Scalable Infrastructure Build on Cloud with Python

Python Conference 2014 at Seoul, Korea

Seungjin Kim

Hello! Developers and SysAdmins

Seung-jin Kim

- Cloud Architecture @ Fancy.com
(e-commerce / Internet industry)

- Infrastructure Services Integrator @ iPlant Collaborative
(Cyber Infrastructure Project by NSF)

- Application Analyst @ Arizona Research Laboratories
(Research/Academic Institution)

Scalable Infrastructure

- How Big is Big?
- What is scalability to you?
- Scalability as a Tool
- Scalability as a Service
- What’s the benefit from it?

Cloud

- Elastic Resources
- Resources as a service / tools
- Various meanings and terms
- Infrastructure as a Service (IaaS)

Cloud; Infrastructure as a Service (IaaS)

- Private Cloud
  - OpenStack
  - Eucalyptus
- Public Cloud
  - Amazon Web Services
  - Windows Azure
  - Google Cloud Engine
  - Rackspace

What do you do with Cloud?

- Characteristics by Industry
  - Start-ups
  - Web Services
  - Science / Research

- Characteristics by Service Provider
  - Amazon Web Services
  - Rackspace

Python with Scalable Infrastructure

- Python for Scalability
- Why Python in cloud world?
- Python is fast (practically)!
- Cloud+Python -> boto / libcloud / webservices

Python for Orchestration

Python as an orchestration tool

- boto: Python interface to Amazon Web Services
- Apache Libcloud: Python library for interacting with many of popular cloud service providers using a unified API
- pyrax: Python SDK for the RAckspace Cloud
- Azure: https://github.com/Azure/azure-sdk-for-python

Automation with Python System call

System Call

```
>>> from subprocess import call
>>> call(["uname", "-s"])
Linux
0

>>> import os
>>> os.system("uname -s")
Linux
```

Automation with Fabic

Fabric is a Python library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.

```
from fabric.api import run
def host_type():
    run('uname -s’)


$ fab -H localhost host_type
```

Automation with Ansible

Ansible?

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. (http://docs.ansible.com)

Automation with Salt Stack

Sale Stack (http://docs.saltstack.com/en/latest/topics/)?

- A configuration management system, capable of maintaining remote nodes in defined states (for example, ensuring that specific packages are installed and specific services are running)
- A distributed remote execution system used to execute commands and query data on remote nodes, either individually or by arbitrary selection criteria

Python as a DevOps Tool

Orchestrating (Cloud + Automation)

Architecting (Size does NOT matter)

Python as Cloud Tool

Scalable Infrastructure

Elastic Resource

Python Toolsets

Automation