance related queries are processed by `nova-compute`.

2.1.1 API Services (nova-api)

The `nova-api` service provides an interface to the outside world to interact with the cloud infrastructure. The API server is the only component that the outside world uses to manage the infrastructure. Management is done through RESTful calls using the EC2 API. The API server then, in turn, communicates with the relevant components of the cloud infrastructure by using the Message Queue.

2.1.2 Message Queuing (rabbitmq-server)

The OpenStack Cloud Controller communicates with other nova components such as the Scheduler, Network Controller, and Volume Controller by using AMQP (Advanced Message Queue Protocol). OpenStack uses the `rabbitmq-server` for this purpose. Nova uses asynchronous calls for request-response, with a call-back that gets triggered once a response is received. Since asynchronous communication is used, none of the user actions get stuck for long in a waiting state. This is especially true since many actions expected by the API calls such as launching an instance or uploading an image are time consuming.

2.1.3 Computing Services (nova-compute)

Servers providing computing services via `nova-compute` deal with instance management life cycle. They receive requests for life cycle management via the Message Queue and carry out appropriate operations. There are several servers providing computing services in a typical production cloud deployment. An instance is deployed on any of the available compute workers based on the scheduling algorithm used by Nova.

2.1.4 Network Services (nova-api)

The network services provided by `nova-api` deals with the network configuration of host machines. It does operations like allocating IP addresses, configuring VLANs for projects, implementing security groups and configuring networks for compute nodes.

2.1.5 Block Storage Services (nova-volume)

Block storage services performed by `nova-volume` include creation, deletion, attaching a volume to an instance, and detaching a volume from an instance. Volumes provide a way of providing persistent storage for use by instances, as the main disk attached to an instance is non-persistent and any changes made to it are lost when the volume is detached or the instance is terminated. When a volume is detached from an instance or when an instance, to which the volume is attached, is terminated, it retains the data that was stored on it when it was attached to an instance earlier. This data can be accessed by reattaching the volume to the same instance or by attaching it to another instances.

As such, any valuable data that gets accumulated during the life cycle of an instance should be written to a volume, so that it can be accessed later.

2.1.6 Scheduling Services (nova-scheduler)

The `nova-scheduler` maps API calls to the appropriate OpenStack components. It picks a server from a pool depending upon the scheduling algorithm in place. A scheduler can base its decisions on various factors such as load, memory, physical distance of the availability zone, CPU architecture, etc.