

## Computação em Nuvem - 2018/2

Cap. 0 - Public Cloud Intro - 2 Aulas

Raul Ikeda - rauligs@insper.edu.br

### Grupo:

### Objetivos

1. Introduzir os conceitos básicos sobre cloud computing.
2. Entender os conceitos básicos sobre IaaS.

### Pré-requisitos:

1. Ter os grupos formados.
2. Realizar leitura prévia:  
[<https://aws.amazon.com/pt/products/>]  
[<http://redhatstackblog.redhat.com/2015/05/13/public-vs-private-amazon-compared-to-openstack/>]
3. Realizar a leitura sobre Cloud Computing. [Kavis - cap. 2 e Earl et al - cap. 4]

## Explorando a AWS

Cada grupo receberá um login e senha para acessar o AWS (Amazon Web Services - <http://aws.amazon.com>).

- Link:
- Login:
- Senha:

### Tarefas:

- Crie um grupo com permissões de Administrador, ou seja, acesso total.
  - Crie um usuário para cada integrante e insira no grupo criado no item anterior.
  - **Atenção (Achtung!):** Nunca divulgue *Access Key ID* e *Secret Access Key*. O par de chaves permite realizar qualquer operação dentro da AWS e pode gerar custos extras. NUNCA fazer o upload das chaves em um repositório Git público.
  - Teste o acesso de cada usuário.
1. Descreva passo a passo o processo utilizado.

2. O que são *Policies*? Por que elas são importantes e devem ser bem definidas?

## Primeira Instância

Sugestão: fazer individual, mas respondam em grupo.

- Acessar o dashboard do EC2 de *North Virginia* (**us-east-1**).
- Criar uma instância t2.micro com Ubuntu 16.04 LTS server e 8Gb de disco.
- Verifique se o *security group* liberou a conexão SSH externa.
- Criar um novo key pair, guardar com muito carinho a private key.

– Endereço IP público da máquina:

- Teste a conexão SSH (não se esqueça da chave privada).

1. Descreva passo a passo o processo utilizado.

2. Dentro do contexto de Cloud Computing, defina os termos: *instance*, *image*, *region*, *VPC*, *subnet*, *security-group*.

3. O poder computacional das instâncias é medido em vCPU. O que é vCPU?

## Primeiro Deploy - Ghost Blog Platform

Acesse o console da instância criada e execute os seguintes comandos:

- Atualize o S.O.
    - `$ sudo apt update && sudo apt upgrade -y`
  - Instale o conjure-up usando o snap
    - `$ sudo snap install conjure-up --classic`
  - Agora instale a plataforma Ghost
    - `$ conjure-up battlemidget/ghost`
    - Selecione aws e a região correspondente (us-east-1)
    - Entre com o par *Access Key ID/Secret Access Key*
    - Selecione *Self-hosted controller*
    - Deploy!
    - Mantenha a URL como localhost.
  - Teste o deploy
    - `$ juju status`
    - Anote o IP da aplicação HAProxy.
    - Acesse a plataforma: `https://[IP Público]/admin`
1. Quantas instâncias foram criadas automaticamente? Criar instâncias automaticamente é um atributo positivo ou negativo?

2. Quanto custou o protótipo? Assuma que usou uma hora de cada instância.

## Limpando a bagunça (força bruta)

- Selecione as instâncias no dashboard do EC2.
- Terminate'em all!

1. Dada a quantidade de computadores apontada na questão anterior, descreva como você montaria um ambiente Ghost em um Datacenter próprio. Assuma que você ainda não possui nenhum hardware disponível, apenas um orçamento aprovado.
2. Agora, somando o fato de que hardware deprecia com o tempo e possui um custo mensal de manutenção, compare em termos de custo uma *Public Cloud* e um Datacenter próprio.

## Escalabilidade

Conforme visto na Aula 01, vamos agora fazer uso de uma das mais poderosas características da nuvem.

- Crie uma instância t2.micro com Ubuntu 16.04 LTS:
    - \$ sudo apt update; sudo apt install nodejs build-essential
    - Edite o arquivo hello.js:

```
#!/usr/bin/env nodejs
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8080, ' ');
console.log('Server running at http://localhost:8080/');
```
  - Execute o serviço:
    - \$ chmod +x ./hello.js
    - \$ ./hello.js
  - Libere a porta no *security group* correspondente. Teste o serviço.
  - Ajustando:
    - \$ crontab -e
    - @reboot /home/ubuntu/hello.js
    - \$ sudo reboot
  - Teste novamente.
1. O que faz o crontab?

## Montando o Autoscaling Group

- Crie uma AMI da instância. Café.
  - Crie um *LoadBalancer Classic* que mapeia instâncias:8080 para loadbalancer:80.
  - Adicione a instância existente.
  - Teste o acesso ao *LoadBalancer*:
    - Endereço:
  - Remova a instância do LoadBalancer e a destrua.
  - Crie um *Autoscalling Group*:
    - Criar um *Launch Configuration* da mesma configuração em que foi criada a instância original. Reutilizar o *Security Group* com as devidas liberações.
    - Começar com 2 instâncias.
    - Selecionar uma *subnet*.
    - Anexar o *LoadBalancer* criado anteriormente.
    - Selecionar: *Use scaling policies to adjust the capacity of this group*.
    - Entre 2 a 5 instâncias com média de CPU a 30% e *warm up* com 60s.
  - Aguarde as instâncias subirem e teste o *LoadBalancer* de novo.
1. O que é uma AMI?

2. O que faz o *LoadBalancer*? Explique o algoritmo utilizado.

## Fazendo uso da Escalabilidade Horizontal

- Entre em uma das instâncias.
- Altere a mensagem do webserver e reinicie.
- Verifique se a instância ficou out of service no load balancer.
- Se sim, aguarde até que volte InService.
- Teste novamente o LoadBalancer várias vezes e veja a nova mensagem aparecer esporadicamente.
- Entre em uma das instâncias novamente.
- Rode um loop infinito. Espere alguns minutos.
- Verifique se alguma nova instância surgiu.
- Cesse o loop infinito. Espere alguns minutos e veja o número de instâncias diminuindo.
- Remova: Autoscaling Group, Launch Configuration, Load Balancer, Todas as instâncias.

1. Enfim, como funciona o *Autoscaling Group* da AWS?
2. Qual a diferença entre escalabilidade horizontal e escalabilidade vertical?
3. Qualquer serviço pode fazer uso desse modelo?

### Questões Complementares

1. O que é um *VPS*? Qual a diferença entre uma instância e um *VPS*?
2. Defina *Platform as a Service* (PaaS) e *Software as a Service* (SaaS).

3. O modelo *LoadBalancer* possui um custo fixo elevado. Como você justificaria o uso e a configuração de um *LoadBalancer* para uma empresa?

## Concluindo

1. Defina *Public Cloud*. Cite a principal vantagem e desvantagem.

2. Defina *Infrastructure as a Service* (IaaS).

3. Defina Escalabilidade.

**Conclusão:** Imagine como é o processo de alocação de uma instância. O que é realmente uma instância? Como você montaria um ambiente similar a AWS em um Datacenter próprio?