

Cloud Computing

Devdatta Kulkarni

What is Cloud Computing?

- On-demand provisioning and management of resources
- *Resources*
 - Compute servers, database instances, object stores, load balancers
 - Compute servers can be virtual machines and/or dedicated hardware
- *Provisioning and management*
 - creation, scaling, deletion
- *On-demand*
 - When needed by application developer
 - When needed by running application
 - Automatic scale up/scale down

How does one consume the Cloud?

- If you are an application developer:
 - You can provision a virtual machine (VM)
 - This is called -- “Host” or “VM”
 - Login
 - Install Tomcat on it
 - Deploy your web application to Tomcat
 - Lets say your web app is available at the context root “myeavesdrop” and tomcat is running on port 8080
 - Adjust firewall rules on the Host to allow traffic to the Host’s IP address and port 8080
 - Your web app will be available at
 - `http://<ip-address:8080/myeavesdrop/`

Cloud computing categories

- Two basic categories
 - Infrastructure-as-a-Service (IaaS)
 - Platform-as-a-Service (PaaS)

Infrastructure-as-a-Service (IaaS)

- In this model you provision a VM, ssh into it, install application container such as Tomcat, deploy your code
 - Advantage:
 - As an application developer you have complete control over how to setup your application's environment
 - Disadvantage:
 - You have to deal with issues outside of core application development
 - Setting up Tomcat
 - Modifying firewall rules
 - Examples:
 - Commercial
 - Amazon AWS, Google Compute Engine (GCE), Microsoft Azure, IBM Softlayer, Rackspace cloud servers
 - Open source
 - OpenStack, CloudStack

Platform-as-a-Service (PaaS)

- In this model you develop your code locally and push it to the “platform”. The platform takes care of deploying it to an appropriate application container
 - Advantage:
 - You just need to worry about developing your application’s code
 - Appealing model from developers' point of view
 - Disadvantage:
 - Hard to troubleshoot since no access to the VM or the application container
 - Typically, the platforms make logs available via an API
 - Examples
 - Commercial
 - Heroku, Amazon Elastic Beanstalk, Google App Engine (GAE), Redhat OpenShift, Pivotal CloudFoundry, IBM BlueMix
 - Open source
 - OpenStack Solum, Redhat OpenShift, Pivotal CloudFoundry

IaaS Example: AWS EC2

- Deploying web application on Amazon EC2 instance
 - Check Webapp-deployments-to-public-clouds.pdf on Canvas

PaaS Example: Heroku

- Deploying web application on Heroku
 - Check [Webapp-deployments-to-public-clouds.pdf](#) on Canvas

OpenStack

What is OpenStack?

- Open source software for creating private and public clouds (<http://www.openstack.org/>)
 - For instance, using OpenStack it is possible for any organization to provide cloud computing capabilities to its users
 - CS department can install OpenStack and all those who have CS accounts can spin up virtual machines as and when needed -- similar to what you get currently from commercial vendors (Amazon, Azure, Rackspace, etc.)
- Founded in 2010 by Rackspace and NASA
 - Many more companies are involved now
 - <http://www.openstack.org/foundation/companies/>

OpenStack

- Current release
 - <http://www.openstack.org/software/ocata/>
- Integrated projects (Main projects)
 - Compute (Nova)
 - Object storage (Swift)
 - Image Service (Glance)
 - Database Service (Trove)
 - Orchestration (Heat)
 - Identity Service (Keystone)
 - Networking (Neutron)
 - Dashboard (Horizon)
 - Block storage (Cinder)
 - Application PaaS (aPaaS) (Solum)

OpenStack: High-level details

- Code
 - written in Python
 - available on github
- Projects are available in the “openstack” organization on Github
 - <https://github.com/openstack>

OpenStack: Development steps

- Sign the CLA
- Setup code submission and code review hooks on your development machine
 - Gerrit is used for code submission and code reviews
- Clone the project that you want to work on
- Work on your patch
- Test the patch using tox
- Submit the patch for review

OpenStack: Development Resources

- Devstack
 - <http://docs.openstack.org/developer/devstack/>
- Vagrant
 - <https://www.vagrantup.com/>
- VirtualBox
 - <https://www.virtualbox.org/>
- IRC channels
 - <https://wiki.openstack.org/wiki/IRC>

Contributing to OpenStack

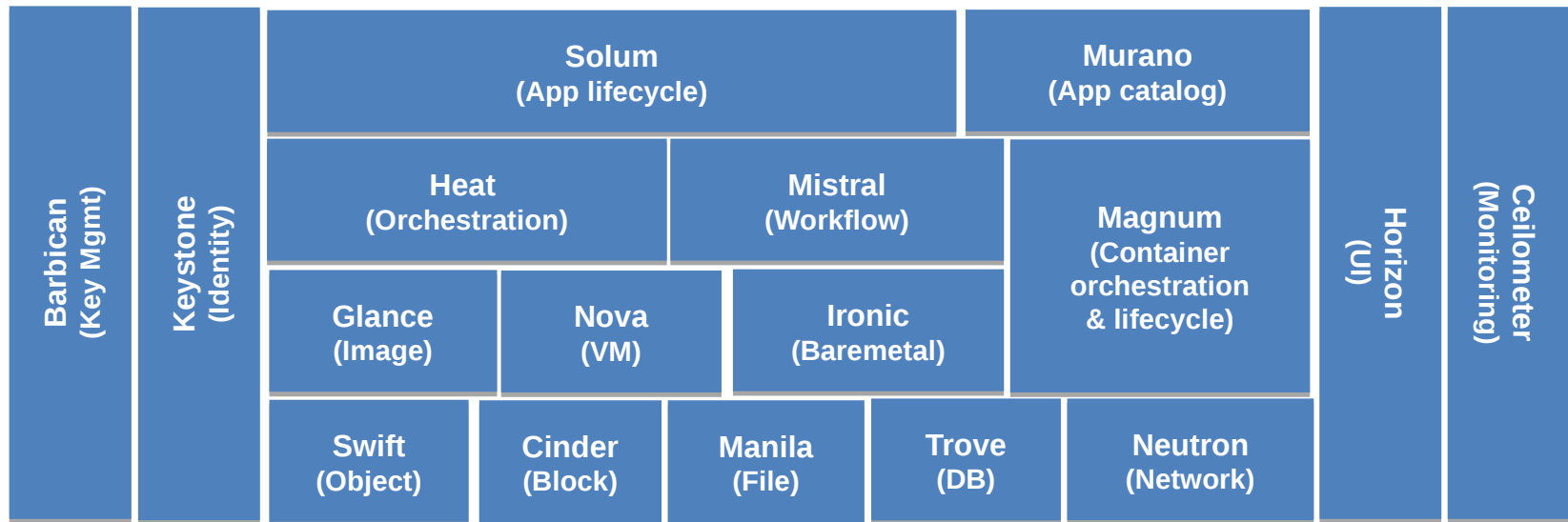
- <http://www.slideshare.net/devkulkarni/contributing-to-openstack-59590072>

PaaS Example: Solum

- System that supports testing, building, deploying of applications starting from the source code to OpenStack clouds
- Main concepts
 - App
 - Language Pack
 - Workflows

Solum in context with other OpenStack services

Solum - System for building and deploying applications to OpenStack Clouds



Solum: App

- Specifies data associated with an application
 - Github repository of the code
 - Languagepack to be used to build the application
 - Workflow to be associated with the application
 - Unit testing
 - Unit testing and build
 - Unit testing, build, and deploy
 - Commands
 - To run/start the application
 - To unit test the application

Solum: Languagepack

- It is the environment for running/testing the application
- It includes
 - Compilers (e.g.: javac)
 - Runtime libraries (e.g.: jre, jars)
 - Unit testing libraries (e.g.: tox)
- Built as a Docker (<https://www.docker.com/>) image

Solum Resources

- Solum Overview
 - <https://drive.google.com/file/d/0BxmgNoI8UMz5elk4ZXdhXzZiRWM/view>
- Solum Demo
 - <https://drive.google.com/file/d/0BxmgNoI8UMz5NXpaTFQzYXYwUzA/view>

<https://wiki.openstack.org/wiki/Solum#Resources>

http://docs.openstack.org/developer/solum/getting_started/