



Computação em Nuvem

Arquitetura, Design e Implementação de Sistemas
Roberto Ribeiro Costa Lima Junior, MSc.

Breve Apresentação

- O que você faz atualmente? Trabalha? Estuda?
Onde?
- Motivação para fazer essa pós-graduação?
- O que você espera desse módulo?

Objetivos

- Apresentação de definições e conceitos sobre computação em nuvem
- Prática!!!
- No final:
 - Instanciar máquinas virtuais na nuvem
 - Configurar balanceamento de carga para aplicações Web
 - Configurar alarmes
 - Criação de Imagens customizadas (AMIs)
 - Configurar auto escalabilidade a partir de parâmetros especificados
 - Configurar banco de dados na nuvem
 - Configurar backup no S3
 - **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem

Artigo

- Above the Clouds: A Berkeley View of Cloud Computing

Avaliação

- Trabalho em grupo no último dia

Detalhamento da Agenda

Dia 1: Apresentação do curso, Introdução, breves conceitos, uma breve demonstração da plataforma da AmazonWS e criação das contas

Exposição: Introdução, Motivação e Principais conceitos básicos.

Demonstração: Amazon EC2.

Laboratório: Criação das contas de usuários na Amazon WS (Free). Inicializar máquina virtual, instalar programas, configurar grupos de segurança, acessar remotamente aplicação web.

Dia 2: Estado da Arte

Exposição: Exposição do estado da arte e artigo "Above the Clouds".

Laboratório: Criação de AMI, Load Balance de aplicação

Dia 3: Virtualização e Laboratório

Exposição: Virtualização

Laboratório: Autoscaling, RDS e S3

Detalhamento da Agenda

Dia 4: Amazon RDS e Heroku
(PaaS)

Laboratório: Heroku e ElasticBeanstalk

Dia 5: API's (SaaS) para
alavancar aplicações

Exposição: API's diversas da AWS, Azure e Google
Laboratório: Utilização de uma API da Azure para alavancar
aplicações

Dia 6: Trabalho

Laboratório: Trabalho

Dia 1



Motivação

Apache

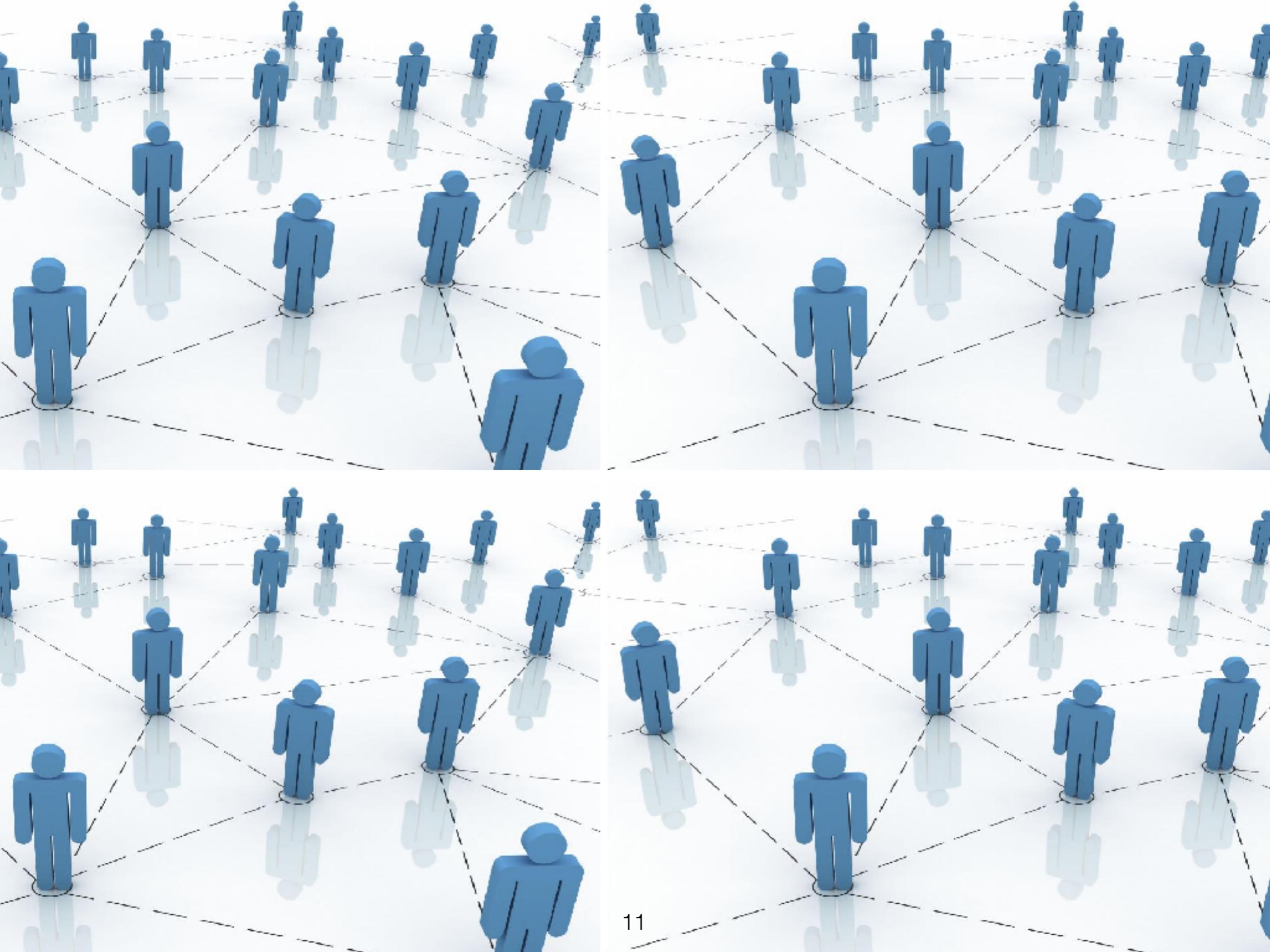
HTTP SERVER



MySQL®

php













E se eu crescer rápido demais e precisar de muitos recursos?

E se eu não crescer o esperado?

E a segurança das minhas aplicações?

E os backups?

E os picos de utilização X os horários “mortos”?

<http://www.youtube.com/watch?v=QJncFirhjPg>

“There is no cloud just other people's computers.”

–Unknown author

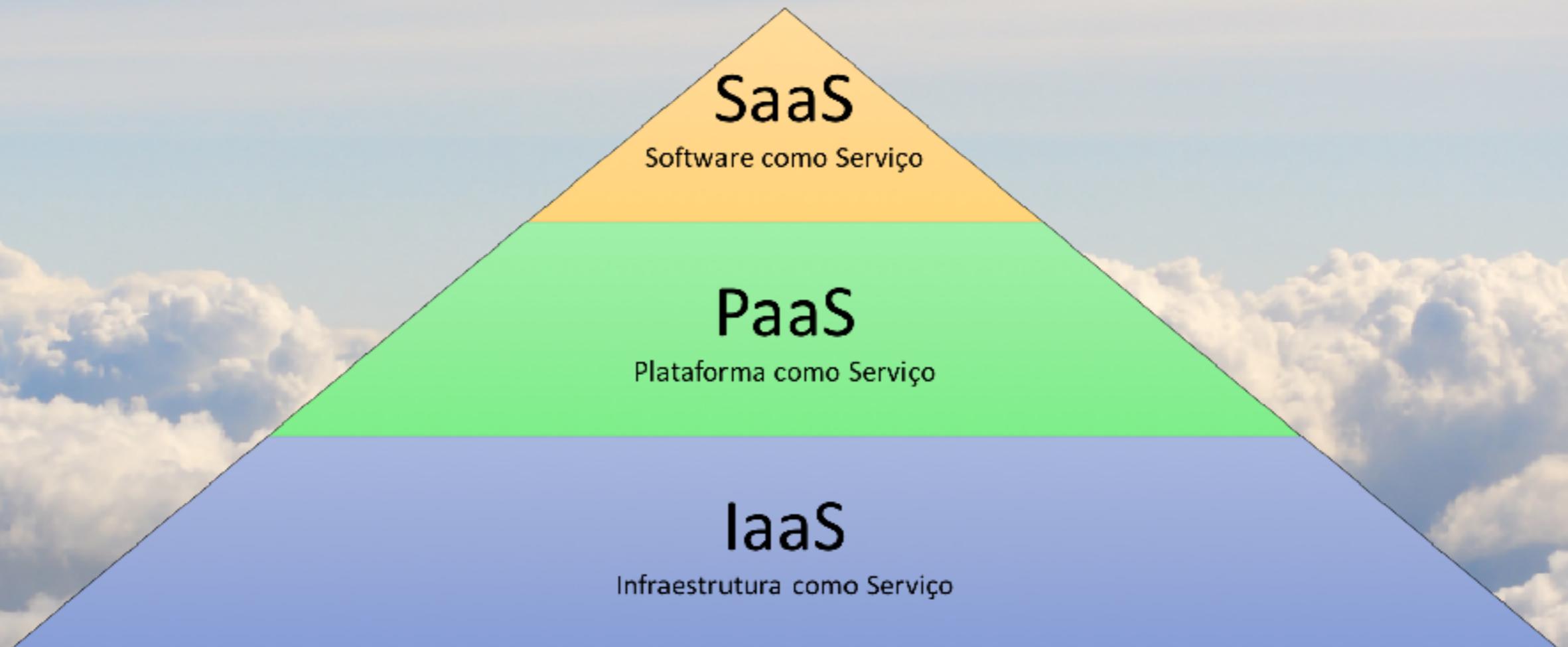
“A computação em nuvem é um modelo computacional que permite o acesso, sob demanda e através da rede (comumente na internet), a um “pool” de recursos computacionais (redes, servidores virtuais ou físicos, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados”

–Américo Sampaio

“Poder computacional infinito pago sob demanda”

Características

- Self Service
- Elasticidade
- Pagamento pelo Uso





Google Compute Engine

Infraestrutura como Serviço (IaaS)



 API Gateway	 DynamoDB	 Mobile Hub
 AppStream	 EC2	 OpsWorks
 AWS IoT	 EC2 Container Service	 RDS
 Certificate Manager	 Elastic Beanstalk	 Redshift
 CloudFormation	 Elastic File System <small>PREVIEW</small>	 Route 53
 CloudFront	 Elastic Transcoder	 S3
 CloudSearch	 ElastiCache	 Service Catalog
 CloudTrail	 Elasticsearch Service	 SES
 CloudWatch	 EMR	 SNS
 CodeCommit	 GameLift	 SQS
 CodeDeploy	 Glacier	 Storage Gateway
 CodePipeline	 IAM	 SWF
 Cognito	 Import/Export Snowball	 Trusted Advisor
 Config	 Inspector <small>PREVIEW</small>	 VPC
 Data Pipeline	 Kinesis	 WAF
 Device Farm	 Lambda	 WorkDocs
 Direct Connect	 Machine Learning	 WorkMail
 Directory Service	 Mobile Analytics	 WorkSpaces
 DMS		



Amazon Web Services

Modelo	vCPU	Créditos de CPU/hora	Mem (GiB)	Armazenamento
t2.nano	1	t2.nano = US\$ 4,84 / mês t2.micro = US\$ 9,67 / mês t2.large = US\$ 77,38 / mês		Somente EBS
t2.micro	1			Somente EBS
t2.small	1	12	2	Somente EBS

	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GB)	Usa do Linux/UNIX
Uso geral – Geração atual					
t2.nano	1	Variável	0,5	Somente EBS	\$0.0065 por hora
t2.micro	1	Variável	1	Somente EBS	\$0.013 por hora
t2.small	1	Variável	2	Somente EBS	\$0.026 por hora
t2.medium	2	Variável	4	Somente EBS	\$0.052 por hora
t2.large	2	Variável	8	Somente EBS	\$0.104 por hora

24	4	Somente EBS
Recursos:		
36		
		<ul style="list-style-type: none"> Processadores Intel Xeon de alta frequência com turbo de até 3,3 GHz CPU com capacidade de intermitência, regida por créditos de CPU e desempenho de linha de base constante Tipo de instância de uso geral de custo mais baixo e elegível ao nível gratuito (apenas t2.micro) Equilíbrio entre recursos de computação, memória e rede



Plataforma como Serviço (PaaS)



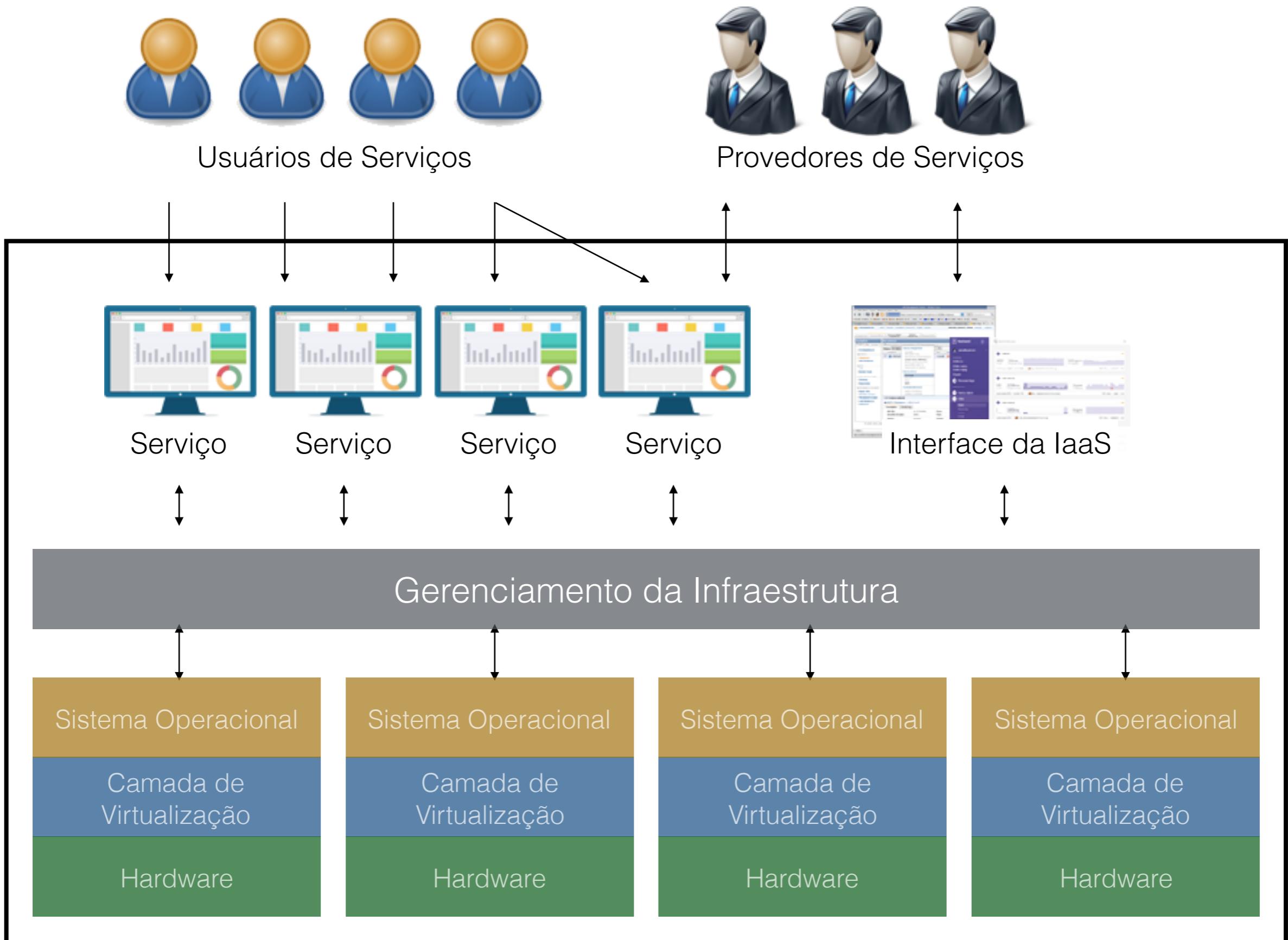


Google Apps for Work



Software como Serviço (SaaS)

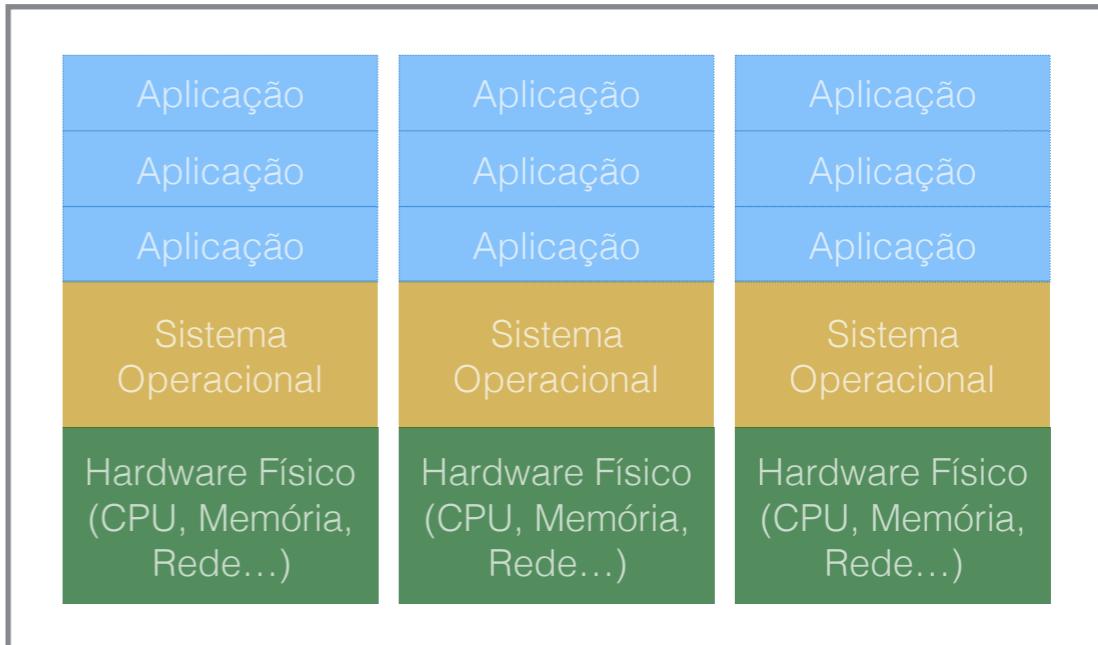




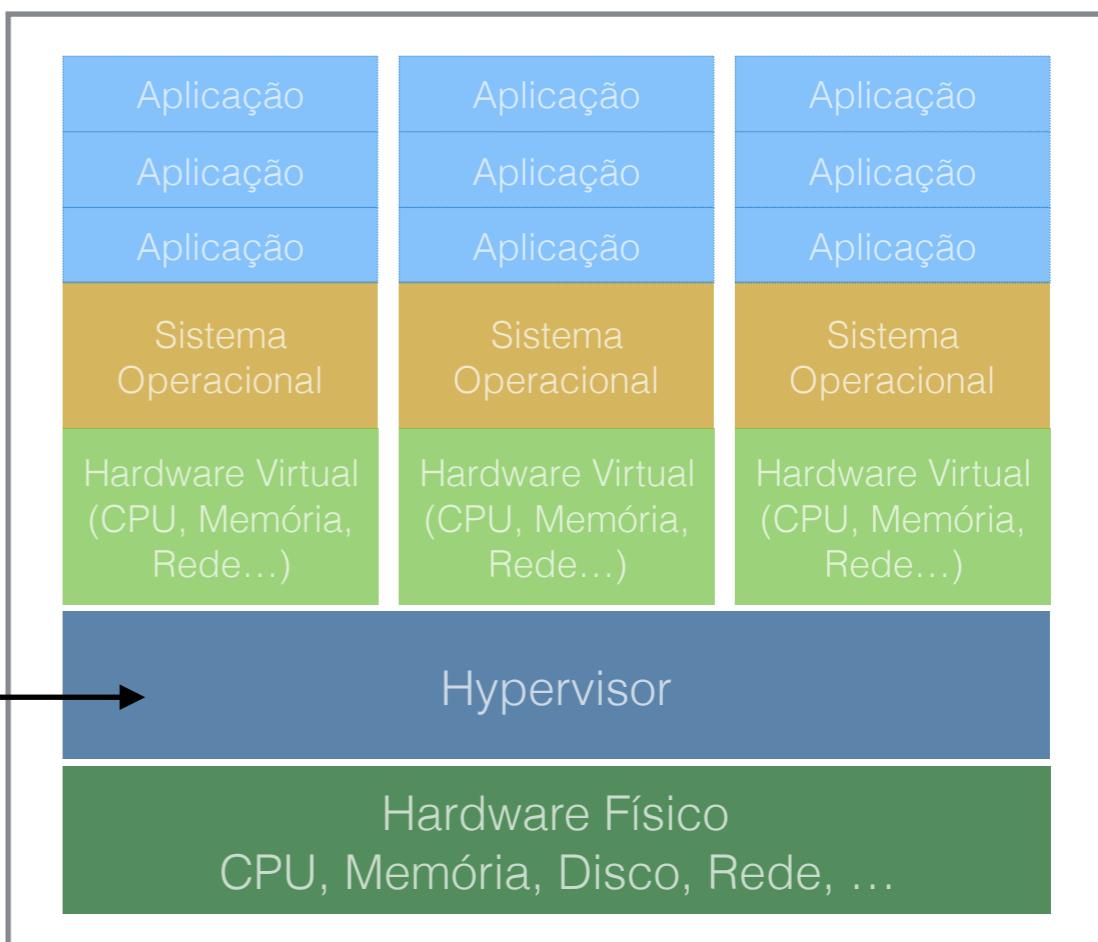
Virtualização

- Adiciona uma camada de abstração entre as aplicações e o hardware
- Isola os recursos computacionais
- Segurança entre as máquinas virtuais
- Maximiza o uso da infraestrutura de TI
- VMWare, Xen, HyperV, KVM...

Sem Virtualização



Com Virtualização



Nuvem X Grid

Grid Computing

- Modelo mais voltado para a pesquisa que envolve cooperação entre instituições
 - Também está presente em aplicações comerciais como as operações financeiras
- A idéia do grid também é a de compartilhar recursos computacionais
 - Para que pesquisadores submetem processos (jobs)
 - Para resolver problemas computacionais complexos

Diferenças entre Nuvem e Grid

- Modelo de Negócios

Nuvem	Grid
Pagar pelo uso (Pay per use)	Oferece recursos computacionais gratuitos entre os membros que participam do projeto

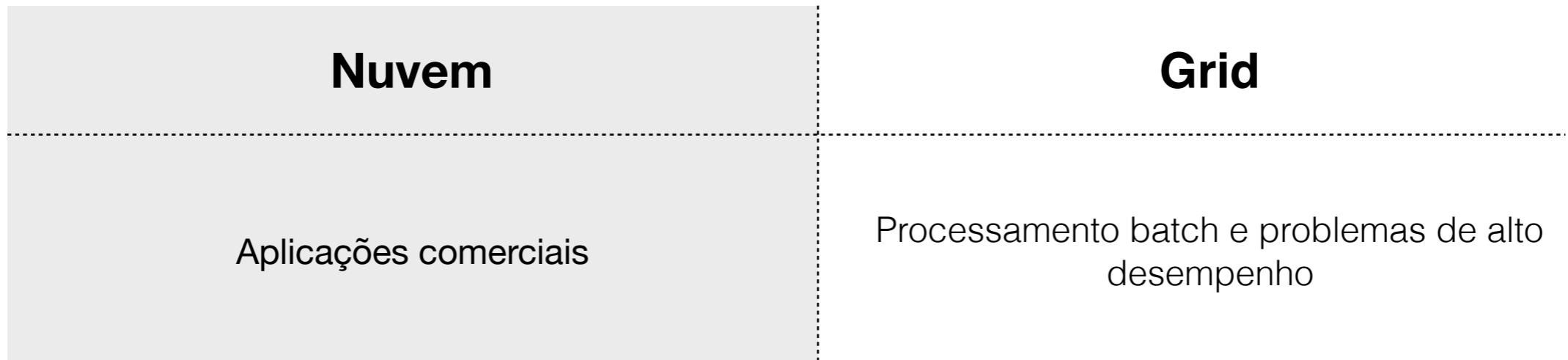
Diferenças entre Nuvem e Grid

- Arquitetura e Distribuição

Nuvem	Grid
Soluções oferecidas pela mesma empresa (embora que possivelmente distribuídas geograficamente). A Amazon, por exemplo possui datacenters espalhados em vários lugares do mundo inclusive no Brasil	Recursos computacionais compartilhados (como por exemplo máquinas) podem estar em regiões geográficas diferentes e pertencem a instituições distintas
Distribuição das aplicações por máquinas virtuais	Distribuição através de jobs submetidos para processamento em batch

Diferenças entre Nuvem e Grid

- Tipos de Aplicação



Nuvem X Hosting

Diferenças entre Nuvem e Hosting

Nuvem	Hosting
Pay per use	Cliente paga mensalidade
Cliente aluga máquinas virtuais	Cliente aluga um servidor físico
Provisão da aplicação se dá em minutos	Provisão lenta da aplicação

Demonstração Amazon EC2

www.amazon.com/ec2

Criação de Conta na AWS

Segurança

- **Exemplo da Amazon EC2**
 - **Acesso as máquinas virtuais**
 - Clientes tem root access
 - Clientes controlam o acesso as suas AMIs
 - **Firewall**
 - EC2 possui mecanismo de grupos de segurança
 - Os grupos de segurança podem restringir faixas de IPs para acessos a determinadas portas
 - Ex.: IP 177.23.45.67 pode acessar a porta 22 (ssh) e o restante da Web só pode acessar a porta 80

Segurança

- Exemplo Amazon EC2
 - Usuários precisam registrar no site e baixar o certificado
 - Chamadas a API passam o certificado

Bonus 1

Criar uma instância na Amazon, acessar a máquina, instalar um Apache Http Server e liberar as portas de acesso

```
ssh -i arquivo.pem ubuntu@IP  
sudo apt-get install apache2
```

Bonus 2

Liberar acesso da máquina (ssh) apenas para o IP da FA7 e liberar o acesso http para todo mundo

Dia 2

Aula passada

- Motivação para a Computação em Nuvem
- Principais conceitos
- Nuvem X Grid X Hosting
- Demonstração da Amazon EC2
- Criação das Contas
- Laboratório para abertura de instância na EC2

Objetivos Gerais

-  Instanciar máquinas virtuais na nuvem
 - Configurar balanceamento de carga para aplicações Web
 - Configurar alarmes
 - Criação de Imagens customizadas (AMIs)
 - Configurar auto escalabilidade a partir de parâmetros especificados
 - Configurar banco de dados na nuvem
 - Configurar backup no S3
 - **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem

Objetivos desta aula

- Conceitos: Above the Clouds: A Berkley View of Cloud Computing
- **Prática:**
 - Revisão do EC2
 - Criação de imagens (AMIs)
 - Load balance de aplicação

Executive Summary

“Cloud Computing, the long-held dream of Computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased.”

Executive Summary

“Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it”

Executive Summary

“using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT”

Executive Summary

“The illusion of infinite computing resources available on demand, thereby eliminating the need for Cloud Computing users to plan far ahead for provisioning”

Why Now, Not Then?

- Na Web 1.0, aceitar pagamento por cartões de crédito exigia contratos com as operadores (VeriSign ou Authorize.net). Era muito difícil para um pequeno negócio aceitar pagamentos por cartão de crédito;
- Com a chegada do PayPal, qualquer empresa ou até indivíduo pode receber pagamentos por cartão de crédito com mínimo esforço;
- Isso pode parecer uma diferença pequena, mas possibilitou a entrada de uma multidão no negócio online.

Why Now, Not Then?

- Similarmente, páginas de “Indivíduos” podem utilizar o Google AdSense para vender anúncios sem a necessidade de construírem um relacionamento com os anunciantes, agências, etc;

Why Now, Not Then?

- Amazon percebeu esse movimento e passou a oferecer a mesma filosofia que a PayPal e a Google, porém com recursos computacionais;
- Qualquer um pode ter uma “super infraestrutura computacional” para disponibilizar os seus serviços.

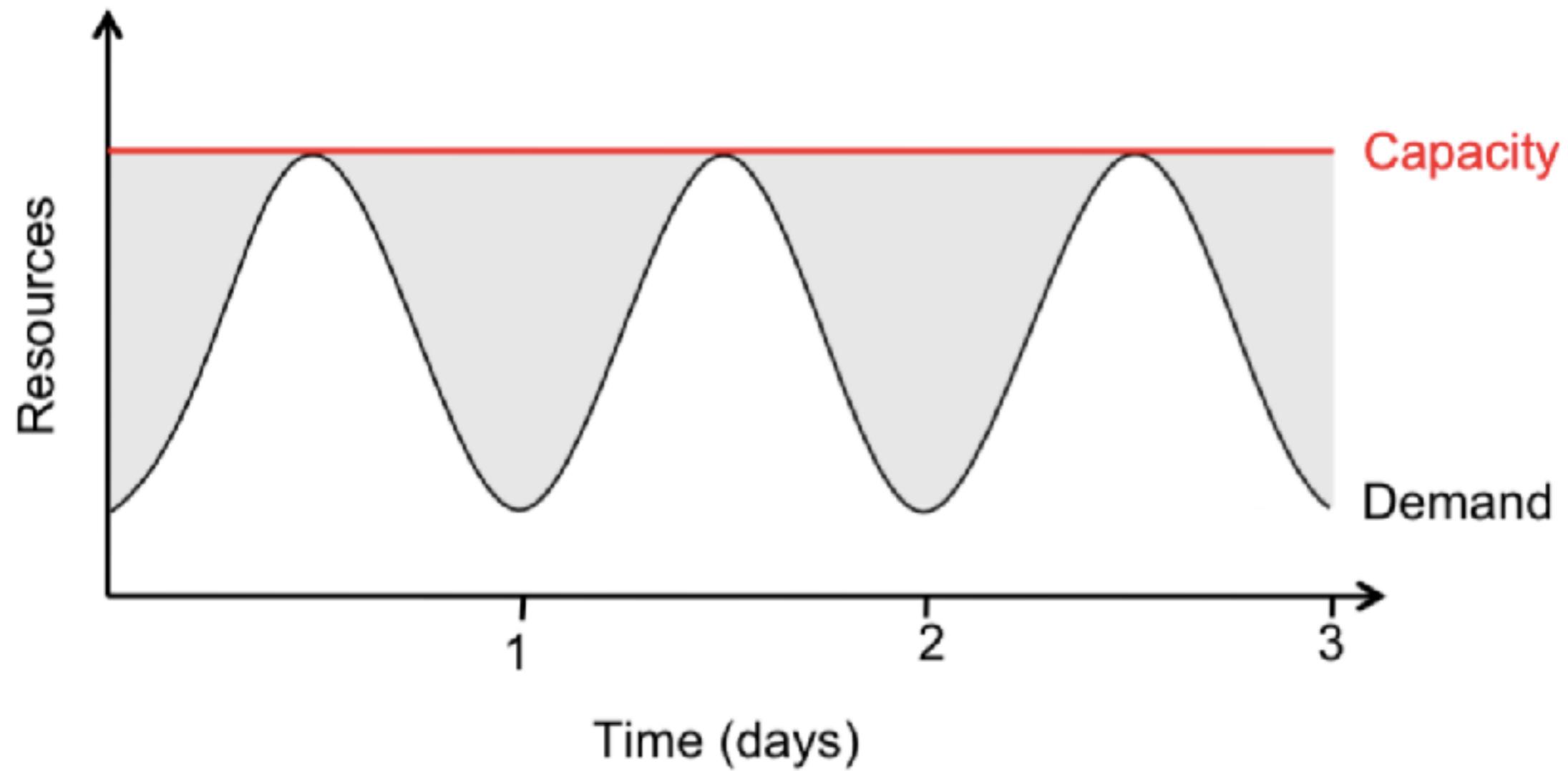
Cloud Computing Economics

- “*Although the economic appeal of Cloud Computing is often described as “converting capital expenses to operating expenses” (CapEx to OpEx), we believe the phrase “pay as you go” more directly captures the economic benefit to the buyer.*”

Cloud permite que se troque despesas de capital e focar no negócio
(as despesas são convertidas de investimento para operacionais)

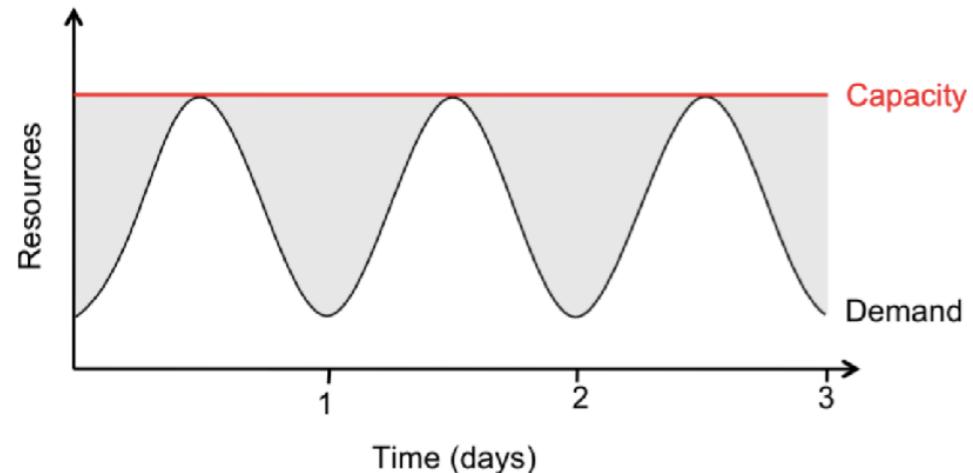
- “*Therefore, even though Amazon’s pay-as-you-go pricing (for example) could be more expensive than buying and depreciating a comparable server over the same period, we argue that the cost is outweighed by the extremely important Cloud Computing economic benefits of elasticity and transference of risk, especially the risks of overprovisioning (underutilization) and underprovisioning (saturation).*”

Apesar do custo EC2 ser maior que comprar um servidor mas possui os seguintes benefícios: elasticidade e transferência de riscos

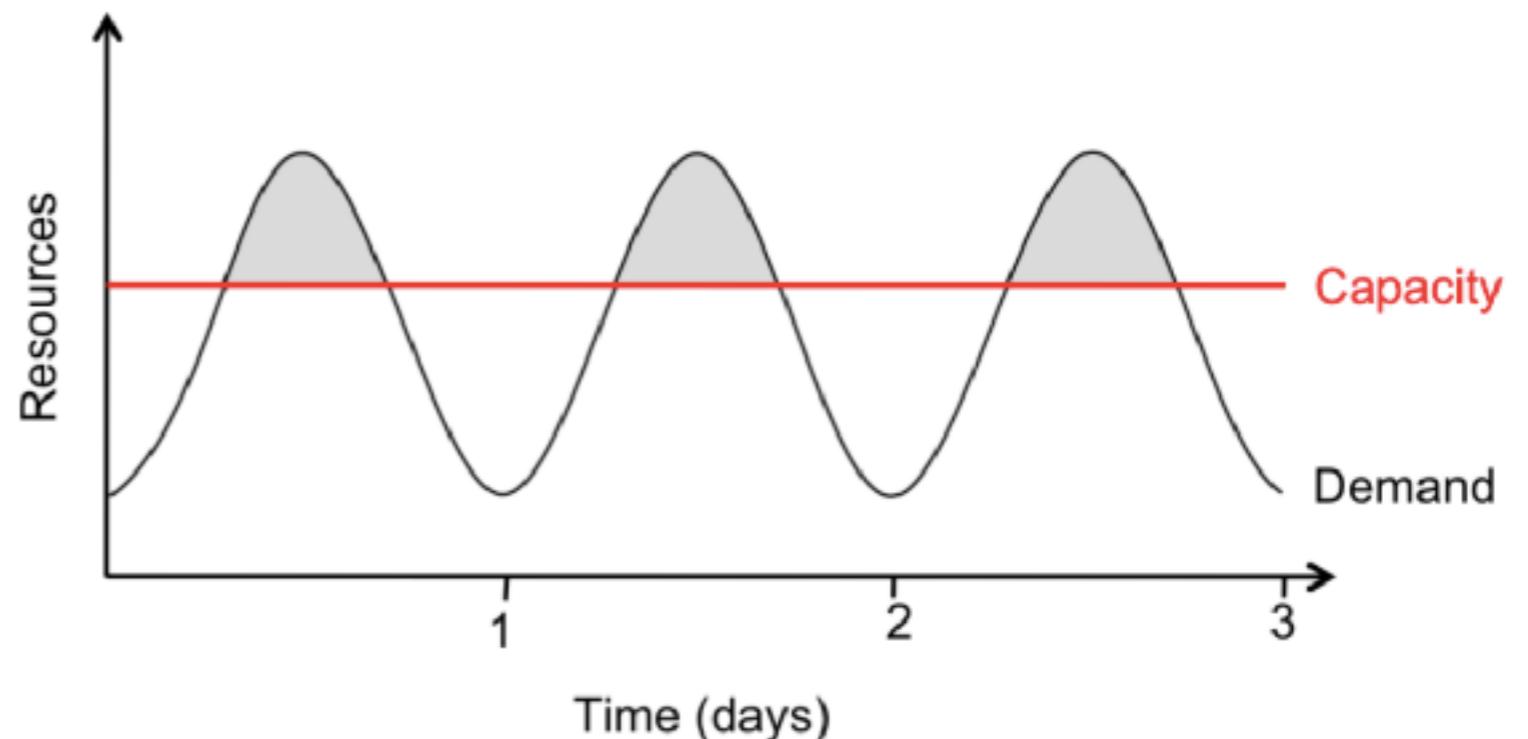


(a) Provisioning for peak load

Provisionamento pelos picos



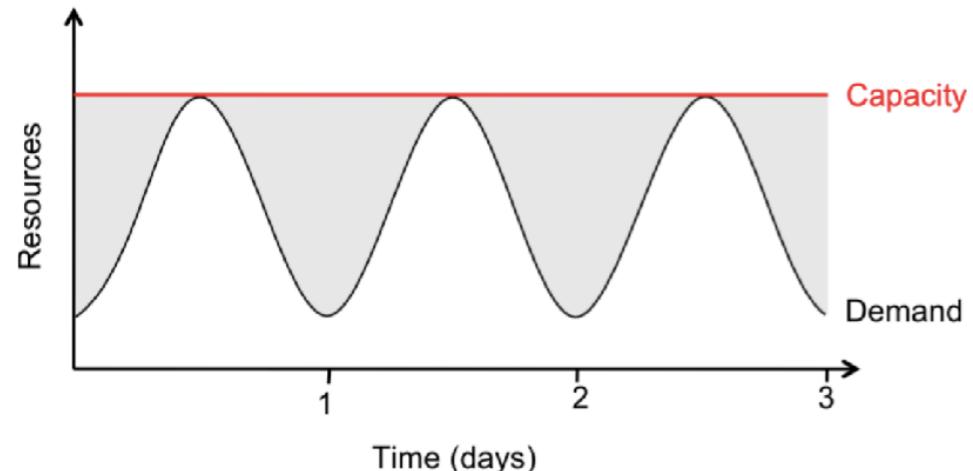
(a) Provisioning for peak load



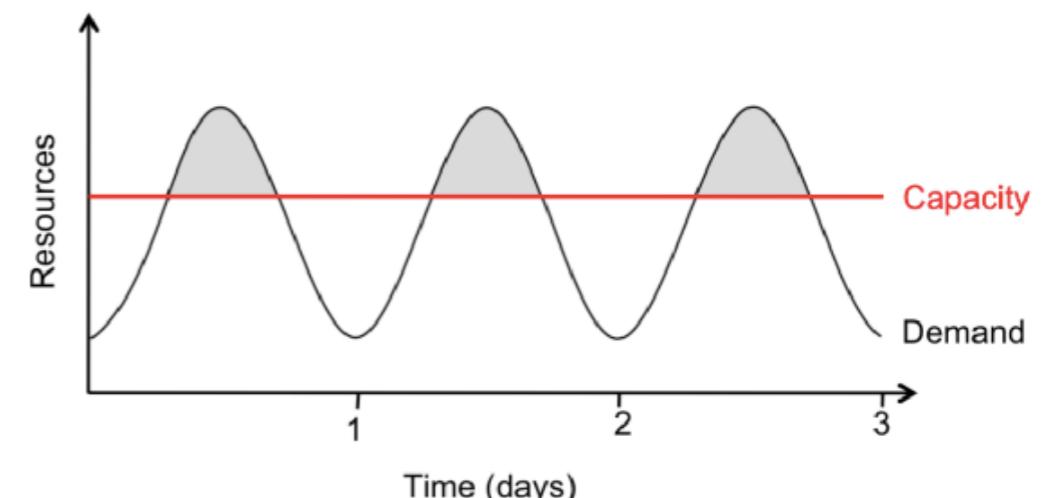
(b) Underprovisioning 1

Provisionamento pelos picos

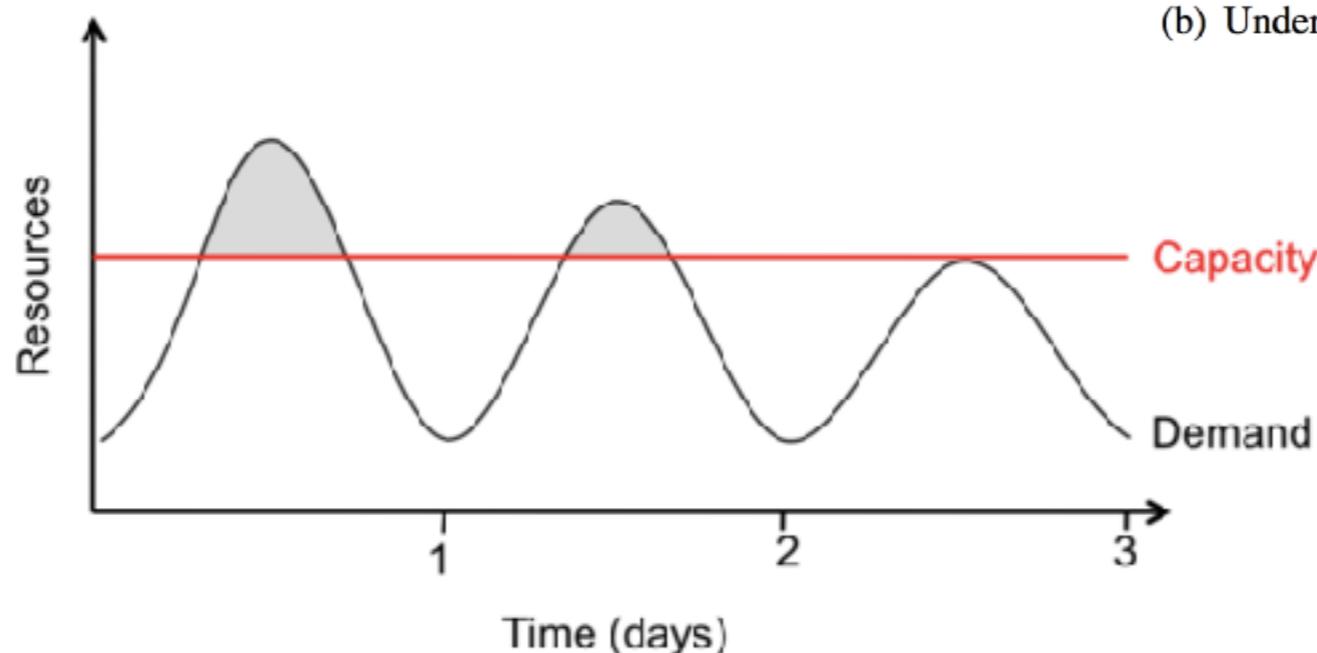
Perda de Venda



(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2

Provisionamento pelos picos

Perda da Demanda

Overprovisioning resulta em
má utilização dos recursos e
consequente **perda de dinheiro**

Underprovisioning resulta em perda de clientes e consequente prejuízo financeiro além de perda de clientes e reputação

Cases



Video Case

Easy Taxi

Laboratório

Revisão EC2

Criação de Imagens (AMIs)

Load Balance de Aplicação

Revisão EC2

Instalar Apache Server

Objetivos:

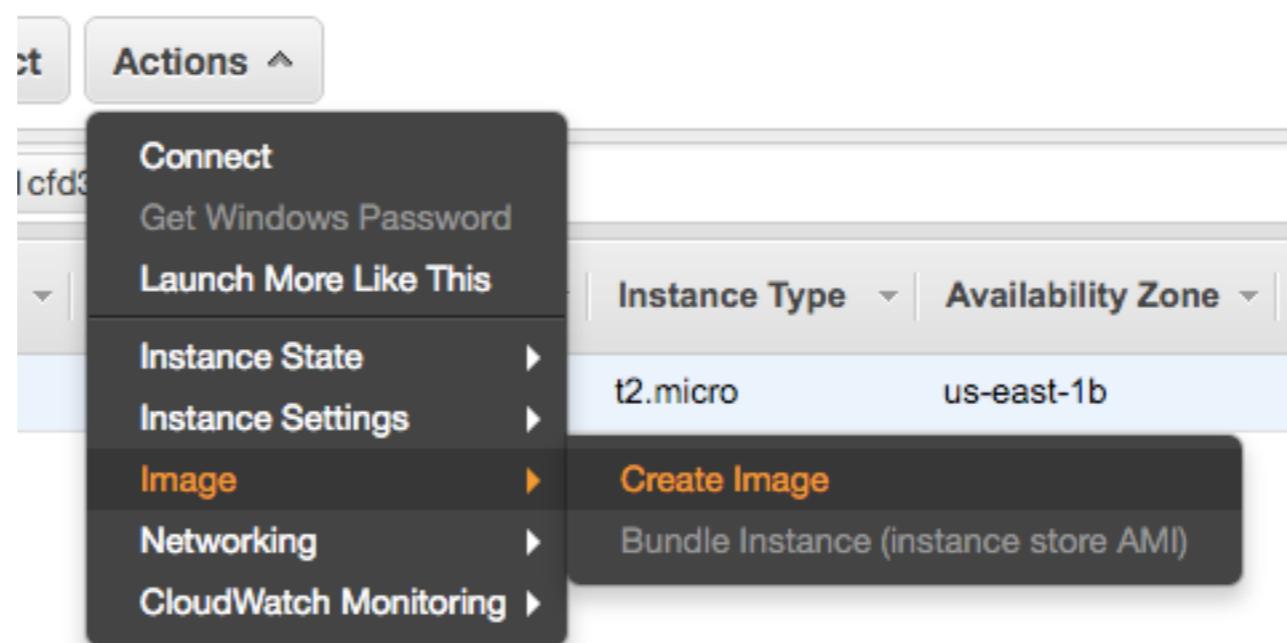
- Acessar servidor
- Instalar Apache Server
- Configurar página para apresentar o número 1 quando acessar a Home (index.html)

- ssh -i **chave.pem** ubuntu@ip
- sudo apt-get update
- sudo apt-get install apache2
- cd /var/www/html
- sudo vim index.html

Imagens AMIs

AMI

- Selecione a instância que servirá de base para a criação da imagem



AMI

- **Image Name:** Nome da instância para referência futura
- **Image Description:** Descrição da imagem (explicativo, programas instalados, propósito, etc)
- **No reboot:** Se marcado, a Amazon não irá pausar parar a máquina para a criação da imagem. Não é garantido, no entanto, a integridade dos arquivos (Não recomendo a menos que seja impossível fazer o reboot da máquina)
- **Instance Volumes:** Discos

Load Balance

Load Balance

LOAD BALANCING
Load Balancers

Step 1

- **Load Balancer name:** Nome para referenciar o LB
- **Create LB Inside:** VPC
- **Create an internal load balancer:** Se deseja criar o LB interno (apenas IP privado para acesso dentro da própria rede da Amazon) ou externo (público)
- **Enable advanced VPC configuration:** Se marcado você poderá configurar para quais sub-redes deseja ativar o LB
- **Listener Configuration:** Portas e protocolos disponíveis no LB

Load Balance

Step 2

- **Security Group**

Load Balance

Step 3

- **Health Check**
 - Configurações de ping para verificar se uma instância do平衡amento de carga está disponível
 - Ping Protocol: Protocolo (geralmente http)
 - Ping Port: Porta (geralmente 80 ou 8080)
 - Ping Path: Caminho da URL, por exemplo /index.html
 - Response Timeout: Quanto tempo aguarda uma resposta antes de dizer que foi sem sucesso
 - Health check interval: Frequência de verificação
 - Unhealthy Threshold: Quantas vezes precisa falhar para a instância ser marcada como indisponível
 - Healthy Threshold: Quantas vezes precisa ter sucesso para a instância ser marcada como disponível

Load Balance

Step 4

- **Instâncias**
 - Adicionar as instâncias que farão parte do balanceamento de carga

Load Balance

Step 5 e 6

- **Key**
 - Adicione uma chave para esse平衡ador de carga para futura identificação (opcional)
- **Review and Launch**
 - Revise os dados e initialize o load balance

Dia 3

Objetivos Gerais

-  Instanciar máquinas virtuais na nuvem
-  Configurar balanceamento de carga para aplicações Web
 - Configurar alarmes
-  Criação de Imagens customizadas (AMIs)
 - Configurar auto escalabilidade a partir de parâmetros especificados
 - Configurar banco de dados na nuvem
 - Configurar backup no S3
- **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem

Objetivos desta aula

- Conceitos
 - Virtualização
 - EC2 (Apresentação de tipos de instâncias e tamanhos)
- **Prática:**
 - Configurar alarmes
 - Configurar auto escalabilidade a partir de parâmetros especificados

Virtualização

- Máquina virtual é um computador criado por software com hardware virtual (Memória, CPU, Disco, etc)

- VMs rodam isoladas
- Fácil de modificar a capacidade (memória, disco, cpu, etc)
- S.O. e aplicativo enxergam a VM como um hardware físico
- Switch virtual para troca de informações entre VMs de uma mesma máquina hospedeira
- Hardware físico roda com carga alta e máquinas virtuais rodam com carga baixa

Motivação para Virtualizar

- Hardware é subutilizado
- Data Centers estão ficando sem espaço físico
- Preço da Energia vem aumentando
- Redução de Custos com administração de sistemas
- “Green Data Centers”

Hardware é subutilizado

- Moore's Law:
 - A cada 12-18 meses uma nova geração de chips é capaz de dobrar o poder de processamento pelo mesmo preço;
 - Ou seja, máquinas aumentam sua capacidade exponencialmente
- Data Centers possuem muitas máquinas poderosas e ociosas
 - Dados mostram que alguns usam apenas 10 a 15% da sua capacidade
- Portanto, deve haver uma melhor forma de balancear
 - Capacidade total de Processamento VS Carga
 - Ideal seria utilizar 70 a 80% e não 10 a 15%
 - Solução: Virtualização

Espaço físico nos Data Centers

- Estão abarrotados de máquinas ociosas ocupando espaço físico
- Fica difícil adicionar novos servidores para atender a novas demandas
 - Novas aplicações
 - Mais dados
 - Estima-se que necessidade de armazenamento cresce 50% por ano
- Portanto, virtualização deve consolidar servidores
 - Menos servidores físicos com maior poder de processamento

Custos com administração de sistemas

- Quanto mais servidores físicos maior o trabalho para manter
 - Instalar SO e aplicações
 - Instalar patch de SO
 - Atualizar SO e aplicações
 - Consertar defeitos em hardware
 - Segurança contra vírus, etc.
- Virtualização ajuda a reduzir os custos com administração
 - Menos máquinas
 - Facilidades para instalação de máquinas virtuais

Demo VirtualBox

Prática

- Configurar alarmes
- Configurar auto escalabilidade a partir de parâmetros especificados

Cloud Watch

(Alarms)

All AWS Services

Compute

Storage & Content Delivery

Database

Networking

Developer Tools

Management Tools

Security & Identity



CloudWatch

Amazon CloudWatch provides monitoring for resources and applications.



CloudFormation

AWS CloudFormation lets you create and update a collection of related AWS resources in a predictable fashion.



OpsWorks

AWS OpsWorks is a platform for managing the infrastructure of any application running on AWS cloud services.

Auto Scaling

(Escalabilidade Automática)

AUTO SCALING
Launch Configurations
Auto Scaling Groups

Welcome to Auto Scaling

1

You can use Auto Scaling to manage Amazon
operate a healthy group of instances, and scal
[Learn more](#)

[Create Auto Scaling group](#)

2



Step 1: Create launch configuration

First, define a template that your Auto Scaling group will use to launch instances.

You can change your group's launch configuration at any time.



Step 2: Create Auto Scaling group

Next, give your group a name and specify how many instances you want to run in it.

Your group will maintain this number of instances, and replace any that become unhealthy or impaired.

You can optionally configure your group to adjust its capacity according

Auto Scaling

Step 1

- Criação de Imagem (AMI) que será utilizada para a auto escalabilidade

Auto Scaling

Step 2

- Launch Configuration (Esquema padrão para inicialização das instâncias)
 - **AMI** (geralmente uma AMI customizada dentro de “My AMIs”)
 - **Size**: Tamanho das instâncias que serão disparadas
 - **Name**: Nome do padrão para referência futura
 - **Purchasing option**: Se marcado o request spot instance, usará esse modelo para inicializar as máquinas (modelo de compra de processamento em horários de baixa utilização”)
 - **Monitoring**: habilita monitoramento mais detalhado pelo Cloud Watch

Auto Scaling

Step 3

- Launch Configuration (Esquema padrão para inicialização das instâncias)
 - **Storage**
 - **Security Group**
 - **Review**

Auto Scaling

Step 4

- Auto Scaling Group
 - **Regras para a auto escalabilidade**
 - **Group Name:** Nome para referência futura

Auto Scaling

Step 5

- Auto Scaling Group
 - **Políticas para a auto escalabilidade**
 - **Increase Group Size**
 - Crie um alarme para disparar o auto scaling
 - Diga o que fazer caso o alarme seja disparado aumentando o número de instâncias
 - **Decrease Group Size**
 - Crie um alarme para disparar a redução do número de instâncias
 - Diga como quer reduzir as instâncias

Auto Scaling

Step 6

- Auto Scaling Group
 - **Notifications**
 - Avisos a respeito do que está sendo disparado, máquinas sendo inicializadas, máquinas sendo desligadas

Auto Scaling

Step 7

- Load Balance
 - Configurar o Load Balance juntamente com o Auto Scaling, de forma que fique transparente para os usuários a utilização dos serviços que serão escaláveis a partir das políticas configuradas

Laboratório

- Nova instância
- Instalar Java 8
- Instalar Tomcat 8
- Instalar aplicação Números Primos
- Gerar Imagem
- Configurar autoescalabilidade a partir de parâmetros de uso de CPU

Comandos Úteis

- **Java 8**
 - sudo apt-get install python-software-properties
 - sudo add-apt-repository ppa:webupd8team/java
 - sudo apt-get update
 - sudo apt-get install oracle-java8-installer
- **Tomcat 8**
 - sudo apt-get install tomcat8
- **Aplicação Números Primos**
 - sudo service tomcat8 stop
 - cd /home/ubuntu
 - wget <http://upload.enovar.biz/uni7.war>
 - sudo mv uni7.war /var/lib/tomcat8/webapps/
 - sudo service tomcat8 start
- **http://<IP>:8080/uni7/teste**
 - 1) Configurar instância padrão
 - 2) Criar Imagem (AMI) para replicação
 - 3) Configurar autoescalabilidade a partir de parâmetros de uso de CPU (> 5% de média)

Dia 4

Aula passada

- Conceitos de Virtualização
- Configuração de Alarmes
- Configuração de um ambiente auto escalável (Auto Scaling)

Objetivos Gerais

-  Instanciar máquinas virtuais na nuvem
-  Configurar balanceamento de carga para aplicações Web
-  Configurar alarmes
-  Criação de Imagens customizadas (AMIs)
-  Configurar auto escalabilidade a partir de parâmetros especificados
 - Configurar banco de dados na nuvem
 - Configurar backup no S3
 - Heroku
- **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem

Objetivos desta aula

- **Prática:**
 - Configurar um banco de dados na nuvem (Amazon RDS)
 - Utilizar o S3 como repositório para armazenamento na nuvem
 - Heroku (PaaS)

Amazon Relational Database Service (RDS)

“O Amazon RDS é um serviço de banco de dados relacional gerenciado que disponibiliza seis mecanismos de banco de dados conhecidos para você escolher”

Texto retirado do próprio site da Amazon RDS

**Amazon
Aurora**

 PostgreSQL

 MySQL

 MariaDB

ORACLE

 Microsoft®
SQL Server®

- Aplicação de patches;
- Backup;
- Recuperação;
- Detecção de falhas e reparo

- Alta disponibilidade (Multi-AZ)
- Escalabilidade (Vertical e Horizontal)
- Monitoramento

Tipo de instância	vCPU	Memória (GiB)	Otimizado para PIOPS	Desempenho de rede
Standard – Geração mais recente				
db.m4.large	2	8	Sim	Moderada
db.m4.xlarge	4	16	Sim	Alto
db.m4.2xlarge	8	32	Sim	Alto
db.m4.4xlarge	16	64	Sim	Alto
db.m4.10xlarge	40	160	Sim	10 Gigabit

Exemplos de Instâncias

Standard

Tipo de instância	vCPU	Memória (GiB)	Otimizado para PIOPS	Desempenho de rede
Otimizadas para memória – Geração atual				
db.r3.large	2	15	–	Moderada
db.r3.xlarge	4	30,5	Sim	Moderada
db.r3.2xlarge	8	61	Sim	Alto
db.r3.4xlarge	16	122	Sim	Alto
db.r3.8xlarge	32	244	–	10 Gigabit

Exemplos de Instâncias

Otimizadas para memória

Tipo de instância	vCPU	Memória (GiB)	Otimizado para PIOPS	Desempenho de rede
Microinstâncias				
db.t2.micro	1	1	–	Baixo
db.t2.small	1	2	–	Baixo
db.t2.medium	2	4	–	Moderada
db.t2.large	2	8	–	Moderada

Exemplos de Instâncias

Micro Instâncias

Prática

- Configuração de um banco de dados Postgres na RDS
- Configurar o backup do Postgres no S3
- Heroku

Prática 1

Instanciar e testar a utilização de um PostgreSQL no RDS

“Dica para o laboratório: Utilizar a porta 8080 como sendo a porta do postgres”

1

All AWS Services

Compute

Storage & Content Delivery

Database

Networking

Developer Tools

2

RDS

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale familiar relational databases in the cloud.

Ela

Amazon ElastiCache application allowing you to store and retrieve information in memory using an in-memory caching system.

DynamoDB

Amazon DynamoDB is a scalable, highly available, fully managed NoSQL database service.

Redis

Amazon Redis Cache

RDS Dashboard

Instances

CloudWatch Metrics

Launch DB Instance

S

Filter: All Instances

Select Engine

To get started, choose a DB Engine below and click Select.



PostgreSQL

Select

PostgreSQL is a powerful, open-source object-relational database system with a strong reputation of reliability, stability, and correctness.

- High reliability and stability in a variety of workloads.
- Advanced features to perform in high-volume environments.
- Vibrant open-source community that releases new features multiple times per year.
- Supports multiple extensions that add even more functionality to the database.
- The most Oracle-compatible open-source database.

Do you plan to use this database for production purposes?

Production

- PostgreSQL

Use [Multi-AZ Deployment](#) and [Provisioned IOPS Storage](#) as defaults for high availability and fast, consistent performance.

Para ambientes de produção. Permite Multi-AZ Deployment, que significa alta disponibilidade

Dev/Test

- PostgreSQL

This instance is intended for use outside of production or under the [RDS Free Usage Tier](#).

Para ambientes de teste e Free

[Cancel](#)

[Previous](#)

[Next Step](#)

Instance Specifications

DB Engine	postgres
License Model	postgresql-license
DB Engine Version	9.3.1
DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	No
Storage Type	General Purpose (SSD)
Allocated Storage*	10 GB

Escolher a versão 9.3.1 para que seja facilitada a nossa próxima atividade de backup e armazenamento no S3

Settings

DB Instance Identifier*	fa7posgrad
Master Username*	fa7posgrad
Master Password*
Confirm Password*



This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the [new certificate bundle](#). Learn more [here](#)

VPC*	Default VPC (vpc-05a1b661)
Subnet Group	default
Publicly Accessible	Yes
Availability Zone	No Preference
VPC Security Group(s)	<ul style="list-style-type: none">Create new Security Groupdefault (VPC)launch-wizard-1 (VPC)

Database Options

Database Name	db_fa7
Database Port	5432
DB Parameter Group	default.postgres9.3
Option Group	default:postgres-9-3
Copy Tags To Snapshots	<input type="checkbox"/>
Enable Encryption	No

Backup

Backup Retention Period days

Backup Window

Monitoring

Enable Enhanced Monitoring

Maintenance

Auto Minor Version Upgrade

Maintenance Window

Launch and Test it!!!

- Download PgAdmin - <http://www.pgadmin.org/>
- Configurar conexão
 - Name: Nome da sua conexão
 - Host: Endereço do seu banco de dados no RDS
 - Port: 5432 (porta padrão do postgres)
 - Username:
 - Password

Endpoint: fa7posgrad.co9wjkjgefl.us-east-1.rds.amazonaws.com:5432 (authorized) i

Publicly accessible

The screenshot shows the database structure for 'db_fa7'. The tree view includes:

- Databases (3)
 - db_fa7 (selected)
 - Catalogs (2)
 - Event Triggers (0)
 - Extensions (1)
 - Schemas (1)
 - public
 - Slony Replication (0)
 - Tablespaces (2)
 - Group Roles (2)
 - Login Roles (2)
- postgres
- rdsadmin

Endpoint para acessar o banco (host)

```

WITH OWNER = rdsadmin
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'en_US.UTF-8'
LC_CTYPE = 'en_US.UTF-8'
CONNECTION LIMIT = -1;
  
```

Prática

✓ Configuração de um banco de dados Postgres na RDS

- Configurar o backup do Postgres no S3
- Heroku

S3

(Simple Storage Service)

All AWS Services

Compute

Storage & Content Delivery

Database



S3

Amazon Simple Storage Service (S3) can be used to store and retrieve any amount of data.



Ama
(Am
serv
Cor
insta

Nível de uso gratuito da AWS*

Como parte do Nível de uso gratuito da AWS, você pode começar a utilizar o Amazon S3 gratuitamente. Após a inscrição, novos clientes da AWS recebem 5 GB de armazenamento padrão do Amazon S3, 20.000 solicitações de Get, 2.000 solicitações de Put, 15 GB de transferência de dados para fora a cada mês, por um ano.

Região:

Leste dos EUA (Norte da Virgínia)



Armazenamento padrão

Primeiro 1 TB/mês \$0.0300 por GB

Próximos 49 TB/mês \$0.0295 por GB

Próximos 450 TB/mês \$0.0290 por GB

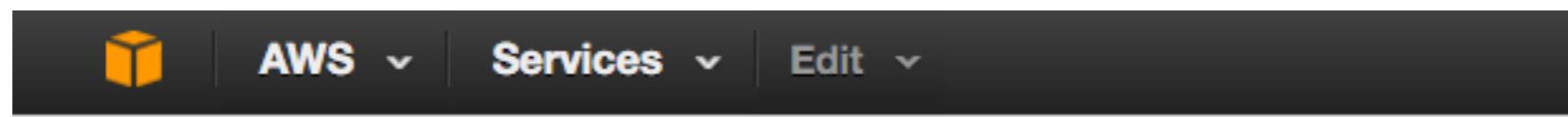
Próximos 500 TB/mês \$0.0285 por GB

Próximos 4.000 TB/mês \$0.0280 por GB

Mais de 5.000 TB/mês \$0.0275 por GB

Pelo que você será cobrado?

- Espaço
- Solicitações
- Transferência (entre diferentes regiões da AWS e para a Internet)



Buckets são “caixas” com nomes **únicos** dentro de toda a Amazon

Create a Bucket - Select a Bucket Name and Region Cancel

A bucket is a container for objects stored in Amazon S3. When creating a bucket, you can choose a Region to optimize for latency, minimize costs, or address regulatory requirements. For more information regarding bucket naming conventions, please visit the [Amazon S3 documentation](#).

Bucket Name: fa7aula4

Region: US Standard

Dica: Utilize o nome da sua organização como prefixo. (Ex: posgradfa7.aula)

Set Up Logging > **Create** **Cancel**

S3 on EC2

Security Credentials

1 Roberto Costa Lima Jr ▾

- My Account
- Billing & Cost Management
- Security Credentials
- Sign Out

2 + Access Keys (Access Key ID and Secret Access Key)

3 Create New Access Key

Create Access Key

Your access key (access key ID and secret access key) has been created successfully.

Download your key file now, which contains your new access key ID and secret access key. Once you download the key file now, you will not be able to retrieve your secret access key again.

To help protect your security, store your secret access key securely and do not share it with anyone.

▼ Hide Access Key

4 Access Key ID: AKIAIYSSDNV75TPNLDQQ
Secret Access Key: 2v56LGKD5Djj3KQEkYvGiDM0FRHjCg0a9Wp866L

Download Key File Close

- Instalar AWS Cli
 - **sudo apt-get install awscli**
- AWS Configure Credentials
 - **aws configure**
- Copy file
 - **aws s3 cp teste.txt s3://posgradfa7.aula --region us-east-1**

Principais Comandos

- Copy
 - cp
- Move
 - mv
- Delete
 - rm

Regiões

Region Name	Region
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1
EU (Frankfurt)	eu-central-1
Asia Pacific (Tokyo)	ap-northeast-1

Prática 2

Criar um backup do Postgres **RDS** em uma máquina do **EC2** e Salvar no **S3**

BACKUP POSTGRES

- sudo apt-get update
- sudo apt-get install pgadmin3
- pg_dump -h {HOST} -p 5432 -Ft -d {BANCO} -U {USUÁRIO} -W > {ARQUIVODESTINO}
- aws s3 cp teste.txt s3://posgradfa7.aula --region us-east-1

Prática

-  Configuração de um banco de dados Postgres na RDS
-  Configurar o backup do Postgres no S3
 - Heroku

Heroku

Comandos Úteis

- **Java 8**

<http://upload.enovar.biz/Heroku.txt>

- sudo apt-get install python-software-properties
- sudo add-apt-repository ppa:webupd8team/java
- sudo apt-get update
- sudo apt-get install oracle-java8-installer

- **Maven**

- sudo apt-get install maven

- **Heroku**

- sudo add-apt-repository "deb https://cli-assets.heroku.com/branches/stable/apt ./"
- curl -L https://cli-assets.heroku.com/apt/release.key | sudo apt-key add -
- sudo apt-get update
- sudo apt-get install heroku

Dia 5

Aula passada

- Configurar um banco de dados na nuvem (Amazon RDS)
- Utilizar o S3 como repositório para armazenamento na nuvem
- Heroku (PaaS)

Objetivos Gerais

-  Instanciar máquinas virtuais na nuvem
-  Configurar balanceamento de carga para aplicações Web
-  Configurar alarmes
-  Criação de Imagens customizadas (AMIs)
-  Configurar auto escalabilidade a partir de parâmetros especificados
-  Configurar banco de dados na nuvem
-  Configurar backup no S3
-  Heroku
- **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem
- **Bônus2:** ElasticBeanstalk

Objetivos desta aula

- **Apresentação e Prática:**
 - Elastic Beanstalk (PaaS)
 - API's diversas em ambientes de nuvem
 - Video sobre o CloudFormation

Elastic Beanstalk



O AWS Elastic Beanstalk é um serviço de fácil utilização para implantação e escalabilidade de aplicações e serviços da web desenvolvidos com **Java, .NET, PHP, Node.js, Python, Ruby, Go e Docker** em servidores familiares como Apache, Nginx, Passenger e IIS.

* Definição do próprio site do fabricante

Não há custos adicionais pelo Elastic Beanstalk, você só paga pelos recursos da AWS necessários para executar e armazenar seus aplicativos.

* Retirado do próprio site do fabricante

Video Demo

Elastic Beanstalk

Prática

All AWS Services

Compute

Storage & Content Delivery

Database



EC2

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.



Elastic Beanstalk

AWS Elastic Beanstalk is an application container for deploying and managing applications.

<https://dl.dropboxusercontent.com/u/11502680/cloud.war>

1)

Application Information

To create a new application, enter the details of your application.

Application name: Must be less than 100 characters and cannot contain a /

Description: Optional.

[Cancel](#) [Next](#)

2)

Web Server Environment

Provides resources for an AWS Elastic Beanstalk web server in either a single instance or [load-balancing, auto scaling](#) environment. [Learn more](#)

[Create web server](#)

3)

Environment Type

Choose the platform and type of environment to launch.

Predefined configuration:

Looking for a different platform? [Let us know.](#)

AWS Elastic Beanstalk will create an environment running Tomcat 8 Java 8 on 64bit Amazon Linux 2017.09 v2.7.0. [Change platform version.](#)

Environment type:

[Learn more](#)

1)

Deployment Preferences

Elastic Beanstalk will update your application in batches so as to avoid downtime when deploying.

Deployment policy: **Rolling** [Learn more](#)

Healthy threshold: **Ok**

Ignore health check: **False**

Batch size:

- Percentage
30 % of the fleet at a time
- Fixed
1 instances at a time



Sobre a implantação

- Contínua
- Completa
- Mista

2) Environment Information

Enter your environment information.

Environment name: **uni7cloud-env**

Environment URL: **uni7cloud-env.us-east-1.elasticbeanstalk.com**

Check availability

Description:

Optional: 200 character maximum

Ambiente

1)

Additional Resources

Select additional resources for this environment.

- Create an RDS DB Instance with this environment [Learn more](#)
- Create this environment inside a VPC [Learn more](#)

2)

Configuration Details

Modify the following settings or click Next to accept the default configuration. [Learn more](#)

Instance type: **t2.micro**

Determines the processing power of the servers in your environment.

EC2 key pair:

uni7-memcached

[Refresh](#)

Optional: Enables remote login to your instances.

Email address:

roberto@enovar.com.br

Optional: Get notified about any major changes to your environment.

Application health check

URL:

/home

Enter the relative URL that ELB continually monitors to ensure your application is available.

Rolling updates type:

Rolling based on Health

[Learn more](#)

Cross zone load
balancing:

Enables load balancing across multiple Availability Zones. [Learn more](#)

Connection draining:

Enables the load balancer to maintain connections to an Amazon EC2 instance to complete in-progress requests while also stopping new requests. [Learn more](#)

Connection draining
timeout:

20 seconds

Maximum time that the load balancer maintains connections to an Amazon EC2 instance before forcibly closing connections.

1)

Additional Resources

Select additional resources for this environment.

- Create an RDS DB Instance with this environment [Learn more](#)
- Create this environment inside a VPC [Learn more](#)

upload.enovar.biz/numerosprimos.war

2) VPC Configuration

Select the VPC to use when creating your environment. [Learn more](#)

VPC: [Refresh](#)

Associate Public IP Address

Select different subnets for ELB and EC2 instances in your Availability Zone.

AZ	Subnet	ELB	EC2
us-east-1a	subnet-4b3f1403 (172.30.0.0/24)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
us-east-1b	subnet-ce234a94 (172.30.1.0/24)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
us-east-1c	subnet-fd18ea99 (172.30.2.0/24)	<input type="checkbox"/>	<input type="checkbox"/>
us-east-1d	subnet-cd6d70f1 (172.30.3.0/24)	<input type="checkbox"/>	<input type="checkbox"/>
us-east-1e	subnet-eed7a1c2 (172.30.4.0/24)	<input type="checkbox"/>	<input type="checkbox"/>
us-east-1f	subnet-5a89f566 (172.30.5.0/24)	<input type="checkbox"/>	<input type="checkbox"/>

VPC security group: [Refresh](#)

ELB visibility: [Refresh](#)

Select Internal when load balancing a back-end service that should not be publicly available.

API's

<http://upload.enovar.biz/compruvem.pdf>

<http://upload.enovar.biz/avaliacaoRestaurante.zip>

Vídeo Demo do Cloud Formation

Dia 6

Objetivos Gerais

-  Instanciar máquinas virtuais na nuvem
-  Configurar balanceamento de carga para aplicações Web
-  Configurar alarmes
-  Criação de Imagens customizadas (AMIs)
-  Configurar auto escalabilidade a partir de parâmetros especificados
-  Configurar banco de dados na nuvem
-  Configurar backup no S3
-  Heroku
-  **Bônus:** API's (SaaS) para alavancarem as aplicações utilizando a computação em nuvem
-  **Bônus2:** ElasticBeanstalk

Objetivos desta aula

Trabalho

Obrigado!!!
roberto@enovar.com.br