

# Computação Em Nuvem

Raul Ikeda

## Cap. 2 – Deployment Orchestration – 3 Aulas

### 1. Qual o S.O. utilizado na máquina juju? Quem o instalou?

S.O. : Ubuntu 18.04.LTS

Esse sistema operacional foi instalado pelo MaaS

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

O juju client roda no MaaS e representa a interface do sistema, já o juju service roda na própria nuvem do juju, e é responsável pela implementação do sistema de serviços. Ambos interagem através da API REST.

### 3. O que é LXC e LXD?

O LXC (Linux Containers) é um método de virtualização em nível de sistema operacional para executar vários sistemas Linux isolados (containers) em um host de controle usando um único kernel do Linux.

O kernel do Linux fornece a funcionalidade cgroups que permite a limitação e priorização de recursos (CPU, memória, E / S de bloco, rede, etc.) sem a necessidade de iniciar máquinas virtuais e também a funcionalidade de isolamento de namespace que permite o isolamento completo de um aplicativo 'visão do ambiente operacional, incluindo árvores de processo , rede , IDs de usuário e sistemas de arquivos montados

O LXD é uma extensão de gerenciamento de container de software livre para Linux Containers (LXC). O LXD melhora os recursos existentes do LXC e fornece novos recursos e funcionalidades para criar e gerenciar containers do Linux.

LXD apresenta uma interface de programação de aplicativo de transferência de estado representacional (API REST) que se comunica com o LXC por meio da biblioteca liblxc. O LXD também fornece um daemon de sistema que os aplicativos podem usar para acessar o LXC e possui um sistema de distribuição de modelos para permitir a criação e operação mais rápidas de containers.

4. Explique o conceito por traz do HAProxy (reverse proxy). Vocês já fizeram algo parecido?

O HAProxy é um software livre e de código aberto que fornece um balanceador de carga de alta disponibilidade e um servidor proxy para aplicativos baseados em TCP e HTTP que distribui solicitações em vários servidores.

O HAProxy é usado por uma série de websites de alto perfil, incluindo GoDaddy, GitHub, Bitbucket, Stack Overflow, Reddit, Speedtest.net, Tumblr, Twitter e Tuenti e é usado no produto OpsWorks da Amazon Web Services.

Este foi utilizado para aplicar o load balancing no wordpress.

5. Na instalação, o Juju alocou automaticamente 4 máquinas físicas, duas para o Wordpress, uma para o Mysql e uma para o HAProxy. Considerando que é um Hardware próprio, ao contrário do modelo Public Cloud, isso é uma característica boa ou ruim?

O alojamento de 4 máquinas físicas em um modelo próprio pode ser uma característica prejudicial ao pensar-se que pode haver problemas de banda, estabilidade e manutenção, auto fornecidos por clouds públicas, e ainda assim não há backups. Nesse modelo não é garantido uma escalabilidade automática, sendo possivelmente necessário um aumento de estrutura, a manutenção deve ser realizada pela própria empresa, e não é garantido que a banda vá suportar o aumento do número de máquinas alocadas.

6. Crie um roteiro de implantação do Wordpress no seu hardware sem utilizar o Juju.

1. Instalação e Configuração do servidor web (PHP versão 5.2.4 ou superior)
2. Instalação e Configuração do banco de dados MySql versao 5.0.15 ou superior
1. Instalação do wordpress
2. Gerenciamento de arquivos (FTP)
4. Port forward ou um tunel para o wordpress
5. Efetuar login no dashboard do wordpress

6. Instalação do puglins do wordpress para a melhora de desempenho (cache, gzip etc)
7. Implementação do CDN (Content Delivery Network)
8. Gerar backup do wordpress

## Questões Complementares

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

O Juju não é uma aplicação distribuída, embora esta trabalhe com sua nuvem para garantir que, em caso de paralisação de IaaS, nem todas as unidades de um aplicativo caiam ao mesmo tempo, maximizando a disponibilidade dos outros serviços.

O MaaS pode ser uma aplicação distribuída, rodando em varias maquinas, mas no caso em que esta sendo usado (apenas uma máquina para o MaaS) ele não é um sistema distribuído, caso a única máquina do MaaS pare de funcionar o sistema do MaaS também para.

### 2. Qual a diferença entre REST e RPC?

Primeiramente, ambas podem ser usadas como protocolos que APIs vão querer usar para implementar chamadas via web.

O REST (Representational State Transfer) é um estilo de arquitetura que define padrões de comandos para serviços WEB. Ele é padronizado para conter os métodos que o HTTP usa, o famoso CRUD (Get, update, delete etc). Originalmente foi definido neste artigo (<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>) de Roy Thomas Fielding, mas hoje é um conceito que se expandiu muito além do texto. Resumindo, REST é uma relação entre cliente e servidor, onde o lado do server tem dados que são disponibilizados em formatos simples como XML ou JSON.

Já o RPC (Remote Procedure Call) é um protocolo de sub-rotinas que requisita dados de programas rodando em outra máquina ou rede. Ele usa um formato server client onde quem requisita é o client e quem manda é o server. Sem aprofundar muito no funcionamento do RPC, as maiores diferenças entre ambos métodos são:

RPC é síncrono e roda com procedures locais, podendo usar várias threads

REST é stateless (suas requisições são independentes uma das outras), enquanto RPC possui GET e POST, métodos stateful.

Finalmente, ambos são usados, apenas para propósitos diferentes, não é possível dizer que um é intrinsecamente melhor que outro.

### 3. O que é SOAP?

SOAP ou Protocolo Simples de Acesso a Objetos é um protocolo que usa a linguagem XML e HTTP para trocar informações entre dois computadores, e consegue fazer isso usando o modelo apresentado acima chamado RPC. O SOAP especifica exatamente como fazer uma chamada, então uma função pode se comunicar com outra de outra máquina e trocar informações. O HTTP não é um protocolo de transporte obrigatório, mas é muito popular. Não necessita de nenhuma especialidade de OS, então pode haver tranquila comunicação entre Windows e Linux.

## Concluindo

### 1. O que é e o que faz um Deployment Orchestrator? Cite alguns exemplos.

A tradução ao pé da letra já explica grande parte de sua funcionalidade, que é coordenar como as aplicações serão implantadas em um servidor ou até hardware. Quando são muitas, algumas precisam de prioridade, e as que são complexas precisam de configurações de instalação especiais que o orquestrador lida. Além disso, implementa as configurações necessárias para o deploy. Alguns exemplos são o JuJu e também o SSH.

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

O JuJu precisa fazer um deploy, e para tanto, ele requisita ao MaaS que ele o destine uma máquina para fazer tal deploy (virtual ou não). Quando o MaaS recebe o pedido do JuJu, liga (caso esteja desligado) um aparelho, e o JuJu envia seus arquivos para o node (no nosso caso). Toda a comunicação entre Juju, maas e nodes é feita via rede. No caso de haver um load balancer, o aparelho que será ligado/destinado ao JuJu implantar é o que o LB decidir que tem menos carga (na maior parte dos casos).

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

Alta disponibilidade é quando fazemos um servidor que consegue estar disponível 24/7 sem nunca ficar offline. Ou seja, é um termo que vem necessariamente acompanhado de gastos maiores para a empresa. Felizmente, com o IaaS e serviços como a AWS, a alta disponibilidade está cada vez mais acessível. Os gastos maiores, quando feito em um server privado, se dão por conta da quantidade de máquinas extras que serão necessárias para manter backups e serviços operando sem cair.

Load Balancing é um hardware que distribui a carga de requisições de um serviço. Como descrito várias vezes em outros roteiros, ele recebe diversas requests e decide, com base em diferentes tipos de algoritmos, para qual das máquinas disponíveis tal request será encaminhada.

Aplicação Distribuída é em conceito, uma forma de usar várias máquinas (hardwares) para rodarem a mesma aplicação, de forma a rodar mais rápido, com segurança, e com o mesmo acesso de dados que a aplicação teria se rodasse em somente uma máquina. Além disso, se um processo cair caso uma máquina falhe, ele pode ser retomado por outra máquina, garantindo integridade do uso do serviço para o cliente. Outro ponto positivo é permitir o acesso de múltiplos clientes de uma vez (considerando a aplicação como o server).

**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 32Gb 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?

No caso de um servidor grande como um Datacenter, as máquinas podem até ser super potentes, mas rodam com máquinas virtuais dentro, o que garante que uma aplicação só estará usando parte da capacidade delas. Obviamente o tamanho das aplicações muda, então o tamanho das máquinas virtuais varia. Existe também um tipo de “máquina virtual” que é modelável em tamanho, o famoso Container, não é fixo, mas roda um layer acima das VMs comuns. Ele é mais frágil que a VM, portanto nem sempre é recomendado. Datacenters e serviços como AWS costumam oferecer ambos VM e Containers ao usuario.