**Amazon Web Services**

**vs.**

**OpenStack**

Askar Serikov, PERCCOM Cohort 5

ITMO University

June 6, 2018

**Introduction**

According to Oxford Dictionary, cloud computing is the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. Nowadays cloud computing is a huge business where companies provide clouds to individuals and organizations, so they can focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Main modern cloud computing service models are:

* Software-as-a-Service (SaaS)
* Platform-as-a-Service (PaaS)
* Infrastructure-as-a-Service (IaaS)

Cloud computing market has been growing rapidly in the last decade. Before 2006, clouds were a thing only in big enterprises which built their own infrastructures using proprietary software for remote computing. Things have changed in 2006 when Amazon Inc. introduced their new business – Amazon Web Services, which offered users an ability to use the company’s infrastructure for their own needs. Since then, many companies entered the market of cloud computing, including tech giants like Google, IBM, Microsoft. However, there is still a common fallacy that Amazon owns the cloud solutions market. A report from Forbes states[1] that Microsoft actually earned the most in revenue last year.



Figure 1. Cloud Revenue in 2017

Nevertheless, Amazon is still unbeaten on the Infrastructure-as-a-Service market with stunning 44 per cent of the market share by the year 2016[2]. It raises a reasonable question among individuals and companies of different sizes: “Should I use AWS for my work by default or are there any alternatives that better fit my needs?”. The question is especially relevant in enterprises, as in most cases so called “private clouds”, clouds that completely belong to certain organizations or individuals, are required. Main alternative to AWS on the IaaS market are OpenStack-based solutions.

OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service, whereby virtual servers and other resources are made available to customers.[3]

In this report, brief overviews of Amazon Web Services and OpenStack will be given along with general comparison and recommendations on which solution fit which tasks better.

**Amazon Web Services**

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis. The technology allows subscribers to have at their disposal a full-fledged virtual cluster of computers, available all the time, through the Internet.

Despite the fact that AWS uses open source software and contributes to many open source projects, it is a commercial service and its core software is not open source.

Here is a list of services AWS provides:

**Computing**

* Amazon Elastic Compute Cloud (EC2) scalable virtual private servers using Xen
* Amazon Elastic MapReduce (EMR) Hadoop-based big data analytics

**Networking**

* Amazon Route 53 scalable DNS
* Amazon Virtual Private Cloud (VPC) isolated EC2 instances with the ability to extend corporate networks VPN
* Amazon Elastic Load Balancing (ELB)

**Content Delivery**

* Amazon CloudFront CDN

**Storage**

* Amazon Simple Storage Service (S3)
* Amazon Glacier low-cost, long-term storage for data archival
* Amazon Elastic File System (EFS) to accompany EC2

**Database**

* Amazon DynamoDB low-latency NoSQL SSD-backed databases
* Amazon Relational Database Service (RDS) with MySQL, Oracle, SQL Server, and PostgreSQL support
* Amazon SimpleDB distributed database with EC2 and S3 interoperability, written in Erlang

**Deployment**

* AWS Elastic Beanstalk for quick deployment and cloud app management
* AWS OpsWorks EC2 configuration services via Chef, which we discussed previously

**Management**

* Amazon Identity and Access Management (IAM) to authenticate into the various services
* AWS Directory Service for tying into an on-premises Microsoft Active Directory or for setting up a new stand-alone AWS directory
* Amazon CloudWatch for application and resource monitoring
* Amazon CloudHSM Hardware Security Module for data security and for meeting regulatory compliance requirements
* AWS Key Management Service (KMS) for creating and managing encryption keys

**Application Services**

* Amazon DevPay (beta) for billing and account management
* Amazon Elastic Transcoder (ETS) for mobile video transcription from S3
* Amazon Simple Email Service (SES) for sending bulk and transactional email
* Amazon Simple Notification Service (SNS) multi-protocol application "push" notifications
* Amazon Cognito secure application-user data management and synchronization tool

**Analytics**

* Amazon Machine Learning for building regression models from publicly-available datasets

The main advantage of AWS is the hassle-free set up of customers’ clouds: everything is done via web interface where users can pick type of the cloud and available configurations. No need for building an infrastructure for the cloud and hiring specialists who will maintain it. User can have its cloud ready in minutes, which makes AWS a perfect platform for fast development and testing. Another strong side of AWS is Amazon itself. Having the biggest computing capacity in the world, the company not only offers geographically distributed network of servers across the globe, but also remains reliable partner that will not discontinue its business any time soon.

Speaking of drawbacks, even though AWS provides an Infrastructure-as-a-Service, customers do not have any control over hardware, they cannot pick certain parts for their cloud or tweak them. Another disadvantage comes from implications of “public” clouds – as we know, AWS serves hundreds of organizations on their servers. Situations where, for example, a DDOS attack on one customer can down a whole server where several other customers host their clouds are not uncommon. Same happens when a certain resource gets blocked somewhere – all adjacent resources are getting blocked too because they share a pool of IP addresses. Last but not least in the list of drawbacks is the proprietary nature of AWS. If one will decide to move their cloud off AWS, the process is not going to be as easy as setting up a new AWS cloud. It is still possible, there are examples of big companies like Dropbox and Spotify moving their clouds from AWS to other services or private clouds. However, due to proprietary software solutions AWS incorporates, the process becomes trickier.

**OpenStack**

OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS), whereby virtual servers and other resources are made available to customers.The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center. Users either manage it through a web-based dashboard, through command-line tools, or through RESTful web services.

OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA. As of 2016, it is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012to promote OpenStack software and its community.More than 500 companies have joined the project.

Here are the main components of the modular OpenStack architecture:

**Compute (Nova)**

* An Infrastructure as a Service (IaaS) system
* Management and automation of pools of computer resources
* Bare metal and high-performance computing (HPC) configurations
* KVM, VMware, and Xen hypervisor virtualization
* Hyper-V and LXC containerization
* Python-based with various external libraries: Eventlet for concurrent programming, Kombu for AMQP communication, SQLAlchemy for database access, etc.
* Designed to scale horizontally on standard hardware with no proprietary hardware or software requirements
* Interoperable with legacy systems

**Image Service (Glance)**

* OpenStack Image Service for discovery, registration, and delivery of services for disk and server images
* Template-building from stored images
* Storage and cataloging of unlimited backups
* REST interface for querying disk image information
* Streaming of images to servers
* VMware integration, with vMotion Dynamic Resource Scheduling (DRS) and live migration of running virtual machines
* All OpenStack OS images built on virtual machines
* Maintenance of image metadata
* Creation, deletion, sharing, and duplification of images

**Object Storage (Swift)**

* Scalable redundant storage system
* Automatic replication of content from failed disks to other active nodes
* Suitable for inexpensive commodity hard drives and servers

**Dashboard (Horizon)**

* GUI for access, provision, and automation of cloud-based resources for administrators and users
* Third-party billing, monitoring, management tool integration
* Customizable (brandable) dashboard
* EC2 compatibility

**Identity Service (Keystone)**

* Unified authentication system across the cloud OS
* Integration with existing backend directory services such as LDAP
* Various authentication methods: username/password, token-based systems, and AWS-style logins
* Queryable, single registry of all deployed services, with programmatic determination of access for users and third-party tools

**Networking (Neutron)**

* Manual and automatic management of networks and IP addresses
* Distinct networking models for different applications and user groups
* Flat networks (VLAN's) for separating servers and traffic.
* Static IP addresses, DHCP
* Floating IP addresses for dynamic rerouting to resources on the network
* Software-defined networking (SDN), OpenFlow, for multi-tenancy and scalability.
* Management of intrusion detection systems (IDS), load balancing, firewalls, VPN's, etc.

**Block Storage (Cinder)**

* Persistent block-level storage for databases and expandable file systems
* Block storage integration into OpenStack Compute and Dashboard for allocation of storage
* Various storage platforms supported: Ceph, CloudByte, Coraid, EMC (ScaleIO, VMAX and VNX), GlusterFS, Hitachi Data Systems, IBM Storage (Storwize family, SAN Volume Controller, XIV Storage System, and GPFS), Linux LIO, NetApp, Nexenta, Scality, SolidFire, HP (StoreVirtual and 3PAR StoreServ families) and Pure Storag
* Snapshot management for backing up data stored on block storage volumes
* Restoring of snapshots, use of snapshots as templates for new block storage volumes

**Orchestration (Heat)**

* Orchestration of multiple composite cloud applications using templates
* OpenStack-native REST API
* Cloud Formation-compatible Query API

**Telemetry (Ceilometer)**

* Billing system Single Point of Contact
* Traceable, auditable delivery of counters for billing
* Counters extensible to new projects
* Independent data collection

**Database (Trove)**

* Database-as-a-service (DaaS) provisioning relational database engine
* DaaS non-relational database engine

**Elastic Map Reduce (Sahara)**

* Hadoop cluster provisioning
* Setting of parameters based on: Hadoop version, cluster topology, node hardware details, etc.
* Cluster deployment in minutes
* Scaling of already-provisioned clusters by adding and removing worker nodes on demand

**Bare Metal Provisioning (Ironic)**

* Provisioning of bare metal machines (as opposed to virtual machines)
* Bare-metal hypervisor API
* Plugins for interacting with bare-metal hypervisors
* PXE and IPMI simultaneous provisioning, turning machines on and off as needed
* Extensible with vendor-specific plugins for additional functionality

**Multiple Tenant Cloud Messaging (Zaqar)**

* Multi-tenant cloud messaging service for Web developers
* Some components inspired by Amazon's SQS, with additional semantics for event broadcasting
* Fully RESTful API for sending messages between various components of their SaaS and mobile applications
* Surfacing of events to end users and guest agents that run in the "over-cloud" layer

**Shared File System Service (Manila)**

* Vendor-agnostic share management API
* Create, delete, give/deny access to a share
* Support for commercial storage appliances from: EMC, NetApp, HP, IBM, Oracle, Quobyte, and Hitachi Data Systems
* Support for Red Hat's GlusterFS filesystem

**DNSaaS (Designate)**

* DNS as a Service

**Security API (Barbican)**

* REST API for secure storage, provisioning and management of secrets
* Built for use in all environments, including large ephemeral clouds

**AWS Compatibility**

* Interoperability with Amazon EC2 and Amazon S3
* Minimal effort to port AWS client applications to OpenStack

As we can see, OpenStack provides an extensive list of modules and components that allows individuals and organizations to build their own clouds which, if set up correctly, are not any worse than AWS-based solutions, and all of it comes for free and it is open source. This is the main reason why enterprises invest in and support OpenStack project. They need private, independent clouds on their own infrastructure which they can shape as they want, and OpenStack provides that ability. Big advantage of OpenStack is its community, being an open source project it’s backed by hundreds of developers all over the world who constantly develop new features for OpenStack. Thus, it becomes easier and faster for OpenStack users to incorporate new features in their clouds.

OpenStack’s versatility allows users to use the platform differently:

* Use a managed service offering
* Buy hardware to run an OpenStack cloud and freely download OpenStack software and employ engineers to install, maintain, enhance, upgrade etc.
* License a distribution from a vendor. This involves an upfront license cost, annual support costs and a subsequent license renewal.
* Purchase a predictable subscription from Red Hat and receive support, maintenance, consulting, upgrades.

Speaking of drawbacks of OpenStack, installation and setting up of the system is a complicated task that requires knowledge about the whole infrastructure of the system. Maintenance of the OpenStack is also a non-trivial task that, again, can only be done by someone who has a good understanding of how the system works. Of course, capital expenses is another issue that makes the majority of small organizations, startups and individuals to choose commercial clouds as they do not require large initial investments in hardware. Also, despite of being an open-source platform, OpenStack based solutions, no matter which way you use them (through a vendor or building your own cloud), are not cheap. One must think thoroughly if their needs actually require an OpenStack-based solution.

**Suitable use cases for each cloud**

There is no the best cloud platform for everyone. One must decide on which cloud is better for them only based on their needs. Therefore, in this section lists of suitable use cases are given for both AWS and OpenStack.

|  |  |
| --- | --- |
| **Amazon Web Services** | **OpenStack** |
| - Fast Development- Hosting websites- Small organizations or individuals with limited budget- Short term projects- Projects with uncertain future- Organizations without an IT staff- Running separate modules of an application (e.g. push notifications server)- Need of a wide geographical distribution- Uncertain hardware demand | - Long-term development- Academic, scientific projects- Enterprises- High-Performance Computing- Grid Computing- Managing Big Data- DevOps-style development (where manipulating hardware is a must)- Private archiving/backing up- Having an existing infrastructure- Need to move the cloud from one infrastructure to another |

**References**

1 [Forbes](https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/sites/bobevans1/2018/02/05/why-microsoft-is-ruling-the-cloud-ibm-is-matching-amazon-and-google-is-15-billion-behind/)

2 [Gartner](https://www.gartner.com/newsroom/id/3808563)

3 ["OpenStack Open Source Cloud Computing Software"](http://www.openstack.org/)