OpenStack Cloud Security

OpenStack is a system that controls large pools of computing, storage, and networking resources, allowing its users to provision resources through a user-friendly interface. OpenStack helps developers with features such as rolling upgrades, federated identity, and software reliability.

You will begin with basic security policies, such as MAC, MLS, and MCS, and explore the structure of OpenStack and virtual networks with Neutron. Next, you will configure secure communications on the OpenStack API with HTTP connections. You will also learn how to set OpenStack Keystone and OpenStack Horizon and gain a deeper understanding of the similarities/differences between OpenStack Cinder and OpenStack Swift.

By the end of this book, you will be able to tweak your hypervisor to make it safer and a smart choice based on your needs.

Who this book is written for
If you are an OpenStack administrator or developer, or wish to build solutions to protect your OpenStack environment, then this book is for you. Experience of Linux administration and familiarity with different OpenStack components is assumed.

What you will learn from this book
- Secure your servers, data center, and network to improve your environment for the cloud
- Gain insights into ISP intercept and social engineering
- Explore automated attacks with the help of mass phishing, brute force, and automated exploitation tools
- Secure your OpenStack installation from a networking perspective at both low and high levels
- Get to know how to secure your OpenStack to use only encrypted communications for APIs
- Configure secure communications on the OpenStack API
- Harden OpenStack Keystone and Horizon for a more secure environment
- Protect the Swift replication mechanism through network hardening

Build a secure OpenStack cloud to withstand all common attacks

Fabio Alessandro Locati

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 2 'OpenStack Security Challenges'
- A synopsis of the book’s content
- More information on OpenStack Cloud Security
**About the Author**

**Fabio Alessandro Locati** is an Italian IT external consultant. His main areas of expertise are Linux, networking, security, data centers, and OpenStack. With more than 10 years of working experience in this field, he has experienced different IT roles, technologies, and languages. Fabio has worked for many different companies, starting from a one-man company to huge companies such as Tech Data and Samsung. This has allowed him to consider various technologies from different points of view, helping him develop critical thinking and understand whether a particular technology is the correct one in a very short span of time.

Since he is always looking for better technologies, he also tries new technologies to see their advantages over the old ones. Two of the most important things Fabio evaluates in a technology are its internal security and the possibility of additional security through configuration or interaction with the other technologies. For virtualization, he often uses OpenStack due to its power and simplicity, ever since he first tried it in 2011. Fabio has used OpenStack for the public-facing cloud, as well as the internal clouds.
Preface

As our society moves from an analog world to a digital world, it is easier for ill-intentioned people to attack privates, companies, banks, and government for their advantage or for the other party's damage. Since the beginning of information technologies, we have seen a shift to digitalize our world, and this process has been accelerating ever since. The virtualization has concentrated more data on even less systems, making these systems very nice targets for attacks. Making the clouds secure will be one of the biggest security challenges for the next 10 years, from my point of view. The goal of this book is to prepare cloud administrators for this challenge.

The structure of this book is designed to give you a wide perspective on security. This has multiple reasons. First of all, programs change, but a secure mindset is often more important than knowing how to secure a very specific software, also because very often people specialize in a particular part of the IT sector and kind of lose track of what there is around the technology they master. This often leads to huge security problems in between the areas of expertise of the various people of the team.

OpenStack allows very powerful infrastructures, but tends to be pretty complex, being a solution to many different situations, making it, often, very interesting from a business point of view, but very hard to manage safely.

What this book covers

Chapter 1, First Things First – Creating a Safe Environment, teaches you about a lot of basic security concepts. Also, you'll see a lot of things to be kept in mind while designing a data center as well as new security policies.

Chapter 2, OpenStack Security Challenges, allows you to discover the different kinds of clouds and how this affects security and also the possible types of attacks. In the second part of the chapter, you'll see the various parts of OpenStack and what they do.
Preface

Chapter 3, Securing OpenStack Networking, shows you how the OSI networking model works from a security perspective and a lot of possible network attacks for each OSI level. In the second part of the chapter, you can see how to harden OpenStack and a few utilities OpenStack networking can provide to make your workflow more secure.

Chapter 4, Securing OpenStack Communications and Its API, explains how the encryption works in our world, and, therefore, what its strengths and weaknesses are. You’ll also learn how to enable encryption for the OpenStack APIs.

Chapter 5, Securing the OpenStack Identification and Authentication System and Its Dashboard, shows you how the identification, authentication, and authorization systems work, as well as how OpenStack can be configured to meet your needs from this point of view.

Chapter 6, Securing OpenStack Storage, explains how the different kinds of storage work from a security standpoint and the options you have to implement them in OpenStack. Also, you will see some configuration to make the storage more resilient to attacks.

Chapter 7, Securing the Hypervisor, lists all the hypervisors that can be used with OpenStack. You'll find a lot of insight on how to choose the right hypervisor for you and how to secure it.
OpenStack Security Challenges

As we have seen in the first chapter, each level of your infrastructure can be an object of the unwanted attention for an attacker. Software is no exception to this. There are a lot of attacks that aim to find bugs or misconfigurations in software and exploit them to gain access to the machines that run the software, or to data. OpenStack, with all its parts and all the software it relies on, can be a very effective attack vehicle if not safely configured, due to its very trustful policy that allows nodes to access all data if the node requires it. So, an attacker can quickly compromise and obtain your data if he or she is able to compromise a single node.

Before looking at OpenStack directly, I would like to deal with a critical aspect: security in cloud environments; that is, the ownership of machines.

In this chapter, we will to cover:

- The differences between the private and the public cloud with a focus on the security aspects
- The possible security threats to a cloud components of OpenStack

Private cloud versus public cloud security

Very often people say cloud when they actually mean public cloud. For this reason, in the book, we’ll always specify private cloud or public cloud and when we do not specify anything, the word cloud is used in both senses at the same time.
This is a necessary disclaimer because when speaking of security, private and public clouds have completely different issues, but let's start from the beginning.

**The private cloud**

A private cloud environment is operated solely for a single organization (or person) by internal or third-party personnel. In a private cloud situation, all machines are owned (or leased) by the organization and will run that organization’s software exclusively.

From an economical perspective, private clouds are less flexible; in fact, the number of machines will stay pretty stable over time compared to public clouds.

From a scalability perspective, private clouds are not very flexible because you can’t use more processing power than that you have installed it with. Very often, private clouds are kept with an average of 80-90 percent load and this means you can burst only 10-25 percent of your average load.

From a security perspective, private clouds grant you full access (and full responsibilities) to create a safe environment. This means that no one can look at your data if you create a safe environment, and you will have to spend money to create a safe environment. Usually, these clouds are created behind a company’s firewall, so this helps secure them. This security advantage is negated if the cloud contains the Web-readable/writable content because you’ll have to open your firewall ports in this case. This is often mitigated by creating two different clouds, one for web-accessible data (in a DMZ) and one that is accessible only by internal users (in the internal network).

**The public cloud**

“There is no [public] cloud, only other people's computers.” – Free Software Foundation Europe

A public cloud has very different problems and opportunities as compared to a private cloud.

From an economical perspective, with a public cloud, you pay exactly what you use as you go, so no upfront costs.

From a scalability perspective, public clouds can be considered as limitless because they usually have so many resources available that you can start up all the machines you need without worrying about cloud capabilities.
From a security perspective, the public cloud is more complex to analyze. Since cloud providers usually provide to millions of machines at any given moment, they can invest way more than the average company for security. Thus, their cloud is very secure. The drawback is that you have to trust the **Cloud Service Provider (CSP)** completely with your data. If the CSP would like to see your data and everything you run on their machines, they can. If they are interested in selling your data to your competitor, there are very limited things you can do. Also, we have to remember that public clouds can be attacked from inside, since an attacker can lease a virtual machine directly into the cloud for a few dollars and without any questions asked.

Since all users of a public cloud are not in the company network of the cloud service provider, public clouds have to be accessible from the Web, increasing the attack surface of public clouds.

### Private cloud versus public cloud

The following is an easy-to-remember schema that will help you immediately understand the advantages and disadvantages of public and private clouds:

<table>
<thead>
<tr>
<th>Prospective</th>
<th>Public cloud</th>
<th>Private cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economical</td>
<td>Pay as you go</td>
<td>Pay upfront</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Usually very high</td>
<td>Limited</td>
</tr>
<tr>
<td>Scalability</td>
<td>Virtually unlimited</td>
<td>Limited</td>
</tr>
<tr>
<td>Security</td>
<td>Usually very high</td>
<td>Limited to your budget</td>
</tr>
<tr>
<td>Data confidentiality</td>
<td>Not under your control</td>
<td>Under your control</td>
</tr>
<tr>
<td>People you have to trust</td>
<td>Yourself and the cloud provider</td>
<td>Yourself</td>
</tr>
</tbody>
</table>

As we can see, public clouds and private clouds are very different and there isn't a choice that is always right and one that is always wrong. It depends on the specific software you have to deploy. If you integrate a private cloud with a public cloud, you'll have an **hybrid cloud**. Usually, the public part of a hybrid cloud has the same characteristics as that of a public cloud, as the private part has the same characteristics of a private cloud.
The different kinds of security threats
As we have seen in the previous chapter, when we speak about security, we can mean multiple things. Also, as we have just seen that private and public clouds present different kinds of security issues. We are now going to analyze the various attacks that you can encounter when administering an OpenStack cloud.

Possible attackers
Let's start by identifying the possible attackers we can face. They can be divided in different ways based on their goals; in this case, we will distinguish them as the following:

- **Automated attacks/Script kiddies**: Automated vulnerability scanning/exploitation.
- **Motivated individuals**: This includes multiple kinds of attackers, such as small-scale industrial espionage, rogue or malicious employees, or disaffected customers. They act alone.
- **Highly capable groups**: These groups often refer to themselves as Hacktivist and are not typically commercially funded, but can pose a serious threat to service providers and cloud operators. Many groups of hackers have organized themselves lately, such as Lulzsec and Anonymous.
- **Organized hackers**: These are groups of hackers who are usually highly capable. These groups are financially driven and able to fund in-house to exploit development and target research. Multiple groups fall in this category, from the Russian Business Network to the various organized groups that undertake industrial espionage.
- **Intelligence agencies/services**: They usually have capabilities greater than any other attacker, because they can bend rules without breaking them and can be authorized to violate rules. Intelligence agencies and other governmental players are comparable to organized hackers, but usually have far more money they can spend on those operations, making them more effective.

The possible attacks
There are multiple kinds of attacks that can be put into action. The main kinds are as follows:

- Denial of Service
- 0-day
• Brute force
• Advanced Persistent Threat
• Automated exploitation tools
• ISP intercept
• Supply chain attack
• Social engineering
• Hypervisor Breakout

**Denial of Service**

A **Denial of Service (DoS)** attack is an attack that aims to make some service unavailable. In the last few years, usually we speak about DDoS, since those are very effective and cheap and for those reasons have become the most used DoS attack. DDoS attacks consist in multiple machines trying to overload a server or its connection to make the services that are running on that server unavailable.

The good part about DoS attacks is that in majority of the cases, as soon as they end it, all is back to normal. When this is not true, small actions have to be executed by system administrators, such as restarting a service or rebooting a machine.

```
There is no way to completely protect a server from a DoS attack.
```

Even if you cannot protect your company completely from such attacks, you can mitigate them in two ways:

• Having a lot of spare resources such as CPU, RAM, and bandwidth makes harder to knockdown the service
• Writing rules on firewalls (or having an **Intrusion Prevention System (IPS)** or an **DoS Defense System (DDS)** that do it for you) that drop all traffic coming from IPs that are currently attacking your servers

There are companies that provide **clean pipes** that are connected with only good traffic since have already been filtered by the ISP using IPSes and DDSes.

Usually, DDoSes are used by automated attacks/script kiddies, motivated individuals, and highly capable groups. It could so happen that organized hackers too use DDoS attacks, but, in this case, it's usually an **Advanced Persistent DoS (APDoS)**, where the attack lasts for long periods (the longest APDoS registered was 38 days), moves from server to server to be harder to detect, and involves a huge amount of traffic (usually more than 50Pb in total).
OpenStack Security Challenges

In the history of DoS attacks, the following methods have been heavily used:

- **Buffer overflow attacks**: In this kind of attack, the attacker looks for buffers that are filled with input from the user without prior validation. Since buffers have a fixed length, we can't put only a certain amount of data that can fit in the buffer; the rest of the data will be written in other parts of RAM and could be executed by the program.

- **SYN Flood attacks**: As we will see more deeply in the next chapter, the computers expect certain handshakes at the beginning of a communication. This attack violates this convention forcing the server to open more connections than needed. At a certain point, the server will not be able to open a new connection, making the service unavailable.

- **Teardrop attacks**: Network packages should be of certain sizes. If bigger packages are found, the machines split them into smaller packages to manage them properly. Old machines have problems recognizing and properly managing packages that are smaller than expected. In this attack, this bug is exploited by sending smaller packages than expected to the machines, which in old systems often resulted in system crashes and reboots.

- **Smurf attacks**: In this kind of attack, the attacker uses badly configured machines in the network to amplify the attack. Usually, the attacker sends a forged package (that is, ICMP ECHO package) that seems to arrive from the victim to a broadcast address. All the machines in the broadcast domain that are tricked by this package, will respond to the victim. So, if in the network there are 100 machines with poor configuration, an attacker could be able to create an amount of traffic that is 100 times its maximum amount of traffic.

- **Viruses/Worms**: In this kind of attack, the attacker creates a self-replicating program that can consume resources or destroy the systems.

**0-day**

A 0-day attack is an attack that exploits a vulnerability that was not known (or thought not to be exploitable) until that day. In these cases, there is no patch available when the attack is used the first time.

In a 0-day case, no specific measurement can protect a company, but all general security measurements we already talked about in the previous chapter will help mitigate this risk.

0-day attacks can only be done by highly capable groups, organized hackers, and intelligence agencies/services because those are the only players that have the resources needed to do such an attack.
Brute force
Since many attackers cannot afford to invest to research 0-day attacks, they use Brute force. Brute force is very noisy and the majority of system administrators, and IDSes will recognize and block them.

To prevent these kinds of attacks, you should have an IDS and good policies for passwords.

These attacks are so noisy that only automated attacks/script kiddies and motivated individuals will use them.

Advanced Persistent Threat
An Advanced Persistent Threat (APT) is a kind of attack in which expert attackers use stealthy and continuous attacks targeting a specific entity.

During an APT attack, it is common to find many attacks that we already discussed, such as APDoS, and 0-day exploitations. Often also involved are Social Engineering techniques and Supply Chain attacks, which we will talk about shortly.

Since these are very expensive attacks that require multiple people, they can only be done by highly capable groups, organized hackers, and intelligence agencies/services.

Automated exploitation tools
Since APT attacks are very expensive, automated attacks/script kiddies, and motivated individuals will prefer Automated exploitation tools. These tools allow the attacker to test multiple already known exploits to search for a known exploit that the system administrator has not yet patched.

Examples of Automated exploitation tools are Metasploit and Nessus.

To prevent these kinds of attacks, you need to always keep your system updated and frequently check online whether there are new ways to exploit software applications you use become known.
The ISP intercept

The ISP intercept is a category in which a lot of possible attack vectors fall into. The baseline is that somehow the attacker is able to see all traffic moving into your connection at the border of your property. This attack can be executed legally by Intelligence agencies/services with a warrant, or could be executed illegally.

I've seen an example of an illegal execution of this kind of attack in a company, where the attacker has cut the company's Internet connection cable and has added its own box that allowed the traffic to be normally received and sent, but also copied all the passing data to the attacker systems.

Preventing those kinds of attacks is impossible because they are executed outside your competence limits. The only possible way to mitigate these attacks is to encrypt all the data you share with the outside world.

Sometimes even companies' private networks can be compromised with this attack. Recently, there have been rumors that the NSA was able to retrieve data from Google and Yahoo! by tapping their fiber optics cables that connect the datacenters. Even if there hasn't been any official confirmation, in a few months' since these rumors, both Google and Yahoo! announced that they now encrypt all traffic between datacenters to prevent this from happening.

The only kind of attacker that can do this legally are the Intelligence agencies/services. But organized hackers can also perpetrate such attacks.

The supply chain attack

In a supply chain attack, the attacker tampers a cryptographic component, such as a device that performs encryption or secure transactions, when it is still in the supply chain of the device, so that it is not yet in the hands of the client. This could happen during the manufacture of the device or at a certain point before it is put into the production environment. For this kind of attack, the attacker needs physical access to the device. A common type of tempering is the installation of a rootkit or specific hardware design to spy on the user.

From the documents written by Edward Snowden, it seems that the NSA has been able to perform multiple Supply Chain Attacks in the last few years. This has not been confirmed as of today by the NSA itself.

Due to its complexity, only Organized hackers and Intelligence agencies/services can perform this kind of attack.
Social engineering
As we have already seen in the previous chapter, social engineering could be a good option for an attacker who would like to attack an organization.

All kinds of attackers can perform social engineering attacks, but the most effective will be the ones perpetrated by the most skilled groups.

The Hypervisor breakout
Since we are focusing on OpenStack, the Hypervisor breakout is an attack your company could suffer from.

In an Hypervisor breakout, the user of a virtual machine is able to escape from his virtual machine and connect to the host that is running the virtual machine. In the history of virtualization, there have been multiple cases of possible Hypervisor Breakout attacks and pretty much all hypervisors have been objects of such unwanted attention.

At the moment, there is no known case of Hypervisor Breakout in real-world attacks, but it is possible that some companies have been compromised by this kind of feature but has not made it public due to the possible consequences to the company's image.

The real risk with an attack of this kind is that a person with such a level of access will probably be able to attack every machine in the cluster and will be able to access all resources available to those machines.

We will see how to prevent this kind of attack in the last chapter of this book.

These attacks are really hard to perform and are very expensive, so only organized hackers and intelligence agencies/services will be able to perform them.

The OpenStack structure
OpenStack is an orchestration suite to create clouds mainly focused to create Infrastructure as a Service (IaaS) solutions. OpenStack has multiple components, each one aiming to provide a piece to the cloud. As I write, last OpenStack stable version is Juno that has the following components:
OpenStack Compute Service – Nova

Computing is one of the core parts of any IaaS solution, as well as OpenStack. This is also one of the two oldest modules of OpenStack, since it has been part of the project since its first version, Austin, which was released in October, 2010. Nova derives from NASA's Nebula platform.

Nova is a cloud computing fabric controller. It is designed to manage and automate pools of computer resources and can work with many hypervisors such as KVM, VMware, and Xen.

It is written in Python and uses many external libraries. Nova was created with horizontal scalability in mind; in fact, it's able to scale horizontally on commercial off-the-shelf (COTS) components. This allows you to keep the hardware costs down and to easily integrate with legacy hardware.

Starting from the Havana release, Nova is able to run docker containers directly, but due to some Continuous-Integration problems, this feature will be in the main source code only since the Kilo release.

Nova can be compared to Amazon's Amazon Elastic Compute Cloud (EC2). As for the Docker addition in Kilo, Amazon provides the AWS Elastic Beanstalk service.

OpenStack Object Storage Service – Swift

The other component available since the first release of OpenStack is Swift, a scalable redundant storage system. Swift was developed in the first place by Rackspace Hosting itself and derives from the Rackspace expertise, and is built for creating and managing the Rackspace Hosting Cloud File service. Currently, Swiftstack is leading the development of Swift.

Swift is an object storage capable to ensure data integrity, thanks to its ability to write the files to multiple disks spread throughout the nodes in the cluster. Swift is also able to manage multiple regions for the same pool, so it's possible to create real-time, off-site replicas of data to prevent possible data losses in case of problems in the main region.

Due to its design, Swift—like Nova—is created with horizontal scalability in mind, and works with COTS components.

Swift can be compared to Amazon's Amazon Simple Storage Service (S3).
OpenStack Image Service – Glance

Glance has been added in the second release of OpenStack (Bexar), and since its first version, it has greatly improved. Glance is useful to save disk and server images to make the users able to run multiple equal servers without having to reconfigure them each time.

The purpose of Glance is to help you manage the Nova images in a simpler and more efficient way. In fact, Glance allows you to use the images as templates for new instances, take snapshots, and backups.

Glance is not a storage service for those images and can rely on multiple storage services, such as the OpenStack Object Storage Service. Due to this fact, Glance can be easily integrated with the current storage architecture and can contain a large number of images, based on the amount of free space available in your backend storage.

Glance provides a REST API interface to integrate with other components to allow other components to manage (indirectly) machines, images, and templates.

Glance can be compared to Amazon's Amazon Machine Image (AMI) system.

OpenStack Dashboard – Horizon

Horizon is the OpenStack dashboard and can help users to handle OpenStack resources without the need for command-line access. Horizon has been added in Essex, the fifth release of OpenStack.

Horizon is a web interface for OpenStack and all components of OpenStack can be managed in Horizon. This allows OpenStack end users to access their account and to manage their OpenStack resources without the need of a system administrator and of connecting via terminal to the cluster. This improves OpenStack security.

Horizon is designed to allow easy integration with other products and services, in order to allow an easy deployment and usage with third-party software.

Horizon can be compared to Amazon's AWS Management Console.
OpenStack Security Challenges

OpenStack Identity Service – Keystone

Keystone is the identity server of OpenStack. It has been added to OpenStack in Essex.

Keystone is a service that catalogs the available API endpoints and allows a centralization of user permissions in OpenStack. Due to the high sensibility of these information, it will be very costly and unsafe to let each component manage them. To do so, Keystone keeps all information in a secure way and all the other components that need them will be able to access it using the Keystone REST API. Keystone allows multiple authentication methods such as username and password, token-based system and Amazon Web Services (AWS) login.

Keystone supports multiple backends to store this data, such as LDAP.

Keystone can be compared to Amazon's AWS Identity and Access Management (IAM).

OpenStack Networking Service – Neutron

In Folsom (the sixth release of OpenStack), a networking module called Quantum has been added. Due to some branding issues, since Havana (the eighth OpenStack release), this module has been renamed as Neutron.

Neutron allows you to create and manage virtual networking in an easy yet powerful way. It allows to have global networks that are valid for all users and managed by administrators and user networks that are usable and manageable by a single user. In the case of user networks, the network will be visible and usable only by that specific user.

Neutron does not only provide basic networking, but also provides advanced networking tools, such as floating IPs. Also, it provides an extension framework allowing the deployment and management of other network services such as Intrusion Detection Systems (IDS), load balancers, firewalls, and virtual private networks (VPN). For administrators, there is the possibility to use software-defined networking (SDN) technology such as OpenFlow to support multitenancy and horizontal scaling.

Neutron can be compared to Amazon's Amazon Virtual Private Cloud (VPC).
OpenStack Block Storage Service – Cinder

Cinder is a Block Storage for OpenStack. It has been included in OpenStack since Folsom (the sixth release of OpenStack).

Cinder is able to provide block-level storage devices to Nova. Cinder interface and its features are comparable to the block storage providers available in commercial SAN products, so any user is able to create, manage, and use their block storage devices. Cinder does support multiple backends, such as Ceph, GlusterFS, NFS, and multiple proprietary SAN systems.

Cinder can be compared to Amazon's Amazon Elastic Block Store (EBS).

OpenStack Orchestration – Heat

Heat has been a part of OpenStack since Havana (the eighth release of OpenStack). It can be used to orchestrate cloud applications using templates, and to automatically create machines on demand.

Heat can be used to create machines on demand from templates to allow an application to grow horizontally without any need for direct commands from the administrators.

To help the administrators that have to manage multiple infrastructure on OpenStack and Amazon, or are migrating the infrastructure from Amazon to OpenStack, Heat does support Amazon CloudFormation template syntax.

Heat can be compared to Amazon's Amazon CloudFormation.

OpenStack Telemetry – Ceilometer

Ceilometer has been added to OpenStack in Havana (the eighth release of OpenStack) with Heat, since they are complementary. In fact, Ceilometer provides data about the user's usage of resources, so as to be able to bill the people based on the actual resources used.

Ceilometer provides a single service that centralizes each service counter, so it's possible to export the usage data that will be needed to calculate the customer billing. All data available in Ceilometer are traceable and the whole process can be audited. Ceilometer data can also help companies using OpenStack in their private cloud to understand which processes and Business Units use more resources.

Ceilometer can be compared to Amazon's Amazon CloudWatch.
OpenStack Security Challenges

OpenStack Database Service – Trove

Trove is a database-as-a-service that is able to provide databases that are both relational and nonrelational. It has been added in Icehouse (the ninth release of OpenStack) and has been heavily improved in Juno (the tenth release of OpenStack).

Trove manages the database for the user, so it's capable of migrating a database from a machine to another or to scale the machine size based on the required resources. It also provides a RESTful API to communicate to the databases to completely abstract the database and its management. Also, the native interface of the chosen database is always available. Currently, it supports relational databases such as MySQL, NoSQL databases such as MongoDB, Cassandra, Redis, CouchDB, CouchBase, and in-memory databases such as MemCached and VoltDB.

Trove can be compared to Amazon's Amazon Relational Database Service (RDS), but Amazon's service only supports relational databases.

OpenStack Data Processing Service – Sahara

Sahara is a Hadoop-as-a-service system. It's very new; in fact, it has been added in Juno (the tenth release).

Sahara allows the user to create Hadoop clusters quickly and easily. It also allows the user to be fully in control of the clusters, being able to set a lot of settings such as Hadoop version, cluster topology, and node's hardware details. After the user completes this information, Sahara deploys the cluster in a few minutes.

Sahara also allows the user to launch and manage MapReduce jobs on the clusters that have created.

Sahara can be compared to Amazon's Amazon Elastic MapReduce (EMR).

Future components

Since the OpenStack community is increasing its size very quickly, the OpenStack Technical Committee has created a procedure to accept new components as part of OpenStack.

To grant maximum safety and code-continuity, it has been decided that the new components have to pass a given time in incubation. In this period, the component has to show a few releases as if they were already parts of OpenStack before they can be promoted to official components. This makes the incubation process pretty long but you can be assured that only high quality components are allowed to be officially part of OpenStack.
Due to this long process, we already know that some components that will soon become part of OpenStack. The following components are being considered to be part of the next release (Kilo) and some of them will very likely be integrated. These components are explained next.

**Ironic – bare metal provisioning**

The ironic goal is to provide the same interface that is used to create virtual instances in OpenStack to create real (bare metal) machines as well. The main goal of this is to help a system administrator to centralize the administration of the machines.

Amazon does not provide any service that is comparable to Ironic.

**Zaqar – cloud messaging**

Zaqar is a cloud messaging service for web developers. The service features a RESTful API, which developers can use to send messages between various components of their software and mobile applications. During the early phase of the development of Zaqar, it was known as Marconi, but since has been renamed.

Zaqar can be compared to Amazon's Amazon Simple Queue Service (SQS) service, but with the additional support for event broadcasting. Also, some features of Zaqar can be found in Amazon's Amazon Simple Notification Service (SNS).

**Manila – file sharing**

Manila is a file sharing service provider. Manila volumes are accessible as NFS and CIFS volumes, as well as through the RESTful interface. Manila also supports ACL at the file level. It can use GlusterFS, NetApp, and IBM GPFS volumes as backends.

Manila can be compared to Amazon's AWS Storage Gateway.

**Designate – DNS**

Designate is a DNS-as-a-service provider. It is able to manage multiple DNS instances for redundancy reasons and to keep them all synchronized properly. Multiple backends can be used such as PowerDNS, NSD4, FreeIPA, DynECT, and BIND9.

Designate can be compared to Amazon's Amazon Route 53.
Barbican – key management
Barbican is a key (secrets) manager for OpenStack. Barbican handles many types of secrets, including:

- Symmetric keys that can be used to encrypt Swift containers and Cinder block storages
- Asymmetric keys that can be used for secure communications such as SSL/TLS, encrypted e-mails, and SSH
- Raw secrets that can be used to keep secure data in Barbican

Barbican can be compared to Amazon's AWS Key Management Service (KMS).

Summary
In this chapter, we have seen the differences between private and public cloud with a specific focus on security, the different kind of attackers, and attacks with a focus on cloud computing, as well as the components of OpenStack.

In the next chapter, we will focus on networking security for OpenStack.
Where to buy this book

You can buy OpenStack Cloud Security from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.