

# Capítulo 1

## Introducao ao Openstack

### 1.1 O que e Openstack

Em um primeiro momento poderíamos definir vagamente OpenStack como um software de cloud computing. Sendo um pouco mais específico: um Sistema Operacional projetado para nuvem capaz de controlar uma enorme quantidade de recursos computacionais (processamento, armazenamento e rede) em um Datacenter, ou seja, OpenStack é um projeto aberto para a criação de nuvens privadas e públicas, um sistema operacional para gestão e controle nuvem. Um cenário típico de um ambiente de cloud.

O termo projeto aberto neste contexto merece destaque, pois implica que o projeto nasceu com o objetivo de ser transparente e independente de fabricante. O site do projeto destaca os seguintes valores: **open source, open design, open development and an open community**.

Como “plataforma base” poderíamos empregar o software OpenStack, por exemplo, para construção de uma solução de Infraestrutura como Serviço (Infrastructure as a Service – IaaS).

Basicamente uma plataforma de IaaS fornece meios para criação/provisionamento de máquinas virtuais (ou nós computacionais) sob demanda. Permitindo ainda que sua infraestrutura de servidores possa expandir ou encolher de forma elástica de acordo com a necessidade da sua aplicação.

#### 1.1.1 Arquitetura

Como uma plataforma de computação em nuvem o OpenStack foi projetado para ser extremamente escalável e flexível. A plataforma é composta por vários “sub projetos” que juntos formam o seu núcleo (core). Sua arquitetura é modular formada por vários componentes que juntos implementam as funcionalidades três pilares que sustentam uma infraestrutura de nuvem: Processamento (compute), Rede (networking) e Armazenamento (storing).

Estruturas de processamento	
Nome	Funcao
nova	Gerencia o ciclo de vida das instâncias de computação em um ambiente OpenStack. As responsabilidades incluem a geração, programação desmantelamento de máquinas virtuais sob demanda.

Estruturas de networking	
Nome	Funcao
Neutron(nova-compute)	Permite Network-Conectividade-as-a-Service para outros serviços OpenStack, como OpenStack Compute. Fornece uma API para que os usuários definam as redes e os anexos neles. Tem uma arquitetura conectável que suporta muitos fornecedores de redes populares e tecnologias.

Estruturas de storage	
Nome	Funcao
Swift	Armazena e recupera objetos de dados não estruturados arbitrariamente através de um HTTP baseado API RESTful. É altamente tolerante a falhas com a sua replicação de dados e arquitetura scale-out. A sua implementação não é como um servidor de arquivos com diretórios montáveis. Neste caso, ele grava objetos e arquivos para várias unidades, garantindo que os dados são replicados em um cluster de servidor
Cinder	Fornece armazenamento em bloco persistente para instâncias em execução. Sua arquitetura condutora conectável facilita a criação e gestão de dispositivos de armazenamento de bloco.

Além dos componentes que formam o core do OpenStack a plataforma conta com um conjunto de serviços que integra cada componente para fornecer uma plataforma de IaaS completa. Essa integração é possível porque cada componente disponibiliza um conjunto APIs que permite o acesso às suas funcionalidades. Os serviços são os seguintes:

Estruturas de processamento	
Serviço	Funcao
Keystone	Fornece um serviço de autenticação e autorização para outros serviços OpenStack. Fornece um catálogo de pontos de extremidade para todos os serviços OpenStack.
Glance	Armazena e recupera imagens de disco de máquina virtual. OpenStack Compute faz uso deste durante o exemplo de provisionamento.
Ceilometer	Monitora e faz medidas do OpenStack nuvem para o faturamento, o benchmarking, escalabilidade e fins estatísticos.
Heat	Orquestra múltiplas aplicações em nuvem compostas usando o formato de modelo HOT nativo ou o formato de modelo AWS CloudFormation, tanto através de uma API REST OpenStack-nativo e uma consulta API CloudFormation-compatível.
Trove	Fornece dados-as-a-Service funcionalidade escalável e confiável nuvem para ambos os motores de banco de dados relacionais e não-relacionais.

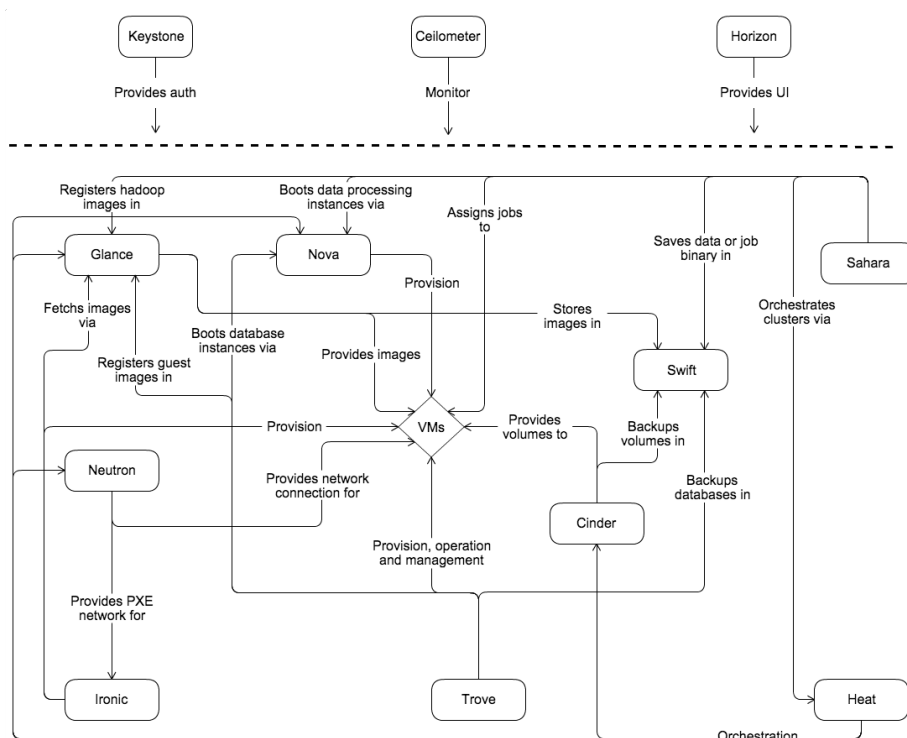


Figura 1.1: Arquitetura do OpenStack

### 1.1.2 Exemplo de Arquitetura

OpenStack é altamente configurável para atender às necessidades diferentes, com vários nós Compute, rede e opções de armazenamento. Neste tutorial será utilizado as seguintes configurações:

- O nó *Controller* executa o serviço de Identidade, serviço de imagem, parte do Compute gestão, e o painel de instrumentos. Ele também inclui suporte de serviços como um banco de dados SQL, fila de mensagens, e Network Time Protocol (NTP).
- O nó *Compute* executa parte da do Hypervisor do *Compute* que opera os "tenants" das máquinas virtuais, ou instancias. Por padrão, Compute usa KVM como o hypervisor. Nó *Compute* também dispõe "tenants" de rede fornece firewall (grupos de segurança) serviços. Pode-se executar mais de um nó. *Compute*

Opcionalmente, o nó Compute executa um agente de telemetria para recolher metros. Além disso, ele pode conter uma terceira interface de rede em uma rede de armazenamento separado para melhorar o desempenho dos serviços de armazenamento.

- O nó *Block Storage* opcional contém os discos que o serviço Block Storage providões para instâncias de máquinas virtuais inquilino. Pode-se executar mais de um desses nós;

- O nó *Object Storage* contém os discos que o serviço de armazenamento objeto usa para conta de armazenamento, recipiente e objeto. A arquitetura mínima são dois nós.

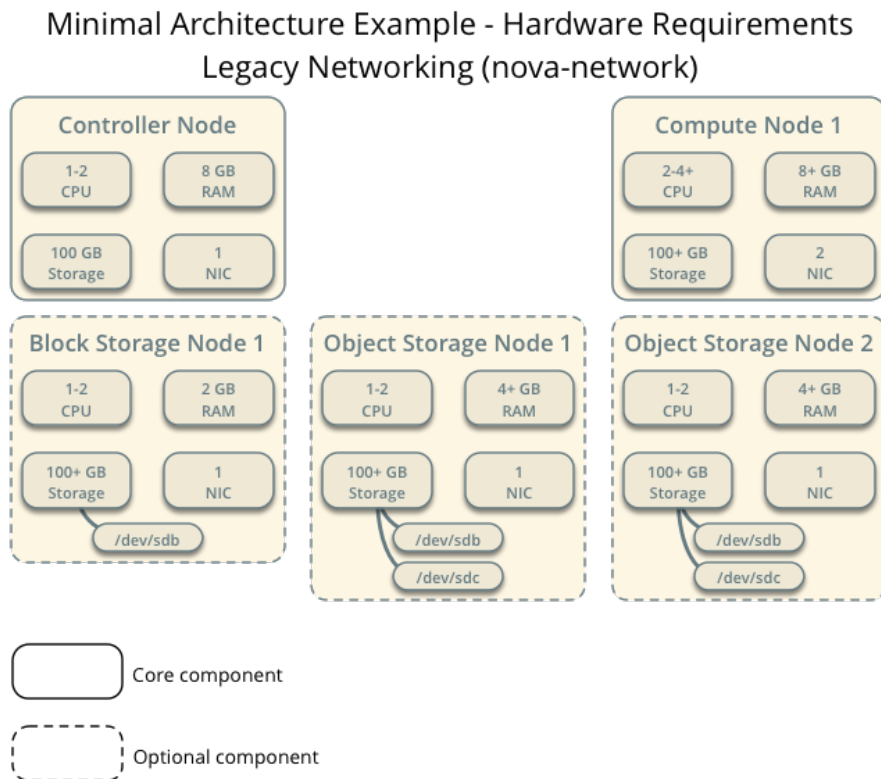


Figura 1.2: Arquitetura minima de hardware para suportar OpenStack

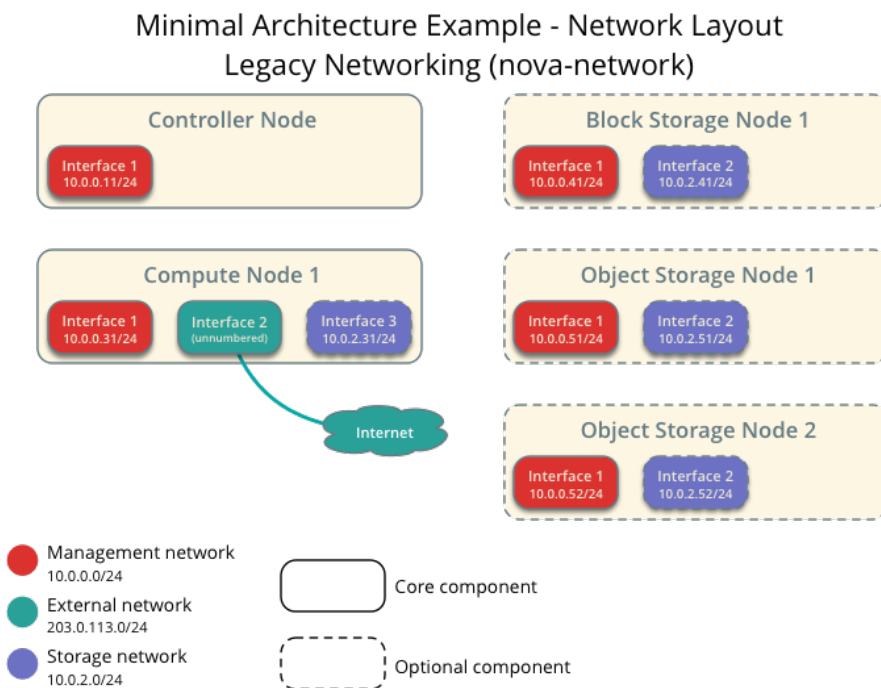


Figura 1.3: Arquitetura de rede para o OpenStack



## Capítulo 2

# Ambiente Basico

### 2.1 Passos iniciais de instalacao

Para melhor performance, é recomendado que o ambiente seja parecido ou exceda os requisitos de hardware na figura 1.2 do capítulo anterior. No entanto, OpenStack não requer uma quantidade significativa de recursos e os seguintes requisitos mínimos devem suportar um ambiente de "proof-of-concept" com os serviços centrais e várias instâncias CirrOS

- *nó Controller*: 1 processador, 2GB de memória e 10GB de armazenamento
- *nó Compute*: 1 processador, 2GB de memória e 20GB de armazenamento

Para poupar espaço em disco e fornecer mais recursos para OpenStack, recomendamos uma instalação mínima de sua distribuição Linux. Além disso, é altamente recomendável que você instale uma versão de sua distribuição de 64 bits em, pelo menos, o nó *Compute*. Se você instalar uma versão de sua distribuição no nó *Compute* de 32 bits, a tentativa de iniciar uma instância usando uma imagem de 64 bits falhará. Muitos usuários constroem seus ambientes de teste em máquinas virtuais (VMs). Os principais benefícios de VMs incluem o seguinte:

- Um servidor físico pode suportar vários nós, cada um com praticamente qualquer número de interfaces de rede.
- Capacidade de tomar "snap shots" periódicas durante todo o processo de instalação e "voltar" para uma configuração de trabalho, no caso de um problema.

```
$ openssl rand -hex 10
```

Para as senhas dos serviços do OpenStack, use *SERVICE\_PASS* e para as senhas de banco de dados *DB\_PASS*.

Nome da Senha	Descricao
<i>ADMIN_PASS</i>	Senha do usuario <i>admin</i>
<i>CEILOMETER_DBPASS</i>	
<i>CEILOMETER_PASS</i>	
<i>CINDER_DBPASS</i>	
<i>CINDER_PASS</i>	
<i>DEMO_PASS</i>	Senha do usuario <i>demo</i>
<i>GLANCE_DBPASS</i>	Senha do banco de dados para o servico de Imagem
<i>GLANCE_PASS</i>	Senha do usuario de servico de Imagem <i>glance</i>
<i>KEYSTONE_DBPASS</i>	Senha do banco de dados para o servico de autenticacao
<i>NOVA_DBPASS</i>	Senha do banco de dados para o servico <i>Compute</i>
<i>NOVA_PASS</i>	Senha do usuario de servico de <i>Compute</i>
<i>RABBIT_PASS</i>	Senha do usuario convidado do <i>RabbitMQ</i>

OpenStack e serviços de suporte exigem privilégios administrativos durante a instalação e operação. Em alguns casos, os serviços de realizar modificações para o host que pode interferir com ferramentas de automação de implantação, tais como Ansible, Chef e Puppet. Por exemplo, alguns serviços OpenStack adicionar um envoltório de "root" para sudo que podem interferir com as políticas de segurança.

Depois de instalar o sistema operacional em cada nó para a arquitetura que optar por utilizar, é necessário configurar as interfaces de rede. É recomendado que desativar todas as ferramentas de gerenciamento de rede automatizados e editar manualmente os arquivos de configuração apropriados para distribuição.

## 2.2 Distribuicao da Rede no modo Legacy (nova-network)

O exemplo da arquitetura com a rede no modo Legacy requer um nó controlador e pelo menos um nó *Compute*. O nó controlador contém uma interface de rede na rede de gestão. O nó *Compute* contém uma interface de rede na rede de gestão e um na *rede externa*.

A arquitetura em exemplo assume o uso da seguintes redes:

- Gerenciador de rede:
- Rede externa:

### 2.2.1 Controller Node

#### Configuracao da Rede

- Endereco de Ip:
- Mascara de Rede: 255.255.255.0 (ou /24)
- Gateway:

Reinicie o sistema para ativar as mudancas



## 2.2. DISTRIBUICAO DA REDE NO MODO LEGACY (NOVA-NETWORK)9

### Configurar nome da Rede

- Configurar o nome do computador (hostname) do nó para Controller
- Editar o arquivo /etc/hosts com as seguintes configuracoes

```
#controller

<endereço_de_ip>controller

#compute1#

<endereço_de_ip>compute1
```

**OBS:** Algumas distribuições adicionam uma entrada estranha no arquivo /etc/hosts que resolve o nome da máquina para outro endereço IP de auto-retorno, como 127.0.1.1. Você deve comentar ou remover esta entrada para evitar problemas de resolução de nomes.

### 2.2.2 Nó Compute

#### Configuracao da Rede

Configurar a primeira interface como interface de gerenciamento

- Endereço de Ip:
- Mascara de rede: 255.255.255.0 (ou /24)
- Gateway:

Configurar a segunda interface como externa utilizando uma configuracao especial sem um endereço de Ip assinalado.

Reescreva *NOME\_INTERFACE* com o nome da interface atual. Exemplo: eth1, ens224

- Edite o arquivo /etc/network/interfaces com as seguintes configuracoes:

```
#Ainterfaceexternadarede
auto NOME_INTERFACE
iface NOME_INTERFACE inet manual
    up ip link set dev $IFACE up
    down ip link set dev $IFACE down
```

Reinicie o sistema para ativar as mudancas

#### Configurar nome da rede

- Coloque o nome do computador (hostname) do nó como *compute1*
- Edite o arquivo /etc/hosts com as seguintes configuracoes

```
#controller

<endereço_de_ip>controller

#compute1#

<endereço_de_ip>compute1
```

## 2.3 Pacotes OpenStack

Distribuições liberam pacotes OpenStack como parte da distribuição ou usando outros métodos devido às diferentes datas de lançamento. Realizar esses procedimentos em todos os nós

**OBS:** Desativar ou remover qualquer serviço de atualização automática, pois pode modificar o ambiente OpenStack

### Permitir OpenStack Repositorio

- Instalar o arquivo "keyring" Ubuntu Cloud e repositório

```
# apt - getinstallubuntu - cloud - keyring
# echo "deb http://ubuntu-cloud.archive.canonical.com/ubuntu" "trusty-
updates/kilo main" >/etc/apt/sources.list.d/cloudarchive-kilo.list
```

### Finalizar Instalacao

- # apt-get update && apt-get dist-upgrade

**OBS:** Se a atualização inclui um novo kernel, reiniciar o sistema para ativar

## 2.4 Banco de dados SQL

A maioria dos serviços OpenStack usa um banco de dados SQL para armazenar informações. O banco de dados tipicamente executa no nó Controller. O procedimento nesta documentação usa o MariaDB ou o MySQL dependendo da distribuição. Os serviços do OpenStack também suportam outros bancos de dados SQL.

### Instalar e configurar o banco de dados do Servidor

1. Instalar pacotes

```
# apt-get install mariadb-server python-mysqldb
```

2. Escolhe uma senha para a conta de super-usuário

3. Criar e editar o arquivo `/etc/mysql/conf.d/mysql_d.openstack.cnf` com as seguintes configurações:

```
[mysqld]
...
bind-address = <endereço_de_ip>
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
```

### Finalizar Instalação

1. Reiniciar o banco de dados

```
service mysql restart
```

2. Guardar o banco de dados

```
# mysql_secure_installation ...

Set root password? [Y/N] ...

Remove anonymous users? [Y/N] y
... Sucess!

...

Disallow root login remotely? [Y/N] y
... Sucess!

Remove test database and access to it? [Y/N] Y
- Dropping test database ...
... Sucess!
- Removing privileges on test database ...
... Success!

...

Reload privilege tables now? [Y/N] Y
... Sucess!

...

Thanks for using MariaDB!
```

## 2.5 Fila de Mensagens (Message QUEUE)

OpenStack usa o *message queue* para coordenar operações e status da informação através dos serviços. O serviço *message queue* tipicamente é executado no nó Controller. OpenStack suporta vários serviços de fila de mensagem, incluindo RabbitMQ, Qpid e ZeroMQ. Contudo, a maioria das distribuições que o pacote OpenStack suporta um particular serviço de fila de mensagens. Esta documentação implementa o RabbitMQ serviço de *message queue*, pois a maioria das distribuições o suportam.

### 2.5.1 Instalar o pacote de fila de mensagens

- Instalar o pacote

```
# apt-get install rabbitmq-server
```

### 2.5.2 Configurar o serviço de fila de mensagens

1. Adicionar o usuário *openstack*

```
# rabbitmqctl add_user openstack RABBIT_PASS
Creating user "openstack" ...
... done.
```

2. Permitir configuração, escrita, e acesso a leitura for *user*

```
# rabbitmqctl set_permissions openstack ".*" ".*" ".*"
Setting permissions for user "openstack" vhost "/"
```