ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Developing Modern Application Architectures with OpenStack

Course Summary

Description

This is a fast paced, technical overview of the OpenStack cloud landscape. The focus is not necessarily OpenStack, but rather how to build a modern web application that can serve millions of users via open source tools. This survey course is targeted towards mostly technical people who want to understand the emerging world of Cloud Computing Application Development. The class will be a mix of 70% lecture and 30% labs/demos.

After introducing students to the building blocks of cloud computing (database options, queues, virtual servers, etc), we will walk through ideal use cases that use the different building blocks to achieve a holistic solution. The students will be given a ~100 page slide deck which can be used as reference material after the course.

The last third of the class focuses on NoSQL databases (especially Hadoop and Cassandra) and how to choose the correct one for your use case with some technical background on each. Students will also work on two hands-on labs exploring OpenStack using Python APIs.

Objectives

By the end of this course, students will be able to:

- Introduce students to the core concepts of Cloud Computing with OpenStack
- Provide a general overview of the most common cloud architectures
- Explain how to choose the correct cloud tools and databases for specific use cases
- Deep dive into OpenStack products
- Complete two hands on labs (~2 hours) with installing Openstack and using Openstack
- Complete two hands on demos of NoSQL databases (Hadoop + Cassandra) (~1 hour) to make the audience familiar with the interfaces of the databases
- Complete optional, time permitting: 2 additional hands on demos of Couchbase (Document database) and Neo4J (Graph Database)

Topics

- Intro to the Cloud
- OpenStack Intro
- Intro to Application Architectures
- Databases
- Putting it all together

Audience

This course is designed for sales engineers with a technical background, especially on the networking side, but not as comfortable with modern, scalable software development practices.

Prerequisites

There are no prerequisites for this course. No previous experience with cloud computing is assumed.

Duration

Two days

ProTech "Charting the Course ...
protechtraining.com

... to Your Success!"

Developing Modern Application Architectures with OpenStack

Course Outline

I. Intro to the Cloud
- A. Cloud Computing building blocks: virtual machines, databases, queues, etc.
- B. Programming languages: Java vs. Python vs Ruby
- C. REST and SOAP API technologies
- D. LAMP vs modern, scalable architectures
- E. Multi-tiered architecture use Case Deep Dive: How Netflix moved to the cloud, challenges, strategy, final results
- F. Apache Libcloud: python library to generalize cloud APIs
- G. Intro to labs for the class

Lab #1: Use DevStack to deploy a single-node OpenStack VM

II. OpenStack Intro
- A. Different flavors of OpenStack (RedHat, Dell, Cisco, Nebula, Piston)
- B. OpenStack releases/versions
- C. OpenStack Architecture Overview
- D. Dashboard (Horizon)
- E. Compute (Nova)
- F. Object Storage (Swift)
- G. Block Storage (Cinder), Ephemeral vs Volume
- H. Networking (Quantum)
- I. Images (Glance)
- J. Identity (Keystone)
- K. APIs
- L. Dashboard demo
- M. Limitations of IaaS

Lab #2: Use Openstack Dashboard to deploy VMs from Glance, creating users and roles, exploring the web UI interface for OpenStack

III. Intro to Application Architectures
- A. How to scale different layers of the app: front-end, middle-ware, databases, queues
- B. Elasticity concepts: vertical scaling, sharding, load-balancing, HPC, MapReduce
- C. Understand the differences between traditional deployment and cloud computing (Puppet/Chef)
- D. Load Balancers (HAProxy, RoundRobin, IP/RegEx based)
- E. Determine whether moving existing applications to the cloud makes sense
- F. Things you don't have to do for cloud apps: wait, plan capacity in advance, wait, run out of space/power, wait, having meetings with IT, wait, File tickets, wait, get stuck with wrong configuration
- G. Minimizing synchronous services, maximize asynchronous services
- H. Stateful vs. Stateless app design (components should not keep state across requests)
- I. One DB connection per thread instead of per request
- J. Loose coupling of components
- K. Using Apache ZooKeeper to handle partial failures in general distributed applications
- L. Queuing services & Messaging services
- M. How to use local, global and distributed caches effectively (w/ Memchached)
- N. Using proxies to filter, log and transform requests (includes collapsed forwarding proxies)
- O. High- Availability and Disaster Recovery
- P. Application integration with legacy and external apps
- Q. Considerations when optimizing performance
- R. Monitoring health and troubleshooting apps (statistics, ganglia, graphite)

IV. Databases
- A. SQL pros and cons
- B. The limitations of SQL/Relational Databases
- C. SQL vs. NoSQL

Developing Modern Application Architectures with OpenStack

Course Outline (cont'd)

D. Database distribution models: Sharding, Master/Slave replication, Peer to Peer
E. NoSQL flavors: Key-Value Databases (Riak, Redis, Memcached DB, Amazon Dynamo), Document Databases (MongoDB, CouchDB), Column Family Databases (Cassandra, HBase, Accumulo), Graph Databases (Neo4J, Infinite Graph, FlockDB)
F. How to select a NoSQL database for your use case
G. Hadoop fundamentals: HDFS, MapReduce, Hive, Pig
H. What are the ideal Hadoop use cases

V. Putting it all together

A. How to think about porting traditional apps to the cloud
B. How do cloud based apps scale? Review of the different layers
C. Cloud design patterns
D. Limitations of cloud computing
E. One more use case deep dive explaining how a company uses Rackspace to deploy a cloud app
F. Next steps for learning more about cloud computing (books, blogs, videos)
G. Q&A with students