cloud STACK
COLLABORATION CONFERENCE

Towards an automated testing environment

Amogh Vasekar
amogh.vasekar@citrix.com

LINUX FOUNDATION

Agenda

- Motivation
- Architecture
- Implementation
- Replication
- Proposal
- Enhancements

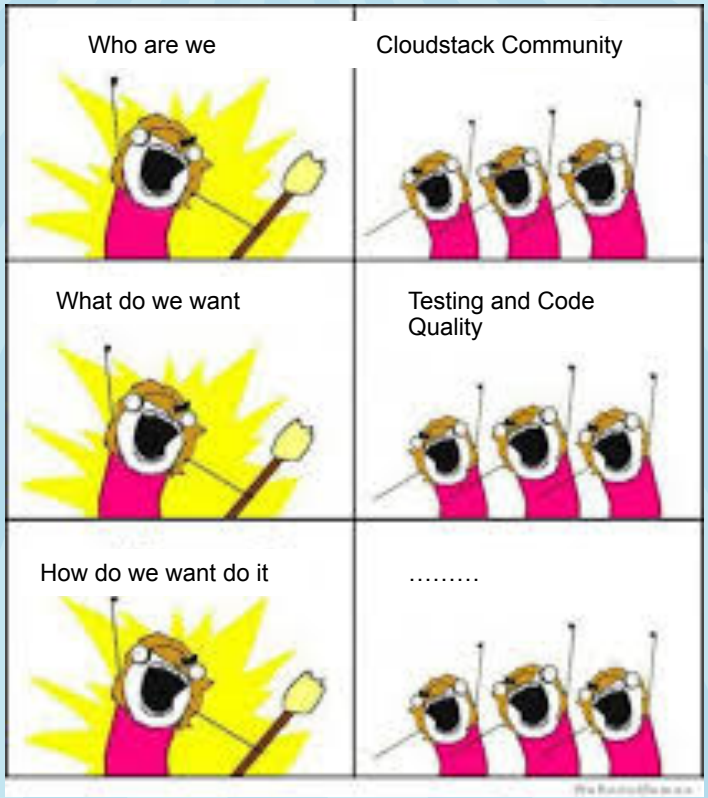
Agenda

- Motivation
- Architecture
- Implementation
- Replication
- Proposal
- Enhancements

Motivation

- Master is invariably unstable
- Unchecked commits
- No provision for developer to test feature branch
- Testing is not yet a community effort
- Not easy to test plug-ins privately
- CloudStack set-up is involved – Management server, hypervisor, network, storage

Just for laughs



cloud STACK COLLABORATION CONFERENCE

LINUX FOUNDATION

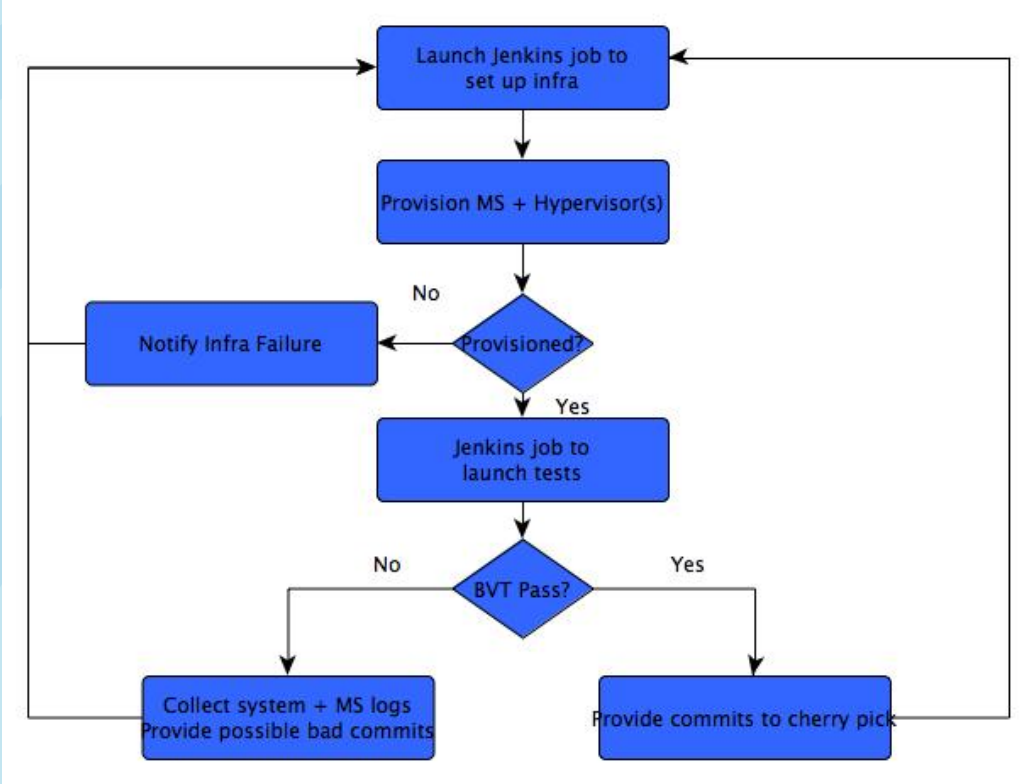
Goals

- Continuously running BVTs
- Enable community testing
- Keep master stable
- Isolate test-runs from each other, and from other infrastructures
- Easy to replicate and set-up
- Provide commits to cherry pick / buggy commits

Agenda

- Motivation
- **Architecture**
- Implementation
- Replication
- Proposal
- Enhancements

Architecture



Architecture

Phase 1 : Set-up infrastructure

- Provision management server from scratch
- Provision hypervisor from scratch
- Storage and networking set-up

Phase 2 : Run Marvin tests

- Track bad commits
- Cherry-pick good commits

Need to isolate test-runs

Based on Prasanna's effort (Big Thanks!!)

Agenda

- Motivation
- Architecture
- **Implementation**
- Replication
- Proposal
- Enhancements

Implementation

- Jenkins for integration We all like Jenkins
- Glue to hold all together
- Reporting, alerting etc.
- Separate jobs for phases 1 and 2

Implementation

Provisioning management server & hypervisors

- Need a way to deploy an OS
- Need a way to configure the systems
- Need a way to provision storage and networking in an isolated manner

Implementation

Quick overview of technologies involved

- Cobbler
- IPMI tool
- Puppet
- DNSMASQ
- Squid

Cobbler

- Open source OS installation server
- Provides baremetal provisioning via PXE
- Components
  - Distro – the OS to provision
  - Profile – associate kickstart / configuration with OS
  - System – MAC to profile mapping. Additional configs
- Snippets to run post install scripts etc
- Manages TFTP and PXE seamlessly

Others…

Puppet

- Management server configuration
- Hypervisor configuration like KVM agent etc.

IPMI tool

- To bootstrap baremetal provisioning

Squid

- HTTP proxy server (needed for isolated network requirement)

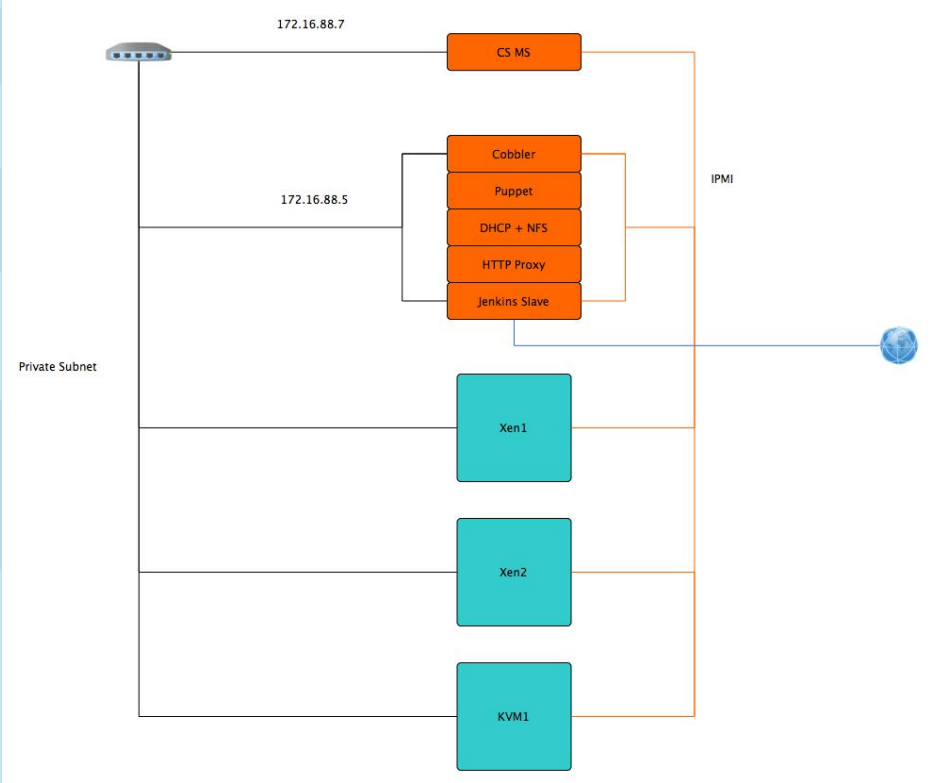
DNSMASQ

- DHCP / DNS management underneath Cobbler

Implementation

- Put all these and related technologies on a single VM
- May contain additional storage (NFS server)
- Seed system VMs
- Python code acting as launcher script

Implementation



Implementation

Minimum Requirements

- XenServer to host VM and management server
- Hosts for hypervisors
- Isolated network
- IPMI network

Implementation

Phase 2

- Run Marvin tests
- Track commits from previous stable build to current build
- If fails, one of the commits broke BVT
- If pass, auto cherry-pick commits to (stable) master
- All code in Python

Agenda

- Motivation
- Architecture
- Implementation
- **Replication**
- Proposal
- Enhancements

Replication

Setting up from scratch requires

- Familiarity with CI tools
- Familiarity with CloudStack specifics
- Lots of time
- Many DHCP / network configs

Replication

Propose to push VM like DevCloud (QACloud?)

Thus, steps become

- Load VM on XenServer
- Provision network using conf provided
- Add IPMI network
- Change MAC addresses for Cobbler (has easy to use interface)
- Add to Jenkins master
- Optionally add NFS volumes
- Optionally change domain name for hosts

Replication

To set-up from scratch

https://cwiki.apache.org/confluence/display/CLOUDSTACK/QA+Infra+from+scratch

Agenda

- Motivation
- Architecture
- Implementation
- Replication
- **Proposal**
- Enhancements

Proposal

To keep master stable

- Maintain a staging branch
- Developer commits go to staging
- If all BVTs pass, push commits
- May need more BVTs
- Reduce BVT running time (Simulator?)

Agenda

- Motivation
- Architecture
- Implementation
- Replication
- Proposal
- Enhancements

Enhancements

1. Better integration of VMWare hypervisor
2. Pull latest systemvm templates
3. Make VM size manageable
4. Scheduling mechanism for resource pools
5. Throw open to developers via a service

(Thanks to Bharath for 2,3,4,5)

Only the first step

Lets all keep master stable!!

Lets all test together!!

Big Thanks to AlexH for guiding ☺