

## Computação em Nuvem - 2018/2

### Cap. 3 - Private Cloud Stack - 4 Aulas

Raul Ikeda - rauligs@insper.edu.br

#### Grupo:

#### Objetivos

1. Entender os conceitos básicos de Private Cloud.
2. Aprofundar conceitos sobre redes virtuais SDN.

#### Pré-requisitos:

1. Terminar o capítulo anterior (Juju)
2. Realizar a leitura sobre o Openstack. [<https://www.openstack.org/>].
3. Realizar a leitura sobre o Openstack. [Jackson et al - Cap. 1 até Cap. 5]

#### Instalando - Canonical Distro

- Faça o *download* do charm **Openstack Base #57** via SSH no maas.
  - Edite o arquivo *bundle.yaml*:
    - verifique quantas máquinas serão alocadas.
    - verifique cada um dos serviços que serão instalados e em quais máquinas.
    - modifique o nome do adaptador de rede conectado na rede externa de *eno2* para *eth1* em neutron-gateway:options:data-port.
  - Verifique no MaaS se os nomes dos adaptadores estão corretos e que o DNS **interno da subnet** não esteja preenchido no dashboard (Roteiro 1).
  - Vocês ainda tem o bootstrap do Juju do roteiro anterior?
  - Instale o bundle customizado.
  - Vocês já sabem o que fazer agora: café, mas de olho no status do Juju.
  - Instale também o client do Openstack no maas e verifique:
  - \$ openstack catalog list
1. Faça um desenho de como é a sua arquitetura de solução, destacando o hardware, sistema operacional/container e respectivas alocações dos serviços.

## Configurando o Openstack

- Usando como base a página do charm instalado:
    - Importe a imagem do Ubuntu 18 e 16.
    - Configure a rede externa. Usar uma faixa de 256 IPs que não esteja sendo usado pelo DHCP do MaaS.
    - Crie a tenant-net (subnet) e roteador. Usar a subnet 192.169.0.0/24.
    - Crie agora os flavors (instance type):
      - \* m1.tiny: 1 core/1Gb/20Gb
      - \* m1.small: 1 core/2Gb/20Gb
      - \* m1.medium: 2 cores/4Gb/40Gb
      - \* m1.large: 4 cores/8Gb/40Gb
    - Crie um Key-pair, usando *public key* do próprio MaaS.
  - Acesse o Horizon (*dashboard*) como administrador:
    - Adicione a liberação do SSH no *security group default*.
    - Verifique se a topologia de rede faz sentido. Ela deve ser semelhante ao que você projetou para a sua rede física, porém agora é tudo definido por software (virtual).
    - Dispare uma instância m1.tiny sem volume adicional.
    - Aloque um floating IP para a instância. Não se esqueça do *port-forwarding*.
    - teste a conexão SSH! Em caso de erro, volte e verifique se as etapas anteriores estão corretas.
2. Faça um desenho de como é a sua arquitetura de rede, desde a conexão com o Insper até a instância alocada.

## Criando Usuários

- Entre no Horizon como administrador.
- Crie um usuário e um projeto para cada membro do grupo.
- Para cada usuário:
  - Configure a rede padrão. Não pode usar a mesma rede que o admin e outros usuários.
  - Dispare uma instância m1.small como teste.
  - Faça uma conexão SSH.
  - Remova a instância.

3. Monte um passo a passo de configuração de rede via Horizon.

## Protótipo II

- Façam individualmente o tutorial abaixo: <https://developer.openstack.org/firstapp-shade/>
- Crie um repositório no Git e coloque o material.
- **Usuários/Repositórios:**

## Deja-vu (Juju Reborn)

O Dashboard do Openstack possui alguns termos (region, instance type, security group, etc) que são semelhantes ao da AWS. Como visto anteriormente, Juju consegue operar sobre Public Cloud, Private Cloud, Bare-metal e Container. Para o setup ficar completo, agora vamos utilizar o Juju sobre o Openstack.

- Acesse a instância criada pelo admin no início do roteiro.
  - Fazer a instalação do juju. Instale também o juju-gui.
  - Adicione o Openstack como *Cloud Provider* no Juju.
4. Escreva as configurações utilizadas para incluir o Openstack como *Cloud Provider* no Juju.

- Crie um serviço Simplestreams no Openstack para prover imagem ao Juju.
  - Faça o bootstrap e instale o charm *Kubernetes Core*. Não remover ao final.
  - Testar o acesso.
5. Escreva o comando de bootstrap.

## Escalando o Kubernetes

- Verique no Hypervisor do Openstack se há espaço para mais um Kubernetes-worker.
  - Se não houver, mova uma instância para outro nó, abrindo espaço.
  - Escale horizontalmente o Kubernetes conforme os detalhes na página do charm.
6. O que é um Hypervisor? Qual o hypervisor do Openstack, da AWS e da Azure?

## Habilitando o LoadBalancer

- Acesse: <https://docs.openstack.org/neutron/queens/admin/config-lbaas.html>
  - Leia a documentação e habilite: *LBaaSv2 with an Agent*. Você irá precisar do acesso SSH via juju aos serviços correspondetes.
  - Instale também o Dashboard no Horizon.
  - Teste o serviço.
7. Escreva o seu roteiro detalhado de instalação e testes

## Questões Complementares

1. Assistir o vídeo: <https://youtu.be/ZlCoIIgLzYQ>
2. Dado que vocês trabalharam com Nuvem Pública e com Nuvem Privada, descreva com detalhes como você montaria uma Nuvem Híbrida. Como seria a troca de dados?
3. É possível somar todo o hardware disponível e disparar uma instância gigante (ex: mais memória do que disponível na melhor máquina)? Discorra sobre as possibilidades.
4. Como visto é possível rodar o Juju sobre o Openstack e o Openstack sobre o Juju. Quais os empecilhos de ter um Openstack rodando sobre outro Openstack?

## Concluindo

1. Cite e explique **pelo menos** 2 circunstâncias em que a *Private Cloud* é mais vantajosa que a *Public Cloud*.

2. Openstack é um Sistema Operacional? Descreva seu propósito e cite as principais distribuições?

3. Quais são os principais componentes dentro do Openstack? Descreva brevemente suas funcionalidades.

**Conclusão:** A arquitetura em nuvem permite diminuir o desperdício de hardware e ganhar na mobilidade de recursos. Contudo existem sérios riscos que podem paralisar as operações de uma empresa. Todo equipamento e arquiteturas complexas são passíveis de falhas tanto operacionais quanto de segurança. Como seria possível mitigar esses riscos?