

Cap. 2 - Juju - 4 Aulas

Objetivos

1. Entender os conceitos básicos sobre uma plataforma de gerenciamento de aplicações distribuídas.
2. Entender os conceitos básicos de comunicação entre aplicações e serviços.

Pré-requisitos:

1. Terminar o capítulo anterior (Bare metal)
2. Realizar a leitura sobre o Juju. [<https://jujucharms.com/>].
3. Realizar a leitura sobre redes de computadores. [Kavis - Cap. 6. Tanenbaum & Steen - Cap. 1 e Cap. 12]

Perguntas preliminares:

1. O que é e o que faz o Juju?
2. Como é o o processo de interação entre o MaaS e o Juju?
3. Defina Aplicação Distribuída, Alta Disponibilidade e *Load Balancing*?

Instalando Juju

- Acesse o maas via SSH e instale o Juju
 - `$ sudo apt install juju`
- Verifique se o Juju enxerga o MaaS como um provedor de recursos
 - `$ juju clouds`
- Caso não possua o MaaS como opção, deve adicioná-lo
 - Criar um arquivo de configuração: `maas-hw.yaml`

```
clouds:
  maas:
    type: maas
    auth-types: [oauth1]
    endpoint: http://192.168.0.3/MAAS/
```
 - `$ juju add-cloud maas ~/maas-hw.yaml`
 - `$ juju add-credential maas`
- Crie agora o *juju controller*:
 - `$ juju bootstrap maas main --to juju`
- Uma vez montada a máquina que controla o Juju, adicione-a como um recurso também utilizável.
 - `$ juju status`
 - `$ juju switch main:controller`
 - `$ juju status`
- Note que apareceu uma máquina de ID 0 disponível para uso. Instale agora o *Dashboard* de administração.
 - `$ juju deploy juju-gui --to lxd:0`
- Para acompanhar o processo, utilize:
 - `$ watch -c juju status --color`
- Para abrir o dashboard:
 - `$ juju expose juju-gui`
 - Acesse o *dashboard* do juju via browser.
- Qual o S.O. utilizado na máquina Juju? Quem o instalou?

1. O programa juju client roda aonde? E o juju service? Como eles interagem entre si?

1. O que é LXD?

Deploying Wordpress com *Load Balancing*

- Faça a instalação do Wordpress (Você pode acompanhar o processo via web)
 - `$ juju deploy wordpress`
 - `$ juju deploy mysql`
 - `$ juju add-relation wordpress mysql`
 - `$ watch -c juju status --color`
- Agora fazendo escalabilidade horizontal e *load balancing*
 - `$ juju add-unit wordpress`
 - `$ juju deploy haproxy`
 - `$ juju add-relation wordpress:website haproxy:reverseproxy`
 - `$ juju expose haproxy`
- Acesse o wordpress via Browser (com qual IP?)
- Agora remova o Wordpress
 - `$ juju remove-application haproxy`
 - `$ juju remove-application wordpress`
 - `$ juju remove-application mysql`

1. Na instalação, o Juju alocou automaticamente 4 máquinas físicas, duas para o Wordpress, uma para o Mysql e uma para o HAProxy. Fazendo um paralelo com o modelo *Public Cloud*, isso é uma característica boa ou ruim?

2. Crie um processo de implantação do Wordpress no seu hardware sem utilizar o Juju.

Protótipo I

- Cumpra o seguinte tutorial:
 - <https://jujucharms.com/docs/stable/authors-charm-writing>

Garbage Collector

- Limpando a ambiente
 - `$ juju kill-controller main`
- Desligue a máquina Juju se permanecer ligada.

Questões Complementares

1. Juju é uma aplicação distribuída? E o MaaS?

2. Qual a diferença entre REST e RPC?

3. O que é SOAP?

Concluindo

1. O que é e o que faz o Juju?

2. Como é o o processo de interação entre o MaaS e o Juju?

3. Defina Aplicação Distribuída, Alta Disponibilidade e *Load Balancing*?

Conclusão: O Juju utilizou o MaaS como provedor de recursos. O MaaS por sua vez forneceu o que havia disponível no rack. Você acha que seria necessária uma máquina de 16Gb para rodar um *Apache Webserver* ou um *Load Balancer*? Extrapole a resposta para um Datacenter real, onde as máquinas possuem configurações muito superiores. Como resolver esse problema?