

Webvirtcloud

Michal Švamberg, Jan Krčmář
Západočeská univerzita v Plzni

OpenAlt 2016, Brno



Úvod

- Základní požadavek - žádný kanón na vrabce.
- Prozkoumání existujících cloud řešení,
 - často složitě upravitelné,
 - nutnost agentů.
- Cílem bylo zjednodušit současnou správu virtualizační platformy xen.
 - Několik xen hypervisorů se společným polem.
 - Ruční propagace LVM.
 - Ruční registrace v DNS, DHCP.
 - Nutnost nastavit parametry instalace v systému FAI
 - Ruční příprava instalační a provozní konfigurace virtuálního stroje pro 'xl create <soubor>'

Srovnání existujících řešení

- opennebula, openstack
 - příliš komplexní,
- ovirt
 - Primárně pro RHEL, agent není připraven pro debian.
- mist.io
 - Řešení pro velké cloudy, potřeba registrovat účet na mist.io.
- archipel
 - Ovládání virtuálních strojů přes XMPP vyžaduje XMPP server, což je nevhodné.
- webvirtcloud

Webvirtcloud

- Webový frontend k libvirt, který splnil naše požadavky.
- Proti ostatním malé a snadno rozšiřitelné.
 - Napsané v python web frameworku django s využitím knihovny libvirt-python.
- apache2, mod_wsgi, sqlite
 - Apache se ukázalo jako stabilnější řešení, než nginx (http), gunicorn (wsgi server).
 - Podporuje i další databáze. Nám sqlite vyhovuje, protože velikost dat je malá.
- Přístup na konzoli přímo z webu pomocí javascript VNC klienta, novnc.
 - Novnc stojí na HTML5 Websockets a Canvas.
 - Na straně serveru běží novncd, implementace novnc serveru (novncd).

Webvirtcloud

- Snadněji se začlení do stávající infrastruktury.
 - Přidána autentizace webauthem (django modul pro basic auth).
 - Rozšíření o uživatelské kvóty, logy.
 - Navázáno na DNS a DHCP.
- Přístup na hypervisory přes ssh.
 - Standardní vlastnost libvirt.
 - Umí i další typy autentizace, které nepoužíváme.
- <https://github.com/honza801/webvirtcloud>
 - Fork z retspen/webvirtcloud (53 forků, 10 přispěvovatelů).
 - Snažíme se udržovat kompatibilní s ostatními forkly.

Navázání na DNS a DHCP

- Předregistrované hostname, mac, ip.
 - ourea1-128
- Ve webovém rozhraní možnost zvolit tyto předregistrované záznamy.
 - Pravidelně se stahuje konfigurace dhcp serveru.
 - REST api poskytuje data webovému frontendu.

Cloud architektura - HW

- Používáme výkonný hardware se sdíleným rychlým polem.
- Sítě definované v OS hypervisoru (bonding, vlan, bridge).
 - Nepotřebujeme SDN, swift.
 - Host je zapojen dvěma interfaci, které jsou v bondingu. Bonding zařízení se potom taguje do VLAN, které jsou navázány na samostatné bridge.
- Debian Linux
- RHEL cluster (cman)
 - Balíček pro Debian redhat-cluster-suite.
- CLVM (Clustered Logical Volume Manager)
 - Stará se o synchronizaci LVM metadat mezi nody.

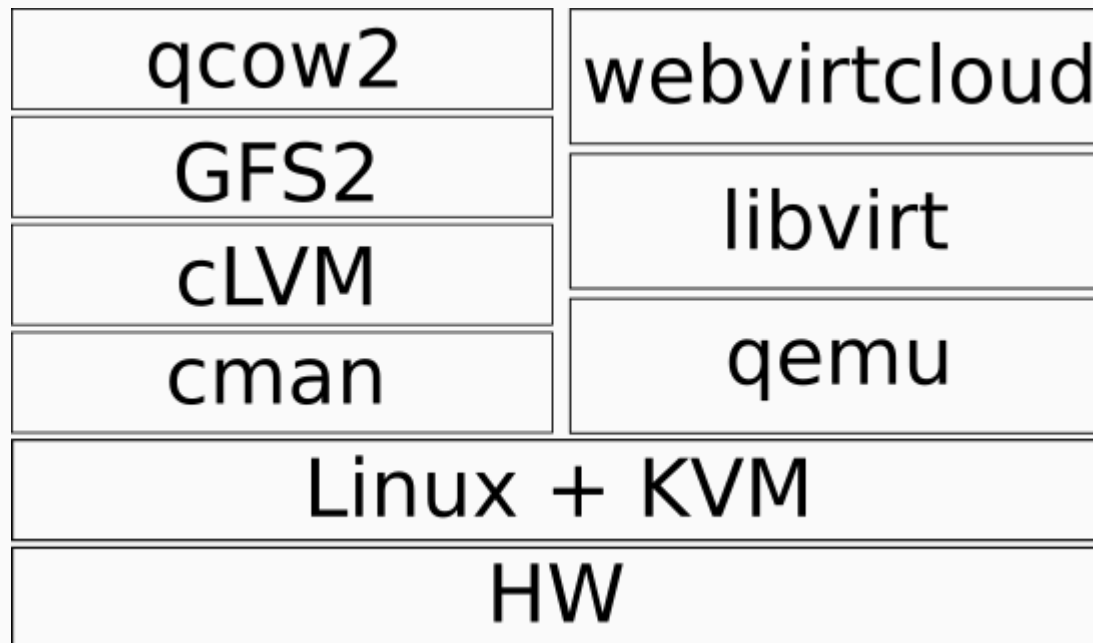
Cloud architektura - storage

- GFS2 (Global File System 2)
 - ocfs2 se ukázalo jako nestabilní.
 - Pro storage máme rychlé, pomalé LUNy. Každé ve své volume group a logical volume.
 - Z toho plynou dva filesystemy (mountpointy), které jsou nadefinované v libvirt jako directory storage.
- Testovali jsme obrazy v LVM, ale nakonec jsme se přiklonili ke qcow2.
- QCOW (QEMU Copy and write) je formát obrazu disku v souboru.
 - Velikost souboru je na začátku malá a narůstá postupně zápisem dat.
 - Možnost snapshotů.
 - Libvirt neumí lvm resize.

Cloud architektura - virtualizace

- Testovali jsme hypervisory Xen a KVM.
- Xen v debian jessie nespolupracuje dobře s libvirt.
 - Nastávaly timeouty při migraci, což šlo sice obejít, ale pak se ukázalo, že xen driver není v distribuci kompatibilní s libvirt.
 - libxl: error: libxl.c:855:libxl_domain_unpause: unpausing domain 5: Invalid argument.
- Qemu machine emulator s kvm rozšířením.
- Libvirt spravuje virtuální stroje na daném hostu.
 - V jessie problémy se systemd. Je nutné použít sysv. (Debian bug #799922, #773313)
 - error from service: CreateMachine: Activation of org.freedesktop.machine1 timed out

Cloud architektura



Další možnosti přístupu ke cloudu

- Díky tomu, že používáme libvirt, můžeme použít i další cesty ke správě virtuálních strojů.
 - Pozor při smazání stroje není konzistentní databáze webvirtcloudu.
- Virsh
 - Připojení lokálně z hosta nebo vzdáleně přes ssh.
 - Podporuje všechny operace libvirt API.
- Virt-manager
 - GTK GUI
 - Bug při vytváření qcow disku. Vyrobit pouze RAW.

Template

- Primárně provozujeme Debian Linux (jessie, stretch).
 - grub-mkimage pro bezpartisnové disky.
- guest-init skript nastaví základní parametry a služby, při prvním startu.
 - hostname, ip, config management,
- Částečně podporujeme CentOS, FreeBSD.
- Prázdný disk s možností připojení iso image.
- Windows 7, výhledově Windows 2k12r2.

Příprava Linux template

- Vytvoření qcow image (qemu-img create).
- Připojení image nelze přímo přes mount, musí se použít qemu-nbd.
 - Tímto způsobem nesedí názvy disků (/dev/vda).
- Používáme nástroj virt-rescue, který spustí virtuální stroj s initramfs s připojeným diskem a sítí (nat).
 - `virt-rescue -a debian9-template.qcow2 --network`
 - Spustíme síťování, připravíme disk, debootstrap.
 - Nastavíme základní parametry (fstab, síť, kerberos, afs, ntp).

Instance s vynucenou bezpečností

- Výhradně pro Linux.
- Vytvoření na požadavek helpdesku, protože musíme nakonfigurovat další přidružené služby.
 - Config management, monitoring, zálohování.
- Centrální správa firewallu.
- Centrální nastavení systému
 - V základu ntp, krb5, openafs, ssh, hesla, mail, fail2ban, software.
 - Dále podle požadavku apache, mysql, bacula, bind, webauth, kdc, afs server, ldap.

Správa uživatelů

- Založení uživatele automaticky po autentizaci apache.
 - Používáme SSO řešení webauth.
- Role
 - superuser, staff, clone instances
- Přidělení zdrojů (kvóty)
 - Počet instancí, paměť, cpu, velikost disků.
 - Každému uživateli přidělujeme konkrétní template/y a/nebo existující stroj.
- U každého přiděleného stroje můžeme zakázat delete, resize per uživatel.
- Logování akcí uživatele.
 - Abychom někoho usvědčili, když se nechce přiznat.

Z pohledu uživatele

- Vidí pouze ty template a instance, které mu byly přiděleny.
- Má kontrolu nad life cycle instance.
 - Power on, off, cycle, suspend.
- Nové stroje vyrábí klonováním templatů nebo vlastněných strojů.
 - Může používat pouze předgenerované hostname.
- Může změnit Přidělení zdrojů instancím.
 - Resize, delete.
- Přístup ke konzoli z webového rozhraní.
 - vnc

Statistika

- 40 virtuálních strojů na 3 hostech.
 - 72 cpu, 1T RAM, 2T diskového prostoru (fast + mirror).
- 13x Windows (i jako VDI). Základní parametry:
 - 2 CPU,
 - 4G RAM,
 - 35+10G disk,
- zbytek Linux,
 - 1 CPU,
 - 2G RAM,
 - 10G disk, 256M swap.
- Virtuální stroje mají celkem přiděleno 136 cpu, 216G RAM, 2T diskového prostoru. Díky qcow2 je skutečně zabráno 1.5T.

Co jsme získali?

- Snadnější správa oproti původnímu řešení.
 - Dříve jsme vyráběli ručně - konfigurační soubory pro xen hosty, lvm, image.
 - Administrátor na webu přehledně vidí, kde který virtuál aktuálně běží.
- Lepší podpora pro Windows a další operační systémy.
 - Full virtualizace
 - Použím paravirtualizovaných driverů se sníží režie.
- Uživatel má možnost vytvořit si vlastní stroje.
 - Může sám klonovat, nebo mít pouze kontrolu nad life cycle.
 - Můžeme uživatelům nastavovat kvóty.

Problémy

- Problém s filesystem sync. Vypadá to na problém s cache, bude chtít ještě doladit. Teď používáme directsync (podobné jako writethrough).
- Nefunkční migrace virtuálních strojů s chipsetem Q35 v rámci cloudu.
 - Tento chipset je navázaný na hardware hypervisoru.
 - Týká se strojů Windows 2012r2 po přesunu virtuálního stroje z Xen na KVM.

Financování

Projekt je spolufinancován z FR CESNET 571R1/2015.

<http://fondrozvoje.cesnet.cz/projekt.aspx?ID=571>