cloud STACK COLLABORATION CONFERENCE USA

The Future of CloudStack VR

Rohit Yadav

Software Architect, ShapeBlue

rohit.yadav@shapeblue.com

$ whoami: Rohit Yadav

- **Software Architect** @ ShapeBlue.
- From Gurugram, India.
- Background:
  - Committer and PMC, 7 years and counting!
  - RM and maintainer for several minor and major releases
  - Specialize in design and architecture, development work, framework, tooling, APIs, KVM, VR/networking, debugging.
  - Author of cloudmonkey 🐵 and several features in CloudStack.
- Love cats 😸 and programming.

Topics

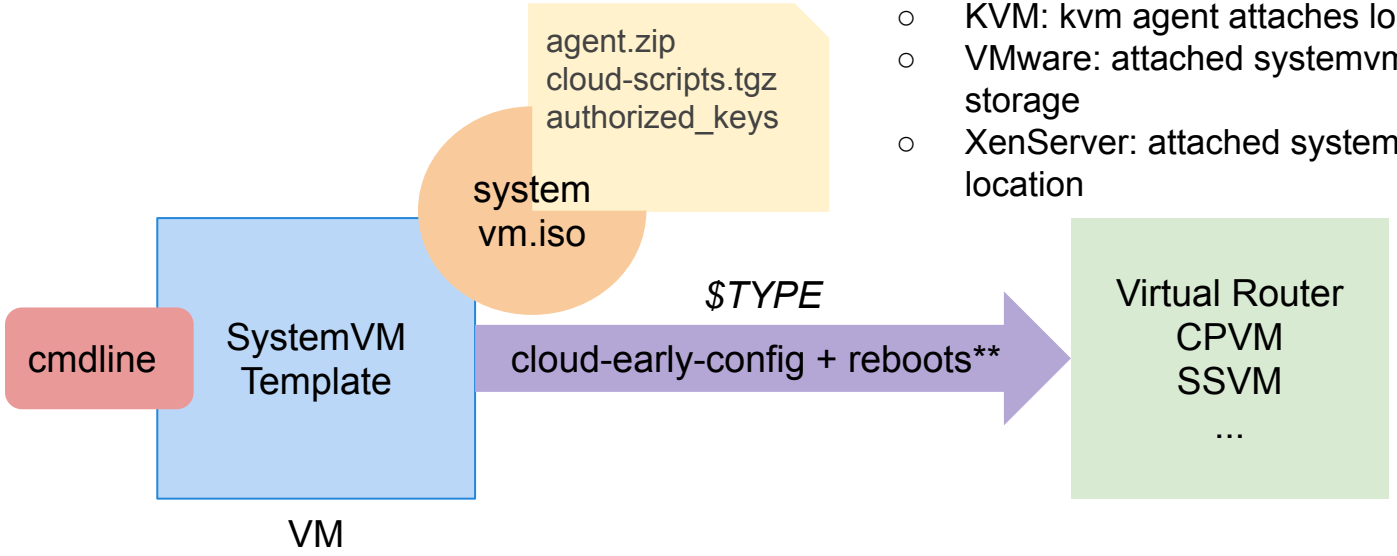
- Review of current implementation
- Zero Downtime VRs
- Proposal and Ideas
- Technical Debt and Refactoring
- Complex Use-Cases
- Q&A

Present SystemVM Template (4.11-4.13)

- Debian 9 based, GNU/Linux 4.9.0, Fixed 2G disk size
- Java RE 1.8, misc custom packages
- Custom/hypervisor specific patching (cmdline), programming, uses systemvm.iso
- Builds: CPVM, SSVM, Types of VR(s)
- Issues: Upgrade, HA/redundancy, fragmented python-based codebase

How SystemVMs are patched?

- Non-standard files, patching process
- Attached to a systemvm while booting:
  - KVM: kvm agent attaches local systemvm.iso
  - VMware: attached systemvm.iso on sec storage
  - XenServer: attached systemvm.iso from local location



Future VR: Zero Downtime

Can we reduce to “0”s downtime during upgrades at all?

Present VR: Towards near zero-downtime upgrade

ENVIRONMENT	ACS 4.9.3 AVG	ACS 4.11.1 AVG (LOWEST)	REDUCTION AVG (HIGHEST)
VMware 5.5	119s	21s (12s)	82% (90%)
KVM / CentOS7	44s	26s (9s)	40% (80%)
XenServer 7.0	181s	33s (15s)	82% (92%)

Non-redundant VR Downtime Stats
(source: previous CCC talk)

General Idea: Deploy a new VR, program it, destroy old VR, reprogram it

Availability: 4.11.2.0, 4.11.2.0, 4.13.0.0+

Reference:
https://www.shapeblue.com/working-towards-cloudstack-zero-downtime-upgrades/

What causes downtime?

- Time between old VM is unavailable/destroyed and new VM being provisioned and services come up
- Downtime due to ARP caching (~30s)

What happens during SystemVM upgrade?

- A new SystemVM template is seeded**
- On upgrade, an upgrade path changes internal config to use the new template
- On CloudStack upgrade, a new systemvm.iso
- On rolling restart, old is replaced with new service VM based on new template

When a new SystemVM Template is necessary?

- Change in package/dependency version? Such as JRE etc.
- A Security Issue
- New CloudStack version (changes in systemvm.iso)
- Fundamental changes (VR/systemvm patching, programming etc)

VR Lifecycle and Features

- SystemVM Template
  Build, Patch, Upgrade
- CloudStack + Hypervisor + Networking
- VR programming

- VM DHCP + DNS (dnsmasq)
- VM Password (cloud-password-service)
- VM Metadata (apache2)
- Guest Network (iproute2)
- Network ACLs (iptables: filter)
- Firewall Rules (iptables: filter)
- Forwarding Rules (iptables: nat, filter)
- Static NAT Rules (iptables: nat, iproute2)
- Load Balancer (haproxy)
- VPN: S2S, Remote Access, Users (ipsec:strongswan, xl2tpd, ppp)
- Static Routes (iproute2)
- Redundancy (keepalived, conntrackd)
- Health/Monitoring (monitor_service.sh)
- Network Stats (netusage.sh)

☠ DON’T EVEN TRY TO READ THIS

Present VR Programming

- Orchestration: *VirtualRoutingResource*, *VirtualRouterDeployer*
- Executable scripts at **/opt/cloud/bin/** in VR
- Executable scripts run via **router_proxy.sh** or directly in the /opt/cloud/bin path
- Commands sent as json saved at **/var/cache/cloud/** and updated in VR by **update_config.py**. On updation, they are moved and gzip-ed at /var/cache/cloud/processed.
- VR Config (VR-<uuid>.cfg) file has aggregated file+contents and commands in a custom xml format, processed by vr_cfg.sh.
- VR config jsons are stored at **/etc/cloudstack/** which is used to compare existing vs new config and only diffs (changes) are applied that are calculated by per-command type *databag* handlers (in cs_*.py, merge.py).

VR: Core vs Non-Core Services?

What causes network downtime?

- Firewall, ACLs
- NAT, SNAT/DNAT
- Forwarding Rules
- Guest Networking
- Static Routes
- VPN (ipsec, strongswan)**

What causes service downtime?

- DNS, DHCP (dnsmasq)
- Password Server
- Metadata (apache2)
- LB (haproxy)
- Redundancy (Keepalived, conntrackd)
- Misc (health, monitoring, network stats)

Typical Recovery Times

{n, number of VRs in env} x

- Services: milliseconds to seconds
- OS: few seconds to minutes
- VM: seconds to minutes
- Host: few minutes to hours
- ...

Future VR: Zero Downtime

Can we reduce to “0”s downtime during upgrades at all?

Yes, No, it **depends**!
DO NOT destroy the VRs!

Idea 💡 : Get rid of systemvm.iso?

- Multiple sources of truth:
  KVM: available on KVM host (via cloudstack-common)
  VMware: copied to secondary storage
  XenServer: copied to host
- Introduce a standard programming mechanism: config drive (iso based programming)
- Custom JRE for embedded CPVM, SSVM agents (use JDK11/jlink)
- Live Patch payload, packages, files etc against local copy from management server (source of truth)

Idea  : Get rid of VR codebase?

- Python and shell-script based
- Custom, non-standard input/output interfaces
- Hard to unit-test, test VR codebase in isolation
- Technical Debt: Maintenance, extension issues
- Fragmented codebase and services
- Upgrade issues: Versioning, sanity checks...
- Executed manually by management server or agent via SSH

Idea 💡 : Live Patch

- A small VR agent
- Non-core services lifecycle management (may run in containers)
- Kernel, iproute2, iptables/nft: Implement core VR services
- Patching mechanism(s): VR based (TLS-secured tunnel), ssh, rsync, bittorrent

Live Patch: Experiments 

- Standardise API/RPC interface, secured by CloudStack CA framework and direct non-ssh based programming (MS -> {Agent} -> VR)
  GRPC based VR agent
  (https://github.com/shapeblue/cloudstack/commits/vr-agent)
- Non-core services in containers: Exporting/Importing container images via iso
  (https://github.com/shapeblue/ccs/blob/master/scripts/create-binaries-iso.sh)
- Kernel Live Patch: Debian/Ubuntu…
  (https://linux-audit.com/livepatch-linux-kernel-updates-without-rebooting/)

Discuss: VR Programmability

- Compiled agent (Java)
- Script/codebase (Shell-scripts, Python, Py 2 vs 3 vs...)
- Best of both worlds: Go
  Go as compiled binary!
  Go as runnable script! `//usr/bin/env go run "$0" "$@"; exit "$?"`
  (https://gist.github.com/posener/73ffd326d88483df6b1cb66e8ed1e0bd)

Idea 💡 : Get rid of SystemVM Template?

- Thin and lighter template: Minimal (~50-100MB) Alpine Linux Image (https://github.com/alpinelinux/alpine-make-vm-image)
- Ship the version specific systemvm template as deb/rpm
- SystemVM Template management APIs (https://github.com/shapeblue/cloudstack/pull/45)
- Read-only ISO: No more hypervisor specific template, ship a minimal and custom live-ISO that runs with a read-only rootfs and attachable data-disk for logs, configs etc

Idea 💡 : Get rid of VRs, CPVM, SSVM?

- **CPVM**: noVNC based service on management server(s), or separate host(s)
- **SSVM**: Run agent (via deb/rpm) on separate host or on management server host
- **VR**:
  - Core-services provided by (KVM, XenServer:dom0) host kernel, non-core via containers/appliances
  - Dedicated VR infra for VMware infra

Challenges

- Live Patch and VR agent:
  Removing link-local NIC on KVM and XenServer:
  Remove link-local NIC to allow direct non-ssh based programming (upgrade issue)

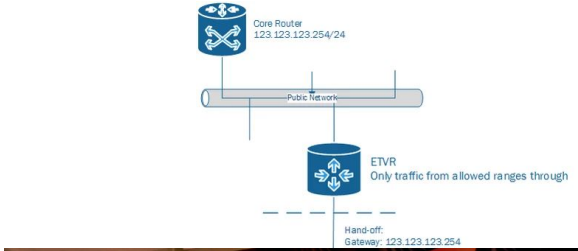
  No link-local NIC for VMware VRs
- Testing changes against other network plugins
- Not all ideas, good ideas 🥺

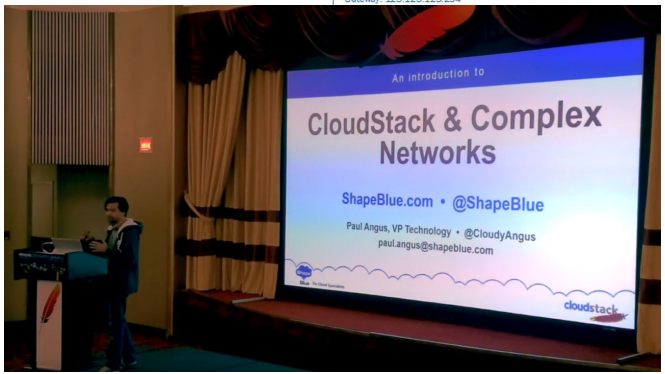
CloudStack Network Types

- L2 Networks
- L3 Networks
  - Isolated Network (Single tier/cidr)
  - VPC Network (Multi-tier/cidrs)
- Shared Networks
  - Basic Zone + Adv Zone
  - Optional with Security Groups (L2/bridge on host, only supported on XenServer and KVM)

Complex Networks

- ETVR: Like ISP-provided router
- BYO Router in CloudStack
- Routed, NAT-ed, Tunneled
- May/does not provide PF, FW, VPN etc.
- Misc: Request a range of public IPs for a network than single IPs

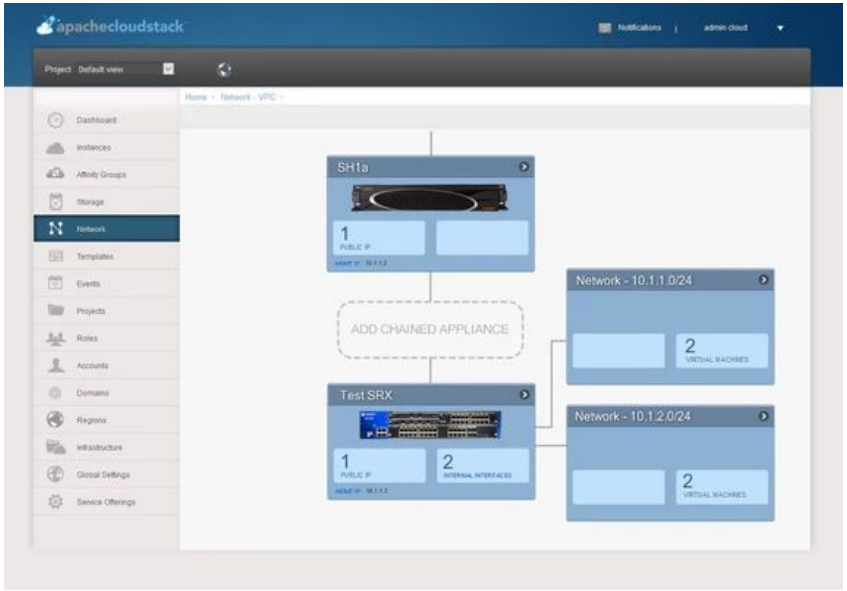




https://www.youtube.com/watch?v=XNEdjlbHQLI

BYO Router

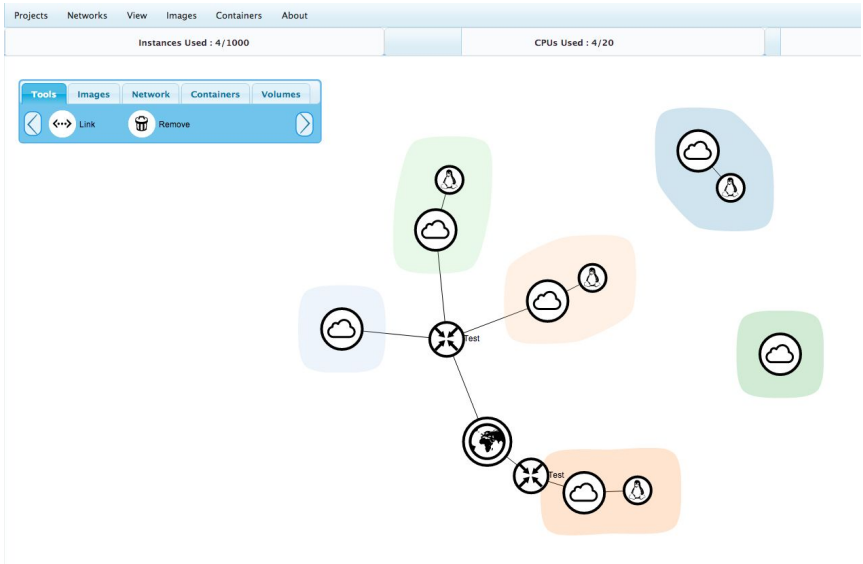
- Create special VM appliances based on templates/ISOs
- Allow users to add and use custom appliances (wizard)



Mocked UI

Network Designer and Flexible Topologies

- Design and Implement complex topologies
- Work with multiple network devices (standard and external appliance)
- Inspiration: Curvature, vCloud, custom portals



Proposal: Network Refactoring

- Refactor and amalgamate {basic, advanced} zone to just zones
- Refactor and amalgamate codebase around isolated and VPC networks, redundant and non-redundant network(s)
- Flexible network topologies and network designer
- All networks are L2 network (with isolation)
- L3 network is L2 network with capabilities provided by an appliance/router:
  - Isolated network: single-tier L2 network managed by VR
  - VPC network: multi-tier L2 networks managed by VR
  - Shared network: L2 network with DHCP/DNS provider (via VR, config drive etc)
- Discuss, gather support on dev@

CCCNA19 Hackathon

Let’s Discuss!

Q&A - Thanks!

We're Hiring!

https://www.shapeblue.com/careers/