Aula 14 - Gerenciamento de ambientes virtualizados

Sobre

- Objetivos:
  - Apresentar os cuidados e necessidades de se ter uma ferramenta de gerenciamento.
  - Mostrar uma comparativo sobre as diferentes soluções.
  - Apresentar o Ganeti e especificar o projeto de cluster de máquinas virtuais para os alunos desenvolverem.

Ambientes pequenos

- Apenas um servidor (barato e rápido)
- Gerenciamento manual
- Backup limitado
- Sem redundância
- Não tem hardware para reposição rápida
- Alocação de VMs é manual (cp + rsync + lvcreate = magia negra)
- Sem acompanhamento da carga do hardware e se o que foi alocado é realmente necessário

Mais máquinas chegando, e agora?

- Com a chegada de mais servidores e o acúmulo de máquinas virtuais, se se continua trabalhando manualmente, o ambiente resultante simplesmente não tem como ser gerenciado.
- Toda a alocação de máquinas virtuais seria manual, não existirá redundância de nada e se caminhará cada vez mais para o colapso, pois concentremos mais serviços em menos hardware real.
- O volume de dados para se voltar de backup seria cada vez mais colossal, incluindo aí as próprias máquinas virtuais.

Quesitos para um ambiente campeão

- Escalabilidade
- Disponibilidade
- Monitoramento

- Balanceamento
- Acessibilidade
- Suporte

Escalabilidade

- Esse crescimento tem que ser sustentável e estruturado (aplica-se a qualquer ambiente, não?)
- A expansão do número de hosts tem que agregar recursos a todo o ambiente.
- Se forem compradas máquinas novas, essas máquinas tem que entrar no conjunto de hosts e estarem facilmente disponíveis para receber mais máquinas virtuais.
- A visão que precisamos ter sobre o total de disco e memória é do conjunto dos hosts, não individualmente de cada um.
- Se comprar mais máquinas, simplesmente terei mais recursos disponíveis (escalabilidade horizontal).

Disponibilidade

- Precisamos de redundância. Se pelo menos um for desligado, é possível manter todas as máquinas virtuais desse host em funcionamento, até que volte.
- Isso implica que os dados contidos no host A precisam ser replicados em tempo real para outros hosts.

Monitoramento

- Garantir boa performance e monitoramento dos recursos.
- O ambiente precisa ter um termômetro e um velocímetro para sabermos o que está acontecendo, checagem de integridade e estado geral.

Balanceamento

- O balanceamento dos hosts tem que ser fácil. Se no host A uma máquina virtual está precisando de mais CPU ou memória e no host B existem recursos disponíveis, essa transição tem que ser simples.
- A instalação, criação e alocação de novas máquinas virtuais tem que ser simples, rápida e o mais automatizada possível. Isso garante maior consistência e segurança no geral.
- A localização física da máquina virtual deve ser configurada automaticamente.

Acessibilidade

- Acesso a diferentes redes. Muitas vezes algumas máquinas virtuais precisam ter placas de rede em redes diferentes. Isso tem que ser possível.

Suporte

- Use distribuições sólidas e com suporte abundante, estável e confiável: CentOS , Red Hat, Debian , Ubuntu LTS
- Use ferramentas que também tenham suporte abundante

E quanto custa?

- Não menos importante:
  - Podemos ter tudo isso com baixo custo de implementação?
  - Não precisar de hardware sofisticado como um storage ou uma SAN?
  - Não depender de licenciamento de software proprietário?

Ferramentas livres de gerenciamento e cloud computing

- Existe uma selva de opções.
- Estão em constante e rápida evolução.
- Cada uma com suas vantagens e desvantagens.

O que procurar nessas ferramentas?

- Suporta diferentes Hypervisors? De maneira transparente?
- Tem comunidade ativa? É mantido apenas por uma empresa ou grupo fechado?
- Qual é a licença?
- Quantas pessoas e quem está desenvolvendo o código?
- É de fácil instalação? Já existe empacotado nas distribuições mais populares?
- Seria possível entender o código fonte?
- Quais são os requisitos de funcionamento? Storage? Servidores de armazenamento?
- Esquecemos de algo?

Projetos

- libvirt + Virt-Manager
- Ganeti + Ganeti Web Manager

- OpenStack
- RHEV-M 3.0
- oVirt
- ConVirt 2.0 OpenSource
- Eucalyptus
- OpenNebula
- Karesansui
- abiCloud
- OpenQRM
- Outros projetos, menções honrosas:
  - Xen Cloud Platform
  - SolusVM
  - Enomaly
  - Domain Technologie Control (DTC)
  - XenServer
  - OpenNode
  - Nimbus
  - LxCenter (Xen e OpenVZ)

Libvirt

- É o projeto mais antigo.
- Oferece uma API agnóstica de hypervisor para gerenciar máquinas virtuais.
  - Suporta: Xen, QEMU/KVM, LXC, UML, OpenVZ, VirtualBox, VMware ESX e GSX.
- Anatomia da biblioteca de virtualização libvirt: Uma API para facilitar a virtualização Linux
- Usa arquivos XML com definições razoavelmente abstratas, mas é possível especificar detalhes de cada Hypervisor quando necessário.
- É uma muito madura e robusta.
- Desenvolvida primeiramente pela Red Hat, mas com diversas outras contribuições externas.
- É utilizada por quase todos os gerenciadores como base.
- Para saber mais:
  - Libvirt: Hypervisor Independent Virtual Machine Management
  - Non-intrusive Virtualization Management using libvirt
  - Secure remote management with virtualization

Ganeti

- Ganeti foi pensado justamente para se fazer clusters de virtualização com baixo orçamento e funcionar em hardware humilde, sem depender de storage e SAN.
- Ele provê um mecanismo de replicação, baseado no DRBD
- Ele não reinventa a roda, usa apenas comandos e recursos disponíveis em qualquer distro Linux.
- O projeto é mantido principalmente por desenvolvedores do Google e outros.
- Puro Python
- Ganeti: An Open Source (GPLv2+) cluster manager for KVM
- Documentação do Ganeti
- http://www.lancealbertson.com/2010/05/creating-a-scalable-virtualization-cluster-with-ganeti/
- Creating a low-cost clustered virtualization environment
- The evolution of Ganeti, an Open Source manager for clusters of virtual machines
- DRBD: a distributed block device

Projeto do curso

- Implantar um ambiente de virtualização utilizando o Ganeti 2.4
- Documentar
  - Dependências
  - Instalação
  - Configuração e ajustes de cada nó
  - Comandos do Ganeti utilizados para criar o cluster, e o que é feito a cada etapa.
  - Como criar máquinas virtuais com e sem replicação.
  - Live Migration
- Grupos de até 4 pessoas
- Postar no wiki, será avaliado no final de semana dos dias 2 e 3 de julho.