



Prefácio

José Papo

AWS América Latina

MANOEL VERAS

# Arquitetura de Nuvem

AMAZON WEB SERVICES (AWS)

Explica a arquitetura AWS

Relaciona os modelos IaaS e PaaS com a AWS

Descreve os principais serviços e produtos da AWS

Ensina a montar a arquitetura e construir o DATACENTER com recursos da AWS

Apresenta o caso AWS Peixe Urbano



# Arquitetura de Nuvem

AMAZON WEB SERVICES (AWS)

MANOEL VERAS

# Arquitetura de Nuvem

AMAZON WEB SERVICES (AWS)

Prefácio:

José Papo  
AWS América Latina



Copyright© 2013 por Brasport Livros e Multimídia Ltda.

Todos os direitos reservados. Nenhuma parte deste livro poderá ser reproduzida, sob qualquer meio, especialmente em fotocópia (xerox), sem a permissão, por escrito, da Editora.

Editor: Sergio Martins de Oliveira

Diretora: Rosa Maria Oliveira de Queiroz

Gerente de Produção Editorial: Marina dos Anjos Martins de Oliveira

Revisão: Maria Inês Galvão

Editoração Eletrônica: Abreu's System Ltda.

Capa: Paulo Vermelho

Produção de ebook: S2Books

Técnica e muita atenção foram empregadas na produção deste livro. Porém, erros de digitação e/ou impressão podem ocorrer. Qualquer dúvida, inclusive de conceito, solicitamos enviar mensagem para [brasport@brasport.com.br](mailto:brasport@brasport.com.br), para que nossa equipe, juntamente com o autor, possa esclarecer. A Brasport e o(s) autor(es) não assumem qualquer responsabilidade por eventuais danos ou perdas a pessoas ou bens, originados do uso deste livro.

**BRASPORT Livros e Multimídia Ltda.**

Rua Pardal Mallet, 23 – Tijuca

20270-280 Rio de Janeiro-RJ

Tels. Fax: (21) 2568.1415/2568.1507

e-mails: [brasport@brasport.com.br](mailto:brasport@brasport.com.br)

[vendas@brasport.com.br](mailto:vendas@brasport.com.br)

[editorial@brasport.com.br](mailto:editorial@brasport.com.br)

site: [www.brasport.com.br](http://www.brasport.com.br)

**Filial**

Av. Paulista, 807 – conj. 915

01311-1))0 – São Paulo-SP

Tel. Fax (11): 3287.1752

e-mail: [mailto:filialsps@brasport.com.br](mailto:mailto:filialsps@brasport.com.br)

**Para Marcus e Anália.**  
**In memoriam.**

# Sumário

[Capa](#)

[Folha de Rosto](#)

[Dedicatória](#)

[Agradecimentos](#)

[Notas do Autor](#)

[Prefácio](#)

[Introdução](#)

[Objetivos](#)

[Estrutura](#)

## [PARTE I – ASPECTOS BÁSICOS](#)

### [1. Visão Geral](#)

[1.1. Introdução](#)

[1.2. Evolução](#)

[1.3. SOA, web services e API](#)

[1.4. APIs SOAP, REST e QUERY](#)

[1.5. NIST e AWS](#)

[1.6. Estrutura](#)

[1.7. Imagens e instâncias](#)

[1.8. Marketplace](#)

[1.9. Partner Network](#)

[1.10. Referências bibliográficas](#)

### [2. Infraestrutura](#)

[2.1. Introdução](#)

[2.2. Componentes](#)

[2.3. Regiões e zonas de disponibilidade](#)

[2.4. Serviços por região](#)

[2.5. Suporte AWS](#)

[2.6. Referências bibliográficas](#)

### [3. Identidade e Acesso](#)

[3.1. Introdução](#)

[3.2. Conceitos](#)

[3.3. Identidade e acesso tradicional \*versus\* identidade e acesso na AWS](#)

[3.4. Credenciais de segurança](#)

[3.5. IAM](#)

[3.6. Referências bibliográficas](#)

### [4. Precificação e Faturamento](#)

[4.1. Introdução](#)

[4.2. Precificação](#)

[4.3. Faturamento](#)

[4.4. Referências bibliográficas](#)

## [PARTE II – SERVIÇOS DE INFRAESTRUTURA](#)

### [5. Computação](#)

[5.1. Introdução](#)

[5.2. Elastic Compute Cloud \(EC2\)](#)

[5.3. Elastic Load Balancing \(ELB\)](#)

[5.4. Elastic MapReduce \(EMR\)](#)

[5.5. Referências bibliográficas](#)

### [6. Armazenamento](#)

[6.1. Introdução](#)

[6.2. Armazenamento tradicional \*versus\* armazenamento na AWS](#)

[6.3. Elastic Block Store \(EBS\)](#)

[6.4. Simple Storage Service \(S3\)](#)

[6.5. Opções de armazenamento](#)

[6.6. Storage Gateway \(SG\)](#)

[6.7. Import/Export \(I/E\)](#)

[6.8. Glacier](#)

## [6.9. Referências bibliográficas](#)

## [7. Rede](#)

### [7.1. Introdução](#)

### [7.2. Rede tradicional \*\*versus\*\* rede na AWS](#)

### [7.3. Virtual Private Connect \(VPC\)](#)

### [7.4. Route 53](#)

### [7.5. Direct Connect \(DC\)](#)

### [7.6. Opções de conectividade com a VPC](#)

### [7.7. Referências bibliográficas](#)

## [8. Banco de Dados](#)

### [8.1. Introdução](#)

### [8.2. Bancos de dados tradicionais \*\*versus\*\* bancos de dados na AWS](#)

### [8.3. RDS \(Relational Database Service\)](#)

### [8.4. ElastiCache](#)

### [8.5. Referências bibliográficas](#)

## [9. Gerenciamento](#)

### [9.1. Introdução](#)

### [9.2. Gerenciamento tradicional \*\*versus\*\* gerenciamento na AWS](#)

### [9.3. CloudWatch](#)

### [9.4. Referências bibliográficas](#)

## [PARTE III – ASPECTOS AVANÇADOS](#)

## [10. Arquitetura](#)

### [10.1. Introdução](#)

### [10.2. Conceito](#)

### [10.3. Arquitetura tradicional \*\*versus\*\* arquitetura AWS](#)

### [10.4. Melhores práticas com AWS](#)

### [10.5. Elasticidade com Auto Scaling](#)

### [10.6. Aplicações web](#)

### [10.7. Aplicações empresariais](#)

### [10.8. Migração para a AWS](#)

## 10.9. Referências bibliográficas

## 11. Governança

### 11.1. Introdução

### 11.2. Governança tradicional e governança com AWS

### 11.3. SLAs tradicionais e SLAs com a AWS

### 11.4. Questões específicas de conformidade

### 11.5. Referências bibliográficas

## 12. Segurança

### 12.1. Introdução

### 12.2. Conceitos

### 12.3. Segurança tradicional *versus* segurança na AWS

### 12.4. Segurança em domínios na AWS

### 12.5. Segurança da plataforma Microsoft na AWS

### 12.6. Referências bibliográficas

## 13. Continuidade

### 13.1. Introdução

### 13.2. Fundamentos

### 13.3. Práticas tradicionais *versus* práticas com a AWS

### 13.4. Cenários AWS

### 13.5. PCN e PRD

### 13.6. Referências bibliográficas

## PARTE IV – CASO NACIONAL

## 14. Caso Peixe Urbano

### 14.1. Introdução

### 14.2. Site

### 14.3. Desafios

### 14.4. Benefícios

### 14.5. Startups

### 14.6. Ágil

### 14.7. Startups enxutas (lean startups)

[14.8. Infraestrutura](#)

[14.9. Arquitetura](#)

[14.10. Referências bibliográficas](#)

*The movement to the cloud is a one-way street.*

Vivek Kundra, ex-CIO do governo americano

# Agradecimentos

Agradeço a todos que me apoiam e me ajudam a prosseguir. Este livro aborda a principal arquitetura de nuvem disponível hoje, a Amazon Web Services (AWS). A Amazon Web Services LCC, do ponto de vista organizacional, é uma subsidiária da Amazon.com, Inc. A Amazon é uma *startup* de internet cujo principal negócio é o varejo eletrônico. A AWS tecnicamente é caracterizada principalmente como Infraestrutura como Serviço (IaaS), um dos modelos de serviço de nuvem.

A AWS adotou o conceito de Cloud Computing na sua essência, mantendo a simplicidade do ambiente para o desenvolvedor. Através de diversas interfaces é possível gerenciar todo o sistema. Pode-se dizer que a AWS permite programar o DATACENTER sem exagero.

Agradeço ao José Papo e ao Jeff Barr da Amazon pelas contribuições dadas ao livro.

Agradeço ao Sérgio Martins e à Rosa Queiroz da Brasport pelo apoio de sempre.

**Dr. Manoel Veras.**

# Notas do Autor

Este livro pretende ser uma contribuição para a formação de profissionais de TI no Brasil. Boa parte do material utilizado como referência foi gerado pela própria Amazon e publicado em seu site. Parabenizo a Amazon pela qualidade de seus artigos e manuais. Meu trabalho foi o de articular uma visão e criar uma referência para o entendimento da arquitetura sem necessariamente ter que compará-la com produtos da concorrência ou esgotar os aspectos técnicos. Em alguns casos e funcionalidades abordadas, tento criar uma visão mais simples da arquitetura AWS para que você, caro leitor, entenda rapidamente o poder deste sistema.

O foco deste livro é a arquitetura da infraestrutura da AWS. Eu mostro como construir o DATACENTER com uma plataforma de nuvem, fazendo naturalmente uma equivalência entre o DATACENTER convencional e o DATACENTER baseado na nuvem. São grandes as diferenças, mas a principal é a abstração permitida pela introdução da camada de web services no DATACENTER baseado na nuvem.

## **Para quem é este livro?**

Se você for um aluno universitário ou de escola técnica matriculado em cursos de ciência da computação, engenharia de sistemas ou mesmo tecnologia da informação e entender a essência deste livro, estará se preparando para lidar com a arquitetura de TI que predominará nos próximos anos.

Se você for um possível cliente desta plataforma, poderá adquirir uma visão do estágio desta nova arquitetura e influenciar decisões sobre a sua utilização na sua própria organização.

Se você está do lado de um provedor, mesmo que não seja a Amazon, poderá aqui verificar o estágio atual desta tecnologia, constatando a excelência da AWS.

Se você for um profissional de TI do tipo CIO, CTO, Dev, DevOps, OPs e está interessado em crescer, entender a mudança e se manter no mercado, estará se preparando para os desafios no futuro.

É importante ressaltar que por diversas vezes consultei a Wikipédia nas suas versões em inglês e português. Ela é uma referência para diversos conceitos encontrados nesta obra.

# Prefácio

Estamos vivenciando uma revolução na Tecnologia da Informação. Essa revolução está mudando não somente os departamentos de TI das organizações, como também está gerando grandes mudanças socioculturais em nosso planeta. Essa revolução está baseada em algumas inovações de ruptura: smartphones, tablets, redes sociais, e-commerce, Big Data e a computação em nuvem. Podemos dizer que a computação em nuvem é a fundação que permitiu o crescimento e a inovação presente nas outras tecnologias de ruptura.

A computação em nuvem (que começou a ser oferecida de forma pioneira pela Amazon através da Amazon Web Services em 2006) permitiu o surgimento de centenas de milhares de apps para smartphones e tablets, apps para redes sociais, aplicações web e até mesmo novas redes sociais de cunho específico (como o Pinterest, uma rede social que está 100% na nuvem da Amazon). Empresas como Netflix, Pinterest, Peixe Urbano, Foursquare, Flipboard, Instagram, Airbnb, Dropbox, Slideshare, Órama, portal Terra, Chaordic, entre tantas outras, nasceram já suportadas pela computação em nuvem da AWS.

Grandes corporações e agências de governo já notaram os benefícios da computação em nuvem e estabeleceram estratégias – empresas como Shell, Unilever, Nasdaq, Samsung, NASA, SEGA, Amazon.com, The New York Times, Grupo Pão de Açúcar, Gol Linhas Aéreas, Sul América Seguros e muitas outras. Segundo o Instituto de Pesquisas Gartner, a computação em nuvem já representa um mercado de 109 bilhões de dólares em 2012 e estima-se que chegará a 206 bilhões de dólares em 2016. Já está crescendo a um ritmo de praticamente 20% ao ano.

Mas por que a computação em nuvem cresce tanto no mundo? Quais são os seus benefícios? Por que ela revolucionou nossas estruturas sociais, culturais e o modo como trabalhamos com os recursos computacionais? Podemos responder essas perguntas a partir do entendimento das suas características básicas: elasticidade, pagamento apenas pelos recursos usados, infraestrutura de autosserviço e APIs/automação. Elas permitem que as organizações tenham DATACENTERS automatizados e gastem somente aquilo que utilizam. Diminuem os custos de capital em TI e os transformam em custos operacionais.

Outro ponto fundamental é que a computação em nuvem facilita a inversão de uma tendência descrita pelo Gartner: a maioria das organizações está gastando 80% do seu investimento e tempo de TI com manutenção e

sustentação de projetos e DATACENTERS e não com projetos e soluções inovadoras. A computação em nuvem permite às organizações diminuir o seu custo total de propriedade com infraestrutura de TI. Além disso, a nuvem aumenta a agilidade e diminui os riscos de desenvolvimento e implantação de projetos inovadores. Com a nuvem é possível mudar o foco dos investimentos de TI para mais inovação e mais foco no negócio (o “core business”) da organização. Assim é possível transformar o departamento de Tecnologia em um centro de resultados e inovação, e não simplesmente ser mais um centro de custos. Segundo um estudo do IDC, em cinco anos de utilização da nuvem o TCO pode chegar a 70% de economia em comparação a um ambiente dentro de casa. Além disso, o estudo concluiu que a TI aumenta a agilidade e a produtividade em 52% e reduz o tempo de parada de sistemas em até 72%.

No Brasil e na América Latina estamos dando os primeiros passos na nuvem. A vinda da Amazon Web Services ao Brasil em 2011, inclusive com uma região contendo DATACENTERS, acelerou muito a adoção da nuvem. A AWS é atualmente considerada a líder em serviços de cloud na modalidade IaaS pelo Gartner, por conta de seus preços baixos e do número de soluções, funcionalidades, recursos e segurança que oferece. Com sua chegada em 2011, notamos o crescimento acelerado com que a América Latina está aproveitando os benefícios da nuvem.

Por todos os motivos citados é que acredito ser providencial o lançamento do livro do Professor Doutor Manoel Veras. Um livro prático e detalhado sobre a nuvem da Amazon, contendo uma análise de aspectos básicos e avançados dos serviços da AWS. Com seu livro, o professor Veras ajuda na divulgação do conhecimento sobre a computação em nuvem na língua portuguesa. Presta, desse modo, uma contribuição valiosa para a Tecnologia da Informação em nosso país.

Acredito que a TI e a computação em nuvem podem revolucionar mercados e indústrias no Brasil, ajudando na sua competitividade. E a Amazon Web Services está presente no país para ajudar empresas, órgãos de governo e pessoas a crescerem e a prosperarem através da tecnologia. Por isso, leia o livro do professor Manoel Veras e conheça em detalhes todas as soluções disponíveis na AWS. O mercado de computação em nuvem precisa de pessoas altamente qualificadas para acelerar a adoção de Tecnologia da Informação. E enquanto estiver lendo o livro, mantenha-se sempre conectado às novidades da nuvem através do blog oficial da AWS em português (<http://aws.typepad.com/brasil>) e do site da AWS (<http://aws.amazon.com/pt>).

## José Papo

Evangelista Técnico da AWS para a América Latina

# Introdução

Este livro foi escrito com foco em descrever a arquitetura Amazon Web Services (AWS). A AWS é um sistema fantástico. Seu sucesso é fruto de um trabalho árduo de cientistas, engenheiros e analistas comandados por Werner Vogels, CTO e vice-presidente da Amazon, iniciado em 2006. A Amazon tem o mérito de ter acreditado cedo no conceito de cloud computing e a sua atuação no varejo on-line testou e aprimorou o modelo. Gerenciar e operar uma infraestrutura de TI de um varejo baseado na internet com operação em diferentes partes do globo trouxe à Amazon um grande know-how e lastreou sua operação como provedor de serviços de nuvem. Sem dúvida, garantir desempenho, disponibilidade e segurança em ambientes globais é aspecto-chave da cloud computing e é tratado muito bem pela AWS.

Hoje empresas de pesquisas focadas em TI como o Gartner<sup>[1]</sup> colocam a AWS como uma plataforma líder de IaaS (*Infrastructure as a Service*), um dos modelos de serviço de cloud computing.

## Objetivos

Os principais objetivos deste livro são:

- Explicar como funciona a arquitetura AWS.
- Ajudar a formar arquitetos de nuvem.
- Ajudar a formar mão de obra qualificada em TI no Brasil.

## Estrutura

O livro é dividido em quatro partes e quatorze capítulos e é escrito em uma sequência lógica que facilita o entendimento dos diversos conceitos e a estrutura da AWS.

As partes do livro, os grandes assuntos tratados e os capítulos incluídos em cada uma das partes são descritos na **Tabela 0-1**.

**Tabela 0-1 Estrutura do livro**

Partes	Assuntos	Capítulos
<b>Parte I</b>	Aspectos Básicos	Capítulos 1, 2, 3 e 4.
<b>Parte II</b>	Serviços de Infraestrutura	Capítulos 5, 6, 7, 8 e 9.

<b>Parte III</b>	Aspectos Avançados	Capítulos 10, 11, 12 e 13.
<b>Parte IV</b>	Caso Nacional	Capítulo 14.

A Tabela 0-2 descreve os títulos dos treze capítulos.

**Tabela 0-2 Descrição dos Capítulos**

<b>Capítulos</b>	<b>Descrição</b>
<b>Capítulo 1</b>	Visão Geral
<b>Capítulo 2</b>	Infraestrutura
<b>Capítulo 3</b>	Identidade e Acesso
<b>Capítulo 4</b>	Precificação e Faturamento
<b>Capítulo 5</b>	Computação
<b>Capítulo 6</b>	Armazenamento
<b>Capítulo 7</b>	Rede
<b>Capítulo 8</b>	Banco de Dados
<b>Capítulo 9</b>	Gerenciamento
<b>Capítulo 10</b>	Arquitetura
<b>Capítulo 11</b>	Governança
<b>Capítulo 12</b>	Segurança
<b>Capítulo 13</b>	Continuidade
<b>Capítulo 14</b>	Caso Peixe Urbano

Neste livro optou-se por manter o termo em inglês cloud computing e não utilizar os termos equivalentes em português: computação de nuvem ou computação nas nuvens. Também optou se por utilizar o termo web services e não serviços web ou serviços da web.

Utilizam-se aplicativo e aplicação como sinônimos. Utiliza-se também o conceito de sistema de informação como um conjunto de aplicativos ou de aplicações construído com uma determinada finalidade. Deste ponto de vista,

pode-se considerar a AWS também um sistema. Neste livro quase sempre a AWS é tratada como uma arquitetura de tecnologia da informação.

Utiliza-se carga de trabalho para o termo workload. Carga de trabalho ou workload é a carga do aplicativo a ser processada pelo ambiente computacional.

Ressalta-se que normalmente, quando não se faz uma citação específica, o livro refere-se à arquitetura AWS e aos conceitos de nuvem pública e Infraestrutura como um Serviço (IaaS).

A **Tabela 0-3** sugere o significado de siglas utilizadas no livro. Estas siglas são utilizadas pela Amazon para descrever seus web services.

**Tabela 0-3 Significado das siglas da AWS**

Sigla AWS	Significado
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
ELB	Elastic Load Balancing
EMR	Elastic MapReduce
IAM	Identity and Access Management
RDS	Relational Database Service
S3	Simple Storage Service
SES	Simple E-mail Service
SNS	Simple Notification Service
SQS	Simple Queue Service
SWF	Simple Workflow Service
VPC	Virtual Private Cloud

# ***PARTE I – ASPECTOS BÁSICOS***

---

# 1. Visão Geral

## 1.1. Introdução

Cloud computing ou Computação em Nuvem pode ser considerada uma nova arquitetura de TI, uma evolução natural dos modelos baseados em mainframe e cliente/servidor.

Com o modelo cloud computing muda-se a forma de adquirir a TI. Agora paga-se pelo uso dos recursos em um modelo que se adequa à demanda, o que se convencionou chamar de elasticidade. A **Tabela 1-1** ilustra a evolução da arquitetura de TI em termos de tecnologia e economia.

**Tabela 1-1 Evolução da arquitetura de TI**

	<b>Tecnologia</b>	<b>Economia</b>
<b>Mainframe</b>	Utiliza computação centralizada.	Otimizada para eficiência por causa do alto custo.
<b>Cliente/servidor</b>	Utiliza computação distribuída.	Otimizada para agilidade devido ao baixo custo.
<b>Cloud computing</b>	Utiliza grandes DATACENTERS.	Otimizada para eficiência e agilidade. Pode reduzir custos e riscos.

Observa-se que o modelo cloud computing é um avanço natural dos modelos anteriores e busca essencialmente eficiência e agilidade.

O livro “Cloud Computing: Nova Arquitetura de TI”, de minha autoria, reforça os conceitos relacionados à nuvem e trata dos tipos de serviços oferecidos por provedores de nuvem, dos riscos e das oportunidades.

Amazon Web Services (AWS) é a principal oferta de arquitetura do tipo cloud computing da atualidade. Esta arquitetura permite às empresas o acesso a serviços de infraestrutura na forma *on demand*. A ideia é que esta forma de aquisição de serviços de infraestrutura de TI, conhecida como serviços de infraestrutura de nuvem, reduza custos, minimize os riscos do negócio e maximize as oportunidades. Do ponto de vista teórico, o serviço tradicional de DATACENTER, quando comparado ao serviço de nuvem, é pouco eficaz, considerando que o uso e a capacidade dos recursos não podem ser otimizados como no modelo baseado em nuvem.

As empresas podem, ao utilizar a arquitetura AWS:

- Continuar a utilizar aplicações existentes baseadas em sistemas operacionais, banco de dados, arquiteturas de software e linguagens de programação já comuns. Estas aplicações devem ser movidas para a nuvem AWS.
- Desenvolver novas aplicações que desde o início aproveitam os recursos da nuvem AWS e que podem até misturar-se com arquiteturas “legadas” para servir a diferentes modelos de negócio.

A proposta da Amazon para a AWS é fornecer serviços baseados em nuvem com flexibilidade, efetividade, escalabilidade, elasticidade e segurança.

O IDC[\[2\]](#) recentemente relatou a experiência de onze empresas de médio e pequeno porte com a utilização da arquitetura AWS. Os resultados obtidos por elas apontam para uma redução de custos, aumento de produtividade em relação à situação de manter a infraestrutura de TI interna, redução do *downtime* e ganhos expressivos de produtividade.

Este capítulo introduz a arquitetura AWS e conceitos fundamentais para o entendimento dos próximos capítulos.

## 1.2. Evolução

A Amazon investiu muitos recursos no desenvolvimento da AWS. O desenvolvimento do sistema começou em 1995, com as demandas do próprio site de vendas da Amazon, uma das maiores plataformas de varejo on-line do mundo.

O negócio AWS iniciou em 2006, ano em que o novo modelo veio a público. Desde então as funcionalidades da AWS são ampliadas praticamente a cada semana. Impressiona a velocidade imposta no desenvolvimento e na otimização do sistema. Estes avanços normalmente são notificados nos blogs da própria AWS. A **Figura 1-1** ilustra o avanço das funcionalidades da AWS de 2006 até os dias de hoje.

Os números não mentem sobre o crescimento da AWS: em junho de 2012 a AWS já armazenava um trilhão de objetos[\[3\]](#) no seu sistema de armazenamento Amazon Simple Storage Service (Amazon S3).

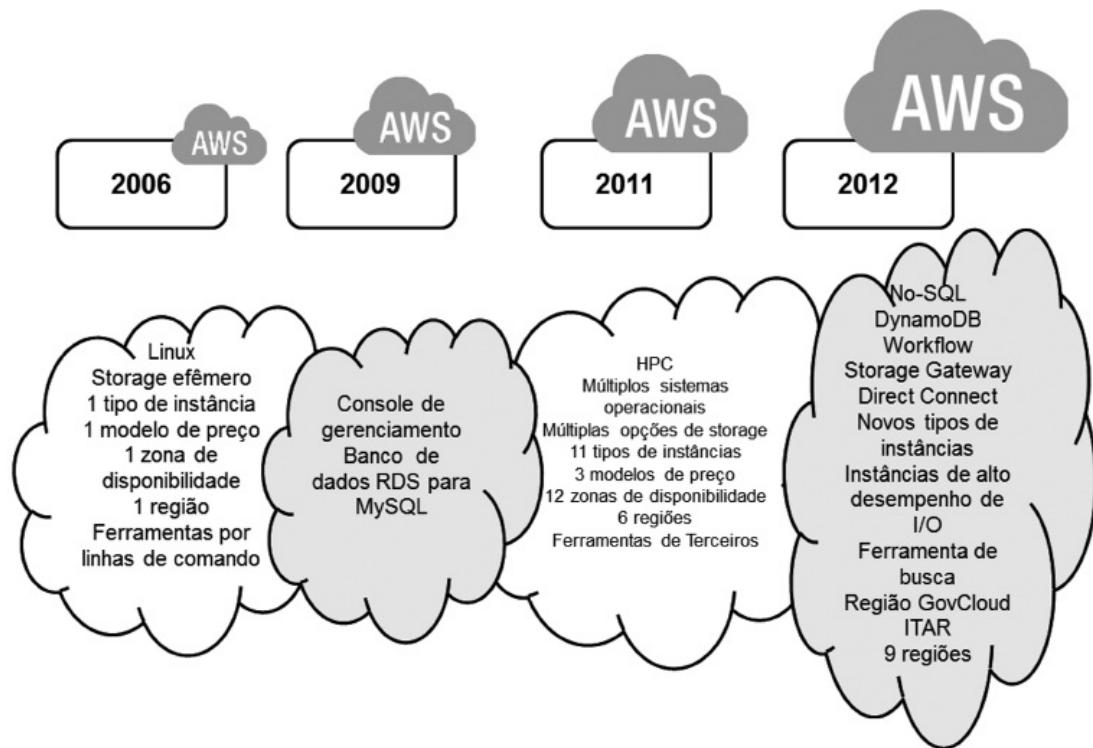


Figura 1-1 Evolução da AWS

### 1.3. SOA, web services e API

Três conceitos são fundamentais neste livro. São eles:

- **Arquitetura orientada a serviços (Service-Oriented Architecture – SOA)** é um estilo de arquitetura de software cujo princípio fundamental prega que as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços. A arquitetura SOA é baseada nos princípios da computação distribuída e utiliza o paradigma *request/reply* para estabelecer a comunicação entre os sistemas clientes e os sistemas que implementam os serviços. SOA independe de qualquer plataforma de tecnologia, mas na prática web services é a plataforma mais associada à realização desta forma de arquitetura.
- **Serviços web (web services)** são blocos funcionais utilizados na construção de aplicativos acessíveis através de protocolos padrões de internet independentemente das plataformas e linguagens de programação. Web services podem implementar a SOA. Neste caso os serviços são conectados através de um barramento de serviços (*enterprise service bus*) que disponibiliza interfaces acessíveis. A arquitetura web services é composta por:
  - **Contrato de serviço técnico:** similar a uma interface de programação de aplicativos (*Application Programming Interface – API*).

- **Corpo da lógica de programação:** pode ser desenvolvida de maneira personalizada para o web service ou pode existir como lógica legada empacotada por um web service. A segunda opção permite disponibilizar a funcionalidade via padrões de comunicação web services.
- **Lógica de processamento de mensagens:** combinação de “parsers”, processadores e agentes de serviço. Parsers transformam texto na entrada em estrutura de dados.
- **API** é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem se envolver em detalhes da sua implementação, mas apenas usar seus serviços. De modo geral, a API é composta por uma série de funções acessíveis somente por programação e que permite utilizar características do software menos evidentes para um desenvolvedor tradicional.

Web API é uma API no contexto de desenvolvimento web; neste caso, a API trata de um conjunto definido de mensagens de requisição e resposta HTTP, geralmente expressado nos formatos XML ou JSON.

XML (*eXtensible Markup Language*) é uma recomendação da W3C para intercâmbio de dados computacionais. É um dos subtipos da Linguagem Padronizada de Marcação Genérica (*Standard Generalized Markup Language – SGML*), capaz de descrever diversos tipos de dados. Seu propósito principal é facilitar o intercâmbio de dados através da internet.

JSON (*JavaScript Object Notation*) é um formato para intercâmbio de dados computacionais. JSON é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript. A simplicidade de JSON tem resultado em seu uso especialmente como uma alternativa para o XML. Uma das vantagens do JSON sobre o XML como um formato para intercâmbio de dados é o fato de ser mais simples escrever um analisador JSON.

## 1.4. APIs SOAP, REST e QUERY

A AWS é baseada em web services: todos eles são controlados por HTTP com diferentes “níveis de abstração” — ou diferentes APIs. Alguns serviços podem ser controlados via API REST, alguns por meio da API SOAP e alguns por meio da API QUERY. Alguns serviços utilizam um mix dessas três APIs.

A Tabela 1-2 relaciona web services AWS e APIs correspondentes.

Tabela 1-2 APIs para web services AWS

Web services	REST	SOAP	QUERY
EC2		X	X

S3	X	X	
Elastic MapReduce			X

- A API REST é uma interface HTTP. Para usar a API REST pode-se utilizar qualquer kit de ferramentas que ofereça suporte a HTTP. Pode-se usar um navegador para buscar objetos, contanto que eles sejam anonimamente legíveis. Deve-se utilizar a API REST para os cabeçalhos HTTP padrão e códigos de status, para que os toolkits e navegadores funcionem conforme o esperado.
- A API SOAP é uma interface que usa uma codificação literal de documento. O caminho mais comum para usar SOAP é fazer download do WSDL (em <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl>) e usar um toolkit SOAP, como Apache Axis ou Microsoft.NET, para criar as ligações e, em seguida, escrever código que use as ligações para chamar o S3, por exemplo.
- Sobre a API QUERY pode-se dizer que são solicitações (*requests*) que dependem de parâmetros, nomes simples e pares de valor para expressar a ação que o serviço irá executar e os dados onde a ação será executada. Quando se usa uma interface QUERY, o envelope HTTP serve apenas como uma forma de entregar esses parâmetros para o serviço.

Sobre utilizar SOAP ou REST pode-se afirmar que REST é mais elegante que SOAP, pois utiliza ao máximo o protocolo HTTP, evitando a construção de protocolos adicionais. REST tem o potencial de ser bem mais simples que uma implementação com SOAP e tende a ser mais performático. SOAP, por outro lado, é um padrão maduro – qualquer ferramenta de integração, ou mesmo um framework encontrado no mercado, normalmente possui funcionalidades para manipular as mensagens que seguem o padrão SOAP.

## 1.5. NIST e AWS

### 1.5.1. Introdução

A definição do NIST (*National Institute of Standards and Technology*), órgão pertencente ao Departamento de Comércio americano, para cloud computing pode ser representada pela **Figura 1-2**. A parte tracejada na figura corresponde ao posicionamento da AWS com relação ao modelo NIST.

O conceito do NIST trata das características essenciais da nuvem, das modalidades de serviço e das formas de implementação e é o conceito mais aceito na atualidade para cloud computing.

A AWS pode ser classificada como uma arquitetura IaaS com algumas opções de PaaS e quase sempre como nuvem pública. Ela também possui diversas funcionalidades de nuvem híbrida e de nuvem privada, funcionando como uma rede privada virtual e até de nuvem comunitária com a opção exclusiva para o governo americano (AWS Gov Cloud).

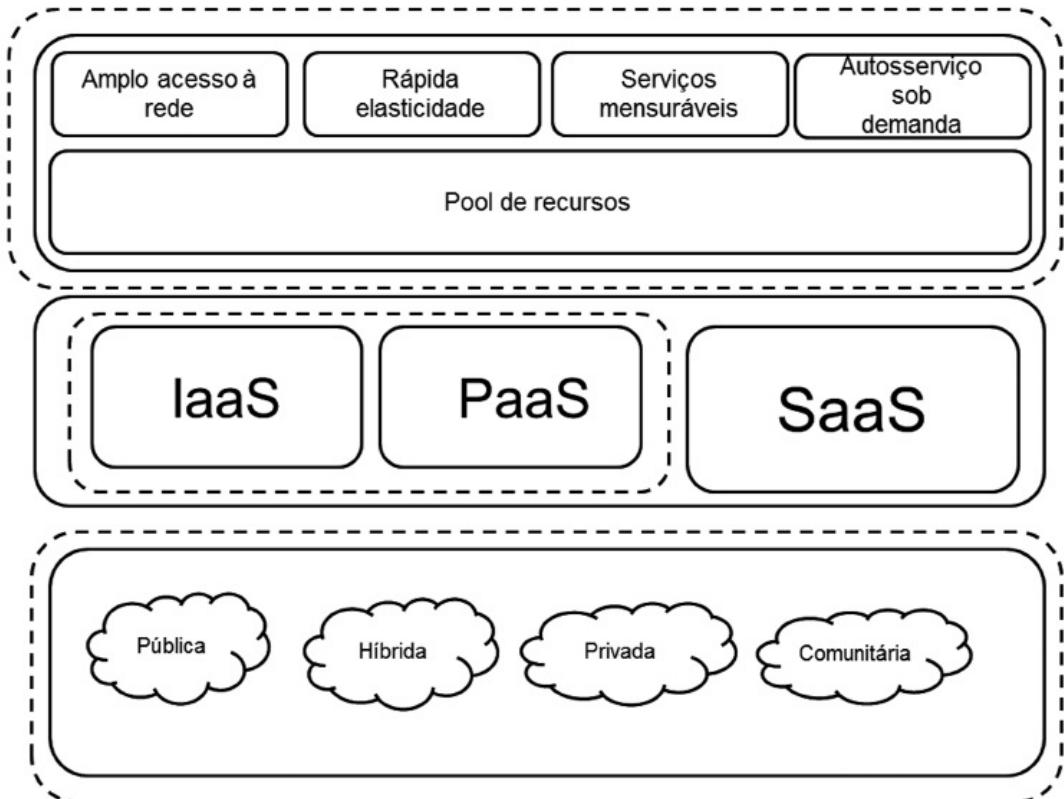


Figura 1-2 Características essenciais da cloud computing e AWS

Outras ofertas estão surgindo com base na AWS. O Eucalyptus, por exemplo, é uma plataforma de nuvem compatível com as APIs da Amazon. Usuários de nuvem podem mover instâncias entre uma nuvem privada de Eucalyptus e a nuvem pública da Amazon para criar uma nuvem híbrida. Ele aproveita a virtualização do sistema operacional para obter isolamento entre os aplicativos. Ele também pode ser utilizado com nuvem privada, abstraindo os recursos de computação, armazenamento e rede para oferecer serviços do tipo IaaS.

### 1.5.2. Características essenciais

Segundo o NIST, um modelo de cloud computing deve apresentar algumas características essenciais:

- **Autoserviço sob demanda:** funcionalidades computacionais são providas automaticamente sem a interação humana com o provedor de serviço.
- **Amplo acesso à rede:** recursos computacionais estão disponíveis através da internet e são acessados via mecanismos padronizados,

para que possam ser utilizados por dispositivos móveis e portáteis, computadores, etc.

- **Pool de recursos:** recursos computacionais (físicos ou virtuais) do provedor são utilizados para servir a múltiplos usuários, sendo alocados e realocados dinamicamente conforme a demanda do usuário.
  - Multitenancy é a essência da característica “pool de recursos”. Multitenancy na camada de infraestrutura é a própria virtualização e na camada de aplicação é a utilização de uma arquitetura que serve a múltiplos inquilinos (tenants) e, portanto, otimiza o uso dos recursos.
- **Rápida elasticidade:** prega que as funcionalidades computacionais devem ser rápidas e elasticamente providas, assim como rapidamente liberadas. O usuário dos recursos deve ter a impressão de que possui recursos ilimitados que podem ser adquiridos (comprados) em qualquer quantidade e a qualquer momento.

Elasticidade é uma propriedade fundamental da nuvem. A elasticidade é o poder para dimensionar recursos computacionais diminuindo ou expandindo-os facilmente e com o mínimo de atrito. É importante compreender que a elasticidade acabará por propiciar a maioria dos benefícios da nuvem.

Elasticidade também pode ser a somatória dos esforços para obter:

- Escalabilidade linear.
- Utilização *on demand*.
- Pagamento conforme o uso.

Rápida elasticidade impõe rapidez a estes esforços. A natureza elástica e rápida da abordagem de nuvem deve permitir que a infraestrutura de TI possa ser alinhada com a demanda real, aumentando assim a utilização, reduzindo os custos e naturalmente beneficiando o negócio.

A **Figura 1.3** ilustra curvas de vendas e de TI *versus* tempo que traduz uma situação ideal. Observe que a curva de TI acompanha a curva de vendas. A proposta da cloud computing é possibilitar este alinhamento otimizando o uso dos recursos.

Jinesh Varia, consultor da AWS, sugere no artigo “Projetando para a nuvem: práticas recomendadas”, publicado pela AWS com versão em português, que, se você for um arquiteto de nuvem, você precisará internalizar este novo conceito na sua organização e fazer com que os desenvolvedores trabalhem na arquitetura do aplicativo a fim de aproveitar os benefícios da nuvem.

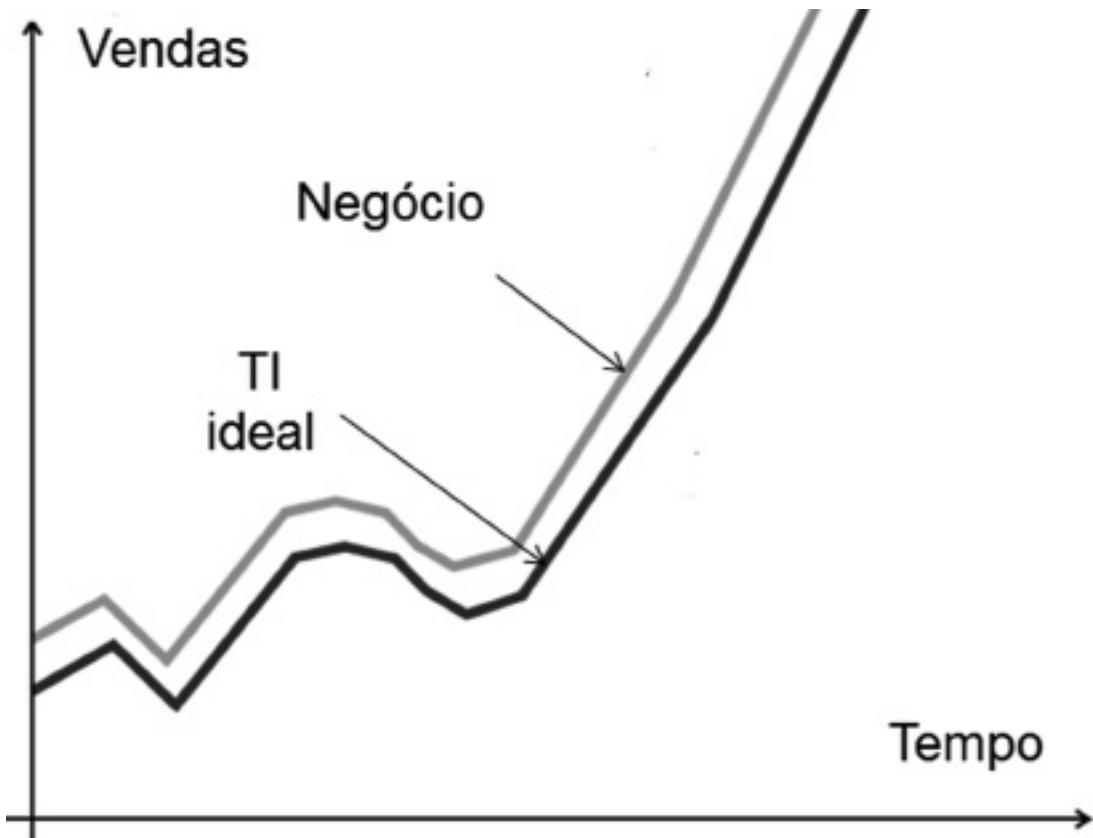


Figura 1-3 Negócio e TI

- **Serviços mensuráveis:** os sistemas de gerenciamento utilizados pela cloud computing controlam e monitoram automaticamente os recursos para cada tipo de serviço. Esse monitoramento do uso dos recursos deve ser transparente para o provedor de serviços, assim como para o consumidor do serviço utilizado.

### 1.5.3. Modelos de serviço

Existem três principais modelos de serviços para cloud computing:

- **Infraestrutura como um serviço (*Infrastructure as a Service – IaaS*):** é a capacidade que o provedor tem de oferecer uma infraestrutura de processamento e armazenamento de forma transparente e representa uma abstração da infraestrutura propriamente dita. Neste cenário, o usuário não tem o controle da infraestrutura física, mas, através de mecanismos de virtualização, possui controle sobre as máquinas virtuais, o armazenamento, os aplicativos instalados e algum controle limitado sobre os recursos de rede.
- **Plataforma como um serviço (*Platform as a Service – PaaS*):** são capacidades oferecidas pelo provedor para o desenvolvedor de aplicativos. Aplicativos estes que serão executados e disponibilizados na nuvem. A plataforma na nuvem oferece um modelo de computação, armazenamento e comunicação para os aplicativos.
- **Software como um serviço (*Software as a Service – SaaS*):** são aplicativos de interesse para uma grande quantidade de usuários que passam a ser hospedados na nuvem como uma alternativa ao processamento local. Os aplicativos são oferecidos como serviços por provedores e acessados pelos clientes através de aplicações como o browser. Todo o controle e gerenciamento da rede, sistemas operacionais, servidores e armazenamento é feito pelo provedor de serviço.

### 1.5.4. Modelos de implantação

Existem quatro principais modelos de implantação para cloud computing descritos a seguir:

- **Nuvem privada:** comprehende uma infraestrutura de cloud computing operada e quase sempre gerenciada pela organização cliente. Os serviços são oferecidos para serem utilizados pela própria organização, não estando publicamente disponíveis para uso geral. O Gartner alerta que a nuvem privada é definida por privacidade, não propriedade, localização ou responsabilidade de gestão.
- **Nuvem pública:** é disponibilizada publicamente através do modelo pague-por-uso. São oferecidas por organizações públicas ou por grandes grupos industriais que possuem grande capacidade de processamento e armazenamento.
- **Nuvem comunitária:** neste caso a infraestrutura de cloud computing é compartilhada por diversas organizações e suporta uma comunidade que possui interesses comuns. A nuvem comunitária pode ser administrada pelas organizações que fazem parte da

comunidade ou por terceiros e pode existir tanto fora como dentro das organizações.

- **Nuvem híbrida:** a infraestrutura é uma composição de duas ou mais nuvens (privadas, públicas ou comunitárias) que continuam a ser entidades únicas, porém conectadas através de tecnologias proprietárias ou padronizadas que propiciam a portabilidade de dados e aplicações. A nuvem híbrida impõe uma coordenação adicional a ser realizada para uso das nuvens privadas e públicas com impactos na governança.

A **Figura 1-4** mostra o tamanho do mercado mundial para serviços de nuvem pública de 2009 até 2015<sup>[4]</sup>.

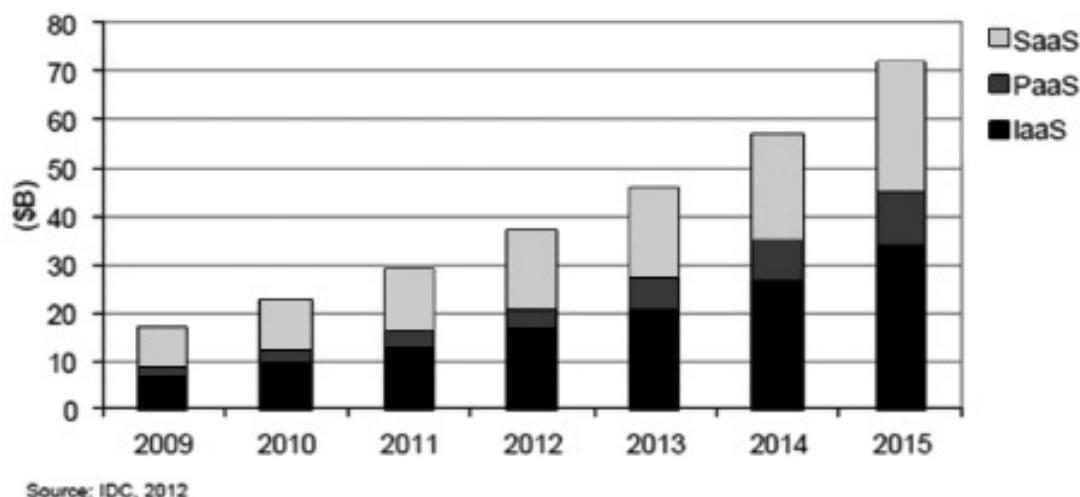


Figura 1-4 Mercado global para nuvem pública até 2015

### 1.5.5. Arquiteturas de referência

Diversos modelos estão surgindo e tentando criar uma referência conceitual de arquitetura. Entre eles destacam-se:

- NIST Cloud Computing Reference Architecture
- IBM Cloud Reference Architecture
- Cisco Cloud Reference Architecture e Framework
- Cloud Reference Model da CSA

O modelo NIST, mais aceito atualmente, define cinco principais atores na nuvem: provedor, cliente, broker, auditor e carrier.

### 1.5.6. Aderência

Uma forma de validar o modelo AWS é verificar sua aderência ao modelo de referência cloud computing sugerido pelo NIST.

A AWS oferece amplo acesso a serviços de rede com sua oferta global

de infraestrutura incluindo serviços DNS e diversos DATACENTERS ao redor do mundo. Os recursos básicos e avançados são ofertados através de uma camada de virtualização que possibilita a utilização dos recursos em pool.

A elasticidade é oferecida através de instâncias virtuais, sistemas de balanceamento de carga e sistemas de monitoramento que permitem o acréscimo e a redução de instâncias. O sistema de mensageria também permite a construção de aplicações escaláveis na nuvem. Os serviços também são mensuráveis na AWS.

A especificação da AWS é baseada na utilização das instâncias, no armazenamento e no tráfego de dados. A ideia do autosserviço sob demanda também é a essência do modelo AWS, e a sua oferta para aquisição de instâncias torna o autoatendimento uma realidade.

A capacidade finita da nuvem AWS, um aspecto importante, fez com que surgisse uma oferta com diferentes tipos de instâncias (servidores virtuais) que permite a reserva de capacidade e também a aquisição em preços de mercado.

A AWS também tem avançado com o lançamento de funcionalidades que a tornam também uma plataforma como um serviço (PaaS). Também nota-se um grande investimento em serviços de apoio à nuvem híbrida com ênfase em sistemas de armazenamento que permitam fazer a replicação de dados entre sistemas locais e a nuvem AWS e propiciam a consequente continuidade dos negócios.

Considerando o serviço IaaS, pode-se afirmar que a arquitetura AWS possui boa aderência à definição do NIST.

## 1.6. Estrutura

### 1.6.1. Classificação dos serviços

Para efeito didático, a estrutura da AWS pode ser dividida em três grandes partes:

- Infraestrutura global de suporte para a plataforma. Aqui incluem-se as regiões, zonas de disponibilidade, localizações de conteúdo e DNS.
- Serviços de infraestrutura que são os serviços de processamento, armazenamento, rede, banco de dados e gerenciamento. Serviços de infraestrutura de TI, mas com outro nível de abstração e acesso baseado em web services.
- Serviços básicos de plataforma que estão no limite do serviço IaaS e acabam por completar a oferta de serviços incluindo mensageria, *workflow* e gerenciamento dos aplicativos.

A Figura 1-5 ilustra a relação entre processos de negócio, aplicações e a estrutura da AWS.

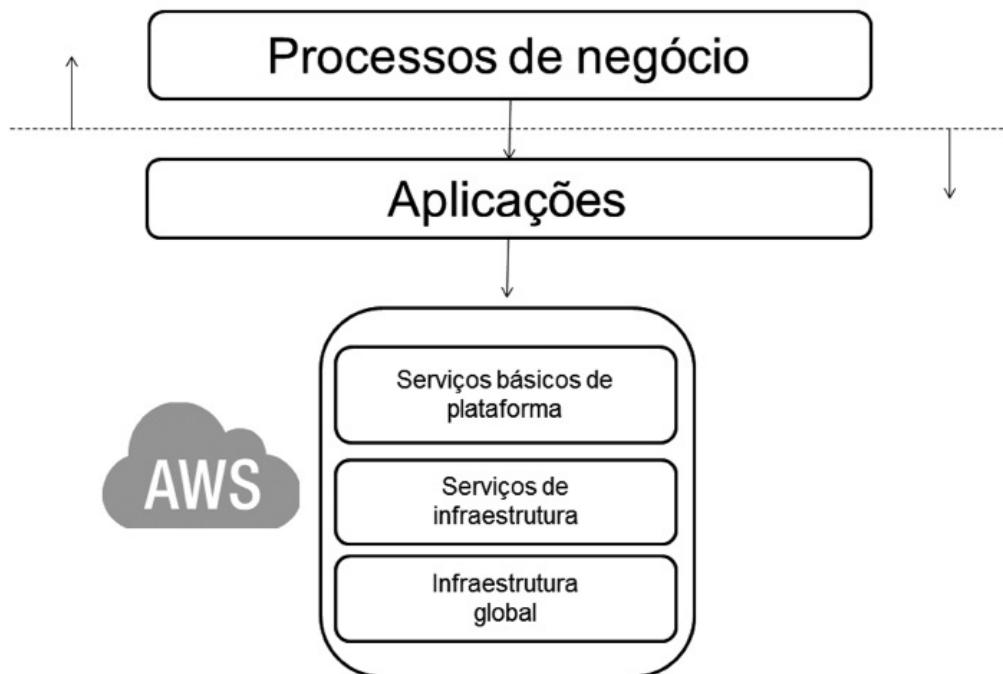


Figura 1-5 Posicionamento da estrutura da AWS

A **Figura 1-6** detalha os principais serviços contidos na estrutura AWS. Essa estrutura é a base do livro.

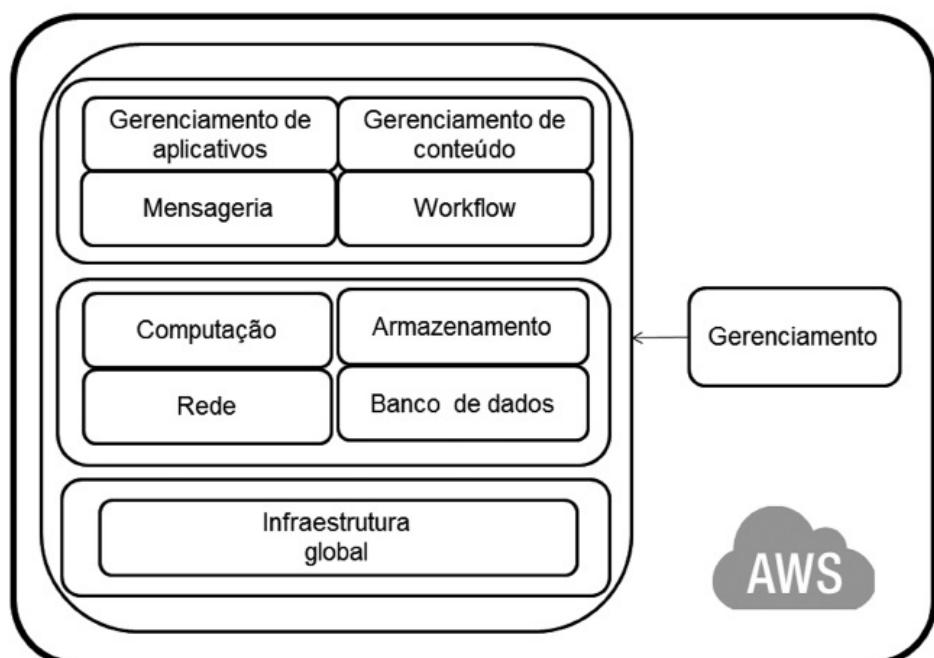


Figura 1-6 Estrutura detalhada da Amazon AWS

- **Infraestrutura global:** os elementos centrais da infraestrutura são as regiões, as zonas de disponibilidade e os pontos de presença de conteúdo e de DNS (*Domain Name System*).
- **Serviços de infraestrutura:** são fundamentalmente os serviços de processamento, armazenamento, rede, banco de dados e gerenciamento que representam a infraestrutura central de TI do DATACENTER e são fornecidos *on demand*. No modelo AWS pode-se contratar estes serviços de acordo com a necessidade. Um servidor virtual, uma instância virtual, por exemplo, pode ser contratado com diferentes níveis de processamento e diferentes tipos de armazenamento local. Os serviços de armazenamento na AWS podem ser utilizados mediante a necessidade, sem precisar antecipar a capacidade. Os elementos centrais dos serviços de infraestrutura da AWS que fornecem os blocos de construção mais comuns necessários às aplicações são resumidos a seguir:

- **Computação.**

- O Amazon Elastic Compute Cloud (EC2) é o servidor virtual AWS e fornece a capacidade de processamento necessária para a aplicação. Permite também, utilizando ferramentas associadas, aumentar ou diminuir recursos de computação baseando-se na demanda.
- O Amazon Elastic MapReduce (EMR) permite realizar o processamento de grandes quantidades de dados.

- **Armazenamento.**

- É possível armazenar qualquer tipo de dado que o aplicativo necessite no *Simple Storage Service* (S3) ou no *Elastic Block Store* (EBS) e tirar vantagem do armazenamento escalável, confiável, altamente disponível e de baixo custo. O AWS Glacier oferece uma opção baseada em web services para backup e archive.

- **Rede.**

- A Amazon Virtual Private Cloud (VPC) permite definir uma topologia de rede virtual para os recursos EC2 que aumenta a segurança da rede.
- O Amazon Route 53 é um web service e uma opção ao DNS fornecido por provedores e operadoras de telecomunicações. A proposta da Amazon é fornecer um serviço de DNS com alta disponibilidade, baixa latência e alto desempenho.

- **Gerenciamento de banco de dados.**

- O Amazon Relational Database Service (RDS) é um web service de gerenciamento de banco de dados relacional baseado no MySQL, Microsoft SQL Server ou Oracle.
- O Amazon ElastiCache (beta) é um web service que torna fácil implantar, operar e escalar um cache de memória na nuvem.
- O Amazon DynamoDB (BETA) fornece armazenamento de dados NoSQL *on demand*, indexado e sem manutenção, em conjunto com processamento e enfileiramento para conjuntos de dados. O detalhamento do web service DynamoDB não é objeto deste livro.

NoSQL é um termo genérico para uma classe definida de banco de dados não relacionais. Os bancos de dados NoSQL não utilizam esquemas de tabela fixa e, geralmente, não suportam instruções e operações de junção SQL. NoSQL atende a necessidade crescente de prover serviços escaláveis. Bancos de dados NoSQL não visam eliminar bancos de dados relacionais do tipo SQL, mas oferecer uma alternativa a estes bancos.

O DynamoDB é um web service de banco de dados NoSQL cuja proposta é fornecer um desempenho rápido e previsível com facilidade de escalabilidade. Segundo a Amazon, o DynamoDB permite aos clientes redirecionar as cargas administrativas de operação, assim como escalar bancos de dados distribuídos para a AWS. Dessa forma, os clientes ficam isentos de preocupações com provisionamento, instalação e configuração de hardware, replicação, patches de software ou escalabilidade de cluster. A ideia é que o esforço de administração seja reduzido.

A **Figura 1-7** ilustra os serviços de infraestrutura.

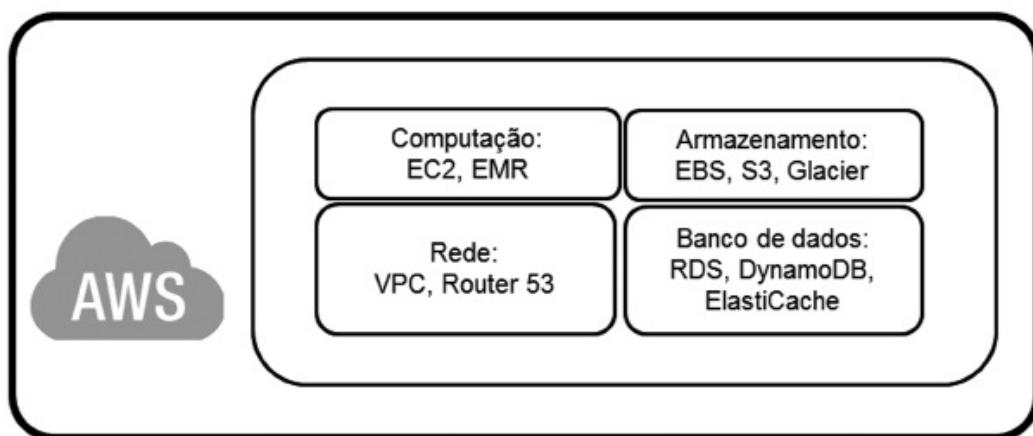


Figura 1-7 Serviços de infraestrutura

- **Gerenciamento.**

◦ Serviços de gerenciamento baseados em métricas e alarmes fornecidos pelo Amazon CloudWatch. O CloudWatch oferece monitoramento dos recursos de nuvem da AWS e de aplicativos que os clientes executam na AWS. O *Auto Scaling*, um recurso do EC2, permite o balanceamento da carga do aplicativo de forma automática.

- **Serviços básicos de plataforma:** os elementos centrais dos serviços básicos de plataforma são resumidos a seguir:

- **Gerenciamento de aplicativos.**

- O Amazon Elastic Beanstalk (beta) permite gerenciar os detalhes de implantação do fornecimento de capacidade, balanceamento de carga, escalonamento automático e monitoramento do status do aplicativo.
    - A AWS CloudFormation oferece aos desenvolvedores e administradores de sistemas uma maneira fácil de criar e gerenciar um grupo de recursos relacionados à AWS e fornecê-los e atualizá-los de uma forma organizada e previsível.
    - O Amazon CloudSearch (beta) permite integrar uma funcionalidade de pesquisa rápida e escalável nos aplicativos.

- **Gerenciamento de conteúdo.**

- O Amazon CloudFront é um web service para distribuição de conteúdo. Ele se integra a outros Amazon Web Services para oferecer aos desenvolvedores e às empresas uma maneira fácil de distribuir conteúdo aos usuários finais com baixa latência e altas velocidades de transferência de dados.

- **Mensageria.**

- O Amazon Simple Queue Service (SQS) permite desacoplar os componentes das aplicações usando um sistema de mensagens. A mensageria permite a construção de aplicações escaláveis baseadas em filas de mensagens. As mensagens são armazenadas pela AWS em servidores virtuais e DATACENTERS múltiplos para fornecer a redundância e a confiabilidade necessárias para um sistema de mensagens.
    - O Amazon Simple E-mail Service (SES) (beta) é um serviço de envio de e-mails transacional para empresas e desenvolvedores. O SES elimina a complexidade e a despesa de criar uma solução de e-mail interna ou licenciar, instalar e operar um serviço de e-mail terceirizado. Ele se integra com outros serviços AWS, facilitando o envio de e-mails de

aplicativos hospedados em serviços como o EC2.

- O Amazon Simple Notification Service (SNS) (beta) é um web service que facilita a configuração, a operação e o envio de notificações. O SNS fornece uma interface simples de web service que pode ser usada para criar tópicos desejados para notificar aplicativos (ou pessoas), inscrever clientes nesses tópicos, publicar mensagens e fazer com que essas mensagens sejam entregues ao protocolo de escolha dos clientes. O SNS entrega notificações aos clientes usando um mecanismo “push” que elimina a necessidade de verificação periódica ou “pull” para novas informações e atualizações. O SNS pode ser utilizado para criar fluxos de trabalho acionados por eventos e aplicativos de mensagens sem a necessidade de fazer a gestão do *middleware* e dos aplicativos.

- **Workflow.**

- O Amazon Simple Workflow Service (SWF) (beta) é um web service de fluxo de trabalho para construção de aplicativos escaláveis e flexíveis. O SWF coordena de modo seguro as etapas de processamento em um aplicativo, incluindo a automatização de processos de negócios para aplicativos de seguros ou financeiros e a construção de aplicativos de análise de dados.

A **Figura 1-8** ilustra os serviços básicos de plataforma.

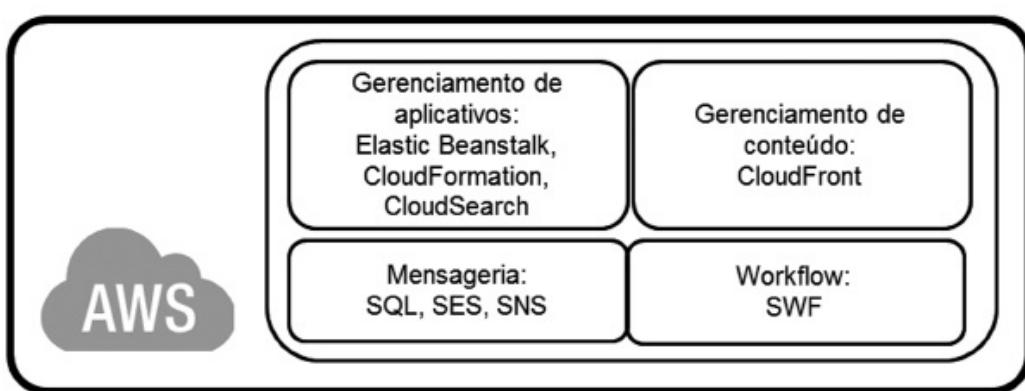


Figura 1-8 Serviços básicos de plataforma

Existem ainda os serviços de gerenciamento de identidade e acesso realizado pelo *Identity and Access Management* (IAM), que permite fazer o controle do acesso aos serviços e recursos da AWS pelos usuários. O IAM permite criar e gerenciar usuários vinculados a uma conta AWS e também possibilita a concessão de acesso a recursos para usuários gerenciados fora

da AWS, mas que fazem parte do diretório corporativo. O IAM oferece maior segurança, flexibilidade e controle ao usar a AWS.

## 1.6.2. Interfaces

A AWS fornece diversas ferramentas, bibliotecas e SDKs (*Software Development Kits*) baseados na API AWS para facilitar a programação do DATACENTER. É isso mesmo: a ideia do modelo é que o DATACENTER seja programável com base em uma API. O programador pode até continuar utilizando a ferramenta habitual de desenvolvimento e só incorporar os recursos de nuvem providos pela AWS.

A interface entre os aplicativos e a infraestrutura da AWS pode ser feita basicamente de quatro maneiras distintas:

- **Por APIs.** Pode-se utilizar APIs baseadas nos SDKs da AWS, bibliotecas de terceiros ou utilizar diretamente APIs AWS.
- **Por *Command Line Interface (CLI)***, que é uma interface de linha de comando cliente baseada em Java que encapsula a API AWS. As ferramentas de linha de comando (Amazon EC2 API Tools) estão disponíveis como um arquivo ZIP em:  
<http://aws.amazon.com/developertools/351?encoding=UTF8&jiveRedirect=1>. O arquivo ZIP é autônomo; nenhuma instalação é necessária. Basta fazer download do arquivo e descompactá-lo. Reforçando, as ferramentas de linha de comando do EC2 encapsulam as ações da API EC2 e requerem Java. A Amazon alerta que para utilizar esta interface deve-se ter o Java 1.6 ou posterior instalado.
- **Por SDKs** que facilitam o desenvolvimento de aplicativos e aumentam a produtividade. Estes SDKs incluem: amostras de códigos, bibliotecas, documentação, templates, amostras de aplicações e outras ferramentas relacionadas à plataforma de desenvolvimento. Existem inclusive SDKs para os sistemas operacionais Android e iOS. SDK é a sigla de *Software Development Kit* (kit de desenvolvimento de software). Normalmente os SDKs são disponibilizados por empresas ou projetos “open source” para que programadores externos tenham uma melhor produtividade no desenvolvimento do novo aplicativo em um determinado sistema operacional. Um exemplo de um SDK é a plataforma da Microsoft, que inclui documentação, código e utilitários para que programadores consigam desenvolver suas aplicações de acordo com um padrão de desenvolvimento da Microsoft.
- **Pelo console de gerenciamento (AWS Management Console)**, que é uma interface baseada na web. O console de gerenciamento é a

principal ferramenta disponibilizada pela AWS e trata-se de uma interface para o usuário. Neste livro os exemplos são focados no uso do console de gerenciamento.

A **Figura 1-9** resume as interfaces AWS disponíveis para o usuário. A figura sinaliza que com o console de gerenciamento é possível utilizar a AWS com um maior nível de abstração. Mas deve ficar claro que nem tudo é possível de ser feito utilizando apenas o console de gerenciamento.

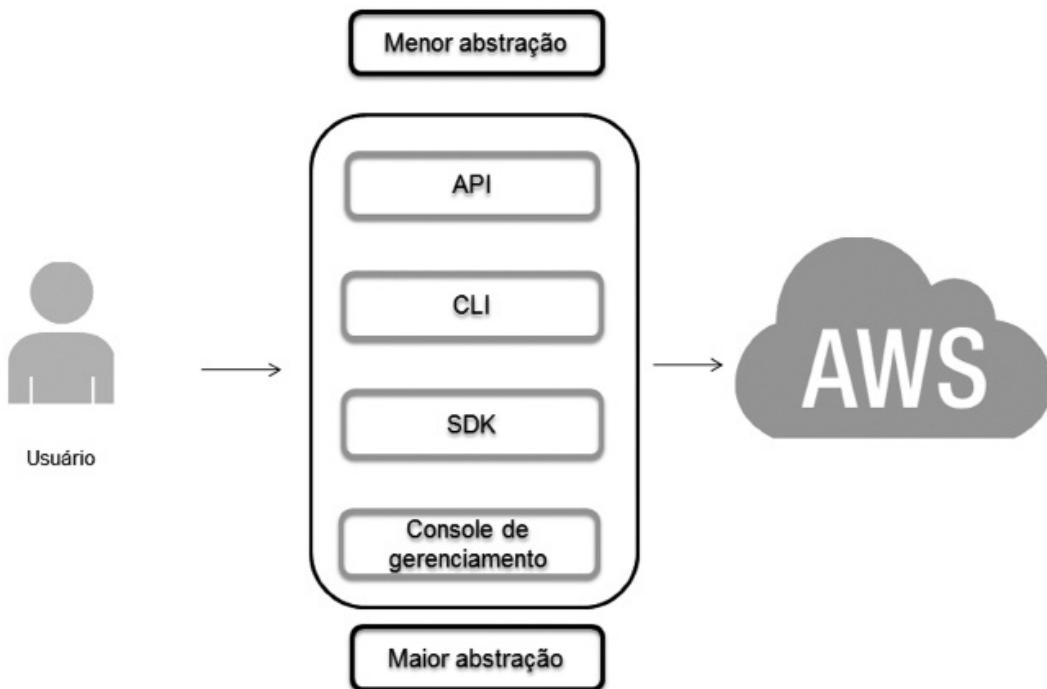


Figura 1-9 Interfaces da AWS

### 1.6.3. Site

O site da Amazon Web Services em português (<http://aws.amazon.com/pt/>) oferece uma série de opções para acesso aos principais produtos e soluções da AWS. Já na página inicial é possível ir para o cadastro da conta e obter o acesso ao console de gerenciamento.

O site fornece links para as principais opções relativas à conta (minha conta; atividade da conta; relatórios de utilização) e para as credenciais de segurança.

O site também fornece links para desenvolvedores, para o suporte (support) e para casos de empresas que estão utilizando a AWS.

A **Figura 1-10** ilustra a página inicial do site AWS em português.



Figura 1-10 Endereço da Amazon AWS

#### 1.6.4. Console de gerenciamento

O console de gerenciamento (*AWS Management Console*) é uma interface de usuário baseada na web. Ele permite fazer boa parte do gerenciamento dos web services AWS através de uma simples e intuitiva interface de usuário.

O console de gerenciamento suporta as funções mais importantes da AWS, incluindo: lançar/reiniciar/encerrar instâncias, fornecer um pacote de imagens AMI, gerenciar volumes EBS, alocar e anexar IPs para máquinas virtuais, definir as configurações de grupos de segurança para instâncias e gerenciar pares de chave para efeito do aumento da segurança no acesso às instâncias.

A página inicial pode ser definida e o usuário pode iniciar o console de gerenciamento por qualquer um dos serviços disponíveis. Pode-se também personalizar a página principal do console de gerenciamento escolhendo os ícones e os textos para acesso direto a alguns dos serviços.

A **Figura 1-11** ilustra a tela do console de gerenciamento AWS.

**Welcome**

The AWS Management Console provides a graphical interface to Amazon Web Services. Learn more about how to use our services to meet your needs, or get started by selecting a service.

Getting started guides

Reference architectures

Free Usage Tier

**Set Start Page**

Console Home ▾

**AWS re:Invent**  
November 27-29, 2012 Las Vegas  
Register Now

**Amazon Web Services**

Compute & Networking

- Direct Connect**  
Dedicated Network Connection to AWS
- EC2**  
Virtual Servers in the Cloud
- Elastic MapReduce**  
Managed Hadoop Framework
- Route 53**  
Scalable Domain Name System
- VPC**  
Isolated Cloud Resources

Storage & Content Delivery

- CloudFront**  
Global Content Delivery Network
- Glacier**  
Archive Storage in the Cloud
- S3**  
Scalable Storage in the Cloud
- Storage Gateway**  
Integrates on-premises IT environments with Cloud storage

Database

- DynamoDB**  
Predictable and Scalable NoSQL

Deployment & Management

- CloudFormation**  
Templated AWS Resource Creation
- CloudWatch**  
Resource & Application Monitoring
- Elastic Beanstalk**  
AWS Application Container
- IAM**  
Secure AWS Access Control

App Services

- CloudSearch**  
Managed Search Service
- SES**  
Email Sending Service
- SNS**  
Push Notification Service
- SQS**  
Message Queue Service
- SWF**  
Workflow Service for Coordinating Application Components

Announcements

- Amazon RDS for MySQL Now Supports Read Replica Promotion
- Amazon RDS for SQL Server Now Supports SSL
- AWS GovCloud (US) Region Announces New Features and Services

More...

**Service Health** [Edit](#)

Click Edit to add at least one service and at least one region to monitor.

[Service Health Dashboard](#)

Figura 1-11 Console de gerenciamento AWS

**Para criar uma conta na AWS:**

Acesse <http://aws.amazon.com> e clique em “Cadastre-se agora”. A **Figura 1-12** ilustra a tela inicial da AWS.

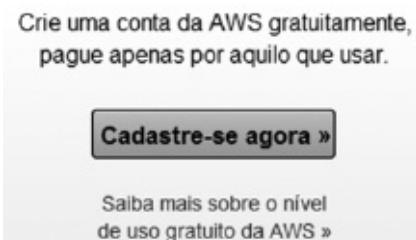


Figura 1-12 Cadastre-se agora

Siga as instruções na tela.

Parte do procedimento de cadastro envolve uma chamada de telefone para você, e a AWS solicita que você digite um PIN usando o teclado do telefone. Quando a conta é criada, a AWS cadastra automaticamente a conta para todos os serviços. A cobrança virá apenas pelos serviços utilizados. A AWS também valida o seu cartão de crédito para efeitos de cobrança.

A Figura 1-13 ilustra a tela inicial para entrar ou criar uma conta na AWS (*Sign in or create an AWS account*).

The screenshot shows the AWS sign-in page. At the top left is the Amazon Web Services logo. Below it, the heading "Sign In or Create an AWS Account" is displayed. A sub-instruction "You may sign in using your existing Amazon.com account or you can create a new account by selecting "I am a new user."" is present. A text input field labeled "My e-mail address is:" contains a placeholder email address. Two radio buttons are shown: one for "I am a new user." (unchecked) and one for "I am a returning user and my password is:" (checked). Below these are two empty text input fields. A large blue "Sign in using our secure server" button is centered. Below the button are links for "Forgot your password?" and "Has your e-mail address changed?". At the bottom, there is a note about AWS Identity and Access Management and AWS Multi-Factor Authentication.

Figura 1-13 *Sign in or create an AWS account*

### 1.6.5. Custos

Há três características fundamentais que determinam os custos de uso da AWS: a computação, o armazenamento e a transferência de dados para fora da nuvem. Os preços variam dependendo dos serviços que estão sendo utilizados e da forma que são adquiridos. Recursos são cobrados mediante o uso. O capítulo 4 trata da precificação e da forma de faturar da AWS.

### 1.6.6. Licenciamento de software

Existem duas formas principais de licenciar software para ser utilizado na nuvem. A forma tradicional, onde a própria AWS já licenciou o software para o uso, e o BYOL (*Bring Your Own License*), que permite que o cliente utilize suas próprias licenças quando da utilização da nuvem. O BYOL trata de um contrato específico relacionado diretamente com o fornecedor.

#### 1.6.6.1. Licença incluída

No modelo de serviço “license included” não é necessário ter adquirido separadamente licenças do fornecedor, pois o software já foi licenciado pela AWS. A definição de preço “license included” por hora inclui o software, os recursos de hardware subjacentes e as capacidades de gerenciamento da

### 1.6.6.2. Traga sua licença

Se o cliente já é proprietário de licenças de um fornecedor específico, pode usar o modelo BYOL (*Bring Your Own License* – em português, “traga sua licença”) e mover as licenças para o provedor de nuvem AWS. O modelo BYOL é projetado para clientes que preferem usar as licenças existentes para aplicativos ou banco de dados ou adquirir novas licenças diretamente do fornecedor.

O BYOL é uma peça-chave na viabilização do modelo de cloud computing. Ele permite que um cliente que resolva ir para a nuvem possa aproveitar um contrato em vigor com um grande fabricante como a Microsoft ou a Oracle, sem custo extra com a utilização da mobilidade de licença.

O BYOL, em resumo, é utilizado para:

- Estender o valor da licença de aplicações de servidor implantando-as na nuvem ou em uma solução com parte interna (*on-premises*) e outra na nuvem.
- Tirar proveito da infraestrutura de cloud computing e mudar a forma de adquirir serviços de infraestrutura de TI.

Com a mobilidade de licenças do *Software Assurance* da Microsoft, por exemplo, pode-se implantar no DATACENTER de um parceiro autorizado algumas licenças de aplicações de servidor compradas por meio do contrato de licenciamento por volume (*Software Assurance*).

A Microsoft sugere as seguintes instruções básicas para os clientes do licenciamento por volume que queiram implantar cargas de trabalho dedicadas em servidores de aplicação com um Parceiro de Mobilidade Autorizado como a AWS, utilizando o benefício da mobilidade de licenças do *Software Assurance*.

#### 1. Avalie as licenças.

Trabalhe com a equipe de aquisição de licenciamento, com um representante da Microsoft ou com o revendedor de sua preferência para entender a posição de suas licenças. Verifique se:

- As licenças de servidor de aplicação que deseja implantar com um Parceiro de Mobilidade Autorizado são identificadas como qualificadas no documento PUR (*Product Use Rights* – direitos de uso do produto) atual.
- As licenças de servidor de aplicação que deseja implantar com um Parceiro de Mobilidade Autorizado têm cobertura ativa do *Software*

*Assurance.*

## **2. Implante as licenças.**

As licenças qualificadas podem ser implantadas no DATACENTER compartilhado da AWS. Permitindo que a AWS use licenças existentes, reduz-se assim o custo da infraestrutura de nuvem.

## **3. Verifique as licenças.**

Em até dez dias a partir da implantação, preencha o Formulário de Verificação de Licenças disponível no site do Licenciamento por Volume e o entregue a um representante da Microsoft ou ao revendedor de sua preferência para que ele o envie à Microsoft. A Microsoft usa o formulário preenchido para verificar se as licenças das cargas de trabalho implantadas na nuvem são qualificadas de acordo com os termos do benefício Mobilidade de Licenças do *Software Assurance* e confirma essa informação com você e a AWS.

Uma variedade de aplicações Microsoft pode rodar na nuvem AWS, incluindo Microsoft Exchange Server, Microsoft SharePoint Server, Microsoft SQL Server Standard e Microsoft SQL Server Enterprise Edition. Já o Microsoft Windows Server atualmente não é compatível com a licença existente do Windows no servidor EC2 ou qualquer outro ambiente de nuvem. A AWS disponibiliza também o AWS Tools para Windows PowerShell, que permite gerenciar e criar scripts dos serviços AWS no ambiente Windows.

Os termos específicos de licença de software variam de acordo com cada fornecedor. Portanto, nesta modalidade, recomenda-se que sejam verificados os termos de licenciamento do fornecedor de software para determinar se as licenças existentes estão autorizadas para uso na AWS.

## **1.7. Imagens e instâncias**

### **1.7.1. Introduzindo as imagens AMI**

A Amazon Machine Image (AMI) é essencialmente uma imagem de servidor que agrupa sistema operacional, muitas vezes aplicativos e configurações associadas. A AMI contém as informações necessárias para pôr no ar uma instância ou servidor virtual. No caso de um servidor web, por exemplo, a AMI deve conter todos os softwares para dar vida a este servidor (por exemplo, Sistema Operacional Linux, Servidor Apache, etc.).

A Amazon publica muitas AMIs que contêm configurações comuns de software. Desenvolvedores também publicam AMIs “customizadas”, que podem ser encontradas facilmente no site AWS, e cada um pode criar suas

próprias AMIs. Os três tipos básicos de AMIs são resumidas na **Tabela 1-3**.

**Tabela 1-3 Tipos de AMIs**

<b>Tipo</b>	<b>Definição</b>
<b>Paga</b>	Imagens que podem ser vendidas a outros usuários. Imagens com funções específicas que podem ser iniciadas por qualquer pessoa que queira pagar por hora de uso com base nas taxas da Amazon.
<b>Privada</b>	Imagens criadas por usuários AWS que são particulares por padrão. É possível conceder acesso a outros usuários para utilizar imagens privadas.
<b>Pública</b>	Imagens criadas por usuários AWS e liberadas para a comunidade da AWS, para que qualquer pessoa possa iniciar instâncias com base nelas e usá-las da forma que desejar. A <i>AWS Developer Connection</i> lista todas as imagens públicas.

A AWS fornece várias ferramentas de linha de comando que facilitam a criação e o gerenciamento de AMIs. As AMIs são armazenadas no Amazon Simple Storage Service (S3) e, após o registro com o EC2, um ID exclusivo é atribuído à imagem, que pode ser usado para identificá-la e iniciar uma instância a partir dela.

Os sistemas operacionais atualmente disponíveis para utilização com instâncias EC2 incluem:

- Red Hat Enterprise Linux.
- Windows Server.
- Oracle Enterprise Linux; SUSE Linux Enterprise.
- Amazon Linux AMI.
- Ubuntu Linux; Fedora; Gentoo Linux; Debian; CentOS; FreeBSD.

Uma pequena amostra de software aplicativo para utilização atual com o EC2 na forma de AMI inclui:

- **Banco de dados:** IBM DB2, IBM Informix Dynamic Server, Microsoft SQL Server Standard, MySQL Server Enterprise.
- **Hospedagem web:** Apache HTTP, IIS/ASP.Net, IBM Lotus Web Content Management, IBM WebSphere Portal Server.

Quando se inicia a criação de uma instância na AWS é possível escolher entre utilizar um *Classic Wizard*, que permite a escolha de diversos parâmetros de configuração da futura instância, ou a utilização de uma opção *Quick Launch Wizard*, que assume alguns parâmetros e lança a instância de forma mais rápida. Nos dois casos a primeira coisa a fazer é escolher uma imagem AMI.

Uma lista com as AMIs disponíveis pode ser encontrada em:

<https://aws.amazon.com/amis/>.

Também pode-se lançar uma instância diretamente do AWS Marketplace, visto a seguir.

## 1.7.2. Introduzindo as instâncias EC2

### 1.7.2.1. Introdução

Instância é uma cópia de uma AMI em execução. É possível iniciar uma instância, visualizar seus detalhes e encerrá-la usando as ferramentas fornecidas pela AWS. É possível também usar uma variedade de bibliotecas de terceiros em diferentes linguagens para controlar o ciclo de vida das instâncias.

As instâncias podem ter como base sistemas operacionais como Windows e Linux em plataformas de 32 e 64 bits. Instâncias existem em diferentes capacidades.

É importante saber que a AWS classifica cada tipo de instância em termos de unidades de cálculo de EC2. Cada unidade de cálculo ECU (*EC2 Computing Unit*) fornece a capacidade de processamento equivalente a um processador Opteron 2007 de 1 – 1,2 GHz ou Intel Xeon 2007.

A relação entre a ECU e a capacidade de processamento de uma instância virtual é fonte de muita discussão na internet. A Amazon não detalha este aspecto. Assim, os tipos de instâncias são definidos em termos de unidades ECU, ou seja, em termos de capacidade. A **Figura 1-14** ilustra a relação entre ECU e capacidade do processador.

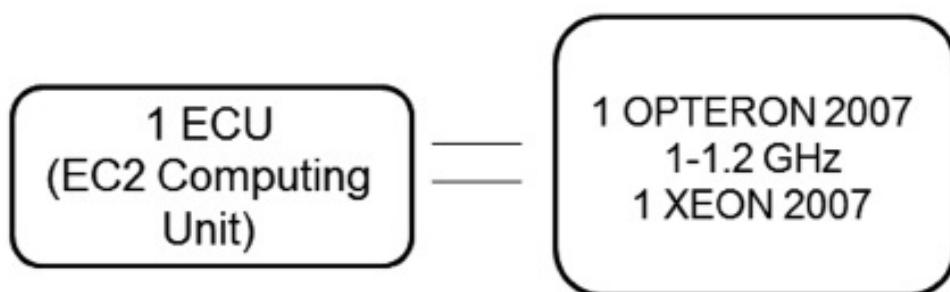


Figura 1-14 Relação entre ECU e capacidade do processador

AMIs idênticas podem ser utilizadas por tipos de instâncias de diferentes capacidades, como ilustrado na **Figura 1-15**.

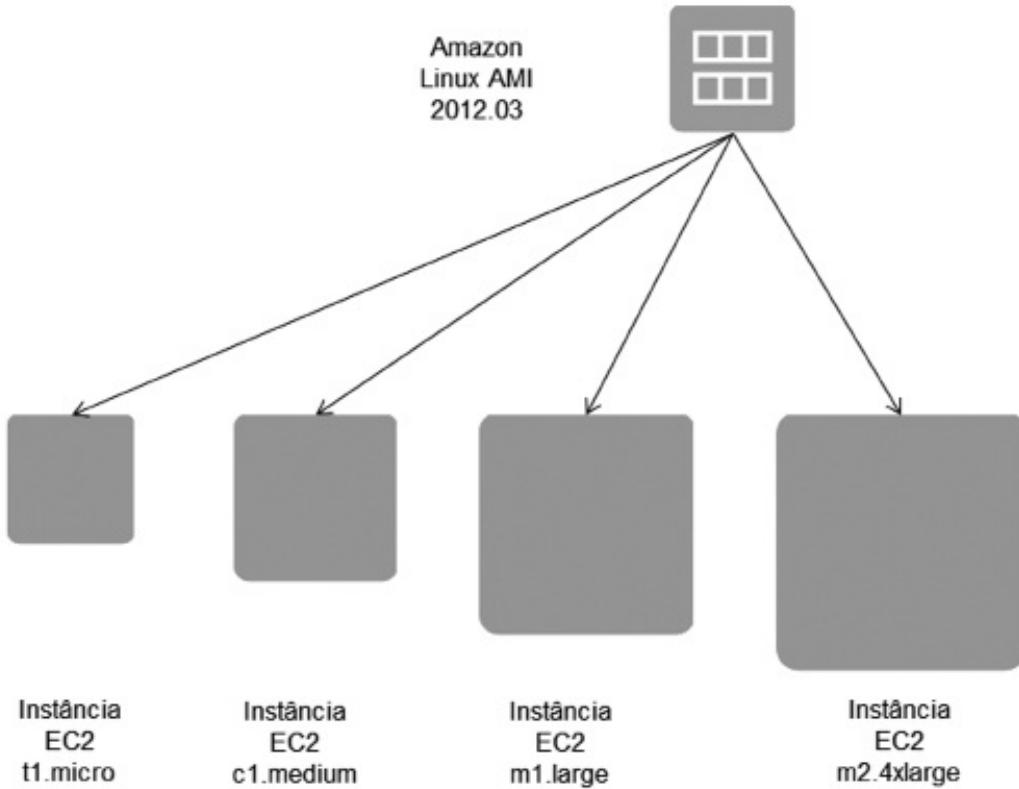


Figura 1-15 AMIs e instâncias

A **Figura 1-16** ilustra uma típica simplificação introduzida pela AWS. Imagine que uma imagem roda em certo tipo de instância e que a carga de trabalho colocada sobre esta instância cresceu mais do que o planejado. Entende-se que se deve trocar o tipo de instância por uma instância maior. Pois bem, na AWS a operação é muito simples para certos tipos de instâncias e imagens AMI relacionadas. Para-se a instância, muda-se o tipo de instância e inicia-se novamente a instância.

Na forma tradicional, o equivalente à troca das instâncias seria a troca dos servidores físicos. No caso do governo brasileiro, por exemplo, supondo que não existissem servidores disponíveis com a nova configuração, isto implicaria em fazer uma licitação para aquisição de um novo servidor – o que poderia levar alguns meses. Mesmo para empresas privadas seria um desafio fazer uma operação como esta em pouco tempo, pois o processo de aquisição de um novo servidor, um componente caro, é naturalmente lento.

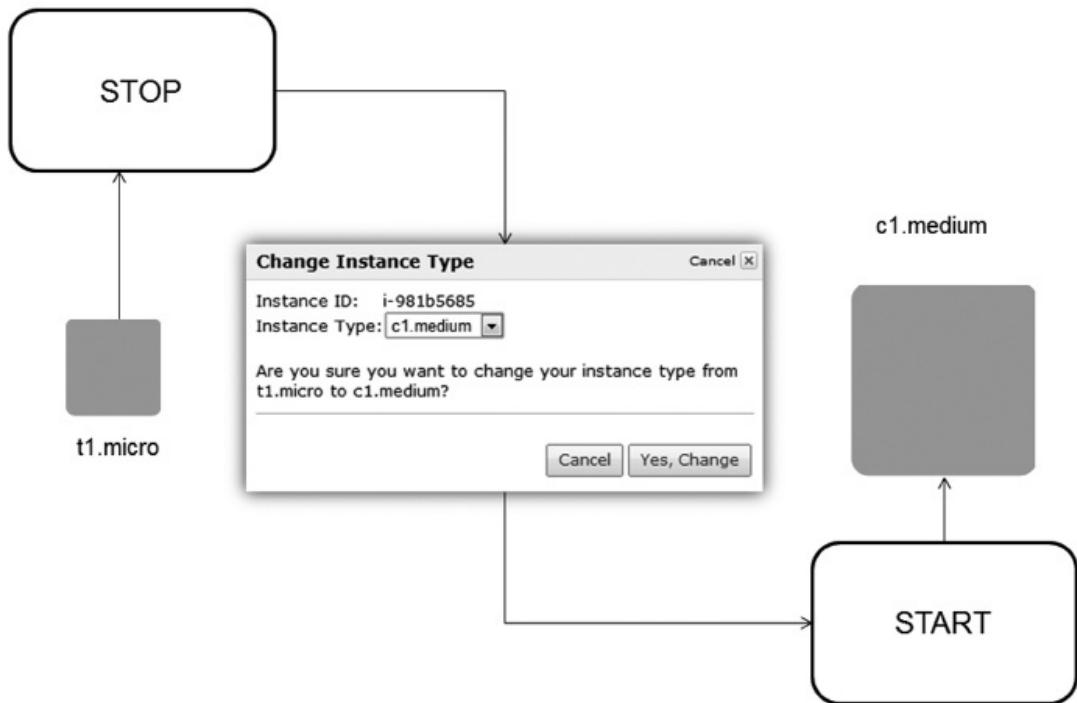


Figura 1-16 Troca do tipo de instância

Recentemente a AWS adicionou o Windows Server 2012 como uma opção para as instâncias Windows. Foram introduzidas 31 AMIs para Windows Server 2012 que incluem dezessete idiomas.

### **1.7.2.2. Par de chaves (*key pairs*) EC2**

Antes de iniciar uma instância no console de gerenciamento AWS, alguns passos devem ser dados. Primeiro deve-se criar um par de chaves EC2, o que pode ser feito facilmente no console de gerenciamento AWS. O par de chaves é uma forma de obter as credenciais de acesso para se conectar às instâncias Windows ou Linux.

### **1.7.2.3. Grupos de segurança (security groups) EC2**

Outro passo a ser dado no sentido de lançar uma instância deve ser o de criar um grupo de segurança. Grupos de segurança permitem configurar as regras de firewall que especificam as restrições de acesso para instâncias executadas dentro desse grupo.

O grupo de segurança define o tráfego de rede de entrada para a instância iniciada. Toda e qualquer instância iniciada no ambiente EC2 é executada dentro de um grupo de segurança. O grupo de segurança é o firewall da instância.

### **1.7.2.4. Endereços IPs elásticos (elastic IPs – EIPs) EC2**

Endereços IPs elásticos (EIPs) são IPs fixos que permitem que instâncias EC2 sejam alcançadas na internet. Se for necessário definir um endereço IP fixo para uma instância, deve-se primeiro alocar um IP no painel do EC2 no

console de gerenciamento, opção “Network & Security/Elastic IPs”, e associá-lo a uma instância. É importante ressaltar que uma instância EC2 pode ter acesso à internet por outros meios que não o estabelecimento de um IP elástico. No capítulo 5 este aspecto será apresentado com maiores detalhes.

### 1.7.3. Recursos persistentes e efêmeros

Os recursos da AWS podem ser agrupados em duas categorias:

- **Recursos persistentes:** permanecem operacionais mesmo que aconteça uma falha de hardware ou de software. São eles: endereços IP elásticos, volumes do tipo *Elastic Block Store* (EBS),平衡adores de carga do tipo *Elastic Load Balancing* (ELB), grupos de segurança e AMIs armazenadas no S3 ou como *snapshots* EBS. Estes conceitos serão vistos mais adiante.
- **Recursos efêmeros:** não possuem redundância e não podem falhar. Quando falham o estado da informação é geralmente perdido. Assim, deve-se garantir a disponibilidade destes recursos utilizando uma arquitetura correta. O EC2, a instância virtual da AWS, é efêmero e é a base de todo o sistema e da sua especificação.

## 1.8. Marketplace

O AWS Marketplace permite encontrar, comparar e iniciar o uso de diversos produtos de software instanciados na plataforma AWS. A Amazon, com o Marketplace, facilitou a descoberta, a implantação e o pagamento desses produtos para que o processo inteiro ficasse mais rápido, simples e interessante para produtores e consumidores de software.

O AWS Marketplace possui softwares de infraestrutura, de negócios, ferramentas de desenvolvimento e de aplicações de negócio. Os preços dos produtos são descritos também. A **Figura 1-17** ilustra o site AWS Marketplace.

O AWS Marketplace foi lançado com quinze categorias de produtos em três grupos:

- **Software de infraestrutura:** desenvolvimento de aplicações, *stacks* de aplicações, servidores de aplicação, bancos de dados e caching, infraestrutura de rede, sistemas operacionais e segurança.
- **Software de negócio:** *Business intelligence*, colaboração, gestão de conteúdo, CRM, e-commerce, *high performance computing*, mídia, gestão de projetos e storage & backup.
- **Ferramentas de desenvolvimento:** ferramentas de *issue & bug tracking*, monitoração, controle de versões e testes.

O AWS Marketplace inclui produtos que se pagam pelo uso e estão disponíveis na forma de AMI ou então por software hospedado, com variados modelos de especificação. Quando se instancia uma AMI, o produto será executado em uma instância EC2 privada e os custos (mensais e/ou por hora) serão detalhados no

relatório de atividades de conta AWS. Já o software hospedado é executado pelo vendedor e acessado através de um cliente tipo browser ou smartphone.

The screenshot shows the AWS Marketplace homepage. At the top, there's a navigation bar with links to 'Amazon Web Services Home', 'Your Account | Help', and a 'GO' button. Below the navigation is a search bar with the placeholder 'Your Software'. To the left, a sidebar lists various software categories such as 'Shop All Categories', 'Software Infrastructure', 'Application Development', 'Application Stacks', etc. The main content area features a large advertisement for 'SAP HANA One', stating it's a 'Real-time data analytics and development platform' now 'certified for production on AWS'. A 'Buy today for \$0.99 per hour' button is visible. Below this, sections for 'Featured Products' and 'Popular Products' show logos for SAP, Alert Logic, aiCache, JumpBox, WordPress, MongoDB, and RubyStack. To the right, there's a section titled 'Development Stacks' with options like 'LAMP Stack - Web Stack (MySQL) prov...', 'JBoss powered by BitNami', and 'JumpBox LAPP Deployment provided by JumpBox'.

Figura 1-17 AWS Marketplace

Recentemente foi criada a categoria de software para BIG DATA com inúmeras ofertas de vários fabricantes. Agora já são 21 categorias no AWS Marketplace.

## 1.9. Partner Network

O AWS *Partner Network* (APN) é um programa global que tem foco em prover informações técnicas e suporte a vendas e marketing aos membros do ecossistema de parceiros AWS.

A afiliação ao APN está disponível a duas categorias de parceiros: de tecnologia (incluindo ISVs, SaaS, fabricantes de ferramentas e/ou plataformas e outros) e de consultoria (incluindo integradores de sistemas, consultorias, provedores de serviços gerenciados e outros). Em qualquer dessas duas categorias, os parceiros podem se qualificar para um de três níveis possíveis (*advanced*, *standard* e *registered*) baseado num conjunto de critérios comuns e públicos.

O programa APN já está ativado. O AWS encorajou todos os atuais parceiros existentes a se pré-qualificarem para os níveis *standard* ou *advanced* via formulário (*APN Upgrade Form*). Foram lançados também os programas *AWS Channel Reseller* e o *Authorized Government Partner Designation*.

## **1.10. Referências bibliográficas**

- Amazon Web Services. **Overview of Amazon Web Services.** Dezembro 2010.
- Barr, Jeff. **Host Your Web Site in the Cloud: Amazon Web Services made easy.** Sitepoint, 2011.
- Buyya, Rajkumar; Broberg, James; Goscinski, Andrzej. **Cloud Computing: Principles and Paradigms.** Wiley, 2011.
- Earl, Thomas. **SOA: Princípios de design de serviços.** Pearson Education do Brasil, 2009.
- Harding, Chris; and members of the Open Group Work Group. **Cloud Computing for Business:** The Open Group Guide, 2011.
- <http://4sysops.com/archives/amazon-ec2-aws-management-console-amazons-cloud-management-tool-reviewed/>
- <http://aws.amazon.com>
- <http://aws.typepad.com/>
- <http://aws.typepad.com/brasil>
- IDC. **The Business Value of Amazon Web Services Accelerates Over Time.** 2012.
- Marston, Sean; Li, Zhi; Zhang, Juheng; Bandyopadhyay, Subhajyoti. **Cloud Computing – The Business Perspective.** Decision Support Systems, 2011.
- Mell, Peter; Grance, Timothy. **The NIST Definition of Cloud Computing:** Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-145, setembro 2011.
- Varia, Jinesh. **Projetando para a nuvem:** práticas recomendadas. Janeiro de 2010 (última atualização em janeiro de 2011).
- Veras, Manoel. **Cloud Computing:** Nova Arquitetura de TI. Brasport, 2012.

# 2. Infraestrutura

## 2.1. Introdução

A infraestrutura global da AWS está na base da oferta das garantias de nível de serviço. Componentes do DATACENTER de qualidade permitem obter serviços de qualidade e bons níveis de garantia. A AWS não publica muitas informações sobre seus DATACENTERS e normalmente prefere mantê-los em sigilo por questões de segurança e conformidade.

A infraestrutura da AWS é baseada nos conceitos de zonas de disponibilidade e de região. As zonas de disponibilidade refletem os DATACENTERS da AWS, e as regiões consistem de uma ou mais zonas de disponibilidade que são separadas em áreas geográficas ou países. Os pontos de presença são os lugares onde a AWS permite armazenar principalmente conteúdos estáticos proveniente dos clientes.

Este capítulo descreve regiões, zonas de disponibilidade, pontos da rede global de DNS e rede de pontos de presença de conteúdo.

## 2.2. Componentes

A **Figura 2-1** ilustra os componentes da infraestrutura global.

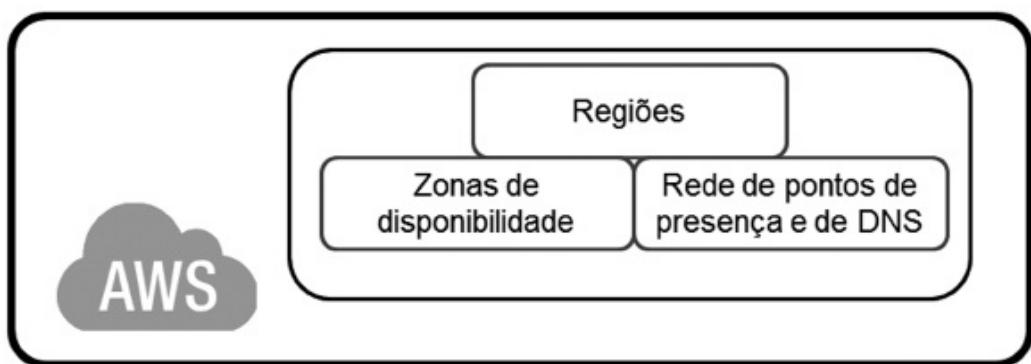


Figura 2-1 Componentes da infraestrutura global

A AWS pode ser considerada um serviço global de IaaS. Mesmo tendo algumas características de PaaS, é mais comum e mais lógico classificá-la como um serviço IaaS. A **Figura 2-2** ilustra esta classificação.

Um dos segredos da AWS é ter criado uma infraestrutura global baseada em DATACENTERS de alta disponibilidade para dar suporte ao sistema AWS. Esta infraestrutura global continua evoluindo junto com o sistema. O padrão do DATACENTER AWS possibilita obedecer a requisitos legais, de

disponibilidade, de desempenho e de segurança. Esta infraestrutura avança rapidamente e é possível que, enquanto eu estou escrevendo este livro, a Amazon já tenha lançado outro DATACENTER aumentando a sua malha de suporte à plataforma. Hoje são vinte DATACENTERS espalhados em oito regiões, sem considerar a região GovCloud, sendo que cada zona de disponibilidade corresponde a um DATACENTER.

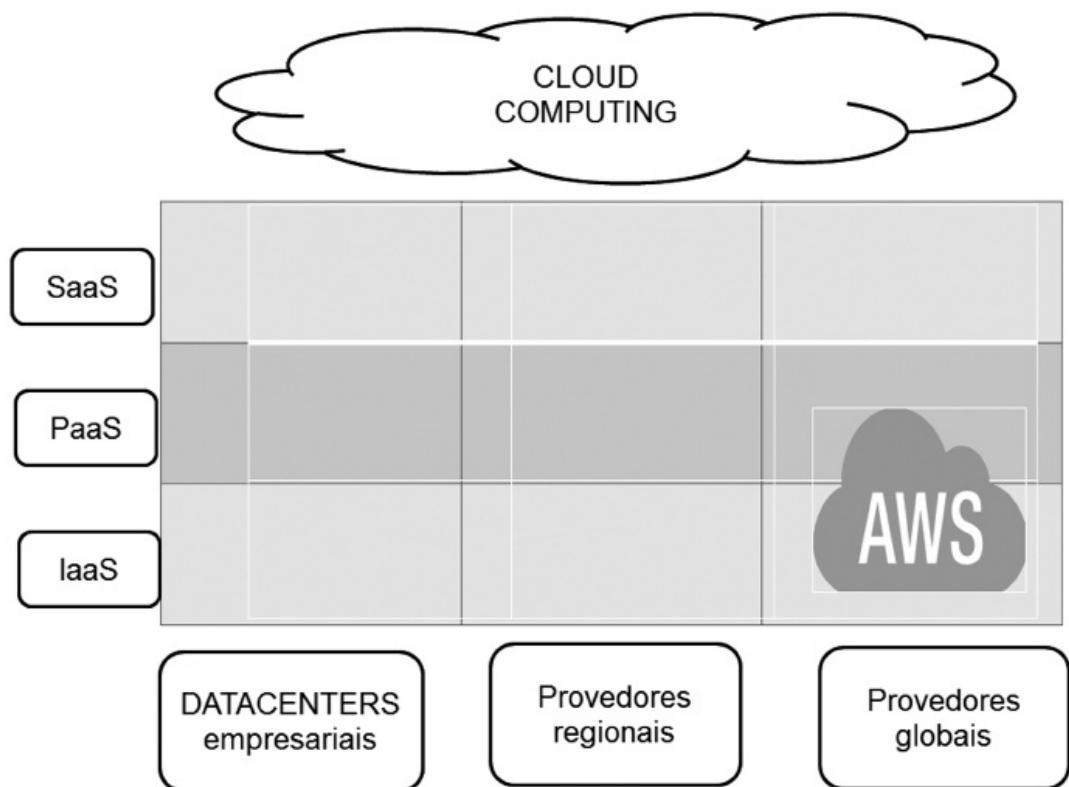


Figura 2-2 Posicionamento da AWS

O investimento realizado pela Amazon na construção de vários DATACENTERS para suporte à arquitetura não garante a total disponibilidade de uma aplicação quando colocada para rodar na AWS. A AWS já enfrentou problemas de indisponibilidade com o DATACENTER AWS de US East (Virginia), o maior e mais antigo, em abril de 2011. Em dezembro de 2011 a Amazon teve que fazer o “reboot” de milhares de instâncias EC2 neste mesmo DATACENTER. Em junho de 2012 foi a vez de uma tempestade trazer problemas para os sistemas de energia da AWS na mesma região. Neste caso específico, as aplicações projetadas para funcionar em zonas de disponibilidade distintas tiveram problema com o *Elastic Load Balancing*. Mais recentemente, em outubro de 2012, os volumes EBS da AWS apresentaram problema em US East (Virginia), deixando diversos sites populares fora do ar.

A indústria de computação de nuvem é jovem, e estes problemas trazem o amadurecimento das soluções. Em 2015 estima-se que o faturamento em serviços de nuvem chegará a mais de US\$ 70 bilhões, conforme previsão do IDC ilustrada no artigo “The Sun Shines on the Cloud”, publicado em <http://wsj.com>. Este dado é coerente com a previsão do IDC mostrada

anteriormente no capítulo 1.

Para aplicações críticas é necessário pensar no projeto de recuperação de desastres utilizando diversas opções fornecidas pela Amazon e até de terceiros. No capítulo 13 serão tratados os aspectos de continuidade e recuperação de desastres da AWS.

## 2.3. Regiões e zonas de disponibilidade

### 2.3.1. Introdução

A AWS permite rodar as instâncias EC2 em múltiplos locais. Os locais da AWS são compostos por regiões e por zonas de disponibilidade (*Availability Zones – AZ*). Regiões consistem de uma ou mais zonas de disponibilidade que são separadas em áreas geográficas ou países. As zonas de disponibilidade são zonas independentes onde estão os DATACENTERS.

Huan Liu[\[5\]](#) estimou que a AWS possuía em março de 2012 cerca de 454 mil servidores. Liu considerou que existem 64 servidores blades por rack nos DATACENTERS AWS. A região US East (Virginia) teria 5.030 racks e 321.920 servidores blades. A região South America (São Paulo) teria cerca de 25 racks para 1.600 servidores blades. A **Tabela 2-1** ilustra a estimativa feita por Huan Liu para todos os sites AWS. A estimativa feita por Liu não considera a região AP (Sydney).

Tabela 2-1 Estimativa da quantidade de servidores na AWS

DATACENTER	Racks	Servidores blade
US East (Virginia)	5.030	321.920
US West (Oregon)	41	2.624
US West (N. California)	630	40.320
EU West (Irlanda)	814	52.096
AP (Japão)	314	20.096
AP (Singapura)	246	15.744
SA (São Paulo)	25	1.600
Total	7.100	454.400

### 2.3.2. Regiões

Regiões são conjuntos isolados de DATACENTERS em uma determinada

geografia.

A **Figura 2-3** ilustra as regiões disponíveis quando da construção deste livro.



Figura 2-3 Regiões AWS

Como visto, instâncias EC2 estão disponíveis em sete regiões e mais a região específica para cidadãos americanos (*GovCloud*). Quando se utiliza a AWS pode-se definir onde os dados devem ficar armazenados, onde devem rodar as instâncias, onde as filas de mensagens são inicializadas e os bancos de dados devem ser instanciados. Este é um aspecto muito relevante para objetivos de conformidade.

Instâncias EC2 podem rodar em múltiplas zonas de disponibilidade ao mesmo tempo. Podem ser utilizados, por exemplo, serviços de balanceamento de carga (*Elastic Load Balancing* – ELB) que se integram a instâncias em múltiplas zonas de disponibilidade para conseguir alta disponibilidade com baixa intervenção humana. O ELB da AWS pode平衡ear tráfego entre múltiplas instâncias e até entre múltiplas zonas de disponibilidade de forma automática utilizando recursos de gerenciamento da plataforma baseados em métricas e alarmes.

Outros web services, como SQS e o RDS, são blocos de construção de alto nível que podem ser incorporados ao desenvolvimento da aplicação e melhorar ainda mais o nível de disponibilidade.

### 2.3.2.1. Região GovCloud

*GovCloud* é uma região só para acesso de funcionários do governo americano. Possui requisitos de segurança e conformidade diferenciados e é conceitualmente uma nuvem comunitária, mesmo a AWS sendo um modelo essencialmente de nuvem pública. Esta região foi projetada para permitir que agências governamentais dos EUA transfiram cargas de trabalho dos aplicativos para a nuvem com garantias de requisitos específicos de regulamentação e conformidade.

A região *GovCloud* é mais sofisticada do ponto de vista da conformidade do que a Amazon VPC, que é baseada em rede privada virtual (*Virtual Private Network* – VPN). A VPC, rede privada virtual da Amazon, será vista detalhadamente no capítulo 8.

A **Tabela 2-2** permite fazer uma comparação entre os modelos AWS do

ponto de vista da conformidade. Observe que o padrão ITAR só é possível de ser obtido com a opção *GovCloud*. Serviços EC2, RDS, DynamoDB, SNS e SQS podem ser utilizados normalmente na região *GovCloud*. A região *GovCloud* possui três zonas de disponibilidade.

**Tabela 2-2 Modelos AWS e conformidade**

	Servidores lógicos e isolamento das aplicações	Política de acesso com informação granular	Isolamento das redes lógicas	Isolamento das redes físicas	Isolamento dos servidores físicos	ITAR compliant (somente cidadãos americanos)
<b>AWS</b>	X	X				
<b>AWS VPC</b>	X	X	X	X		
<b>AWS GovCloud (EUA)</b>	X	X	X	X	X	X

### **2.3.2.2. Região South America (São Paulo)**

A região South America (São Paulo) atende preferencialmente a clientes da AWS no Brasil e na América Latina e suporta os seguintes principais produtos e serviços:

- Elastic Compute Cloud (EC2).
- Elastic MapReduce.
- Simple Storage Service (S3).
- Storage Gateway.
- DynamoDB.
- Relational Database Service (RDS).
- Simple Queue Service (SQS).
- CloudFormation.
- CloudWatch.

Também já existe um ponto para Route 53 (DNS) e CloudFront em São Paulo.

Uma dica da Amazon é que, se existirem restrições com a localização dos dados de certa empresa, será preciso avaliar se o serviço necessário pode efetivamente ser utilizado.

### **2.3.3. Zonas de disponibilidade (*Availability Zones – AZ*)**

As zonas de disponibilidade são posições distintas dentro das regiões que são projetadas para serem isoladas de falhas entre si. As zonas de disponibilidade possuem rede com excelente conectividade e baixa latência entre si, o que facilita na construção de processos de alta disponibilidade e backup entre zonas. Ao iniciar as instâncias em zonas separadas, os aplicativos podem ser protegidos de falha de um único local. Iniciar instâncias em regiões diferentes possibilita obter um nível excelente de recuperação de desastres para estas aplicações.

Instâncias EC2 podem rodar em múltiplas zonas de disponibilidade. Pode-se utilizar, por exemplo, serviços de balanceamento de carga (*Elastic Load Balancing – ELB*) que se integram a instâncias em múltiplas zonas de disponibilidade para conseguir alta disponibilidade com baixa intervenção humana. O ELB da AWS pode平衡ear tráfego entre múltiplas instâncias e até entre múltiplas zonas de disponibilidade de forma automática utilizando recursos de gerenciamento da plataforma baseados em métricas e alarmes.

A **Figura 2-4** ilustra o conceito de zona de disponibilidade. Elas são nomeadas como AZA, AZB, etc. Veja que a região US East, ilustrada, possui quatro zonas de disponibilidade.

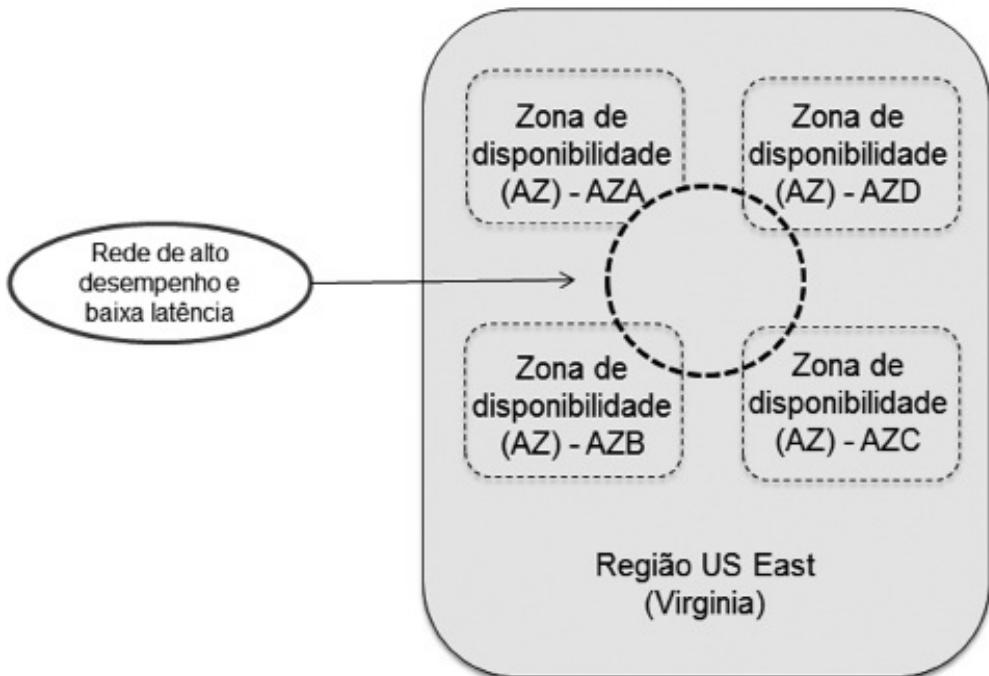


Figura 2-4 Zona de disponibilidade – 1

A **Figura 2-5** ilustra uma região específica com duas zonas de disponibilidade.

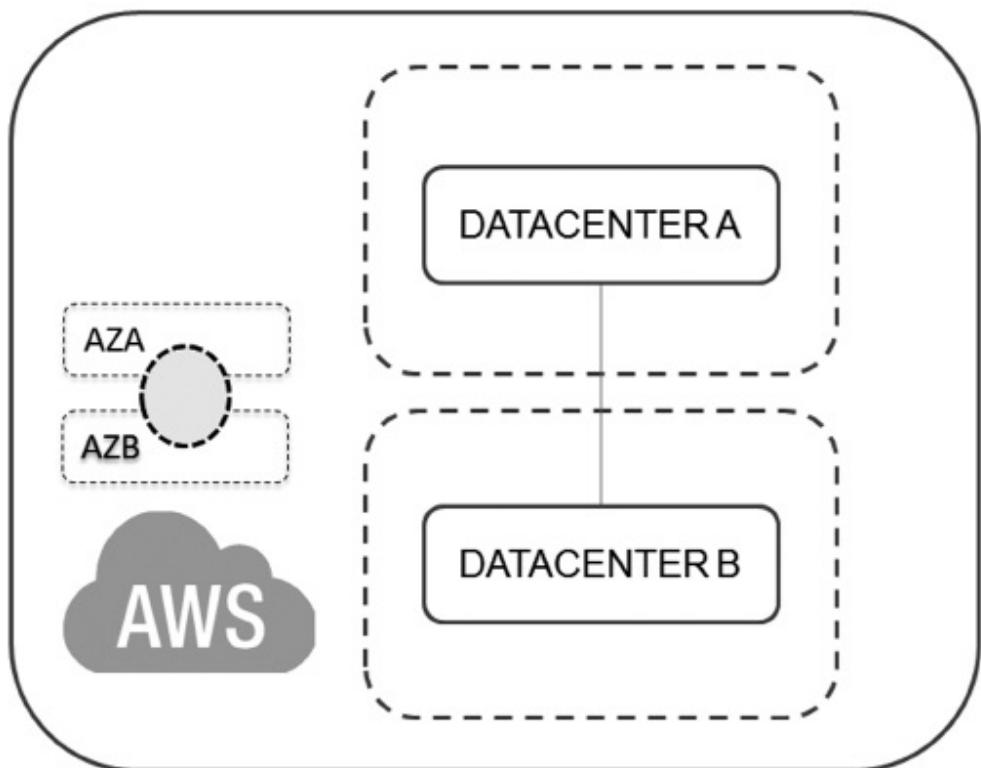


Figura 2-5 Zona de disponibilidade – 2

Cada zona AWS possui seu próprio DATACENTER com infraestrutura independente. As zonas de disponibilidade se confundem com os

DATACENTERS, conforme visto. Pontos comuns de falhas como geradores e equipamentos de refrigeração não são compartilhados nos DATACENTERS de uma mesma região. Além disso, eles são fisicamente separados de tal forma que mesmo em desastres incomuns como incêndios, tornados ou enchentes estes afetariam somente uma única zona de disponibilidade.

As instalações físicas e os conjuntos de equipamentos de energia (EA, EB) e equipamentos de refrigeração (RA, RB) são independentes. As duas zonas de disponibilidade são conectadas por uma rede de baixa latência e alto desempenho. As infraestruturas de Tecnologias da Informação (Infra de TI A, Infra de TI B) são totalmente independentes também.

A **Figura 2-6** detalha os componentes dos DATACENTERS da AWS em uma região específica.

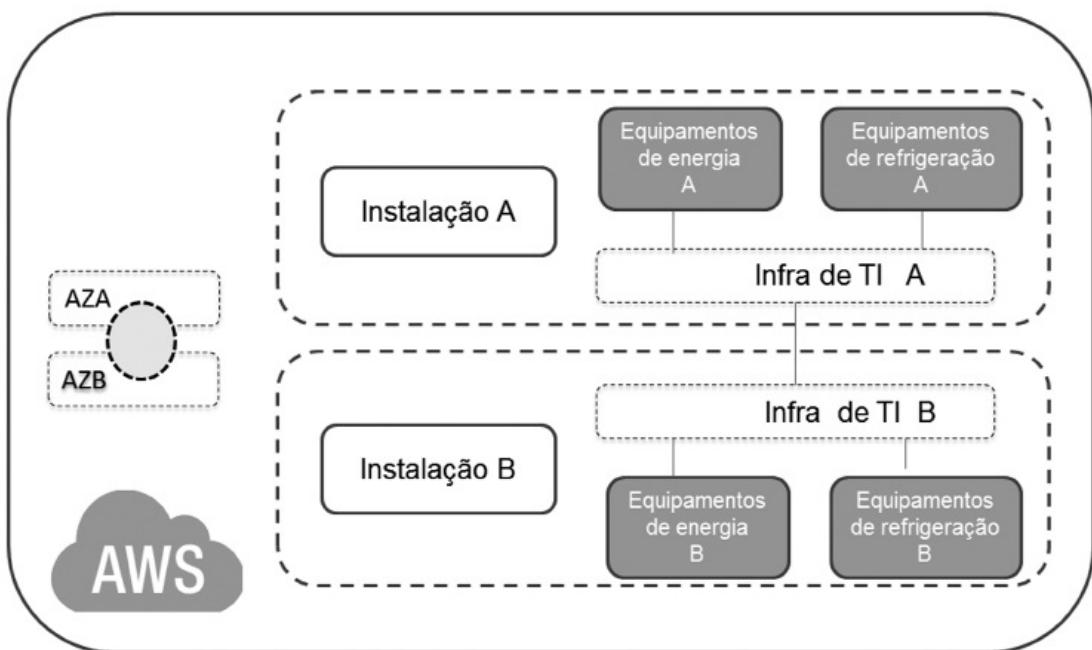


Figura 2-6 Detalhe da zona de disponibilidade

A **Figura 2-7** ilustra regiões e zonas de disponibilidade no contexto da AWS.

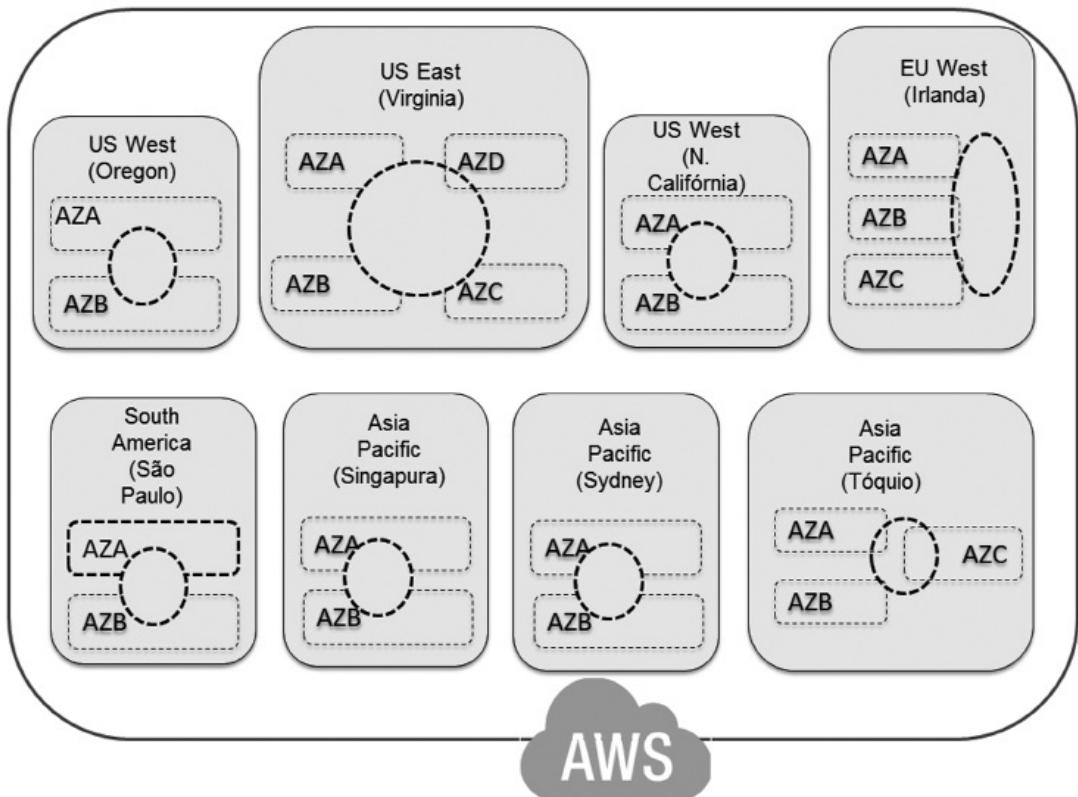


Figura 2-7 Regiões e zonas de disponibilidade

A **Figura 2-8** ilustra como uma arquitetura pode ser construída para utilizar duas zonas de disponibilidade em uma mesma região. O ELB distribui a carga entre duas arquiteturas idênticas de instâncias montadas em cima de duas zonas de disponibilidade.

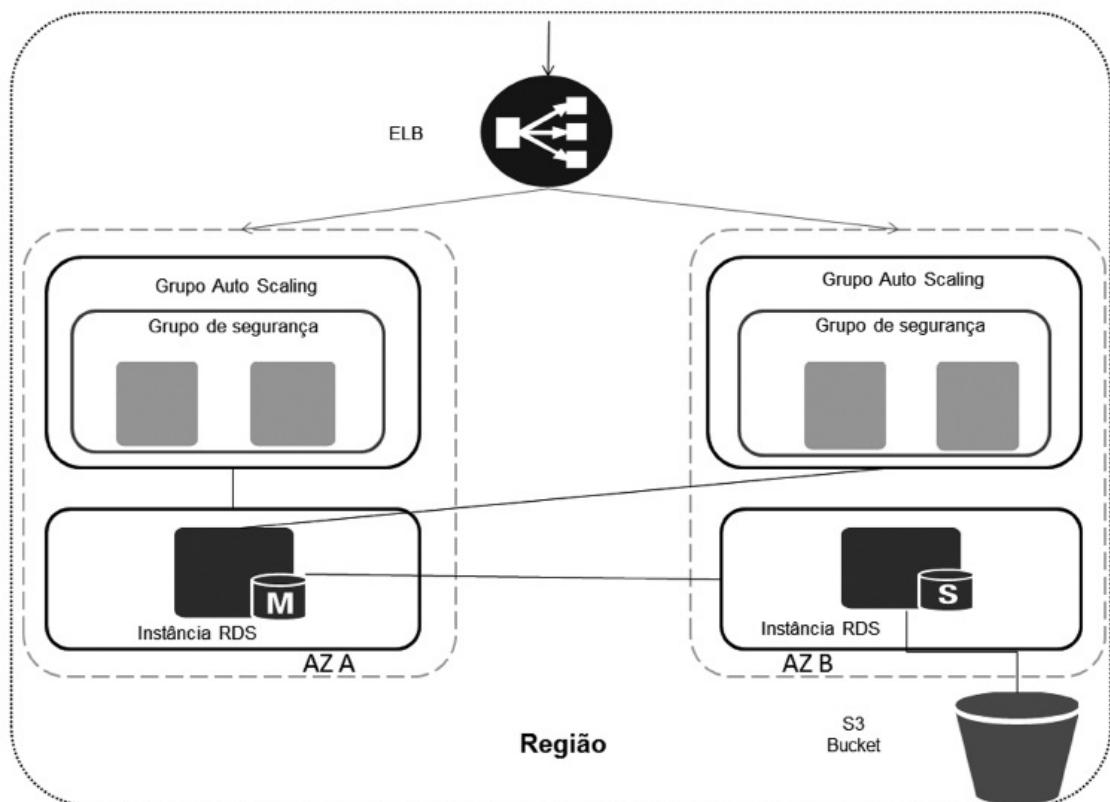


Figura 2-8 Diferentes zonas de disponibilidade utilizadas por uma mesma aplicação

### 2.3.3.1. DATACENTERS tradicionais *versus* DATACENTERS AWS

Um DATACENTER é um conjunto integrado de componentes de alta tecnologia que permitem fornecer serviços de infraestrutura de TI de valor agregado, tipicamente processamento e armazenamento de dados, em larga escala, para qualquer tipo de organização.

Os DATACENTERS e suas conexões de rede formam a infraestrutura da AWS. O DATACENTER consiste de dois núcleos principais: a infraestrutura (também conhecida como Tecnologia de Operação – TO) e a TI (Tecnologia da Informação – TI). A TO inclui instalações físicas, energização e refrigeração. A TI inclui servidores, storage e rede. O gerenciamento inclui aspectos da TO e da TI.

Os DATACENTERS são alimentados por energia e conectados ao mundo externo por uma rede de telecomunicações (upload e download). O objetivo final é alimentar adequadamente a carga de TI.

Independentemente da AWS, o mercado de DATACENTERS deverá crescer nos próximos anos a taxas muito expressivas. Estudo feito por Christian Belady da Microsoft, em 2011, mostra que em 2020 o mercado de DATACENTERS deverá ser de US\$ 218 bilhões, mesmo considerando a redução do preço do DATACENTER por watt ocorrido pelo uso de novas tecnologias e da adoção da modularidade em novos projetos.

Para efeito didático, os DATACENTERS podem ser divididos em três grandes blocos: instalações (inclui instalações físicas, equipamentos de energia e refrigeração), gerenciamento e TI, conforme ilustra a **Figura 2-9**.

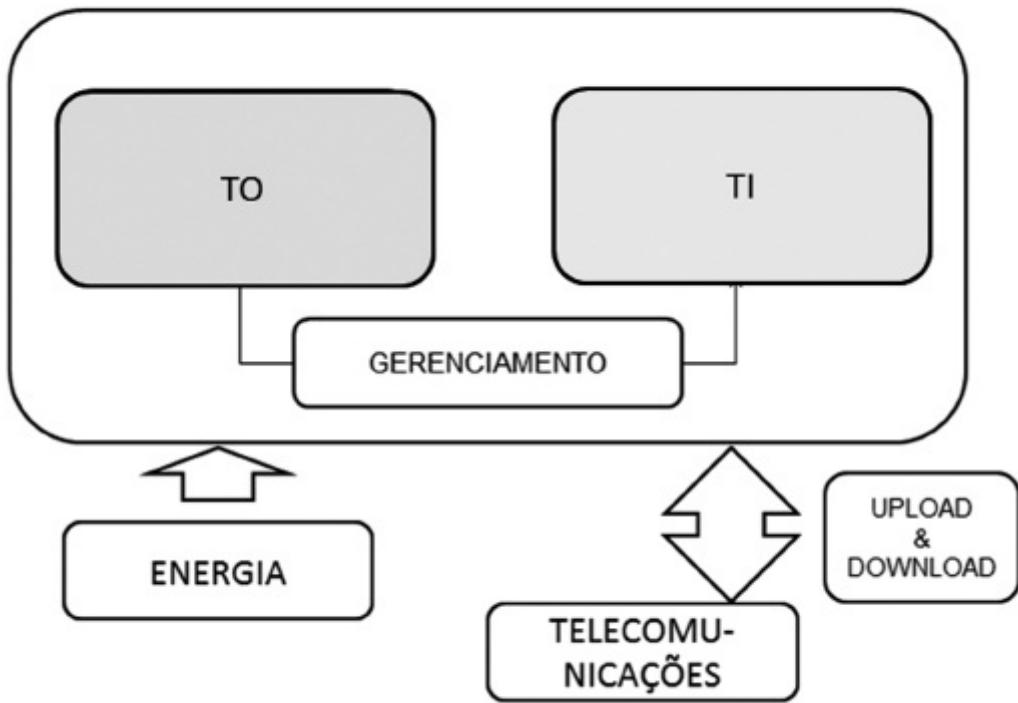


Figura 2-9 Componentes do DATACENTER

A construção de um DATACENTER tradicional envolve o projeto e a implementação das seguintes subpartes:

- **Célula estanque.** Célula estanque e eclusa com paredes e portas corta-fogo.
- **Piso elevado.** Piso elevado para célula estanque, eclusa, telecomunicações, depósito e NOC e corredor de acesso à eclusa.
- **Subsistema de climatização.**
  - Sistema de climatização de precisão redundante para a célula estanque.
  - Sistema de climatização de conforto redundante e que suporte funcionamento ininterrupto (24x7) para as áreas de telecomunicações e no-break.
  - Sistema de climatização de conforto para as demais áreas.
- **Subsistema de provimento ininterrupto de energia elétrica.**
  - Sistema de provimento ininterrupto de energia elétrica para célula estanque, telecomunicações e NOC (*Network Operating Center*), incluindo fornecimento e instalação de gerador, chaves de comutação, UPS redundante, quadros de distribuição, disjuntores, tomadas e fiação necessários.
  - Circuitos de energia elétrica para as demais áreas, incluindo

fornecimento e instalação de quadros de distribuição, disjuntores, tomadas e fiação necessários.

- **Subsistema de detecção e combate a incêndio.**

- Sistema de detecção precoce e combate a incêndio com uso de gás inerte para a célula estanque.
- Sistema de detecção e combate a incêndio com uso de extintores apropriados (eletricidade e mobiliário) para as demais áreas.

- **Subsistema de segurança física.**

- Controle biométrico nas portas de acesso, sendo: uma na eclusa, uma no NOC, uma na sala de telecom e uma na sala de no-break.
- Vigilância por meio de câmeras de vídeo dos pontos de acesso e dos interiores da célula estanque e das áreas de telecom, NOC e no-break.

- **Subsistema de cabeamento estruturado de dados e racks.**

- Cabeamento estruturado, telefonia e rede, composto por cabos de fibra óptica e cobre para todas as áreas.
- Cabeamento vertical composto por cabos de fibra óptica e cobre, interligando o DATACENTER aos centros de distribuição da rede (*wiring closet*).
- Racks padronizados para servidores de rede, equipamentos de telecomunicação e demais equipamentos que compõem a célula estanque e a área de telecomunicações.

O papel principal das subpartes e dos componentes do DATACENTER é possibilitar o atingimento do nível de serviço adequado para cada aplicativo. A ideia central de um projeto de DATACENTER é oferecer níveis de serviço de acordo com a criticidade dos aplicativos, ao mesmo tempo em que é eficiente em termos energéticos. A disponibilidade de DATACENTERS pode ser referenciada e auditada pelo Uptime (<http://www.uptimeinstitute.com>) em camadas (*tiers*) de 1 (menos disponível) a 4 (mais disponível).

Os DATACENTERS consideram como aspectos cruciais de projeto:

- Desempenho.
- Disponibilidade.
- Escalabilidade.
- Segurança.
- Gerenciabilidade.

Outro aspecto importante a ser considerado em projetos de DATACENTERS é a eficiência energética. A eficiência do DATACENTER bem

pouco tempo atrás era medida unicamente em termos de indicadores vinculados a disponibilidade e desempenho. Com os aspectos ambientais sendo cada vez mais considerados, o aumento dos custos de energia e a limitação no fornecimento de energia por parte de alguns provedores de energia é natural que os gerentes de infraestrutura de TI repensem a estratégia para o DATACENTER e considerem o aspecto da eficiência energética nas diversas escolhas que precisam fazer, incluindo engenharia, equipamentos, tecnologias e a própria operação. Na Amazon não é diferente.

O Green Grid (<http://www.thegreengrid.org>), organização criada e mantida por empresas de TI, desenvolveu uma métrica tática já bem utilizada para ajudar a definir a eficiência energética dos DATACENTERS, o PUE (*Power Usage Effectiveness*). A fórmula a seguir relaciona o PUE com a eficiência energética.

$$\text{PUE} = \text{Energia Total da Instalação/Energia dos Equipamentos de TI}$$

A energia total da instalação é a energia aferida pelo medidor que alimenta o DATACENTER. A energia consumida engloba todos os equipamentos de TI incluindo KVMs, monitores e estações de gerenciamento. No caso dos DATACENTERS da Amazon, James Hamilton, vice-presidente da AWS, em palestra recente, estimou para um dos DATACENTERS AWS um PUE da ordem de 1,45.

O desempenho na nuvem depende em certa forma dos componentes físicos do DATACENTER. Ou seja, para que uma instância tenha um excelente desempenho, o servidor do DATACENTER físico deve ter excelente desempenho também. O mesmo vale para a disponibilidade. Isso reflete nos padrões de nível de serviço. A **Figura 2-10** reflete a relação entre níveis de serviço do DATACENTER e da nuvem.

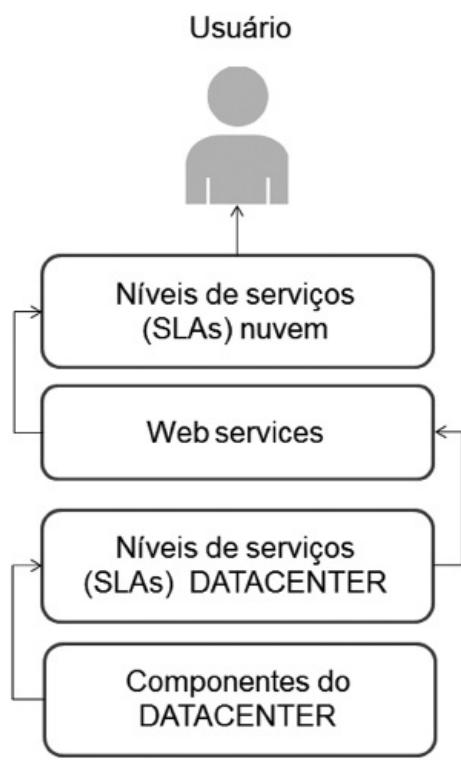


Figura 2-10 Níveis de serviço

No caso da Amazon, os DATACENTERS dos clientes são DATACENTERS lógicos dentro dos DATACENTERS físicos. Em cada zona de disponibilidade podem ser montados vários DATACENTERS lógicos, lembrando que a capacidade da nuvem é finita. A vantagem é que todos os custos associados com as subpartes e os componentes correm por conta da AWS. O DATACENTER lógico só paga em uso e pode ser desligado a qualquer momento.

A Amazon faz investimentos maciços em DATACENTERS e na sua infraestrutura global de rede. Mas mesmo assim os DATACENTERS não são à prova de falha. A Amazon sugere que o projeto da aplicação considere sempre a possibilidade de falha. Aspectos de recuperação de desastres serão tratados no capítulo 13. No capítulo 10 os aspectos gerais de arquitetura serão considerados.

A disponibilidade para a aplicação pode ser melhorada com o uso de mais de um DATACENTER como parte da solução de infraestrutura, conforme mostrado anteriormente. A escalabilidade também é um requisito fundamental. A segurança oferecida aos clientes depende da segurança física implementada. Aspectos de gerenciabilidade são também muito importantes.

O livro “DATACENTER: Componente Central da Infraestrutura”, publicado pela Brasport e de minha autoria, trata o assunto com profundidade.

### **2.3.3.2. Virtualização tradicional *versus* virtualização com AWS**

A virtualização pode ser conceituada de duas principais formas:

- É o particionamento de um servidor físico (normalmente do tipo x86) em vários servidores lógicos. A Figura 2-11 ilustra que cinquenta servidores são substituídos por um servidor virtual que equivale a cinquenta máquinas virtuais (*Virtual Machines – VMs*) ou instâncias virtuais e, portanto, a taxa de consolidação é de 50:1.

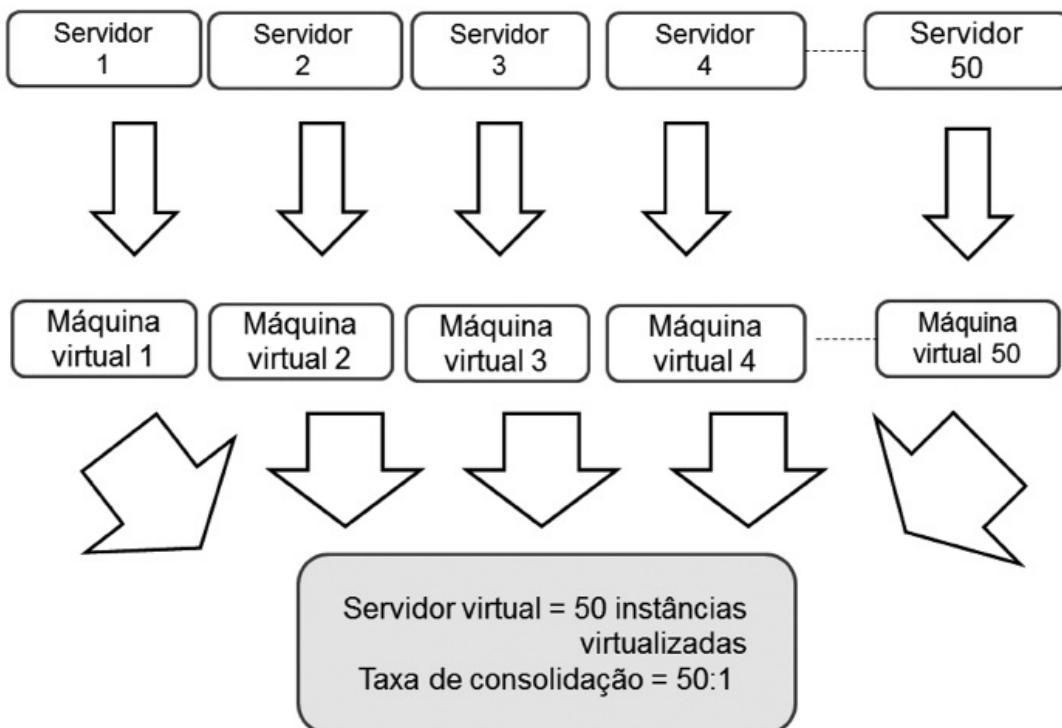


Figura 2-11 O que é a virtualização

- É uma camada de abstração entre o hardware e o software que protege o acesso direto do software aos recursos físicos do hardware. A virtualização permite que a camada de software (aplicações e sistema operacional) seja isolada da camada de hardware. Normalmente a virtualização é implementada por um software. A Figura 2-12 ilustra o conceito.

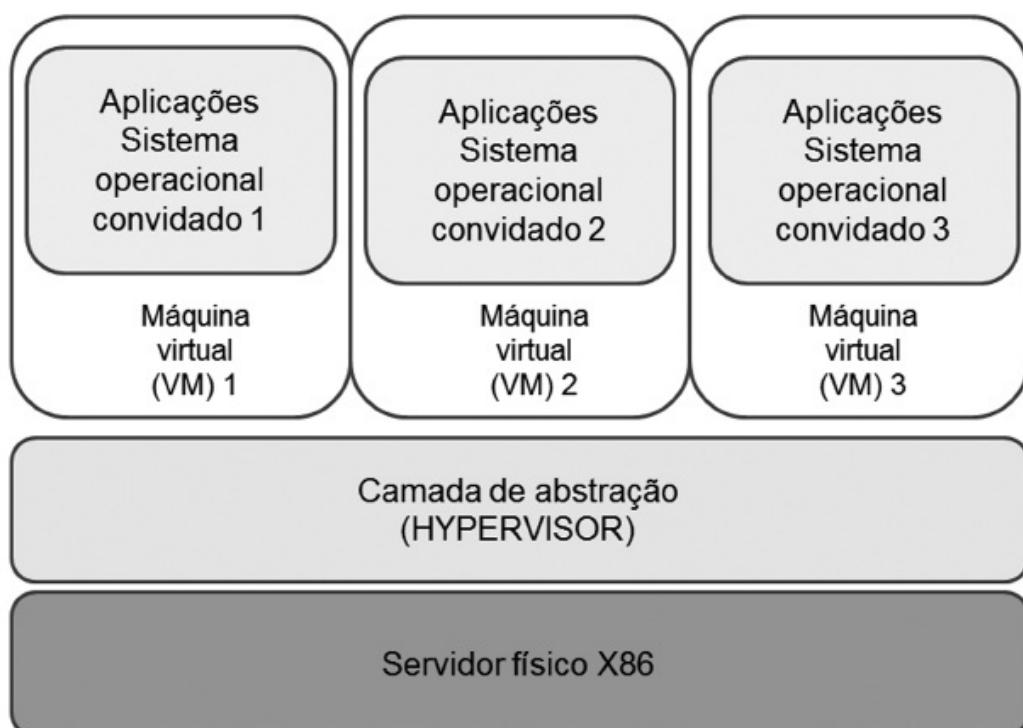


Figura 2-12 Arquitetura da virtualização

O software que implementa a virtualização normalmente é do tipo *hypervisor*. O *hypervisor* é conhecido como monitor de máquina virtual (*Virtual Machine Monitor* – VMM) e representa a camada de abstração que entrega para o sistema operacional convidado um conjunto de instruções de máquinas equivalente ao processador físico. O servidor físico virtualizado pode então rodar várias instâncias virtuais.

A virtualização simplifica o gerenciamento, permite flexibilizar e ampliar o poder de processamento. Funcionalidades contidas nos softwares de virtualização também permitem melhorar a disponibilidade e a recuperação de desastres de ambientes de TI de uma maneira mais simples e com menor custo quando comparado a formas tradicionais.

Com a virtualização cada VM utiliza um sistema operacional e suas respectivas aplicações. Diversas VMs podem coexistir no mesmo servidor físico.

A virtualização é a essência da nuvem AWS. Em comparação com ferramentas de gerenciamento de virtualização de fabricantes como a VMware, as características do console de gerenciamento AWS são até limitadas. No entanto, o fato de que a AWS faz com que as máquinas físicas sejam completamente invisíveis para o administrador do sistema traz enormes vantagens. A instância EC2 (o servidor virtual da AWS) utiliza uma versão personalizada do XEN Hypervisor. A virtualização com XEN utilizada pela AWS permite excelente nível de isolamento entre as VMs e consistência de desempenho. A paravirtualização, um tipo de virtualização que privilegia o desempenho utilizado pelo XEN, otimiza o desempenho para o I/O.

A Amazon também utiliza virtualização baseada em hardware (*Hardware Virtual Machine* – HVM), um dos modos de virtualização do XEN, para instâncias do tipo GPU.

A nuvem AWS fornece um outro nível de abstração quando comparada à virtualização propriamente dita. A **Figura 2-13** ilustra esta característica.

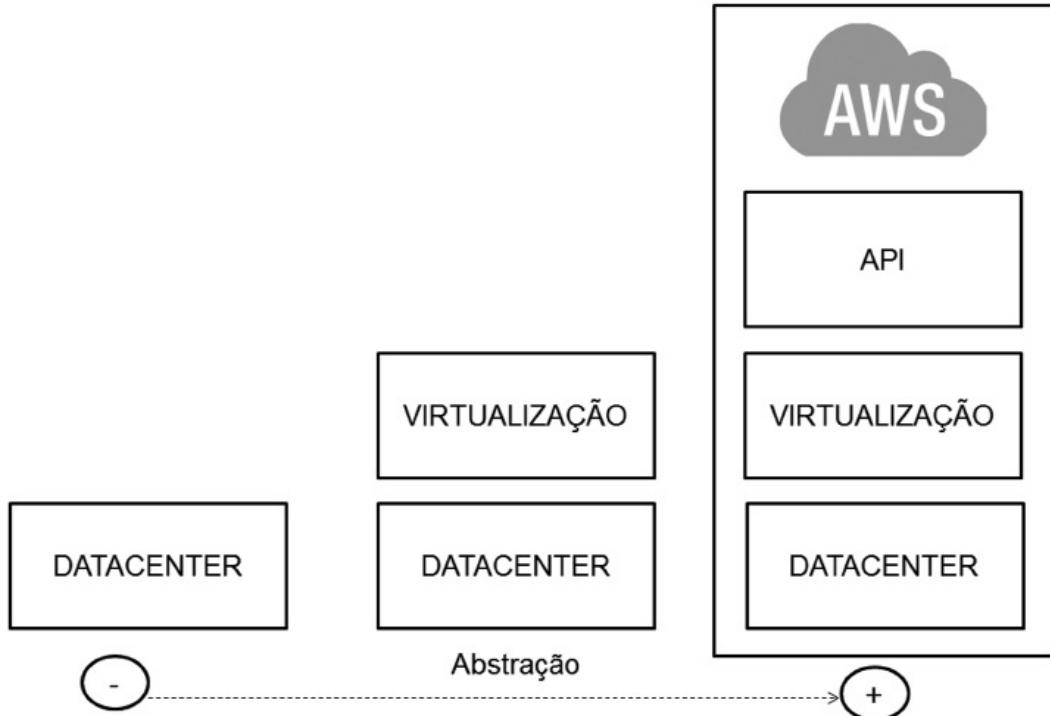


Figura 2-13 DATACENTER, virtualização e AWS

O livro “Virtualização: Componente Central do DATACENTER”, publicado pela Brasport e de minha autoria, trata com profundidade o assunto.

#### 2.3.4. Localização das regiões e da rede CloudFront

A **Figura 2-14** ilustra as regiões da AWS e a rede de pontos de presença (AWS Edge) utilizadas pelo CloudFront. A ilustração ainda não considera a região de Sydney. O serviço CloudFront utiliza esta rede otimizada para entrega de conteúdos estáticos (*downloading*) e *streaming*.

O link <http://aws.amazon.com/pt/about-aws/globalinfrastructure/> permite visualizar a posição atual da infraestrutura global AWS.

<b>América do Norte</b>			
	<b>Região Leste dos EUA (Norte da Virgínia)</b> Zonas de disponibilidade do EC2: 4 Lançado em 2006	<b>Região Oeste dos EUA (Oregon)</b> Zonas de disponibilidade do EC2: 3 Lançado em 2011	<b>Região AWS GovCloud (EUA)</b> Zonas de disponibilidade do EC2: 2 Lançado em 2011
	Região Oeste dos EUA (Norte da Califórnia) Zonas de disponibilidade do EC2: 2 Lançado em 2009		
<b>Locais do AWS Edge</b>	Ashburn, VA (2)	Dallas/Fort Worth, TX	Jacksonville, FL
	Nova York, NY (2)	Newark, NJ	Palo Alto, CA
	South Bend, IN	St. Louis, MO	Los Angeles, CA (2)
			Niami, FL
			San Jose, CA
			Seattle, WA
<b>Europa / Oriente Médio / África</b>			
	<b>Região UE (Irlanda)</b> Zonas de disponibilidade do EC2: 3 Lançado em 2007		
	Região Ásia-Pacífico (Cingapura)		
	Zonas de disponibilidade do EC2: 2 Lançado em 2010		
	Região Ásia-Pacífico (Tóquio)		
	Zonas de disponibilidade do EC2: 2 Lançado em 2011		
<b>Locais do AWS Edge</b>	Amsterdã, Holanda	Dublin, Irlanda	Frankfurt, Alemanha (2)
	Paris, França	Estocolmo, Suécia	Londres, Inglaterra (2) Milão, Itália
<b>Ásia-Pacífico</b>			
	<b>Região Ásia-Pacífico (Cingapura)</b> Zonas de disponibilidade do EC2: 2 Lançado em 2010		
	Região Ásia-Pacífico (Tóquio)		
	Zonas de disponibilidade do EC2: 2 Lançado em 2011		
<b>Locais do AWS Edge</b>	Hong Kong, China	Osaka, Japão	Cingapura (2)
			Tóquio, Japão
<b>América do Sul</b>			
	<b>Região de São Paulo</b> Zonas de disponibilidade do EC2: 2 Lançado em 2011		
	<b>Locais do AWS Edge</b>		
	São Paulo, Brasil		

Figura 2-14 Regiões e localizações de pontos de presença

## 2.3.5. Utilização do CloudFront

O CloudFront é um web service para disponibilização de conteúdo que se integra a outros serviços AWS e possibilita aos desenvolvedores um caminho fácil para distribuir conteúdo para usuários finais com baixa latência e transferência de dados em alta velocidade. Os *requests* dos objetos são automaticamente roteados para a localização mais próxima, otimizando o desempenho.

O CloudFront funciona com qualquer servidor de origem que armazene os arquivos. Como outros Amazon Web Services, não existem contratos nem gastos mensais para seu uso – paga-se apenas pela quantidade de conteúdo que realmente é distribuída através do serviço.

No CloudFront, os objetos são organizados em distribuições. Uma distribuição específica é a localização da versão original dos objetos. Uma distribuição tem um nome de domínio cloudfont.net exclusivo que pode ser usado para fazer referência aos objetos por meio da rede dos pontos de presença. Pode-se também mapear o próprio nome de domínio para a distribuição. Pode-se criar distribuições para fazer download do conteúdo usando os protocolos HTTP ou HTTPS, ou reproduzir o conteúdo usando o protocolo RTMP, comum em *streaming*.

### 2.3.5.1. Distribuição por *downloading*

Quando um cliente solicita uma página usando o nome de domínio, o CloudFront determina o melhor ponto de presença para atender o conteúdo. Se um ponto de presença não tem uma cópia do arquivo que o usuário final solicitar, o CloudFront obterá uma cópia no servidor de origem e a salvará no ponto de presença para que esteja disponível para solicitações futuras.

O conteúdo pode ser distribuído usando o protocolo HTTP ou HTTPS. Por padrão, a distribuição aceitará solicitações em um destes protocolos. No entanto, o conteúdo pode ser distribuído somente através de uma conexão HTTPS. Quando o CloudFront precisa obter um arquivo do servidor de origem, ele usará o mesmo protocolo que foi utilizado para a solicitação do usuário final. Por exemplo, se um usuário final solicitar um arquivo usando o HTTPS que já não está em um ponto de presença, o CloudFront utilizará o HTTPS para obter o arquivo na sua origem.

O CloudFront pode ser utilizado para:

- **Hospedagem dos componentes do website acessados com mais frequência:** os sites geralmente contêm um pequeno número de arquivos que são compartilhados por todas as páginas no site. Devido ao fato de que os mesmos arquivos são acessados com muita frequência, eles são muito suscetíveis a serem mantidos no cache em pontos de presença e são ideais para o CloudFront.

- **Distribuição de software:** o CloudFront é uma boa escolha para desenvolvedores e software que desejam distribuir aplicativos, atualizações ou outros softwares para download por usuários finais. As elevadas taxas de transferência de dados do CloudFront aceleram o download dos aplicativos, melhorando a experiência do cliente e reduzindo custos.
- **Publicação de arquivos de mídia popular:** se o aplicativo envolve conteúdo de mídia – áudio ou vídeo – que é acessado com frequência, o CloudFront possibilita obter preços mais baixos e velocidades maiores para as transferências de dados.

### 2.3.5.2. Distribuições por *streaming*

O CloudFront permite criar “distribuições em *streaming*” projetadas para transmitir conteúdo de mídia avançada. As distribuições em *streaming* disponibilizam conteúdo aos usuários finais em tempo real – os usuários finais assistem o conteúdo à medida que este é transmitido. As distribuições em *streaming* utilizam o protocolo RTMP, em vez dos protocolos HTTP ou HTTPS.

*Streaming* se refere a protocolos proprietários que são usados para fornecer áudio e vídeo para usuários finais da internet. Esses protocolos são diferentes do protocolo HTTP, usado para distribuir páginas da web e outros conteúdos, pois protocolos de *streaming* fornecem conteúdo em tempo real. Realizar *streaming* oferece vários benefícios potenciais para usuários finais: pode oferecer aos usuários mais controle sobre sua experiência de visualização. O *streaming* pode ajudar a reduzir custos, pois oferece apenas partes de um arquivo de mídia a que os usuários realmente assistem.

O CloudFront utiliza o Adobe Flash Media Server para distribuições em *streaming*. A **Figura 2-15** ilustra a ideia de distribuição com o CloudFront.

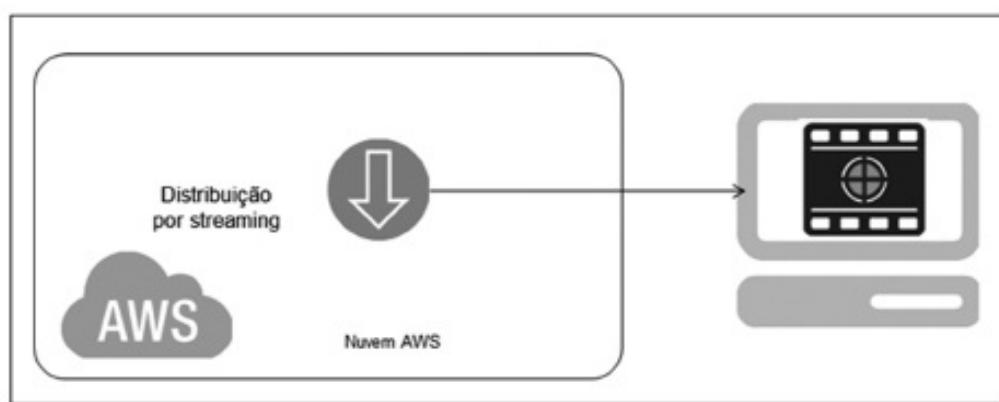


Figura 2-15 Distribuição por *streaming*

As distribuições por *streaming* usam todos os pontos de presença da rede CloudFront. Portanto, o conteúdo é transmitido a partir de um servidor

que está próximo dos usuários finais. Não há nenhum tipo de cobrança adicional pelo *streaming* de conteúdo; paga-se pela quantidade de dados distribuídos de acordo com as taxas estabelecidas.

### 2.3.6. Migração de recursos entre regiões

Recentemente a AWS publicou o artigo “Migrating AWS Resources to a New Region”, que faz considerações importantes sobre a migração de recursos entre regiões. Esse aspecto é importante porque em determinado momento pode surgir uma nova opção mais conveniente em termos de custos e características técnicas, e pode fazer sentido migrar recursos entre regiões aproveitando a nova oferta.

O artigo reforça que só o console de gerenciamento, o AWS Identity and Access Management (IAM) e o CloudWatch são independentes da região.

## 2.4. Serviços por região

É importante reforçar que nem todos os serviços AWS estão disponíveis em todas as regiões. A **Tabela 2-3** ilustra a disponibilidade dos serviços por região sem considerar a região de Sydney.

**Tabela 2-3 Serviços por região**

Serviços	Virginia	Oregon	N. Califórnia	Irlanda	Singapura	Tóquio	São Paulo	Gov Cloud
Amazon Simple Notification Service (SNS)	✓	✓	✓	✓	✓	✓	✓	✓
Elastic Load Balancing	✓	✓	✓	✓	✓	✓	✓	✓
Amazon ElastiCache	✓	✓	✓	✓	✓	✓	✓	✓
AWS Storage Gateway	✓	✓	✓	✓	✓	✓	✓	✓
Amazon DynamoDB	✓	✓	✓	✓	✓	✓	✓	✓
AWS Import/Export	✓	✓	✓	✓	✓	✓		
AWS Elastic Beanstalk	✓			✓	✓			
High Performance Computing	✓							
Amazon Simple Email Service (SES)	✓							
Amazon CloudSearch	✓							
Amazon SWF	✓							

## 2.5. Suporte AWS

### 2.5.1. Tipos de suporte

O suporte AWS cuida da relação com o usuário e permite monitorar o cumprimento dos acordos de nível de serviço estabelecidos com o cliente, com apoio de ferramentas como o *Service Health Dashboard* (SHD). O gerenciamento da nuvem AWS será visto no capítulo 9.

Há quatro níveis de suporte disponíveis para usuários da AWS:

- Suporte básico.
- Suporte desenvolvedor.
- Suporte negócios.
- Suporte empresa.

A **Figura 2-16** ilustra a página obtida em <http://aws.amazon.com/pt/premiumsupport/signup>, que permite fazer a opção por um plano de suporte AWS para desenvolvedor ou para negócios. Para suporte corporativo deve-se entrar em contato com a AWS.

### Selecionar um plano AWS Support

Analise a definição de preço e a funcionalidade das opções disponíveis do AWS Support e selecione um plano usando os botões abaixo, ou saiba mais sobre o AWS Support.

**Selecionar um plano AWS Support:**  Desenvolvedor  
 Negócios

[Continuar](#)

Para o AWS Enterprise Support, entre em contato conosco.

Figura 2-16 Selecionar um plano AWS Support

A **Tabela 2-4** ilustra os recursos do suporte AWS para cada categoria.

**Tabela 2-4 Recursos do suporte AWS**

	Básico	Desenvolvedor	Negócios	Empresa
Atendimento ao cliente – 24 horas por dia, sete dias por semana, 365 dias por ano	✓	✓	✓	✓
Fóruns de suporte	✓	✓	✓	✓
Documentação, <i>white papers</i> , orientações sobre as melhores práticas	✓	✓	✓	✓
Acesso ao suporte técnico	Suporte a verificações de saúde	E-mail (horário comercial local)	Telefone, chat, e-mail (24 horas por dia, sete dias por semana)	Telefone, chat, e-mail, TAM (24 horas por dia, sete dias por semana)
Contatos designados		1	5	Ilimitado
Tempo de resposta mais rápida		Doze horas	Uma hora	Quinze minutos
Supporte à arquitetura		Blocos de criação	Orientação de casos de uso	Arquitetura do aplicativo

	Básico	Desenvolvedor	Negócios	Empresa
Orientação sobre as melhores práticas		✓	✓	✓
Ferramentas de diagnóstico do lado do cliente		✓	✓	✓
Encaminhamento direto para engenheiros de suporte sênior			✓	✓
Suporte de software terceirizado – beta			✓	✓
AWS Trusted Advisor – beta			✓	✓
Gerenciamento de evento de infraestrutura			Entrar em contato para ter acesso à definição de preço	✓
Acesso direto ao gerente técnico de conta (TAM)				✓
<i>White-glove case routing</i>				✓
Análises empresariais de gerenciamento				✓

Os conceitos utilizados na tabela são descritos a seguir:

- **Contatos designados.** Adiciona nomes à conta da empresa para que mais pessoas possam entrar em contato com a AWS em vez de fazer perguntas através de um único usuário.
- **Suporte à arquitetura.**
  - **Blocos de criação:** orientação sobre como usar todos os produtos, recursos e serviços da AWS. Inclui orientação sobre práticas recomendadas.
  - **Análises de serviço:** análise de produtos, recursos e serviços da AWS específicos que estão sendo usados. Inclui orientação básica quanto ao alinhamento do uso atual da AWS para práticas recomendadas.
  - **Orientação de casos de uso:** orientação sobre quais produtos, recursos e serviços da AWS usar para melhor oferecer suporte aos casos de uso específicos. Inclui orientação quanto à otimização de produtos AWS e à configuração para atender a necessidades específicas.
  - **Arquitetura de aplicativo:** parceria consultiva oferecendo suporte a casos de uso e aplicativos específicos. Inclui análises de projeto. A equipe de suporte a clientes empresa inclui um gerente técnico de conta (*Technical Account Manager* – TAM) dedicado e acesso a arquiteturas de soluções da AWS. A proposta do TAM é fornecer expertise técnico para uma grande gama de serviços da AWS e obter um entendimento detalhado do caso de uso e da arquitetura utilizada pelo cliente. Os TAMs funcionam como arquitetos de

soluções AWS para ajudar a lançar novos projetos e fornecer recomendações de melhores práticas.

- **Suporte de software terceirizado.** O suporte negócios ou empresa permite fazer perguntas sobre sistemas operacionais e softwares.
- **Trusted Advisor.** Monitora serviços de infraestrutura da AWS, identifica configurações do cliente, compara-as às melhores práticas conhecidas e, em seguida, notifica os clientes sobre onde podem existir as oportunidades de economizar dinheiro, melhorar o desempenho do sistema ou falhas de segurança próximas. As verificações do recurso *Trusted Advisor* são comunicadas aos clientes através de casos proativos, relatórios programados regularmente e através da interação direta com os engenheiros. A Amazon reforça que a lista de melhores práticas disponíveis no *Trusted Advisor* é atualizada continuamente e expandida com os aprendizados mais recentes. O *Trusted Advisor* usa uma biblioteca de melhores práticas sobre como utilizar a AWS, também inspeciona o ambiente AWS do cliente e faz recomendações. As verificações são agrupadas em três famílias: tolerância a falhas, segurança e otimizações de custo. A Amazon ressalta que o *Trusted Advisor* não faz nenhum tipo de acesso aos dados do cliente. A **Figura 2-17** ilustra a arquitetura do *Trusted Advisor*.

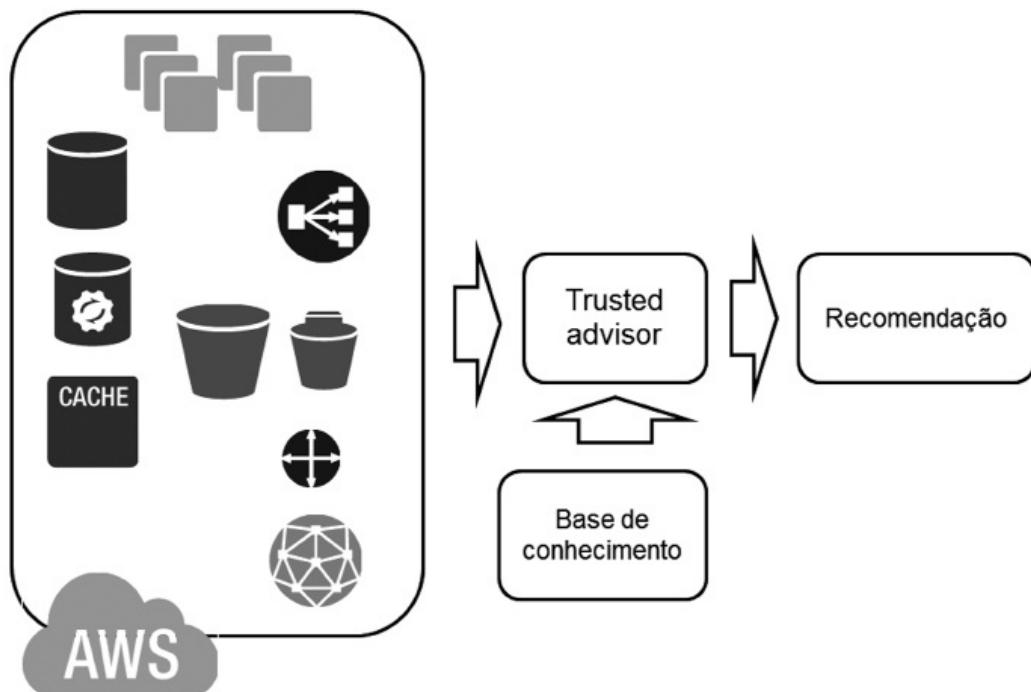


Figura 2-17 *Trusted Advisor*

- **White-Glove Case Routing.** Os casos apresentados pelos clientes Platinum serão reconhecidos e encaminhados diretamente para os engenheiros especialmente treinados para garantir uma solução rápida e precisa para as questões críticas.
- **Análise empresarial de gerenciamento.** É possível obter ajuda com o planejamento de infraestrutura. Para o suporte empresarial o TAM realiza avaliações de desempenho, relatórios de métricas, colabora em lançamentos e conecta o usuário aos arquitetos de soluções AWS.

### 2.5.2. Service Health Dashboard (SHD)

A AWS publica o status de funcionamento de todos os seus web services no SHD. Os status dos serviços podem ser obtidos em <http://status.aws.amazon.com/>.

Durante a interrupção de um serviço, a equipe da AWS atualiza o status a cada 15-30 minutos, enquanto trabalha para solucionar o problema.

A **Figura 2-18** ilustra um típico painel para a América do Sul.

North America	South America	Europe	Asia Pacific	Contact Us		
Current Status			Details			
	Amazon CloudFront	Service is operating normally.				
	Amazon CloudWatch (Sao Paulo)	Service is operating normally.				
	Amazon Elastic Compute Cloud (Sao Paulo)	Service is operating normally.				
	Amazon Elastic MapReduce (Sao Paulo)	Service is operating normally.				
	Amazon ElastiCache (Sao Paulo)	Service is operating normally.				
	Amazon Relational Database Service (Sao Paulo)	Service is operating normally.				
	Amazon Route 53	Service is operating normally.				
	Amazon Simple Notification Service (Sao Paulo)	Service is operating normally.				
	Amazon Simple Queue Service (Sao Paulo)	Service is operating normally.				
	Amazon Simple Storage Service (Sao Paulo)	Service is operating normally.				
	Amazon SimpleDB (Sao Paulo)	Service is operating normally.				
	Amazon Virtual Private Cloud (Sao Paulo)	Service is operating normally.				
	AWS CloudFormation (Sao Paulo)	Service is operating normally.				
	AWS Direct Connect (Sao Paulo)	Service is operating normally.				
	AWS Management Console	Service is operating normally.				
	AWS Storage Gateway (Sao Paulo)	Service is operating normally.				

Service is operating normally   
 Performance issues   
 Service disruption   
 Informational message

Figura 2-18 SHD – Status

Também existe a opção de verificar o histórico dos serviços e das interrupções nos últimos 35 dias. A **Figura 2-19** ilustra o painel com o histórico dos serviços para a região América do Sul.

	<a href="#"></a>	Oct 12	Oct 11	Oct 10	Oct 9	Oct 8	Oct 7	Oct 6	<a href="#"></a>
Amazon CloudFront									
Amazon CloudWatch (Sao Paulo)									
EC2 (Sao Paulo)									
EMR (Sao Paulo)									
Amazon ElastiCache (Sao Paulo)									
RDS (Sao Paulo)									
Amazon Route 53									
SNS (Sao Paulo)									
SQS (Sao Paulo)									
S3 (Sao Paulo)									
Amazon SimpleDB (Sao Paulo)									
VPC (Sao Paulo)									
AWS CloudFormation (Sao Paulo)									
AWS Direct Connect (Sao Paulo)									
AWS Management Console									
AWS Storage Gateway (Sao Paulo)									

Figura 2-19 SHD – Histórico do status

## 2.6. Referências bibliográficas

<http://aws.amazon.com>

<http://aws.typepad.com>

<http://aws.typepad.com/brasil/>

<http://gigaom.com/cloud/will-amazon-outage-ding-cloud-confidence/>

Veras, Manoel. **DATACENTER**: Componente Central da Infraestrutura de TI. Brasport, 2009.

Veras, Manoel. **Virtualização**: Componente Central do Datacenter. Brasport, 2011.

# 3. Identidade e Acesso

## 3.1. Introdução

O controle do acesso é peça-chave para o funcionamento com segurança da AWS e consiste na verificação da existência de uma identidade.

Frederick Chong da Microsoft, em seu artigo “Gerenciamento de Identidade e Acesso”, afirma que a identidade digital consiste de um identificador, credenciais, atributos principais e atributos específicos. A identidade é a informação que identifica exclusivamente o objeto desta identidade dentro de um contexto. Credenciais são dados privados ou públicos que podem ser usados para provar a autenticidade de uma solução de identidade. Atributos principais são dados que ajudam a descrever a identidade. Atributos específicos de contexto são dados que ajudam a descrever a identidade, mas utilizados em contextos específicos.

Chong define o gerenciamento de identidade e acesso (IAM) como o processo, as tecnologias e as políticas para gerenciar identidades digitais e controlar como essas identidades digitais podem ser usadas para acessar recursos. O autor reforça que o IAM é um processo dinâmico e deve-se buscar sempre encontrar um novo equilíbrio entre tecnologias, diretrizes e processos.

Gerenciar a identidade e assim controlar o acesso de vários usuários ligados a uma conta AWS é uma funcionalidade importante. O *Identity and Access Management* (IAM) veio resolver isso na arquitetura AWS.

O IAM é um web service que permite que clientes gerenciem usuários e suas permissões na AWS. O foco do IAM são organizações com múltiplos usuários ou sistemas que utilizam web services AWS, como o EC2 e o S3.

Este capítulo trata dos conceitos relacionados ao acesso do IAM e das suas várias funcionalidades. Outros aspectos da segurança serão tratados no capítulo 12.

## 3.2. Conceitos

### 3.2.1. Usuário, identidade e credencial

Os conceitos de usuário, identidade e credenciais são fundamentais para entender como a AWS faz o controle de acesso.

- **Usuário:** pode ser um cliente ou entidade (*host/aplicação*) que quer obter um serviço de um provedor. Um usuário pode ser uma pessoa utilizando um web service. Um requisito necessário para que um usuário possa obter um recurso de um provedor é possuir uma

identidade que o represente.

- **Identidade:** é uma representação de uma entidade em certo contexto. Identidade consiste de identificadores e credenciais de usuários.
- **Credencial:** é um atestado de qualificação, competência ou autoridade, expedido para um indivíduo por terceiros com autoridade relevante para tal. Um exemplo é o certificado X.509, assinado por uma autoridade certificadora.

### 3.2.2. Autenticação e autorização (controle de acesso)

O controle de acesso é baseado nos conceitos de autenticação e autorização. Estes conceitos são distintos e acontecem em sequência.

- **Autenticação:** é o processo de verificar a existência de uma identidade. Para que o processo se inicie é necessário que o usuário forneça uma credencial válida para o método de autenticação utilizado pela entidade autenticadora. De posse da credencial válida, a entidade verifica se existe uma identidade cuja credencial seja a fornecida pelo usuário. Se sim, o usuário é autenticado.
- **Autorização:** é a verificação pela entidade autenticadora das permissões de acesso a um recurso requisitado pelo usuário identificado.

### 3.2.3. Federação e SSO

Federação e SSO (*Single Sign-On*) são conceitos relacionados ao controle de acesso na nuvem.

- **Federação** é um conjunto de organizações que cooperam entre si de acordo com regras de confiança preestabelecidas para a autenticação de usuários e compartilhamento de recursos. A federação é utilizada, por exemplo, para simplificar a autenticação e a autorização de usuários a recursos e aplicações entre domínios parceiros.
- **SSO** se refere à maneira como um usuário de um serviço compartilhado pode ser autenticado sem ter que utilizar diversas credenciais. O SSO é muito útil em ambientes cooperados, facilitando o processo de autenticação.

A **Figura 3-1** ilustra a utilização do conceito de federação de clouds e SSO. O usuário é autenticado com um SSO e possui autorização para utilizar recursos das três nuvens da federação de clouds.

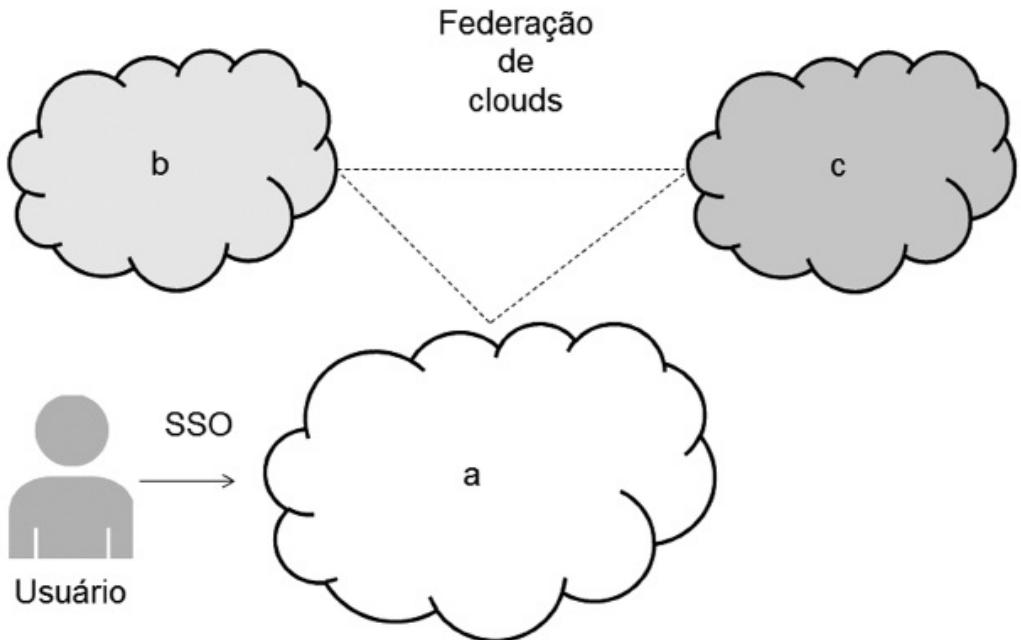


Figura 3-1 Federação de clouds e SSO

### 3.3. Identidade e acesso tradicional *versus* identidade e acesso na AWS

Sistemas de Gerenciamento de Identidade (SGI) tratam do controle de acesso (autenticação e autorização) para cada tipo de usuário, aumentando a segurança e reduzindo os custos através da administração, do controle e da auditoria de forma centralizada, assegurando que somente as pessoas que tiverem o perfil e a alçada adequados terão acesso às informações e aos serviços oferecidos.

Um SGI tradicional, além de autenticar o usuário, também controla o seu acesso, impedindo a consulta e execução de serviços não autorizados. Todas as transações de negócios disponíveis são cadastradas e associadas a um ou mais perfis que definem os direitos do usuário para a execução das transações.

Quando o usuário entra nas áreas de acesso restrito, é solicitado que ele faça o login. A autenticação do usuário pode ser feita de várias formas, incluindo user-ID e senhas, certificados digitais, tokens, identificação biométrica (impressão digital/íris) ou uma combinação destes. Neste momento ele é autenticado e suas ações são controladas pelo sistema.

A implantação de um SGI agiliza a construção de aplicações de nuvem mais seguras, proporcionando aos desenvolvedores uma camada que garante privacidade e segurança, reduzindo a necessidade de codificar a segurança em aplicações múltiplas e ao mesmo tempo garantindo aderência às políticas de segurança e controle de acesso estabelecido.

Com um SGI é possível:

- Autenticar a identificação dos usuários antes de permitir o acesso.
- Implementar SSO baseado em AD, bancos de dados corporativos ou uma combinação destes.
- Controlar não somente o acesso, mas o que cada usuário pode ver ou executar.
- Criar trilhas de auditoria, claras e confiáveis.
- Permitir a implementação de aplicações na nuvem de uma forma flexível, escalável e segura, através de:
  - Criação de políticas de controle de acesso diferenciadas.
    - Delegação de poderes de administração de permissões de acesso e alçadas.
  - Administração descentralizada de usuários, transações, perfis e alçadas.
  - Criação de autocadastramento de usuários, associando-os automaticamente a perfis e domínios adequados a cada tipo de usuário, negócio e interesse.
  - Gerar informações de acesso e utilização para outros sistemas.
    - Permitir a criação de uma política de controle de acesso e perfis de usuários corporativos, eliminando a necessidade da administração de vários sistemas de segurança.

Com o uso da nuvem, reforça-se a necessidade de ter sistemas de gerenciamento de identidade e acesso mais bem estruturados e integrados, pois os riscos aumentam. Empresas de nuvem como a Amazon AWS estão investindo muito nesta área e buscando o aumento das funcionalidades e a simplificação da autenticação (*Single Sign-On – SSO*). O IAM pode ser considerado um SGI.

## 3.4. Credenciais de segurança

As credenciais de segurança (*Security Credentials*) da conta podem ser acessadas diretamente na página principal do site AWS. A **Figura 3-2** ilustra esta opção.



Figura 3-2 Security Credentials

A AWS oferece três tipos de credenciais de segurança:

- **Credenciais de acesso:** chaves de acesso, certificados X.509 e pares de chave.
- **Credenciais de conexão:** endereço de e-mail, senha e *multi-factor authentication*.
- **Identificadores de conta:** ID da conta da AWS e ID de usuário canônico.

A **Tabela 3-1** ajuda a identificar as credenciais de segurança necessárias para a tarefa que se deseja desempenhar na AWS.

**Tabela 3-1 Uso das credenciais da AWS**

Se quer...	Use...
Fazer um <i>request</i> REST ou uma consulta do tipo QUERY para um produto AWS.	Chaves de acesso ( <i>access keys</i> ).
Fazer um <i>request</i> SOAP para um serviço AWS.	Certificados X.509 (exceto S3, que requer <i>access keys</i> ).
Acessar páginas seguras no site AWS ou no console de gerenciamento AWS.	Endereço de e-mail e senha com opção <i>multifactor authentication</i> .
Uso do CLI para o EC2.	Certificados X.509.
Lançar ou conectar-se a uma instância EC2.	<i>EC2 key pairs</i> (pares de chaves).
Compartilhar uma AMI EC2 ou um <i>snapshot</i> Amazon EBS.	<i>Account ID</i> da conta AWS.
Acesso aos fóruns de discussão ou site AWS Premium Support.	Endereço de e-mail e senha.

### 3.4.1. Credenciais de acesso

Cada tipo de credencial de acesso é explicado a seguir.

#### 3.4.1.1. Chaves de acesso

Devem ser utilizadas obrigatoriamente chaves de acesso para tornar as solicitações REST e QUERY seguras para qualquer API de serviço da AWS. A Amazon cria uma chave de acesso quando a conta é criada.

Para proteção, a Amazon reforça que nunca se deve compartilhar chaves

de acesso secretas com ninguém. Além disso, a prática recomendada do setor instrui uma rotação frequente destas chaves.

A **Figura 3-3** ilustra as chaves de acesso de uma conta AWS específica.

The screenshot shows the 'Chaves de acesso' (Access Keys) section of the AWS IAM console. It displays a single access key entry:

Criada	ID de chave de acesso	Chave de acesso secreta	Status
August 6, 2011	AKIAIVWABZLNW25LLJOA	Mostrar	Ativo (Tornar inativo)

Below the table, there is a link to 'Criar uma nova Chave de acesso' (Create a new Access Key). A note at the bottom states: 'Para sua proteção, você nunca deve compartilhar suas chaves de acesso secretas com ninguém. Além disso, a prática recomendada do setor instrui uma rotação frequente de chaves' (For protection, you never share your access keys with anyone. Additionally, the industry standard recommends frequent key rotation). A link 'Saiba mais sobre Chaves de acesso' (Learn more about Access Keys) is also present.

Figura 3-3 Chaves de acesso

#### 3.4.1.2. Certificados X.509

Deve-se utilizar obrigatoriamente certificados X.509 para proteger as solicitações do protocolo SOAP para APIs de serviço da AWS.

**Exceções:** o Amazon S3 exige chaves de acesso para *requests* SOAP.

A **Figura 3-4** ilustra o certificado X.509 de uma conta AWS.

The screenshot shows the 'Certificados X.509' (X.509 Certificates) section of the AWS IAM console. It displays two certificate entries:

Criada	Certificado X.509	Status
July 30, 2012	cert-Y4WPXCCFFDG2EQ2VUSKYXVYYCBDNRF7.pem (Baixar)	Ativo (Tornar inativo)
October 14, 2012	cert-CXJDCUVWIS55UYKYIIMKAWYO2DY5NY2U.pem (Baixar)	Ativo (Tornar inativo)

Below the table, there is a link to 'Exiba seus certificados excluídos' (View deleted certificates). A note at the bottom states: 'Para sua proteção, a AWS não solicita sua chave privada nem a armazena em arquivo. Você também nunca deve compartilhar sua chave privada com ninguém. Além disso, a prática recomendada do setor instrui a rotação frequente do certificado.' (For protection, AWS does not request or store your private key. You never share your private key with anyone. Additionally, the industry standard recommends frequent certificate rotation). A link 'Saiba mais sobre Certificados X.509' (Learn more about X.509 Certificates) is also present.

Figura 3-4 Certificados X.509

#### 3.4.1.3. Pares de chave

Há dois tipos de pares de chave usados no momento com serviços específicos da AWS — um para o CloudFront e outro para o EC2.

Para proteção, a AWS não solicita chave privada nem a armazena em

arquivo. A Amazon reforça que nunca se deve compartilhar a chave privada com ninguém.

A Figura 3-5 ilustra os pares de chave de uma conta específica AWS.

The screenshot shows the AWS IAM 'Pairs of keys' section. It includes tabs for 'Chaves de acesso', 'Certificados X.509', and 'Pares de chave'. Below the tabs, a note states: 'Há dois tipos de pares de chave usados no momento com serviços específicos da AWS — um para o Amazon CloudFront e outro para o Amazon EC2. Eles são explicados a seguir.' The 'Pares de chave do Amazon CloudFront' section contains a table:

Criada	ID do Par de chaves	Status
July 30, 2012	APKAIZZIANHZS5KFQ2LA (Baixar a Chave pública)	Ativo (Tornar inativo)

Below the table are links: 'Criar um novo par de chaves | Upload Your Own Key Pair' and 'Saiba mais sobre Pares de chave do Amazon CloudFront'. The 'Pares de chaves do Amazon EC2' section also has a note about not storing private keys and links to 'Acesse os pares de chave do Amazon EC2 usando o Console de Gestão da AWS' and 'Saiba mais sobre Pares de chave do Amazon EC2'.

Figura 3-5 Pares de chave

### **3.4.2. Credenciais de conexão**

Para se conectar aos sites e aplicativos, a AWS exige o fornecimento do endereço de e-mail e senha cadastrados na Amazon. Além disso, pode-se utilizar a opção autenticação multifator (*Multi-Factor Authentication – MFA*). Cada credencial de conexão é explicada a seguir.

Para efetuar *login* em páginas seguras no site da AWS, no console de gerenciamento AWS, em fóruns de discussão da AWS e no site AWS Premium, basta fornecer o endereço de e-mail e a senha da Amazon.

Para proteção, a Amazon reforça que não se deve compartilhar senha pessoal com ninguém. A prática recomendada do setor instrui a mudança frequente de senhas, e estas devem ter uma combinação de letras e números.

A MFA da AWS é uma credencial opcional que permite aumentar o nível de segurança da conta. Para adicionar essa opção de segurança, deve-se comprar um dispositivo de autenticação compatível junto à Gemalto, um fornecedor autorizado, ou utilizar uma autenticação por software.

A **Figura 3-6** ilustra um dispositivo MFA.



Figura 3-6 Dispositivo MFA

Assim que o dispositivo MFA for ativado, deve-se fornecer um endereço de e-mail, uma senha e um código de autenticação gerado pelo dispositivo sempre que for feita a conexão a páginas seguras no site da AWS e do console de gerenciamento da AWS.

### **3.4.3. Identificadores de conta**

A AWS usa dois tipos de identificadores de conta – ID de usuário canônico e ID de conta da AWS. Esses identificadores são usados para compartilhar recursos entre as contas.

O ID de usuário canônico pode ser usado exclusivamente para recursos do S3 como *buckets* ou arquivos.

O ID de conta pode ser usado para todos os serviços da AWS, exceto o S3. Esses recursos incluem AMIs do Amazon EC2, *snapshots* do Amazon EBS, filas do Amazon SQS etc.

## **3.5. IAM**

### **3.5.1. Introdução**

O *Identity and Access Management* (IAM) permite controlar com

segurança o acesso aos serviços e recursos da AWS pelos usuários. Ele permite criar e gerenciar usuários e também permite a concessão de acesso a recursos da AWS para usuários geridos fora da AWS, mas que fazem parte do diretório corporativo (mencionados como “usuários federados”). O IAM oferece maior segurança, flexibilidade e controle ao usar a AWS e é essencialmente um sistema de autorização.

Com o IAM o gerenciamento do uso dos recursos da AWS pode ser centralizado, e múltiplos usuários (pessoa, sistemas ou aplicação) possuem suas próprias credenciais, controlados e bilhetados por uma conta AWS.

A **Figura 3-7** ilustra como o IAM trabalha. Ele gerencia como os recursos podem ser acessados. Na figura, linhas cheias e linhas tracejadas indicam níveis diferentes de permissões.

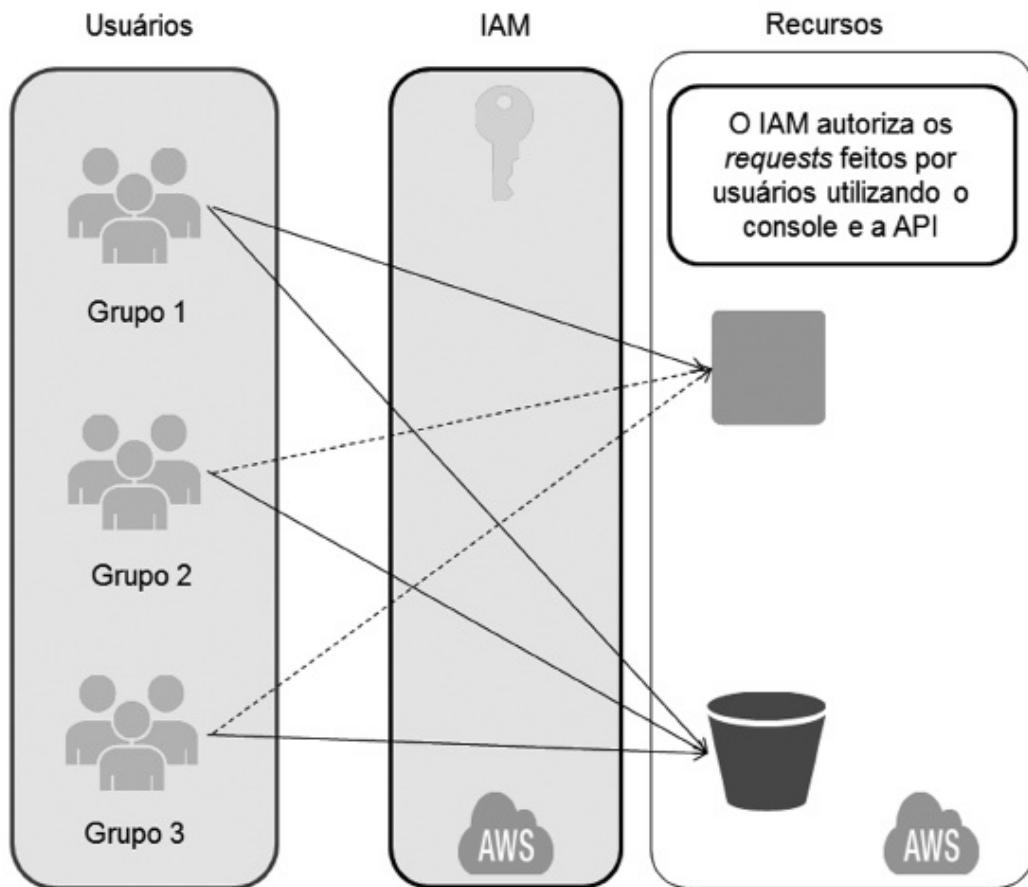


Figura 3-7 Utilização do IAM

A **Figura 3-8** ilustra o IAM *dashboard*. Usuários (*Users*), grupos (*Groups*), permissões (*Permissions*) e funções (*Roles*) são a essência do IAM.

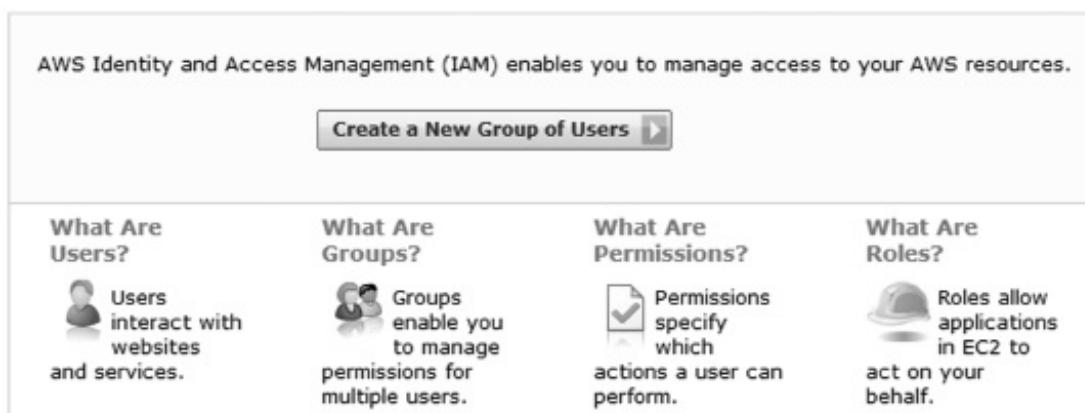


Figura 3-8 IAM dashboard

Sem o IAM só existem duas possibilidades: ou a empresa cria uma conta para cada usuário e cada sistema ou a empresa compartilha as credenciais de segurança entre múltiplos usuários e sistemas (duas opções inadequadas em diversas situações). Ou seja, o IAM é fundamental para o uso da AWS por empresas.

O IAM está disponível com as seguintes interfaces: console AWS, interface de linha de comando (*Command Line Interface – CLI*), API do tipo QUERY e bibliotecas existentes. Na página principal do IAM também é fornecido um status da segurança indicando a opção de gerenciar o dispositivo MFA e se a política de senha está habilitada. A **Figura 3-9** ilustra a opção do status de segurança, mostrando que a opção MFA para a conta raiz está desabilitada e a política de senha está habilitada.



Figura 3-9 Security status

O detentor da conta AWS pode utilizar vários dispositivos MFA. Esses dispositivos podem, em seguida, ser designados a usuários individuais do IAM por meio de APIs, ferramentas de linha de comando ou por meio do console do IAM. A **Figura 3-10** ilustra a caixa de diálogo para ativação do dispositivo MFA.



Figura 3-10 Manage MFA device

Em termos gerais, o IAM permite:

- **Criar identidades para usuários:** adicionar usuários a uma conta da AWS.
- **Organizar usuários em grupos:** criar grupos para facilmente gerenciar permissões para vários usuários em uma conta da AWS.
- **Centralizar o controle do acesso aos usuários:** controlar quais operações os usuários podem desempenhar, como acessar recursos e APIs de serviço da AWS específica.
- **Acesso condicional de usuários:** adicionar condições para controlar como os usuários podem usar a AWS, como horário do dia, seu endereço IP de origem ou se estão usando SSL.
- **Criar e designar credenciais de segurança:** designar credenciais de segurança aos usuários e rotacionar e/ou revogar essas credenciais conforme desejado.
- **Criar credenciais de segurança temporárias:** solicitar credenciais de segurança temporárias com expiração configurável e permissões para usuários, usuários federados ou aplicativos.
- **Ter uma única conta AWS:** ter uma única conta AWS para atividade AWS de todos os usuários de uma organização.

Se a empresa já utiliza a nuvem AWS, a migração para o IAM pode ser fácil ou potencialmente mais desafiadora, dependendo de como a organização atualmente aloca seus recursos AWS:

- Se houver uma única conta AWS para a organização, pode-se migrar facilmente para o IAM, porque todos os recursos da organização AWS já estão juntos em uma única conta da AWS.
- Se a organização tiver várias contas AWS, com cada conta da AWS pertencente a uma divisão na organização, e estas divisões não precisam compartilhar recursos ou usuários, migrar para o IAM é fácil. Cada divisão pode manter sua própria conta da AWS e usa-se o IAM separadamente para cada divisão. Neste caso pode-se utilizar o faturamento consolidado baseado em uma única conta.

- Se a organização tem várias contas AWS que não representam limites lógicos entre divisões, migrar não é uma tarefa trivial. Se for necessário criar contas AWS para compartilhar recursos e usuários comuns, migrar para o IAM será um desafio. Será necessário mover os recursos que precisam ser compartilhados para que estejam sob a posse de uma única conta AWS. No entanto, não há uma forma automática de transferir os recursos AWS de uma conta AWS para outra. Será necessário criar esses recursos novamente sob uma conta única da AWS.

### **3.5.2. Conceitos**

#### **3.5.2.1. Conta (account)**

Uma conta é a primeira entidade criada ao iniciar uma relação com a AWS. A conta AWS controla centralmente todos os recursos criados sob seu guarda-chuva e paga-se por todas as atividades AWS realizadas utilizando esses recursos. Quaisquer permissões criadas para usuários ou grupos dentro da conta AWS não se aplicam à própria conta AWS.

A conta AWS tem permissão para fazer tudo com todos os recursos de conta da AWS. Desta forma, a conta AWS é um conceito similar ao UNIX raiz ou superusuário.

O detentor da conta AWS pode gerenciar usuários, grupos, credenciais de segurança e permissões. Além disso, a permissão poderá ser concedida para usuários individuais fazerem chamadas para APIs do IAM para gerenciar outros usuários. Por exemplo, um usuário administrador poderá ser criado para gerenciar usuários para uma corporação – uma prática recomendada. Quando um usuário tiver recebido permissão para gerenciar outros usuários, ele poderá fazê-lo por meio das APIs do IAM, ferramentas de linha de comando ou por meio do console.

#### **3.5.2.2. Usuário e grupo (users e groups)**

Um usuário é uma identidade exclusiva reconhecida pelos serviços e aplicativos da AWS. Semelhante a um usuário de login em um sistema operacional como Windows ou UNIX, um usuário tem um nome exclusivo e pode se identificar usando credenciais de segurança familiares, como senha ou chave de acesso. Um usuário pode ser um indivíduo, sistema ou aplicativo exigindo acesso a serviços da AWS. O IAM oferece suporte a usuários gerenciados no sistema de gerenciamento de identidades da AWS (usuários do IAM) e também permite conceder acesso a recursos da AWS para usuários geridos fora da AWS no diretório corporativo (usuários federados).

Aspectos importantes:

- Um usuário pode fazer solicitações para web services como o S3 e o EC2. A capacidade do usuário de acessar APIs do web service está

sob controle e responsabilidade da conta AWS sob a qual está definida – um usuário pode obter permissão para acessar qualquer ou todos os serviços da AWS que foram integrados ao IAM e para os quais a conta da AWS foi inscrita. Se for permitido, um usuário terá acesso a todos os recursos na conta da AWS.

- Se a conta AWS tiver acesso a recursos de uma conta diferente da AWS, então seus usuários poderão acessar os dados nessas contas da AWS. Quaisquer recursos da AWS criados por um usuário estão sob controle e são pagos pela conta da AWS. Um usuário não pode se inscrever independentemente nos serviços ou recursos de controle da AWS.
- Os usuários são entidades globais, como uma conta AWS é hoje. Não é exigido que nenhuma região seja especificada ao definir permissões de usuário. Os usuários podem usar os serviços da AWS em qualquer região geográfica.
- Os usuários podem fazer solicitações a serviços da AWS usando credenciais de segurança. A capacidade de um usuário de chamar serviços da AWS é regida por permissões explícitas – como padrão, eles não têm capacidade de chamar APIs de serviço em nome da conta.
- Um usuário pode ter qualquer combinação de credenciais compatível com a AWS – chaves de acesso, certificado X.509, senha para logins no aplicativo da web ou dispositivo de autenticação multifator (MFA). Isso permite que os usuários interajam com a AWS de várias formas – um funcionário pode ter uma chave de acesso da AWS e uma senha; um sistema de software poderá ter somente uma chave de acesso da AWS para fazer chamadas programáticas; e um prestador de serviços externo poderá ter somente um certificado X.509 para usar a interface da linha de comando do EC2.
- As chaves de acesso de um usuário IAM podem ser habilitadas e desabilitadas por meio das APIs do IAM, ferramentas de linha de comando ou por meio do console do IAM. Desabilitar as chaves de acesso significa que o usuário não poderá acessar programaticamente os serviços da AWS.
- As chaves de acesso e os certificados X.509 de um usuário podem ser gerenciados e rotacionados programaticamente por meio de APIs do IAM, ferramentas de linha de comando ou por meio do console do IAM.
- Os nomes de usuários são strings ASCII exclusivos dentro de uma determinada conta da AWS. O detentor da conta da AWS pode designar nomes usando qualquer nomenclatura que selecionar,

incluindo endereços de e-mail.

- Ainda não é possível definir cotas de uso quanto aos usuários do IAM. Todas as cotas estão na conta da AWS. Por exemplo, uma conta da AWS hoje tem um limite de vinte instâncias do EC2 e a conta tem um limite de vinte instâncias do EC2 após o uso do IAM para criar usuários. O número de instâncias está associado à conta da AWS, não aos usuários individuais definidos na conta.
- Uma senha inicial pode ser definida para um usuário do IAM por meio do console, ferramentas de linha de comando ou por meio de APIs do IAM.

A **Figura 3-11** ilustra a opção de definição da senha para um usuário utilizando o console de gerenciamento.

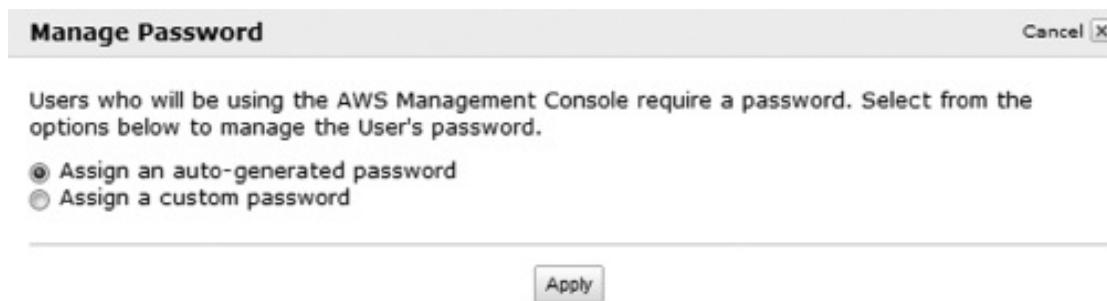


Figura 3-11 Manage password – duas etapas

Um usuário deve se conectar usando uma URL de login fornecida pelo administrador do sistema da conta da AWS. Os usuários que pertencem a uma conta AWS efetuam login usando esta URL, específica para tal.

A URL pode conter o ID da conta ou um apelido (*alias*). O apelido é um nome que o administrador do sistema define para identificar a conta mais facilmente. A URL parece-se com isto:

<https://account-identifier.signin.aws.amazon.com/console/ec2>

Onde substitui-se “account-identifier” pelo ID da conta de doze dígitos ou pelo apelido criado para a conta. O apelido é opcional. Se não quiser criar um apelido, então poderá utilizar o ID da conta para efetuar o login. Sempre é possível criar o apelido posteriormente.

A **Figura 3-12** ilustra o apelido (*alias*) para a conta manoelveras.

## AWS Account Alias



Your AWS Account Alias is **manoelveras**

IAM users sign-in link:

› <https://manoelveras.signin.aws.amazon.com>

[Remove Account Alias](#)

Figura 3-12 Account alias

Os apelidos podem ser criados usando APIs do IAM, ferramentas de linha de comando ou por meio do console do IAM. É possível ter um apelido por conta da AWS.

Na primeira vez que um usuário efetuar login, ele deverá usar a URL específica da conta. Depois disso, ela será armazenada como uma preferência no cookie do navegador do usuário. Isso permite que um usuário retorne para <http://aws.amazon.com> e clique no link “Sign in to the AWS Management Console” (Efetuar login no AWS Management Console) para efetuar login. Se o usuário apagar os cookies do navegador ou usar um navegador diferente, então deverá usar a URL específica da conta.

A **Figura 3-13** ilustra a tela “Amazon Web Services Sign In”.

AWS Account: manoelveras

User Name: MatheusV

Password: [REDACTED]

Sign in using our secure server

Please contact your system administrator if you have forgotten your user credentials.

[Sign in using AWS Account credentials](#)

Figura 3-13 Amazon Web Services Sign In

Os usuários do IAM podem gerenciar suas senhas por meio da caixa de diálogo “My Password” no console do IAM. Os usuários podem acessar essa página selecionando a opção “Security Credentials” na parte superior direita do menu suspenso do console de gerenciamento AWS.

A **Figura 3-14** ilustra a opção “My Password”.

The screenshot shows a web-based password change interface titled "My Password". A message at the top states: "Use this form to change your password. Your new password must meet the requirements defined in the AWS account password policy." Below this, a section titled "Your new password:" lists six requirements for a new password:

- must not be the same as your old password
- must be at least 6 characters long
- must contain at least one symbol (!@#\$%^&\*()\_+=[]{}|`)
- must contain at least one number (0-9)
- must contain at least one uppercase letter (A-Z)
- must contain at least one lowercase letter (a-z)

Below the requirements are three input fields: "Old Password", "New Password", and "Confirm New Password", each with a corresponding label. At the bottom is a "Change Password" button.

Figura 3-14 My password

A **Figura 3-15** ilustra a caixa de diálogo para definição da política de senha para usuários IAM. Os critérios utilizados são:

- Tamanho mínimo da senha.
- Requisitos de complexidade (qualquer combinação de):
  - No mínimo uma letra minúscula.
  - No mínimo uma letra maiúscula.
  - No mínimo um número.
  - No mínimo um símbolo.

A política de senha (*password policy*) será aplicada da próxima vez que um usuário do IAM alterar a senha. Pode-se definir também se os usuários podem mudar sua própria senha. A política se aplica a todos os usuários criados em uma conta da AWS.

A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, go to Using IAM.

Modify your existing password policy below.

Minimum Password Length:

- Require at least one uppercase letter ?
- Require at least one lowercase letter ?
- Require at least one number ?
- Require at least one non-alphanumeric character ?
- Allow users to change their own password ?

Figura 3-15 Política de senha

Um grupo é um conjunto de usuários do IAM. A associação de grupos é gerida como uma lista simples; os usuários podem ser adicionados ou removidos de um grupo. Um usuário pode pertencer a vários grupos. Os grupos não podem pertencer a outros grupos.

Os grupos podem obter a concessão de permissões usando políticas de controle de acesso. Isso facilita a gestão de permissões para um conjunto de usuários em vez de ter de gerenciar permissões para cada usuário individualmente.

Os grupos não têm credenciais de segurança e não podem acessar web services diretamente; eles existem somente para facilitar o gerenciamento de permissões de usuários. Por exemplo, pode-se ter um grupo chamado admin e dar a esse grupo os tipos de permissões que administradores de sistemas geralmente precisam. A **Figura 3-16** mostra a criação de um grupo de usuários chamado “ProjectALPHA”.

Create New Group Wizard Cancel 

GROUP NAME PERMISSIONS REVIEW

Specify a group name. Group names can be edited any time.

**Group Name:**   
Example: Developers or ProjectAlpha  
Maximum 128 characters

Figura 3-16 Create a new group of users – group name

A **Figura 3-17** ilustra a concessão das permissões para os usuários do grupo ProjectALPHA com base no template “Power User Access”.



Figura 3-17 Create a new group – set permissions

A Figura 3-18 ilustra a criação de usuários para o grupo ProjectALPHA.

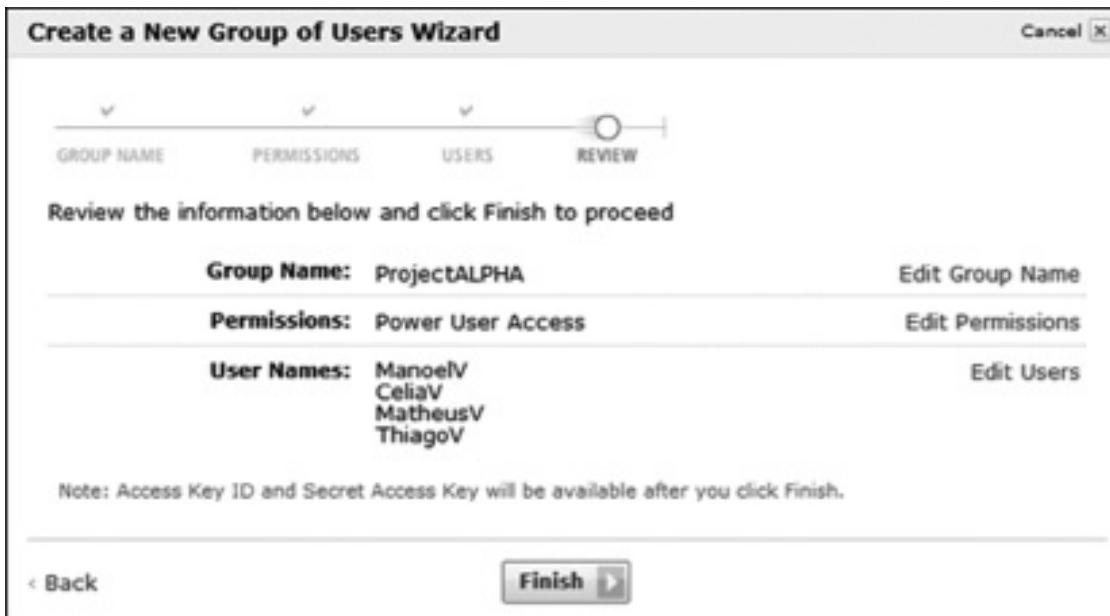


Figura 3-18 Create a new group – create new users

A Figura 3-19 resume a criação do grupo, permissões e usuários.

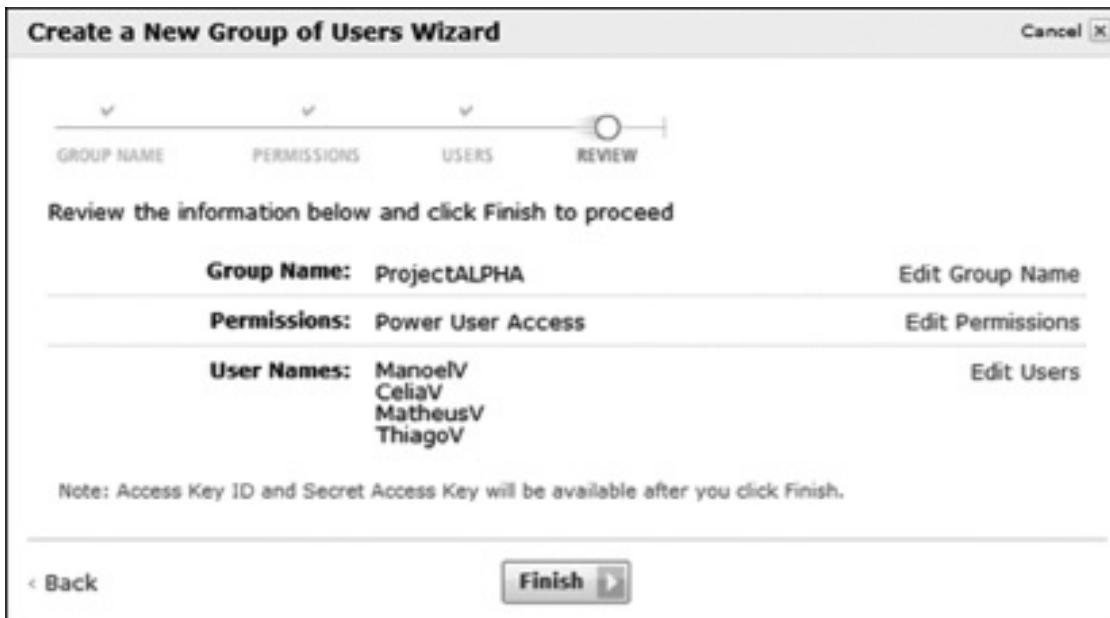


Figura 3-19 Create a new group – review

### 3.5.2.3. Funções (rules)

Uma função do IAM é uma entidade com permissões para fazer solicitações de web services da AWS. As funções do IAM não podem fazer solicitações diretas para web services; elas são consideradas “pressupostas” por entidades autorizadas, como usuários do IAM, aplicativos ou web services, como o EC2.

As funções do IAM para instâncias do EC2 fornecem os seguintes recursos:

- Chaves de acesso da AWS, para uso ao fazer solicitações para web services da AWS, serão automaticamente disponibilizadas em instâncias do EC2 em execução.
- Rotação automática das chaves de acesso da AWS disponibilizadas nas instâncias do EC2.
- Permissões granulares do web service da AWS para aplicativos em execução em instâncias do EC2, que fazem solicitações para web services da AWS.

Um usuário do IAM tem credenciais permanentes e é usado para interagir diretamente com web services AWS. Uma função do IAM não tem quaisquer credenciais e não pode fazer solicitações diretas para web services da AWS.

Um usuário do IAM deve receber duas permissões distintas para executar com êxito instâncias do EC2 com funções:

- Permissão para executar instâncias do EC2.
- Permissão para associar uma função do IAM a instâncias do EC2.

A **Figura 3-20** ilustra o primeiro passo para criar uma função.



Figura 3-20 Create new role – role name

A **Figura 3-21** ilustra a definição das permissões.

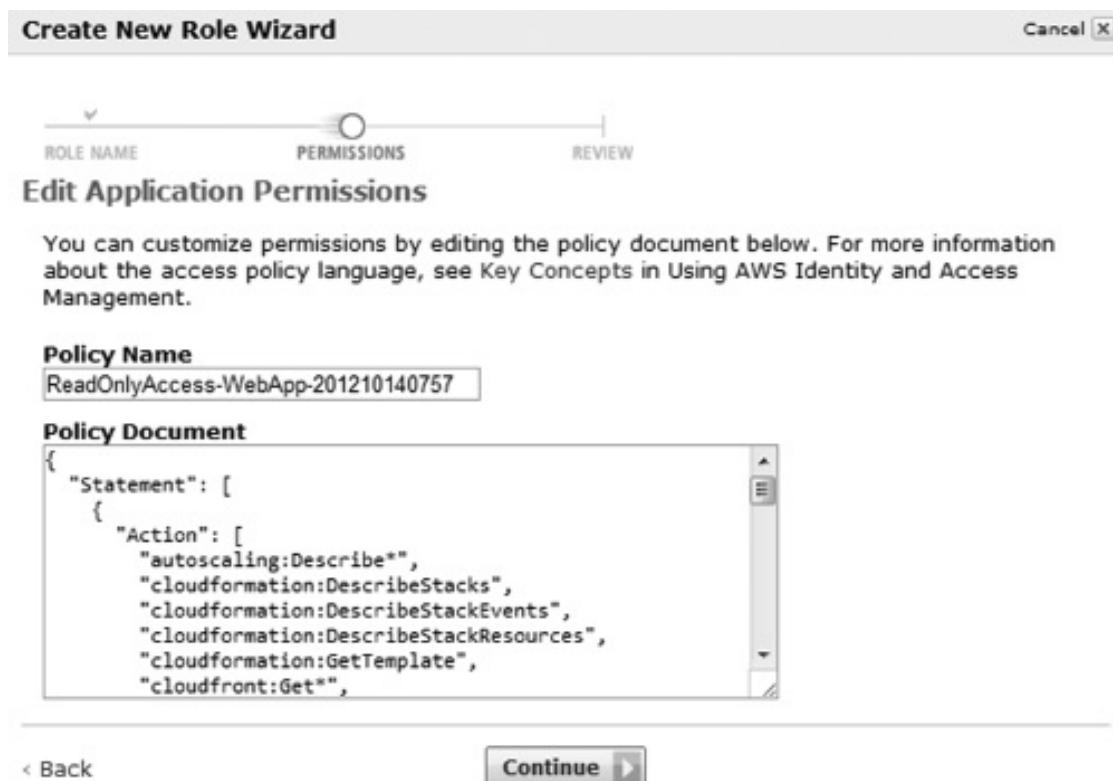


Figura 3-21 Create new role – permissions

A **Figura 3-22** ilustra a opção “Review”.

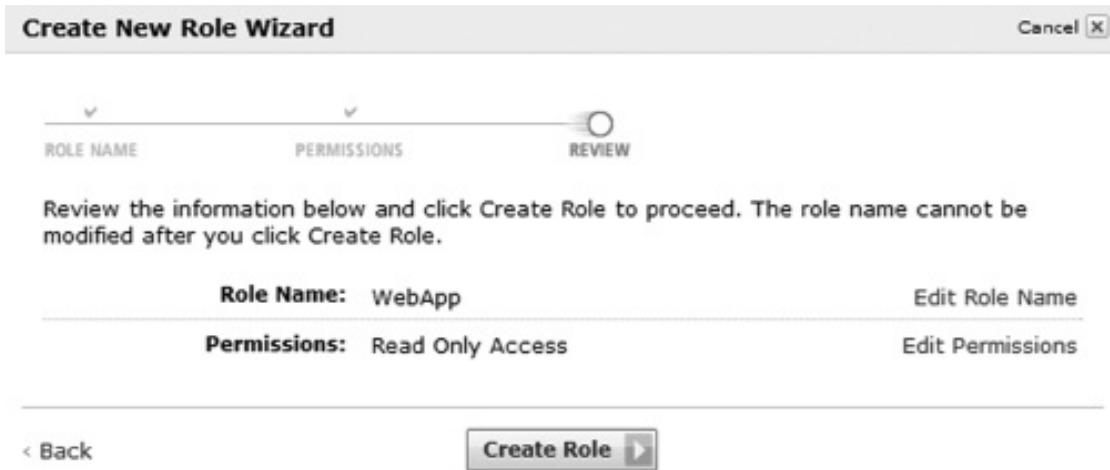


Figura 3-22 Create new role – review

### 3.5.2.4. Permissão (permission)

Uma permissão é o conceito de permitir (ou impedir) um usuário ou grupo de algum tipo de acesso a um ou mais recursos AWS. Por exemplo, João tem permissão para ler e gravar objetos para um determinado *bucket S3* nomeado “example\_bucket”. A permissão é um termo geral utilizado para definir a capacidade de executar uma ação contra um recurso.

Há dois tipos gerais de permissões que podem ser utilizadas dentro da AWS: baseada no usuário e baseada em recursos.

- **Permissão baseada no usuário:** a que um determinado usuário (ou grupo de usuários) tem acesso?
- **Permissão baseada em recursos:** quem tem acesso a um recurso específico?

Com o IAM, trabalha-se inteiramente com permissões de usuário; por outro lado, com o S3 e outros produtos, trabalha-se apenas com permissões baseadas em recursos.

A Figura 3-23 ilustra a diferença entre os dois tipos de permissões. Cada permissão de usuário enfoca um determinado usuário ou grupo; cada permissão baseada em recurso centra-se em um recurso específico.

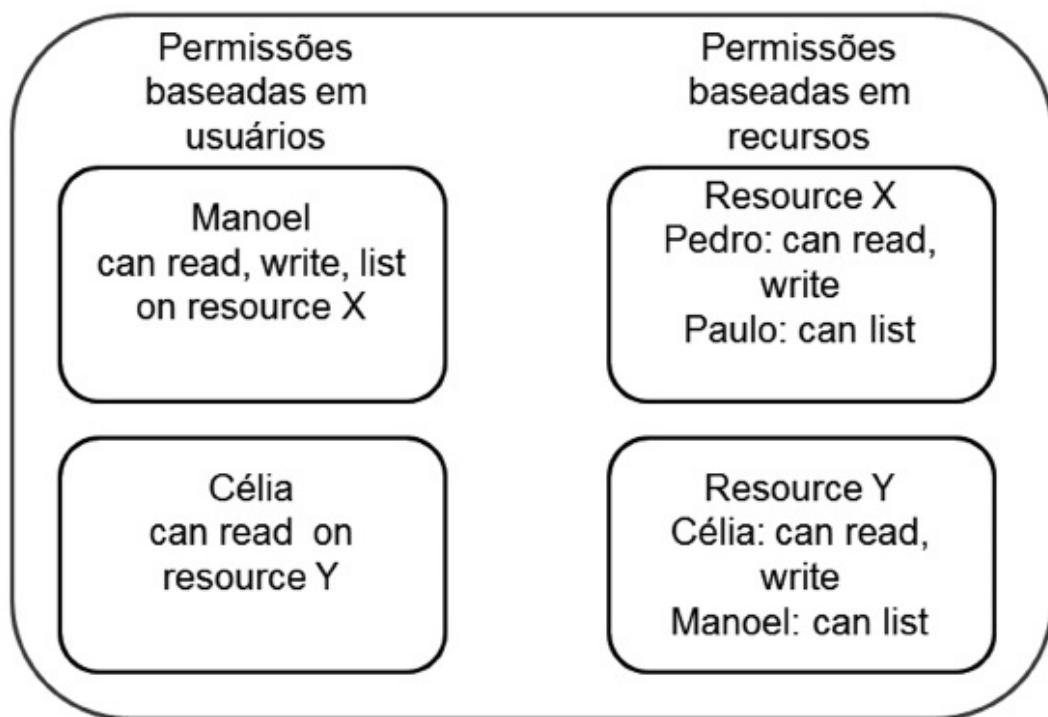


Figura 3-23 Permissões baseadas em usuários e em recursos

É possível ter permissões de recursos e permissões baseadas em usuário que se aplicam ao mesmo recurso ou usuário, mesmo que as permissões estejam localizadas em lugares separados (com o recurso no AWS service *versus* com o usuário no sistema IAM). Os dois tipos de permissões são considerados quando a AWS avalia conceder o acesso de um usuário específico para um determinado recurso.

Como padrão, durante a criação, os usuários do IAM não podem desempenhar nenhuma ação. O administrador da conta tem de conceder explicitamente permissões para o usuário obter acesso a qualquer recurso da AWS. O IAM fornece a capacidade de gerenciar centralmente permissões que controlam quais APIs podem ser acessadas pelo usuário. As permissões são expressas usando o *AWS Access Policy Language*, uma linguagem de controle de acesso flexível. Essas permissões são impostas nas solicitações feitas para os web services da AWS.

O IAM fornece a capacidade de gerenciar permissões para um conjunto de usuários (um grupo), além de um usuário individual. Qualquer política de segurança que possa conceder permissões a um único usuário também pode conceder permissões para um grupo. As permissões de grupos são aplicadas ao verificar a associação – se um grupo tiver recebido permissão para desempenhar uma ação e um usuário estiver naquele grupo, então o usuário

poderá desempenhar aquela ação. É mais fácil gerenciar permissões para um grupo do que aplicar permissões idênticas para cada usuário.

### 3.5.2.5. Política (policy)

Uma política é um documento que declara formalmente uma ou mais permissões. Com o IAM, é possível atribuir uma política a um grupo ou usuário e, em seguida, a entidade atribuída à política recebe as permissões estabelecidas. Pode-se atribuir várias políticas para um usuário ou grupo. Se for desejável atribuir a mesma política a vários usuários, é recomendável colocar os usuários em um grupo e atribuir a política ao grupo.

A distinção entre uma permissão e uma política é importante. Para dar uma permissão a uma entidade particular do IAM, você simplesmente escreve uma política consoante com a linguagem de política de acesso do IAM. Em seguida, deve-se anexar a política à entidade que você quer aplicar (um usuário ou grupo específico na sua conta AWS). Você não deve especificar a entidade na política, pois o ato de anexar a política para a entidade concede a permissão estabelecida na política.

É possível anexar mais de uma política a uma determinada entidade. Se houver várias permissões e for desejável aplicar a uma entidade, pode-se colocá-las em políticas separadas ou colocá-las todas em uma política.

A **Figura 3-24** ilustra uma possível arquitetura para o IAM de uma pequena empresa com três grupos. Neste exemplo, os grupos são admins, developers e teste. Cada grupo tem vários usuários. Cada usuário pode estar em mais de um grupo. Não é permitido colocar grupos dentro de outros grupos.

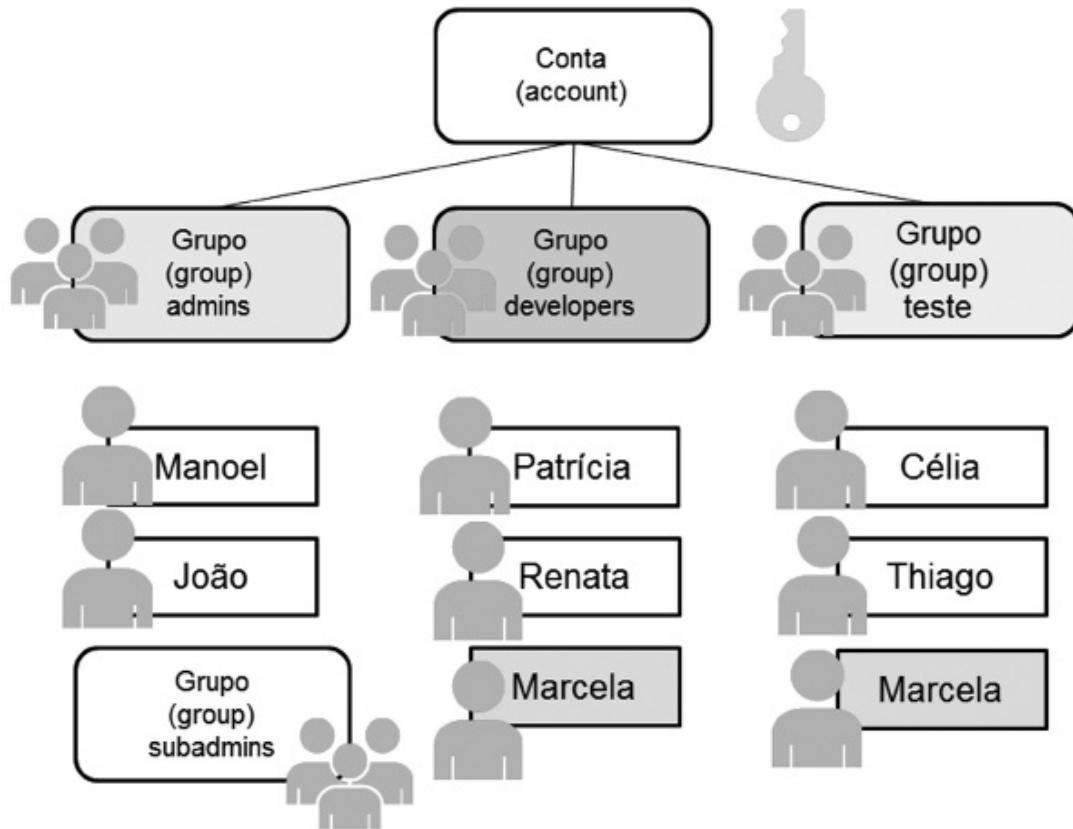


Figura 3-24 Uso do IAM em uma pequena empresa

### 3.5.3. Linguagem de política de acesso (Access Policy Language – APL)

#### 3.5.3.1. ARN

Embora a maioria dos recursos utilize um nome amigável (por exemplo, um usuário chamado *manoel*, um grupo chamado *desenvolvedores*), a linguagem da política de acesso requer especificar o recurso usando o seguinte formato de nome de recurso da Amazon (Amazon Resource Name – ARN):

`arn:aws:<vendor>:<região>:<namespace>:<relative-id>`

onde:

- vendor identifica o produto AWS (por exemplo, “iam”).
- região é o local onde o recurso reside (por exemplo, “us-east-1”).
- namespace é o ID da conta AWS (por exemplo, “123456789012”).
- relative-id é a porção que identifica o recurso específico.

#### 3.5.3.2. Conceitos

- **Instrução (*statement*)**: é a descrição formal de uma permissão

individual, escrita na linguagem de política de acesso.

- **Principal (*principal*)**: é a pessoa (ou pessoas) que recebe a permissão na política.
- **Ação (*action*)**: a atividade que o principal tem permissão para executar.
- **Recurso (*resource*)**: é o objeto a que o principal está solicitando acesso.
- **Condições e chaves (*conditions e keys*)**: são quaisquer restrições ou detalhes sobre a permissão.
- **Solicitante (*requester*)**: é a pessoa que envia uma solicitação para um serviço AWS e pede para acessar um determinado recurso.
- **Avaliação (*evaluation*)**: é o processo que o web service utiliza para determinar se uma solicitação de entrada deve ser negada ou permitida com base nas políticas aplicáveis.
- **Efeito (*effect*)**: é o resultado de uma declaração política que retorna no momento da avaliação.
- **Negação por padrão (*default deny*)**: é o resultado padrão de uma política na ausência de uma permissão ou de uma negação explícita.
- **Negação explícita (*explicit deny*)**: resulta de uma instrução que tem efeito de negar, assumindo as condições encontradas.
- **Permissão (*allow*)**: resulta de uma instrução que tem efeito de permitir, assumindo as condições encontradas.

### 3.5.3.3. Visão geral da arquitetura API

A **Figura 3-25** descreve os principais componentes que interagem para fornecer controle de acesso sobre os recursos.

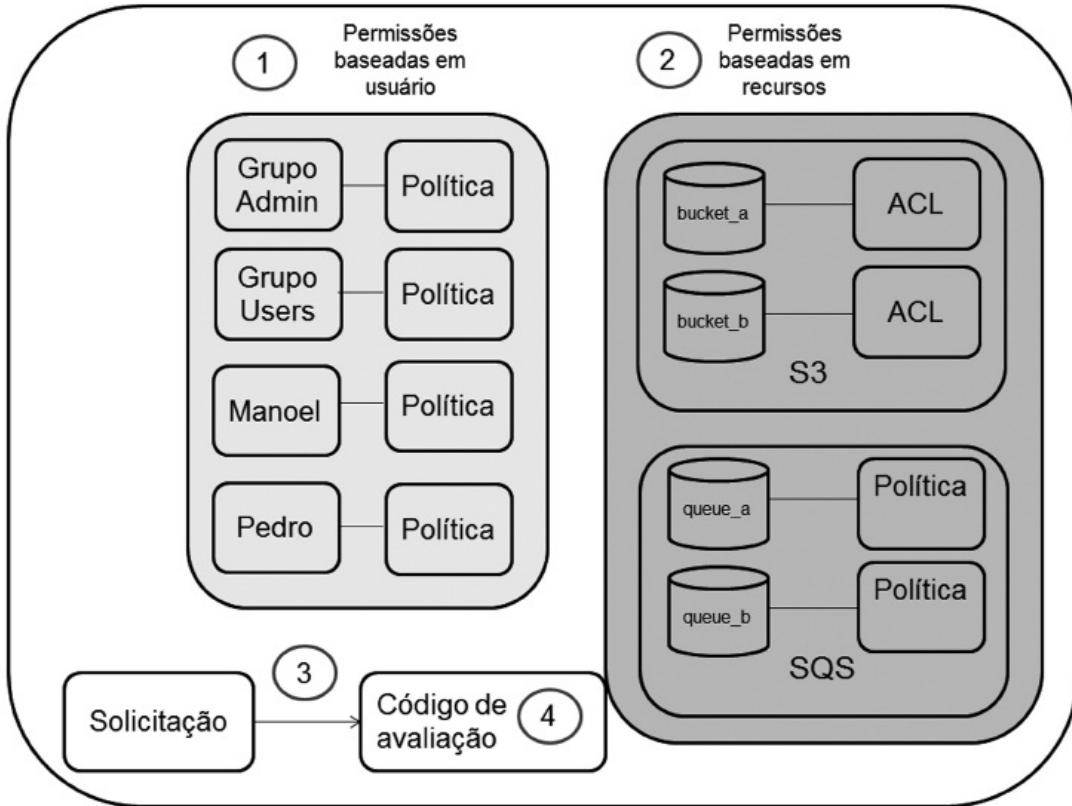


Figura 3-25 Visão geral da arquitetura da APL

1. Permissões baseadas em usuário (condições anexadas a usuários ou grupos dentro do sistema IAM) são criadas por alguém na conta AWS que tenha permissão para gerenciar as políticas da conta AWS. No diagrama ilustrado na figura, cada usuário ou grupo tem uma única política anexada a ele, embora na prática possam haver várias.
2. Recursos baseados em permissões (ACLs anexadas a buckets Amazon S3 e objetos, condições anexadas a filas de Amazon SQS, etc.) são criados por alguém na conta AWS que tem permissão para gerenciar ACLs, políticas, etc., para os recursos na conta AWS.
3. Solicitantes (*requesters*) fazem solicitações de entrada para utilizar os recursos da conta AWS. Eles podem ser de dentro ou fora da conta AWS.
4. O código de avaliação (*evaluation code*) é o conjunto de códigos que avalia as solicitações de entrada contra as permissões aplicáveis com base em usuário e permissões baseadas em recursos e determina se o solicitante tem permissão para acessar o recurso.

#### 3.5.3.4. Lógica de avaliação

O objetivo no momento da avaliação é decidir se uma determinada solicitação deve ser permitida ou negada. A lógica de avaliação segue várias regras básicas:

- Por padrão, todas as solicitações recebem uma negativa, exceto para

as solicitações que usam credenciais de segurança raiz.

- Uma permissão substitui qualquer negação padrão.
- Uma negação explícita substitui qualquer permissão.
- A ordem em que as políticas são avaliadas não é importante.
- A lógica de avaliação nunca resulta em um conflito. Há sempre um resultado de verdadeiro/falso que permite ou nega o acesso solicitado.

### 3.5.4. Credencial temporária

As credenciais de segurança temporárias consistem em ID de chave de acesso, chave de acesso secreta e token. Elas são válidas durante um período específico e para um conjunto específico de permissões. Às vezes, as credenciais de segurança temporárias são chamadas de “tokens”. Os tokens podem ser solicitados para usuários do IAM ou para usuários federados gerenciados no próprio diretório corporativo.

Credenciais de segurança temporárias podem ser solicitadas para usuários federados utilizando um usuário do IAM ou a conta-raiz da AWS para chamar a API *GetFederationToken* do AWS Security Token Service, transmitindo:

- **Nome:** uma identificação de texto representando o usuário federado.
- **Política:** uma política de acesso da AWS especificando as permissões concedidas ao detentor da credencial de segurança temporária. É importante reforçar que as permissões do token não podem ultrapassar as permissões do usuário do IAM que solicita o token.
- **Duração:** o período no qual as credenciais de segurança temporárias são válidas. O padrão é doze horas, o mínimo é uma hora e o máximo é 36 horas. As credenciais de segurança temporárias criadas pela conta-raiz são válidas por uma hora.

Haverá o retorno do seguinte:

- **ID da chave de acesso:** o identificador da chave de acesso com relação à credencial de segurança temporária.
- **Chave de acesso secreta:** a chave usada para efetuar solicitações associadas à credencial de segurança temporária.
- **Token:** o token de segurança.

As credenciais de segurança temporárias permitem que:

- **Usuários federados acessem APIs de web services AWS:** as credenciais de segurança temporárias permitem que você estenda os diretórios de usuários internos para a AWS, possibilitando que os funcionários e aplicativos acessem com mais segurança APIs da AWS, sem a necessidade de criar uma identidade AWS para eles.
- **Escalonem de forma ilimitada:** é possível solicitar credenciais de segurança temporárias para um número ilimitado de usuários federados.
- **Segurança aprimorada:** é possível configurar o período depois do qual as credenciais de segurança temporárias expiram, oferecendo segurança aprimorada ao acessar APIs da AWS por meio de dispositivos móveis onde há risco de perda do dispositivo. Pode-se também integrar a AWS Multi-Factor Authentication (MFA) nos aplicativos para proteger melhor recursos e APIs importantes da AWS. Os usuários solicitando acesso devem primeiro ter realizado a autenticação com o dispositivo de MFA atribuído.

Os tokens do AWS Security Token Service podem ser utilizados na região leste dos EUA. Uma credencial de segurança temporária não pode ser revogada antes que expire.

A Amazon recomenda as seguintes práticas no uso de credenciais de segurança temporárias:

- Uma vez solicitadas, as credenciais de segurança temporárias podem ser usadas durante o período estabelecido para chamar APIs da AWS – não é necessário solicitar um novo token para cada chamada de API.
- Deve-se selecionar a duração em que a credencial de segurança temporária é válida que melhor se adeque ao seu caso de uso.
- Deve-se solicitar uma nova credencial de segurança temporária antes que a antiga expire, para que as chamadas dos serviços da AWS não sejam interrompidas devido a um token expirado.

A Amazon reforça que deve-se seguir o princípio de segurança de menor privilégio – conceda a credencial de segurança temporária somente para o nível de permissões exigido para a tarefa que precise ser desempenhada.

Embora a Amazon permita criar credenciais de segurança temporárias por meio de uma conta-raiz da AWS visando à consistência, ela recomenda enfaticamente a utilização de um usuário do IAM para a realização desta tarefa.

### **3.5.5. Federação de identidades com AWS**

- A federação de identidades é estabelecida quando um sistema confia nos usuários de outro sistema. O IAM permite solicitar credenciais de

segurança temporárias para identidades corporativas, permitindo a federação de identidade do diretório corporativo da empresa para o gerenciador de console e APIs de serviço da AWS sem ter que criar usuários IAM para todas as suas identidades corporativas.

- Usuários federados são usuários gerenciados fora da AWS no diretório corporativo. Eles diferem dos usuários do IAM, que são criados e mantidos no diretório de identidade da AWS. É possível solicitar credenciais de segurança temporárias para os usuários federados para lhes fornecer acesso seguro e direto às APIs da AWS.
- A federação de identidades usa credenciais de segurança temporárias. Após autenticar um usuário e conceder-lhe credencial de segurança temporária, é possível gerar um token de *sign-in* que pode ser usado para acesso à AWS. Isso dá acesso ao console. As ações do usuário no console são limitadas pela política de controle de acesso incorporada à credencial temporária.

É possível utilizar um “Federation Proxy Server” que utiliza funções IAM que criam credenciais temporárias de acesso e que podem ser utilizadas por usuários registrados no AD do Windows para fazer SSO (*Single-Sign-On*) no console de gerenciamento AWS. Esta é uma aplicação desenvolvida em Cº e disponibilizada no site da AWS.

## 3.6. Referências bibliográficas

AWS Identity and Access Management. **Using IAM: API Version 2010-05-08.** Amazon Web Services, 2011.

Chong, Frederick. **Gerenciamento de Identidade e Acesso.** Microsoft, 2004.

<http://www.qualisoft.com.br>.

# 4. Precificação e Faturamento

## 4.1. Introdução

Os fundamentos deste capítulo estão baseados no artigo “How AWS Pricing Works”, publicado pela AWS em dezembro de 2011.

Preço é um aspecto-chave a ser considerado em uma avaliação para adoção do modelo cloud computing. Preços dependem dos custos associados à entrega do serviço que está sendo cobrado. Um modelo eficiente em termos de custos pode propiciar preços de serviços mais atrativos.

Este capítulo estabelece os princípios utilizados pela Amazon para precificar os serviços da AWS. Os preços mostrados nas tabelas a seguir são para a região de São Paulo, e a referência é em dólar e são para os principais serviços da AWS. A precificação do AWS Glacier também é mostrada, mesmo que o serviço ainda não esteja disponível para São Paulo.

Maiores informações sobre precificação ou mesmo a atualização dos preços das tabelas que serão expostas aqui podem ser encontradas em <http://aws.amazon.com/pricing>.

Além dos aspectos de preço, será abordada neste capítulo a opção de faturamento da AWS.

## 4.2. Precificação

### 4.2.1. Custos tradicionais *versus* custos na AWS

Em TI, boa parte das empresas não consegue estimar os custos dos serviços. Como então saber se os preços de utilização de uma solução de nuvem são menores do que uma solução interna, se muitas vezes não existe nem um padrão de comparação?

Estas dificuldades fazem com que muitos executivos de TI, especialmente em empresas grandes e bem estabelecidas, questionem se serviços de nuvem são realmente mais baratos que o custo de provisionar capacidade nos seus próprios DATACENTERS. Boa parte deles não tem como comparar esses custos, conforme mencionado. Ou seja, uma análise só baseada nos custos dos serviços de TI produzidos dentro de casa contra custos de serviços de nuvem será quase sempre pouco conclusiva. Outros fatores também precisam ser considerados quando se pensa em adquirir serviços de TI, incluindo flexibilidade, agilidade e elasticidade para a avaliação de uso do modelo cloud computing.

Custos de utilização dos serviços de nuvem são essencialmente

diferentes de um modelo tradicional de TI. No modelo tradicional os maiores custos são do tipo CapEx, ou seja, custos de capital. No modelo baseado em nuvem, como a AWS, os custos são do tipo OpEx, ou seja, custos operacionais. Esta é uma mudança importante que deve ser bem explicada aos gerentes e diretores financeiros pelo pessoal de TI.

Os DATACENTERS corporativos têm orçamento anual que incluem uma mistura de despesas operacionais e de capital. Serviços de nuvem são, geralmente, baseados em cobrança por uso ou por hora e são basicamente custos operacionais. Mesmo que a área de TI possa de forma aproximada definir gastos com hardware e software para rodar uma determinada aplicação e o seu consequente serviço de TI, em casa ainda precisariam definir outros custos para compor o custo total. Custos do espaço do DATACENTER alocado, custos com o consumo de energia elétrica, custos com a manutenção, por exemplo, devem ser computados. Tarefa complicada. Além disso, quanto tempo e custo de gerenciamento de rede, gerenciamento de banco de dados e armazenamento e mesmo o custo e o tempo de administrar os sistemas devem ser alocados para um serviço de TI que roda internamente? Geralmente, a TI tem uma visão limitada desses fatores quando se pensa em aplicação por aplicação ou mesmo serviço por serviço; portanto, será sempre uma estimativa aproximada. Entendeu?

Diferentemente do modelo tradicional, a AWS oferece uma grande gama de serviços de cloud computing. Cada um destes serviços é cobrado pela quantidade de recursos utilizados durante o tempo de uso. Este modelo de precificação é baseado nos seguintes aspectos:

- **Pay as you go.** Não há compromissos mínimos necessários, nem contratos de longo prazo. Essa flexibilidade reduz a necessidade de realizar um planejamento de uso dos recursos de forma detalhada.
- **Pague pelo uso.** Não há necessidade de pagar adiantado para o excesso de capacidade ou ser penalizado por problemas de falta de planejamento.
- **Pague menos usando mais.** Para transferência de armazenamento de dados, o preço é definido em camadas. Quanto mais usa, menos paga. Pague ainda menos quando for feita uma reserva. Assim, para determinados produtos, pode-se investir em capacidade reservada.
- **Preço personalizado.** Preços personalizados estão disponíveis para projetos de alto volume com necessidades específicas.

#### **4.2.2. Características fundamentais**

Entender a precificação é um aspecto importante da AWS. Há três características fundamentais que determinam os preços de uso da AWS: a computação, o armazenamento e a transferência de dados para fora da nuvem. Os preços variam dependendo dos serviços que estão sendo

utilizados e da forma que são adquiridos.

É importante ressaltar que, embora haja cobrança para a transferência de dados para fora da nuvem, não há custo para transferência de dados para dentro da nuvem ou para transferência de dados entre web services AWS dentro da mesma região. As regiões foram ilustradas e explicadas no capítulo 2.

A AWS possui uma lógica interessante para especificar os serviços. A especificação envolve o que cobrar, quando cobrar e como cobrar. As dimensões utilizadas para cobrar o serviço são [BARR, 2011]:

- **Tempo:** uma hora de tempo de CPU.
- **Volume:** gigabyte de dado transferido.
- **Count:** número de mensagens enfileiradas.
- **Tempo e espaço:** gigabyte-mês de dado armazenado.

A forma de adquirir as instâncias, a computação propriamente dita, baliza o custo e o preço a ser pago.

Um aspecto importante é que os serviços podem possuir mais de uma dimensão para a especificação e podem ter preços diferentes quando se considera a região, ou seja, quando se pensa em preço deve-se considerar a região onde estão armazenados os dados e onde efetivamente roda a aplicação. A AWS mede o uso de cada serviço considerando quando o serviço foi chamado, que serviços foram utilizados e quantos recursos foram consumidos.

Além disso, é possível consolidar todas as contas de usuário de certa empresa ou divisão usando o faturamento consolidado e obter os benefícios de hierarquização das contas, incluindo a redução dos preços dos serviços baseada no ganho de escala.

Ressalta-se que a AWS também oferece diversos serviços sem custo adicional.

#### **4.2.3. Infraestrutura interna *versus* infraestrutura na AWS**

Uma parte do problema de justificar a adoção da nuvem é que uma estrutura interna de TI (*on-premises, self-host*) e de serviços de nuvem como a AWS é difícil de comparar. Isto já foi mencionado.

Ganhos de escala da nuvem repassados aos clientes devem de uma forma geral reduzir os custos quando comparados a uma operação interna de TI. Uma empresa típica, por exemplo, usa a virtualização para reduzir a taxa de administrador por servidor para 1 para 30 ou 1 para 50. Na nuvem pode-se pensar em 1 para 3000 ou 1 para 5000. Esta comparação dá uma ideia do

ganho de escala conseguido.

O artigo “Which is less expensive: Amazon or self-hosted?”, publicado no blog GIGAOM e escrito por Charlie Oppenheimer, traz algumas conclusões importantes sobre a importância do perfil da carga de trabalho na definição do modelo mais eficiente (*self-hosting versus hosting on AWS*) em termos de custo.

A conclusão do artigo de Oppenheimer é que a carga de trabalho e o seu comportamento no tempo são decisivos para a adoção do modelo. Cargas constantes no tempo são mais previsíveis e, se bem utilizadas, podem não justificar a adoção da nuvem quando se verifica só o aspecto de preços. Cargas variáveis com pouca previsibilidade são mais baratas quando rodam na nuvem. A nuvem permite provisionar recursos mediante a demanda e depois liberá-los.

James Urquhart conclui, em outro artigo publicado pelo blog GIGAOM intitulado “Why Amazon and Salesforce are pulling away from the cloud pack”, que, para a maior parte das situações, as vantagens econômicas para IaaS realmente ainda pertencem a aplicações que possuem naturezas dinâmicas — que fixa prazos de execução (por exemplo, processamento de dados) ou de carga variável (por exemplo, web apps).

Ressalta-se que manter a infraestrutura própria recai em dois tipos básicos de custos:

- Custos diretos, que incluem hardware, energia, infraestrutura, redundância, segurança, excesso de capacidade e pessoal.
- Custos indiretos, que são traduzidos em custos de oportunidade. Custo de possuir, manter e explorar uma infraestrutura tradicional de TI. Este tipo de custo normalmente não é considerado.

Boa parte desses custos some no modelo de nuvem.

#### **4.2.4. Precificação EC2**

O preço do EC2 considera vários aspectos importantes.

##### **4.2.4.1. Aspectos importantes**

- **Computação:**
  - **Horas de relógio de tempo do servidor.** Recursos incorrem em custos quando eles estão sendo executados. Por exemplo, a partir do momento em que instâncias EC2 são lançadas até a sua conclusão, ou a partir do momento em que IPs elásticos são distribuídos até o momento em que são desalocados.
  - **Configuração da máquina.** Deve-se considerar a capacidade física da instância EC2 escolhida. As características variam de acordo

com o SO, o número de núcleos, a memória e o armazenamento local.

- **Opção de compra.** Existem vários modelos de opção de compras. As opções de compra serão explicadas no próximo tópico.
- **Número de instâncias.** Podem ser configuradas várias instâncias do EC2 para lidar com grandes cargas de trabalho.
- **Armazenamento:**
  - **Armazenamento adicional.** O Amazon Elastic Block Store (EBS) fornece volumes de blocos de armazenamento para uso com instâncias EC2. Volumes EBS persistem independentemente da vida de uma instância.
  - **Backups.** O EBS oferece a capacidade de fazer backup dos *snapshots* dos dados no storage S3 visando à recuperação de desastres.
- **Transferência de dados:** deve-se levar em conta a quantidade de dados transferidos para fora da aplicação. A transferência de dados de entrada é gratuita e as taxas de transferência de saída de dados são escalonadas.
- **Balanceamento de carga:** um ELB pode ser usado para distribuir o tráfego entre as instâncias EC2. O número de horas que o ELB é executado e a quantidade de dados que processa contribuem para o custo mensal.
- **Monitoramento detalhado:** pode-se utilizar o CloudWatch para monitorar instâncias EC2. Por padrão, o monitoramento básico está habilitado (e disponível, sem custo adicional), porém, por uma taxa fixa mensal, pode-se optar por acompanhamento detalhado do EC2.
- **Auto Scaling:** ajusta automaticamente o número de instâncias do EC2 em sua implantação, de acordo com as condições definidas. Este serviço está disponível sem custo adicional, além da taxa cobrada pelo CloudWatch.
- **Endereços IP elásticos (EIPs):** EIPs são endereços IP projetados para cloud computing. Este é um serviço gratuito, desde que os endereços IP elásticos sejam utilizados. Cobram-se os endereços IP elásticos apenas se estes não forem usados.
- **Sistemas operacionais e pacotes de software:** o preço do sistema operacional está incluído no preço da instância EC2. Não há custos adicionais de licenciamento para executar os seguintes sistemas operacionais comerciais: Red Hat Enterprise Linux, SUSE Linux Enterprise, Windows Server e Oracle Enterprise Linux. Além disso, a AWS fez uma parceria com a Microsoft, IBM e vários outros

fornecedores para que pacotes de softwares comerciais possam ser executados em instâncias EC2. Neste caso é necessário obter uma licença dos fornecedores. Também se pode trazer uma licença já adquirida para a nuvem por meio de programas específicos de fornecedores como a Microsoft, através do Programa Mobilidade de Licença do *Software Assurance*, como explicado no capítulo 1.

#### **4.2.4.2. Opção de compra EC2**

A escolha da opção de compra das instâncias EC2 determina o preço de referência a ser pago pelo uso da AWS. A necessidade do aplicativo, ou seja, da condição da carga de trabalho e o seu comportamento no tempo devem definir o modelo adequado de aquisição dessas instâncias.

Além da flexibilidade de escolher o número, o tamanho e a configuração de instâncias computacionais necessárias para o aplicativo, o EC2 fornece aos clientes quatro modelos de compra diferentes que proporcionam flexibilidade e otimização dos preços.

As instâncias *on demand* permitem que se pague uma taxa fixa por hora de uso sem nenhum compromisso a longo prazo; com as instâncias reservadas, paga-se uma taxa única baixa e, em troca, recebe-se um desconto significativo na cobrança do uso por hora daquela instância; e as instâncias *spot* permitem que se proponha o preço que se deseja pagar pela capacidade da instância, proporcionando uma economia ainda maior se os aplicativos tiverem períodos de início e de término flexíveis. Existem também as instâncias dedicadas que podem ser utilizadas para instâncias EC2 que precisam estar isoladas por questões de conformidade. Criar opções diferentes para aquisição de instâncias foi a forma que a AWS encontrou de entregar a sua capacidade finita para milhares de clientes.

O cliente da AWS não precisa fazer o planejamento da capacidade da infraestrutura do DATACENTER da forma usual, mas deverá encontrar a melhor forma de comprar esta capacidade do provedor AWS. Se a forma de adquirir as instâncias for otimizada poderá se pagar um preço ótimo.

O segredo para obter um bom preço está na previsibilidade do comportamento da carga de trabalho do aplicativo, que permite comprar a quantidade ideal de instâncias reservadas. A **Figura 4-1** ilustra a mudança de paradigma.

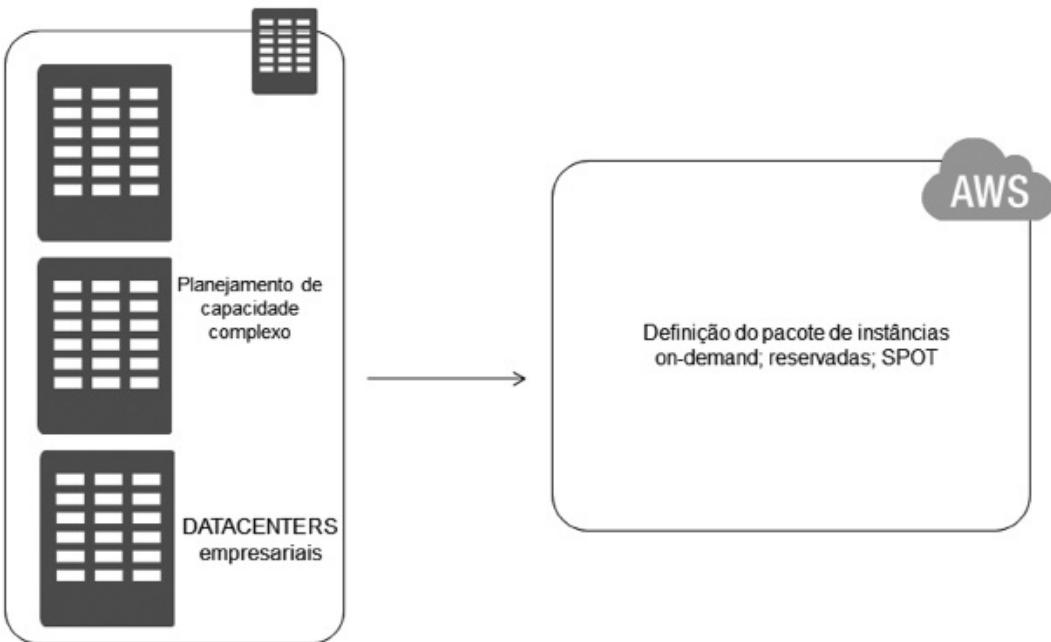


Figura 4-1 Mudança no planejamento da capacidade com a AWS

#### 4.2.4.3. Instâncias *on demand*

As instâncias *on demand* permitem o pagamento para utilização da capacidade computacional por hora, sem compromissos a longo prazo ou pagamentos prévios. É a essência do modelo AWS para cloud computing.

As instâncias *on demand* permitem aumentar ou diminuir a capacidade computacional dependendo das demandas do aplicativo e pagar somente pelas instâncias usadas de acordo com a taxa especificada pela AWS. A Amazon reforça que o EC2 sempre busca ter capacidade *on demand* disponível para atender às necessidades dos clientes, mas durante períodos de demanda muito alta talvez não seja possível iniciar tipos de instância específicos em certas zonas de disponibilidade. Este é um aspecto importante.

As instâncias *on demand* são recomendadas para:

- Aplicativos cujos usuários desejam o custo baixo e a flexibilidade do EC2 sem nenhum pagamento prévio ou compromisso de longo prazo.
- Aplicativos utilizados por pouco tempo mas importantes ou com cargas de trabalho imprevisíveis que não podem ser interrompidas.
- Aplicativos sendo desenvolvidos ou testados no EC2 pela primeira vez.

A **Tabela 4-1** ilustra os preços para instâncias *on demand* na região América do Sul (São Paulo). Os preços em dólar incluem o custo para executar AMIs públicas e privadas baseadas no sistema operacional escolhido.

**Tabela 4-1 EC2 – Preço das instâncias *on demand***

### **Instâncias on demand – padrão**

	<b>Linux/Unix</b>	<b>Windows</b>
Pequeno	\$0,115 por hora	\$0,150 por hora
Médio	\$0,230 por hora	\$0,300 por hora
Grande	\$0,460 por hora	\$0,600 por hora
Extragrande	\$0,920 por hora	\$1,200 por hora

### **Instâncias on demand – micro**

Micro	\$0,027 por hora	\$0,037 por hora
-------	------------------	------------------

### **Instâncias on demand – mais memória**

Extragrande	\$0,680 por hora	\$0,800 por hora
Dupla Extragrande	\$1,360 por hora	\$1,600 por hora
Quádrupla Extragrande	\$2,720 por hora	\$3,200 por hora

### **Instâncias on demand – mais CPU**

Médio	\$0,230 por hora	\$0,350 por hora
Extragrande	\$0,920 por hora	\$1,400 por hora

O capítulo 5 explica as diferenças entre instâncias padrão, micro, instância de mais memória e instância de mais CPU.

#### **4.2.4.4. Instâncias reservadas**

As instâncias reservadas do EC2 permitem que sejam preservados os benefícios da computação elástica ao mesmo tempo em que diminuem os custos da reserva de capacidade. Com as instâncias reservadas, paga-se uma pequena taxa única e, por sua vez, recebe-se um desconto significativo sobre a cobrança de utilização por hora para tal instância. As instâncias reservadas

podem oferecer uma economia substancial em relação a possuir infraestrutura interna ou a execução única de instâncias *on demand*, assim como podem ajudar a garantir que a capacidade necessária esteja disponível.

Funcionalmente, as instâncias reservadas e as instâncias *on demand* são exatamente as mesmas. Elas são iniciadas e encerradas da mesma forma e funcionam identicamente assim que estiverem em execução. Isso facilita o uso de instâncias reservadas e *on demand* em conjunto sem que o desenvolvedor precise fazer qualquer alteração na aplicação.

As instâncias reservadas são recomendadas para:

- Aplicativos com condição estável ou uso previsível.
- Aplicativos que exigem capacidade reservada, incluindo aplicativos desenvolvidos para recuperação de desastres.
- Aplicativos cujos usuários podem fazer pagamentos prévios para reduzir os custos computacionais.

As instâncias reservadas são fáceis de usar e a forma de uso do EC2 permanece as mesmas. Ao fazer o cálculo, o sistema de preços da AWS automaticamente irá aplicar primeiro as taxas de instâncias reservadas para minimizar os preços finais. A hora de instância será cobrada somente na taxa *on demand* quando a quantidade total de instâncias em execução naquela hora ultrapassar o número adquirido de instâncias reservadas.

Sobre as instâncias reservadas:

- As instâncias reservadas podem ser adquiridas por períodos de um ou três anos.
- As instâncias reservadas estão disponíveis em todas as regiões AWS.
- As instâncias reservadas também estão disponíveis para serem utilizadas com a opção de rede privada virtual VPC e para instâncias dedicadas.
- As instâncias reservadas estão atualmente disponíveis para o EC2 em execução nas plataformas Linux/UNIX, SUSE Linux, RedHat Enterprise Linux, Microsoft Windows Server e Microsoft SQL Server.

Caso uma instância reservada esteja sendo comprada e se deseja que a garantia de capacidade se aplique a uma VPC, deve-se selecionar a opção “VPC” em “Launch Instances”, na opção de caixa de diálogo fornecida pelo console de gerenciamento AWS. A **Figura 4-2** ilustra esta opção.

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

**Number of Instances:**  **Instance Type:** Micro (t1.micro, 613 MB) **Launch into:**  EC2  VPC

**Subnet:** subnet-a77485ce (10.0.0.0/24) sa-east-1a 250 available IP addresses

**Request Spot Instances**

Figura 4-2 Instância reservada na VPC

### Tipos de instâncias reservadas (*Reserved Instances – RIs*)

Existem três tipos de oferta para as instâncias reservadas, descritas a seguir, que permitem equilibrar o valor inicial pago e o preço efetivo da hora.

- **RIs de utilização leve:** oferecem o menor pagamento prévio de todos os tipos de instância reservada. Junto com esse baixo pagamento prévio é fornecido um desconto significativo na taxa de uso por hora. Estas RIs permitem desconectar a instância a qualquer momento e não pagar a taxa horária e são ideais para cargas de trabalho periódicas que são executadas apenas por algumas horas por dia, poucos dias por semana, ou muito esporadicamente, como em caso de recuperação de desastres.
- **RIs de utilização média:** apresentam um pagamento inicial maior do que as RIs de utilização leve, mas uma taxa de utilização por hora mais baixa. Estas RIs permitem desconectar a instância a qualquer momento e não pagar a taxa horária e são mais adequadas para cargas de trabalho executadas a maior parte do tempo, mas com alguma variabilidade em uso como boa parte dos aplicativos web.
- **RIs de utilização pesada:** oferecem a maior economia para qualquer tipo de instância reservada. Elas são mais apropriadas para cargas de trabalho de estado estacionário onde sempre se executam essas instâncias, em troca de uma taxa mais baixa de uso por hora. Com esta RI, faz-se um pagamento prévio mais alto do que as RIs de média utilização, mas obtém-se uma taxa significativamente menor de uso por hora.

Alguns valores de parâmetros precisam ser escolhidos antes de adquirir as instâncias reservadas, incluindo: plataforma, tipo de instância, tipo de oferta (pesada, média ou leve), zona de disponibilidade, tipo de locação (default ou dedicada) e se é para reservar para um ou três anos.

A **Figura 4-3** ilustra, como exemplo, o preço a ser pago por uma instância m1.small de utilização leve (*light utilization*) por três anos. Os parâmetros mostrados são a taxa efetiva (*effective rate*), o preço inicial (*upfront price*) e a

taxa horária (*hourly rate*). Observe que os valores estão de acordo com a **Tabela 4-2**.

The screenshot shows the 'Purchase Reserved Instances' interface. The search criteria are set to Platform: Linux/UNIX, Term: 2 years - 3 years, Instance Type: m1.small, Availability Zone: sa-east-1a, and Offering Type: Light Utilization. A single result is shown for AWS, 36 months, \$0.063 upfront price, \$0.055 hourly rate, in sa-east-1a, offering Light Utilization, with an unlimited quantity available and a desired quantity of 1. An 'Add to Cart' button is visible.

Figura 4-3 Compra de instâncias reservadas

A **Tabela 4-2** ilustra o preço para instâncias reservadas Linux de utilização leve para prazos de um e três anos.

**Tabela 4-2 EC2 – Preço de instâncias reservadas – utilização leve – Linux**

Prazo de um ano		Prazo de três anos		
	inicial	por hora	inicial	por hora
<b>Instâncias reservadas – padrão</b>				
Pequeno (padrão)	\$131,63	\$0,07 por hora	\$203	\$0,055 por hora
Médio	\$263,50	\$0,135 por hora	\$405	\$0,11 por hora
Grande	\$527	\$0,27 por hora	\$810	\$0,22 por hora
Extragrande	\$1053	\$0,54 por hora	\$1620	\$0,44 por hora
<b>Instâncias reservadas – micro</b>				
Micro	\$31	\$0,016 por hora	\$47	\$0,016 por hora
<b>Instâncias reservadas – mais memória</b>				
Extragrande	\$749	\$0,38 por hora	\$1151	\$0,308 por hora
Dupla extragrande	\$1485	\$0,77 por hora	\$2302	\$0,616 por hora
Quádrupla extragrande	\$2970	\$1,54 por hora	\$4604	\$1,232 por hora

Instâncias reservadas – mais CPU				
	\$263	\$0,135 por hora	\$405	\$0,118 por hora
Extragrande	\$1053	\$0,54 por hora	\$1620	\$0,472 por hora

A **Tabela 4-3** ilustra o preço para instâncias reservadas Windows de utilização leve para prazos de um e três anos.

**Tabela 4-3 EC2 – Preço de instâncias reservadas – utilização leve – Windows**

	Prazo de um ano		Prazo de três anos	
	inicial	por hora	inicial	por hora

Instâncias reservadas – padrão				
Pequeno (padrão)	\$131,63	\$0,09 por hora	\$203	\$0,075 por hora
Médio	\$263,50	\$0,175 por hora	\$405	\$0,15 por hora
Grande	\$527	\$0,35 por hora	\$810	\$0,30 por hora
Extragrande	\$1053	\$0,70 por hora	\$1620	\$0,60 por hora

Instâncias reservadas – micro				
Micro	\$31	\$0,022 por hora	\$47	\$0,022 por hora

Instâncias reservadas – mais memória				
Extragrande	\$749	\$0,45 por hora	\$1151	\$0,378 por hora
Dupla extragrande	\$1485	\$0,91 por hora	\$2302	\$0,756 por hora
Quádrupla extragrande	\$2970	\$1,82 por hora	\$4604	\$1,512 por hora

Instâncias reservadas – mais CPU				
Médio	\$263	\$0,20 por hora	\$405	\$0,183 por hora
Extragrande	\$1053	\$0,80 por hora	\$1620	\$0,732 por hora

A **Tabela 4-4** ilustra o preço para instâncias reservadas Linux de utilização média para prazos de um e três anos.

**Tabela 4-4 EC2 – Preço de instâncias reservadas – utilização média – Linux**

		Prazo de um ano	Prazo de três anos		
		inicial	por hora	inicial	por hora
<b>Instâncias reservadas – padrão</b>					
Pequeno (padrão)	\$307,13	\$0,04 por hora	\$473	\$0,031 por hora	
Médio	\$614,50	\$0,08 por hora	\$945	\$0,063 por hora	
Grande	\$1229	\$0,16 por hora	\$1890	\$0,124 por hora	
Extragrande	\$2457	\$0,32 por hora	\$3780	\$0,248 por hora	
<b>Instâncias reservadas – micro</b>					
Micro	\$73	\$0,009 por hora	\$111	\$0,009 por hora	
<b>Instâncias reservadas – mais memória</b>					
Extragrande	\$1789	\$0,23 por hora	\$2700	\$0,183 por hora	
Dupla extragrande	\$3578	\$0,46 por hora	\$5400	\$0,366 por hora	
Quádrupla extragrande	\$7155	\$0,92 por hora	\$10800	\$0,732 por hora	
<b>Instâncias reservadas – mais CPU</b>					
Médio	\$614	\$0,08 por hora	\$945	\$0,07 por hora	
Extragrande	\$2457	\$0,32 por hora	\$3780	\$0,28 por hora	

A **Tabela 4-5** ilustra o preço para instâncias reservadas Windows de utilização média para prazos de um e três anos.

**Tabela 4-5 EC2 – Preço de instâncias reservadas – utilização média – Windows**

		Prazo de um ano	Prazo de três anos		
		inicial	por hora	inicial	por hora

<b>Instâncias reservadas – padrão</b>				
Pequeno (padrão)	\$307,13	\$0,06 por hora	\$473	\$0,051 por hora
Médio	\$614,50	\$0,12 por hora	\$945	\$0,103 por hora
Grande	\$1229	\$0,24 por hora	\$1890	\$0,204 por hora
Extragrande	\$2457	\$0,48 por hora	\$3780	\$0,408 por hora

<b>Instâncias reservadas – micro</b>				
Micro	\$73	\$0,015 por hora	\$111	\$0,015 por hora

<b>Instâncias reservadas – mais memória</b>				
Extragrande	\$1789	\$0,30 por hora	\$2700	\$0,253 por hora
Dupla extragrande	\$3578	\$0,60 por hora	\$5400	\$0,506 por hora
Quádrupla extragrande	\$7155	\$1,20 por hora	\$10800	\$1,012 por hora

<b>Instâncias reservadas – mais CPU</b>				
Médio	\$614	\$0,15 por hora	\$945	\$0,135 por hora
Extragrande	\$2457	\$0,58 por hora	\$3780	\$0,54 por hora

A **Tabela 4-6** ilustra o preço para instâncias reservadas Linux de utilização pesada para prazos de um e três anos.

**Tabela 4-6 EC2 – Preço de instâncias reservadas – utilização pesada – Linux**

	<b>Prazo de um ano</b>		<b>Prazo de três anos</b>	
	<b>inicial</b>	<b>por hora</b>	<b>inicial</b>	<b>por hora</b>
<b>Instâncias reservadas – padrão</b>				
Pequeno (padrão)	\$372,94	\$0,03 por hora	\$574	\$0,021 por hora
Médio	\$746	\$0,055 por hora	\$1148	\$0,043 por hora
Grande	\$1492	\$0,11 por hora	\$2295	\$0,084 por hora

Extragrande	\$2984	\$0,22 por hora	\$4590	\$0,168 por hora
-------------	--------	-----------------	--------	------------------

### Instâncias reservadas – micro

Micro	\$84	\$0,007 por hora	\$135	\$0,007 por hora
-------	------	------------------	-------	------------------

### Instâncias reservadas – mais memória

Extragrande	\$2160	\$0,153 por hora	\$3260	\$0,123 por hora
Dupla extragrande	\$4320	\$0,305 por hora	\$6521	\$0,246 por hora
Quádrupla extragrande	\$8640	\$0,61 por hora	\$13041	\$0,492 por hora

### Instâncias reservadas – mais CPU

Médio	\$747	\$0,055 por hora	\$1148	\$0,048 por hora
Extragrande	\$2984	\$0,22 por hora	\$4590	\$0,192 por hora

A **Tabela 4-7** ilustra o preço para instâncias reservadas Windows de utilização pesada para prazos de um e três anos.

**Tabela 4-7 EC2 – Preço de instâncias reservadas – utilização pesada – Windows**

	Prazo de um ano		Prazo de três anos	
	inicial	por hora	inicial	por hora

### Instâncias reservadas – padrão

Pequeno (padrão)	\$372,94	\$0,05 por hora	\$574	\$0,041 por hora
Médio	\$746	\$0,095 por hora	\$1148	\$0,083 por hora
Grande	\$1492	\$0,19 por hora	\$2295	\$0,164 por hora
Extragrande	\$2984	\$0,38 por hora	\$4590	\$0,328 por hora

### Instâncias reservadas – micro

Micro	\$84	\$0,013 por hora	\$135	\$0,013 por hora
-------	------	------------------	-------	------------------

## Instâncias reservadas – mais memória

Extragrande	\$2160	\$0,223 por hora	\$3260	\$0,193 por hora
Dupla extragrande	\$4320	\$0,445 por hora	\$6521	\$0,386 por hora
Quádrupla extragrande	\$8640	\$0,89 por hora	\$13041	\$0,772 por hora

## Instâncias reservadas – mais CPU

Médio	\$747	\$0,12 por hora	\$1148	\$0,113 por hora
Extragrande	\$2984	\$0,48 por hora	\$4590	\$0,452 por hora

### Importante:

- Instâncias reservadas já podem ser compradas ou vendidas no AWS Marketplace.
- Cada instância reservada está associada a um tipo de instância específica e a reserva pode ser aplicada somente àquele tipo de instância durante a vigência do contrato.
- Cada instância reservada está associada a uma região e a uma zona de disponibilidade específica, que é fixada durante a vigência do contrato e não pode ser alterada.
- Pode-se comprar uma instância reservada em uma zona de disponibilidade onde já se tenha uma instância em execução e a instância reservada será aplicada automaticamente àquela instância existente.

### 4.2.4.5. Instâncias *spot*

Instâncias *spot* permitem que os clientes façam uma proposta quanto à capacidade do EC2 e executem essas instâncias desde que sua proposta ultrapasse o preço *spot* atual. O preço *spot* oscila com base no fornecimento e na demanda para instâncias, mas os clientes nunca pagarão mais do que o preço máximo que especificaram, garante a Amazon. As instâncias *spot* são complementares às instâncias *on demand* e reservadas, permitindo mais uma opção para a obtenção da capacidade computacional.

As instâncias *spot* fornecem capacidade para que clientes comprem processamento sem compromisso prévio, por taxas horárias normalmente menores do que a taxa *on demand*. Elas possuem desempenho exatamente igual aos das instâncias *on demand* ou reservadas e podem ser solicitadas usando o console de gerenciamento AWS ou através das APIs do EC2.

A **Figura 4-4** ilustra o detalhe da tela do console de gerenciamento AWS

para definição dos parâmetros da instância *spot*. A AWS fornece o preço corrente para determinada AMI (escolhida na tela anterior do console de gerenciamento) e o tipo de instância requerido.

The screenshot shows the 'Request Instances Wizard' interface. The 'INSTANCE DETAILS' tab is selected. The 'Number of Instances' is set to 1, and the 'Instance Type' is 'Micro (t1.micro, 613 MiB)'. The 'Launch as an EBS-Optimized instance (additional charges apply)' checkbox is unchecked. The 'Request Valid Until' field is highlighted with a red box. Other fields include 'Current Price' (\$0.004), 'Max Price' (\$0.045), 'Launch Group' (empty), 'Launch Into' (set to EC2), 'Availability Zone' (sa-east-1a), and 'Availability Zone Group' (empty). Buttons at the bottom include '< Back', 'Continue >', and 'Cancel'.

Figura 4-4 Definição da instância *spot*

O EC2 muda o preço *spot* periodicamente à medida que novas solicitações são recebidas e à medida que a capacidade disponível do EC2 é alterada. Em geral, o preço *spot* muda uma vez por hora. A Amazon publica o preço *spot* atual e os preços históricos para instâncias *spot* por meio da API ou pelo console de gerenciamento AWS; estes dois preços publicados podem ajudar a avaliar os níveis e as oscilações no preço *spot* ao longo do tempo.

As instâncias *spot* são recomendadas para:

- Aplicativos que têm períodos de início e de término flexíveis.
- Aplicativos que são viáveis somente por preços computacionais muito baixos.
- Aplicativos cujos usuários possuem necessidades computacionais urgentes para grandes quantidades de capacidade adicional.

A instância *spot* permite que a Amazon otimize o uso dos recursos disponíveis ao mesmo tempo em que, para os clientes, oferece uma modalidade de aquisição que flutua conforme oferta e procura. Mas é preciso ter cuidado, pois este tipo de instância para aplicativos que precisam ser utilizados obrigatoriamente em determinados intervalos de tempo pode virar um problema. Como as instâncias *spot* podem ser finalizadas sem aviso, a Amazon sugere que é importante criar aplicativos que possam progredir mesmo se houver uma interrupção e se assegurar de que o aplicativo seja

tolerante a falhas e que lide corretamente com as interrupções.

Para determinar como o preço máximo estipulado se compara aos últimos preços *spot*, pode-se verificar o histórico de preço dos últimos noventa dias por região por meio do console de gerenciamento AWS. Se a oferta de preço máximo exceder o preço *spot* atual, a requisição é atendida e as instâncias serão executadas até que sejam encerradas ou até que o preço *spot* supere o preço máximo (o que ocorrer primeiro). A **Figura 4-5** ilustra a variação dos preços *spot* para uma região e zona específica.

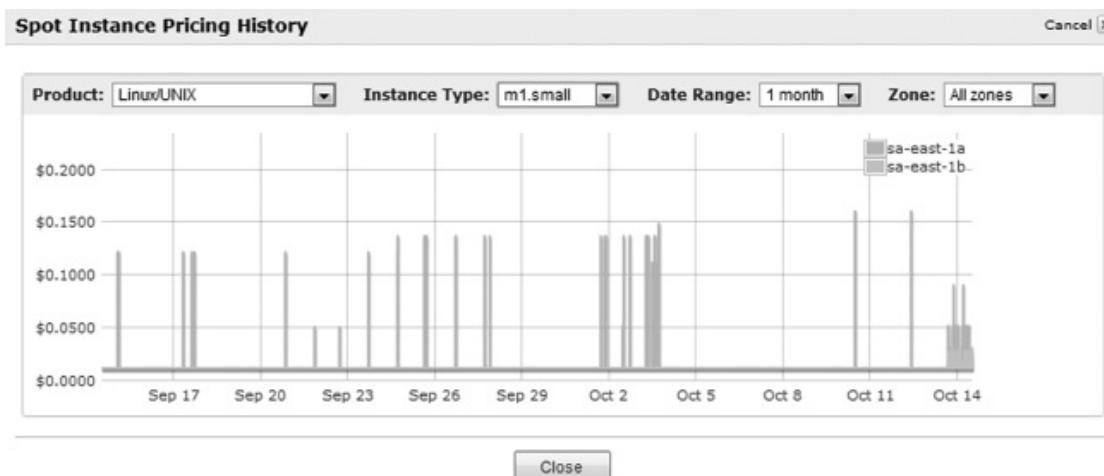


Figura 4-5 Variação de preços *spot* na AWS

Instâncias *spot* estão disponíveis para os sistemas operacionais Linux/UNIX, SUSE Linux e Microsoft Windows Server.

É importante reforçar que as principais diferenças entre as instâncias *spot* e as instâncias *on demand* são o início imediato ou não das instâncias, a variação do preço por hora e a finalização de instâncias *spot* individuais pelo próprio EC2.

A **Tabela 4-8** a seguir mostra o menor preço *spot* obtido por região e por tipo de instância (atualizado a cada cinco minutos). Além de Linux/UNIX e Windows, também são oferecidas instâncias *spot* para o EC2 com o SUSE Linux Enterprise Server.

Tabela 4-8 EC2 – Preço de instâncias *spot*

	Uso do Linux/UNIX	Uso do Windows
<b>Instâncias spot – padrão</b>		
Pequeno (padrão)	\$0,011 por hora	\$0,021 por hora
Médio	\$0,022 por hora	\$0,042 por hora
Grande	\$0,042 por hora	\$0,082 por hora

Extragrande	\$0,084 por hora	\$0,164 por hora
<b>Instâncias spot – micro</b>		
Micro	\$0,004 por hora	\$0,007 por hora
<b>Instâncias spot – mais memória</b>		
Extragrande	\$0,062 por hora	\$0,097 por hora
Dupla extragrande	\$0,123 por hora	\$0,193 por hora
Quádrupla extragrande	\$0,246 por hora	\$0,386 por hora
<b>Instâncias spot – mais CPU</b>		
Médio	\$0,024 por hora	\$0,057 por hora
Extragrande	\$0,096 por hora	\$0,226 por hora

A **Tabela 4-9** ilustra os preços para transferência de dados.

**Tabela 4-9 EC2 – Preço de transferência de dados**

Transferência de dados para fora	
Primeiro 1 GB/mês	\$0,00 por GB
Até 10 TB/mês	\$0,25 por GB
Próximos 40 TB/mês	\$0,23 por GB
Próximos 100 TB/mês	\$0,21 por GB
Próximos 350 TB/mês	\$0,19 por GB

Não há nenhuma taxa de transferência de dados entre o EC2 e outros Amazon Web Services dentro da mesma região. Dados transferidos entre instâncias EC2, localizadas em diferentes zonas de disponibilidade na mesma região, serão cobrados como transferência de dados regional. Para os dados transferidos entre os serviços AWS em diferentes regiões serão cobradas taxas de transferência de dados de internet em ambas as extremidades da transferência. A utilização de outros Amazon Web Services é cobrada separadamente do EC2.

- **Transferência de dados na mesma zona de disponibilidade:** \$0,00 por GB. Todos os dados transferidos entre instâncias na mesma zona de disponibilidade usando endereços IP privados.
- **Transferência de dados na mesma região:** \$0,01 por GB. Todos os dados transferidos entre instâncias em diferentes zonas de disponibilidade na mesma região.
- **Transferência de dados utilizando endereço público, IP elástico e ELB:** \$0,01 por entrada/saída GB. A opção por se comunicar usando um endereço público ou IP elástico ou o ELB dentro da rede do EC2 forçará o pagamento de taxas de transferência de dados regionais, mesmo se as instâncias estiverem na mesma zona de disponibilidade. Para transferência de dados dentro da mesma zona de disponibilidade, pode-se facilmente evitar esta taxa (e obter melhor desempenho da rede) usando um IP privado sempre que possível.
- **Elastic Block Store (EBS):**
  - **Volumes do EBS:**
    - \$0,19 o armazenamento provisionado por GB/mês.
    - \$0,14 por um milhão de requisições I/O.
    - *Snapshots* EBS para Amazon S3: \$0,17 dados armazenados por GB/mês.
- **Endereços IP elásticos (Elastic IPs):**
  - Sem custo para endereços IP elásticos quando em utilização.
    - \$0,01 por endereço IP elástico não inscrito por hora completa.
  - \$0,00 pelos cem primeiros remapeamentos/mês de endereço IP elástico.
  - \$0,10 por remapeamento adicional/mês acima de cem de endereço IP elástico.
- **CloudWatch:**
  - Monitoramento detalhado para as instâncias do EC2: \$4,725 por instância por mês, fornecido em frequência de um minuto.
  - Monitoramento básico para as instâncias do EC2: \$0,00 por instância por mês, fornecido em frequência de cinco minutos.
  - Monitoramento por métricas personalizadas: \$0,675 por métrica por mês.
- **Elastic Load Balancing (ELB):**

- \$0,034 por ELB-hora (ou hora parcial).
- \$0,011 por GB de dados processados por um ELB.

**Auto Scaling:** é ativado pelo CloudWatch e não possui taxas adicionais a serem cobradas. Cada instância iniciada pelo *Auto Scaling* já possui capacidade automática de monitoramento, e as taxas apropriadas do CloudWatch serão aplicadas.

#### 4.2.5. Precificação S3

O preço do S3 leva em consideração o seguinte:

- **Classe do storage.** O armazenamento padrão é projetado para fornecer uma durabilidade de 99,999999999%. O armazenamento de redundância reduzida (*Reduced Redundancy Storage – RRS*) é uma opção de armazenamento dentro do S3 que pode ser utilizado para reduzir os custos de armazenamento de dados não críticos. Armazenamento de redundância reduzida é projetado para fornecer uma durabilidade de 99,99%. Cada classe tem taxas diferentes.
- **Armazenamento.** O número e o tamanho dos objetos armazenados nos *buckets* S3, assim como o tipo de armazenamento.
- **Pedidos (requests).** O número e o tipo de solicitações. Requisições GET incorrem em custos com tarifas diferentes de outros pedidos como PUT e solicitações de cópia.
- **Transferência de dados.** A quantidade de dados transferidos para fora da região do S3.

A Tabela 4-10 ilustra os preços para armazenamento no S3.

Tabela 4-10 S3 – Preço de armazenamento

	<b>Armazenamento padrão</b>	<b>Redundância de armazenamento reduzida (RRS)</b>
Primeiro 1 TB/mês	\$0,130 por GB	\$0,104 por GB
Próximos 49 TB/mês	\$0,110 por GB	\$0,088 por GB
Próximos 450 TB/mês	\$0,095 por GB	\$0,076 por GB
Próximos 500 TB/mês	\$0,090 por GB	\$0,072 por GB

Próximos 4.000 TB/mês	\$0,080 por GB	\$0,064 por GB
Mais de 5.000 TB/mês	\$0,075 por GB	\$0,050 por GB

A **Tabela 4-11** ilustra os preços das solicitações considerando a região de São Paulo.

**Tabela 4-11 S3 – Preço das solicitações (*requests*)**

Definição de preço	
Solicitações PUT, COPY, POST ou LIST	\$0,014 por 1.000 solicitações
Solicitações GET e todas as outras	\$0,014 por 10.000 solicitações
Não há custo por solicitações de exclusão	

A **Tabela 4-12** ilustra os preços para transferência de dados.

**Tabela 4-12 S3 – Preço de transferência de dados**

Definição de preço	
<b>Transferência de dados para fora</b>	
Primeiro 1 GB/mês	\$0,000 por GB
Até 10 TB/mês	\$0,250 por GB
Próximos 40 TB/mês	\$0,230 por GB
Próximos 100 TB/mês	\$0,210 por GB
Próximos 350 TB/mês	\$0,190 por GB

Transferência de dados “para dentro” e “para fora” refere-se à transferência para dentro e para fora de uma região do S3. Não há taxa para dados transferidos dentro de uma região do S3 por meio de uma solicitação COPY. Dados transferidos por meio de uma solicitação COPY entre regiões são cobrados baseados em taxas regulares.

#### **4.2.6. Precificação Glacier**

O Glacier, forma de arquivamento (*archive*) recentemente lançado pela AWS, cobra \$0,01 por GB e \$0,050 por mil solicitações de *upload* e *retrieval*, considerando a região leste dos EUA.

#### 4.2.7. Precificação do suporte AWS

Todos os níveis do suporte AWS incluem um número ilimitado de casos para os quais existe suporte, sem nenhum tipo de contrato a longo prazo. Do mesmo modo, com os níveis Empresa e Negócios, à medida que gastos com a AWS aumentam, ganham-se descontos por volume no suporte AWS.

A **Tabela 4-13** ilustra o preço do suporte AWS.

**Tabela 4-13 Suporte AWS – preços**

	<b>Basic Desenvolvedor</b>	<b>Negócios</b>	<b>Enterprise</b>	
<b>Definição de preço</b>	Incluído	\$49/mês	Mais de \$100 - ou - 10% de uso mensal da AWS para os primeiros de 0 a \$10 mil 7% de uso mensal da AWS de \$10 mil a \$80 mil	Mais de \$15.000,00 - ou - 10% de uso mensal da AWS para os primeiros de 0 a \$150 mil 7% de uso mensal da AWS de \$150 mil a \$500 mil
			5% de uso mensal da AWS de \$80 mil a \$250 mil 3% de uso mensal da AWS de mais de \$250 mil	5% de uso mensal da AWS de \$500 mil a \$1 milhão 3% de uso mensal da AWS de mais de \$1 milhão

#### 4.2.8. Uso gratuito

O uso gratuito permite fazer um “test drive” de vários produtos da AWS. Pode-se ganhar experiência com o sistema durante um ano sem pagar e fazer uma decisão criteriosa sobre o seu uso. Quando o uso livre acaba ou se a aplicação excede o nível de uso gratuito, as taxas normais passam a ser cobradas de acordo com a precificação em vigor. A conta chegará na próxima fatura do cartão de crédito, que deverá ser cadastrado antecipadamente. É importante ressaltar que a Amazon atualmente só fatura via cartão de crédito.

O uso gratuito da AWS é disponibilizado por um ano após a conta AWS ser aberta. As condições de elegibilidade e de uso podem ser encontradas no guia “AWS Free Usage Tier: Getting Started Guide”, disponível para download no site da AWS. A forma de começar a usar a AWS gratuitamente é lançar uma microinstância EC2, a menor instância possível. Esta instância deve ser baseada em uma imagem. Pode-se utilizar o *wizard* disponível para especificar uma AMI. A AWS marca com uma estrela as AMIs disponíveis para uso gratuito.

A **Figura 4-6** ilustra uma típica AMI para uso gratuito.



Figura 4-6 AMI disponível para uso gratuito

Lembre-se que uma instância EC2 é equivalente a um servidor virtual e, portanto, existe um custo quando se aloca uma instância para rodar um aplicativo. Informações de como lançar, se conectar e terminar uma instância EC2 podem ser encontradas no guia “Amazon Elastic Compute Cloud: Getting Started Guide”, disponível para download no site da AWS.

A Amazon permite que novos clientes possam começar a utilizar o EC2 gratuitamente de imediato. Após a inscrição, os usuários receberão os seguintes serviços AWS todos os meses por um ano:

- 750 horas de uso da instância Linux Micro do Amazon EC2 (613 MB de memória e suporte à plataforma de 32 e 64 bits) – horas suficientes para executar continuamente todos os meses.\*
- 750 horas de uso da instância Microsoft Windows Server Micro do EC2 (613 MB de memória e suporte à plataforma de 32 e 64 bits) – horas suficientes para executar continuamente todos os meses.\*
- 750 horas de *Elastic Load Balancer*, além de 15 GB de processamento de dados.\*
- 30 GB de *Elastic Block Storage*, além de dois milhões de E/S e 1 GB de armazenamento de *snapshot*\*.
- 5 GB de armazenamento padrão do Amazon S3, vinte mil solicitações GET e duas mil solicitações PUT\*. GET e PUT são requisições web services feitas na base de dados armazenada no S3.
- Armazenamento de 100 MB, cinco unidades de capacidade de gravação e dez unidades de capacidade de leitura para o DynamoDB.\*\*
- 25 horas de utilização da máquina do SimpleDB e 1 GB de armazenamento.\*\*
- Mil execuções de fluxo de trabalho do Amazon SWF podem ser iniciadas gratuitamente. Um total de dez mil tarefas de atividades, sinais, temporizadores e marcadores e trinta mil dias de trabalho efetivo também podem ser utilizados gratuitamente.\*\*
- Cem mil solicitações de Simple Queue Service.\*\*
- Cem mil solicitações, cem mil notificações de HTTP e mil notificações de e-mail para Simple Notification Service.\*\*
- Dez métricas do CloudWatch, dez alarmes e um milhão de solicitações de API.\*\*
- 15 GB de largura de banda para fora agregada em todos os serviços AWS.\*

\* O nível gratuito está disponível somente para novos clientes da AWS, por doze meses, a partir da data de cadastro na AWS. Quando o uso gratuito expirar ou se o uso do aplicativo ultrapassar os níveis de uso você simplesmente pagará taxas de serviço padrão. Há restrições; consulte os termos da oferta para obter mais detalhes.

\*\* Esses níveis gratuitos não expiram após doze meses e estão disponíveis para clientes AWS existentes e novos indefinidamente.

É importante ressaltar que nem todos os serviços AWS são elegíveis para o uso gratuito e existem restrições de uso que precisam ser observadas e estão relacionadas no guia citado anteriormente. Pode-se também misturar produtos de uso livre com produtos pagos. O uso gratuito vale para todas as regiões da AWS.

A AWS inclui a opção Windows para uso gratuito. É oferecido o uso gratuito das instâncias “t1.micro” para serem executadas no Microsoft Windows Server 2012 ou 2008 R2, com limite máximo de 750 horas de uso da instância por mês. Mais uma vez, é importante ressaltar que é cobrado o preço do EC2 padrão para utilização que ultrapasse estas 750 horas.

Gerenciadores de banco de dados do tipo RDS para MySQL e RDS para SQL Server foram adicionados recentemente à oferta gratuita da AWS. Novos clientes da AWS podem usar um desses bancos de dados em uma instância micro por 750 horas por mês, mais 20 GB de storage, dez milhões de I/Os e 20 GB de backup, tudo isso durante um ano. A combinação dessa nova capacidade com os recursos já disponíveis na oferta gratuita (750 horas por mês de instância micro Linux e 750 horas por mês de instância Windows micro por um ano) permite construir e executar uma aplicação web multicamadas completa sem custo algum.

A AWS também tem uma oferta específica para educação. O professor de uma universidade (de graduação, pós-graduação, mestrado ou doutorado) pode requisitar créditos da AWS para ele e para seus alunos. Os créditos são concedidos para o professor e para cada um dos seus alunos. Basta abrir gratuitamente uma conta ou já ter uma conta na AWS e preencher o formulário para educadores no link <http://aws.amazon.com/pt/education/>.

## 4.2.9. Cálculo de preços

As ferramentas descritas a seguir ajudam a avaliar em termos de custo a migração de aplicativos existentes para a nuvem AWS ou a inicialização de novos aplicativos na nuvem AWS sob uma perspectiva de custos e retorno sobre o investimento. Reforça-se aqui que este não é o único critério a ser considerado para a migração para a nuvem.

### 4.2.9.1. Ferramentas

- **Calculadora mensal simples (*Simple Monthly Calculator*):** a calculadora mensal simples da AWS inclui todos os serviços em todas as regiões para ajudar a calcular a conta mensal na AWS. Com essa ferramenta, é possível adicionar, modificar e remover serviços da “conta” e ela irá recalcular automaticamente cobranças mensais estimadas. A calculadora também mostra os exemplos comuns de uso dos clientes e sua utilização, como recuperação de desastre e backup ou aplicativo da web. A calculadora se encontra disponível em [http://calculator.s3.amazonaws.com/calc5.html?Ing=pt\\_BR](http://calculator.s3.amazonaws.com/calc5.html?Ing=pt_BR).
- **Calculadora de comparação de custo do EC2 (*EC2 Cost Comparison Calculator*):** esta calculadora é baseada no Microsoft Excel e é projetada para ajudar a quantificar os benefícios econômicos diretos (ou custos) da computação em nuvem. A planilha completa é fornecida como ponto inicial, de modo que se possa usar ou modificar as hipóteses padrão com base nos aspectos exclusivos do seu negócio, com o objetivo de determinar o custo anual do EC2 em comparação com o compartilhamento de locação ou recursos de computação locais.
- **Calculadora de comparação de custo do RDS (*RDS Cost Comparison Calculator*):** esta calculadora para comparação de custo é baseada no Microsoft Excel e é projetada para ajudar a quantificar os benefícios econômicos diretos (ou custos) de empregar e gerenciar os bancos de dados em nuvem. Pode-se usar a planilha como ponto inicial para a análise e modificar as hipóteses padrão com base nos aspectos exclusivos do negócio, com o objetivo de comparar o custo anual da implantação dos bancos de dados no Amazon RDS com o compartilhamento de locação ou opções locais.
- **Software PlanForCloud (<http://www.planforcloud.com>):** permite comparar opções de nuvem, incluindo Amazon Web Services, Google Compute Engine, Windows Azure, Rackspace e Softlayer para diversos cenários, promovendo relatórios de custos detalhados. Nesta ferramenta também é possível avaliar impactos nos custos para diferentes taxas de crescimento e configurações.

#### **4.2.9.2. Uso da calculadora mensal simples**

O preço mensal de manter a arquitetura mostrada na **Figura 4-7** pode ser calculado utilizando a calculadora mensal da AWS.

A arquitetura consta de uma instância EC2, uma instância de banco de dados RDS e um *bucket* do storage S3. Utiliza-se também um endereço IP elástico, e a transferência de dados para fora da instância EC2 é de 15 GB/mês. No caso do gerenciador de banco de dados RDS, este possui quinze milhões de requests de I/O por mês e utiliza 40 GB para backup.

A instância EC2 é efêmera do tipo *m1.small* e para este caso o storage também é efêmero. A instância RDS é do tipo efêmera *db.m1.small* e utiliza uma única zona de disponibilidade.



Figura 4-7 Arquitetura para site web

A estimativa da fatura mensal utilizando a ferramenta *Simply Monthly Calculator* neste caso é de US\$ 147,50 para a região US-East (Virginia).

## 4.3. Faturamento

A conta da AWS é responsável pelo pagamento de todo o uso incorrido pelos usuários definidos nela. Um usuário individual não tem sua própria configuração de pagamento ou faturamento e não pode se inscrever individualmente nos serviços da AWS.

A conta da AWS controla e é responsável por todos os dados e recursos como objetos do S3 e instâncias do EC2 criadas por seus usuários. Os recursos criados por um usuário individual parecerão que foram criados pela conta da AWS.

### 4.3.1. Atividade da conta (*account activity*)

A atividade da conta é uma declaração detalhada das taxas. Esta página inclui encargos de uso estimado e faturas para o mês atual. Pode também exibir declarações e faturas de meses anteriores.

As faturas podem vir como arquivos PDF no final de um ciclo de faturamento ou para tarifas únicas.

Para obter uma fatura de arquivo PDF:

1. Vá para “Atividade da conta” (*account activity*).
  2. Selecione o mês de instrução na opção “Selecione uma instrução diferente”.
  3. Na seção “Resumo”, visualize as taxas cobradas.

Pode-se também obter um *Downloadable Report* (arquivo CSV).

Na página de atividades da conta, o titular pode fazer download de um relatório detalhado em formato CSV. O relatório de estimativa é atualizado várias vezes por dia e contém dados de mês-a-dia para custos de cada conta, discriminados por produto AWS e cada tipo de uso. O relatório está disponível para meses anteriores, selecionando o período de instrução.

Se for feita uma inscrição para acesso programático de faturamento, poderá se obter o relatório CSV com encargos estimados e finais em um *bucket S3* especificado. O arquivo contém encargos da conta, discriminados por produto AWS e cada tipo de uso.

A opção *billing preferences* (preferências de pagamento) permite utilizar o programmatic Access.

### **4.3.2. Consolidação do faturamento (*consolidated billing*)**

O faturamento consolidado e o IAM são recursos complementares. O IAM foi tema do capítulo 3. O faturamento consolidado permite consolidar o pagamento de várias contas da AWS dentro da sua empresa ao designar uma única conta de pagamento. Pode-se visualizar uma exibição combinada de custos da AWS incorridos por todas as contas, assim como obter um relatório de custo detalhado para cada uma das contas individuais da AWS associadas à conta de pagamento.

O faturamento consolidado também poderá diminuir os custos totais, já que o uso em todas as contas poderia ajudar a atingir camadas de volume com preços menores mais rapidamente.

Para começar a usar o faturamento consolidado, deve-se efetuar login na conta que se deseja designar como a conta pagante e clicar em “Enviar uma solicitação de faturamento consolidado”. Durante a conexão, verifica-se o número de telefone informado e o método de pagamento. Assim que esta etapa for concluída, pode-se começar a enviar solicitações para outras contas para adicioná-las à fatura consolidada. Os proprietários dessas contas receberão um e-mail com um link para aceitar a solicitação. Quando eles tiverem aceitado, a conta será adicionada à fatura consolidada.

A qualquer momento o administrador da conta ou o proprietário de outra conta poderá remover a conta da fatura consolidada ao retornar à página de faturamento consolidado.

Na página de atividade da conta é possível exibir os custos totais de todas as contas na fatura consolidada. Pode-se também fazer download de um relatório de custo detalhado em formato CSV (*Comma Separated Values*) ou XML.

O faturamento consolidado é um recurso de faturamento, e as permissões normais da conta da AWS ainda serão aplicadas. A conta de pagamento não tem permissões extras ou controle sobre as outras contas na fatura consolidada. Cada conta permanece independente.

Com a conta consolidada é possível ter uma conta simples para todos os recursos AWS. A **Figura 4-8** ilustra esta situação.

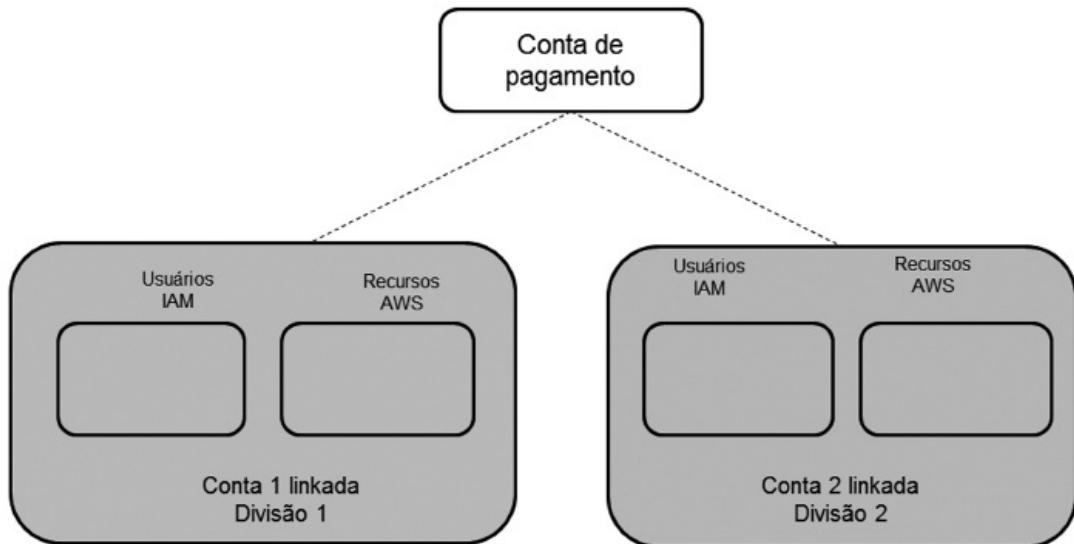


Figura 4-8 Consolidação da conta na AWS – situação 1

Com o IAM também é possível que uma conta seja a conta do pagamento e que pague por seu uso e pelo uso dos usuários linkados. Cada conta IAM linkada não precisa ter um método de pagamento com a AWS, somente a conta de pagamento necessita. A **Figura 4-9** ilustra esta situação.

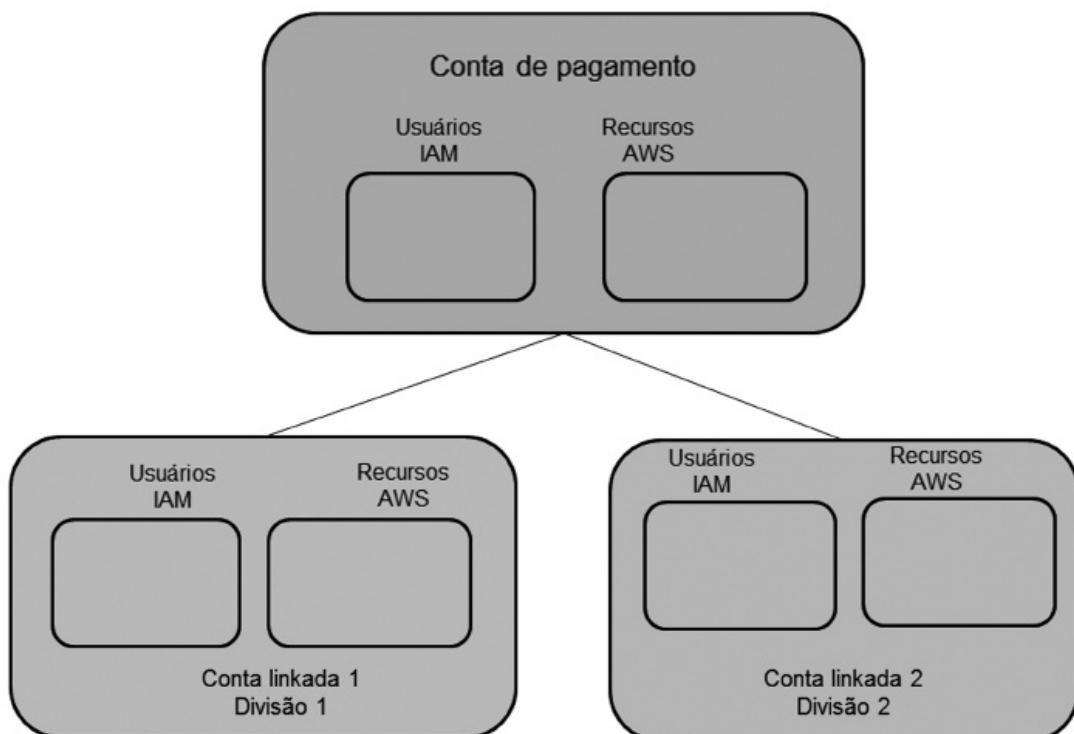


Figura 4-9 Consolidação da conta na AWS – situação 2

Para adicionar uma conta para o faturamento consolidado (e transformá-la em uma conta vinculada), é necessário:

1. Da página de faturamento consolidado, enviar uma solicitação para o proprietário da conta, para adicioná-la ao faturamento consolidado (basta clicar em “Enviar uma solicitação” e seguir as instruções).

A **Figura 4-10** ilustra esta opção.

#### Enviar uma solicitação de faturamento consolidado

Para adicionar uma conta da AWS à sua declaração de faturamento, envie uma solicitação para o proprietário da conta. Quando o proprietário aceitar sua solicitação, você será responsável pelo pagamento dos encargos da conta daí em diante.

[Enviar uma solicitação](#)

Figura 4-10 Solicitação de faturamento consolidado

2. A AWS envia um e-mail para o proprietário da conta.

3. O proprietário da conta clica em um link no e-mail, faz login no site da AWS quando solicitado e aceita ou nega o pedido.

Se o proprietário da conta aceitar a solicitação, a conta passa a integrar o faturamento. O faturamento consolidado pode adicionar até vinte contas.

### 4.3.3. Método de pagamento (*payment method*)

A Amazon trabalha com cartão de crédito internacional como método de pagamento. O cartão de crédito fornecido será usado para todas as cobranças da AWS.

Se você recebeu um cupom de crédito da AWS, poderá adicioná-lo clicando no link fornecido na página mostrada na **Figura 4-11**. A Amazon reforça que fará a cobrança do saldo de crédito da AWS antes de lançar a cobrança no cartão de crédito.

<b>Método de pagamento</b>	<b>Bem-vindo Manoel Veras de Sousa Neto  </b>
	<a href="#">Desconectar</a>
	Número da conta 4520-0109-3923
<b>Método de pagamento atual</b>	
Type: MasterCard	<a href="#">Editar este cartão</a>
Número do cartão de crédito: ****-****-****-****	
Data de validade: 05/2015	
Nome do portador do cartão: manoel veras de sousa nt	
Endereço de faturamento:	Manoel Veras de Sousa Neto Alameda Jardins da Arrabida 1213 APT 9-D Vila Nova de Gaia Afurada, 4400-478 Portugal Phone: 227-724-375

Figura 4-11 Método de pagamento

#### 4.3.4. Relatórios de utilização (*usage reports*)

A Amazon fornece relatórios de utilização acionados via formulário mostrado na **Figura 4-12**. Pode-se fazer download de um relatório do uso para o serviço selecionado.

The screenshot shows the 'Relatórios de utilização' (Usage Reports) page. At the top right, it says 'Bem-vindo Manoel Veras de Sousa Neto | Desconectar' and 'Número da conta 4520-0109-3923'. Below this, a section titled 'Download do relatório de uso' contains instructions: 'Usando o formulário abaixo, você poderá criar um download de um relatório do seu uso para o serviço selecionado:'. A dropdown menu 'Serviço:' is set to 'Amazon ElastiCache'. Below it are four filter dropdowns: 'Usage Types: All Usage Types', 'Operação: All Operations', 'Time Period: Current billing period', and 'Report Granularidade: Hours'. At the bottom are two buttons: 'Download report (XML)' and 'Download report (CSV)'. A note below the filters states: 'A versão CSV do seu Relatório de uso pode ser aberta em qualquer aplicativo de planilha, como Excel.' Another note at the bottom left says: 'OBSERVAÇÃO: relatórios de uso muito grandes poderão ser truncados. Verifique a última linha do arquivo baixado quanto à avisos ou mensagens de erro. Se você vir uma mensagem indicando que o relatório foi truncado, poderá tentar baixar relatórios menores ao solicitar um período menor ou diminuir a granularidade de horas para diariamente ou mensalmente.'

Figura 4-12 Relatórios de utilização

#### 4.3.5. Preferências de faturamento (*billing preferences*)

As preferências de faturamento recaem em quatro possibilidades:

- **Monthly Report:** trata-se de uma declaração detalhada de todas as taxas AWS durante um período de faturamento. O relatório é gerado para custos estimados no final do mês. A AWS atualiza os encargos estimados várias vezes por dia.
- **Programmatic Access:** trata-se de disponibilizar dados do relatório CSV no S3.
- **Detailed Billing Report:** permite obter uma visão por hora de uso dos recursos AWS e taxas incidentes.
- **Cost Allocation Report:** permite à empresa acompanhar e organizar os custos AWS. O programmatic access deve estar habilitado.

### 4.4. Referências bibliográficas

Amazon Web Services. **AWS Account Billing, Version 1.0.** 2012.

Amazon Web Services. **How AWS Pricing Works.** December, 2011.

Amazon Web Services. **The Economics of the AWS Cloud vs. Owned IT Infrastructure.** December, 2009.

Amazon Web Services. **User Guide:** Amazon EC2 Cost Comparison Calculator. July, 2011.

Blog GIGAOM. **Which is less expensive:** Amazon or self-hosted? 2012.

Blog GIGAOM. **Why Amazon and Salesforce are pulling away from the cloud pack.** 2012.

<http://aws.amazon.com/pt/economics/>

## ***PARTE II – SERVIÇOS DE INFRAESTRUTURA***

---

# 5. Computação

## 5.1. Introdução

O serviço de computação responde pela capacidade de processamento da AWS. Ele é um web service adquirido e pago por demanda. Toda a lógica da computação de nuvem passa pela mudança na forma de adquirir e pagar o processamento. O servidor convencional é substituído por uma instância virtual.

Neste capítulo os web services Elastic Compute Cloud (EC2) e Elastic MapReduce, que respondem pela capacidade de processamento da AWS, são explicados. O Elastic Load Balancing (ELB), parte importante do web service EC2, também é explicado de forma separada. Ele é fundamental em projetos de recuperação de desastres.

## 5.2. Elastic Compute Cloud (EC2)

### 5.2.1. Conceito

O EC2 é um web service que permite às empresas clientes e aos desenvolvedores utilizar recursos de processamento de acordo com a demanda do aplicativo e pagar pelo uso. No EC2 paga-se com base no tipo de instância e no uso por hora. Servidores são instâncias no EC2, conforme dito no capítulo 1. Essas instâncias são executadas dentro do ambiente de DATACENTERS próprios da Amazon, descritos no capítulo 2.

O EC2 proporciona aos aplicativos a capacidade de:

- Configurar os requisitos de computação instantaneamente.
- Ajustar a capacidade com base na demanda.

O console de gerenciamento do EC2 permite obter e configurar a capacidade de forma simples. O EC2 reduz para minutos o tempo necessário para lançar e reiniciar novas instâncias, permitindo dimensionar a capacidade, para cima e para baixo, à medida que necessidades computacionais mudam. O EC2 também fornece aos desenvolvedores ferramentas para construir aplicativos resistentes a falhas e isolá-los de situações comuns de falha.

Para utilizar o EC2, deve-se:

- Selecionar um modelo de imagem pré-configurada (Amazon Machine Image – AMI), para começar a usar o serviço imediatamente. Pode-se utilizar um *wizard* do tipo *classic*, um *wizard* do tipo *quick launch* ou o AWS Marketplace, disponibilizados através do console de

gerenciamento da AWS.

- Configurar a segurança e o acesso à rede para a instância EC2 escolhida.
- Escolher o tipo de instância e o sistema operacional associado e em seguida inicializar, monitorar e finalizar quantas instâncias forem necessárias, usando APIs, linhas de comando, SDKs ou o console de gerenciamento AWS.
- Determinar se é necessário executar a instância em vários locais.
- Determinar a necessidade de utilizar IP elásticos e armazenamento persistente.
- Pagar somente pelos recursos utilizados.

O EC2 foi projetado para ser neutro em termos de linguagem de programação, usando interfaces do tipo SOAP e do tipo QUERY.

### **5.2.2. Servidor tradicional *versus* instância AWS**

Uma forma de entender a diferença entre um servidor tradicional e uma instância EC2 é comparar as principais funcionalidades dos dois modelos. Uma comparação básica entre um servidor em rack de 1U<sup>[6]</sup> de altura e uma instância EC2 é mostrada na **Tabela 5-1**.

**Tabela 5-1 Servidor rack de 1U *versus* instância EC2**

Servidor 1U	Instância EC2 Extra Large
1 processador de 2.4Ghz Quad-Core Xeon	Oito ECUs (quatro cores virtuais com duas unidades de computação EC2 cada)
16 GB de memória RAM	15 GB de memória RAM
2 discos SATA de 300 GB	1.690 GB de armazenamento local
1 rede de 1GB Ethernet	1 rede de 1 GB Ethernet

As duas opções parecem similares, mas existem diferenças importantes entre elas:

- A instância EC2 pode ser rapidamente reposta, duplicada e disponibilizada *on demand*.
- A instância EC2 pode crescer rapidamente até oito CPUs lógicas. Para isto só é necessário um reboot e a mudança de configuração via API ou linha de comando, ou através do console de gerenciamento.
- A instância EC2 só custa se estiver ligada.

- Deve-se considerar na escolha da instância EC2 o *overhead* devido ao uso da virtualização.
- Na opção de instância EC2 com storage local, este é efêmero e pode ser excluído se a instância for terminada.

Os serviços de *hosting* baseados em servidores tradicionais fornecem recursos pré-configurados para um período fixo e por um custo predeterminado. O serviço AWS baseado na instância EC2 difere do serviço de *hosting* por permitir utilizar a instância por um período não fixo e por um custo total incluindo armazenamento e transferência de dados que não é predeterminado.

### **5.2.3. Aspectos fundamentais**

#### **5.2.3.1. Referências para a API e para a linha de comando**

É importante saber que existe farta documentação no site da AWS com foco no Amazon EC2. O guia “Amazon Elastic Compute Cloud API Reference (API Version 2012-07-20)” fornece todas as ações da API por função. O guia “Amazon Elastic Compute Cloud CLI Reference (API Version 2012-07-20)” fornece todas as ferramentas para APIs e AMIs.

#### **5.2.3.2. Endereços públicos, privados e elásticos (EIPs)**

Para todas as instâncias EC2 são atribuídos dois endereços IP, um endereço privado e um endereço público, que são mapeados diretamente uns para os outros através de NAT (*Network Address Translation*). Endereços privados só são acessíveis de dentro da rede EC2. Endereços públicos são acessíveis pela internet. O EC2 também fornece um nome DNS interno e um nome DNS público, que mapeiam os endereços IP privados e públicos, respectivamente. O nome DNS interno só pode resolver o endereço IP privado. O nome DNS público resolve o endereço IP público fora da rede EC2 e o endereço IP privado no âmbito da rede EC2. Se for necessário utilizar endereços IP de internet persistentes que podem ser atribuídos e removidos a instâncias de forma imediata, será necessário utilizar endereços IP elásticos.

Para que utilizar um EIP? Se for utilizado um DNS dinâmico para mapear um nome DNS existente para um endereço IP público de uma nova instância, por exemplo, esta operação pode demorar até 24 horas para o endereço IP se propagar através da internet. Como resultado, novas instâncias não podem receber tráfego, enquanto instâncias encerradas continuam a receber pedidos. IPs elásticos vieram para resolver este problema. O EC2 permite utilizar endereços IP elásticos, que são endereços IP estáticos projetados para utilização na nuvem. Endereços IP elásticos são associados a uma conta e não com instâncias específicas.

Quaisquer endereços EIP permanecem associados com a conta até que explicitamente possam ser liberados. Ao contrário dos tradicionais endereços

IP estáticos, no entanto, endereços IP elásticos permitem mascarar uma instância com falha em uma zona de disponibilidade através do remapeamento rápido do endereço IP público (o IP elástico) para qualquer outra instância na mesma conta.

Em resumo, endereços do tipo EIP são IPs públicos que podem ser mapeados ou remapeados para qualquer instância EC2 dentro uma região específica de forma imediata. Os endereços IP elásticos são ideais para construir aplicações tolerantes a falhas. O capítulo 13 exemplifica o uso de IPs elásticos em um projeto de arquitetura tolerante a falhas. A operação para utilizar um EIP é realizada através da API EC2 ou pelo console de gerenciamento.

Três observações importantes sobre EIPs:

- É necessário liberar explicitamente o endereço IP elástico para torná-lo disponível a outros clientes, caso já não seja necessário. Se este não for utilizado paga-se apenas por mantê-lo associado à conta; é importante observar que a qualquer momento é possível ter apenas uma única instância mapeada para um endereço IP elástico.
- Não é necessário ter um endereço IP elástico para todas as instâncias. Como padrão, conforme visto, toda instância vem com um endereço IP privado e um endereço IP público. Todas as contas estão limitadas a ter cinco endereços IP elásticos de rede por região. Endereços IP elásticos são endereços da internet públicos (IPv4) e um recurso escasso. Se for necessário utilizar mais de cinco endereços IP elásticos, é preciso preencher um formulário e enviar para a AWS para que o limite seja aumentado. Quaisquer aumentos serão específicos à região para a qual foram solicitados.
- Na utilização de uma VPC há um “pool” separado de endereços IP elásticos VPC para utilização. Os endereços IP elásticos EC2 não vão trabalhar com instâncias em uma VPC, e os endereços IP elásticos VPC não vão trabalhar com instâncias EC2.

#### 5.2.3.3. Dispositivo raiz (*root device*)

Quando o EC2 foi introduzido, todas as AMIs eram carregadas na própria instância EC2, o que significava que o dispositivo raiz (*root device*) para uma instância iniciada a partir de uma AMI era armazenado na própria instância EC2. Depois da introdução do armazenamento do tipo EBS, introduziram-se também AMIs respaldadas pelo EBS, significando que o dispositivo raiz para uma instância iniciada a partir deste tipo de AMI é um volume do EBS criado de um *snapshot* EBS. Estas duas formas de armazenar o dispositivo raiz traz diferentes implicações na forma de recuperar as instâncias. AMIs respaldadas pelo EBS são persistentes. AMIs armazenadas na instância EC2 não são persistentes.

A **Figura 5-1** ilustra imagens do tipo AMIs dos sistemas operacionais Ubuntu, Windows e RedHat. Observe que quando a AMI utiliza armazenamento do tipo EBS isto está sinalizado no próprio nome da AMI.



Figura 5-1 Tipos de AMIs

#### 5.2.3.4. Mapeamento de dispositivo de bloco (*Block Device Mapping – BDM*)

Cada instância tem uma estrutura de mapeamento de dispositivo de bloco que especifica os dispositivos de bloco anexados à instância. Uma estrutura de mapeamento de dispositivo de bloco contém um ou mais itens, e cada item descreve o mapeamento para um dispositivo de bloco único.

Pode-se exibir o mapeamento de dispositivo de bloco para uma instância no console de gerenciamento da AWS ou descrever a instância com as ferramentas de linha de comando ou com a API. A **Figura 5-2** exibe os dispositivos de bloco de uma instância. Outros dispositivos de bloco podem ser acrescentados à instância nesta mesma caixa de diálogo.

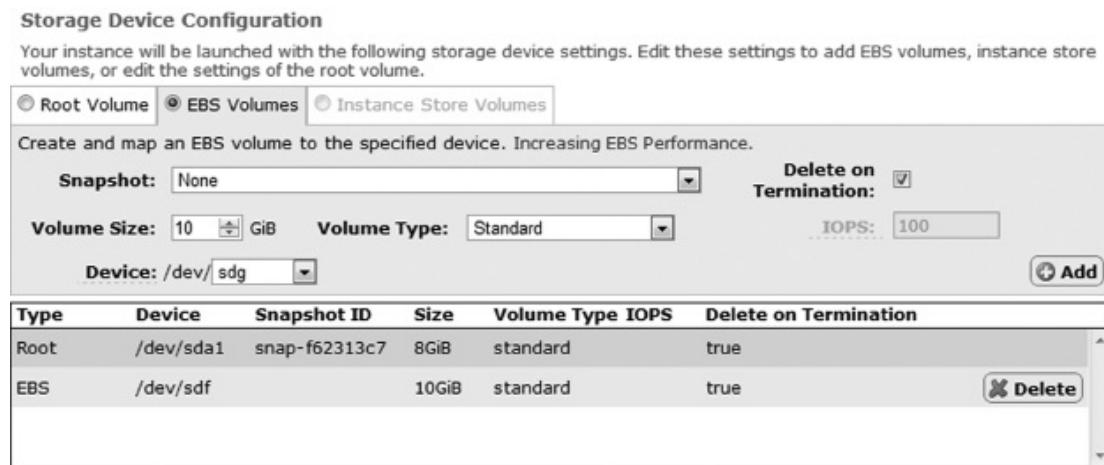


Figura 5-2 Instância e dispositivos de bloco

#### 5.2.4. Tipos de instâncias

As instâncias EC2 existem com diferentes capacidades de desempenho.

Os tipos de instâncias podem ser divididos em:

- Microinstâncias.
- Instâncias padrão.
- Instâncias padrão de segunda geração.
- Instâncias de mais memória.
- Instâncias de mais CPU.
- Instância de mais I/O.
- Instância de mais armazenamento.
- Instância de cluster.
- Instância de cluster GPU.

As microinstâncias fornecem uma pequena quantidade de recursos de CPU e memória. Elas são adequadas para pequenos sites web que utilizam ciclos computacionais periódicos.

As instâncias padrão têm memória, CPU e I/O adequadas para a maioria dos aplicativos de finalidade geral.

Instâncias de mais memória oferecem tamanhos de memória maiores para aplicativos como banco de dados e que utilizam cache em memória; instâncias de mais CPU têm proporcionalmente mais recursos de CPU do que memória e são adequadas para aplicativos que requerem muita computação. Instâncias de mais I/O são adequadas para aplicações intensivas em I/O. As instâncias de mais I/O utilizam discos SSD para aumentar o desempenho e podem fazer parte de um grupo de posicionamento de cluster.

A segunda geração de instâncias padrão tem a mesma relação CPU/memória que as instâncias padrão existentes, mas possui até 50% mais de desempenho absoluto. Essas instâncias são otimizadas para aplicações, tais como codificação de mídia, processamento em lote, cache e serviço web. Essas instâncias lançadas recentemente estão disponíveis por enquanto só na região Northern Virginia.

As instâncias de cluster fornecem uma grande quantidade de CPU vinculada a uma rede de alto desempenho, tornando-as bastante adequadas para aplicativos do tipo *High Performance Compute* (HPC).

As instâncias de cluster GPU fornecem proporcionalmente mais alta utilização de CPU e maior desempenho de rede para aplicativos, beneficiando-se de processamento altamente paralelo, incluindo HPC, renderização e aplicativos de processamento de mídia.

Enquanto instâncias de cluster fornecem a capacidade de criar um grupo de posicionamento de cluster ligado por uma rede de baixa latência, de alta taxa de transferência, as instâncias de cluster GPU fornecem uma opção adicional para aplicativos que podem se beneficiar dos ganhos de eficiência da

potência computacional em paralelo de GPUs. O grupo de posicionamento de cluster é uma entidade lógica que permite a criação de um cluster ao iniciar instâncias como parte de um grupo.

#### **Importante:**

- Para grandes clusters, pode-se aumentar a probabilidade de ser capaz de iniciar o número total de instâncias necessárias em um grupo de posicionamento de cluster ao iniciar o máximo de instâncias possíveis em vez de iniciar poucas instâncias.
- Uma instância interrompida será iniciada como parte do grupo de posicionamento de cluster no qual estava quando da interrupção. Se a capacidade não estiver disponível para ser iniciada em um grupo de posicionamento de cluster, haverá falha na inicialização.

Os usos das instâncias cluster e de cluster GPU diferem de outros tipos de instância do EC2 de três formas:

- As instâncias de cluster e de cluster GPU usam a virtualização baseada em *Hardware Virtual Machine* (HVM) e executam somente AMIs com base neste tipo de virtualização.
- As instâncias de cluster e de cluster GPU exigem a inicialização de uma AMI baseada em armazenamento do tipo EBS.
- As instâncias de cluster e de cluster GPU atualmente estão disponíveis somente na região leste dos EUA (norte da Virgínia) e na região oeste da Europa (Irlanda).

#### **5.2.4.1. Microinstância**

Microinstância (t1.micro) de 613 MB de memória, até dois ECUs, somente utiliza armazenamento EBS, plataforma de 32 bits ou 64 bits.

#### **5.2.4.2. Instância padrão de primeira e segunda geração**

- **Instância pequena (m1.small):** 1,7 GB de memória, um ECU (um virtual core com uma unidade de processamento do EC2), 160 GB de armazenamento de instância local, plataforma de 32 ou 64 bits.
- **Instância média (m1.medium):** 3,75 GB de memória, dois ECUs (um núcleo virtual com duas unidades de processamento EC2 cada), 410 GB de armazenamento de instância local, plataforma de 32 ou 64 bits.
- **Instância grande (m1.large):** 7,5 GB de memória, quatro ECUs (dois núcleos virtuais com duas unidades de processamento EC2 cada), 850 GB de armazenamento de instância local, plataforma de 64 bits.
- **Instância extragrande (m1.xlarge):** 15 GB de memória, oito ECUs (quatro núcleos virtuais com duas unidades de processamento EC2 cada), 1.690 GB de armazenamento de instância local, plataforma de

64 bits.

- **Instância extragrande de segunda geração (m3.xlarge):** 15 GB de memória e treze ECUs em quatro núcleos virtuais.
- **Instância extragrande de segunda geração (m3.2xlarge):** 30 GB de memória e 26 ECUs em oito núcleos virtuais.

#### 5.2.4.3. Instâncias de mais memória

- **Instância extragrande de mais memória (m2.xlarge):** 17,1 GB memória, 6,5 ECUs (dois núcleos virtuais com 3,25 unidades de processamento EC2 cada), 420 GB de armazenamento de instância local, plataforma de 64 bits.
- **Instância dupla extragrande de mais memória (m2.2xlarge):** 34,2 GB memória, treze ECUs (quatro núcleos virtuais com 3,25 unidades de processamento EC2 cada), 850 GB de armazenamento de instância local, plataforma de 64 bits.
- **Instância quádrupla extragrande de mais memória (m2.4xlarge):** 68,4 GB memória, 26 ECUs (oito núcleos virtuais com 3,25 unidades de processamento EC2 cada), 1.690 GB de armazenamento de instância local, plataforma de 64 bits.

#### 5.2.4.4. Instâncias de mais CPU

- **Instância média de CPU de alta performance (c1.medium):** 1,7 GB de memória, cinco ECUs (dois virtual cores com 2,5 unidades de processamento do EC2 cada), 350 GB de armazenamento de instâncias locais, plataforma de 32 ou 64 bits.
- **Instância extragrande de CPU de alta performance (c1.xlarge):** 7 GB de memória, vinte ECUs (oito núcleos virtuais com 2,5 unidades de processamento EC2 cada), 1.690 GB de armazenamento de instância local, plataforma de 64 bits.

#### 5.2.4.5. Instância de mais I/O e de mais armazenamento

- **Instâncias de I/O de alta performance (hi1.4xlarge):** possui oito cores, com total de 35 ECUs, utiliza virtualização HVM e PVM e possui 60,5 GB de RAM com conectividade de 10 Gigabit Ethernet com suporte a grupos de clusters e 2 TB de storage SSD local, visível como dois volumes de 1 TB cada.

Discos SSD são utilizados por este tipo de instância. Usando virtualização PVM, pode-se esperar em torno 120.000 IOPS (*Input/Output Operations per Second*) de leitura randômica e em torno de 10.000 a 85.000 IOPS de escrita randômica, ambos com blocos de 4K. A Amazon recomenda fazer um backup do conteúdo dos discos SSD no S3 regularmente.

As instâncias de mais I/O quádruplas extragrandes podem ser lançadas

em US East (Virginia) e EU West (Irlanda). É possível também comprar instâncias reservadas de mais I/O.

Adrian Cockcroft, em seu artigo “Benefits of moving Cassandra Workloads to SSD”, disponível na internet, sugere as seguintes vantagens que justificam a migração da aplicação Cassandra para a instância de mais I/O.

- A configuração *hi1.4xlarge* é metade do custo da opção anterior utilizada para o mesmo *throughput*.
- A latência das solicitações de leitura foi reduzida de 10ms para 2.2ms.
- A latência de solicitação de 99% foi reduzida de 65ms para 10ms.
- Instâncias de armazenamento maior (*hs1.8xlarge*) são interessantes para aplicações com grandes demandas de armazenamento e desempenho otimizado de I/O sequencial. Cada instância inclui 117 Gb de RAM, 16 virtual cores (35 ECUs) e 48 Tb de armazenamento local de instância, usando 24 HDs que podem disponibilizar 2.4 GB/s de I/O.

A **Figura 5-3** ilustra os tipos principais de instâncias EC2.

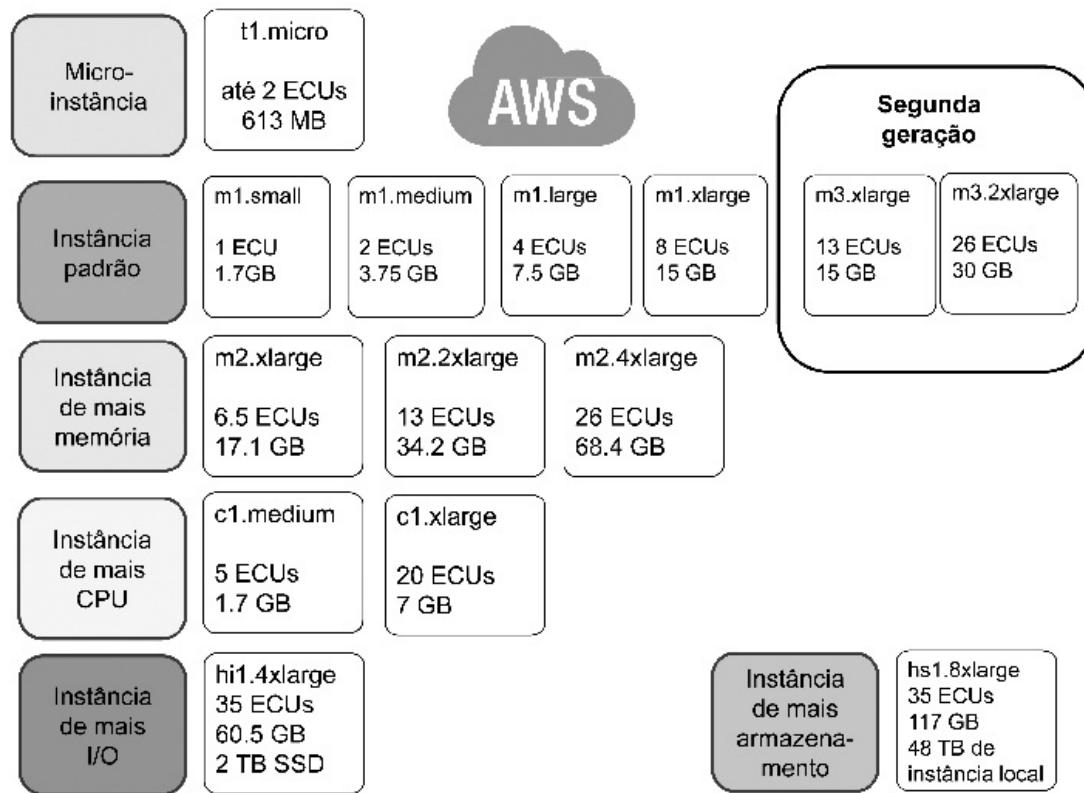


Figura 5-3 Tipos de instâncias

#### 5.2.4.6. Instâncias de cluster compute e cluster GPU

- **Cluster compute quádrupla extragrande (cc1.4xlarge):** 23 GB memória, 33,5 ECUs, 1.690 GB de armazenamento de instância local,

plataforma de 64 bits, Ethernet de 10 Gigabit.

- **Cluster compute óctupla extragrande (cc2.8xlarge):** 60.5 GB de memória, 88 ECUs, 3.370 GB de armazenamento de instância local, plataforma de 64 bits, Ethernet de 10 Gigabit.
- **Cluster GPU compute quádrupla extragrande (cg1.xlarge):** 22 GB memória, 33,5 ECUs, 2 x GPUs NVIDIA Tesla “Fermi” M2050, 1.690 GB de armazenamento de instância local, plataforma de 64 bits, Ethernet de 10 Gigabit.

Não há limite específico imposto pela AWS para utilização de instâncias de cluster compute quádrupla extragrande. Para instâncias de cluster GPU e instâncias de cluster compute óctupla extragrande, pode-se lançar até duas instâncias por conta. Se for necessário utilizar mais instâncias, é necessário preencher um formulário de solicitação de instância do EC2 (selecionando o tipo apropriado de instância).

**Importante:**

- A opção Amazon VPC ainda não está disponível para instâncias de cluster ou de cluster GPU.

### **5.2.5. Cluster de alta performance (*High Performance Cluster – HPC*)**

*High Performance Cluster* (HPC) ou cluster de alta performance é um tipo de supercomputador com altíssima velocidade de processamento e grande capacidade de memória. É usado para cálculos muito complexos e tarefas intensivas, como problemas envolvendo física quântica, mecânica, meteorologia, pesquisas de clima, modelagem molecular e simulações físicas, como simulação de aviões em túneis de vento, simulação da detonação de armas nucleares e investigação sobre a fusão nuclear.

Os HPCs são normalmente baseados na arquitetura *Symmetric Multiprocessors* (SMP), constituída de processadores comerciais conectados a uma memória compartilhada. A arquitetura SMP utiliza amplamente memória cache, e todos os processadores têm igual acesso ao barramento e à memória compartilhada. São mais fáceis de programar que máquinas que se comunicam por troca de mensagens, já que a forma de programação se aproxima daquela feita em sistemas convencionais, mas tem como desvantagem o uso de um barramento de interconexão.

Tradicionalmente, os HPCs foram construídos baseados na aquisição de hardware de baixo custo. O hardware, neste caso, é adquirido em grande quantidade e necessita de grandes instalações para poder funcionar. Nesse tipo de aplicação o software específico de cluster se encarrega de utilizar o poder de processamento das máquinas que compõem o cluster. A

infraestrutura fica toda por conta de quem quer montar o cluster e a manutenção também. Agora é possível montar um cluster HPC em pouco tempo, utilizando os recursos da nuvem AWS.

Recentemente a Amazon, utilizando o chip Intel Xeon E5 (Sandy Bridge), conseguiu obter um poder de processamento elevado para um conjunto destas instâncias em cluster. Este cluster foi construído com 1064 *cc2.8xlarge* instâncias (17.024 núcleos) e alcançou 240.09 TFlops rodando o Linpack, um *benchmark* de alta performance. Este cluster AWS ficou na posição 42 da última lista Top 500 de novembro de 2011 publicada em <http://www.top500.org>.

A **Figura 5-4** ilustra as principais características do cluster AWS.

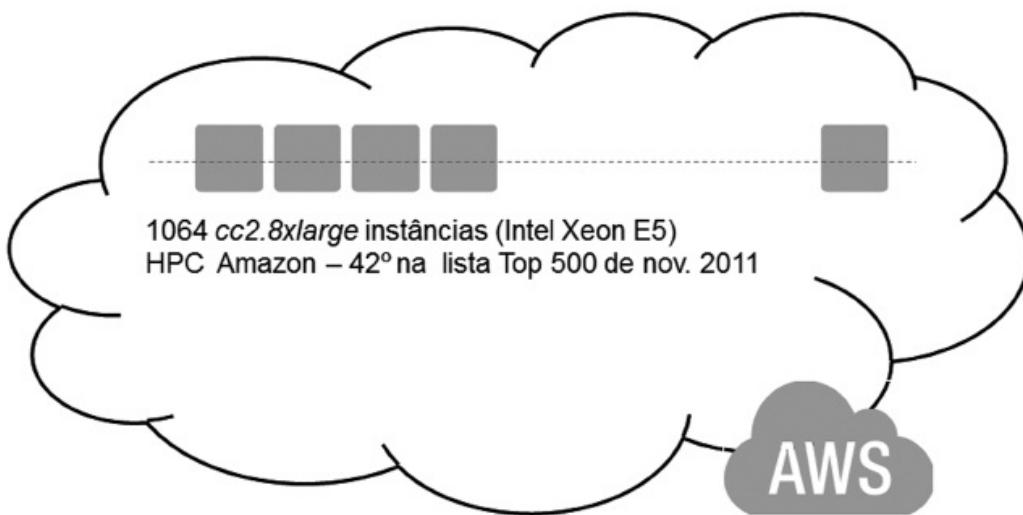


Figura 5-4 HPC na AWS

A Amazon ressalta que HPC com a AWS permite que cientistas e engenheiros resolvam problemas complexos de ciência, de engenharia e de negócios usando aplicativos que exigem grande largura de banda, redes de baixa latência e recursos de computação muito altos adquiridos baseados na demanda. Normalmente, os cientistas e engenheiros esperam em longas filas para acessar clusters compartilhados ou adquirir sistemas HPC muito caros. Agora, utilizando instâncias de cluster EC2, os clientes podem definir suas cargas de trabalho HPC com base em recursos elásticos conforme necessário e economizar, escolhendo modelos de preços acessíveis que atendam às necessidades de utilização.

O HPC no EC2 é montado com a família de computação de cluster e de cluster GPU. Essas instâncias podem ser usadas como qualquer outra instância do EC2, mas também oferecem os seguintes recursos otimizados para aplicativos HPC:

- Podem ser lançadas dentro de um grupo de posicionamento de

cluster. Todas as instâncias lançadas dentro destes grupos têm baixa latência e largura de banda completa de 10 Gbps entre as instâncias. Como muitos outros recursos do EC2, os grupos de posicionamento de cluster são dinâmicos e podem ser redimensionados se necessário.

- Incluem uma arquitetura de processador específica.
- Permitem que os clientes aproveitem ao máximo o desempenho paralelo da NVidia Tesla GPU.

Uma das principais vantagens do EC2 é a capacidade de otimizar recursos sem estar limitado pelo acesso a um cluster de tamanho fixo. É possível iniciar vários clusters simultaneamente sem a necessidade de enviar trabalhos para uma fila. Escolhe-se entre executar um grande conjunto único ou vários clusters menores simultaneamente para resolver limites de escala de aplicação.

### 5.2.6. Importação e exportação de máquinas virtuais

A importação de máquinas virtuais (*VM import*) é uma função disponibilizada pela API AWS que permite que os clientes da AWS importem imagens da máquina virtual VMware para dentro do EC2. Assim, estes podem recuperar investimentos anteriores na criação dessas máquinas virtuais.

O processo de importação *VM import* atualmente suporta imagens do VMware ESX VMDK, do Citrix Xen VHD, do Microsoft Hyper-V VHD ou do tipo RAW.

- VMDK (*Virtual Machine Disk*) é um formato de arquivo que especifica o disco rígido de uma máquina virtual encapsulado dentro de um único arquivo. Normalmente ele é usado por infraestruturas de TI virtuais, como o vSphere da VMware.
- VHD (*Virtual Hard Disk*) é um formato de arquivo que especifica o disco rígido de uma máquina virtual encapsulado dentro de um único arquivo. O formato de imagem VHD é usado por plataformas de Virtualização, como o Microsoft Hyper-V e o Citrix Xen.
- RAW são imagens em estado bruto.

O *VM export* permite exportar instâncias EC2 importadas previamente de volta para o ambiente *on-premise* ou mesmo fazer a exportação de máquinas virtuais criadas no EC2.

### 5.2.7. Controle

O IAM é um web service que permite que os clientes da AWS gerenciem usuários e suas permissões. O IAM foi tema do capítulo 3. O serviço destina-se a organizações com vários usuários ou sistemas que usam os produtos AWS. Com o IAM, é possível gerenciar centralmente usuários, utilizando credenciais

de segurança, tais como chave de acesso e permissões, que controlam quais usuários podem acessar os recursos AWS.

### 5.2.7.1. Permissões EC2

O EC2 tem seu próprio sistema de permissões que abrange os *snapshots* EBS e Amazon Machine Images (AMIs). Não há nenhum sistema ACL ou política para conceder permissões para lançar AMIs ou criar volumes a partir de *snapshots*. Em vez disso, a API EC2 permite modificar os atributos de uma AMI ou de um *snapshot* para dar essas permissões a uma conta AWS. Todos os usuários da conta AWS terão esta permissão.

A **Figura 5-5** ilustra como funciona o sistema de permissões no EC2. Cada AMI tem um atributo *LaunchPermission* que pode ser definido para um ou mais IDs de conta AWS para partilhar a AMI com estas contas. Neste caso, as contas AWS 1 e AWS 2 possuem esta permissão. No outro exemplo cada *snapshot* EBS possui um atributo *VolumePermission* semelhante. Neste caso as contas AWS 3, AWS 4 e AWS 5 possuem esta permissão.

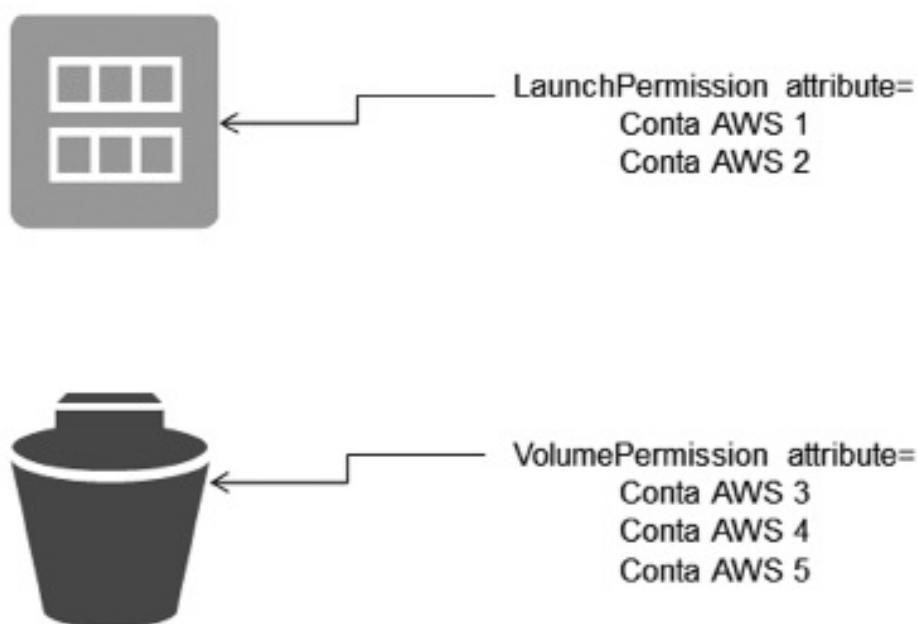


Figura 5-5 Permissões EC2

### 5.2.7.2. Permissões EC2 com IAM

Usar o IAM com o EC2 não muda como a API EC2 compartilha AMIs e *snapshots* com outras contas AWS. No entanto, condições IAM podem ser utilizadas para especificar as ações que um usuário em uma conta AWS pode realizar com recursos do EC2.

A **Figura 5-6** ilustra uma típica política baseada no uso do IAM. Uma política pode dar permissão ao Developers Group, por exemplo, para usar unicamente as funções *RunInstances*, *StopInstances*, *StartInstances*,

*TerminateInstances* e *DescribeInstances*. Eles, os participantes do grupo, podem utilizar essas funções com qualquer AMI vinculada à conta AWS, incluindo AMIs públicas ou qualquer AMI compartilhada com a conta.

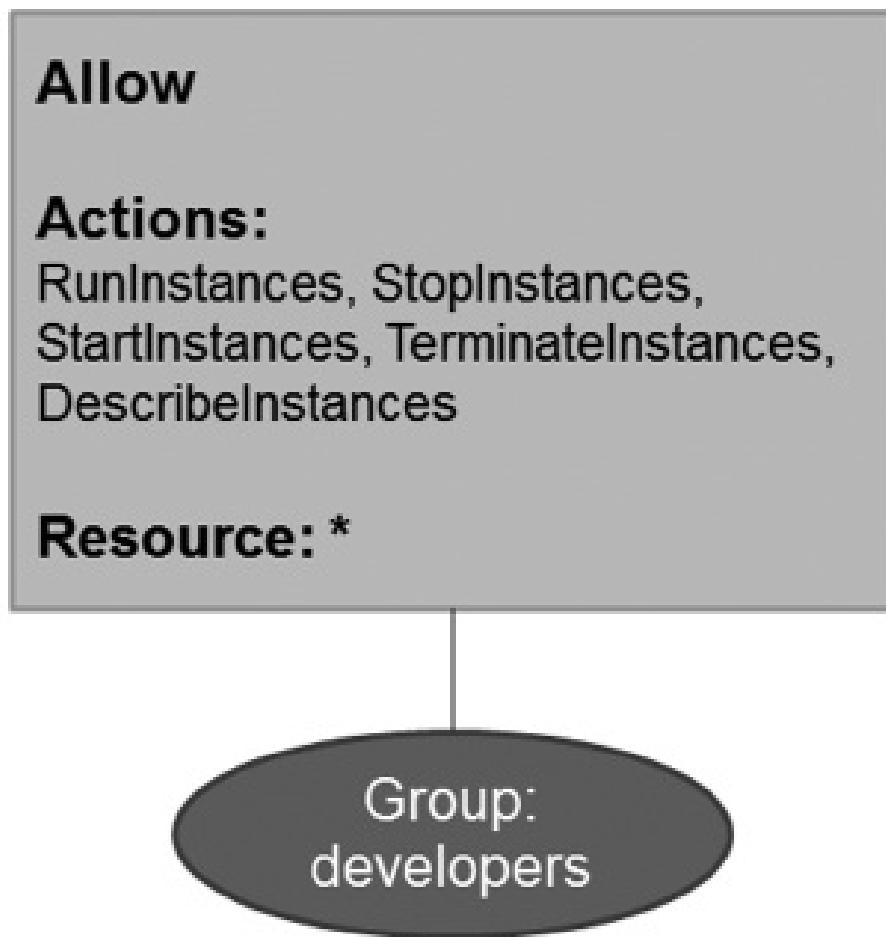


Figura 5-6 Permissões EC2 com IAM

#### 5.2.7.3. Credencias temporárias de segurança

Além de criar os usuários IAM com suas próprias credenciais de segurança, o IAM também permite que sejam concedidas credenciais de segurança temporária para qualquer usuário que não possui conta AWS, permitindo a este usuário acessar os serviços AWS e recursos. Assim, pode-se gerenciar os usuários de uma conta AWS que são usuários IAM e também usuários que não pertencem a uma conta AWS; esses são chamados de usuários federados. Na AWS, “usuários” também podem ser aplicativos criados para acessar recursos AWS. Isto foi assunto do capítulo 3.

#### 5.2.8. EC2 na prática

O guia de conceitos básicos “Amazon Elastic Compute Cloud (API Version 2012-03-01)” ensina a implementar o ciclo inteiro de uma instância EC2. Isto será mostrado aqui. Inicialmente é necessário ter uma conta na AWS.

### 5.2.8.1. Lançar uma instância

A partir do painel do console de gerenciamento AWS, clique em “Launch Instance” para iniciar o *wizard*.

Para lançar uma instância existem três opções disponíveis no console de gerenciamento AWS, mostradas na **Figura 5-7**.

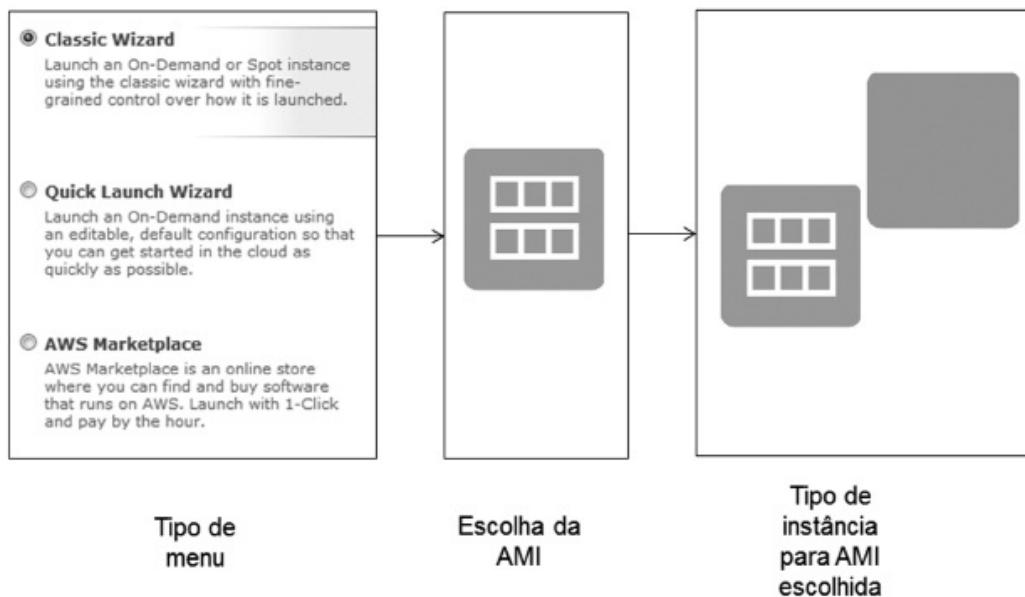


Figura 5-7 Opções para lançar instância no console de gerenciamento AWS

Na opção *Classic Wizard* será necessário definir detalhes para as instâncias, incluindo o par de chaves e o grupo de segurança. Esta opção deve ser utilizada quando se quer fazer maior detalhamento das opções de configuração para lançamento da instância.

A **Figura 5-8** ilustra a tela com a opção *Classic Wizard*.

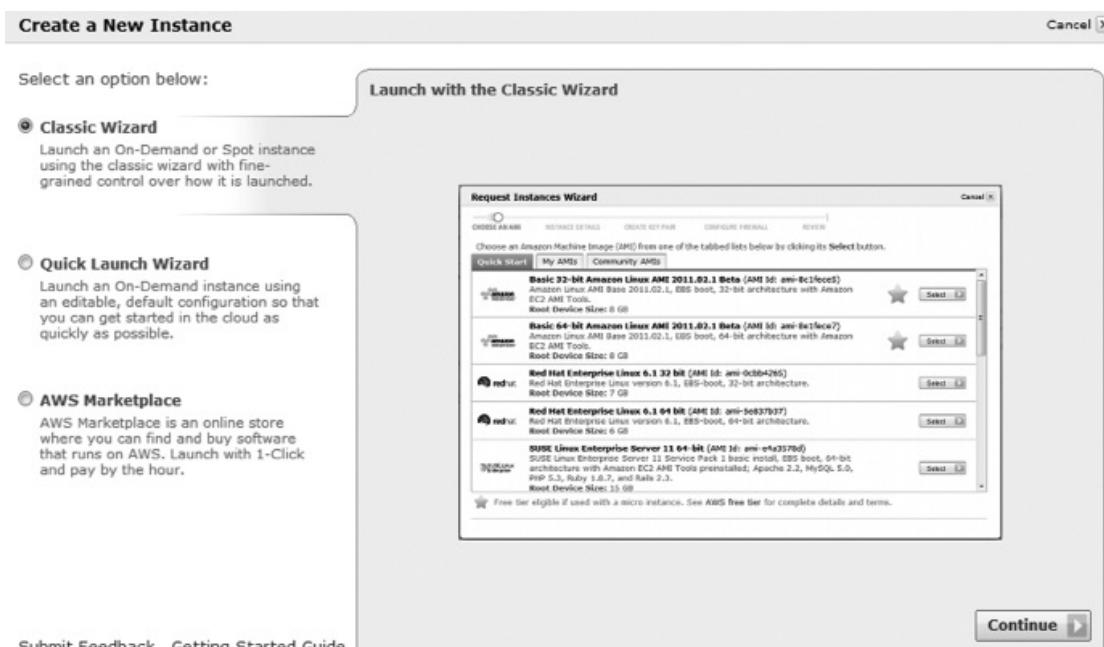


Figura 5-8 Create a new instance – *Classic Wizard*

No *Quick Launch Wizard* pode-se já na primeira tela (“Create a New Instance”) escolher o nome da instância e o par de chaves. Nesta mesma primeira tela exibe-se uma lista de AMIs, conforme ilustra a **Figura 5-9**.

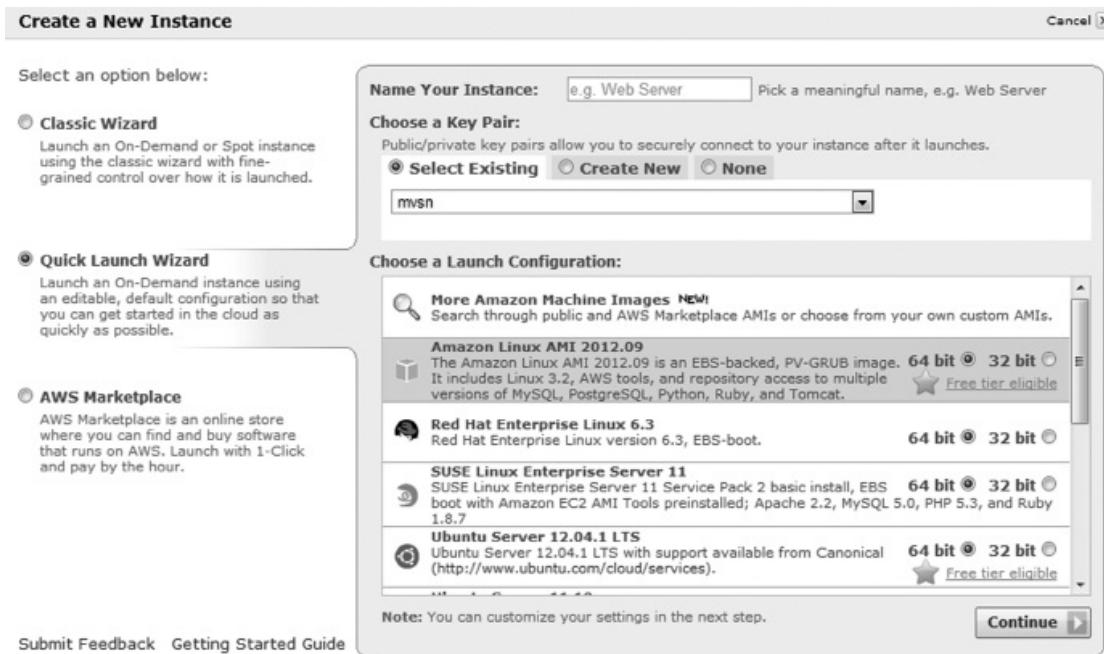


Figura 5-9 Create a new instance – *Quick Launch Wizard*

Depois de definir o nome da instância, deve-se definir o par de chaves (*key pair*), que pode ser um já existente, ou pode ser criado um novo par de chaves – ou mesmo, em um caso especial, pode-se não escolher o par de chaves (neste caso não será possível se conectar à instância de imediato quando ela for lançada).

Também é possível adquirir softwares disponibilizados no AWS Marketplace diretamente do console de gerenciamento AWS. A **Figura 5-10** ilustra esta opção.

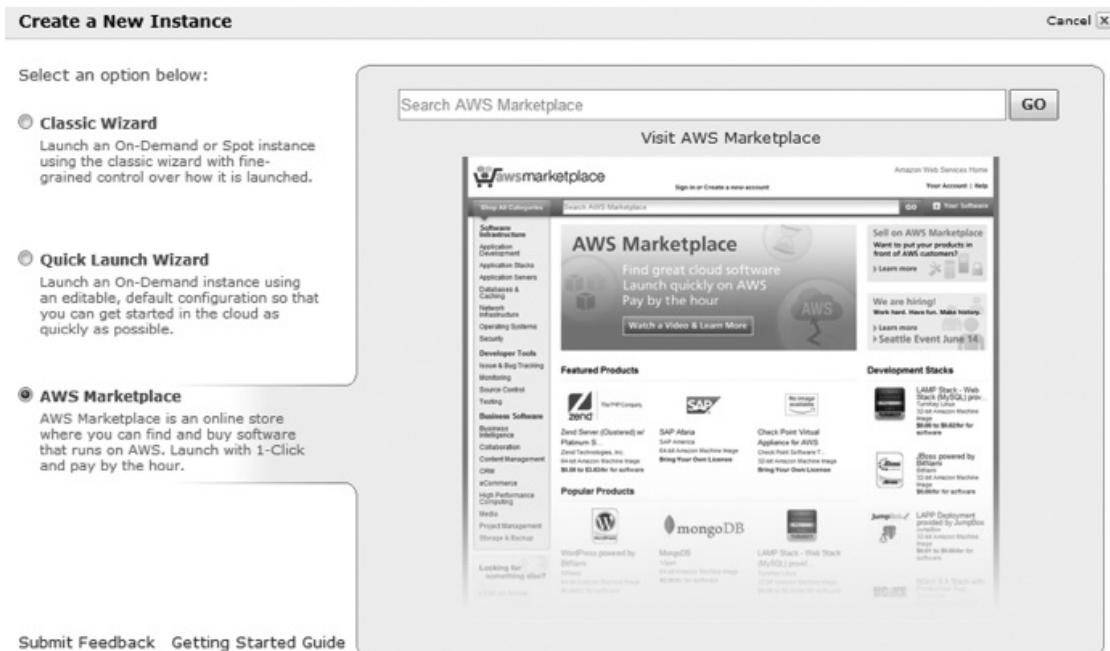


Figura 5-10 Create a new instance – AWS Marketplace

Considera-se a opção *Classic Wizard* daqui para frente.

Deve-se também escolher a AMI. Pode-se escolher uma AMI baseada no Windows, por exemplo. A **Figura 5-11** ilustra esta opção.

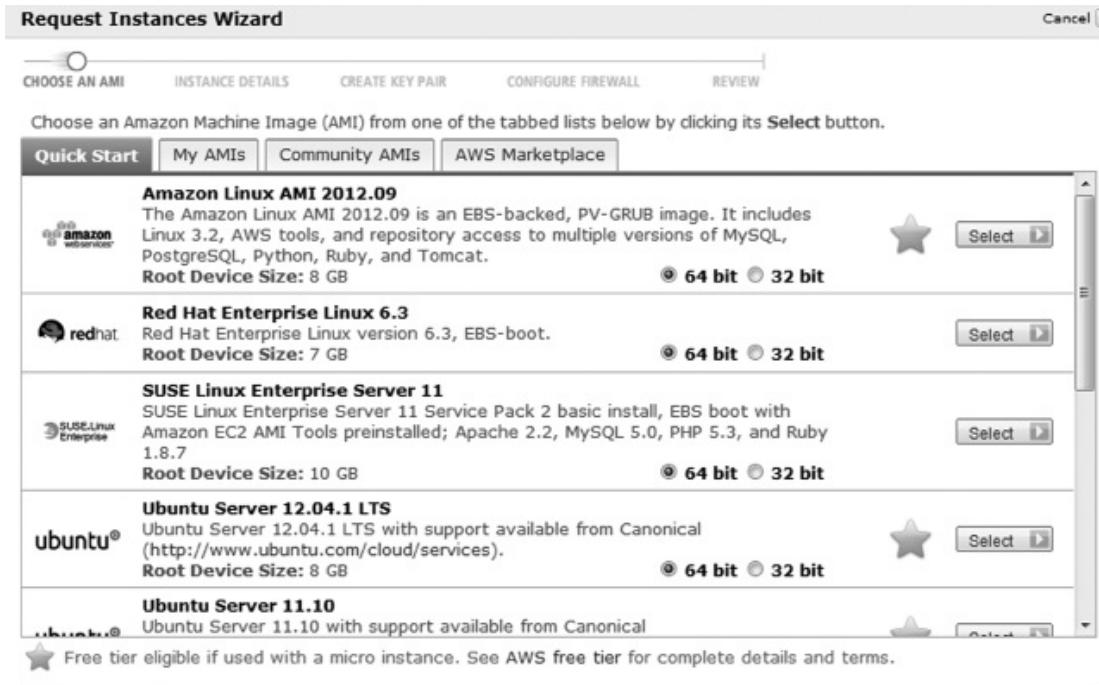


Figura 5-11 Choose an AMI

Na opção “Instance Details” pode-se escolher o número de instâncias (“Number of Instances”), o (“Instance Type”), lançar a instância no EC2 ou na rede VPC e até requisitar *Spot Instances*. A **Figura 5-12** ilustra essas opções.

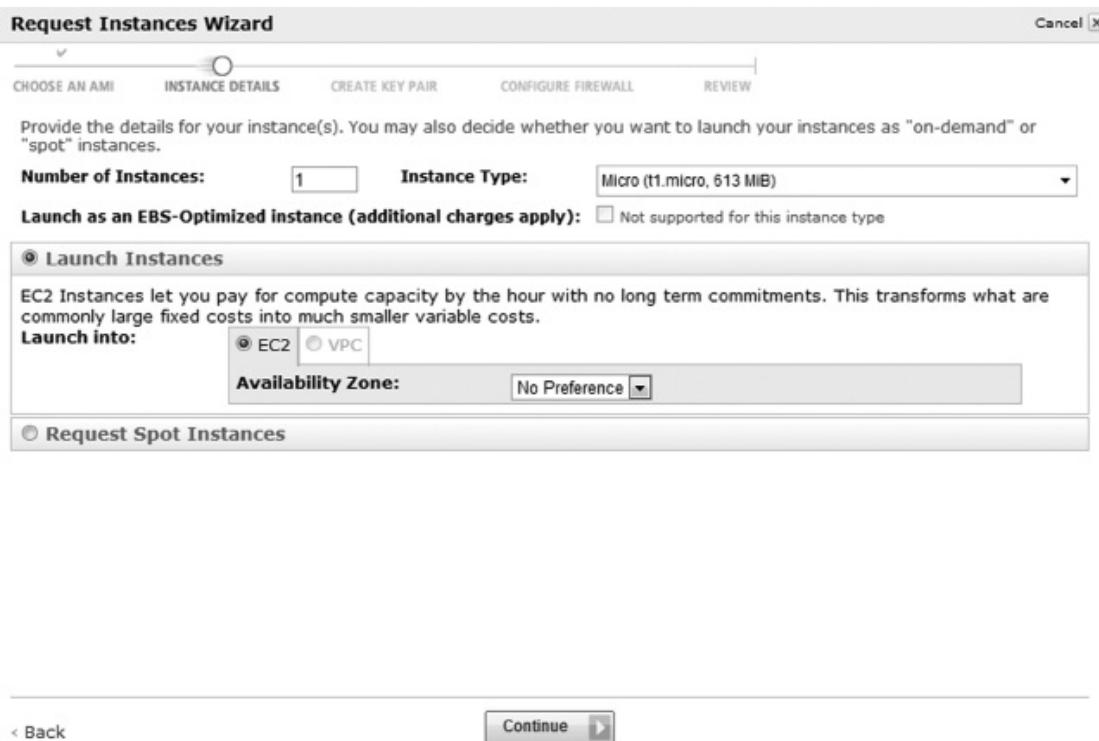


Figura 5-12 Instance details

Precisam ser feitas outras escolhas em uma próxima caixa de diálogo, incluindo a opção de utilizar kernel específico (“Kernel ID”) ou um disco RAM (“RAM Disk ID”) para utilizar com a instância. Pode-se escolher também a opção “Monitoring”, que permite o detalhamento da monitoração pelo CloudWatch. E ainda:

- **User Data:** permite definir dados de usuário que serão disponibilizados para a instância após o seu lançamento.
- **Termination Protection:** se habilitado, não permite terminar a instância via API.
- **Shutdown Behavior:** permite definir o comportamento da instância quando do *shutdown*.
- **IAM Role:** possibilita gerenciar permissões para os aplicativos que rodam no EC2.

A **Figura 5-13** ilustra a caixa de diálogo com a configuração do dispositivo de armazenamento. Esta opção será mais bem explicada no próximo capítulo.

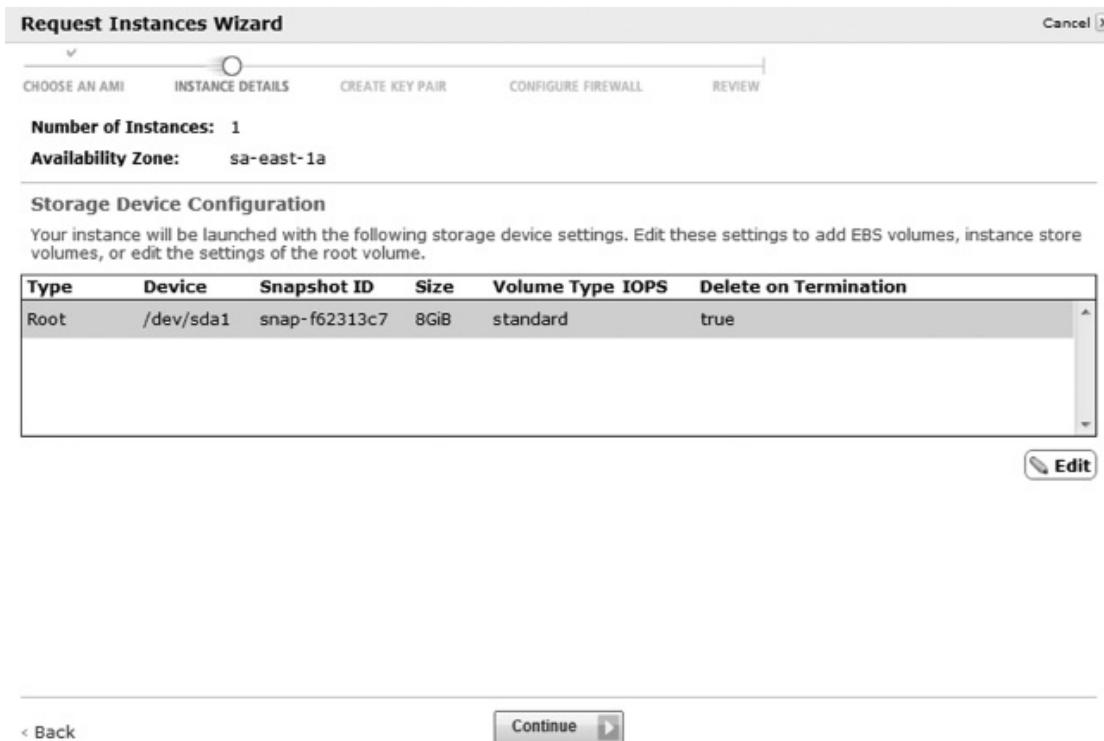


Figura 5-13 Storage device configuration

Na próxima caixa de diálogo é possível adicionar tags à instância para simplificar a administração da infraestrutura EC2.

O *wizard* permite criar o par de chaves (*key pair*) em seguida. Deve-se definir se será um par de chaves já existente ou se será criado um, ou mesmo, em um caso especial, proceder sem escolher o par de chaves (neste caso não será possível se conectar à instância).

Na opção “Create a new Key Pair” (criar um novo par de chaves) insira um nome para o par de chaves. O nome escolhido para o arquivo será associado ao par.

A **Figura 5-14** ilustra a caixa de diálogo para a criação do novo par de chaves.



Figura 5-14 Create a new key pair

Ainda na opção *Classic Wizard*, deve-se nomear um grupo de segurança

(*firewall*) onde estão definidas as regras de *firewall* para as instâncias. Essas regras definem o tráfego de entrada para a instância e podem ser modificadas a qualquer momento.

Neste ponto o *wizard* sugere escolher um ou mais grupos de segurança (“Choose one or more of your existing Security Groups”) ou criar um novo grupo de segurança (“Create a New Security Group”).

A **Figura 5-15** ilustra a opção de criar um novo grupo de segurança (“Create a new Security Group”). Observe que existe a opção “Inbound Rules”, que permite definir as regras de *firewall* para a instância em questão. As opções são:

- “Create a new rule” (criar nova regra).
- “Port range” (faixa das portas).
- “Source” (fonte).

Ao final deve-se adicionar a regra criada.

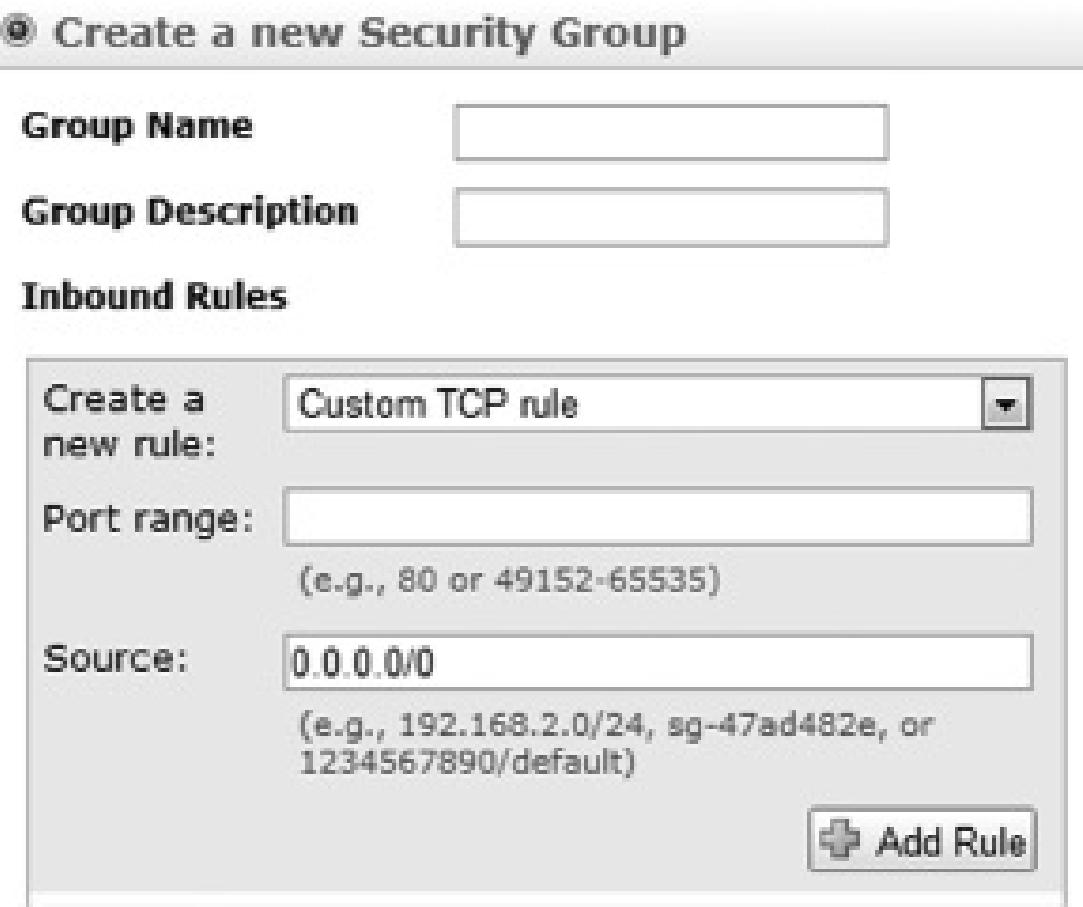


Figura 5-15 Create a new Security Group

A **Figura 5-16** ilustra a opção de revisão (“Review”) antes do lançamento da instância Windows.

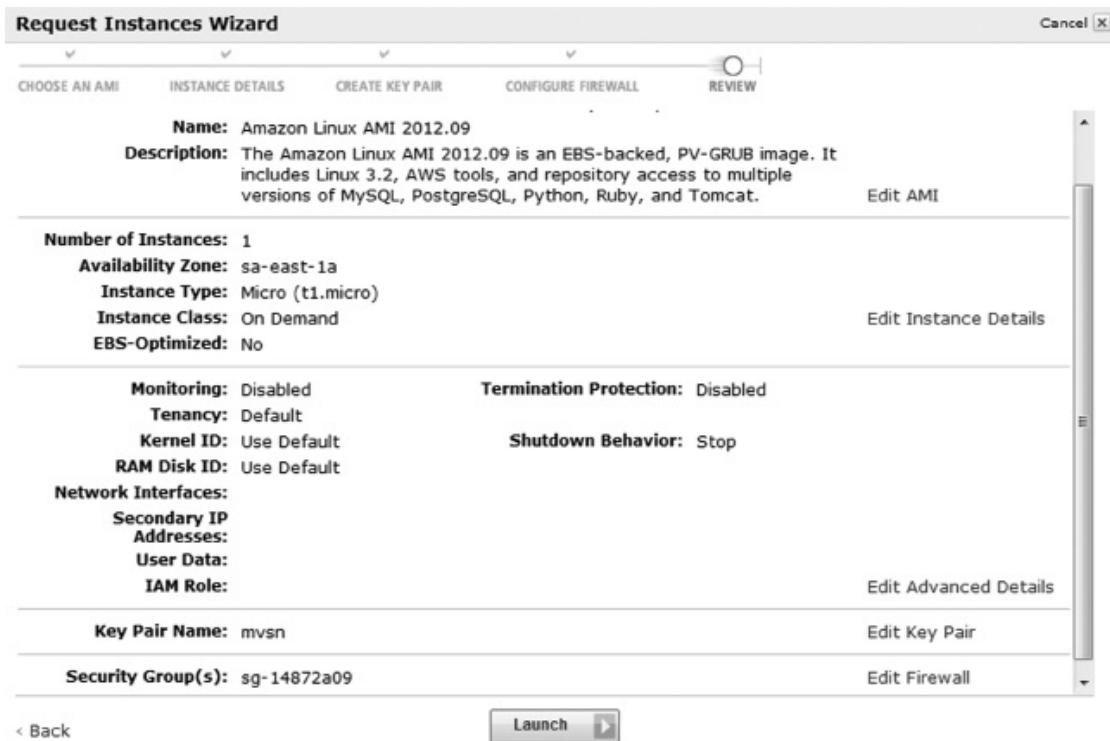


Figura 5-16 Request instances – review

Quando se clica com o mouse em “Launch” (lançar) inicia-se o processo de lançamento da instância. Como última opção, é possível realizar ainda algumas tarefas enquanto a instância está sendo lançada.

A **Figura 5-17** ilustra esta opção.

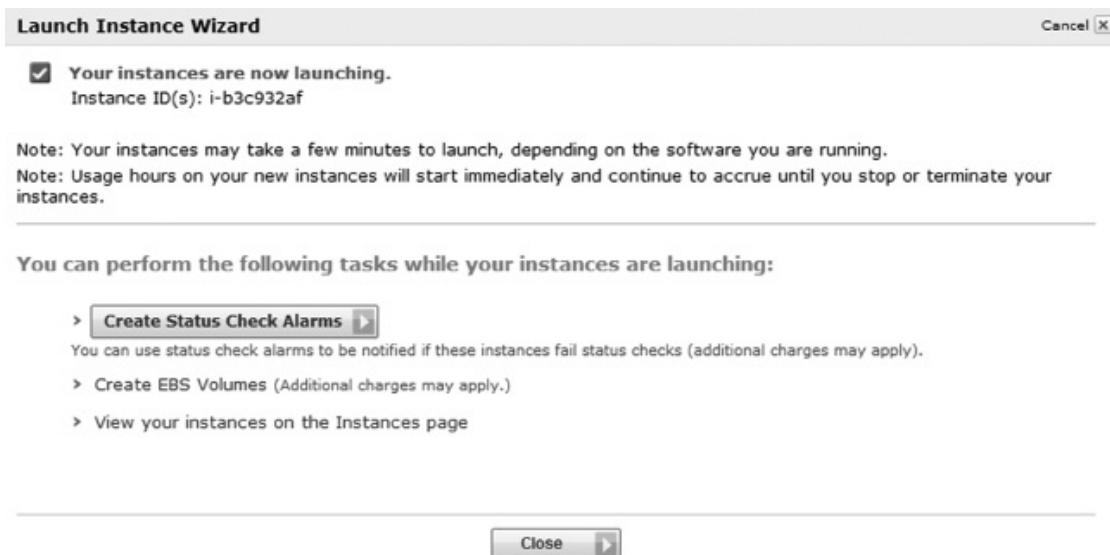


Figura 5-17 Create a new instance

Clique em “Close” para fechar a página de confirmação e em seguida clique em “instances” (instâncias) no painel de navegação para exibir o status da instância.

O status será “pending” (pendente) enquanto a instância estiver iniciando. Após um curto período de tempo o status da instância muda para

"running" (executando).

É importante registrar o nome do DNS público para a instância criada. A seleção da instância permite exibir suas propriedades no painel inferior, incluindo o DNS público.

### 5.2.8.2. Conexão a uma instância Linux/UNIX

Para se conectar a uma instância EC2 Linux/UNIX de uma estação Linux/UNIX ou Windows pode-se utilizar o cliente SSH.

Estações Linux/UNIX incluem um cliente SSH por padrão. Se a sua estação não inclui, o projeto OpenSSH oferece implementação grátis da suíte completa de ferramentas SSH. Informações em <http://www.openssh.org>.

Estações Windows, por exemplo, podem utilizar PuTTY, um cliente SSH gratuito. No caso de uso do PuTTY, é necessário converter as chaves privadas geradas pelo EC2 para o formato PuTTY. A ferramenta PuTTYgen permite a conversão. Para baixar e instalar o PuTTY vá em <http://www.chiark.greenend.org.uk/~sgtatham/putty/>.

Recentemente o cliente SSH MindTerm foi integrado no console de gerenciamento AWS para simplificar o processo de se conectar a uma instância EC2 Linux/UNIX. Isso facilita a vida de quem usa uma estação Windows que não tem um terminal/SSH default. No caso da estação Windows, a utilização do cliente MindTerm (Java) torna desnecessária a utilização do PuTTY.

### 5.2.8.3. Conexão a uma instância Windows

Para se conectar a uma instância Windows, deve-se recuperar a senha do administrador inicial e em seguida usá-la na área de trabalho remota. Será necessário obter o conteúdo do arquivo de chave privada criado quando da iniciação da instância.

Para conectar-se a uma Instância Windows:

1. Recupere a senha inicial de administrador.

- Navegue até o diretório onde você armazenou o arquivo de chave privada quando iniciou a instância.
- Abra o arquivo em um editor de texto e copie todo o conteúdo (incluindo as primeiras e últimas linhas, que contêm BEGIN RSA PRIVATE KEY e END RSA PRIVATE KEY).
- Vá para o console de gerenciamento AWS e localize a instância na página "Instâncias".
- Clique com o botão direito do mouse na instância e selecione "Obter senha do Windows". A caixa de diálogo "Recuperar senha padrão de administrador do Windows" é exibida (pode demorar alguns minutos

depois de a instância ser iniciada até que a senha esteja disponível).

- Cole o conteúdo do arquivo de chave privada no campo “Chave privada”.
- Clique em “Descriptografar senha”. O console retorna a senha de administrador padrão para a instância.
- Salve a senha. Você vai precisar dela para se conectar à instância.

2. Conecte-se a uma instância usando a área de trabalho remota utilizando o nome DNS público e efetuando o login usando “administrator” como nome de usuário e a senha do administrador obtida no item anterior.

É importante ressaltar ainda que se a instância for lançada, ela será cobrada para cada hora ou hora parcial que for mantida em execução, mesmo se ela estiver ociosa. Se a instância não é mais necessária pode-se encerrá-la.

#### **5.2.8.4. Encerrar uma instância**

Para encerrar uma instância, deve-se localizá-la na opção “Instances” na guia EC2 do console de gerenciamento AWS e clicar com o botão direito do mouse escolhendo “Terminate” (encerrar). Clique em “Yes” quando solicitada a confirmação. Quando o status mudar para “shutting down” ou para “terminated” a cobrança cessará.

#### **5.2.9. Importante**

- Volumes de armazenamento de instância local são fixos em tamanho para um dado tipo de instância EC2 e vinculados a uma instância específica, de modo que este tipo de armazenamento é inelástico.
- É possível adquirir até vinte instâncias reservadas por zona de disponibilidade por mês. Se for necessário utilizar mais instâncias reservadas, deve-se preencher um formulário fornecido pela AWS.
- É possível solicitar apenas uma ou até cem instâncias *spot* por meio do console de gerenciamento ou pela API do EC2. Se for necessário solicitar mais de cem instâncias *spot*, deve-se preencher o formulário de solicitação de instâncias do EC2.
- Atualmente, as instâncias reservadas não podem ser usadas com instâncias em execução no Red Hat Enterprise Linux, em IBM e em AMI paga.
- Instâncias *spot* podem utilizar AMIs com o *root device* em EBS, mas não suportam operações Stop/Start.
- Não é possível utilizar uma instância *spot* com uma AMI paga para softwares de terceiros (como pacotes de software da IBM).

### **5.3. Elastic Load Balancing (ELB)**

### 5.3.1. Introdução

O *Elastic Load Balancing* (ELB) distribui automaticamente o tráfego de entrada dos aplicativos em várias instâncias EC2. Ele permite melhorar a tolerância a falhas do ambiente e ainda fornece capacidade de carga em resposta ao tráfego de entrada dos aplicativos. Quando se define um nome único para o ELB, quaisquer solicitações enviadas a esse nome de *host* são delegadas a um pool de instâncias EC2.

O ELB também detecta instâncias com problema dentro de um conjunto e redireciona automaticamente o tráfego para instâncias saudáveis até que as instâncias com problema sejam restauradas. Ele é disponibilizado por região e não por zona de disponibilidade.

### 5.3.2. Arquiteturas ELB

Pode-se habilitar o ELB para atuar com instâncias EC2 em uma única zona de disponibilidade ou para atuar com instâncias EC2 disponibilizadas em várias zonas de disponibilidade. A **Figura 5-18** ilustra uma arquitetura padrão para o ELB e instâncias web EC2. Os usuários acessam a arquitetura via serviço de DNS.

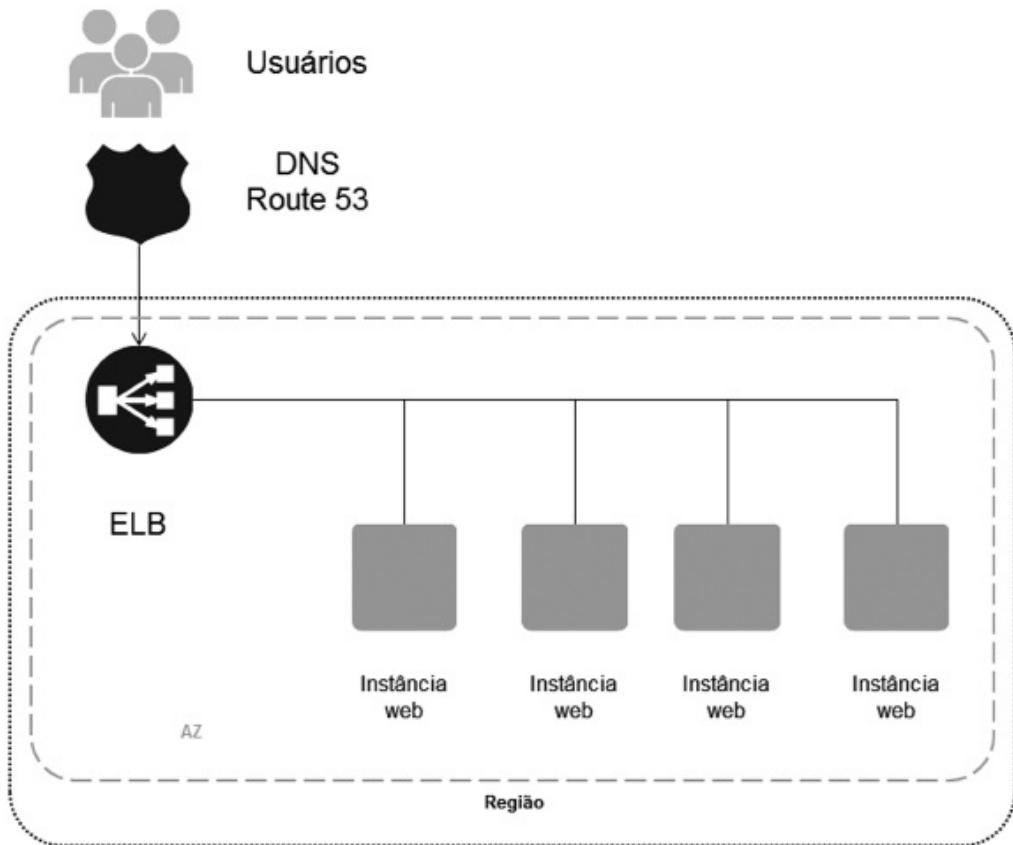


Figura 5-18 ELB

Outra opção é estabelecer as instâncias EC2 para o ELB em duas ou mais zonas de disponibilidade distintas, aumentando a disponibilidade do ambiente.

A **Figura 5-19** ilustra esta possibilidade. Quando se chama a função `EnableAvailabilityZonesForLoadBalancer` o balanceador de carga começa a encaminhar o tráfego igualmente entre as zonas de disponibilidade habilitadas. A Amazon alerta que se as instâncias não foram registradas anteriormente, as solicitações que envolvem as novas instâncias não vão funcionar.

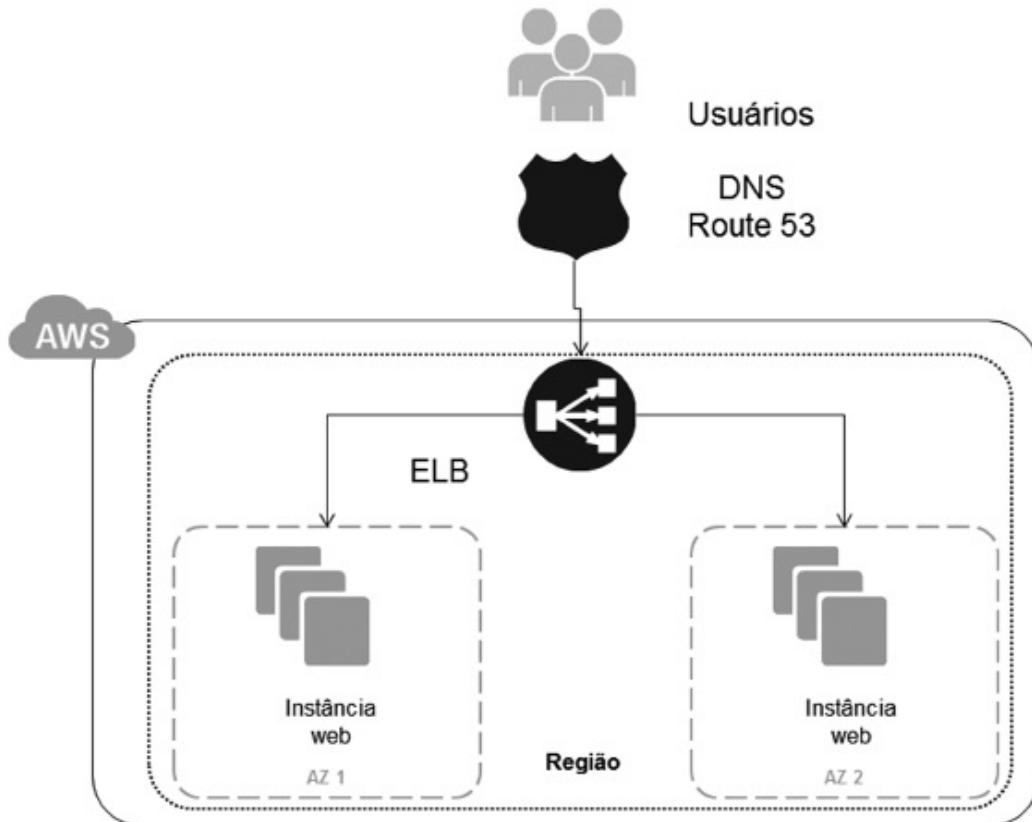


Figura 5-19 ELB com instâncias em diferentes AZs

O ELB oferece suporte a instâncias EC2 baseadas em qualquer sistema operacional compatível com o serviço do EC2.

Pode-se também criar um *Load Balancing* na AWS sem utilizar os recursos ELB. Em geral esta opção, quando comparada à opção que utiliza o ELB da AWS, será mais cara e apresentará um ponto único de falha (*Single Point Of Failure* – SPOF). Também neste caso a elasticidade terá que ser implementada manualmente. Deve-se ressaltar que o uso do ELB está vinculado a um tráfego oriundo da internet, o que pode ser considerado uma limitação de uso. A **Figura 5-20** ilustra esta possibilidade.

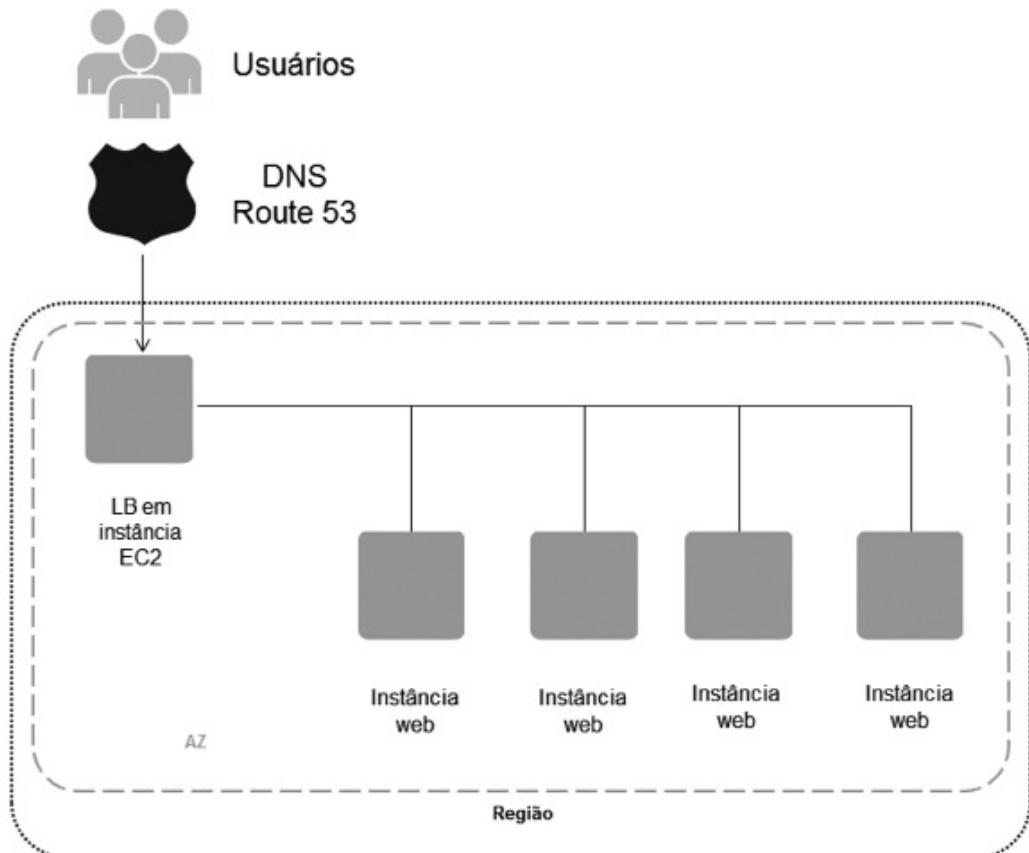


Figura 5-20 Outra forma de construir o *load balancing*

O *Auto Scaling* pode ser utilizado em conjunto com o ELB para ajustar o número de instâncias necessárias para o pleno funcionamento do ELB. O gerenciamento dos clusters ELB com o CloudWatch/*Auto Scaling* será visto no capítulo 10.

### 5.3.3. ELB na prática

#### 5.3.3.1. Criar um ELB

- Selecione *Load Balancers* na coluna da esquerda do console de gerenciamento.
- No painel superior direito, selecione “Create Load Balancer”.
- Crie um nome exclusivo para o *Load Balancer*.

O ELB oferece suporte ao balanceamento de carga de aplicativos usando os protocolos HTTP, HTTPS, SSL e TCP (opção *Load Balancer Protocol*). O ELB também pode ser utilizado dentro de uma VPC (opção “VPC” em “Create LB Inside”).

A **Figura 5-21** ilustra a tela inicial do *wizard* (disponível a partir da opção “Network & Security” no console de gerenciamento EC2). É necessário possuir uma conta na AWS.

Create a New Load Balancer

Cancel X

DEFINE LOAD BALANCER    CONFIGURE HEALTH CHECK    ADD EC2 INSTANCES    REVIEW

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

**Load Balancer Name:** e.g. MyLoadBalancer

**Create LB inside:** EC2

**Create an internal load balancer:**  (what's this?)

**Listener Configuration:**

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	Actions
HTTP	80	HTTP	80	<input type="button" value="Remove"/>
HTTP	<input type="text"/>	HTTP	<input type="text"/>	<input type="button" value="Save"/>
HTTPS (Secure HTTP)				
TCP				
SSL (Secure TCP)				



Figura 5-21 Define Load Balancer

O passo seguinte (“Configure Health Check”) permite configurar as verificações de integridade para o *Load Balancer* e somente rotear o tráfego para instâncias que passam na verificação de integridade. As instâncias que falham no *health check* são automaticamente removidas do *Load Balancer*. O *health check* pode ser customizado nesta tela. A opção “Listener Configuration” exige que os servidores HTTP EC2 estejam ativos e aceitem *requests* na porta 80. Pode-se também criar um LB interno, ou seja, dentro de uma VPC. A Figura 5-22 ilustra as opções.

Create a New Load Balancer

Cancel X

DEFINE LOAD BALANCER    CONFIGURE HEALTH CHECK    ADD EC2 INSTANCES    REVIEW

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

**Configuration Options:**

**Ping Protocol:**

**Ping Port:**

**Ping Path:**

**Advanced Options:**

<b>Response Timeout:</b> <input type="text" value="5"/> Seconds	Time to wait when receiving a response from the health check (2 sec - 60 sec).
<b>Health Check Interval:</b> <input type="text" value="0.5"/> Minutes	Amount of time between health checks (0.1 min - 5 min)
<b>Unhealthy Threshold:</b> 	Number of consecutive health check failures before declaring an EC2 instance unhealthy.
<b>Healthy Threshold:</b> 	Number of consecutive health check successes before declaring an EC2 instance healthy.



Figura 5-22 Configure Health Check

As opções avançadas (“Advanced Options”) servem para configurar parâmetros específicos de *health check*. São eles:

- “Response Timeout” vai de 2 a 60 segundos.
- “Health Check Interval” vai de 0,1 minuto a 5 minutos.
- “Unhealthy Threshold” trata do número de falhas antes de declarar que a instância não está saudável.
- “Healthy Threshold” trata do número de sucessos antes de declarar que a instância está saudável.

A opção “Add EC2 Instances” mostra as opções de instâncias EC2 que ainda não são utilizadas por outros *Load Balancers* ou fazem parte de um grupo *Auto Scaling*. É necessário marcar os boxes para adicionar instâncias EC2 que irão fazer parte do *Load Balancer*. A **Figura 5-23** ilustra esta opção. Observe que as duas instâncias marcadas estão em zonas de disponibilidade diferentes.

**Create a New Load Balancer**

Cancel

Define Load Balancer Configure Health Check ADD EC2 INSTANCES REVIEW

The table below lists all your running EC2 Instances that are not already behind another load balancer or part of an auto-scaling capacity group. Check the boxes in the Select column to add those instances to this load balancer.

**Manually Add Instances to Load Balancer:**

Select	Instance	Name	State	Security Groups	Availability Zone
<input type="checkbox"/>	i-7c013361		<input checked="" type="radio"/> stopped	quick-start-1	sa-east-1a
<input type="checkbox"/>	i-68383b75		<input checked="" type="radio"/> stopped	quick-start-1	sa-east-1b
<input type="checkbox"/>	i-6eb4b973	Teste	<input checked="" type="radio"/> stopped	quick-start-1	sa-east-1b
<input checked="" type="checkbox"/>	i-b3c932af		<input checked="" type="radio"/> running	ipoip	sa-east-1a
<input checked="" type="checkbox"/>	i-2d778d31		<input checked="" type="radio"/> running	quick-start-1	sa-east-1a

[select all](#) | [select none](#)

**Availability Zone Distribution:**

2 instances in sa-east-1a  
0 instances in sa-east-1b

[< Back](#)

Figura 5-23 Add EC2 instances

O passo “Review” permite então criar o *Load Balancer*. Observe que as várias opções escolhidas podem ser mudadas via opção “Edit”. A **Figura 5-24** ilustra esta opção.

Create a New Load Balancer Cancel

---

▼ DEFINE LOAD BALANCER    ▼ CONFIGURE HEALTH CHECK    ▼ ADD EC2 INSTANCES    REVIEW

---

**DEFINE LOAD BALANCER**

**Load Balancer Name:** MyLoadBalancer  
**Scheme:** Internet-facing  
**Port Configuration:** 80 (HTTP) forwarding to 80 (HTTP)

[Edit Load Balancer Definition](#)

---

**CONFIGURE HEALTH CHECK**

**Ping Target:** HTTP:80:/index.html  
**Timeout:** 5  
**Interval:** 0.5

**Unhealthy Threshold:** 2  
**Healthy Threshold:** 10

[Edit Health Check](#)

---

**ADD EC2 INSTANCES**

**EC2 Instances:** i-b3c932af, i-2d778d31

[Edit EC2 Instance Selection](#)

---

**VPC INFORMATION**

**VPC:**  
**Subnets:**

Please review your selections on this page.  
 Clicking "Create" will launch your load balancer.  
 Check the Amazon EC2 product page for load balancer pricing info

< Back

**Create**

Figura 5-24 Create a new load balancer – review

Agora é só criar o ELB via opção “Create”. A caixa de diálogo *Load Balancers* abre após o clique na opção “Close”. A **Figura 5-25** ilustra a opção LB criada e o DNS name.

Load Balancer Name	DNS Name	Port Configuration
<input type="checkbox"/> TESTE	TESTE-1119757259.sa-east-1.elb.amazonaws.com	80 (HTTP) forwarding to 80 (HTTP)
<input checked="" type="checkbox"/> MyLoadBalancer	MyLoadBalancer-1105641722.sa-east-1.elb.amazonaws.com	80 (HTTP) forwarding to 80 (HTTP)

---

**DNS Name:**  
 MyLoadBalancer-1105641722.sa-east-1.elb.amazonaws.com (A Record)

Note: Because the set of IP addresses associated with a LoadBalancer can change over time, you should never create an “A” record with any specific IP address. If you want to use a friendly DNS name for your LoadBalancer instead of the name generated by the Elastic Load Balancing service, you should create a CNAME record for the LoadBalancer DNS name, or use Amazon Route 53 to create a hosted zone. For more information, see the Using Domain Names With Elastic Load Balancing

---

**Scheme:** internet-facing

---

**Status:** 0 of 2 instances in service

---

**Port Configuration:** 80 (HTTP) forwarding to 80 (HTTP)  
 Stickiness: Disabled [\(edit\)](#)

---

**Availability Zones:** sa-east-1a

---

**Source Security Group:** amazon-elb/amazon-elb-sg  
 Owner Alias: amazon-elb  
 Group Name: amazon-elb-sg

---

**Hosted Zone ID:** Z2ES78Y61JGQKS

Figura 5-25 Load balancer name e DNS name

O tráfego para o nome DNS fornecido pelo *Elastic Load Balancer* é automaticamente distribuído entre as instâncias íntegras do EC2 e com equilíbrio de carga.

O nome DNS permite testar o ELB copiando o endereço DNS no campo de endereço de um web browser.

Para usar um nome de subdomínio que direciona o tráfego para o平衡ador de carga, crie um registro CNAME que associa o nome do subdomínio com o balanceador de carga. A vantagem deste método é que a criação de registros CNAME pode ser um processo simples. A desvantagem deste método é que não se pode usar um registro CNAME para associar uma instância de balanceamento de carga em uma zona apex (por exemplo, manoel.com). Para criar um *alias* da zona apex que aponta para a instância do serviço de balanceamento de carga elástico, use o Route 53.

Todos os passos mencionados também estão disponíveis como APIs de ELB e como operações de linha de comando.

### 5.3.3.2. Deletar um ELB

Na página “Load Balancers”, selecione a caixa de seleção ao lado do balanceador de carga que deseja apagar e, em seguida, clique em “Excluir”.

### 5.3.4. Importante

- É possível mapear a porta 80 HTTP e a porta 443 HTTPS para um único ELB.
- O ELB oferece suporte à terminação SSL no balanceador de carga.
- O ELB facilita a criação de um ponto de entrada de internet na VPC. Podem ser atribuídos grupos de segurança ao ELB para controlar quais portas estarão abertas para uma lista de fontes permitidas.
- Cada ELB tem um nome DNS IPv4, IPv6 e *dualstack* (IPv4 e IPv6) associado. O IPv6 não é suportado quando o ELB está na VPC.

## 5.4. Elastic MapReduce (EMR)

### 5.4.1. Conceito

O *Elastic MapReduce* (EMR) é um web service que permite realizar o processamento de grandes quantidades de dados. Ele utiliza uma estrutura *Hadoop* sendo executada na infraestrutura do EC2 e do S3. O S3, o serviço de armazenamento AWS, será visto no capítulo 6.

O *Hadoop* é uma plataforma de software de computação distribuída voltada para clusters e processamento de grandes massas de dados. Foi inspirado pelo *MapReduce* e *GoogleFS* (GFS). Trata-se de um projeto de alto nível da Apache que está sendo construído por uma comunidade de contribuidores utilizando a linguagem de programação Java.

O EMR é um modelo de programação criado para processar grandes volumes de dados em paralelo, dividindo o trabalho em um conjunto de tarefas independentes. Trata-se de um estilo de programação paralela que é suportado por algumas nuvens de capacidade *on demand* como o

## *Hadoop.*

Usando o EMR é possível fornecer imediatamente quanta capacidade desejar para melhorar o desempenho de tarefas que requerem muitos dados para aplicativos como indexação web, mineração de dados, análise de arquivos de log, depósito de dados, análise financeira e simulação científica.

O EMR permite que o usuário se concentre na análise dos dados sem ter que se preocupar com a configuração, a gestão ou a sintonia de clusters *Hadoop*, o que consome tempo, ou com a capacidade computacional na qual estão baseados.

Na essência, o EMR realiza de forma automática uma implementação *Hadoop* do framework do *MapReduce* nas instâncias do EC2, subdividindo os dados de um fluxo de trabalho em partes menores para que a função mapa possa ser processada paralelamente e, por fim, recombinando os dados processados na solução final (a função redutor). O S3 atua como fonte dos dados que estão sendo analisados, logs e scripts e como destino de saída dos resultados finais.

### **5.4.2. Utilização do EMR**

A utilização do EMR pode ser feita obedecendo aos seguintes passos:

- Desenvolva seu aplicativo de processamento de dados. O EMR permite que fluxos de trabalho sejam desenvolvidos por linguagens do tipo SQL, facilitando a gravação de scripts analíticos de dados sem o conhecimento aprofundado do paradigma de desenvolvimento do *MapReduce*.
- Faça o upload dos dados no *bucket* do S3.
- Faça login no console de gerenciamento AWS para iniciar um “fluxo de trabalho” do EMR: basta selecionar o número e o tipo de instâncias do EC2 desejadas, especificar a localização dos dados e/ou aplicativos no S3 e, em seguida, clicar no botão “Create a Job Flow” (criar fluxo de trabalho).
- Monitore o progresso do seu fluxo de trabalho diretamente no console de gerenciamento AWS, através das ferramentas de linha de comando ou das APIs. Por fim, após o fluxo de trabalho ser concluído, recupere a saída pelo S3.
- Pague somente pelos recursos utilizados. O EMR monitora o fluxo de trabalho e, a não ser que seja especificado o contrário, desliga as instâncias do EC2 após o trabalho ser concluído.

### **5.4.3. BIG DATA**

BIG DATA refere-se a conjuntos de dados (*data sets*) que são muito grandes para serem processados por bancos de dados relacionais tradicionais

e são inefficientes para serem analisados por aplicações não distribuídas. Parte desses conjuntos de dados possui grandes volumes, grande variedade e grande velocidade e muitas vezes é pouco estruturada. Dados recentes publicados pelo IDC dão conta de que o mercado de BIG DATA já é de 26 bilhões de dólares em todo o mundo.

Aplicações focadas em tratar o BIG DATA devem permitir processar e analisar rapidamente um grande número de informações.

Um caminho popular para analisar grandes *data sets* é utilizar cluster de servidores rodando em paralelo. Cada servidor processa uma porção dos dados e depois os resultados são agregados. Uma técnica que utiliza esta estratégia é o *MapReduce*, já explicado no item anterior. Como visto, o *Hadoop* é uma implementação de software livre para o *MapReduce* que suporta o processamento distribuído de grandes *data sets*. Esta seria uma das formas de tratar o BIG DATA na AWS. A ideia é ilustrada na **Figura 5-26**.

A AWS lançou recentemente o AWS Data Pipeline, que permite automatizar o movimento e processamento de qualquer quantidade de dados utilizando workflows. Cada pipeline é composto de fontes de dados, pré-condições, destinos de dados e tarefas de processamento. As tarefas de processamento podem ser executadas em clusters EMR.

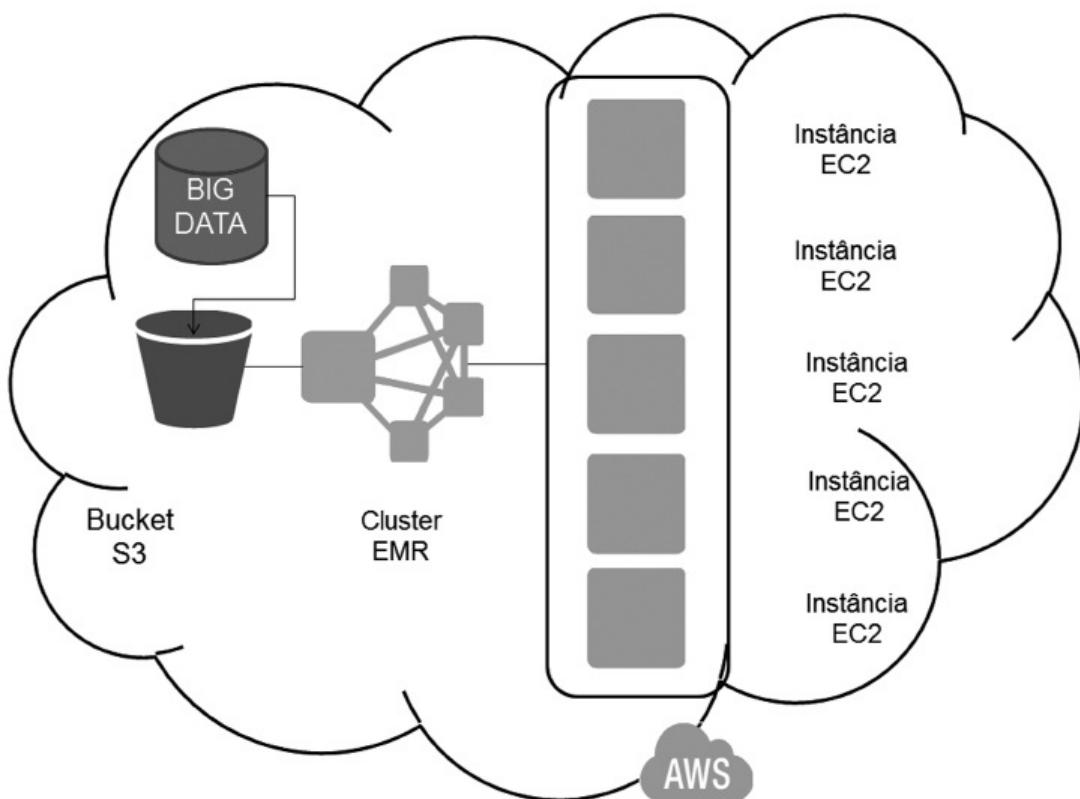


Figura 5-26 BIGDATA com AWS

#### 5.4.4. Importante

- O EMR funciona com qualquer tipo de instância Linux/UNIX EC2. Ele

permite utilizar instâncias *on demand* e reservadas; se houver instâncias reservadas, elas serão usadas em primeiro lugar pelos fluxos de trabalho.

- Se for necessário executar fluxos de trabalho com mais de vinte instâncias no EMR, é necessário preencher um formulário de solicitação de instância fornecido pela AWS.

## 5.5. Referências bibliográficas

Amazon Elastic Compute Cloud CLI Reference (API Version 2012-07-20).

Amazon Elastic Compute Cloud API Reference (API Version 2012-07-20).

Amazon Web Services. **Amazon Elastic Compute Cloud: Getting Started Guide**, API Version 2011-07-15.

Amazon Web Services. **Amazon Elastic Compute Cloud User Guide**. API Version 2012-04-01.

Amazon Web Services. **An Introduction to Spot Instances**. API version 2011-05-01.

Amazon Web Services. **Elastic Load Balancing Developer Guide**. API version 2011-11-15.

Amazon Web Services. **Elastic Load Balancing: Getting Started Guide**. API version 2011-11-15 .

Amazon Web Services. **Amazon Elastic MapReduce Developer Guide**. API Version 2009-11-30.

Amazon Web Services. **Overview of Amazon Web Services**. December 2010.

<http://aws.amazon.com/pt/ec2/>

<http://aws.amazon.com/pt/ec2/faqs/>

<http://aws.amazon.com/pt/elasticloadbalancing/>

<http://aws.amazon.com/pt/elasticmapreduce/>

# 6. Armazenamento

## 6.1. Introdução

O armazenamento de dados, quer seja na forma tradicional, quer seja na nuvem, é um aspecto crucial em qualquer projeto de infraestrutura de TI.

Segundo estudo realizado pelo IDC[7], a disponibilização de meios de armazenamento não acompanha a velocidade de criação da informação, e a tendência desta lacuna é aumentar. Pode-se concluir então que esta é uma questão relevante, e a opção por armazenamento na nuvem que permita melhorar a escalabilidade é uma saída.

Outro aspecto importante é que os dados que originam as informações podem ser estruturados ou não. Os dados estruturados são provenientes de sistemas de bancos de dados e são mais simples de serem manipulados e recuperados. Os dados não estruturados, gerados por redes sociais e por mecanismos de busca, por exemplo, por sua vez, demandam maior espaço de armazenamento e maior esforço de gerenciamento, e são justamente esses dados que fazem crescer as necessidades de armazenamento. Novas soluções precisam contemplar este aspecto.

Neste capítulo tratamos dos serviços de armazenamento na AWS, incluindo o web service Glacier, uma opção para fazer arquivamento de dados (*archive*).

## 6.2. Armazenamento tradicional *versus* armazenamento na AWS

Existem diversas formas de fazer o armazenamento de dados em uma infraestrutura tradicional de DATACENTER. As formas mais conhecidas são:

- **DAS:** trata-se do armazenamento em unidades de disco rígido locais ou *arrays* que residem em cada servidor. A durabilidade para este tipo de conexão é menor do que em uma rede SAN.
- **SAN:** trata-se do armazenamento em unidades lógicas (blocos) baseadas em LUNs (*Logical Unit Number*). O LUN é um identificador exclusivo usado em um barramento SCSI que permite diferenciar até oito dispositivos separados (cada um dos quais é uma unidade lógica). SANs dedicadas fornecem muitas vezes um bom nível de desempenho e durabilidade tanto para o armazenamento de dados quanto para o banco de dados de arquivos essenciais para os negócios, mas são sistemas ainda caros. Em relação ao DAS, a SAN

permite melhor utilização dos discos e gerenciabilidade do ambiente de armazenamento, mas normalmente tem pior desempenho, pois o armazenamento é compartilhado.

- **NAS:** fornece uma interface no nível de arquivo para armazenamento que pode ser compartilhado por vários sistemas operacionais através de uma rede de área local (*Local Area Network – LAN*). O NAS tende a ser mais lento que as opções SAN ou DAS, mas é a principal solução para armazenamento de dados em forma de arquivos.

A **Figura 6-1** resume as opções para armazenamento tradicional.

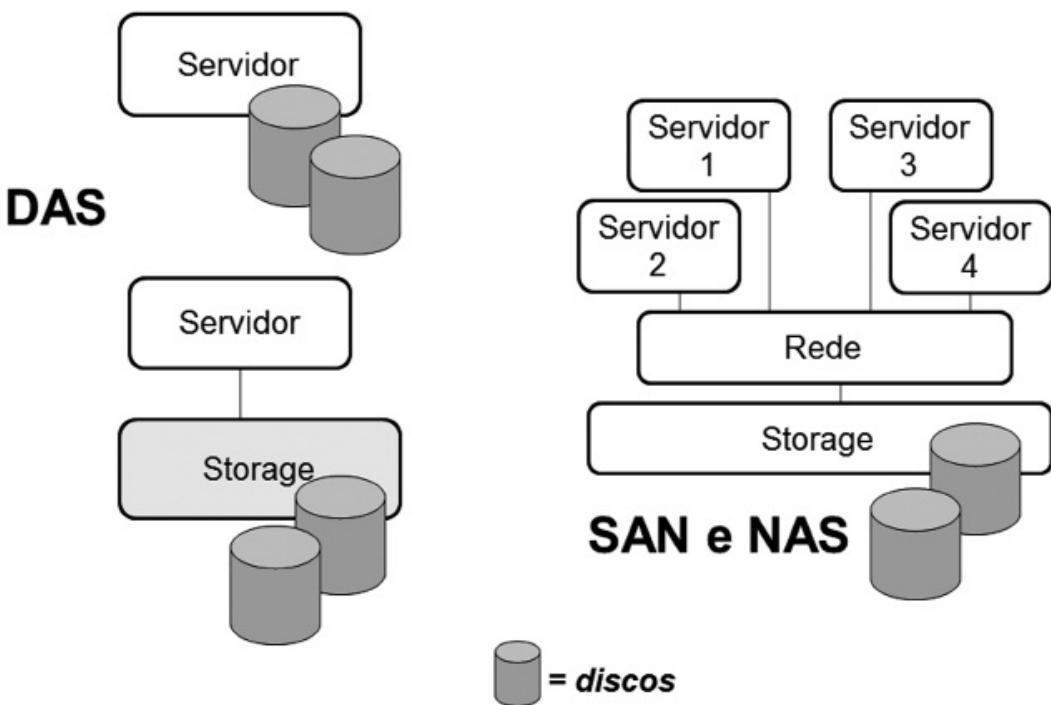


Figura 6-1 Opções de armazenamento

Nas redes SAN a infraestrutura de rede de armazenamento pode ser baseada em Fibre Channel (FC) ou Gigabit Ethernet (FCoE) e o dado a ser transportado é preferencialmente do tipo “bloco”. Nas redes NAS a infraestrutura é quase sempre Gigabit Ethernet do tipo LAN e o dado a ser armazenado é do tipo “arquivo” com sistemas NFS (*Network File System*) e CIFS (*Common Internet File System*). O entendimento de quando utilizar uma infraestrutura ou outra é complexo e confuso, muitas vezes causando problemas em projeto quando soluções que deveriam ser baseadas em NAS são baseadas em SAN e vice-versa.

Arquitetos de infraestrutura consideram diversas opções quando selecionam a solução de armazenamento adequada para a tarefa de armazenar dados. Grandes empresas utilizam várias tecnologias de armazenamento em conjunto, cada uma delas sendo selecionada para satisfazer as necessidades de uma subclasse específica de armazenamento de

dados. Essas combinações formam uma hierarquia de níveis de armazenamento de dados. O livro “DATACENTER: Componente Central da Infraestrutura de TI”, publicado pela Brasport e de minha autoria, trata do armazenamento hierárquico para DATACENTERS.

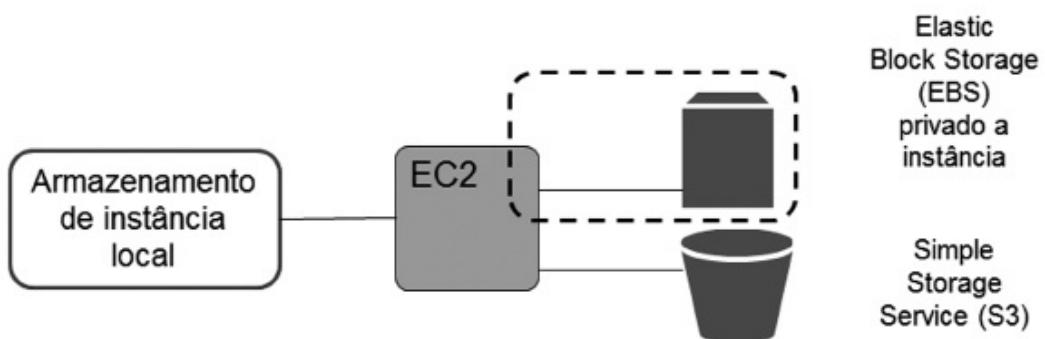
Cada uma das opções de armazenamento tradicional difere em desempenho, durabilidade e custo, bem como em suas interfaces de uma solução de armazenamento em cloud computing como a AWS. Boa parte das preocupações do arquiteto de infraestrutura não faz sentido quando se faz uso do armazenamento na nuvem, que é adquirido de acordo com a necessidade e pago mediante o uso. É uma mudança radical na forma de pensar o armazenamento.

O armazenamento AWS fornece opções equivalentes (mesmo que um pouco diferentes) ao armazenamento tradicional, só que adquiridas e pagas *on demand*. A **Tabela 6-1** relaciona as opções tradicionais de armazenamento com as opções AWS.

**Tabela 6-1 Opções de armazenamento**

Tradicional		AWS
DAS	Dispositivos de bloco local	Armazenamento de instância local EC2
SAN	LUNs	EBS, S3
NAS	arquivos NFS e CIFS	EBS, S3

A **Figura 6-2** ilustra as opções de armazenamento da AWS.



**Figura 6-2 Opções de armazenamento AWS**

O armazenamento de instância permite utilizar discos de estado sólido (*Solid-State Drives – SSD*) para instâncias de mais I/O visando o aumento de desempenho. Discos SSD possuem tempo de acesso reduzido, mas menor capacidade de armazenamento e são mais caros quando comparados aos discos convencionais baseados em sistemas magnéticos.

Joseph Baron e Robert Schneider fazem uma comparação interessante entre as opções do storage na AWS no artigo “Storage Options in the AWS Cloud”, de dezembro de 2010. Nos próximos itens são descritos resumidamente os principais resultados do levantamento feito pelos autores citados para as opções de armazenamento EBS e S3 da AWS.

## 6.3. Elastic Block Store (EBS)

### 6.3.1. Conceito

O EBS é uma forma de armazenamento para as instâncias EC2. Os volumes EBS oferecem armazenamentos que persistem independentemente da vida de uma instância.

Os volumes EBS apresentam alta disponibilidade e confiabilidade e podem ser aproveitados como uma partição de inicialização do EC2 ou ligados a uma instância em execução do EC2 como um dispositivo de bloco padrão. Por padrão, qualquer volume EBS criado quando a instância é lançada é automaticamente excluído quando a instância é terminada.

Quando o EBS é utilizado como uma partição de inicialização, as instâncias EC2 podem ser interrompidas e posteriormente reiniciadas, permitindo pagar apenas pelos recursos de armazenamento utilizados, mantendo o estado da instância.

Depois que um volume EBS é anexado a uma instância EC2, pode-se interagir com ele como se faria com uma unidade física de disco rígido, geralmente formatando-o com um sistema de arquivo escolhido. Pode-se usar um volume EBS para iniciar uma instância EC2 (apenas para AMIs EBS) e anexar vários volumes EBS a uma única instância EC2. Observe, no entanto, que qualquer volume único do EBS pode ser ligado a apenas uma instância do EC2 a qualquer momento. Um volume do EBS não pode ser compartilhado com outros usuários, a menos que seja criado um *snapshot* EBS. Os tamanhos para volumes do EBS variam de 1 GB a 1 TB e os volumes EBS são alocados em incrementos de 1 GB.

A **Figura 6-3** ilustra as situações comumente encontradas na manipulação de volumes EBS. Essas operações são facilmente realizadas pelo console de gerenciamento AWS.

- No caso (a) o volume EBS é criado no lançamento da instância A.
- No caso (b) é criado um *snapshot* do volume EBS.
- No caso (c) o *snapshot* criado dá origem a um novo volume e este é anexado à instância B.
- No caso (d) o volume é “desanexado” da instância A e anexado à instância B.

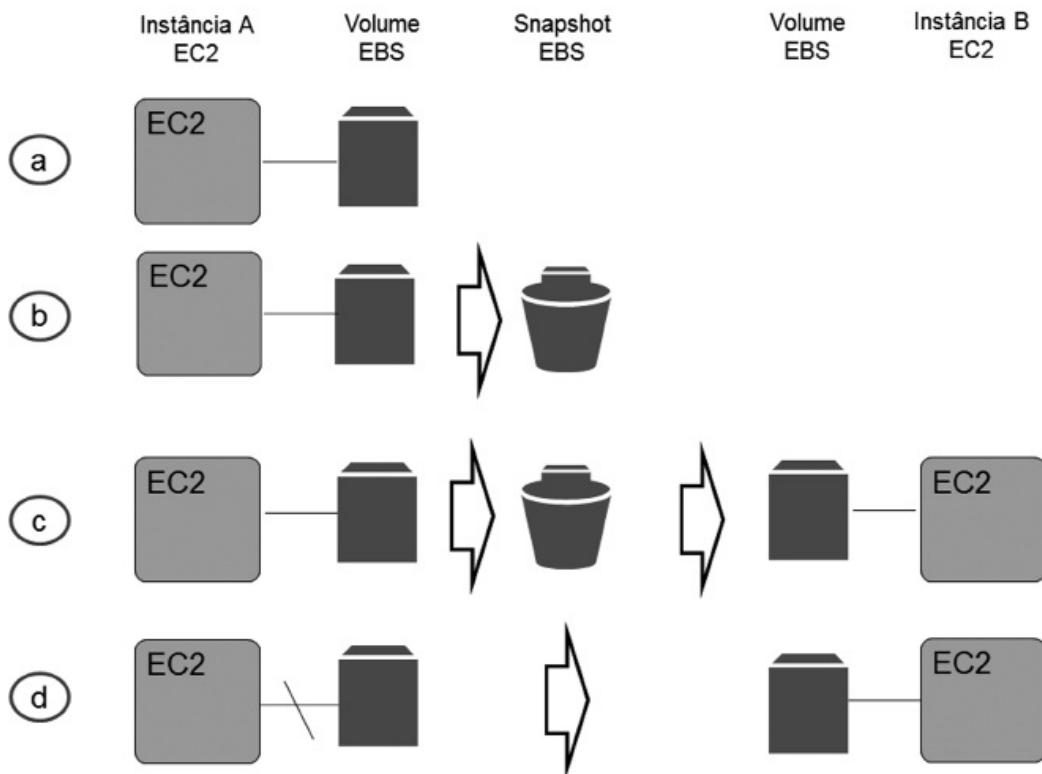


Figura 6-3 Utilização do EBS

O EBS destina-se principalmente a dados que mudam com frequência e precisam de persistência a longo prazo. A Amazon reforça que o EBS é particularmente interessante para uso como armazenamento principal para um sistema de arquivos, banco de dados ou para todos os aplicativos que necessitem de atualizações granulares finas e acesso ao armazenamento de blocos não formatados.

Cada volume EBS é automaticamente replicado na mesma zona de disponibilidade, para evitar perda de dados devido à falha de qualquer componente de hardware. O EBS também permite criar *snapshots* de volumes em um determinado momento, que persistem no S3. Esses *snapshots* podem ser usados como ponto inicial para criação de novos volumes EBS e para proteger dados em sistemas de backup a longo prazo.

A durabilidade do volume EBS depende tanto do seu tamanho quanto da porcentagem de dados que foi alterada desde o último *snapshot*. *Snapshots* do EBS são backups incrementais contendo apenas os blocos de dados alterados desde o último *snapshot*. Segundo a Amazon, os volumes EBS que operam com 20 GB ou menos de dados modificados desde o mais recente *snapshot* possuem uma taxa anual de falha (*Annual Failure Rate* – AFR) entre 0,1% e 0,5%. Caso ocorra uma falha no volume EBS, o que é uma situação improvável, todos os *snapshots* daquele volume permanecerão intactos. Lembre-se de que eles estão armazenados no S3 e permitirão a recriação do volume a partir do momento do último *snapshot*.

Os volumes EBS são projetados para serem altamente confiáveis e

disponíveis. No entanto, em função de serem criados em uma determinada zona de disponibilidade, eles não estarão disponíveis se a própria zona de disponibilidade não estiver disponível. Observe que, embora qualquer volume único EBS seja restrito a uma única zona de disponibilidade, um *snapshot* de um volume do EBS poderá estar disponível em todas as zonas de disponibilidade dentro de uma região e pode-se usar um *snapshot* do EBS para criar um ou mais novos volumes do EBS em qualquer zona de disponibilidade. Os *screenshots* EBS também podem ser compartilhados com outras contas de usuário. Isso facilita utilizar o *snapshot* para fazer backup e restore.

A **Figura 6-4** ilustra a caixa de diálogo no console de gerenciamento do EC2 para criação de volumes e *screenshots*.

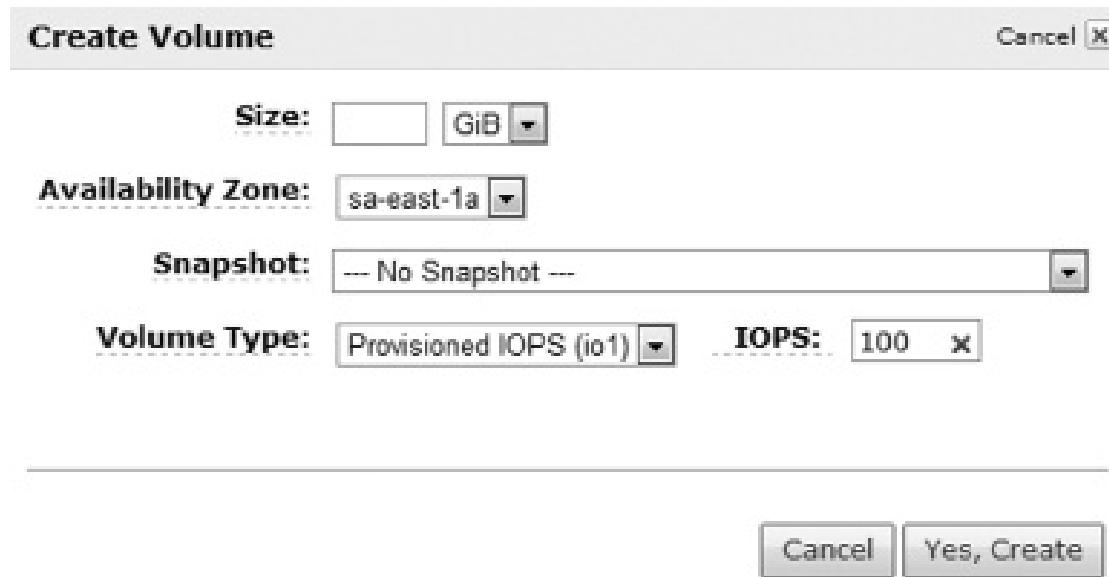


Figura 6-4 Create volume – 1

O EBS facilita a expansão de espaço disponível de armazenamento para a instância EC2. Pode-se criar um novo volume EBS e anexá-lo à instância e em seguida começar a utilizá-lo.

- Para expandir o tamanho de um volume, faça um *snapshot* do volume EBS para o S3 (utilizando a opção “Create Snapshot”). Crie um novo volume EBS a partir do *snapshot* (opção “Create Volume from Snapshot”) e especifique um tamanho maior do que o do volume original. As Figuras 6-5 (a) e Figura 6-5 (b) ilustram a operação em duas fases.

## Delete Snapshot

## Snapshot Permissions

---

## Create Volume from Snapshot

---

## Create Image from Snapshot

---

## Add/Edit Tags

(a) Primeira fase

**Create Volume** Cancel 

**Size:**  GiB 

**Availability Zone:**  

**Snapshot:**  

**Volume Type:**   **IOPS:**  

---

Cancel Yes, Create

(b) Segunda fase

Figura 6-5 Create volume – 2

- Anexe o novo volume à instância do EC2 (“Attach Volume”) e depois delete o volume EBS original (“Delete Volume”).

### 6.3.2. EBS otimizado

A Amazon verificou, desde o lançamento dos volumes EBS em 2008, que certas cargas de trabalho que precisam de muito I/O de forma consistente e outras que necessitam de muito I/O em bases absolutas acabam por precisar de uma garantia maior de alto desempenho do I/O.

Um volume EBS padrão pode entregar cerca de 100 IOPS (*Input/Output Operations Per Second*) em média, com habilidade de atender centenas de IOPS em uma base de melhor esforço. Volumes EBS padrão são ótimos para aplicações com requisitos de I/O moderado ou intermitente, bem como para

volumes de inicialização.

Os novos volume EBS provisionados permitem definir o nível de *throughput* necessário e correntemente suportam o provisionamento de até 1.000 IOPS (para blocos de 16 K). Para maior performance, pode-se fazer RAID de múltiplos volumes IOPS, possibilitando entregar um IOPS maior por volume lógico. Esses volumes provisionados oferecem desempenho consistente e são adequados para o armazenamento de banco de dados, processamento de transações e outras cargas pesadas de I/O aleatórias.

Para o máximo desempenho de IOPS para um volume EBS, pode-se agora solicitar o lançamento de instâncias EC2 otimizadas para uso do EBS. Uma instância com EBS otimizado é provisionada com *throughput* dedicado para um determinado volume EBS.

Os tipos disponíveis atualmente como instâncias EBS otimizadas são: *m1.large*, *m1.xlarge* e *m2.4xlarge*. Instâncias *m1.large* podem transferir dados para e do EBS a uma taxa de 500 Mbits/segundo; instâncias *m1.xlarge* e *m2.4xlarge* podem transferir dados a uma taxa de 1000 Mbits/segundo. A Amazon alerta que esta é uma taxa de transferência adicional e não afeta outro *throughput* de rede de uso geral já disponível para a instância.

É possível criar volumes EBS com IOPS provisionado utilizando o console de gerenciamento AWS, ferramentas de linha de comando, ou através de APIs EC2. Com o console, é preciso apenas criar e definir o tamanho do volume a ter o IOPS provisionado e, em seguida, digitar o número desejado de IOPS conforme ilustra a **Figura 6-6**.

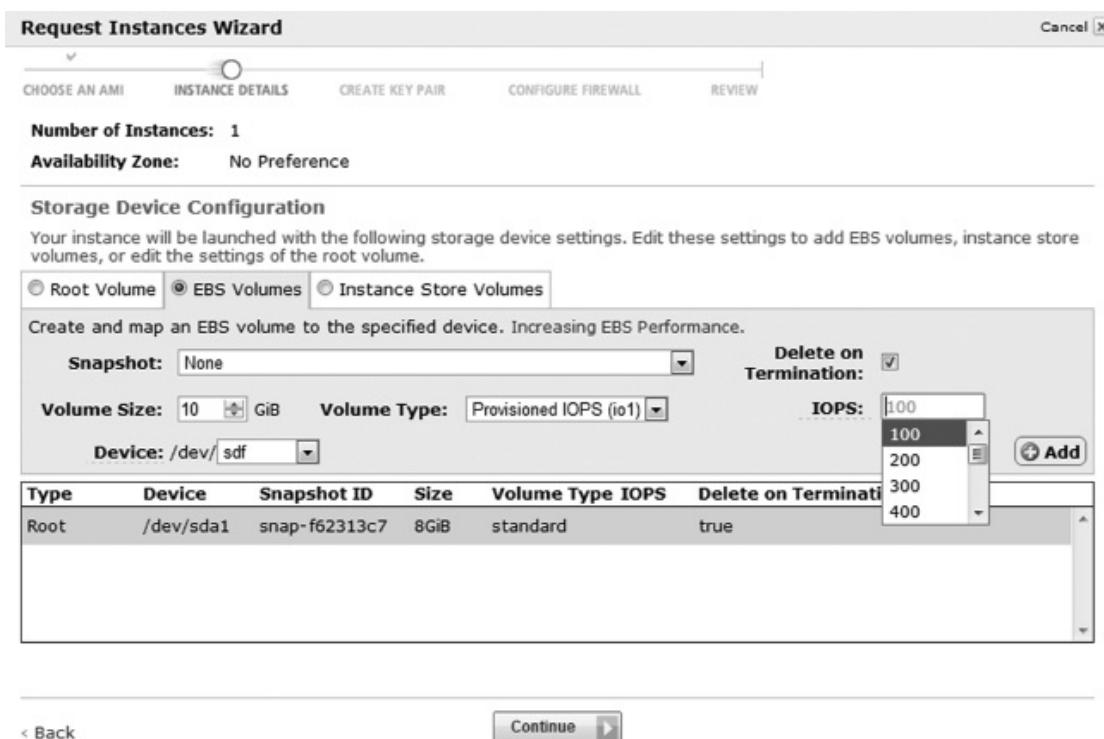


Figura 6-6 Storage device configuration

Pode-se fazer de forma fácil um *upgrade* para uma instância EC2 que utiliza um volume EBS otimizado.

1. Faça *shut down* nas aplicações que rodam na instância.
2. Pare a instância com a opção “Stop”.
3. Modifique a instância usando o comando *ec2-modify-instance-attribute* e o flag EBS-Optimized. Mude o tipo de instância para uma das instâncias suportadas, se necessário.
4. Inicie a instância com a opção “Start”.

A API EC2 foi atualizada e agora suporta também a criação de volumes IOPS provisionados. Por exemplo, o seguinte comando pode criar um volume de 500 GB com um IOPS provisionado de 1000:

```
ec2-create-volume --size500--availability-zone us-east-1b --type io1 --iops 1000.
```

O link <http://alestic.com/2012/08/ec2-provisioned-iops-ebs> explica detalhadamente a migração de uma instância não provisionada para uma instância provisionada.

### 6.3.3. EBS root device

Os dados armazenados em um volume EBS persistirão independentemente da vida útil da instância. A opção *DeleteOnTermination* flag é setada para “true” por padrão.

A **Figura 6-7** ilustra operações com o EBS *root device*. Observa-se que as opções *stopped* e *terminated* têm diferentes efeitos nos volumes anexados à instância que está rodando.

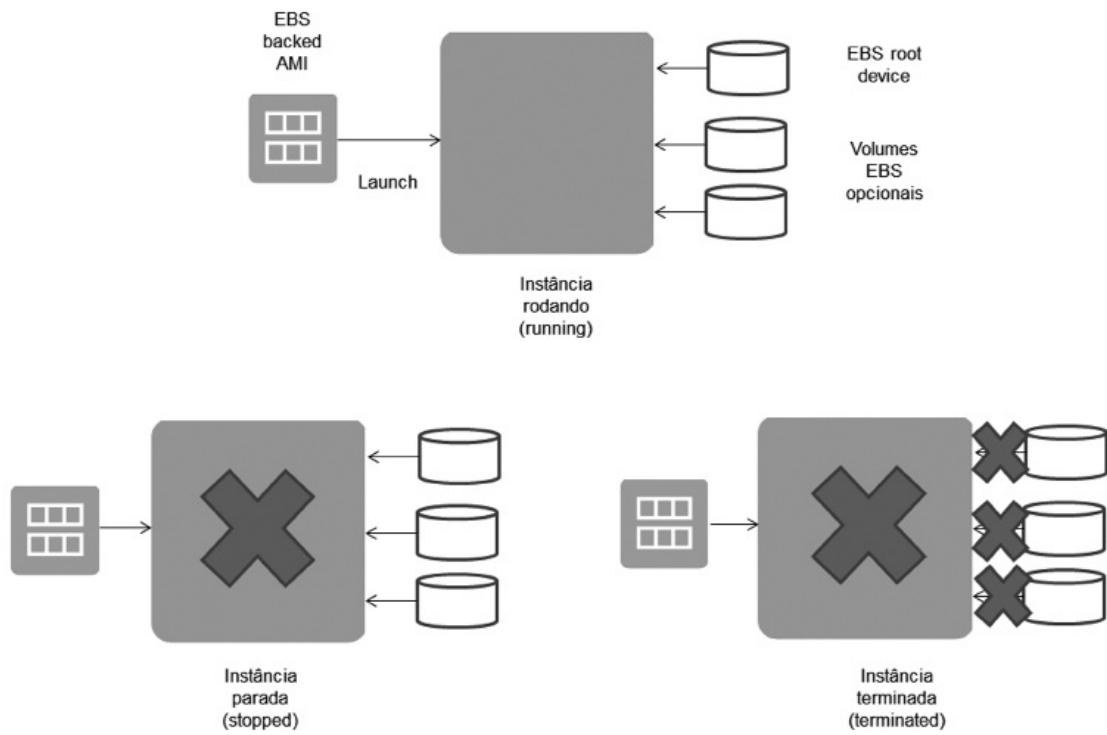


Figura 6-7 Operações com EBS root device

A **Tabela 6-2** compara a opção de *root device* de EBS com a opção de *root device* de instância local. Observe que em geral a opção EBS é melhor.

**Tabela 6-2 Root device de EBS versus root device de instância local**

Características	Root device de EBS	Root device de instância local
<b>Tempo de boot</b>	Menos de um minuto	Menos de cinco minutos
<b>Limite de tamanho</b>	1 TB	10GB
<b>Localização do root device</b>	Dado persiste na falha da instância	Dado persiste durante a vida da instância
<b>Upgrading</b>	Tipo de instância, kernel, RAM disk e dados de usuário podem ser modificados parando a instância	Atributos da instância são fixos durante a sua vida
<b>Taxas</b>	Uso da instância, uso do volume EBS, volume do snapshot EBS	Uso da instância e taxas do S3 para o armazenamento AMI
<b>Criação da AMI</b>	Command/call	Requer ferramentas de instalação de AMIs
<b>Estado <i>stopped</i></b>	Pode ser utilizado se a instância não estiver sendo utilizada	Instâncias só estão rodando ou não. Não existe estado <i>stopped</i> .

### **6.3.4. Importante**

- Volumes de armazenamento de instância local no EC2 oferecem armazenamento temporário no nível de bloco para instâncias do EC2. Ao criar uma instância EC2 a partir de determinada AMI, ela é acompanhada de um bloco pré-configurado de armazenamento de disco que já vem pré-anexado à instância. Ao contrário dos volumes EBS, os dados nos volumes de armazenamento de instância local persistem apenas durante a vida da instância associada ao EC2.
- Volumes de armazenamento de instância local no EC2 não se destinam a serem usados como armazenamento duradouro em disco e são ideais para o armazenamento temporário de informações que estão mudando continuamente, tais como buffers, caches, dados e outros conteúdos temporários, ou para dados que são replicados em todas as instâncias oriundos de um balanceamento de carga de servidores web.
- Para criar, deletar e anexar volumes EBS a uma instância EC2, a AWS também oferece APIs nos formatos SOAP e REST. As APIs são utilizadas para manipular volumes e *snapshots* do EBS.
- Se a criptografia for importante, recomenda-se executar um sistema de arquivos criptografado no topo do volume EBS. Aspectos de segurança serão vistos no capítulo 13.
- Uma consideração importante é que os *snapshots* do EBS podem ser feitos em tempo real enquanto o volume estiver anexado e em uso. No entanto, os *snapshots* capturam somente dados que foram registrados no volume EBS, o que poderá excluir quaisquer dados que foram localmente armazenados em cache pelo aplicativo ou SO.
- Por padrão, cada cliente EC2 tem um limite de 5.000 volumes ou um tamanho agregado de 20 TB (o que for menor) que podem ser executados a qualquer momento. Se for necessário aumentar esses limites, preencha um formulário específico disponível no site AWS para enviar uma solicitação de aumento do limite do EBS.
- A opção EBS Snapshot Copy permite copiar snapshots EBS através de regiões AWS. Esta opção permite criar uma estratégia de continuidade para o negócio.

## **6.4. Simple Storage Service (S3)**

### **6.4.1. Conceito**

O *Simple Storage Service* (S3) é a principal forma de armazenamento da AWS. Ele armazena objetos distribuídos e foi projetado para armazenamento de dados primários, secundários e de missão crítica através de uma interface

web service fácil de usar.

Em ambientes tradicionais, conforme visto, dados críticos seriam normalmente armazenados em redes SAN ou NAS. No entanto, segundo a Amazon, um mecanismo baseado em nuvem como o S3 é bem mais ágil, flexível e georredundante. O S3 fornece uma interface web service que pode ser usada para armazenar e recuperar qualquer quantidade de dados, a qualquer momento, de qualquer lugar na web. Pode-se gravar, ler e deletar objetos contendo de um byte até cinco terabytes de dados cada, e o número de objetos que você pode armazenar em um *bucket* do S3 é ilimitado. O S3 também é altamente escalável, permitindo o acesso simultâneo à leitura e à gravação dos dados por diferentes clientes ou *threads* de aplicativo.

A arquitetura do S3 é projetada para ser neutra em termos de linguagem de programação, usando interfaces para armazenar e recuperar objetos. A API S3 fornece interfaces SOAP e REST.

#### 6.4.2. Utilização

O S3 é uma plataforma abrangente que está disponível para diversas necessidades de armazenamento de dados. Ela é excelente para:

- **Armazenamento de conteúdo estático da web.** O conteúdo estático pode ser enviado diretamente a partir do S3 por meio de um servidor web, visto que cada objeto do S3 possui seu próprio endereço URL baseado em HTTP, ou enviado por meio de uma rede de fornecimento de conteúdo como o CloudFront. O S3 funciona muito bem para hospedagem de conteúdo da web com exigências de largura de banda. Além disso, como o S3 não faz provisionamento de armazenamento, este funciona bem para websites em rápido ritmo de crescimento de conteúdo de dados gerados pelo usuário.
- **Armazenamento de dados para computação em larga escala.** Por causa de escalabilidade horizontal do S3, os dados podem ser acessados a partir de múltiplos nós de computação simultaneamente, sem ficar restrito a uma conexão única.
- **Armazenamento de dados de missão crítica.** Armazenamento altamente confiável para *snapshots* de volumes EBS, aplicações de armazenamento de backup e soluções “quentes” de recuperação de desastres para continuidade de negócios. Dado que o S3 armazena objetos de forma redundante em múltiplos dispositivos através de várias zonas de disponibilidade, ele fornece uma infraestrutura de armazenamento altamente durável necessária para essas situações.

Segundo a Amazon, há vários fatores a serem considerados na hora de hospedar os dados produzidos por um aplicativo específico no S3:

- Proximidade dos clientes para reduzir latências de acesso aos dados.

- Afastamento de outras instalações para fins de recuperação de desastres e redundância geográfica.
- Atendimento de requisitos específicos, legais e normativos.
- Redução de custos de armazenamento. Pode-se optar por uma região de custo mais acessível para economizar.
- Deve-se inicialmente, quando do uso do S3, definir o nome do *bucket* e a região onde os dados ficam armazenados. Esta escolha da região é um aspecto importante para efeito de conformidade. A Figura 6-8 ilustra esta opção.



Figura 6-8 Select a bucket name and region

#### 6.4.2.1. Uso do S3 com o EC2

A combinação do S3 com o EC2 fornece capacidade computacional redimensionável com infraestrutura de armazenamento de dados altamente dimensionável e rápida.

O EC2 utiliza o S3 para armazenar AMIs. É possível utilizar AMIs para o lançamento de instâncias EC2. Em caso de falha de instância, pode-se usar a AMI armazenada e imediatamente iniciar outra instância, permitindo desse modo a rápida recuperação e continuidade de negócios.

O EC2 também utiliza o S3 para armazenar *snapshots* de volumes. Pode-se usar *snapshots* para recuperação de dados de forma rápida e confiável em caso de falhas de sistema ou aplicativo.

#### 6.4.2.2. Uso do S3 com site web

O S3 pode hospedar sites estáticos de forma muito simples, sem depender de ter uma instância no ar. Para rotear as consultas para um site hospedado em um *bucket* S3 deve-se:

- Criar um *bucket* S3 e configurar o site.
- Criar um domínio ou subdomínio que usa o Route 53 como o serviço DNS, ou migrar um domínio existente ou subdomínio para o Route 53.
- No console do Route 53, criar um registro CNAME, que redireciona o tráfego do nome de domínio para o nome de domínio do S3.

Importante ressaltar a funcionalidade do S3, que permite redirecionar páginas web alterando a URL no site hospedado no S3 (por exemplo, de [www.example.com/oldpage](http://www.example.com/oldpage) para [www.example.com/newpage](http://www.example.com/newpage)) sem quebrar

links ou deixar favoritos apontando para as URLs antigas. Os usuários, quando acessam a URL antiga, serão automaticamente redirecionados para a nova. O ranking de busca de uma página web também não sofre impacto com esta funcionalidade. Pode-se utilizar regras de redirecionamento para automatizar o processo.

O site hospedado no S3 pode ser acessado sem especificar o “www” no endereço Web. Para utilizar esta opção o Route 53 deve hospedar os dados do DNS.

### 6.4.3. Estrutura

Os três conceitos básicos que formam a estrutura do S3 são *buckets*, objetos e chaves.

#### 6.4.3.1. Buckets

*Buckets* são containers para objetos S3. Cada objeto armazenado no S3 está contido em um *bucket*, e ele funciona como um diretório em um sistema de arquivos. Uma das principais distinções entre uma pasta de arquivos e um *bucket* é que cada *bucket* e seu conteúdo podem ser acessados usando uma URL.

Cada conta do S3 pode conter um máximo de cem *buckets*. *Buckets* não podem ser aninhados um dentro do outro; assim, não é possível criar um *bucket* dentro de outro *bucket*.

É possível definir a localização geográfica dos *buckets* especificando uma restrição de localização ao criá-los. Isso automaticamente garantirá que quaisquer objetos armazenados dentro daquele *bucket* sejam armazenados naquela localização geográfica. Se não especificar uma localização ao criar o *bucket*, o seu conteúdo será armazenado na localização mais próxima do endereço de cobrança da conta.

Nomes de *bucket* precisam atender aos seguintes requisitos:

- O nome deve iniciar com um número ou uma letra.
- O nome deve ter entre três e 255 caracteres.
- Um nome válido pode conter somente letras minúsculas, números, pontos, sublinhados e travessões.
- Apesar de ser possível usar números e pontos no nome, eles não podem ter o formato de um endereço IP.
- O nome do *bucket* é compartilhado com todos os *buckets* de todas as contas no S3. O nome deve ser exclusivo em todo o S3.
- Os *buckets* estão sujeitos a políticas que permitem o controle centralizado de permissões e objetos baseados em várias condições de operação, solicitadores (*requesters*) e recursos. As políticas são

expressas em termos de uma *Access Policy Language (APL)* vista no capítulo 3.

#### 6.4.3.2. Objetos

Objetos contêm os dados armazenados dentro de *buckets* no S3. Pense em um objeto como o arquivo que se deseja armazenar. Cada objeto armazenado é composto de duas entidades: dados e metadados.

Os dados são a coisa real a ser armazenada, como um arquivo PDF, por exemplo. Os dados armazenados também têm metadados associados para descrever o objeto. Os metadados armazenam o tipo de conteúdo do objeto que está sendo armazenado, a data em que o objeto foi modificado pela última vez e quaisquer outras informações específicas do usuário ou do aplicativo. Os metadados de um objeto são especificados pelo desenvolvedor como pares chave-valor quando o objeto é enviado ao S3 para armazenamento.

Diferentemente da limitação no número de *buckets*, não existem restrições no número de objetos. É possível armazenar um número ilimitado de objetos nos *buckets* e cada objeto pode conter até 5 GB de dados. Os dados nos objetos S3 publicamente acessíveis podem ser recuperados por HTTP, HTTPS ou Bit Torrent. É possível fazer o download de quaisquer dados publicamente disponíveis no Amazon S3 através do protocolo *BitTorrent*, para complementar o mecanismo de distribuição padrão de cliente/servidor. Para utilizá-lo basta adicionar o parâmetro *?torrent* no final da solicitação GET na API REST.

O *BitTorrent* é um protocolo de distribuição de internet de código aberto que permite que os desenvolvedores economizem com os custos de largura de banda para uma porção popular de dados, permitindo que os usuários façam download a partir da Amazon e de outros usuários simultaneamente.

A **Figura 6-9** ilustra as opções disponibilizadas através do clique com o botão direito do mouse sobre o objeto.

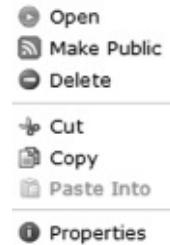


Figura 6-9 Opções do S3

### Acessar *buckets* e objetos

Para acessar *buckets* e objetos S3 que foram criados usando *CreateBucketConfiguration* ou pelo console de gerenciamento, basta fazer a seguinte solicitação em todas as regiões:

<http://yourbucket.s3.amazonaws.com/yourobject>

## Exclusão de objetos

Pode-se excluir um ou mais objetos diretamente do S3:

- **Eliminar um único objeto:** S3 fornece a API de excluir, que pode ser usada para excluir um objeto em uma única solicitação HTTP.
- **Excluir vários objetos:** S3 também fornece a API de excluir multiobjetos, que pode ser usada para eliminar até mil objetos em uma única solicitação HTTP.

## Recuperação de objetos

O controle de versão permite preservar, recuperar e restaurar todas as versões de cada objeto armazenado em um *bucket* do S3. Depois de habilitar o controle de versão para um *bucket*, o S3 preserva objetos existentes sempre que seja realizada uma operação PUT, POST, COPY ou DELETE. Por padrão, solicitações GET irão recuperar a versão mais recentemente gravada. Versões mais antigas de um objeto substituído ou excluído podem ser recuperadas ao se especificar a versão na solicitação.

O S3 oferece aos clientes uma infraestrutura de armazenamento altamente durável. O controle de versão oferece um nível adicional de proteção, fornecendo um meio de recuperação caso os clientes substituam ou excluam objetos acidentalmente. Isso permite recuperar os dados armazenados de ações não intencionais de usuário e de falhas do aplicativo. Pode-se usar o controle de versão para arquivamento e retenção de dados também.

## Expiração de objetos

Esta função do S3 permite definir regras para agendar a remoção de objetos após um período de tempo predefinido. As regras são especificadas na política de configuração do ciclo de vida que se aplica a um *bucket*. Pode-se atualizar esta política através da API do S3 ou através do console de gerenciamento da AWS.

Cada regra tem os seguintes atributos:

- **Prefixo:** parte inicial do nome da chave (por exemplo, "logs /"), ou o nome completo da chave. Qualquer objeto no *bucket* com um prefixo correspondente estará sujeito a esta regra de validade. Um prefixo vazio irá corresponder a todos os objetos no *bucket*.
- **Status:** ativado ou desativado. Pode-se optar por habilitar as regras de vez em quando para realizar a coleta de exclusão ou de lixo dos

*buckets* e deixar as regras de deficiência em outros momentos.

- **Vencimento:** especifica um período de validade para os objetos que estão sujeitos à regra, como o número de dias da data do objeto da criação.
- **Id:** opcional, dá um nome à regra. Pode-se definir até cem regras de expiração para cada um dos *buckets* S3; no entanto, as regras devem especificar prefixos distintos para evitar ambiguidade. Depois de uma regra de expiração do objeto ser adicionada, a regra é aplicada para objetos que já existem no *bucket*, bem como todos os novos objetos adicionados para o *bucket* após a regra ser criada. Pode-se usar esse recurso para expirar objetos criados ou objetos que a AWS criou em seu nome, incluindo registros, logs e dados criados pelos CloudFront e o Import/Export.

#### 6.4.3.3. Chaves

Cada objeto armazenado dentro de um *bucket* S3 é identificado usando uma chave exclusiva. Isso é similar em conceito ao nome de um arquivo em uma pasta de um sistema de arquivos.

O nome do arquivo dentro de uma pasta em um disco rígido deve ser exclusivo. Cada objeto dentro de um *bucket* tem uma chave exclusiva. O nome do *bucket* e da chave são usados em conjunto para fornecer a identificação exclusiva de cada objeto armazenado na S3.

O endereço de cada objeto dentro do S3 é uma URL que combina a URL de serviço do S3, o nome do *bucket* e a chave exclusiva. Apesar de serem conceitos simples, *buckets*, objetos e chaves, juntos, fornecem flexibilidade para criar soluções de armazenamento de dados.

É possível aproveitar esses blocos modulares para simplificar o armazenamento de dados no S3 ou usar da flexibilidade para criar camadas e construir armazenamento ou aplicativos mais complexos que utilizam o S3 para fornecer funções adicionais.

#### 6.4.4. CORS

A Amazon AWS permite compartilhar recursos de origem cruzada, também conhecidos por CORS (*Cross-Origin Resource Sharing*), para seu serviço de armazenamento S3, permitindo aos desenvolvedores criar mais facilmente aplicações web que acessam dados armazenados no S3.

O CORS permite aos desenvolvedores criar aplicativos web que fazem solicitações para outros domínios diferentes do que forneceu o conteúdo principal. Páginas web externas, folhas de estilo e aplicações HTML 5 hospedadas em diferentes domínios podem agora usar fontes da web e imagens armazenadas no S3, permitindo que esses recursos sejam usados em vários sites.

Para ser capaz de usar solicitações *cross-origin*, os desenvolvedores têm que criar uma configuração CORS, um documento XML com regras que identificam o que pode acessar as configurações S3. Podem ser adicionadas até cem regras para o documento de configuração.

#### 6.4.5. Otimização de desempenho

É possível otimizar o desempenho do S3 de duas formas:

- Utilizando o *TCP Window Scaling*, que permite melhorar o desempenho de *throughput* de rede entre o sistema operacional, a camada de aplicativo e o S3, oferecendo suporte a janelas maiores que 64 KB. Apesar do dimensionamento de uma janela TCP poder melhorar o desempenho, pode ser um desafio defini-la corretamente. Certifique-se de ajustar as configurações no aplicativo e no nível do kernel. Para obter mais informações sobre redimensionamento da janela TCP, consulte a documentação do sistema operacional e a RFC 1323.
- Utilizando o *TCP Selective Acknowledgement*, que permite aumentar o tempo de recuperação após um grande número de perdas de pacote. O *TCP Selective Acknowledgement* é suportado por sistemas operacionais mais novos, mas pode precisar ser habilitado. Para obter mais informações sobre confirmações seletivas TCP, consulte a documentação que acompanha o sistema operacional e a RFC 2018.

#### 6.4.6. Controle

##### 6.4.6.1. Políticas IAM, políticas de *bucket* e ACLs

O S3 suporta permissões baseadas em recursos para objetos e *buckets* com políticas de *buckets* e lista de controle de acesso (*Access Control Lists* – ACLs). As ACLs e as políticas de *bucket* definem quais contas AWS têm acesso ao S3 e o tipo de acesso. É possível usar ambas em conjunto com as políticas de usuário IAM para controlar o acesso a recursos do S3. A **Tabela 6-3** descreve as relações entre ACLs, políticas de *bucket* e políticas IAM.

Com as ACLs, pode-se conceder acesso a outras contas AWS para os recursos S3. Com políticas IAM, pode-se conceder permissão aos usuários dentro da conta AWS para os recursos S3. Com políticas de *bucket*, é possível fazer ambos.

Tabela 6-3 ACLs, políticas IAM e políticas de *bucket*

Tipo de controle de acesso	Controle no nível da conta AWS	Controle no nível de usuário
ACLs	Sim	Não

<b>Políticas IAM</b>	Não	Sim
<b>Políticas de <i>bucket</i></b>	Sim	Sim

#### 6.4.6.2. Políticas IAM e de *buckets* juntas

Políticas de IAM e *bucket* podem ser utilizadas juntas. Quase sempre se conseguem os mesmos resultados com as duas alternativas. A **Figura 6-10** ilustra políticas equivalentes feitas com o IAM e diretamente só com política para o *bucket*.

A política IAM (à esquerda) permite a ação *PutObject* para o *bucket* chamado *bucket\_xyz* em uma conta AWS, e a política está conectada aos usuários Manoel e Paulo (significa que Manoel e Paulo têm as permissões estabelecidas na política).

O *bucket\_xyz* acompanha a política de *bucket* (à direita). Como a política IAM, a política de *bucket* fornece a Manoel e a Paulo a permissão para a ação *PutObject* no *bucket\_xyz*.

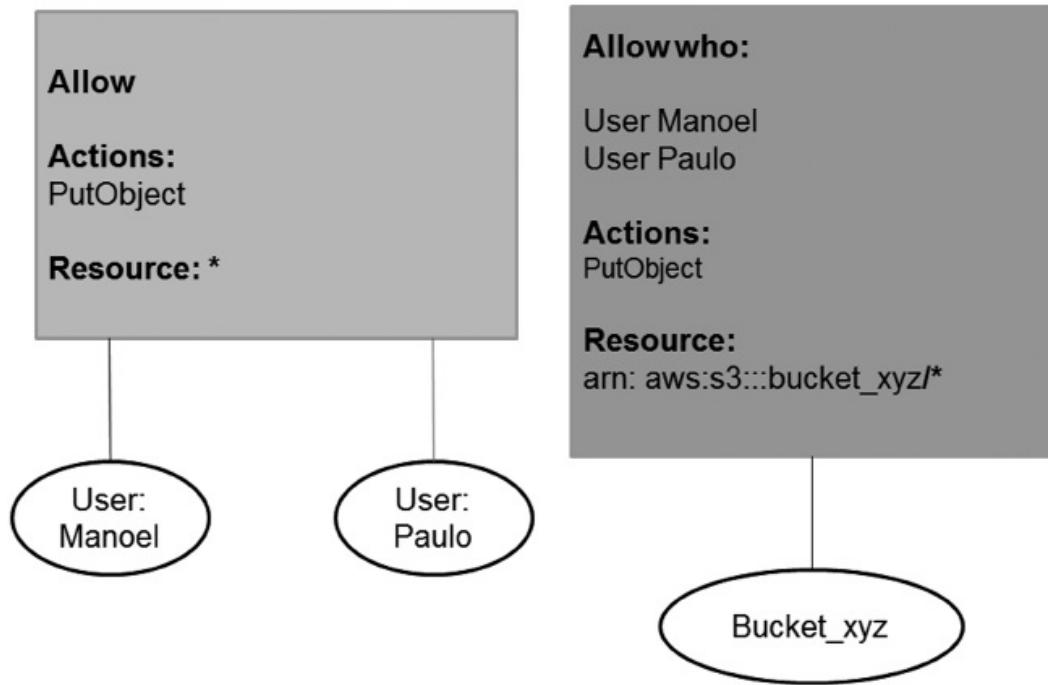


Figura 6-10 Políticas para *buckets* e IAM

#### 6.4.7. S3 na prática

O guia de conceitos básicos “Simple Storage Service (API Version 2006-03-01)” descreve as operações básicas com o S3. É necessário possuir uma conta na AWS para se inscrever no S3.

##### 6.4.7.1. Criar um *bucket*

A partir do painel do console AWS:

- Na caixa de lista suspensa, selecione Amazon S3.
- Na guia do S3, clique com o mouse em “Create a Bucket – Select a Bucket Name and Region”.
- Insira um nome de *bucket* no campo “Bucket Name”. Uma vez criado um *bucket*, não é possível mudar o seu nome. Além disso, o nome do *bucket* é visível na URL que aponta para os objetos nele armazenados. Verifique se o nome de *bucket* escolhido é adequado.
- Na caixa de lista suspensa “Region”, selecione a região onde deverão ficar os dados. Por padrão, o S3 cria *buckets* na região EUA. É possível escolher uma região que possa otimizar a latência, minimizar custos ou atender aos requisitos regulatórios. Os objetos armazenados em uma região nunca saem dela, exceto se for explicitamente desejado transferi-los para outra região.
- Clique em “Create”.

##### 6.4.7.2. Adicionar um objeto a um *bucket*

- Na guia do S3 no console de gerenciamento AWS, clique no *bucket* para o qual se deseja fazer o carregamento de um objeto e, em seguida, clique em “Upload – Select Files”.
- Clique em “Add Files” para selecionar o arquivo para upload.
- Selecione o arquivo que deseja enviar e clique em “Open”. O assistente de upload “Select Files” exibe os arquivos e pastas selecionados para carregar.
- Clique em “Start Upload”.

Pode-se ver o progresso do upload usando o painel “Transfers”, que aparece na parte inferior da tela assim que o upload começa.

O arquivo será adicionado ao *bucket*. Verifique se tudo correu bem.

#### **6.4.7.3. Abrir ou fazer download de um objeto**

- No guia S3 no console de gerenciamento AWS, clique com o botão direito do mouse no objeto a ser aberto.
- Clique em “Open” para abrir o objeto no browser ou faça o download para salvar o objeto localmente. O objeto abre no browser ou, se estiver fazendo o download do objeto, uma caixa de diálogo abre para que seja especificada a localização da pasta onde se deseja salvar o objeto. O objeto está aberto.
- Por padrão os *buckets* do S3 são objetos privados. Para exibir o objeto usando uma URL, utilize a seguinte forma:  
<https://s3.amazonaws.com/Bucket/Object>

O objeto deve ser legível publicamente. Caso contrário, será necessário criar uma URL assinada que inclui uma assinatura com informações de autenticação.

#### **6.4.7.4. Mover e excluir um objeto**

- Para mover um objeto:
  - Na guia S3 no console de gerenciamento AWS, clique com o botão direito do mouse no objeto a ser movido. Clique em “Cut”.
  - Navegue até o *bucket* (e folder) para onde deseja mover o objeto e clique com o botão direito do mouse no folder e/ou no *bucket* para o qual deseja mover o objeto. Clique em “Paste Info”.
  - O S3 move os arquivos para o novo local.
- Para excluir um objeto:
  - Clique com o botão direito do mouse no objeto a ser excluído. Uma caixa de diálogo mostra as ações possíveis com relação ao(s) objeto(s) selecionado(s).

- Clique em “Delete”. Confirme a exclusão quando o console solicitar.

#### 6.4.7.5. Excluir um *bucket*

- Clique com o botão direito do mouse no *bucket* a ser excluído. Uma caixa de diálogo abre as ações possíveis com relação ao *bucket* selecionado.
- Clique em “Delete”. Confirme a exclusão quando o console solicitar.
- Agora foi excluído o *bucket* e todo o seu conteúdo.

#### 6.4.8. Importante

- Os clientes do S3 podem, opcionalmente, configurar os *buckets* para criar acesso a registros de log para todas as solicitações feitas a ele. Esses registros de log de acesso podem ser usados para fins de auditoria e contêm detalhes sobre a solicitação, tais como o tipo, os recursos especificados na solicitação e a hora e data em que foi processada.
- Os objetos nunca deixam a região onde foram armazenados, a menos que se queira transferi-los para fora. No entanto, é de responsabilidade do cliente garantir a conformidade com a legislação de privacidade da sua região.
- Se for feita a opção de utilizar o SSE (*Server Side Encryption*), a AWS lidará com o gerenciamento e a proteção de chave. Assim, deve-se escolher o SSE se preferir que a AWS gerencie suas chaves.

### 6.5. Opções de armazenamento

A **Tabela 6-4** resume as diferenças entre os tipos de armazenamento na AWS considerando também o armazenamento de instância local. Perde-se em desempenho no S3, mas se ganha em durabilidade, disponibilidade e escalabilidade.

**Tabela 6-4 Tipos de armazenamento AWS**

Parâmetros	Armazenamento de instância local	EBS	S3
Custo	Incluído no EC2	Provisionado por GB/mês	Armazenado por GB/mês
Desempenho	Alto	Moderado ou alto (com discos SSD)	Moderado
Durabilidade	Baixa	Moderada	Alta

<b>Disponibilidade</b>	Baixa	Moderada (utilizando <i>snapshots</i> )	Alta
<b>Escalabilidade</b>	Não	Manual	Automático
<b>Limites de tamanho</b>	160 GB a 1.6 TB	1 GB a 1 TB por volume	5 TB por objeto
<b>Persistência</b>	Não	Sim	Sim
<b>Interfaces</b>	Acesso via SO	Acesso via SO	Acesso via HTTP, REST ou SOAP
<b>Segurança (criptografia em repouso)</b>	Executar FS criptografado	Executar FS criptografado	Criptografar utilizando AES de 256 bits
<b>Segurança (criptografia em trânsito)</b>	N/D	N/D	SSL (HTTPS)
<b>Modelo</b>	Bloco	Bloco	Objeto
<b>Grau de automação</b>	Nenhum	Autoespelhamento	Autorreplicação
<b>Grau de redundância</b>	Não redundante	Redundante em uma zona de disponibilidade	Redundante entre zonas de disponibilidade
<b>Gerenciamento e administração</b>	Manual	Automático	Automático

## 6.6. Storage Gateway (SG)

O Storage Gateway (SG) pode ser visto como um “appliance virtual” que fornece integração entre *appliances* locais de armazenamento e a infraestrutura de armazenamento da AWS. É uma solução para nuvem híbrida, pois integra o DATACENTER corporativo com a nuvem AWS. Ele permite fazer backup dos dados corporativos para a nuvem AWS de forma rápida e econômica. Backup e restore serão tratados com detalhes no capítulo 12.

O SG permite criar volumes de armazenamento iSCSI locais, armazenando os dados no hardware, enquanto transfere e armazena de forma assíncrona esses dados no S3 em formato de *snapshots* do EBS. É possível acessar esses dados pela rede local ou por instâncias EC2 na nuvem AWS. A **Figura 6-11** ilustra o uso do Storage Gateway.

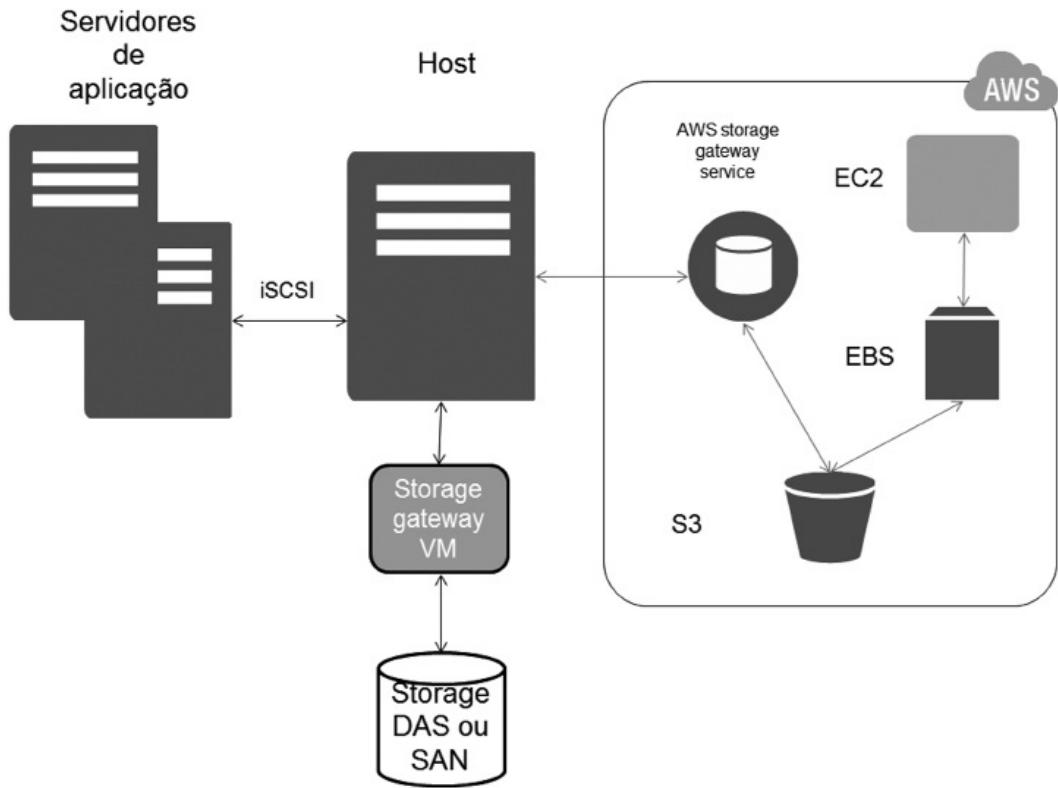


Figura 6-11 AWS storage gateway

O SG é compatível com os protocolos de armazenamento padrão que funcionam com os aplicativos de backup existentes. Ele fornece desempenho com baixa latência ao manter dados no hardware de armazenamento interno da organização enquanto carrega, de modo assíncrono, esses dados na AWS, onde eles são criptografados e armazenados de forma segura no S3.

Usando o SG, pode-se fazer backup para o S3 de *snapshots* pontuais dos dados dos aplicativos internos da organização visando a recuperação futura. Caso seja necessário obter capacidade de substituição para fins de recuperação de desastres, ou se for interessante impulsionar a capacidade computacional sob demanda do EC2 (visando obter uma capacidade extra durante períodos de pico para novos projetos ou como uma maneira mais econômica de executar as cargas de trabalho normais), é possível usar o SG para espelhar dados internos e utilizá-los com instâncias EC2 na AWS.

Os principais recursos do SG são:

- **Volumes armazenados:** fornecem aos aplicativos *in loco* acesso com baixa latência a conjuntos de dados completos, enquanto oferecem backups remotos duráveis. É possível criar volumes de armazenamento de até 1 TB e instalá-los como dispositivos iSCSI com base nos servidores de aplicativos *in loco*. Os dados gravados nos volumes armazenados no gateway são armazenados no hardware de armazenamento *in loco* e, de forma assíncrona, são armazenados no S3 na forma de *snapshots* do EBS.

- **Snapshots de dados:** o SG fornece a capacidade de criar e armazenar *snapshots* pontuais dos volumes de armazenamento no S3. Estes são armazenados como *snapshots* do EBS. Os *snapshots* são backups incrementais, reduzindo os encargos de armazenamento. Ao obter um novo *snapshot*, somente os dados que foram alterados desde o último *snapshot* são armazenados. Todo armazenamento de *snapshots* também é compactado, minimizando ainda mais os encargos de armazenamento.

A máquina virtual do Storage Gateway deve ser instalada em um *host* com os seguintes requisitos mínimos:

- VMware ESXi Hypervisor (v4.1).
- quatro processadores virtuais designados para a máquina virtual.
- 7,5 GB de RAM designados para a máquina virtual.
- 75 GB de espaço em disco para a instalação dos dados e do sistema.

O SG é compatível com a instalação de volumes de armazenamento usando iniciadores de software iSCSI do Microsoft Windows ou do RedHat.

A sequência de uso para o SG envolve:

- Provisionar um *host*.
- Fazer download e preparar uma máquina virtual (*Virtual Machine – VM*).
- Provisionar disco de armazenamento local.
- Ativar o Storage Gateway.

Maiores informações sobre o uso do SG podem ser encontradas na documentação técnica da AWS.

## 6.7. Import/Export (I/E)

O Import/Export (I/E) transfere os dados diretamente para dispositivos de armazenamento usando a rede interna de alta velocidade da AWS e ignorando a internet. A utilização do I/E com frequência é mais rápida do que a transferência via internet e mais vantajosa do que fazer um upgrade dos componentes da rede buscando aumento do desempenho para resolver problemas temporários.

O I/E oferece suporte à importação e à exportação de dados para dentro e para fora dos *buckets* do S3 na região onde é disponibilizado.

Se existem grandes volumes de dados para serem carregados e uma conexão de internet com largura de banda limitada, o tempo necessário para preparar e enviar um dispositivo de armazenamento portátil para a AWS pode ser uma pequena porcentagem do tempo levado para transferir seus dados

pela internet. A Amazon recomenda usar o AWS Import/Export para carregar dados que levariam uma semana ou mais para transferir pela internet.

A **Tabela 6-5** fornece orientação sobre velocidades de conexão de internet comuns em relação a:

- Quanto tempo levará a transferência de 1 TB de dados através da internet para a AWS utilizando uma capacidade de rede de 80%.
- Que volumes de dados totais exigem uma semana para serem transferidos pela internet para a AWS e, portanto, permitem considerar o uso da AWS Import/Export.

Por exemplo, se existir uma conexão de 10 Mbps e espera-se utilizar 80% da capacidade de rede para a transferência de dados, a transferência de 1 TB de dados pela internet para a AWS levará treze dias. O volume para que esta mesma configuração leve pelo menos uma semana é o de 600 GB.

A Amazon reforça então que se existem 600 GB de dados ou mais para transferir e deseja-se que essa transferência de dados para a AWS leve menos de uma semana, é recomendado utilizar o Import/Export.

**Tabela 6-5 Utilização do AWS Import/Export**

Conexão de internet disponível	Número mínimo de dias para transferir 1 TB com 80% de utilização da capacidade da conexão	Volume que exige uma semana para ser transferido pela internet para a AWS
T1 (1,544 Mbps)	82 dias	100 GB ou mais
10 Mbps	Treze dias	600 GB ou mais
T3 (44,736 Mbps)	Três dias	2 TB ou mais
100 Mbps	Um a dois dias	5 TB ou mais
1.000 Mbps	Menos de um dia	60 TB ou mais

Em resumo, o Import/Export facilita a rápida transferência de grandes volumes de dados para dentro e para fora da nuvem AWS. Pode-se utilizá-lo para:

- **Migração de dados:** se os dados precisam subir para a nuvem AWS pela primeira vez, o Import/Export costuma ser muito mais rápido do que a transferência de dados através da internet.
- **Distribuição de conteúdo:** envie dados para clientes a partir de

dispositivos de armazenamento portáteis.

- **Intercâmbio direto de dados:** é usual receber conteúdo em dispositivos de armazenamento portáteis de parceiros de negócios. É possível escolher que eles sejam enviados diretamente à AWS para importá-los no S3 ou no EBS.
- **Backup externo:** envie backups completos ou incrementais para o S3 ou para o EBS para armazenamento externo confiável e redundante.
- **Recuperação de desastres:** caso seja necessário recuperar rapidamente um grande backup armazenado no S3, use o AWS Import/Export para transferir os dados para um dispositivo de armazenamento portátil e enviá-lo ao seu site.

## 6.8. Glacier

### 6.8.1. Introdução

O Amazon Glacier é um serviço de custo baixo que fornece armazenamento seguro e durável para backup e principalmente arquivamento de dados. Para manter os custos baixos, o Glacier é otimizado para dados que raramente são acessados e para os quais tempos de recuperação de várias horas são adequados. Com o Glacier, os clientes podem armazenar com segurança grandes ou pequenas quantidades de dados por apenas US\$ 0,01 por gigabyte, por mês, o que representa uma economia significativa em comparação a soluções locais.

As empresas geralmente pagam além do necessário pelo arquivamento de dados. Normalmente elas são forçadas a fazer um alto pagamento inicial por sua solução de arquivamento (que não inclui o custo contínuo de despesas operacionais, como energia, instalações, equipe e manutenção). Depois precisam “adivinhar” quais serão seus requisitos de capacidade – assim elas acabam provisionando armazenamento para mais, a fim de se certificar de que terão capacidade suficiente para redundância de dados e crescimento inesperado. Esse conjunto de circunstâncias resulta em capacidade não utilizada e dinheiro desperdiçado. Com o Glacier, paga-se somente pelo que é usado.

### 6.8.2. Funcionamento

Os dados no Glacier são armazenados como arquivos. Um arquivo pode representar um único arquivo ou pode combinar vários arquivos a serem carregados como um único arquivo. Os arquivos são organizados em *vaults* (cofres).

A **Figura 6-12** ilustra a ideia de organizar os arquivos em *vaults* (cofres).

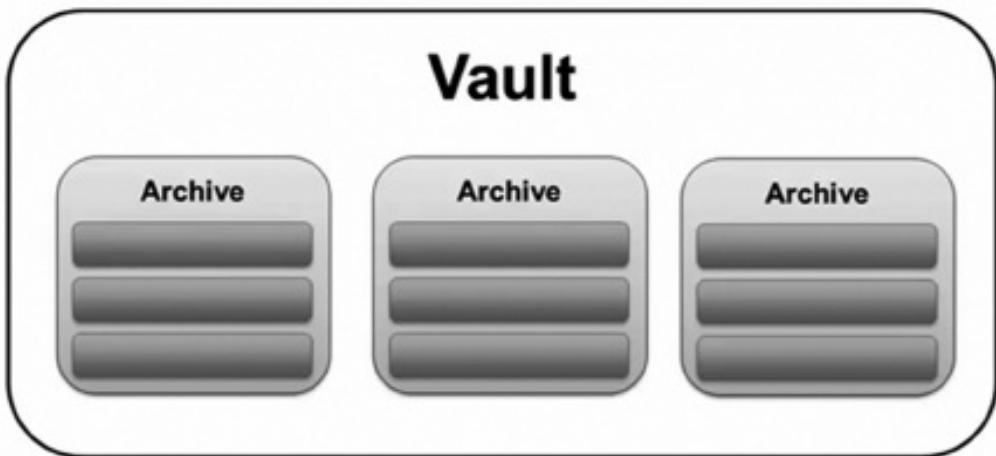


Figura 6-12 Vault

Recuperar arquivos do Glacier requer a iniciação de um *job*. É possível controlar o acesso aos cofres com o IAM. Os *jobs* geralmente são concluídos entre três horas e meia e quatro horas e meia.

Para usar o Amazon Glacier, simplesmente:

- Use as APIs do Glacier ou o console de gerenciamento para criar cofres. Os cofres organizam os arquivos a serem enviados para o Glacier.
- Use as APIs do Glacier para carregar e recuperar arquivos.
- Monitore o status dos *jobs* usando as APIs do Glacier. Opcionalmente, você também pode configurar seu cofre para enviar para si mesmo uma notificação através do SNS quando os trabalhos forem concluídos.
- Pague somente por aquilo que usar. Sua conta mensal será baseada na quantidade de dados armazenados e transferidos.

A **Figura 6-13** ilustra o funcionamento do Glacier. Os dados trafegam entre o DATACENTER corporativo e o Glacier na AWS.

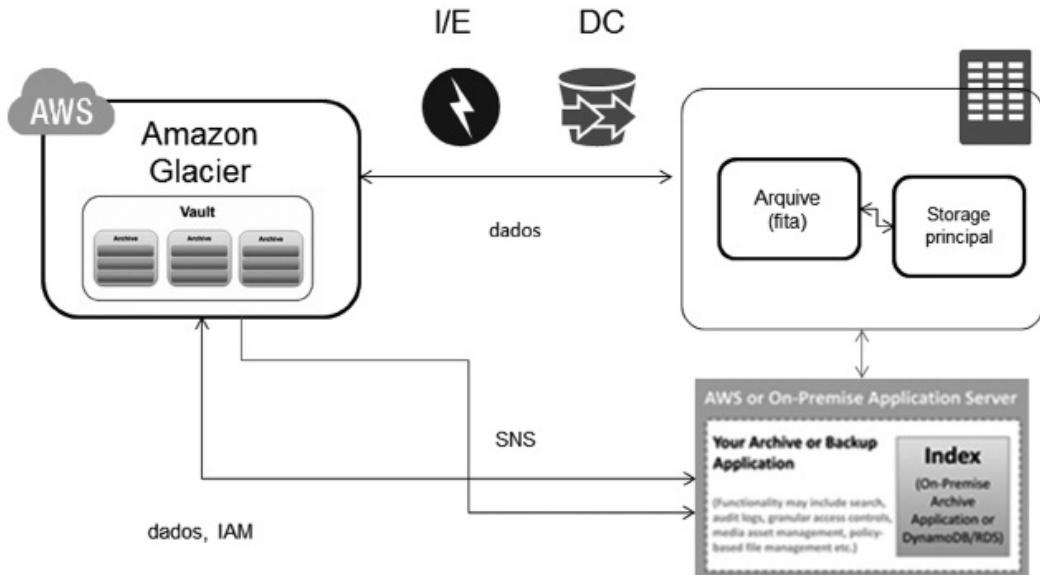


Figura 6-13 Glacier

### 6.8.3. Casos de uso

Sobre possíveis utilizações do Glacier a Amazon identificou os seguintes casos de uso:

- **Substituição da fita magnética:** o Glacier pode substituir bibliotecas de fita. Embora o armazenamento baseado em fita magnética possa ser econômico quando operado em escala, ele pode “sugar” recursos, pois uma ou mais bibliotecas de fita precisam ser mantidas (com frequência em locais distintos geograficamente), exigindo equipe especializada e ocupando espaço valioso em DATACENTERS. Além disso, as fitas por si só devem ser cuidadosamente armazenadas e gerenciadas, o que pode incluir cópia periódica de dados de fitas antigas para novas, a fim de garantir que seus dados ainda possam ser lidos à medida que os padrões de tecnologia de fita evoluem. Substituir a biblioteca de fita pelo Glacier elimina a carga de gerenciar esses desafios operacionais. Conjuntos inteiros de dados podem ser movidos de bibliotecas de fita para o Glacier, um processo que pode ser economicamente acelerado pelo uso do AWS Import/Export.
- **Arquivamento externo de informações empresariais:** as organizações estão arquivando cada vez mais dados em virtude de necessidades normativas e comerciais, bem como devido à quantidade cada vez maior de dados que produzem. O Glacier permite armazenar dados empresariais externamente, com economia e segurança, tornando simples, barato e seguro manter dados arquivados pelo tempo que desejar. O custo baixo de armazenamento do serviço permite manter os dados que podem ser

importantes no futuro, mas que poderiam ser descartados para reduzir custos ou liberar espaço para dados adicionais.

- **Arquivamento de ativos de mídia:** os principais ativos de empresas de mídia são seu conteúdo, que inclui livros, filmes, músicas, imagens, filmagem de notícias e programas de TV. Armazenar esses ativos com segurança e proteção é de extrema importância. A acessibilidade aos dados também é essencial. Tradicionalmente, o arquivamento de mídia tem exigido cofres externos e DATACENTERS de dados redundantes, em vários locais e de alto custo. O Glacier reduz o custo de armazenamento desses ativos, aumentando ao mesmo tempo a durabilidade, a facilidade de uso e a acessibilidade do conteúdo. Os clientes não precisam se preocupar em transportar a mídia de armazenamento de instalações externas para restaurar dados.
- **Arquivamento de dados científicos:** organizações científicas e de pesquisa têm necessidades de arquivamento de grande volume de dados. Tradicionalmente, esses dados eram armazenados em sistemas inflexíveis com base em fita, com cópias mantidas em vários locais, e, com frequência, com uma cópia em um cofre externo também. O Glacier reduz o custo de armazenar esses conjuntos de dados ao eliminar a sobrecarga operacional envolvida no gerenciamento de hardware e DATACENTERS. O serviço reduz automaticamente dados redundantes em várias instalações e em vários dispositivos em cada instalação. Além disso, foi criado para realizar recuperação automática, executando verificações regulares e sistemáticas de integridade de dados e usando dados redundantes para realizar reparos automáticos se erros forem detectados.
- **Preservação digital:** os responsáveis pela preservação digital estão cada vez mais aumentando seus esforços para preservar conteúdo digital antigo, porém valioso, como sites, código-fonte de software, conteúdo gerado por usuário e outros itens digitais que não estão mais prontamente disponíveis. Esses volumes de arquivo podem começar pequenos, mas podem aumentar para petabytes ao longo do tempo. O Glacier torna o armazenamento econômico, extremamente durável e acessível para volumes de dados de qualquer porte. Isso contrasta com soluções tradicionais de arquivamento de dados que exigem planejamento de capacidade preciso e uso em grande escala para ser econômico.

#### 6.8.4. Utilitários de terceiros

O IceBox (<https://www.iceboxpro.com/>) elimina a necessidade de instalar um cliente Glacier separado. O serviço linka o Amazon Glacier com sua conta Dropbox e qualquer arquivo adicionado a uma pasta no Dropbox será

automaticamente enviado a um cofre Glacier.

O FastGlacier (<http://fastglacier.com/>) é um cliente freeware Windows que permite fazer o upload de arquivos para o Glacier usando a largura de banda disponível. Com o FastGlacier pode-se também baixar arquivos do Glacier e gerenciar os cofres de forma fácil.

O WinGlacier (<http://winglacier.com/>) é um utilitário que permite comprimir os arquivos antes da realização do upload. O WinGlacier também oferece validação de *checksum*, sincronização automática de informações de arquivo, criptografia de senha e múltiplas transferências de arquivo simultâneas.

### 6.8.5. Armazenamento com base em políticas

É possível utilizar o Glacier como uma opção de armazenamento do S3 com a criação de regras de ciclo de vida dos objetos do S3 no console de gerenciamento. Em um certo momento esses objetos podem ser transferidos para o Glacier de forma automática. Depois os objetos podem ser restaurados em uma data definida. A **Figura 6-14** ilustra a utilização do Glacier com o S3.

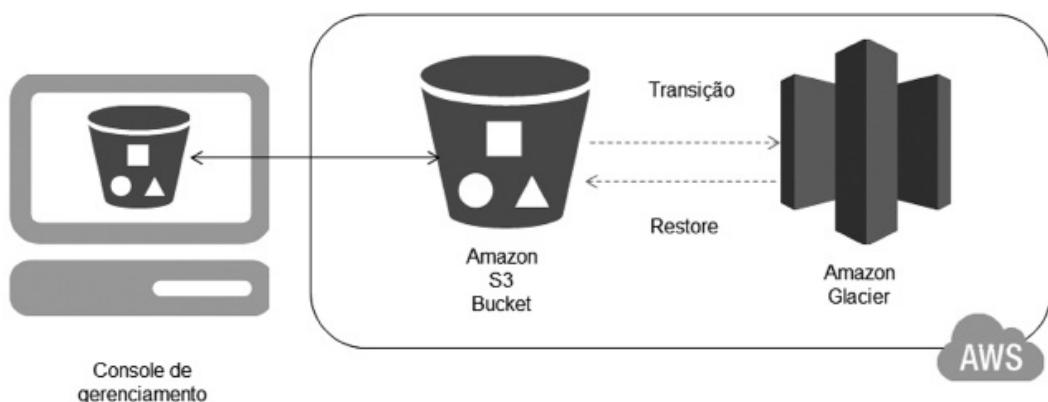


Figura 6-14 Glacier e S3

A opção *Lifecycle Rule* (regra de ciclo de vida) obtida acessando as propriedades de um *bucket* é mostrada na **Figura 6-15**.

A regra de ciclo de vida para agendar o arquivamento e/ou a remoção de objetos será aplicável a todos os objetos contidos dentro do prefixo especificado. É possível configurar o arquivo e a agenda de vencimento de duas maneiras: expressar o período de tempo após a ação ou arquivar a ação de vencimento. O período de tempo pode ser definido como: 1) o número de dias a partir da data de criação do objeto; ou 2) simplesmente como uma data específica.



Figura 6-15 Lifecycle rule

A classe de storage no S3 de um objeto que foi arquivado no Glacier será definido como GLACIER.

São três os valores possíveis no S3:

- **STANDARD**: durabilidade de 99,99999999%. É a opção de storage default do S3.
- **RRS**: durabilidade de 99,99%. Opção *Reduced Redundancy Storage* do S3.
- **GLACIER**: durabilidade de 99,99999999%. Opção de objeto arquivado no Glacier.

É possível recuperar os objetos arquivados no Glacier através do console de gerenciamento ou por meio das APIs da AWS.

## 6.9. Referências bibliográficas

Amazon Web Services. **Amazon Simple Storage Service**: user guide. API Version 2006-03-01.

Amazon Web Services. **Overview of Amazon Web Services**. December 2010.

Amazon Web Services. **Storage Options in the AWS Cloud**: Use Cases. December 2010.

<http://aws.amazon.com/pt/ebs/>

<http://aws.amazon.com/pt/ec2/faqs/>

<http://aws.amazon.com/pt/glacier/>

<http://aws.amazon.com/pt/s3/>

<http://aws.amazon.com/pt/s3/faqs/>

# 7. Rede

## 7.1. Introdução

As principais opções de rede fornecidas pela AWS são a rede privada virtual (*Virtual Private Cloud* – VPC) e o sistema de nomes de domínios (*Domain Name System* – DNS) Route 53.

A rede privada virtual VPC é um web service que permite criar uma seção isolada de rede que inicia recursos da AWS em uma rede privada virtual. Esta é uma opção que reforça a segurança de uma solução projetada para utilizar os recursos da AWS. Vale ressaltar que a AWS essencialmente é uma arquitetura para nuvem pública, e segurança é uma questão chave.

O Route 53 é um web service do tipo DNS com alto desempenho, escalabilidade e baixa latência que dá aos desenvolvedores um caminho para rotear usuários finais para aplicações de internet fazendo a tradução de nomes para endereços IP. O Route 53 utiliza uma rede global mantida pela AWS, conforme visto no capítulo 2.

Tanto a rede privada virtual (VPC) como o sistema de nomes de domínios (Route 53) são pagos conforme o uso.

## 7.2. Rede tradicional *versus* rede na AWS

Tradicionalmente, a rede interna do DATACENTER tem sido fonte de preocupação dos projetistas de DATACENTER. A estrutura em camadas a ser criada, ilustrada na **Figura 7-1**, implica em altos investimentos em componentes de rede e de segurança, incluindo *firewall* e sistemas de detecção de intrusão. Switches de acesso, agregação e core precisam ser corretamente dimensionados para evitar gargalos de funcionamento. Switches de acesso precisam agora lidar com a virtualização e switches virtuais que precisam manter políticas de rede em ambientes dinâmicos. Máquinas virtuais podem precisar rodar em servidores diferentes por questões de平衡amento de carga ou mesmo de disponibilidade. Switches de agregação precisam ser cuidadosamente projetados, e decisões sobre a forma de integração com serviços de rede, switches de acesso, disposição física nos racks e tipo de cabeamento podem ser complexas. Switches core representam um alto investimento e, para complicar, normalmente são adquiridos aos pares para efeito de redundância, podendo mesmo assim ser um gargalo com o aumento repentino do tráfego em muitos negócios.

A grande dificuldade é prever o tamanho necessário dos componentes em função das dificuldades de previsão de tráfego. Muitas vezes estes

componentes precisam ser atualizados em menos de dois anos de funcionamento do DATACENTER. Investimentos em segurança também são altos e muitas vezes replicados em outras instalações.

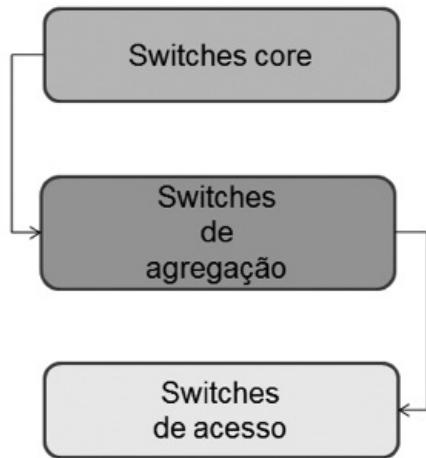


Figura 7-1 Rede em camadas no DATACENTER

A rede AWS muda todo este contexto. Toda a construção da estrutura em camadas fica por conta da AWS. Basicamente, o cliente da AWS passa a gerenciar e operar aspectos importantes da rede, incluindo a definição de IPs, definição e utilização de instâncias NAT, criação de *subnets*, utilização de listas de controle de acesso, definição de regras de *firewall* e utilização e design da rede privada virtual construída de forma lógica sem investimentos em dispositivos de hardware ou mesmo licenciamento de software. A sobrecarga sobre o projetista de rede diminui de forma acentuada.

A AWS possibilita a utilização de interfaces elásticas de rede que permitem que uma instância possa ter até trinta endereços IP, o que introduz uma grande flexibilidade para ambientes *dual-homed* para servidor web, servidor de aplicação e servidor de banco de dados.

Redes privadas virtuais (VPNs) são construídas normalmente para utilizar a internet como meio de comunicação. O tráfego de dados é levado pela rede pública utilizando protocolos padrão que devem ser seguros, como o IPSec.

Quando uma rede empresarial A quer enviar dados para uma rede empresarial B através de uma VPN, um protocolo como o IPSec, por exemplo, encapsula os dados. Quando esses dados encapsulados chegam à outra extremidade, é feito o desencapsulamento do IPSec e os dados são encaminhados ao referido destino da rede local. A VPN também pode ser acessada remotamente através de um provedor de acesso.

A AWS utiliza recursos de VPN para conexão a redes empresariais e usuários remotos.

A AWS também fornece uma opção de serviços DNS escalável e de fácil utilização que emprega rede de servidores ao redor do mundo, conforme mostrado no capítulo 2.

## 7.3. Virtual Private Connect (VPC)

### 7.3.1. Introdução

Com a VPC é possível definir uma topologia de rede virtual para os recursos EC2 equivalente a uma rede tradicional interna segura de um DATACENTER. A **Figura 7-2** ilustra o aumento da segurança de rede com a utilização de recursos aprimorados na AWS, incluindo a própria rede AWS, rede privada virtual VPC e *subnets*.

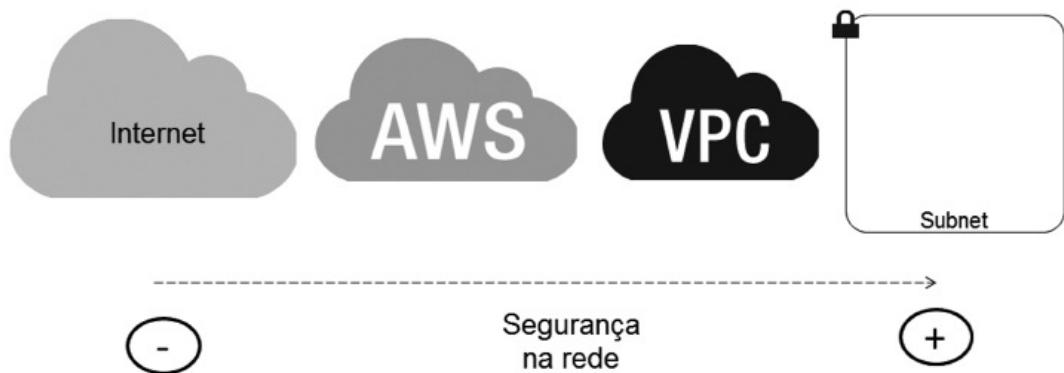


Figura 7-2 Segurança de rede com a AWS

Da mesma forma que em uma rede interna, na VPC pode-se ter controle total sobre o ambiente de rede, incluindo a seleção do intervalo de endereços IP, criação de *subnets* e configuração de tabelas de rotas e gateways de rede. Em resumo, a VPC permite criar um DATACENTER isolado dentro da nuvem AWS, sem que nenhuma solução específica de VPN, hardware específico ou DATACENTER físico sejam exigidos. Tudo é providenciado de forma virtual pela própria AWS e pago pelo uso.

A **Figura 7-3** ilustra o DATACENTER baseado na VPC.

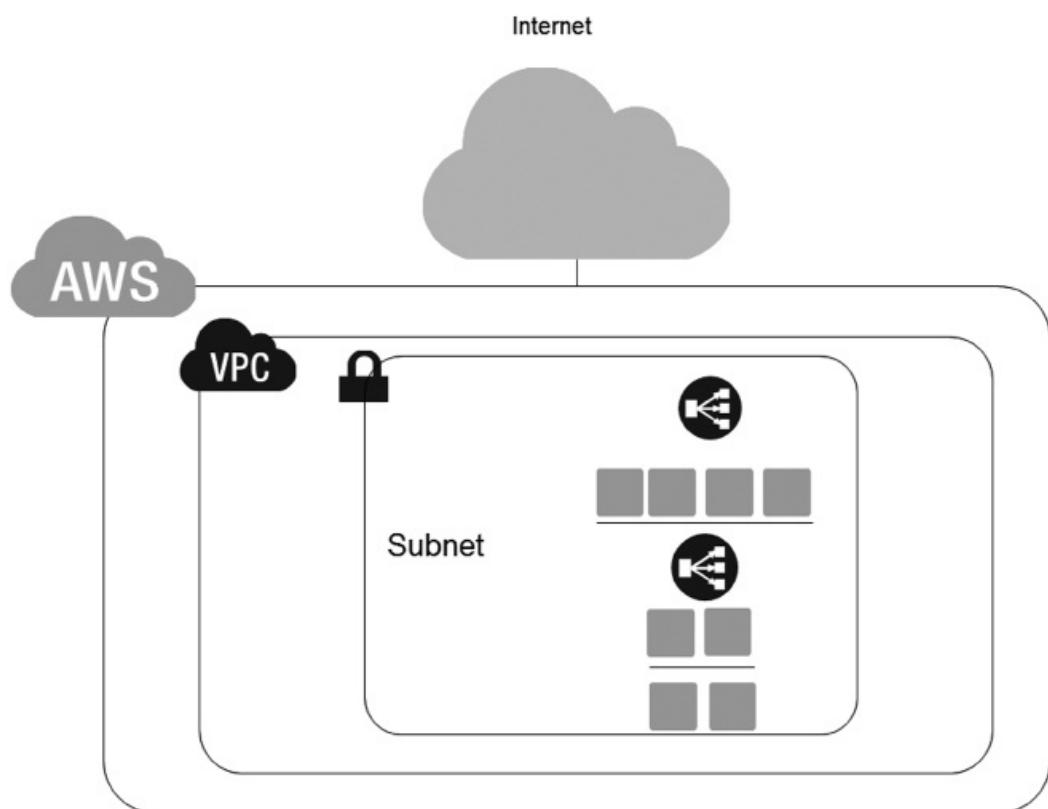


Figura 7-3 DATACENTER na AWS

A VPC fornece recursos avançados de segurança, como grupos de segurança e listas de controle de acesso de rede (ACLs), realizando filtragem

de entrada e de saída no nível da instância e no nível de *subnet*. Além disso, é possível armazenar dados no S3 e restringir o acesso para que estes dados só sejam acessíveis por instâncias na VPC, como se o S3 fosse parte da VPC. É possível também iniciar instâncias dedicadas na VPC que funcionam como um servidor dedicado de um único cliente AWS para isolamento adicional.

As instâncias EC2 recebem um endereço IP interno e um IP externo. Os endereços IP externos fornecem conectividade à internet, enquanto os endereços IP internos permitem que as instâncias comuniquem-se entre si. Na VPC, as instâncias têm somente um endereço IP privado. Para que a instância acesse a internet, deve-se designar um endereço do tipo EIP, que faz o papel do endereço IP externo.

É possível também que instâncias internas na VPC acessem a internet sem um endereço EIP de duas formas:

- As instâncias sem EIPs podem direcionar seu tráfego de internet por meio de uma instância NAT. Essas instâncias sem EIPs usam o EIP da instância NAT para acessar a internet. A instância NAT permite a comunicação externa, mas não permite que máquinas na internet iniciem uma conexão com as máquinas endereçadas privatamente usando a instância NAT.
- As instâncias sem EIPs podem direcionar seu tráfego de internet para o VPN gateway do DATACENTER existente (conexão hardware VPN). Lá elas podem acessar a internet por meio dos pontos de saída existentes e dos dispositivos de segurança/monitoramento de rede.

A primeira coisa a fazer quando se cria uma VPC é estabelecer a faixa de endereços IP a ser utilizada. Esta escolha deve ser baseada na forma de um bloco CIDR (*Classless Inter-Domain Routing*). O tamanho do bloco da VPC pode ir de dezesseis a 65.536 endereços IP.

CIDR, definido no RFC 1519, utiliza máscaras de *subnet* de comprimento variável (*Variable Length Subnet Masks* – VLSM) para alocar endereços IP em *subnets* de acordo com as necessidades individuais e não baseado nas regras de uso generalizado em toda a rede. Assim, a divisão de rede/host pode ocorrer em qualquer fronteira de bits no endereço. Porque as distinções de classes normais são ignoradas, o novo sistema foi chamado de *routing* sem classe, acarretando com isso que o sistema original passasse a ser chamado de *routing* de classes.

A notação *standard* para o intervalo de endereços CIDR começa com o endereço de rede (na direita com o número apropriado de bits com valor zero – até quatro octetos para IPv4 e até campos hexadecimais de oito octetos de 16 bits para IPv6). Isto é seguido por um caractere e comprimento de um prefixo, em bits, definindo o tamanho da rede em questão (o prefixo é, na verdade, o comprimento da máscara de *subnet*).

Por exemplo:

- 192.168.0.0 /24 representa os 256 endereços IPv4 de 192.168.0.0 até 192.168.0.255 inclusive, com 192.168.0.255 sendo o endereço de *broadcast* para a rede.
- 192.168.0.0 /22 representa os 1.024 endereços IPv4 de 192.168.0.0 até 192.168.3.255 inclusive, com 192.168.3.255 sendo o endereço de *broadcast* para a rede.

Atualmente não é possível alterar o tamanho de uma VPC ou das suas *subnets*. Se uma VPC for criada muito pequena e for necessário aumentá-la, deverão ser encerradas todas as instâncias que rodam dentro da VPC, excluir a VPC e seus componentes e, em seguida, criar uma nova e maior VPC.

Se na VPC existe um prefixo de endereço IP que se sobrepõe a um dos prefixos de uma rede interna, qualquer tráfego para o prefixo da rede interna será descartado. Veja o exemplo seguinte:

- Uma VPC com bloco CIDR 10.0.0.0/16.
- Uma *subnet* na VPC com bloco CIDR 10.0.1.0/24.
- Instâncias rodando na *subnet* com endereços IP 10.0.1.4 e 10.0.1.5.
- *Host on-premises* na rede doméstica usando blocos CIDR 10.0.37.0/24 e 10.1.38.0/24.

Quando as instâncias na VPC tentarem “falar” com os *hosts* no espaço de endereço 10.0.37.0/24, o tráfego será descartado porque 10.0.37.0/24 é parte do mesmo prefixo atribuído para a VPC (10.0.0.0/16). As instâncias podem “falar” com *hosts* no espaço de 10.1.38.0/24 porque esse bloco não é parte do 10.0.0.0/16. Portanto, recomenda-se criar uma VPC com um intervalo CIDR grande o suficiente para o crescimento futuro esperado, mas não um que se sobreponha às atuais ou futuras *subnets* da sua rede doméstica.

### 7.3.2. Componentes da rede VPC

A **Figura 7-4** ilustra os componentes de uma rede VPC.

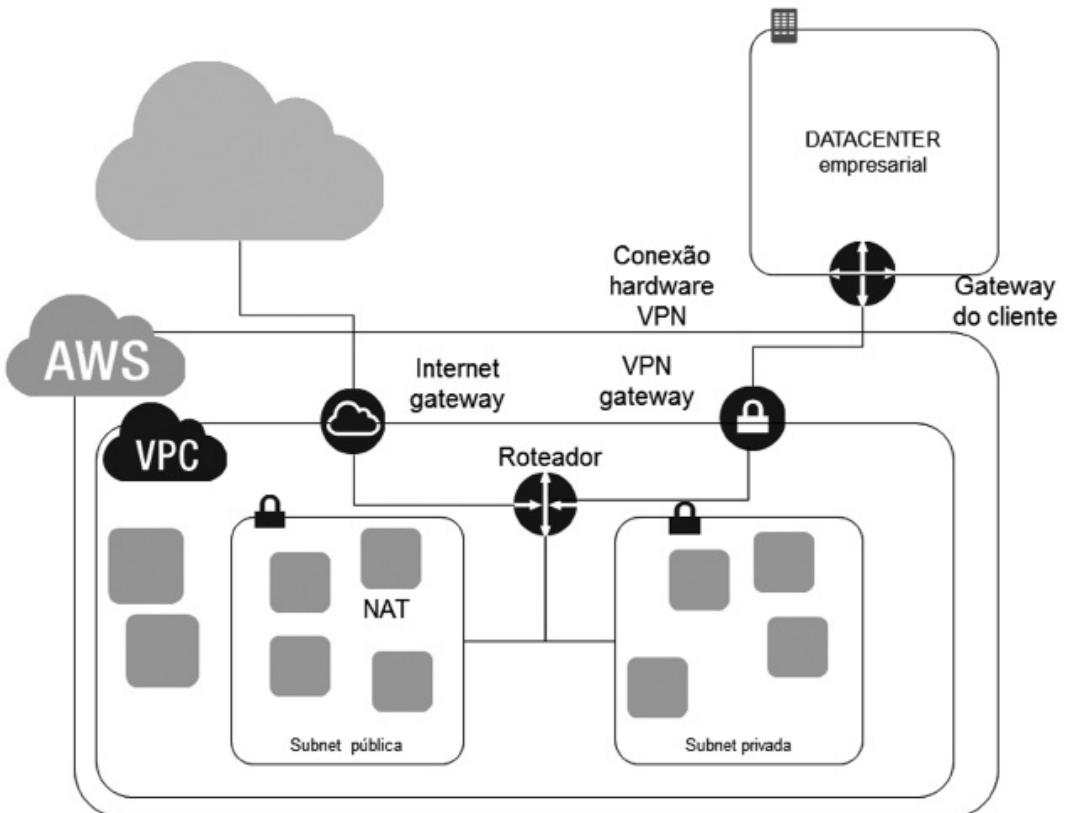


Figura 7-4 VPC e componentes

Os componentes são descritos na **Tabela 7-1**.

**Tabela 7-1 Componentes de uma rede VPC**

### Componentes Descrição

<i>Subnet</i>	Segmento com intervalo de endereços IP da VPC onde é possível disponibilizar grupos de recursos isolados.
Internet Gateway	Lado da VPC de uma conexão à internet pública.
VPN Gateway	Lado da VPC de uma conexão VPN.
Gateway do cliente	Lado do cliente de uma conexão VPN.
Instância NAT	Instância EC2 que fornece conversão de endereços de porta para instâncias não EIP para acessar a internet via Internet Gateway.
Conexão hardware VPN	Conexão VPN entre a VPC e o DATACENTER corporativo. É baseada em hardware.
Roteadores	Dispositivos que interconectam subnets e direcionam o tráfego entre Internet Gateways, gateways de VPN, instâncias NAT e subnets.

### 7.3.3. Subnets

Configurar uma rede VPC de forma correta é uma tarefa fundamental para usufruir da plataforma AWS. A criação de *subnets* é parte importante desta tarefa. Por exemplo, pode-se criar uma *subnet* voltada para o público contendo servidores web que tenham acesso à internet e colocar sistemas *back-end*, como bancos de dados e servidores de aplicação, em outra *subnet* de uso privado sem acesso à internet. É possível inclusive utilizar várias camadas de segurança para ajudar a controlar o acesso aos servidores (instâncias) em cada *subnet*. Os recursos da VPC permitem aumentar a segurança global da solução AWS utilizada, conforme dito anteriormente.

Pode-se também criar uma VPC que abrange várias zonas de disponibilidade, onde cada *subnet* reside exatamente em uma zona de disponibilidade. Através do lançamento de instâncias em zonas de disponibilidade separadas, os aplicativos são protegidos de falha em um único local. A segurança será tratada especificamente no capítulo 13.

A **Figura 7-5** mostra uma VPC que foi configurada com três *subnets* em três zonas de disponibilidade.

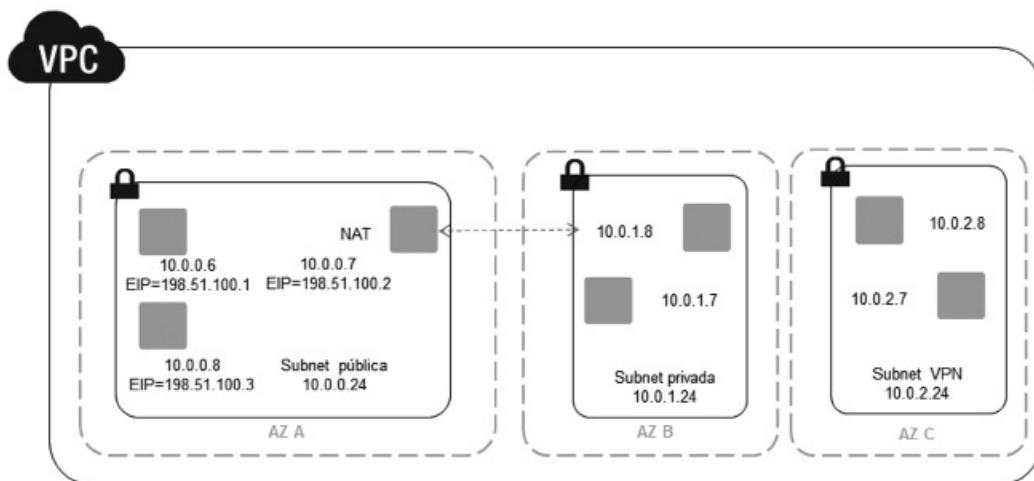


Figura 7-5 VPC com múltiplas AZs

Ao criar cada *subnet*, deve-se fornecer o ID da VPC, a zona de disponibilidade e o bloco CIDR para a *subnet*. O bloco CIDR da *subnet* pode ser o mesmo que o CIDR da VPC (supondo que se queira apenas uma única *subnet* na VPC), ou um subconjunto do CIDR da VPC, se forem necessárias mais de uma *subnet*. Se for criada mais de uma *subnet* em uma VPC, os blocos de CIDR das *subnets* não devem se sobrepor.

Para uma VPC com bloco CIDR 10.0.0.0/24, que fornece 256 endereços, por exemplo, pode-se dividir o bloco CIDR da VPC em duas *subnets*, o que significa que cada uma tem 128 endereços. Uma *subnet* usa o bloco CIDR 10.0.0.0/25 (para endereços 10.0.0.0 – 10.0.0.127) e a outra usa 10.0.0.128/25

de bloco CIDR (para endereços 10.0.0.128 – 10.0.0.255). É importante ressaltar que a AWS reserva os primeiros quatro endereços IP e o último IP em um bloco CIDR; portanto, esses IPs não estão disponíveis para uso.

Por projeto, cada *subnet* deve ter associada uma tabela de rota principal, que especifica as rotas permitidas para o tráfego saindo da *subnet*. Cada nova *subnet* criada é associada automaticamente à tabela de rota principal da VPC.

As *subnets* podem ser públicas, privadas ou somente para uso com VPN, conforme explicado a seguir.

- **Subnet pública:** significa que o tráfego da *subnet* é roteado para o gateway de internet. Pode-se determinar se uma *subnet* é pública observando a tabela de rota associada com a *subnet*. A tabela de rota inclui a rota mostrada a seguir. O destino 0.0.0.0/0 significa todo o tráfego e o *target* é a identificação para o internet gateway.

Destino	Target
0.0.0.0/0	igw-xxxxxxxx

- **Subnet privada:** não tem uma rota para o gateway de internet. Em vez disso, seu tráfego ligado à internet será roteado para uma instância NAT em uma *subnet* pública. A tabela de rota é mostrada a seguir. O *target* é a ID da instância NAT.

Destino	Target
0.0.0.0/0	i-xxxxxxxx

- **Subnet VPN:** é somente para conexão a uma VPN; ele não tem uma rota para o internet gateway. Em vez disso, todo o seu tráfego é roteado para o gateway privado virtual. A tabela de rota é mostrada a seguir. O target é o ID do gateway virtual privado.

Destino	Target
0.0.0.0/0	vgw-xxxxxxxx

Por default, cada *subnet* deve ser associada a um ACL de rede, que propicia segurança no nível de *subnet* para as instâncias.

Quando se adiciona uma nova *subnet* à VPC, deve-se configurar o roteamento e qualquer segurança adicional para a *subnet*. Por padrão, a *subnet* é associada ao ACL da VPC, mas é possível definir um ACL exclusivo para a *subnet*. É possível excluir a VPC a qualquer momento.

### 7.3.4. Roteamento

Quando uma VPC é criada, ela automaticamente possui uma tabela de rota principal modificável. Podem ser criadas outras tabelas de rota na mesma VPC. Uma *subnet* deve ser associada com uma tabela de rota; se nada for configurado, a *subnet* obedece à rota principal.

Inicialmente, a tabela de rota principal contém apenas uma única rota: uma rota local que permite a comunicação dentro da VPC. O diagrama da **Figura 7-6** mostra uma VPC inicial com uma tabela de rota principal.

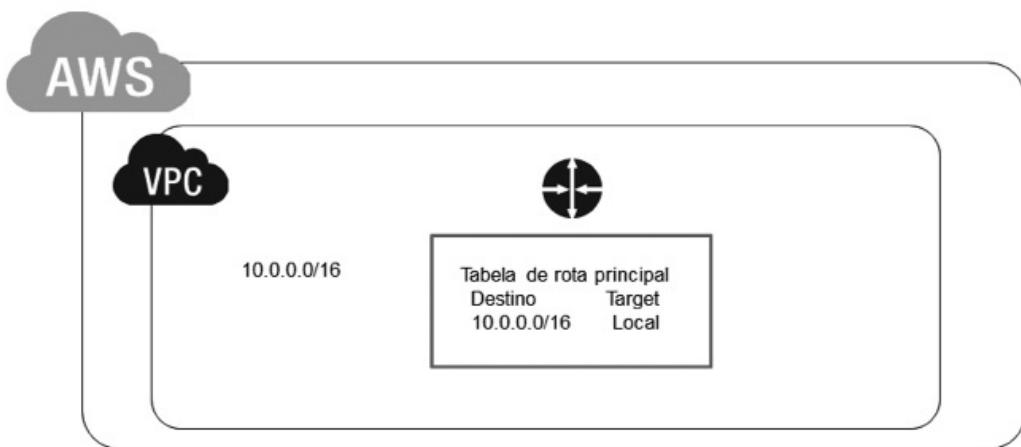


Figura 7-6 VPC e tabela de rota

Não é possível modificar a rota local em uma tabela de rota principal. Sempre que uma instância for iniciada na VPC, a rota local cobre automaticamente essa instância; não é necessário adicionar a nova instância a uma nova tabela de rota, pois isto é feito automaticamente.

Cada *subnet* na VPC deve ser associada a uma tabela de rota; a tabela controla o roteamento para toda a *subnet*. Várias *subnets* podem ser associadas com a mesma tabela de rota, mas uma *subnet* pode ser associada com a tabela apenas em uma rota.

Quando se adiciona um gateway para a VPC (quer seja um gateway de internet ou um gateway VPN), deve ser feita a atualização da tabela de rota para as *subnets* que precisam utilizar o gateway.

Na **Figura 7-7** foi adicionado um gateway VPN e uma *subnet* que precisa utilizar o gateway. A *subnet* utiliza a tabela de rota principal por padrão, assim, deve-se adicionar uma rota para a tabela de rota principal que vai rotear todo o tráfego para o gateway VPN.

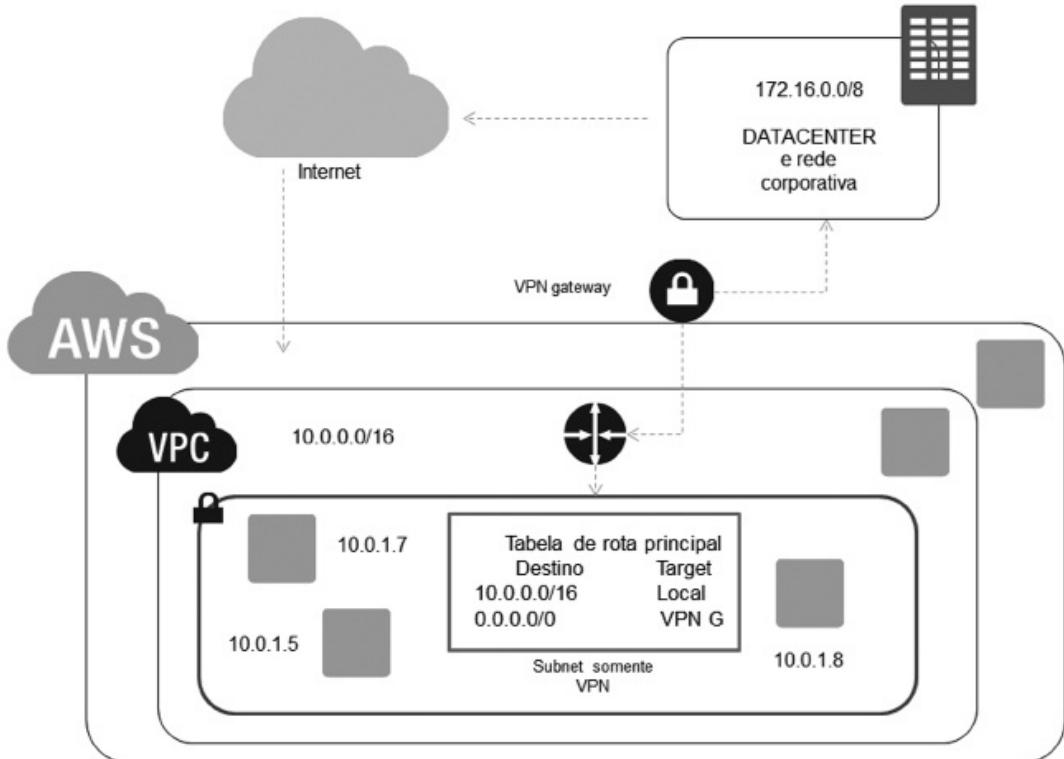


Figura 7-7 Subnet e tabela de rota modificada

É possível adicionar até dez conexões VPN para uma única VPC. Isso permite estabelecer conexões VPN para cada uma das filiais de uma empresa, por exemplo. Além disso, várias conexões VPN fornecem redundância. É possível configurar um segundo gateway de cliente na mesma rede física com o primeiro gateway do cliente. Se um gateway do cliente precisa ser desativado para manutenção, o tráfego continua a fluir entre a rede corporativa e a VPC pelo segundo gateway de cliente. A Amazon reforça que é interessante ter um procedimento de teste antes de substituir a tabela de rota principal.

É possível fazer a comunicação entre múltiplas VPCs utilizando a internet ou mesmo gateways de rede privada.

A **Figura 7-8** ilustra esta possibilidade para VPCs que não falam diretamente via AWS.

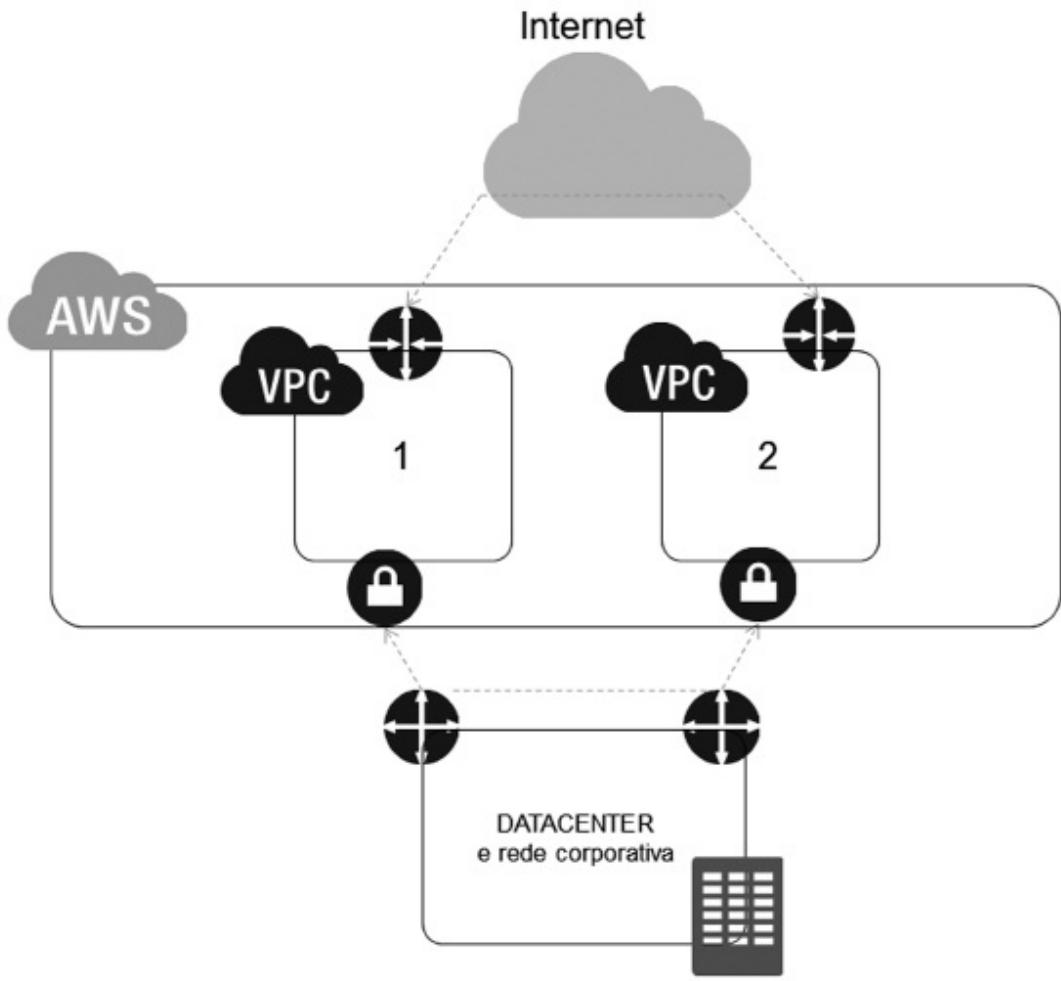


Figura 7-8 Comunicação entre múltiplas VPCs

### 7.3.5. Funcionalidades

#### 7.3.5.1. Endereços IP elásticos VPC (Elastic IPs VPC – EIPs VPC)

Cada instância VPC tem apenas um endereço IP privado, conforme dito; portanto, se for necessário que a instância se comunique utilizando a internet, pode-se alocar um endereço IP elástico para uso com a VPC e, em seguida, atribuir esse endereço à instância. O endereço IP elástico é um endereço IP estático público que pode ser atribuído a qualquer instância na VPC. Com um endereço IP elástico, pode-se mascarar uma falha de instância reatribuindo rapidamente o endereço a outra instância na VPC. Os endereços IP elásticos utilizados com instâncias EC2 fora da VPC (ou seja, endereços EC2) não estão disponíveis para usar na VPC. Deve-se alocar um conjunto separado de endereços IP elásticos para usar na VPC (ou seja, endereços IP elásticos de VPC).

A seguir estão reforçados aspectos básicos sobre endereços IP elásticos utilizados na VPC:

- Qualquer instância que precisa se comunicar com a internet (ou seja, utilizando o internet gateway) deve ter um endereço IP elástico associado a ela.

- Primeiro aloca-se um endereço IP elástico para a VPC e, em seguida, atribui-se este endereço a uma instância na VPC (ele pode ser atribuído a uma única instância por vez).
- Endereços IP elásticos utilizados em uma VPC são diferentes dos endereços IP utilizados fora da VPC.
- É permitido mover um endereço IP elástico de uma instância para outra em uma VPC, ou para qualquer outra VPC que está sendo executada pelo mesmo cliente, mas não para instâncias fora de uma VPC.
- Quaisquer endereços alocados para a VPC permanecem com a VPC até que sejam explicitamente liberados.
- Uma conta está limitada a ter cinco endereços IP elásticos de VPC; para ajudar a conservá-los, pode-se usar uma instância NAT.

A **Tabela 7-2** lista as diferenças entre endereços IP elásticos na EC2 e endereços IP elásticos na VPC.

**Tabela 7-2 Endereços IP elásticos para EC2 e VPC**

EIP EC2	EIP VPC
Se você alocar um endereço, ele é associado com a sua conta AWS, mas para uso só fora da VPC.	Se você alocar um endereço, ele é associado com a sua conta AWS, mas para uso só dentro da VPC.
Se você tenta associar um endereço que já está associado a outra instância, o endereço é automaticamente associado à nova instância.	Se você tenta associar um endereço que já está associado a outra instância, a requisição falha.
Se você parar uma instância, o endereço IP elástico não é mantido mapeado e você deve remapeá-lo quando reiniciar a instância.	Se você parar uma instância, o endereço IP elástico permanece mapeado.

### 7.3.5.2. Network Address Translation (NAT)

NAT (*Network Address Translation*) é uma técnica que consiste em reescrever os endereços IP de origem de um pacote que passam por um roteador ou *firewall* de maneira que um computador de uma rede interna tenha acesso ao exterior ou à internet (rede pública).

Opcionalmente, pode-se usar uma instância do tipo NAT na VPC se for desejado habilitar instâncias particulares (aqueles com apenas um endereço IP privado em uma *subnet* particular) para permitir o tráfego de saída para a internet e para evitar receber tráfego de entrada iniciado por alguém na internet.

As entradas no NAT são geradas apenas por pedidos dos computadores

de dentro da rede privada. Sendo assim, um pacote que chega ao roteador vindo de fora (e que não tenha sido gerado em resposta a um pedido da rede) não encontrará nenhuma entrada no NAT e este pacote será automaticamente descartado, não sendo entregue a nenhum computador da rede. Isso impossibilita a entrada de conexões indesejadas e o NAT acaba funcionando também como um *firewall*.

A tabela de rota principal aponta o tráfego de instâncias na *subnet* particular para a instância NAT. A instância NAT encaminha o tráfego para o internet gateway, para que a fonte do tráfego pareça ser o endereço IP elástico da instância NAT.

A AWS fornece AMIs (nos formatos de 32-bit e 64-bit) Linux que têm sido especialmente configuradas para execução como instâncias NAT. A AMI inclui a cadeia de caracteres *ami-vpc-nat* em seu nome, portanto, pode-se procurá-las facilmente no console de gerenciamento da AWS. Como alternativa, é possível configurar uma instância qualquer como NAT.

### 7.3.5.3. Interface de rede elástica (*Elastic Network Interface – ENI*)

Cada instância EC2 tem uma interface de rede padrão com um endereço IP privado que permite que ela seja inserida em sua rede VPC. Pode-se criar e anexar uma interface de rede adicional, conhecida como interface de rede elástica (*Elastic Network Interface – ENI*). Pode-se ter até trinta endereços IP por interface e oito interfaces por instâncias do tipo *m2.4xlarge* e *cc2.8xlarge*.

Cada ENI vive dentro uma *subnet* particular dentro da VPC – consequentemente, dentro de uma zona de disponibilidade. Uma consequência muito importante da utilização da ENI é que a ideia de lançar uma instância EC2 em uma *subnet* de uma VPC particular é efetivamente obsoleta. Uma única instância EC2 agora pode ser ligada a duas ou mais ENIs, cada uma em uma *subnet* diferente. A ENI (não a instância) está associada a uma *subnet*.

Similar a um volume EBS, as ENIs têm uma vida útil que é independente de qualquer instância específica EC2. Elas também são verdadeiramente elásticas. As ENIs podem ser criadas antes do tempo e, em seguida, ser associadas a uma instância na hora do lançamento. Pode-se também anexar uma ENI a uma instância enquanto ainda estiver em execução (“anexar a quente”). A menos que a ENI seja excluída, ela permanecerá viva depois que a instância é encerrada.

Algumas possíveis utilizações da ENI são descritas a seguir:

- **Otimização de rede:** pode-se criar um ambiente *dual-homed* para o servidor web, aplicação e servidores de banco de dados. A primeira ENI seria ligada a uma *subnet* pública. A segunda ENI seria ligada a uma *subnet* privada. Poderiam ser aplicados grupos de segurança diferentes para cada ENI.

- **Multi-interface de aplicações:** pode-se hospedar平衡adores de carga, servidores proxy e servidores NAT em uma instância EC2, passando cuidadosamente o tráfego de uma *subnet* para a outra.
- **Licenciamento MAC-based:** se um determinado software comercial estiver sendo executado vinculado a um endereço MAC específico, este pode ser licenciado contra o endereço MAC da ENI. Mais tarde, se for necessário mudar uma instância, ou mesmo o tipo de instância, pode-se lançar uma instância de substituição com a mesma ENI e o mesmo endereço MAC.

#### 7.3.5.4. Instâncias dedicadas (*dedicated instances*)

Se for necessário lançar instâncias fisicamente isoladas no nível do hardware do *host*, é possível utilizar instâncias dedicadas dentro da VPC. Instâncias dedicadas na VPC apresentam mais um nível de segurança e possuem o atributo *tenancy* configurado para dedicado.

Em uma VPC existem instâncias dedicadas e não dedicadas.

#### 7.3.6. Implementação

Existem quatro cenários sugeridos pela Amazon para utilização da VPC e disponibilizados no console de gerenciamento VPC. A caixa de diálogo com os quatro cenários é ilustrada na **Figura 7-9**.

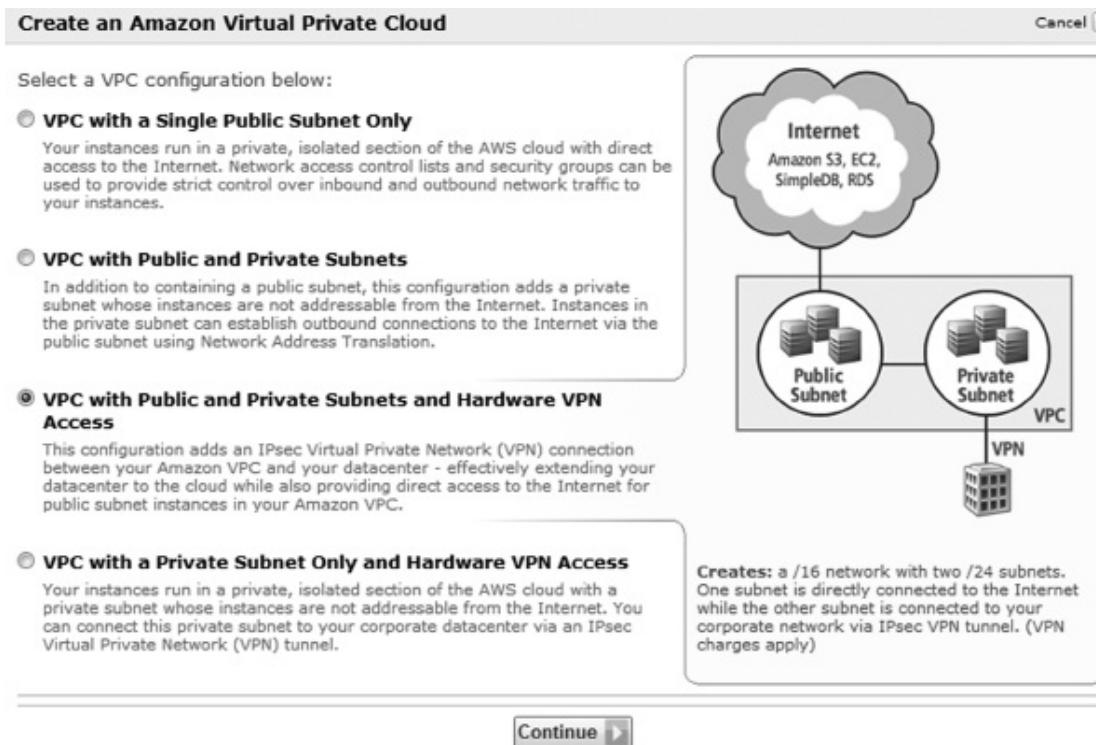


Figura 7-9 Select a VPC configuration

- **Configuração 1:** VPC somente com uma *subnet* pública.

Pode-se hospedar um aplicativo básico de web em uma VPC e usufruir

das camadas adicionais de privacidade e de segurança oferecidas pela VPC. É possível proteger o site web ao criar regras do grupo de segurança que permitam ao servidor web responder a solicitações HTTP e SSL recebidas da internet, enquanto proíbe simultaneamente o servidor web de iniciar conexões de saída com a internet. Pode-se criar uma VPC que suporte este tipo de utilização selecionando “VPC with a Single Public Subnet Only” no console de gerenciamento na opção VPC. A **Figura 7-10** ilustra esta opção.

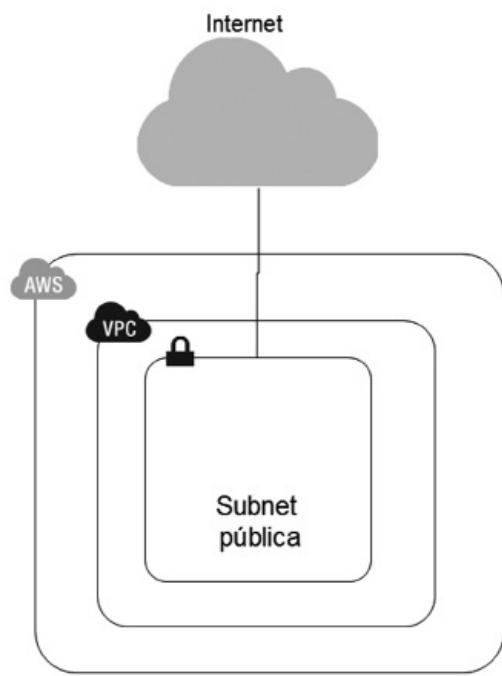


Figura 7-10 VPC com uma única subnet pública

- **Configuração 2:** VPC com *subnets* pública e privada.

Pode-se utilizar a VPC para hospedar aplicativos web multicamadas e estabelecer o acesso e as restrições de segurança entre servidores web, servidores de aplicativos e servidores de banco de dados. É possível iniciar servidores web em uma *subnet* acessível publicamente e servidores de aplicativos e bancos de dados em *subnets* não acessíveis publicamente. Os servidores de aplicativos e banco de dados não podem ser acessados diretamente pela internet, mas ainda podem acessar a web através de uma instância NAT para fazer o download de patches, por exemplo.

É possível controlar o acesso entre os servidores e as *subnets* utilizando filtros de pacote de entrada e de saída fornecidos pelas listas de controle de acesso (ACLs) de rede e grupos de segurança. Para criar uma VPC que aceite esse uso, selecione “VPC with Public and Private Subnets” no console de gerenciamento na opção VPC.

A **Figura 7-11** ilustra esta situação.

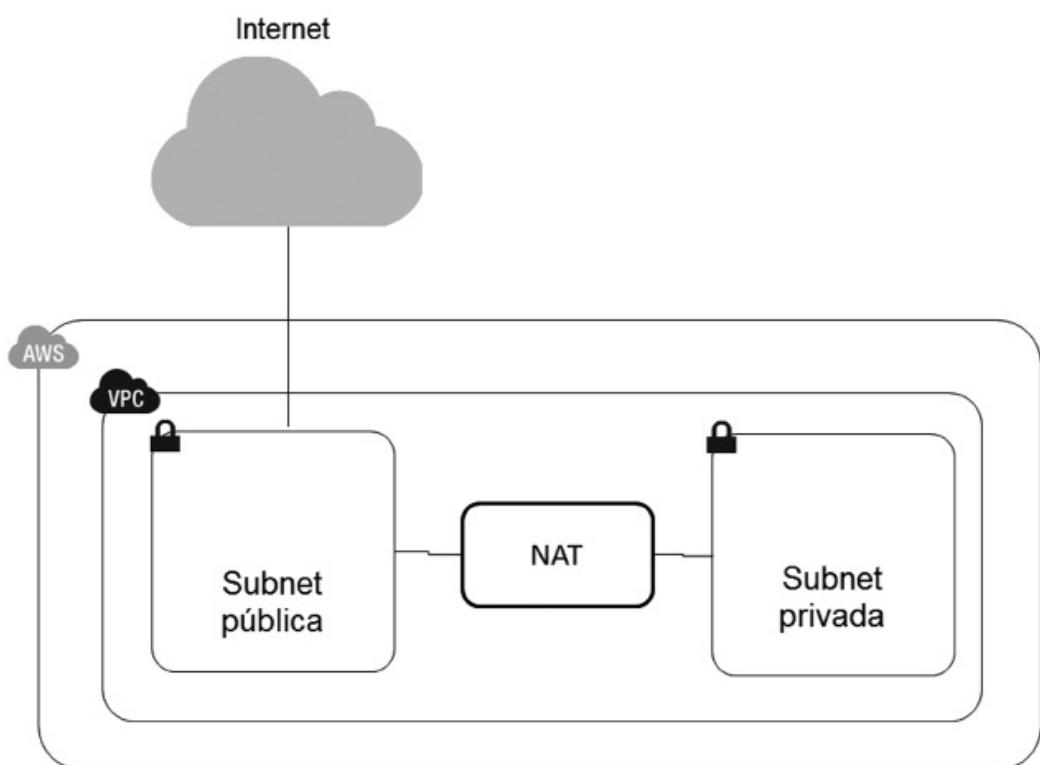


Figura 7-11 VPC com subnets pública e privada

- **Configuração 3:** VPC com *subnets* pública e privada e hardware VPN de acesso.

É possível criar uma VPC onde as instâncias em uma *subnet*, como servidores web, comunicam-se com a internet enquanto outras instâncias em outra *subnet*, como servidores de aplicativos, comunicam-se com banco de dados na rede corporativa.

Uma conexão IPsec VPN entre a VPC e a rede corporativa ajuda a tornar a comunicação segura entre os servidores de aplicativos na nuvem e o servidor de banco de dados no DATACENTER corporativo.

O protocolo de segurança IP (*IP Security Protocol* – IPsec) é uma extensão do protocolo IP considerado padrão para o fornecimento de privacidade de usuário (aumentando a confiabilidade das informações fornecidas pelo usuário para uma localidade da internet), integridade dos dados (garantindo que o mesmo conteúdo que chegou ao seu destino seja o mesmo da origem) e autenticidade das informações, ou mesmo fazendo prevenção de *identity spoofing* (garantia de que uma pessoa é quem diz ser) quando se transferem informações através de redes IP pela internet.

Servidores web e servidores de aplicativos na VPC podem aproveitar os recursos de elasticidade do EC2 para expandir e contrair o pool de instâncias quando for necessário. É possível criar uma VPC para refletir essa situação de uso selecionando “VPC with Public and Private Subnets and Hardware VPN Access” no console de gerenciamento, na opção VPC.

A Amazon recomenda esse cenário para integração do DATACENTER com a nuvem e também para acessar diretamente a internet pela VPC. Esse cenário requer que seja configurado um *appliance* de gateway na rede doméstica para estabelecer uma conexão VPN com a VPC.

A **Figura 7-12** ilustra este cenário.

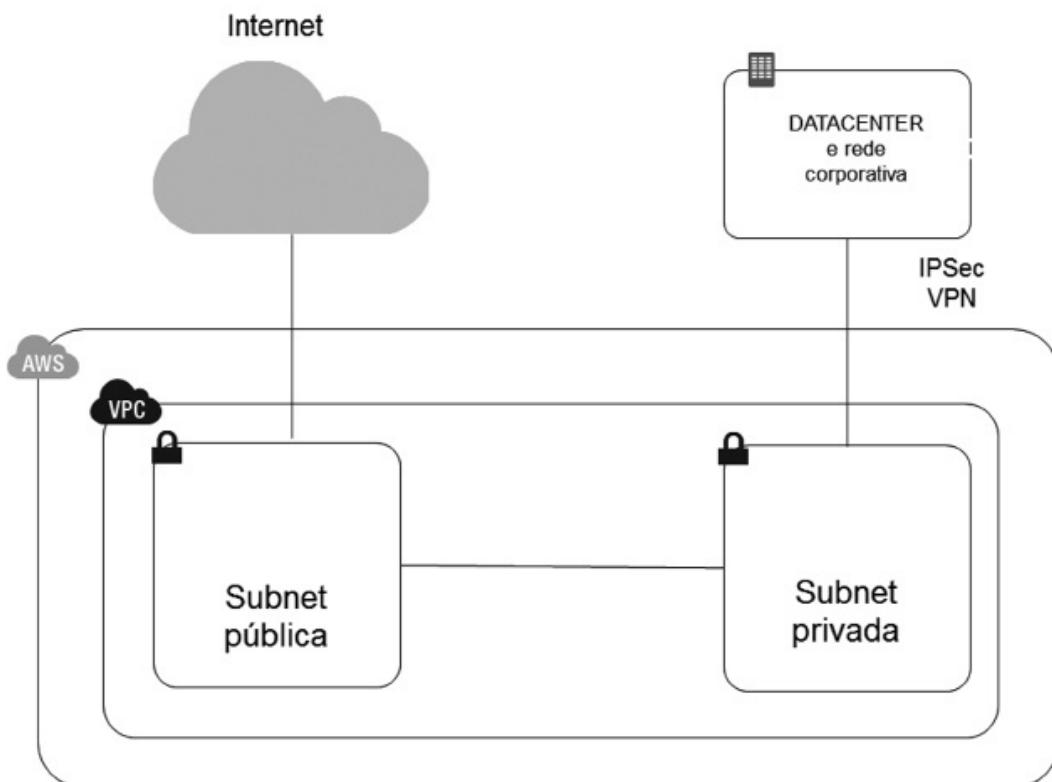


Figura 7-12 VPC com subnets pública e privada e acesso VPN

**Configuração 4:** VPC somente com *subnet* pública e hardware VPN

- de acesso.

É possível mover aplicativos corporativos para a nuvem, iniciar servidores web adicionais ou aumentar a capacidade computacional da rede ao conectar a VPC à rede corporativa. Visto que a VPC pode ser hospedada atrás do *firewall* corporativo, é fácil mover recursos de TI para a nuvem sem alterar a forma com que usuários acessam esses aplicativos.

Selecione “VPC with a Private Subnet Only and Hardware VPN Access” do console de gerenciamento AWS na opção VPC para criar uma VPC que aceite essa situação de uso.

A Amazon recomenda esse cenário para expandir os dados do DATACENTER para a nuvem e aproveitar a elasticidade da AWS sem expor a rede corporativa à internet. Esse cenário requer que seja configurado um *appliance* de gateway na rede doméstica para estabelecer uma conexão VPN com a VPC.

A **Figura 7-13** ilustra este cenário.

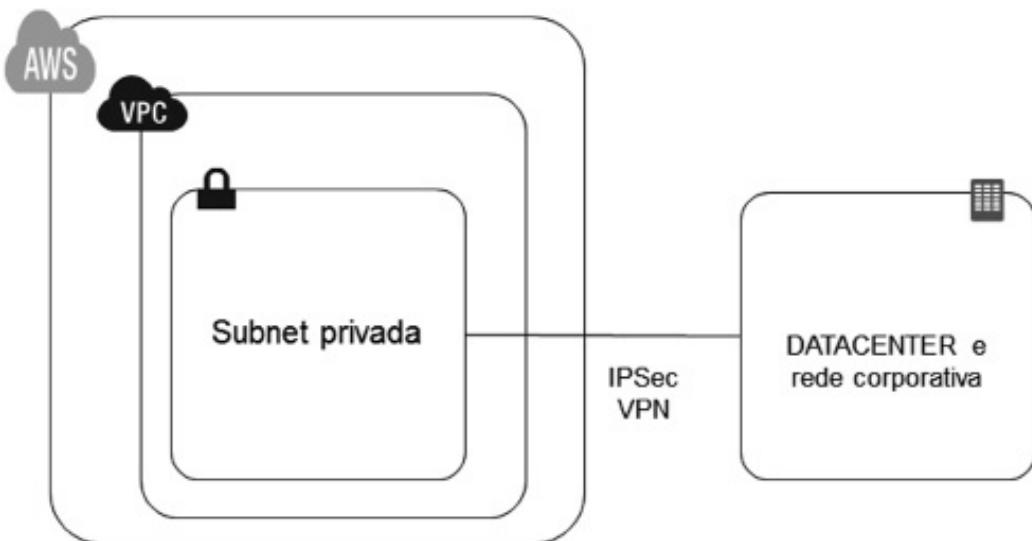


Figura 7-13 VPN com subnet privada e acesso VPN

### 7.3.7. Controle

Atualmente não é possível utilizar o IAM para limitar o acesso do usuário a um recurso específico EC2 ou um recurso VPC. Pode-se limitar o acesso de usuários somente com uma ação individual de API. O IAM se aplica, por exemplo, a todas as instâncias ou a um grupo de segurança.

### 7.3.8. VPC na prática

A configuração 1 mostrada anteriormente é explorada a seguir e permite configurar uma VPC com uma única *subnet* pública que contém uma instância em execução utilizando um endereço elástico EIP – tudo isto com o console de

gerenciamento AWS. Este exemplo é baseado no guia “Amazon Virtual Private Cloud Getting Started Guide, API Version 2011-07-15”.

As tarefas descritas a seguir utilizam o diagrama da **Figura 7-14** como referência e orientam a criar uma VPC na AWS.

- Utilize a opção VPC no console de gerenciamento. Escolha a opção “VPC with a Single Public Subnet Only”. Verifique se o botão está marcado para a primeira opção na lista e clique para continuar.

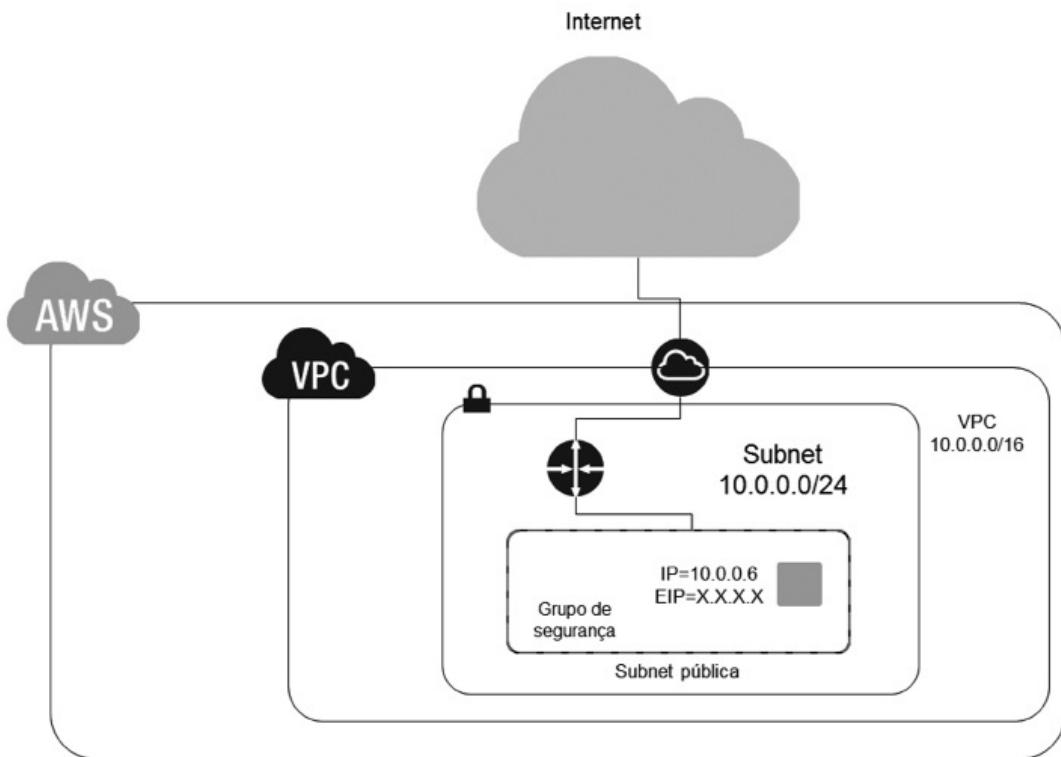


Figura 7-14 Diagrama para apoio na criação da VPC

### 7.3.8.1. Criar uma VPC e um internet gateway

Uma caixa de diálogo será exibida mostrando o intervalo CIDR que será utilizado por padrão para a sua VPC e *subnet* (10.0.0.0/16 e 10.0.0.0/24, respectivamente). É possível alterar qualquer uma dessas configurações nesta caixa.

- Faça as alterações desejadas nos intervalos CIDR de *subnet* e VPC e clique em “Create VPC”. O *wizard* começa a criar a VPC, o internet gateway, a *subnet* e a tabela de rota.
- Uma janela de status mostra o trabalho em andamento. Quando o assistente for concluído, uma página é exibida confirmando que a VPC foi criada.
- Clique em “Close” para retornar ao painel de controle da VPC.
- Clique em “VPC” no painel de navegação à esquerda para exibir informações da VPC.



A **Figura 7-15** ilustra a caixa de diálogo com as definições do internet gateway e da subnet. A AWS já sugere as faixas de endereçamentos IP para a VPC e para a *subnet* pública que podem ser modificadas.



Figura 7-15 Create an Amazon VPC

A VPC tem um conjunto padrão de opções de DHCP, incluindo um servidor DNS padrão (*AmazonProvidedDNS*). As configurações padrão são suficientes para este exemplo. A VPC também possui tabelas de rota e um ACL de rede padrão.

- Clique em “Route Tables” no painel de navegação esquerdo.

A VPC tem duas tabelas de rota. Uma é a tabela de rota principal que a VPC tem como padrão e a outra é uma tabela de rota personalizada que o *wizard* cria. Sua *subnet* é associada com a tabela de rota personalizada, o que significa ser possível usar as rotas nessa tabela para determinar como o tráfego da *subnet* vai fluir. Marque a caixa de seleção para a tabela de rota personalizada (aquele com “No” na coluna “Main”) e observe as informações de rota exibidas no painel inferior.

A **Figura 7-16** ilustra o detalhe desta tabela de rota.

Route Table: rtb-43397828				
Routes	Associations	Route Propagation		
Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	active	No	<button>Remove</button>
0.0.0.0/0	igw-4e397825	active	No	<button>Remove</button>
	select a target			<button>Add</button>

Figura 7-16 Route Table

A primeira linha da tabela é a rota local que permite a comunicação dentro do VPC. Essa rota está presente em cada tabela de rota por padrão e não é possível removê-la.

A segunda linha mostra a rota que o *wizard* adicionou à tabela para habilitar o tráfego destinado a qualquer endereço IP fora da VPC (ou seja, 0.0.0.0/0) para fluir da *subnet* para o internet gateway.

### 7.3.8.2. Criar grupo de segurança na VPC

Configure um grupo de segurança para controlar o tráfego de entrada e saída das instâncias iniciadas.

Um grupo de segurança é apenas um grupo de instâncias que compartilha um conjunto comum de regras que determinam o tráfego permitido para dentro e fora das instâncias. Para usar grupos de segurança cria-se um grupo, adicionam-se as regras de entrada e de saída e, em seguida, iniciam-se as instâncias no grupo. É possível adicionar e remover regras do grupo a qualquer momento. Essas alterações se aplicam automaticamente para as instâncias do grupo.

As instâncias em um grupo de segurança não têm de estar na mesma *subnet* na VPC. Por outro lado, as instâncias na mesma *subnet* não precisam pertencer ao mesmo grupo de segurança. A **Figura 7-17** ilustra como uma *subnet* pode ter instâncias em mais de um grupo de segurança: duas das instâncias na *subnet* estão no grupo A, enquanto as outras duas instâncias na mesma *subnet* estão no grupo B.

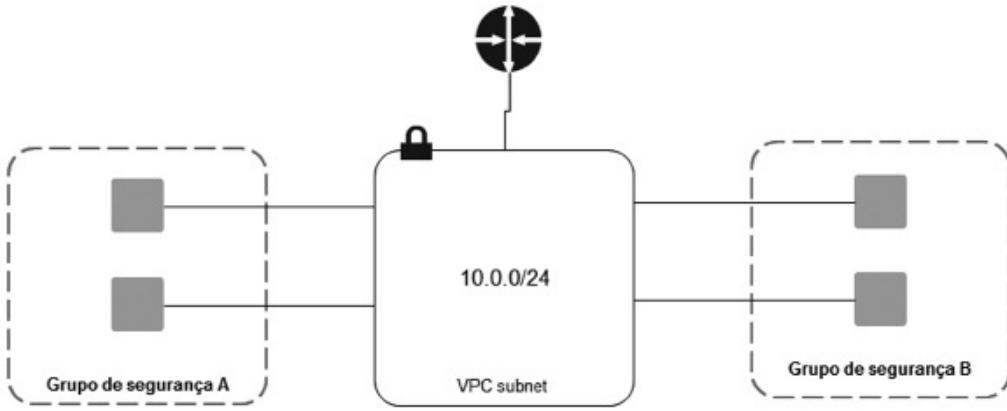


Figura 7-17 Grupos de segurança na VPC

Deve-se criar novos grupos de segurança especificamente para uso na VPC. No entanto, os nomes de grupo na VPC podem ser duplicados no EC2, pois cada grupo tem uma ID exclusiva atribuída à AWS.

A **Figura 7-18** mostra as setas apontando para dentro e para fora do grupo de segurança *WebServerSG*. Elas representam as regras de entrada e de saída definidas para o grupo.

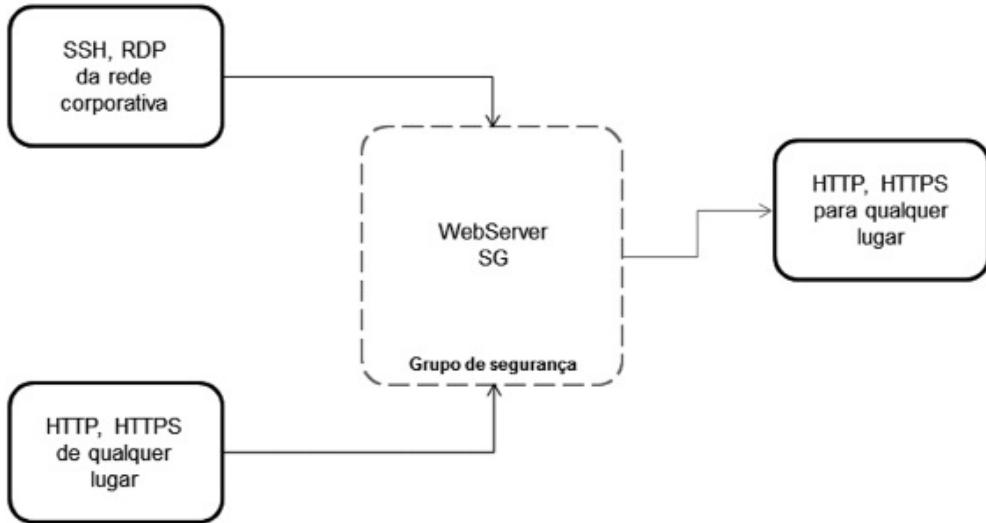


Figura 7-18 Regras de entrada e saída para grupo de segurança

As regras de entrada regulam o tráfego permitido para dentro de instâncias no grupo (ou seja, a origem do tráfego e a porta de entrada da instância). Todo o tráfego de retorno é permitido.

As regras de saída controlam para quais destinos as instâncias do grupo podem enviar tráfego (ou seja, o destino do tráfego e a porta de destino da instância). Todo o tráfego de retorno (ou seja, uma resposta do *host* que recebeu o tráfego) é automaticamente permitido retornar para as instâncias,

independentemente das regras de entrada definidas no grupo de segurança.

- Clique em “Create Security Group”. A caixa de diálogo para criar um grupo de segurança se abre. Digite o nome do grupo de segurança (por exemplo, *WebServerSG*), digite uma descrição do grupo, selecione a ID da VPC no menu VPC e clique em “Yes, Create” (sim, criar).

O grupo de segurança é criado na VPC e aparece na página grupos de segurança.

A **Figura 7-19** ilustra a tela de criação do grupo de segurança.



Figura 7-19 Create Security Group

- Na lista de grupos de segurança, marque a caixa de seleção para o grupo que acabou de ser criado. O painel inferior exibe detalhes do grupo de segurança. Há também duas guias: uma para trabalhar com as regras de entrada e uma para as regras de saída.

As regras de entrada para o grupo de segurança *WebServerSG* são mostradas na **Tabela 7-3**.

**Tabela 7-3 Regras de entrada**

IP de origem	Protocolo	Intervalo de portas	Comentários
0.0.0.0/0	TCP	80	Permitir acesso HTTP de entrada de qualquer lugar
0.0.0.0/0	TCP	443	Permitir acesso de entrada HTTPS de qualquer lugar
Intervalo IP público da rede doméstica	TCP	22	Permitir acesso de entrada SSH da rede doméstica
Intervalo IP público da rede doméstica	TCP	3389	Permitir acesso de entrada RDP da rede doméstica

A **Tabela 7-4** ilustra as regras de saída para o grupo de segurança *WebServerSG*.

**Tabela 7-4 Regras de saída**

IP de destino	Protocolo	Intervalo de portas	Comentários
0.0.0.0/0	TCP	80	Permitir acesso HTTP para servidores na internet
0.0.0.0/0	TCP	443	Permitir acesso HTTPS para servidores na internet

As regras de entrada regulam o tráfego permitido para dentro das instâncias no grupo. Todo o tráfego de retorno é permitido.

As regras de saída controlam para quais destinos as instâncias do grupo podem enviar tráfego. Todo o tráfego de retorno é permitido.

### 7.3.8.3. Iniciar uma instância na VPC

Inicie uma instância Linux na *subnet*. A instância tem um endereço IP privado no intervalo de endereços da *subnet*. A **Figura 7-20** ilustra a caixa de diálogo para iniciar uma instância Linux na *subnet* criada na VPC.

**Request Instances Wizard**

Cancel

CHOOSE AN AMI    INSTANCE DETAILS    CREATE KEY PAIR    CONFIGURE FIREWALL    REVIEW

Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

**Number of Instances:**     **Instance Type:**

**Launch as an EBS-Optimized instance (additional charges apply):**  Not supported for this instance type

**Launch Instances**

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

**Launch into:**  EC2  VPC

**Subnet:**  251 available IP addresses

**Request Spot Instances**

---

< Back      **Continue**

Figura 7-20 Launch instances

#### 7.3.8.4. Atribuir um endereço EIP na VPC

Atribua um endereço EIP para a instância. A **Figura 7-21** ilustra a opção de cadastrar um EIP para a instância.

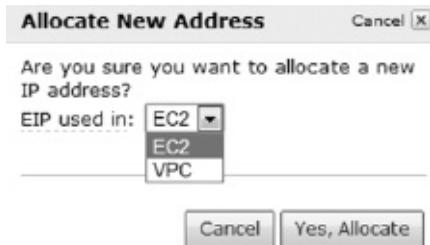


Figura 7-21 Allocate new address

A VPC foi criada com uma *subnet* pública e uma única instância em execução. Essa instância tem um endereço EIP e, por isto, está acessível pela internet. A VPC tem uma tabela de rota personalizada que permite que a instância envie tráfego para a internet. A instância está em um grupo de segurança que permite que o tráfego HTTP e o tráfego HTTPS possam fluir para dentro e para fora da instância e permite que o tráfego SSH e RDP flua para dentro.

#### 7.3.9. Importante

- O gateway do cliente para se conectar à VPC deve ser capaz de:
  - Estabelecer *IKE Security Association* usando chaves pré-compartilhadas.
  - Estabelecer *IPsec Security Associations* no modo *Tunnel*.
    - Utilizar a função de criptografia AES de 128 bits.
    - Utilizar a função de *hashing* SHA-1.
    - Utilizar *Diffie-Hellman Perfect Forward Secrecy* no modo “Group 2” (Grupo 2).
  - Estabelecer pares de *Border Gateway Protocol* (BGP).
    - Vincular túneis a interfaces lógicas (VPN baseado em rota).
      - Utilizar *IPsec Dead Peer Detection*.
        - Desempenhar a fragmentação de pacotes antes da criptografia.
- Os dispositivos de gateway do cliente que atendem aos requisitos mencionados anteriormente são conhecidos por funcionarem com conexões hardware VPN e oferecem suporte a ferramentas de linhas de comando para a geração automática de arquivos de configuração apropriados. Dispositivos já homologados pela Amazon:

- Cisco ISR executando o software Cisco IOS 12.4 (ou mais recente).
- Juniper J-Series Service Router executando o software JunOS 9.5 (ou mais recente).
- Juniper SRX-Series Services Gateway executando software JunOS 9.5 (ou posterior).
- Juniper SSG executando o software ScreenOS 6.1 ou 6.2 (ou mais recente).
- Juniper ISG executando o software ScreenOS 6.1 ou 6.2 (ou mais recente).
- Router Yamaha RTX1200.

■ É possível ter até cinco VPCs e até cinco endereços IP elásticos (EIPs) por conta por região. Pode-se criar até vinte *subnets* por VPC. Uma vez criada uma VPC ou *subnet*, não se pode alterar o intervalo de endereços IP. A VPC não é compatível com *multicast* ou *broadcast*.

- Pode-se também criar conexões VPN de hardware utilizando roteamento estático. Isto significa que é possível estabelecer conectividade utilizando dispositivos VPN que não suportam o protocolo BGP, como Cisco ASA, Microsoft Windows Server 2008m R2 e Linux. Qualquer implementação IPSec deve funcionar.
- Pode-se configurar propagação automática das rotas da VPN ou de links *Direct Connect* (gateways) para as tabelas de roteamento das VPCs. Assim, não é necessário criar entradas de rotas estáticas na tabela de rotas VPC para as conexões VPN.

## 7.4. Route 53

### 7.4.1. Introdução

O Route 53 é uma opção ao DNS fornecido por provedores e operadoras de telecomunicações. A proposta da Amazon é fornecer um serviço DNS com alta disponibilidade, baixa latência, alto desempenho e pago conforme a demanda.

Um aspecto a ser considerado no uso do Route 53 é que ele é integrado aos outros serviços AWS, como o IAM (*Identity and Access Management*). Pode-se com o IAM, por exemplo, criar uma política de acesso para que certos usuários possam fazer modificações nas configurações dos serviços relativos ao Route 53.

DNS significa *Domain Naming Services*, ou serviços de nomes de domínios. É o serviço responsável pela obtenção de um endereço IP a partir de um endereço de domínio e é executado por servidores DNS.

Todo computador conectado à internet tem configurações que indicam qual servidor DNS ele deve consultar para resolver nomes de domínios. Ao digitar o endereço de um site no seu navegador, inicia-se um processo que efetua diversas consultas em servidores DNS para que este endereço seja “resolvido” para o endereço IP correspondente.

As consultas feitas por servidores DNS seguem uma hierarquia que chega sempre ao servidor DNS final responsável pelo domínio. Este servidor final possui todas as informações de um domínio específico, como os registros “www”, “smtp”, “pop”, e retornará o endereço IP desses serviços para o computador de origem. Deve haver no mínimo dois servidores DNS com as informações de um domínio específico. O registro de um domínio é feito em empresas ou órgãos responsáveis, chamados registradores (*registrars*). Ao registrar um domínio em um *registrar*, deve-se informar quais são os servidores DNS responsáveis por este domínio.

O Route 53 é um serviço DNS de nuvem projetado para conectar as solicitações do usuário à infraestrutura em execução na AWS – como uma instância EC2, um volume ELB ou um *bucket* S3 – que também pode ser usado para direcionar usuários para infraestrutura fora da AWS.

O Route 53 realiza duas funções DNS:

- Em primeiro lugar, ele permite gerenciar listas de endereços IP e nomes de domínio correspondentes na internet. Essas listas são os registros DNS.
- Em segundo lugar, como um serviço de assistência de diretório, o Route 53 responde às consultas para traduzir nomes de domínio específicos para endereços IP correspondentes.

#### 7.4.2. Rede global do Route 53

O Route 53 responde a consultas DNS com baixa latência utilizando uma rede global de servidores DNS com a localização dos servidores descrita no capítulo 2. As consultas para um domínio específico são direcionadas automaticamente ao servidor DNS mais próximo e, assim, respondidas com o melhor desempenho possível.

#### 7.4.3. Zonas hospedadas (*hosted zones*)

O Route 53 tem uma interface simples de web service e seus registros DNS são organizados em “zonas hospedadas” que podem ser configuradas com a API do Route 53 ou pelo console de gerenciamento AWS.

Uma zona hospedada é análoga a um arquivo tradicional de zona do DNS; ela representa um conjunto de registros DNS que podem ser gerenciados juntos, pertencendo a um único nome de domínio pai. Todos os conjuntos de registros dentro de uma zona hospedada devem ter o nome de domínio da zona hospedada como um sufixo. Pode-se utilizar a API ou o

console de gerenciamento para criar, inspecionar, modificar e excluir zonas hospedadas.

#### 7.4.4. Registros DNS (*DNS resource records*)

Existem diversos tipos de registros DNS que são configurados para a maioria dos domínios. Os registros DNS são listas de endereços IP e nomes de domínio correspondentes. A forma com que acontece a relação endereço IP-nome de domínio dá origem a diferentes tipos de registros.

No momento, o Route 53 é compatível com os seguintes tipos de registro DNS:

- A (*address record* – registro de endereço).
- AAAA (*IPv6 address record* – registro de endereço IPv6).
- CNAME (*canonical name record* – registro de nome canônico).
- MX (*mail exchange record* – registro de troca de e-mail).
- NS (*name server record* – registro do servidor do nome).
- PTR (*pointer record* – registro do apontador).
- SOA (*start of authority record* – registro de início da autoridade).
- SPF (*sender policy framework* – estrutura da política do remetente).
- SRV (*service locator* – localizador do serviço).
- TXT (*text record* – registro de texto).

Um registro tipo A (*host*) é o mais comum. Este tipo de registro simplesmente aponta um endereço de domínio para um endereço IP.

Um registro CNAME (*Common Name*) é tipo *alias* (apelido), com o qual é possível apontar um endereço de domínio para outro endereço.

Para cada zona hospedada criada, o Route 53 cria automaticamente quatro registros NS e um registro SOA. A Amazon aconselha não mudar esses registros.

#### 7.4.5. Criação de um domínio

Para criar um domínio no Route 53:

- Registre seu nome de domínio. Utilize ICANN.org para ver uma lista de registradores onde se pode registrar um domínio. Após o registrador notificar que o seu nome de domínio está registrado com sucesso, pode-se criar uma zona hospedada para o domínio no Route 53. A zona hospedada criada pode armazenar registros DNS para seu domínio. Ao criar a zona hospedada, recebem-se quatro registros de nomes de servidores (NS) e um registro de início de autoridade (SOA), através de quatro diferentes domínios de nível

superior (*Top-Level Domain – TLDs*) para ajudar a garantir um elevado nível de disponibilidade. O domínio de topo é um dos componentes dos endereços de internet. Cada nome de domínio na internet consiste de alguns nomes separados por pontos, e o primeiro desses nomes é o domínio de topo. Por exemplo, no nome de domínio “exemplo.com”, o TLD é “com” ou “COM”.

- Atualize os registros de servidor de nome (NS) com o registrador para referenciar os servidores de nome do Route 53.

Para transferir o seu domínio de outro serviço de DNS para o Route 53, é preciso obter uma lista dos dados de registros de DNS para o nome de domínio, geralmente disponível sob a forma de um “arquivo de zona” que pode ser obtido do fornecedor de DNS existente.

## 7.4.6. Tipos específicos de registros DNS

Existem três tipos específicos de registros DNS utilizados no Route 53 e descritos a seguir.

### 7.4.6.1. WRR (Weighted Round Robin)

Esta funcionalidade permite que desenvolvedores especifiquem a frequência (pesos) com a qual diferentes respostas DNS são retornadas aos usuários finais. Pode-se usar o WRR para colocar servidores em produção, executar o teste A/B ou equilibrar o tráfego em DATACENTERS ou regiões de tamanhos variados.

O WRR permite designar ponderações aos conjuntos de registros de recursos para especificar a frequência com a qual diferentes respostas são correspondidas. Pode-se usar essa capacidade para desempenhar o teste A/B e enviar uma pequena porção do tráfego para um servidor no qual foi feita uma alteração no software. A Amazon dá um exemplo de uso: suponha que existam dois conjuntos de registros associados a um nome DNS – um com peso 3 e um com peso 1. Nesse caso, em 75% do tempo o Route 53 retornará o conjunto de registros com peso 3 e em 25% do tempo o Route 53 retornará o conjunto de registros com peso 1. Os pesos podem ser qualquer número entre zero e 255.

### 7.4.6.2. LBR (Latency Based Routing)

Esta funcionalidade ajuda a melhorar o desempenho do aplicativo para uma audiência global. LBR funciona roteando os clientes da aplicação para o ponto de extremidade da AWS (por exemplo, instâncias de EC2, IPs elásticos ou ELBs), que fornece a experiência mais rápida com base nas medições de desempenho real das diferentes regiões AWS em que o aplicativo está sendo executado.

### 7.4.6.3. Alias

Os registros *alias* são usados para mapear conjuntos de registros de recursos na zona hospedada para instâncias do ELB. Eles funcionam como um registro CNAME no qual é possível mapear um nome DNS (“meudominio.com”) para outro nome DNS “pretendido” (“elb1234.elb.amazonaws.com”). Eles diferem de um registro CNAME no fato de não estarem visíveis para os resolvedores. Os resolvedores veem somente o registro A e o endereço IP resultante do registro pretendido.

#### 7.4.7. Importante

- O Route 53 não oferece um serviço DNS privado. É possível definir registros DNS que retornarão endereços IP privados (endereços definidos na RFC 5735) no Route 53, mas esses registros não serão privados, e qualquer solicitante poderá consultar seu valor se souber o nome de registro.
- O Route 53 é compatível com registros IPv6 posteriores (AAAA) e anteriores (PTR). No entanto, o próprio serviço Route 53 não está disponível no IPv6 no momento.
- O tempo em que um resolvedor DNS armazena em cache uma resposta é determinado pelo período de ativação (TTL) associado a todo registro. O Route 53 não tem um TTL padrão para nenhum tipo de registro. Sempre deve ser especificado um TTL para cada registro, de forma que os resolvedores do DNS possam armazenar em cache os registros DNS durante o período especificado por meio do TTL.
- Outras operações com o Route 53 podem ser esclarecidas no “Amazon Route 53 Developer Guide”, referenciado no final do capítulo.

### 7.5. Direct Connect (DC)

O Direct Connect (DC) torna fácil estabelecer uma conexão de rede dedicada entre o DATACENTER corporativo e a AWS. Com o uso do AWS Direct Connect, pode-se estabelecer conectividade privada entre a AWS e o DATACENTER, o que, em muitos casos, pode reduzir custos de rede, aumentar a taxa de transferência de largura de banda e fornecer uma experiência de rede mais consistente do que conexões baseadas em internet.

A Figura 7-22 ilustra o uso do DC. A ideia do DC é não utilizar a internet como meio de ligação entre a AWS e o DATACENTER empresarial.

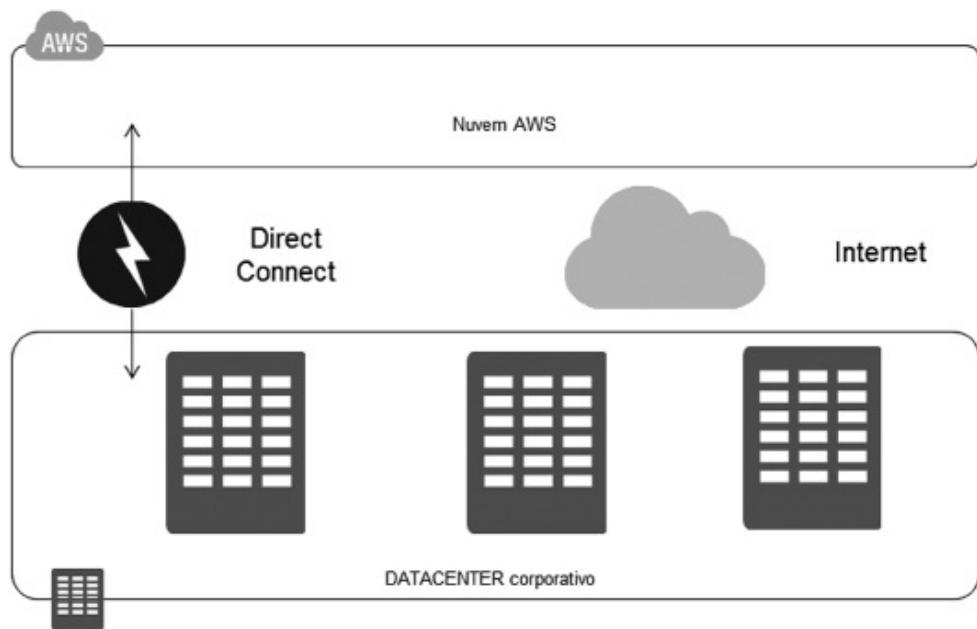


Figura 7-22 Direct Connect

O DC permite estabelecer uma conexão de rede dedicada entre uma rede local e um dos locais do DC. Ao utilizar VLANs 802.1q padrão, esta conexão dedicada pode ser particionada em várias conexões lógicas. Isso permite utilizar a mesma conexão para acessar recursos públicos, como os objetos armazenados no S3, usando o espaço de endereço IP público e recursos privados, como instâncias do EC2 em execução dentro de uma VPC, usando o espaço IP privado, mantendo a separação de rede entre os ambientes públicos e privados. Conexões lógicas podem ser reconfiguradas a qualquer momento para satisfazer essas necessidades.

A Amazon reforça que há várias situações onde o uso do DC pode ser desejável, incluindo:

- **Trabalhar com grandes conjuntos de dados:** transferir grandes conjuntos de dados pela internet pode ser caro e demorado. A maioria das empresas tem uma largura de banda que atende somente às suas necessidades de web e de e-mail e compartilha essa conexão por toda a empresa. Ao usar a nuvem, é possível que a transferência de grandes conjuntos de dados pareça, às vezes, um pouco lenta, pois o trânsito principal da rede da empresa tem que brigar por largura de banda com outros usos da internet. Para reduzir o tempo necessário para transferir os dados, pode-se aumentar a largura de banda do provedor de internet, o que frequentemente requer uma renovação onerosa de contrato e custos. Com o DC é possível transferir dados de negócios importantes diretamente de

DATACENTERS corporativos para a AWS, evitando o provedor de internet e removendo congestionamentos de rede. Além disso, a definição de preço do Direct Connect de somente pagar pelo que se utiliza, e o fato de não haver uma taxa mínima, significa que se paga apenas pelas portas de rede utilizadas e pelos dados transferidos através da conexão, o que pode reduzir significativamente os custos de rede.

- **Notificação de dados em tempo real:** aplicativos que usam notificações de dados em tempo real também se beneficiam do DC. Aplicativos de voz e vídeo, por exemplo, apresentam um melhor desempenho quando a latência de rede permanece constante. A latência de rede sempre pode variar, pois a internet constantemente altera a forma como os dados chegam do ponto A ao ponto B. Com a DC, é possível controlar como dados são direcionados, o que fornece uma experiência de rede mais consistente do que por conexões via internet.
- **Ambientes híbridos:** O DC pode ajudá-lo a construir ambientes híbridos que satisfaçam aos requisitos normativos do uso de conexões privadas. Ambientes híbridos possibilitam a combinação da elasticidade e dos benefícios econômicos da AWS com a capacidade de utilizar outra infraestrutura.

O DC está disponível atualmente em oito locais espalhados pelo mundo. A **Tabela 7-5** mostra opções de conectividade para regiões diferentes da AWS, incluindo São Paulo.

**Tabela 7-5 Locais da AWS Direct Connect**

Local do Direct Connect	Região AWS
TerraMark NAP do Brasil	São Paulo
CoreSite 32 Avenue of the Americas	Virgínia
CoreSite One Wilshire	N. Califórnia
Equinix DC1-DC6	Virgínia
Equinix SV1 & SV5	N. Califórnia
Equinix SG2	Singapura
Equinix TY2	Tóquio
TelecityGroup	Irlanda

A AWS sugere que, para fins de *failover*, sejam configuradas duas conexões Direct Connect para o DATACENTER corporativo. Essas configurações podem ser:

- **ativo-ativo**, via protocolo BGP, onde o tráfego de rede é balanceado entre ambas as conexões. Se uma conexão falha, todo o tráfego é roteado para a outra.
- **ativo-passivo**, onde uma configuração fica em *stand-by*.

A **Figura 7-23** ilustra a utilização de conexões redundantes com o AWS Direct Connect.

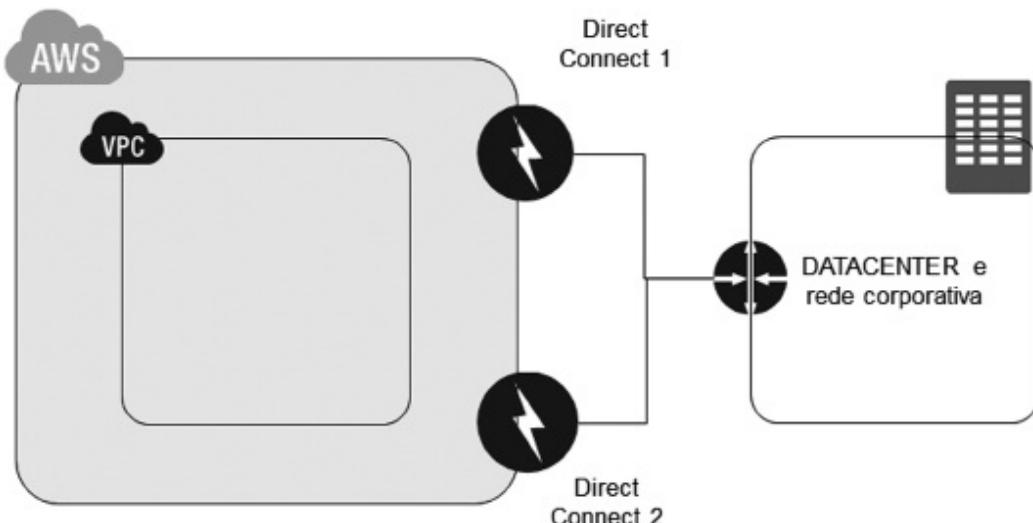


Figura 7-23 Conexão redundante com Direct Connect

## 7.6. Opções de conectividade com a VPC

Existem diversas formas de estabelecer a conectividade entre redes empresariais e redes AWS baseadas na VPC. O artigo “Amazon Virtual Private Cloud Connectivity Options”, publicado pela AWS em outubro de 2012, identifica essas principais formas. As opções de conectividade podem ser classificadas em rede corporativa para VPC; VPC para VPC e usuário interno para VPC.

- **Rede corporativa para VPC:** neste caso, as principais formas de conectividade podem ser resumidas em:
  - **Hardware VPN:** permite estabelecer uma conexão entre uma VPC e uma rede remota do cliente (*Customer Remote Network*) utilizando uma conexão IPSec através da internet. Veja a Figura 7-24.

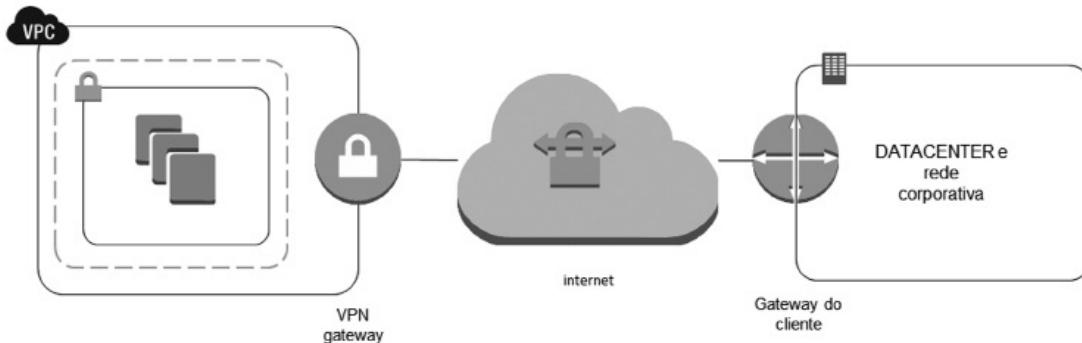


Figura 7-24 Rede de cliente – VPC: hardware VPN

- **Direct Connect:** permite estabelecer uma conexão dedicada entre uma VPC e uma rede corporativa remota de cliente utilizando o AWS Direct Connect. Esta conexão é privada e pode melhorar o desempenho e a consistência da rede. Veja a Figura 7-25.

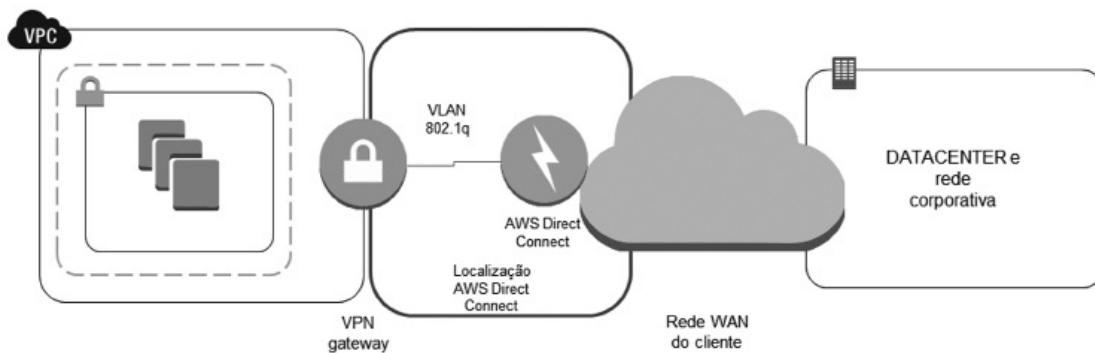


Figura 7-25 Rede de cliente – VPC: Direct Connect

- **Software VPN:** permite estabelecer uma conexão entre uma VPC e a rede corporativa remota do cliente. A VPC utiliza uma *appliance* VPN por software. Esta configuração é recomendada para clientes que querem gerenciar ambos os lados da conexão VPN ou para clientes que utilizam gateways por hardware não suportados pela VPC.
- **VPC-VPC:** neste caso, as principais formas de conectividade podem ser resumidas em:
  - **Software VPN:** permite estabelecer uma conexão entre dois ou mais *appliances* de software VPN em uma grande rede privada virtual. Instâncias em ambos os lados se comunicam utilizando IPs privados. Esta opção é interessante para clientes que querem realizar a conexão utilizando provedores preferenciais de software VPN. Esta configuração ainda utiliza internet gateway de cada lado para facilitar a conexão entre os *appliances*. As opções de conexão podem acontecer dentro de uma região ou entre regiões.

- **Software para hardware VPN:** permite estabelecer uma conexão que utiliza hardware VPN de um lado e software VPN do outro. Este design possibilita a criação de um túnel VPN entre um *appliance* VPN por software e um *Virtual Private Gateway* para conectar múltiplas VPCs em uma grande rede privada virtual utilizando IPs privados.
- **Hardware VPN:** permite criar uma VPN baseada em hardware IPsec para conexão com uma VPC utilizando a internet. Clientes podem utilizar múltiplas conexões VPN por hardware para rotear o tráfego entre suas VPCs.
- **Direct Connect:** permite estabelecer uma conexão dedicada entre uma rede corporativa interna e uma VPC ou entre VPCs. Esta opção pode reduzir custos de rede, aumentar o desempenho e propiciar uma experiência mais consistente de rede.
- **Usuário Remoto – VPC**
  - **Software de acesso remoto VPN:** clientes remotos AWS podem utilizar diversas opções disponibilizadas por fabricantes para se conectarem às redes VPC. Soluções diferem em complexidade, suporte a autenticação de diversos tipos de clientes e integração com outras ferramentas. Veja a **Figura 7-26**.

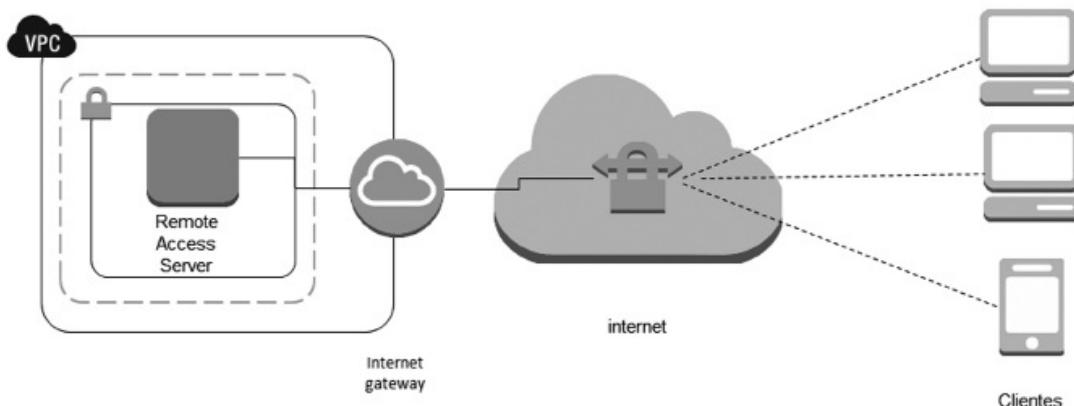


Figura 7-26 Usuário remoto – VPC: software de acesso remoto VPN

## 7.7. Referências bibliográficas

- Amazon Web Services. **Amazon Route 53 Developer Guide.** API Version 2012-02-29.
- Amazon Web Services. **Amazon Virtual Private Cloud: Getting Starter Guide.** API Version 2011-07-15.
- Amazon Web Services. **Amazon Virtual Private Cloud: User Guide.** API Version 2011-07-15.

**Amazon Web Services. Extend Your IT Infrastructure with Amazon Virtual Private Cloud.** Jan 2010.

<http://aws.amazon.com/pt/route53/>

<http://aws.amazon.com/pt/route53/faqs/>

<http://aws.amazon.com/pt/vpc>

<http://aws.amazon.com/pt/vpc/faqs/>

<http://aws.typepad.com/>

<http://aws.typepad.com/brasil/>

# 8. Banco de Dados

## 8.1. Introdução

Serviços de gerenciamento de banco de dados são fundamentais numa plataforma de nuvem. A AWS fornece esses serviços na forma de web service.

O Relational Database Service (RDS) é um web service que torna fácil de configurar, operar e dimensionar um banco de dados relacional na nuvem. O serviço fornece capacidade a um custo adequado e é redimensionável enquanto gerencia tarefas de administração de banco de dados normalmente demoradas, liberando o analista para se concentrar na camada de inteligência dos negócios.

O ElastiCache é um web service que permite operar e escalar um cache de memória na nuvem. O serviço melhora o desempenho de aplicativos web, permitindo que as informações sejam recuperadas de um sistema de cache rápido em memória e gerenciável, em vez de depender inteiramente de bancos de dados mais lentos que fazem o armazenamento em disco.

Este capítulo trata dos serviços de banco de dados relacional RDS e do serviço de cache de memória ElastiCache, pagos conforme o uso.

## 8.2. Bancos de dados tradicionais *versus* bancos de dados na AWS

Os gerenciadores de banco de dados (GBDs) podem ser classificados em duas grandes famílias: SQL e NoSQL.

Os bancos de dados relacionais do tipo SQL adquiridos de forma tradicional (aquisição do software e do hardware necessário) operam muito bem em situações normais. Em situações de grandes demandas de processamento e armazenamento esses bancos não conseguem manter o desempenho e garantir a escalabilidade. Diversas técnicas são utilizadas para corrigir este problema, incluindo *sharding* e *caching*, e até novas aquisições de hardware. Em certas situações, o banco de dados relacional do tipo SQL pode não ser a solução adequada.

Bancos de dados NoSQL foram projetados para obter escalabilidade e armazenam os dados construindo e utilizando as tabelas de forma diferente. Adquiridos da forma convencional, também não permitem a otimização do uso da infraestrutura, pois variações de demanda não são atendidas por uma infraestrutura rígida.

Na AWS é possível implementar as duas opções SQL (RDS) e NoSQL

(DynamoDB) na forma de web services. Neste caso, o benefício do uso da AWS seria o de utilizar a infraestrutura como serviço com base na demanda. A **Tabela 8-1** ilustra as diferenças entre bancos de dados tradicionais ou baseados na AWS.

Pode-se também utilizar uma instância Windows ou Linux EC2 na AWS e instalar um GBD qualquer aproveitando o benefício da licença BYOL (traga sua própria licença) permitida por alguns fornecedores de softwares gerenciadores de banco de dados. Esta forma de licenciamento foi abordada no capítulo 1.

**Tabela 8-1 Opções de gerenciadores de banco de dados (GBD)**

Tipo	GBD			GBD DynamoDB NoSQL
	convencional	GBD no EC2	GBD RDS	
	SQL ou NoSQL	SQL ou NoSQL	SQL	
Qual GBD	Escolhe GBD	Escolhe GBD	MySQL, Oracle ou SQL Server	Dynamo DB
Forma de licença	Licenciamento convencional	BYOL	BYOL ou licença incluída	Sem licenciamento
Forma de aquisição	Aquisição de capital	Infraestrutura como serviço	GBD como serviço	GBD como serviço

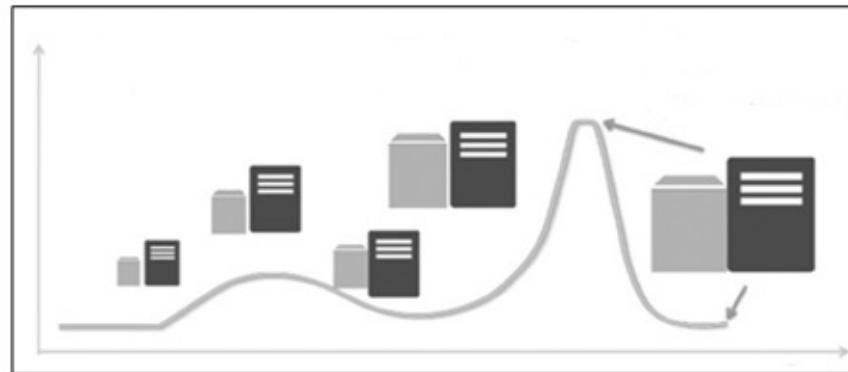
O RDS permite acesso aos recursos e mecanismos dos bancos de dados relacionais do tipo MySQL, SQL Server e Oracle de forma *on demand*. Isso significa que o código, as aplicações e as ferramentas que o usuário já usa hoje para manipular estes bancos podem continuar existindo e continuar sendo utilizados com o RDS.

O DynamoDB é uma solução NoSQL da AWS projetada para tratar dos principais problemas de desempenho e escalabilidade de grandes bancos de dados. Os desenvolvedores podem criar uma tabela de banco de dados que pode armazenar e recuperar qualquer quantidade de dados, assim como servir níveis distintos de solicitação de tráfego. O DynamoDB já é oferecido na região de São Paulo, mas não será foco deste capítulo.

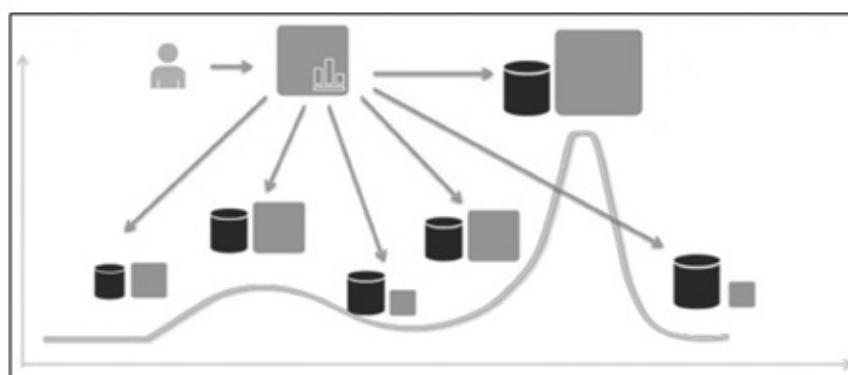
O artigo “The Total Cost of (Non) Ownership of a NoSQL Database Cloud Service”, publicado pela AWS, trata de demonstrar as vantagens em termos de menor custo total de propriedade (*Total Cost of Ownership – TCO*) para o DynamoDB.

A **Figura 8-1** ilustra a diferença de funcionamento entre um banco de dados no modelo tradicional (curva demanda *versus* tempo) e um banco de dados no modelo baseado na demanda (curva demanda *versus* tempo), a

proposta da AWS para o DynamoDB e para o RDS. A opção (a) ilustra que as variações de demanda não são acompanhadas pela infraestrutura do banco de dados (processamento, armazenamento) em um modelo tradicional. A opção (b) ilustra o contrário, com demanda e infraestrutura alinhadas à utilização de um sistema baseado na demanda e com monitoramento em tempo real.



(a) tradicional



(b) *on demand* com AWS

Figura 8-1 Banco de dados tradicional (a) Bancos de dados *on demand* (b)

## 8.3. RDS (Relational Database Service)

### 8.3.1. Visão geral

No RDS é possível se beneficiar da flexibilidade de poder dimensionar os recursos de computação ou da capacidade de armazenamento associada à instância do banco de dados relacional através de uma única chamada de API. Além disso, o RDS permite implantar uma instância de banco de dados em mais de uma zona de disponibilidade, para melhorar a disponibilidade e a confiabilidade em implantações críticas, aplicar automaticamente patches do software de banco de dados e fazer o backup do banco de dados, armazenando-o por um período de retenção definido pelo usuário.

A **Figura 8-2** ilustra a arquitetura de um DATACENTER considerando o RDS como o gerenciador de banco de dados. Observe que a arquitetura

baseada no RDS segue o mesmo princípio da arquitetura convencional de DATACENTER, só que agora é virtual e paga pelo uso. O banco de dados é *on demand*. O desenho da arquitetura será visto no capítulo 10.

É importante ressaltar que o RDS também é integrado ao IAM (*Identity Access and Management*) da AWS.

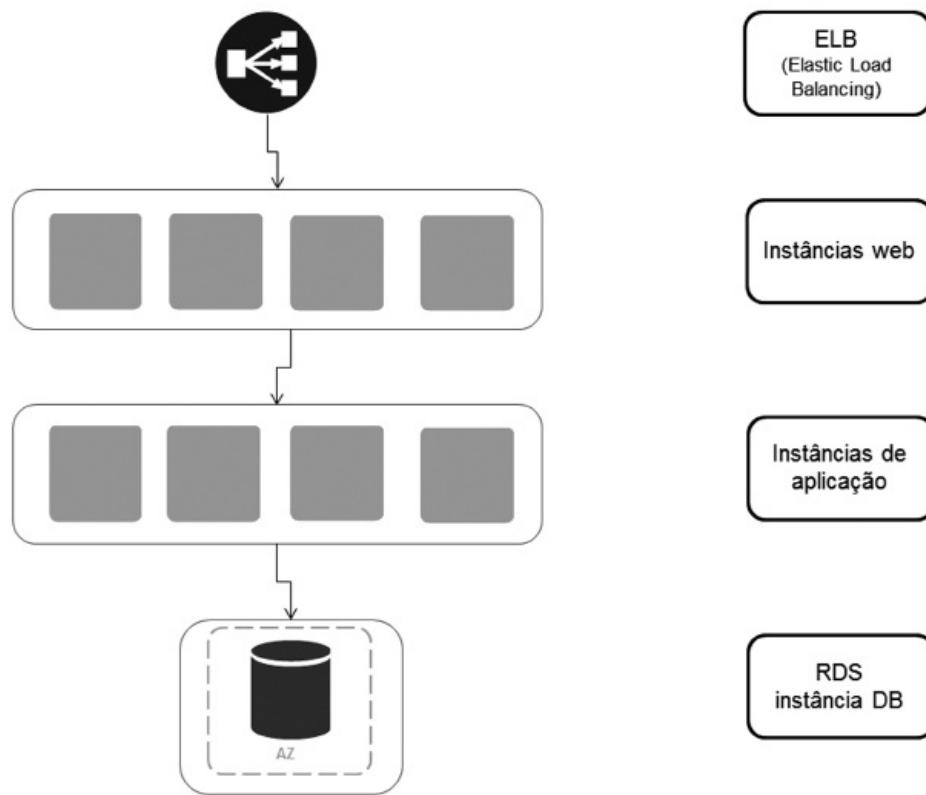


Figura 8-2 Arquitetura de DATACENTER com RDS

### 8.3.2. Conceitos

- **Instância DB (*DB instance*):** é um ambiente de banco de dados rodando na nuvem que pode conter múltiplos bancos de dados criados pelo usuário. Instâncias DB podem ser criadas ou modificadas utilizando ferramentas de linha de comando, APIs ou o console de gerenciamento AWS. O RDS cria uma conta de usuário “master” como parte do processo de criação de uma instância DB. O usuário master possui permissão para criar bancos de dados e criar, deletar, selecionar, atualizar e inserir operações nas tabelas do banco criado. Uma senha deve ser definida obrigatoriamente quando da criação da conta master.
- **Identificador de instância DB (*DB instance identifier*):** é fornecido pelo cliente para uma instância DB. Este identificador especifica uma instância particular do DB quando interagir com a API do RDS e com os comandos. O identificador de instância DB deve ser exclusivo para cada cliente em uma região AWS.
- **Nome do banco de dados (*Database Name*):** a definição do nome do banco de dados depende do mecanismo em uso.
  - Para o MySQL, o nome é o banco de dados hospedado em sua instância DB. Uma instância DB Amazon pode hospedar vários bancos de dados. Bancos de dados hospedados pela mesma instância DB devem ter um nome único dentro dessa instância.
  - Para o Oracle, o nome do banco de dados é usado para definir o valor do ORACLE\_SID, que deve ser fornecido quando se conectar à instância Oracle RDS.
  - Para o Microsoft SQL Server, o nome do banco de dados não é um parâmetro com suporte.
- **Mecanismo de banco de dados (*Database engine*):** o RDS é projetado para eliminar o *overhead* da manutenção associada à execução de um serviço de banco de dados relacional e oferecer suporte a vários mecanismos de banco de dados (*DB engines*).
- **Manutenção na instância DB (*DB instance maintenance*):** periodicamente, o RDS realiza manutenção na instância DB. Esta manutenção ocorre durante uma janela definida pelo usuário e pode incluir lançamentos de patches para o mecanismo de banco de dados ou sistema operacional em uso, bem como implementações de alterações pendentes. A manutenção da instância DB normalmente requer um curto tempo de inatividade para as instâncias do tipo Single-AZ. Se isto for um problema, pode-se pensar em utilizar uma implantação *Multi-AZ*, que permite obter um

tempo de inatividade menor durante a manutenção do sistema. Essas opções são explicadas adiante neste capítulo.

- **Grupo de segurança DB (*DB security group*):** O RDS possibilita o controle de acesso às instâncias DB usando os grupos de segurança DB. Um grupo de segurança DB atua como um *firewall* controlando o acesso à instância DB pela rede. Por padrão, o acesso de rede não é possível para as instâncias DB. As aplicações podem acessar a instância DB respeitando as regras dos grupos de segurança. Uma vez o ingresso configurado, as mesmas regras aplicam-se a todas as instâncias DB de um grupo de segurança.
- **Snapshots DB (*DB snapshots*):** os *snapshots* DB de uma instância DB são iniciados pelo usuário. Eles são mantidos até serem apagados pelo usuário.
- **Grupos de parâmetros DB (*DB parameter group*):** o RDS permite controlar a configuração do mecanismo do banco de dados utilizando os grupos de parâmetros DB, que atuam como um container para valores de configuração aplicados a uma ou mais instâncias.
- **Monitoramento da instância DB (*DB instance monitoring*):** o RDS coleta informações de desempenho para todas as instâncias DB. Utilizando as APIs do CloudWatch, pode-se acessar a CPU, o armazenamento e as métricas de conexões de banco de dados.
- **Eventos (Events):** eventos de log do RDS registram instâncias DB, *snapshots*, grupos de segurança e parâmetros de grupo. Esta informação inclui a data e o horário do evento, o nome da fonte, o tipo de fonte do evento e a mensagem associada com o evento. Pode-se facilmente recuperar os eventos de log usando o comando *rds-describe-events* ou a função API *DescribeEvents*.
- **Regiões e zona de disponibilidade (regions e available zone):** cada região é completamente independente. Qualquer atividade do RDS iniciada (por exemplo, criação de instâncias de banco de dados ou lista de instâncias de banco de dados disponível) é executada somente na região atual. Para criar ou trabalhar com uma instância DB do RDS em uma região específica, deve-se utilizar o *endpoint* do serviço regional correspondente. O *endpoint* para São Paulo é <https://rds.sa-east-1.amazonaws.com>.

### 8.3.3. Utilização

A Amazon sugere para utilização do RDS a sequência relatada a seguir:

- Use o console de gerenciamento AWS ou as APIs do RDS para iniciar uma instância de banco de dados (*DB instance*) selecionando o

mecanismo de banco de dados (MySQL, SQL Server ou Oracle), o tipo de licença, a classe da instância de banco de dados e a capacidade de armazenamento que melhor atende às necessidades do aplicativo.

- Conecte-se à instância usando uma ferramenta de banco de dados ou linguagem de programação. A maioria das ferramentas desenvolvidas para mecanismos MySQL, Oracle ou SQL Server fora da nuvem deve funcionar sem modificações com o RDS.
- Monitore a utilização de recursos de computação e de armazenamento da instância de banco de dados, sem custos adicionais, através das métricas do CloudWatch. O CloudWatch será visto no capítulo 9.
- Pague somente pelos recursos utilizados, com base nas horas de instância de banco de dados utilizadas, no armazenamento de banco de dados, no backup e na transferência de dados.

#### 8.3.4. Mecanismos de DB (*DB engines*)

As opções de *DB engines* (mecanismos de banco de dados) para o RDS podem ser resumidas na **Tabela 8-2**.

**Tabela 8-2 *DB engines* no RDS**

DB engine	Edição do DB
<i>mysql</i>	MySQL Community Edition <b>5.1.x &amp; 5.5.x</b>
<i>oracle-se1</i>	Oracle Database Standard Edition One 11.2.x
<i>oracle-se</i>	Oracle Database Standard Edition 11.2.x
<i>oracle-ee</i>	Oracle Database Enterprise Edition 11.2.x
<i>sqlserver-ex</i>	Microsoft SQL Server Express Edition 2008 R2 e 2012
<i>sqlserver-web</i>	Microsoft SQL Server Web Edition 2008 R2 e 2012
<i>sqlserver-se</i>	Microsoft SQL Server Standard Edition 2008 R2 e 2012
<i>sqlserver-ee</i>	Microsoft SQL Server Enterprise Edition 2008 R2 e 2012

Uma funcionalidade importante do RDS é o gerenciamento de versões, que permite o controle de quando e como o software do mecanismo de banco de dados deve ser corrigido e atualizado. Este recurso dá a flexibilidade para manter a compatibilidade com versões específicas de patch, testar novas

versões de patch para garantir que eles funcionem eficazmente com o aplicativo antes da implantação em produção e realizar upgrades da versão em seus próprios termos e prazos.

O gerenciamento de versões do RDS é feito utilizando a chamada de API *ModifyDBInstance*, o utilitário de linha de comando *modify-db-instance* ou através do console de gerenciamento AWS.

O licenciamento do RDS para Oracle e Microsoft SQL Server segue os padrões ditados no capítulo 1. Para o RDS baseado no MySQL não existe a necessidade de licenciamento.

#### **8.3.4.1. EBS com IOPS provisionado de alta performance para RDS**

A opção EBS com IOPS provisionado de alta performance permite especificar o tamanho e o desempenho do volume em termos de número de operações de I/O por segundo (IOPS). O EBS consistentemente vai entregar o desempenho desejado durante a vida útil do volume.

Pode-se agora criar uma instância de banco de dados RDS e especificar o nível desejado de IOPS para obter desempenho e *throughput* mais consistente.

É possível provisionar novas instâncias RDS de banco de dados com 1.000 a 10.000 IOPS e com 100 GB a 1 TB de armazenamento em bancos de dados MySQL e Oracle. Se estiver usando o SQL Server, o IOPS máximo que pode ser provisionado é 7.000. Todos os outros recursos do RDS, incluindo *Multi-AZ*, réplicas de leitura e a nuvem privada virtual, também são suportados.

Em um futuro próximo, a Amazon pretende fornecer uma maneira automatizada para migrar instâncias de bancos de dados existentes para o armazenamento do tipo IOPS provisionado para MySQL e Oracle. Neste momento a migração de uma instância existente do banco de dados para armazenamento do tipo IOPS provisionado pode ser feita exportando os dados e reimportando em uma nova instância de banco de dados, preparada com armazenamento de IOPS provisionado.

A **Figura 8-3** ilustra a opção na caixa de diálogo “Launch DB Instance Wizard”. Observe as opções “Use Provisioned IOPS” e “Provisioned IOPS”. A Amazon reforça que a relação entre a área de armazenamento alocada e o IOPS provisionado deve ser de dez para um. Também sinaliza as classes de instâncias otimizadas para uso do IOPS provisionado.

Launch DB Instance Wizard

Cancel

ENGINE SELECTION    DB INSTANCE DETAILS    ADDITIONAL CONFIGURATION    MANAGEMENT OPTIONS    REVIEW

To get started, choose a DB engine below and click **Continue**

**DB Engine:** oracle-se1

**License Model:** Bring Your Own License

**DB Engine Version:** Oracle 11.2.0.2.v5 (default)

**DB Instance Class:** db.m2.2xlarge

**Multi-AZ Deployment:** Yes

**Auto Minor Version Upgrade:**  Yes  No

---

Provide the details for your RDS Database Instance.

**Allocated Storage:\*\*** 100 GB (Minimum: 10 GB, Maximum: 1024 GB) Higher allocated storage may improve IOPS performance.

**Use Provisioned IOPS:**

**DB Instance Identifier:\*\***  (e.g. mydbinstance)

**Master Username:\*\***  (e.g. awsuser)

**Master Password:\*\***  (e.g. mypassword)

---

< Back Continue

Figura 8-3 EBS com IOPS provisionado no RBS

### 8.3.4.2. Amazon RDS para MySQL

Os recursos fornecidos pelo RDS dependem do mecanismo selecionado. As opções para MySQL são:

- **Parâmetros pré-configurados:** as instâncias do RDS são pré-configuradas com um conjunto sensível de parâmetros e definições apropriados para a classe da instância de banco de dados selecionada. É possível iniciar uma instância MySQL e conectar o aplicativo em instantes sem nenhuma configuração adicional. Se for necessário um controle adicional, pode-se utilizar o grupo de parâmetros do banco de dados.
- **Monitoramento:** o RDS fornece métricas CloudWatch para monitoração da instância de banco de dados gratuitamente. Pode-se usar o console de gerenciamento AWS para exibir métricas-chave operacionais para implementações da instância de banco de dados, incluindo utilização da computação/memória/capacidade de armazenamento, atividade I/O e conexões de instância de banco de dados.
- **Correção automática do software:** o RDS certificará que o software do banco de dados relacional que capacita a implementação permaneça atualizado com as correções mais recentes. Pode-se exercer controle opcional com relação a quando e se a instância de banco de dados é corrigida por meio do gerenciamento de versão do mecanismo de banco de dados.

**Backups automatizados:** backups são ativados como padrão. O

- recurso de backup automatizado do RDS permite a recuperação pontual para a instância de banco de dados. O RDS fará backup do banco de dados e de logs de transação e os armazenará por um período de retenção especificado pelo usuário. Isso permite restaurar a instância de banco de dados para qualquer segundo durante o período de retenção, até os últimos cinco minutos. O período de retenção de backup automático pode ser configurado para até trinta dias.
- **DB snapshots:** são backups da instância de banco de dados iniciados pelo usuário. Esses backups completos do banco de dados serão armazenados pelo RDS até serem excluídos explicitamente. Pode-se criar uma nova instância de banco de dados com base em um DB snapshot onde e quando for necessário.
- **Escalonamento com o toque de um botão:** usando as APIs do RDS ou clicando algumas vezes no console de gerenciamento AWS, pode-se escalar os recursos de computação e de memória, aumentando ou diminuindo a capacidade da implementação. As operações de aumento de capacidade normalmente são concluídas em alguns minutos. À medida que os requisitos de armazenamento aumentam, pode-se também fornecer armazenamento adicional imediato com um período de interrupção zero.
- **Substituição automática de host:** o RDS substituirá automaticamente a instância de computação no caso de uma falha no hardware.
- **Replicação:** o RDS para MySQL fornece dois recursos de replicação distintos, porém complementares: implementações *Multi-AZ* e *Read Replicas* que podem ser usadas em conjunto para obter uma disponibilidade aprimorada do banco de dados, proteger as atualizações mais recentes contra interrupções não planejadas e escalar além das limitações de capacidade de uma única instância de banco de dados para cargas de trabalho que exigem muita leitura. A replicação será explicada a seguir.

Os recursos fornecidos pelo RDS para o MySQL também existem para o Oracle e para o SQL Server. As exceções são as implementações baseadas em *Multi-AZ* e *Read Replica*. A implementação *Multi-AZ* já está disponível para Oracle.

Após a instância de banco de dados estar disponível, é possível recuperar o ponto de extremidade através da descrição da instância de banco de dados no console de gerenciamento AWS ou com a API *DescribeDBInstance*.

#### **8.3.4.3. Amazon RDS para Oracle**

O RDS para Oracle facilita a configuração, a operação e o escalonamento de implantações do banco de dados Oracle na nuvem. Com o RDS pode-se implantar em instantes várias edições do Oracle Database 11g, conforme mostrado em tabela anterior, com capacidade de hardware redimensionável e econômica.

O RDS é compatível atualmente com várias edições do Oracle. O suporte para uma determinada edição varia de acordo com o modelo de licença utilizado (*License Included* ou *License Mobility – Bring Your Own License*), já explicado no capítulo 1.

As opções para as licenças com Oracle são descritas a seguir:

- **Standard Edition One:** licença incluída, “Bring Your Own License” (traga sua própria licença).
- **Standard Edition:** *Bring Your Own License*.
- **Enterprise Edition:** *Bring Your Own License*.

O Oracle, com a funcionalidade *Oracle Data Pump*, permite mover dados em qualquer direção de forma fácil nas seguintes situações:

- Transferência entre bancos de dados Oracle *on-premises* e instâncias RDS.
- Transferência entre banco de dados rodando em instâncias EC2 e em instâncias RDS.
- Transferência de dados entre instâncias RDS.

A Amazon reforça que transferências usando *Data Pump* devem ser consideravelmente mais rápidas do que aquelas via utilitários de importação e exportação originais. O *Oracle Data Pump* está disponível em todas as novas instâncias de banco de dados executando a versão 11.2.0.2. do Oracle.

#### **8.3.4.4. Amazon RDS para MS SQL Server**

O RDS para Microsoft SQL Server torna fácil de configurar, operar e escalar implantações do SQL Server na nuvem. Com o RDS, pode-se implantar várias edições do SQL Server em poucos minutos, com uma ótima relação custo-benefício e capacidade de computação redimensionável.

O RDS para Microsoft SQL Server suporta várias edições do SQL Server, conforme mostrado em tabela anterior. O suporte para uma determinada edição varia de acordo com o modelo de licença utilizado.

- O RDS para Microsoft SQL Server utiliza o modelo *License Included* para instâncias DB, executando o SQL Server Express Edition, o SQL Server Web Edition e o SQL Server Standard Edition (SE). Neste modelo, conforme visto no capítulo 1, a licença é mantida pela AWS

e está incluída no preço da instância do banco de dados.

- O RDS para Microsoft SQL Server utiliza também o programa de mobilidade de licença da Microsoft (*License Mobility – Bring Your Own License* – BYOL), que permite mover facilmente cargas de trabalho do aplicativo de servidor Microsoft local atual para o AWS, sem custos adicionais de licença. Atualmente, o SQL Server Standard Edition e o SQL Server Enterprise Edition são as edições elegíveis para este programa. Este modelo também foi visto no capítulo 1.
- O RDS para Microsoft SQL Server permite utilizar um SQL Server Agent para programar e executar tarefas administrativas nas bases de dados SQL. O SQL Server Agent é uma ferramenta desenhada para reduzir o trabalho manual de otimização e manutenção de serviços de banco de dados.
- O RDS para Microsoft SQL Server já pode ser utilizado dentro de uma VPC. Ou seja, as instâncias RDS para SQL Server podem estar dentro da rede privada virtual.

O RDS já suporta o Microsoft SQL Server 2012. As edições Express, Web e Standard podem ser lançadas a partir do console de gerenciamento. Basta escolher a versão SQL 2012 na opção “DB Engine Version”. O SQL Server 2008 R2 continua disponível.

### **8.3.5. Replicação**

O RDS fornece duas opções de replicação distintas que servem a propósitos diferentes.

Se o desejo é utilizar a replicação para aumentar a disponibilidade do banco de dados enquanto protegem-se as mais recentes atualizações contra interrupções inesperadas, uma opção é executar a instância de banco de dados como uma implantação *Multi-AZ*. Esta forma de replicação está intimamente ligada à recuperação de desastres do banco de dados. A replicação será detalhada no capítulo 13.

Se o desejo for o de aproveitar a vantagem da replicação incluída no MySQL para dimensionar além das limitações de capacidade de uma única instância de banco de dados para cargas de trabalho de leitura pesada, o RDS para MySQL facilita isso com as réplicas de leitura.

#### **8.3.5.1. Arquitetura Multi-AZ**

Ao criar ou modificar uma instância de banco de dados para ser executada como uma implantação *Multi-AZ*, o RDS automaticamente providencia e mantém uma réplica “em *stand-by*” simultânea em uma zona de disponibilidade diferente.

Atualizações para a instância de banco de dados são replicadas simultaneamente na réplica em *stand-by*, a fim de manter ambos em sincronia e proteger as últimas atualizações contra falhas da instância de banco de dados. Durante a manutenção programada, ou no caso de falha de instância de banco de dados ou falha da zona de disponibilidade, o RDS automaticamente fará o *failover* para a opção *stand-by*, para que seja possível retomar gravações e leituras de banco de dados assim que a réplica *stand-by* for promovida. Visto que o registro de nome para a instância de banco de dados permanece o mesmo, o aplicativo pode retomar as operações sem a necessidade de intervenção manual.

A Figura 8-4 ilustra a opção *Multi-AZ*.

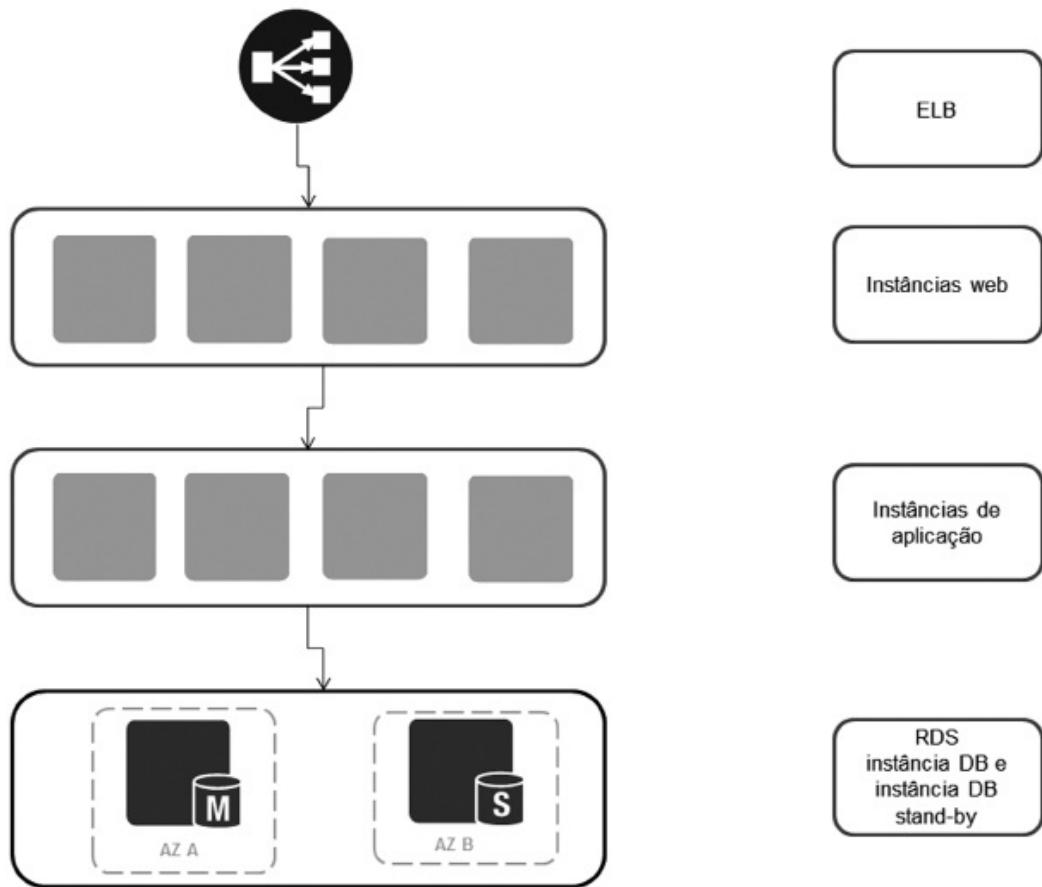


Figura 8-4 Opção Multi-AZ no RDS

Com implantações *Multi-AZ* a replicação é transparente: não existe interação diretamente com a instância em *stand-by* e ela não pode ser utilizada para servir ao tráfego de leitura em nenhum momento.

Os principais benefícios de executar uma instância de banco de dados como uma implantação *Multi-AZ* são a durabilidade e a disponibilidade aprimoradas de banco de dados. Essa maior disponibilidade e a tolerância a falhas tornam esta a melhor opção para ambientes de produção.

Executar uma instância de banco de dados como uma implantação *Multi-AZ* protege os dados, caso ocorra inesperadamente uma falha de componentes ou uma perda de disponibilidade. Por exemplo, se um volume de armazenamento da instância principal falhar, o RDS automaticamente inicia um *failover* para a instância *stand-by*, onde todas as atualizações do banco de dados estão intactas. Isso fornece uma durabilidade de dados adicional relativa às implantações padrão *Single-AZ*, em que uma operação de restauração feita pelo usuário seria necessária e atualizações feitas após o último momento restaurável (geralmente dentro dos últimos cinco minutos) não estariam disponíveis.

Existe também o benefício da disponibilidade aprimorada de banco de

dados ao executar a instância de banco de dados como uma implantação *Multi-AZ*. Se ocorrer uma falha de zona de disponibilidade ou uma falha da instância de banco de dados, o impacto da disponibilidade é limitado ao tempo que o *failover* automático leva para ser concluído (segundo a Amazon, geralmente três minutos).

Os benefícios de disponibilidade da replicação *Multi-AZ* também se estendem à manutenção planejada e aos backups. Com backups automatizados, por exemplo, a atividade de I/O não é mais suspensa na instância principal durante a janela de manutenção preferencial, pois os backups são retirados da instância *stand-by*. No caso de realizar um patch ou um dimensionamento de classe de instância de banco de dados, essas operações ocorrem primeiro na instância *stand-by*, antes do *failover* automático. Como resultado, o impacto da disponibilidade é limitado ao tempo necessário para o *failover* automático ser concluído.

Outro benefício incluído ao executar a instância de banco de dados como uma implantação *Multi-AZ* é que o *failover* de instância de banco de dados não requer nenhuma administração. No contexto do RDS, isso significa não precisar monitorar eventos de instância de banco de dados e iniciar a recuperação manual caso haja uma falha de zona de disponibilidade ou da instância de banco de dados.

Para criar uma implantação de instância de banco de dados *Multi-AZ*, basta clicar na opção “Yes” para “Multi-AZ Deployment” ao iniciar uma instância de banco de dados com o console de gerenciamento AWS, conforme ilustra a **Figura 8-5**.

The screenshot shows the 'Launch DB Instance Wizard' interface, specifically the 'DB INSTANCE DETAILS' step. The 'DB Engine' is set to 'oracle-se1'. Under 'Multi-AZ Deployment', the 'Yes' option is selected. Other settings include 'Allocated Storage' at 100 GB, 'Master Username' as 'awsuser', and 'Master Password' as 'mypassword'. The 'Continue' button is visible at the bottom.

Figura 8-5 DB instance details

Alternativamente, pode-se utilizar a API do RDS, chamar a API *CreateDBInstance* e configurar o parâmetro “Multi-AZ” para o valor “verdadeiro”. Para converter uma instância de banco de dados padrão *Single-AZ* para *Multi-AZ*, é necessário modificar a instância de banco de dados no console de gerenciamento AWS ou utilizar a API *ModifyDBInstance* e configurar o parâmetro *Multi-AZ* para “verdadeiro”.

As implantações *Multi-AZ* são propícias para os seguintes cenários de falha:

- Perda de disponibilidade na zona de disponibilidade principal.
- Perda de conectividade de rede da instância principal.
- Falha da unidade computacional da instância principal.
- Falha de armazenamento da instância principal.
- Expansão ou redução da classe computacional da sua instância de banco de dados.
- Realização de patching de software.

O RDS cria um evento de instância de banco de dados para informar que houve um *failover*. O *failover* é automaticamente controlado pelo RDS para

que seja possível retomar operações de banco de dados rapidamente e sem intervenção administrativa. Ao ocorrer um *failover*, o RDS simplesmente troca o registro de nome canônico (CNAME) para a instância de banco de dados para apontar para a opção *stand-by*, que, em troca, é promovida e se torna a nova instância principal.

O backup automatizado e a funcionalidade de *snapshot* de banco de dados funcionam da mesma forma em uma implantação padrão *Single-AZ* ou *Multi-AZ*. Se estiver executando uma implementação *Multi-AZ*, backups automatizados e *snapshots* de banco de dados são simplesmente realizados na instância *stand-by* para evitar suspensão do I/O na instância principal. A Amazon reforça que poderá haver aumento de latência de I/O (normalmente durante alguns minutos) durante a realização dos backups.

Iniciar uma operação de restauração (restauração de um momento exato ou restauração de um *snapshot* de banco de dados) também funciona da mesma maneira com implantações *Multi-AZ* e *Single-AZ*. Novas implantações de instância de banco de dados podem ser criadas com as APIs *RestoreDBInstanceFromSnapshot* ou *RestoreDBInstanceToPointInTime*. Essas novas implantações de instância de banco de dados podem ser *Single-AZ* ou *Multi-AZ*, independentemente do backup de origem ter sido iniciado em uma implantação *Single-AZ* ou *Multi-AZ*.

### 8.3.5.2. Arquitetura para réplicas de leitura

O recurso de réplica de leitura facilita o escalonamento além das limitações de capacidade de uma única instância de banco de dados para cargas de trabalho que exigem muita leitura. Pode-se criar uma ou mais réplicas de uma determinada instância de banco de dados de origem e atender ao tráfego de leitura de aplicativos de alto volume, aumentando portanto o processamento agregado de leitura. O RDS usa a replicação nativa do MySQL para propagar as alterações realizadas em uma instância de banco de dados de origem para qualquer réplica de leitura associada.

Pode-se criar uma réplica de leitura com alguns cliques no console de gerenciamento AWS ou com a API *CreateDBInstanceReadReplica*. Após a criação da réplica de leitura, atualizações na instância de banco de dados de origem serão replicadas utilizando a replicação assíncrona e nativa do MySQL.

A Figura 8-6 ilustra a utilização de réplicas de leitura.

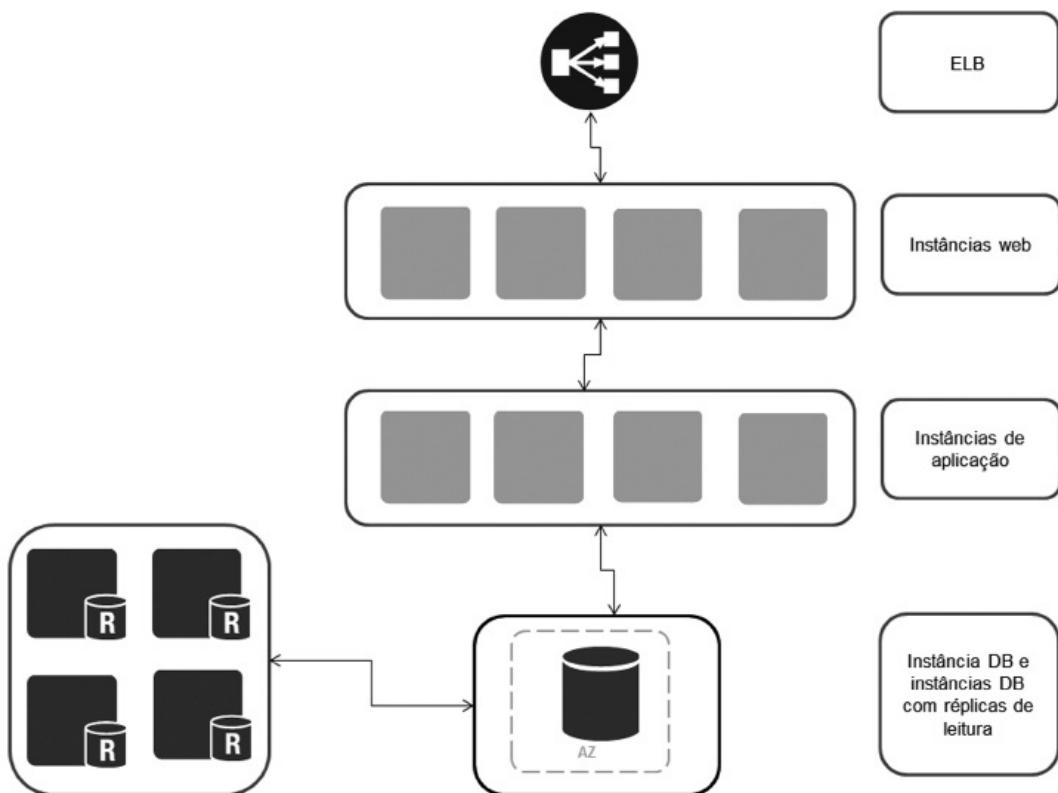


Figura 8-6 Rápidas de leitura no RDS

É possível criar múltiplas réplicas de leitura para uma determinada instância de banco de dados e distribuir o tráfego de leitura do aplicativo entre elas. Visto que réplicas de leitura utilizam replicação incluída no MySQL, elas estão sujeitas às suas capacidades e limitações. Mais especificamente, as atualizações são aplicadas à(s) réplica(s) de leitura após estas ocorrerem na instância de banco de dados de origem, e o atraso de replicação pode variar significativamente.

Razões comuns para implantar uma réplica de leitura incluem:

- Expandir além da capacidade computacional ou de I/O de uma única instância de banco de dados para cargas de trabalho de leitura pesadas de banco de dados. Esse tráfego de leitura excessivo pode ser direcionado a uma ou mais réplicas de leitura.
- Atender ao tráfego de leitura enquanto a instância de banco de dados de origem está indisponível. Se a instância de banco de dados de origem não consegue atender às solicitações de I/O (por exemplo, devido à suspensão de I/O para backups ou à manutenção programada), é possível direcionar o tráfego de leitura para uma réplica de leitura.
- Casos de relatórios de negócios ou de armazenamento de dados. Pode ser desejado que consultas aos relatórios de negócios sejam

executadas em uma réplica de leitura, em vez de ser na instância de banco de dados principal e de produção.

É possível criar uma réplica de leitura de forma rápida utilizando a API *CreateDBInstanceReadReplica* ou com apenas alguns cliques no console de gerenciamento RDS. A **Figura 8-7** ilustra a caixa de diálogo para criação da réplica de leitura após clicar na opção *Create Read Replica* na janela “My DB Instances”.

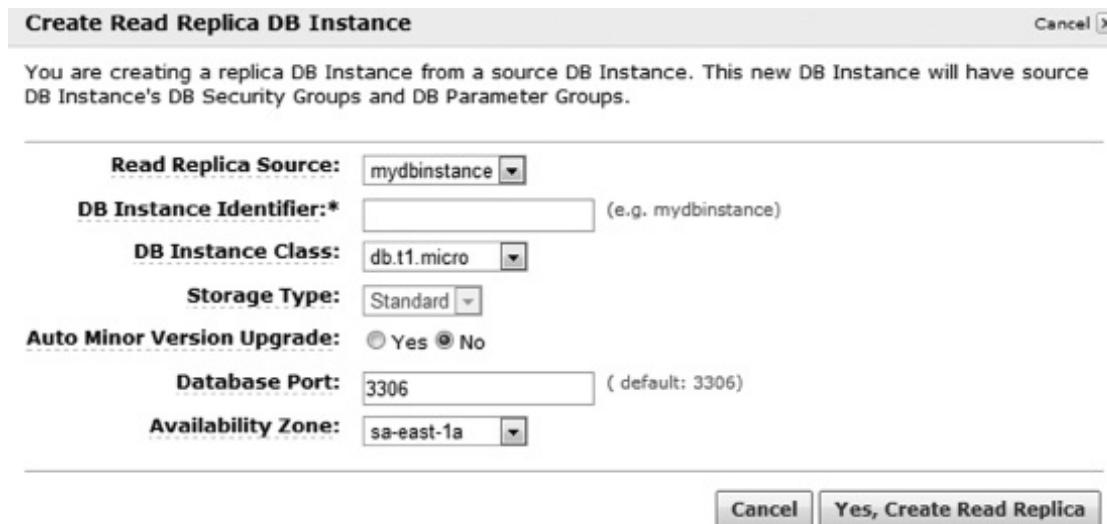


Figura 8-7 Create read replica DB instance

Ao criar uma réplica de leitura, pode-se identificá-la ao especificar um *SourceDBInstanceIdentifier*, que é o identificador da instância de banco de dados de origem a partir da qual você deseja fazer a replicação. Da mesma forma que com uma instância de banco de dados padrão, também é possível especificar a zona de disponibilidade, a classe de instância de banco de dados e a janela de manutenção preferida para uma réplica de leitura.

A versão MySQL e a alocação de armazenamento de uma réplica de leitura são herdadas da instância de banco de dados de origem. Ao iniciar a criação de uma réplica de leitura, o RDS faz um *snapshot* da instância de banco de dados de origem e inicia a replicação. Como resultado, ocorrerá uma breve suspensão de I/O da instância de banco de dados de origem à medida que ocorrer o *snapshot*. Essa suspensão de I/O geralmente dura cerca de um minuto e pode ser evitada se a instância de banco de dados de origem for uma implantação *Multi-AZ* (no caso de implantações *Multi-AZ*, *snapshots* são realizados a partir da opção *stand-by*).

As réplicas de leitura do RDS são tão fáceis de excluir quanto de criar; basta utilizar o console de gerenciamento do RDS ou chamar a API *DeleteDBInstance* (especificando-o para a réplica de leitura que deseja excluir).

É possível conectar-se a uma réplica de leitura da mesma maneira que a uma instância de banco de dados padrão utilizando a API *DescribeDBInstance* ou o console de gerenciamento AWS para recuperar o(s) ponto(s) de extremidade para a(s) réplica(s) de leitura.

Se forem criadas múltiplas réplicas de leitura, o aplicativo terá que determinar como o tráfego de leitura será distribuído entre elas. O RDS permite criar até cinco réplicas de leitura para uma determinada instância de banco de dados de origem.

A replicação é assíncrona; as gravações no banco de dados ocorrem em uma réplica de leitura após serem realizadas na mesma instância de banco de dados de origem, e esse “atraso” de replicação pode variar significativamente. Em contraste, a replicação utilizada pelas implantações *Multi-AZ* são simultâneas, de forma que todas as gravações de banco de dados são concomitantes na instância principal e na instância *stand-by*. Isso protege atualizações de banco de dados mais recentes, pois elas devem estar disponíveis na espera caso um *failover* seja necessário.

As implantações *Multi-AZ* e as réplicas de leitura usam tecnologias de replicação diferentes adequadas aos seus respectivos fins. Entretanto, elas podem ser utilizadas em conjunto para implantações de produção confiáveis e escaláveis. Basta designar uma implantação *Multi-AZ* como a origem de uma ou mais réplicas de leitura para obter as vantagens de durabilidade e disponibilidade de uma implantação *Multi-AZ* e os benefícios de escalonamento de réplicas de leitura. Essas opções são muito relevantes para ambientes críticos de banco de dados. A **Figura 8-8** ilustra esta opção.

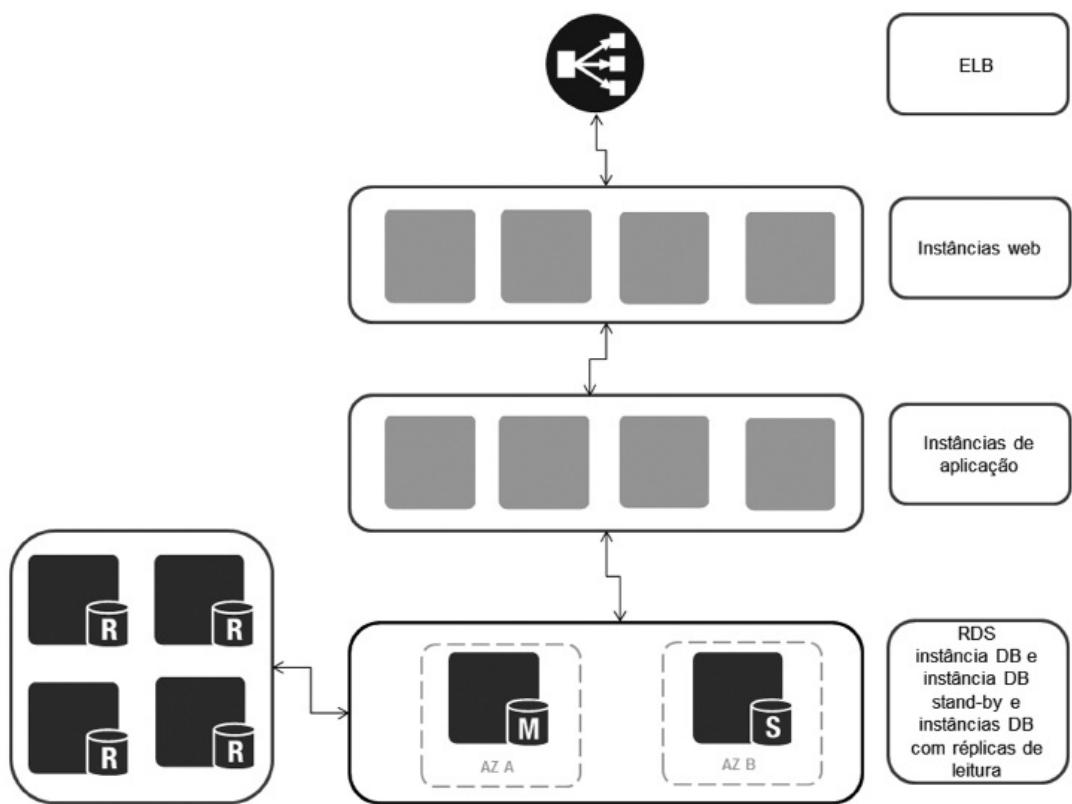


Figura 8-8 Multi-AZ e réplicas de leitura

### **8.3.6. Backup e restore**

O RDS fornece dois métodos diferentes de fazer backup. Ou faz o backup de forma automatizada ou faz o backup manual baseado em *snapshots* criados pelo usuário. Naturalmente, o restore deve ser contemplado nas duas opções.

Backup e restore são temas do capítulo 13.

#### **8.3.6.1. Backup automatizado**

O backup automatizado do RDS permite a recuperação dos dados de um momento exato da instância de banco de dados. Ao ativar backups automatizados, o RDS automaticamente realiza um *snapshot* diário completo dos dados (durante a janela de backup preferencial) e captura os logs de transação (à medida que instâncias de banco de dados são atualizadas).

Ao iniciar a recuperação de um momento exato, os logs de transação são aplicados ao backup diário mais apropriado, a fim de restaurar a instância do banco para o momento específico solicitado.

O RDS retém backups de uma instância de banco de dados por um período de tempo específico definido pelo usuário, chamado de período de retenção, que por padrão é de um dia, mas pode ser configurado para até 35 dias. Pode-se iniciar a restauração de um momento exato e especificar qualquer segundo durante o período de retenção, até o último momento restaurável. É possível utilizar a API *DescribeDBInstances* para retornar o último momento restaurável à(s) instância(s) de banco de dados, que geralmente ocorre dentro dos últimos cinco minutos. Outra opção para encontrar o último momento restaurável para uma instância de banco de dados é selecioná-lo no console de gerenciamento AWS e procurar na aba “Descrição” no painel inferior do console.

#### **8.3.6.2. Backup baseado em *snapshots***

Os *snapshots* de banco de dados são iniciados pelo usuário e permitem fazer o backup da instância de banco de dados em um estado conhecido e com a frequência que quiser para depois restaurar aquele estado específico a qualquer momento. Os *snapshots* de banco de dados podem ser criados com o console de gerenciamento AWS ou com a API *CreateDBSnapshot* e são armazenados até que voluntariamente sejam excluídos com o console ou com a API *DeleteDBSnapshot*.

A Amazon reforça que, ao realizar uma operação de restauração para um momento exato ou a partir de um *snapshot* de banco de dados, uma nova instância de banco de dados é criada com um novo ponto de extremidade. Isso é feito para criar múltiplas instâncias de banco de dados a partir de um *snapshot* de banco de dados específico ou de um momento exato.

Por padrão e sem custo adicional, o RDS permite realizar backups

automatizados de uma instância de banco de dados para um período de retenção de um dia. O armazenamento de backup gratuito é limitado ao tamanho do banco de dados provisionado e se aplica somente às instâncias de banco de dados ativas. Se for necessário estender o período de retenção de backup além de um dia, pode-se utilizar a API *CreateDBInstance* (ao criar uma nova instância de banco de dados) ou a API *ModifyDBInstance* (para uma instância de banco de dados existente). É possível utilizar essas APIs para modificar o parâmetro *RetentionPeriod* de um para o número de dias desejado.

Pode-se utilizar o console AWS ou a API *ModifyDBInstance* para gerenciar por quanto tempo os backups automáticos serão mantidos, modificando o parâmetro *RetentionPeriod*. Se for necessário desativar completamente os backups automatizados, isso é possível ao configurar o período de retenção para zero (a Amazon não recomenda). É possível visualizar uma lista de *snapshots* de banco de dados de usuário para uma instância de banco de dados específica utilizando a API *DescribeDBSnapshots* e excluir *snapshots* com a API *DeleteDBSnapshot*.

Os backups automatizados do banco de dados e os *snapshots* são armazenados no S3.

### 8.3.7. Tipos de instâncias RDS

O RDS atualmente oferece suporte a sete classes de instâncias de banco de dados:

- Microinstância
  - **Instância de banco de dados micro (db.t1.micro):** memória de 613 MB, um ECU, plataforma de 64 bits, capacidade de I/O pequena (somente MS SQL Server).
- Padrão
  - **Instância de banco de dados pequena (db.m1.small):** memória de 1,7 GB, um ECU (um virtual core com um ECU), plataforma de 64 bits, capacidade de I/O moderada.
  - **Instância de banco de dados grande (db.m1.large):** memória de 7,5 GB, quatro ECUs (dois virtual cores com dois ECUs cada), plataforma de 64 bits, capacidade de I/O elevada.
  - **Instância de banco de dados extragrande (db.m1.xlarge):** memória de 15 GB, oito ECUs (quatro núcleos virtuais com dois ECUs cada), plataforma de 64 bits, capacidade de I/O elevada (somente MySQL).
- Mais memória
  - **Instância de banco de dados extragrande com mais memória**

- **(db.m2.xlarge):** 17,1 GB de memória, 6,5 ECUs (dois núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
- **Instância de banco de dados dupla extragrande com mais memória (db.m2.2xlarge):** 34 GB de memória, 13 ECUs (quatro núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
- **Instância de banco de dados quádrupla extragrande com mais memória (db.m2.4xlarge):** 68 GB de memória, 26 ECUs (oito núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.

Para cada classe de instância de banco de dados, o RDS fornece a capacidade de selecionar entre 5 GB a 1 TB de capacidade de armazenamento associada.

A **Figura 8-9** ilustra os tipos de instâncias RDS.

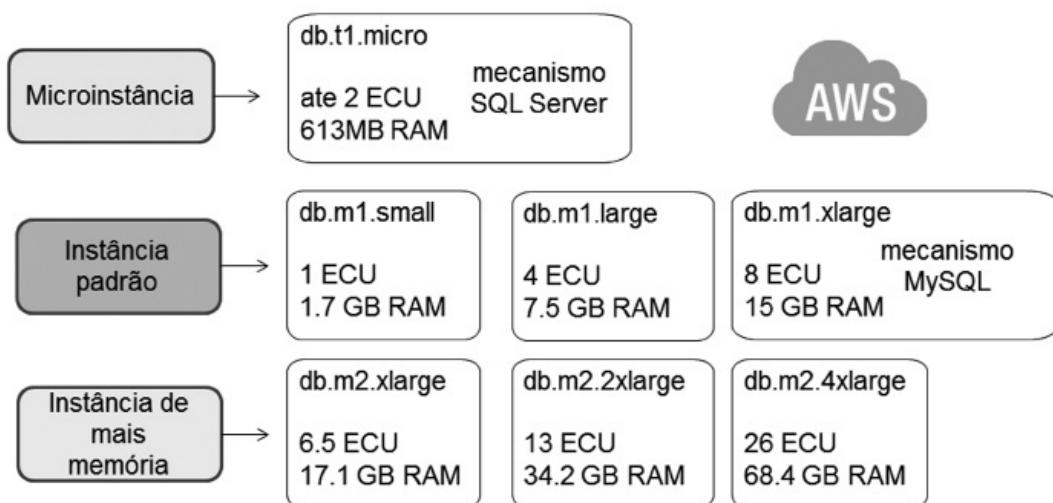


Figura 8-9 Tipos de instâncias RDS

### 8.3.7.1. Escolha da classe de instância

A capacidade de computação e a memória de uma instância DB é determinada pela classe da instância. Como resultado, pode-se aumentar ou diminuir a CPU e a memória disponível para uma instância DB alterando a sua classe. Se não se tem certeza de quanto se precisa de CPU, é recomendável começar com a classe *db.m1.small* e monitorar a utilização da CPU com o CloudWatch. A instância DB pode facilmente ser atualizada para uma classe maior usando o comando *rds-modify-db-instance*.

### 8.3.7.2. Escolha da quantidade de armazenamento

Para determinar a quantidade de armazenamento a ser alocada para uma instância DB, deve-se começar por estimar o armazenamento necessário em

condições típicas de funcionamento (estado estacionário). Acrescente a isso o armazenamento necessário para o crescimento esperado para os próximos três meses, sugestão da Amazon, além de qualquer armazenamento adicional necessário para operações que exigem uma quantidade significativa de armazenamento temporal (tais como trabalhos em lotes ou grandes cargas de dados).

Se os dados já estão em um banco de dados existente, já existe uma boa medida dos requisitos de armazenamento. Quando copiar ou migrar os dados para o RDS, deve-se alocar armazenamento suficiente para atender a demanda de pico do banco de dados atual e adicionar o suficiente para permitir o crescimento previsto. Para dados em arquivos simples, uma regra para estimar a quantidade de espaço necessário para uma instância DB é calcular o tamanho total dos arquivos simples e dobrar. Isso permite que o espaço para armazenamento de trabalho e índices secundários esteja contemplado.

Para os dados armazenados em algum outro formato, determine a quantidade de espaço necessário para armazenar os dados como arquivos simples e use o método de estimativa de arquivos simples. Pode-se monitorar a quantidade de espaço de armazenamento alocado com o CloudWatch. O RDS adiciona armazenamento sem reiniciar a instância DB e sem interromper os processos ativos no caso do MySQL e do Oracle.

O Microsoft SQL Server, devido às limitações de capacidade de extensão de armazenamento distribuído anexado a um Windows Server, não suporta atualmente o aumento de espaço de armazenamento. Se for necessário aumentar o armazenamento de uma instância SQL Server, será necessário exportar os dados, criar uma nova instância DB com maior armazenamento e, em seguida, importar os dados para a nova instância DB.

### **8.3.8. RDS em nuvem privada virtual (*Virtual Private Cloud – VPC*)**

Um dos cenários para usar o RDS é executar a instância dentro de uma VPC. Pode-se assim criar uma *subnet* pública para os servidores web que têm acesso à internet e colocar as instâncias de banco de dados do RDS de *backend* em uma *subnet* privada.

A funcionalidade básica do RDS é a mesma, independentemente do uso da VPC. O RDS gerencia backups, correções de software, detecção e recuperação de falha automática, bem como permite utilizar réplicas de leitura e escala de armazenamento, independentemente de suas instâncias de BD serem implantadas dentro ou fora da VPC. No entanto, atualmente, somente o Amazon RDS para MySQL é suportado pela VPC.

Ao criar uma instância DB na VPC, é necessário selecionar um grupo de *subnet* DB. O RDS usa esse grupo e a zona de disponibilidade preferencial para selecionar uma *subnet* e um endereço IP e associá-lo à instância DB.

### **8.3.9. RDS na prática**

Este item é baseado no guia de conceitos básicos da Amazon RDS (API Version 2010-01-01).

#### **8.3.9.1. Lançar instância DB**

Para utilizar o RDS é necessário ter uma conta AWS.

- No console de gerenciamento selecione “Amazon RDS”. No *dashboard* do console Amazon RDS clique em “Launch DB Instance” (Iniciar uma instância de banco de dados) para iniciar o *wizard*.
- Clique no botão “Select” próximo ao banco de dados MySQL.
- Conforme ilustrado na **Figura 8-10**, defina os detalhes da instância de banco de dados.

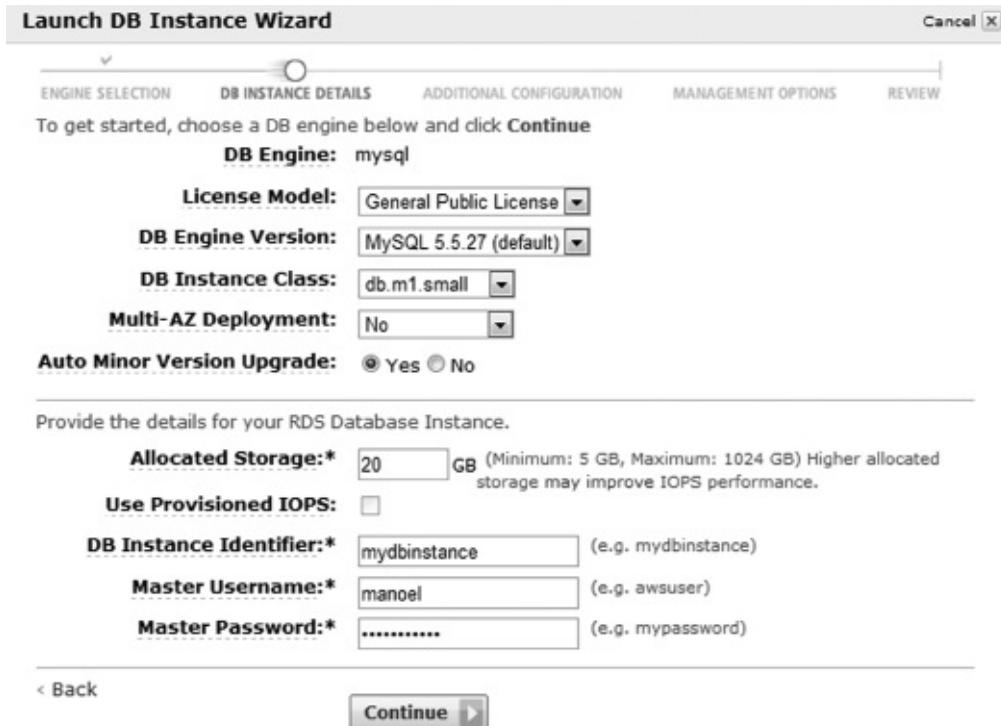


Figura 8-10 DB instance details

- “License Model” (modelo de licença): mantenha o padrão General Public Licence.
- “DB Engine Version” (versão do mecanismo DB): Selecione 5.5.57.
  - “DB Instance Class” (classe de instância): selecione db.m1.small.
- “Multi-AZ Deployment” (implantação Multi-AZ): selecione “No”.
  - “Auto Minor Version Upgrade” (atualização da versão do Auto Minor): selecione “Yes”. Esta opção permite que a instância de banco de dados receba automaticamente as atualizações de versão dos mecanismos utilizados.
- “Allocated Storage” (armazenamento alocado): 20. Esta opção permite especificar o volume de armazenamento em gigabytes desejados inicialmente para a instância.
- “Use Provisioned IOPS” (uso de IOPS provisionado): Não marcar. Foi explicado anteriormente.
- “DB Instance Identifier” (identificador de instância de banco de dados): mydbinstance. Nome para a instância de banco de dados que é exclusivo para uma conta em uma região.
- “Master User Name” (nome de usuário mestre): Manoel. Use o nome de usuário mestre para fazer logon na instância de banco de dados com todos os privilégios do banco de dados.

- “**Master User Password**” (**senha de usuário mestre**): Digite uma senha para o usuário mestre na caixa de texto.
- Depois de clicar na opção “Continue”, a página “Additional Configuration” (Figura 8-11) é mostrada.



Figura 8-11 Additional configuration

- Digite *mydatabase* na caixa de texto “Database Name” (nome do banco de dados). Ao fornecer o nome do banco de dados, o RDS pode criar um banco de dados padrão na nova instância DB.
- Aceite os valores padrões para o resto dos parâmetros disponíveis na página e clique em “Continue”.
- As opções de gerenciamento (“Management Options”) ilustrada na **Figura 8-12** permitem definir o período de retenção do backup (“Backup Retention Period”), a janela de backup (“Backup Window”) e a janela de manutenção (“Maintenance Window”).

**Launch DB Instance Wizard**

**ENGINE SELECTION**   **DB INSTANCE DETAILS**   **ADDITIONAL CONFIGURATION**   **MANAGEMENT OPTIONS**   **REVIEW**   **Cancel**

**Enabled Automatic Backups:**  Yes  No

The number of days for which automated backups are retained.

Please note that automated backups are currently supported for InnoDB storage engine only . If you are using MyISAM, refer to details here .

**Backup Retention Period:**  days

The daily time range during which automated backups are created if automated backups are enabled

**Backup Window:**  Select Window  No Preference

Start Time  :  UTC  
Duration :  hours

The weekly time range (in UTC) during which system maintenance can occur.

**Maintenance Window:**  Select Window  No Preference

Start Time   :  UTC  
Duration :  hours

[« Back](#) **Continue** 

Figura 8-12 Management options

A Amazon sugere que a janela de manutenção do RDS seja uma oportunidade de controlar modificações de instâncias de banco de dados (como, por exemplo, dimensionar uma classe de instância de banco de dados) e a ocorrência de patches de software, caso alguma dessas atividades seja solicitada ou exigida. Se um evento de “manutenção” está programado para uma determinada semana, ele será iniciado e concluído em algum momento durante a janela de manutenção definida.

Os únicos eventos de manutenção que exigem uma instância de banco de dados *offline* são os que tratam de dimensionar operações computacionais (que geralmente levam poucos minutos do início ao fim) ou implantar patches de software. A aplicação necessária de patches é automaticamente programada somente para patches relacionados à segurança e à durabilidade. Segundo a Amazon, as aplicações de patches não ocorrem com frequência (geralmente uma vez a cada dois meses) e raramente exigem mais do que uma fração de tempo da janela de manutenção.

Se não for especificada uma janela de manutenção semanal preferencial ao criar sua instância de banco de dados, é atribuído uma valor padrão de trinta minutos. Se for desejado modificar quando a manutenção será realizada, deve-se alterá-la no console AWS ou utilizando a API *ModifyDBInstance*. Cada uma das instâncias de banco de dados pode ter janelas de manutenção preferenciais diferentes, se assim for desejado.

Para este exemplo do guia aceite os valores padrões e clique em “Continue”. O painel “Review” será exibido conforme ilustra a **Figura 8-13**.

**Launch DB Instance Wizard**

**REVIEW**

Please review the information below, then click **Launch DB Instance**.

**Engine:** mysql  
**Engine Version:** MySQL 5.5.27  
**License Model:** general-public-license  
**Auto Minor Ver. Upgrade:** Yes  
**DB Instance Class:** db.m1.small  
**Multi-AZ Deployment:** No  
**Allocated Storage:** 20  
**Provisioned IOPS:** default  
**DB Instance Identifier:** mydbinstance  
**Master User Name:** manoel  
**Master User Password:** \*\*\*\*\*

---

**Database Name:** mydatabase  
**Database Port:** 3306  
**Availability Zone:** No Preference  
**Option Group:** default:mysql-5-5  
**DB Parameter Group:** default.mysql5.5  
**DB Security Group(s):** default  
**DB Subnet Group:**

---

**Backup Retention Period:** 10  
**Backup Window:** 00:00-00:30  
**Maintenance Window:** mon:05:00-mon:05:30

Figura 8-13 Launch DB instance – review

Se tudo estiver OK, clique no botão “Launch DB Instance”. Caso contrário, clique em “Back” e corrija o que for necessário.

Depois de clicar no botão “Launch DB Instance” será exibida uma mensagem informando que sua instância de banco de dados será criada. Esta operação leva alguns minutos.

Clique no botão “Close”.

O painel “My DB Instances” será exibido. A instância de banco de dados será criada e estará pronta para ser utilizada. Depois da instância estar disponível só falta autorizar o acesso.

### 8.3.9.2. Autorizar acesso à instância DB

Para conceder acesso a um intervalo IP para um grupo de segurança de banco de dados:

- Selecione “DB Security Group” no painel de navegação no lado esquerdo da janela de console.

- Na lista “My DB Security Groups”, marque a caixa de seleção próxima ao grupo de segurança padrão.
- Na guia “Description”, na parte inferior da caixa de diálogo, selecione “CIDR/IP” na lista suspensa “Connection Type”, digite o intervalo CIDR e clique no botão “Add”.

Agora o acesso está autorizado.

#### **8.3.9.3. Conectar a instância DB**

Agora é possível conectar a instância DB utilizando qualquer ferramenta de terceiros para o mecanismo de banco de dados utilizado.

Na página “My DB Instances”, selecione a caixa próxima à instância *mydbinstance*. Na guia “Description”, no painel inferior, observe o ponto de acesso à instância do banco de dados para utilizá-lo no próximo passo. A **Figura 8-14** ilustra este passo.

Utilizando uma ferramenta qualquer acesse o banco de dados.

The screenshot shows the 'Amazon RDS : My DB Instances' interface. On the left, there's a navigation sidebar with 'Region' set to 'South America (Sao Paulo)'. Under 'Databases', it lists 'DB Instances', 'Reserved DB Purchases', 'Orderable DB Options', 'DB Snapshots', 'DB Security Groups', 'DB Parameter Groups', 'Option Groups', 'DB Subnet Groups', and 'DB Events'. The main area shows a table with one item: 'mydbinstance'. The table columns are 'DB Instance', 'VPC ID', 'Multi-AZ Class', 'Status', and 'Storage'. The status is 'available' with '20 GB'. Below the table, it says '1 DB Instances Selected'. A detailed view for 'mydbinstance' is shown with tabs for 'Description', 'Monitoring', 'Recent Events', and 'Tags'. The 'Description' tab is selected, displaying the following details:

DB Instance Name:	mydbinstance	Alarm Status:	None
DB Engine:	mysql	DB Engine Version:	5.5.27
License Model:	general-public-license	Auto Minor Vers. Upgrade:	Yes
DB Security Groups:	default	DB Status:	available
DB Instance Class:	db.m1.small	Endpoint:	mydbinstance.chjodfaznqq.sa-east-1.rds.amazonaws.com
Port:	3306	Zone:	sa-east-1b
Multi-AZ Deployment:	No	DB Storage:	20GiB

Figura 8-14 My DB instances

#### 8.3.9.4. Encerrar a instância DB

Pode-se encerrar uma instância DB a qualquer momento da mesma forma que se encerra uma instância qualquer.

#### 8.3.10. Importante

- Pode-se adquirir até vinte instâncias de banco de dados reservadas. Se for necessário executar mais de vinte instâncias de banco de dados, complete o formulário de solicitação da instância de banco de dados do RDS disponível e envie para a AWS.
- O pagamento único para instâncias reservadas não é reembolsável. Contudo, é possível encerrar uma instância de banco de dados a qualquer momento e, nesse caso, taxas de uso por hora não serão cobradas.
- O RDS suporta o acesso a partir de qualquer aplicativo de cliente SQL padrão. O RDS não permite acesso de *host* direto via Telnet, SSH (*Secure Shell*) ou conexão *Remote Desktop Connection* do Windows.
- A Amazon prepara para 2013 o lançamento do Redshift, um data warehouse na forma de web services em escala de Petabyte.

### 8.4. ElastiCache

#### 8.4.1. Introdução

O ElastiCache é um web service que apresenta protocolo compatível com

o *memcached*, um sistema de memória em cache de objetos amplamente adotado na indústria de TI. A Amazon reforça que códigos, aplicativos e ferramentas populares já utilizados com ambientes *memcached* funcionam com este web service.

O *memcached* é um cache para armazenar informações frequentemente usadas para evitar o carregamento (e processamento) de informações de origens mais lentas, como discos. Ele pode ser implementado em uma situação dedicada ou como um método para usar memória sobressalente em um ambiente existente.

O *memcached* é um projeto de software livre idealizado para fazer uso da RAM sobressalente em muitos servidores para agir como um cache de memória para informações acessadas com frequência. O elemento-chave é o uso do cache: o *memcached* fornece armazenamento temporário, em memória, de dados que podem ser carregados de outro local.

Ao usar o ElastiCache, pode-se adicionar um cache de memória baseado no *memcached* à arquitetura de aplicativo em questão de minutos. Com poucos cliques no console de gerenciamento AWS, pode-se iniciar um cluster de cache consistindo em uma coleção de nós de cache, cada um executando o software *memcached*. Em seguida, dimensiona-se a quantidade de memória associada com o cluster de cache em minutos, adicionando ou excluindo nós de cache para atender às demandas das alterações de carga de trabalho.

Aplicações que não são *stateless* podem se beneficiar do ElastiCache e manter a sessão do usuário (estado) em cache, evitando mantê-la no disco do banco de dados (pouco performático) e também evitando manter a sessão do usuário no servidor web, que pode ser perdida com a quebra do servidor. A **Figura 8-15** ilustra uma arquitetura para ElastiCache.

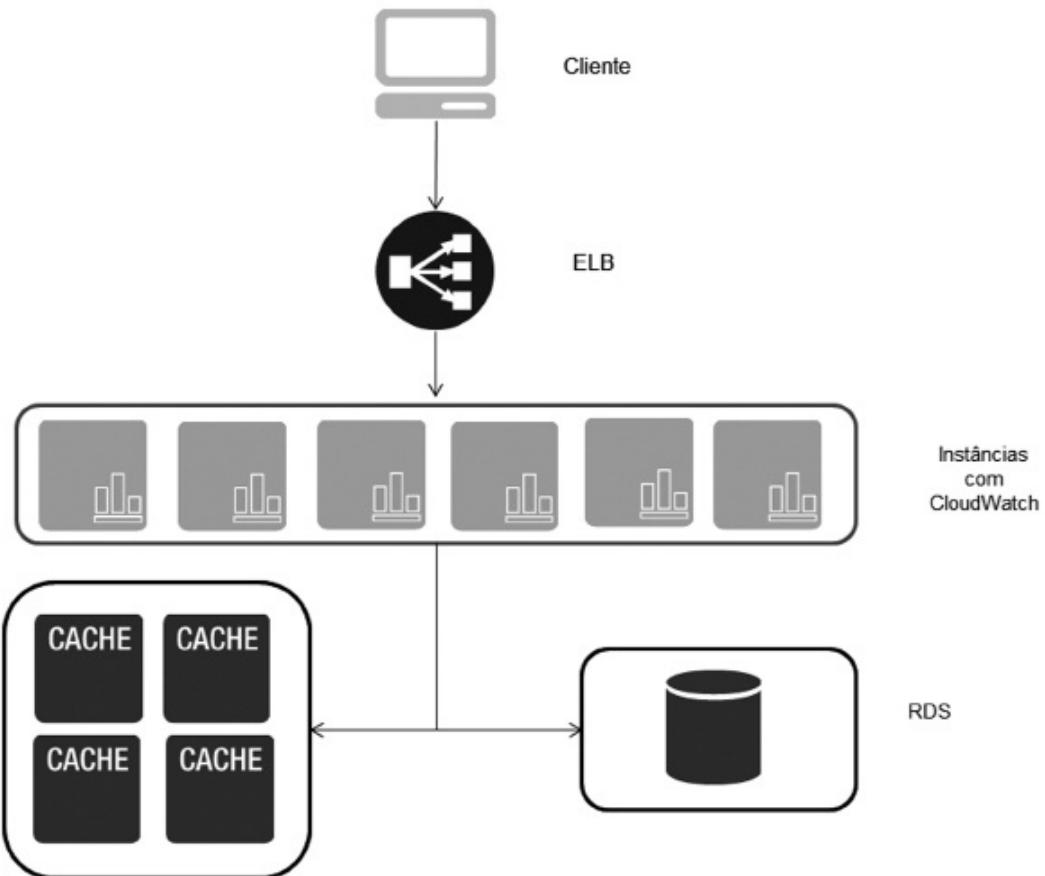


Figura 8-15 Arquitetura para o ElastiCache

O ElastiCache automaticamente detecta e substitui nós de cache com falha, fornecendo um sistema resistente que atenua o risco de sobrecarga de bancos de dados, tornando mais lento o tempo de carregamento de sites e de aplicativos.

Por meio da integração com o CloudWatch, o ElastiCache fornece métricas de desempenho associadas com os nós de cache.

Em resumo, a utilização do ElastiCache tem foco na redução da carga sobre o banco de dados (cache de query), melhorando o seu desempenho e no fornecimento de cache para páginas web dinâmicas (caches de páginas e sessão) e sistemas de gerenciamento de conteúdo (caches de objeto).

A **Figura 8-16** ilustra a caixa de diálogo para lançamento do cluster de cache ("Launch Cache Cluster Wizard"). É preciso providenciar uma série de informações para lançar o cluster, incluindo nome, tipo de nó e número de nós. Também se pode escolher a zona de disponibilidade onde rodará o cache de memória.

**Launch Cache Cluster Wizard**

**CACHE CLUSTER DETAILS**      **ADDITIONAL CONFIGURATION**      **REVIEW**

To get started, provide the details for your Cache Cluster below.

<b>Name*</b> :	CacheManoel
<b>Node Type*</b> :	cache.m1.small (1.3 GB memory)
<b>Number of Nodes*</b> :	6
<b>Engine:</b> Memcached	
<b>Cache Engine Version:</b> 1.4.5	
<b>Cache Port:</b>	11211 (e.g. 11211)
<b>Preferred Zone:</b>	us-east-1a
<b>Topic for SNS notifications:</b>	NotifyMe (manual ARN entry)
<b>Auto Minor Version Upgrade:</b> <input checked="" type="radio"/> Yes <input type="radio"/> No	
Note: "Auto Minor Version Upgrade" only applies to the Cache Engine software. Critical System Software patches (e.g. security related) may be applied irrespective of this selection.	

\*Required

**Continue**

Figura 8-16 Cache cluster details

#### 8.4.2. Conceitos

- **Cluster de cache (*cache cluster*):** é uma coleção de um ou mais nós de cache, cada um executando uma instância do serviço *memcached*. A maioria das operações será executada no nível do cluster de cache. Um cluster de cache pode ser configurado com um número específico de nós de cache e um grupo de parâmetros de cache que controla as propriedades de cada nó de cache. Todos os nós de cache dentro de um cluster de cache são projetados para ter o mesmo tipo de nó e têm o mesmo parâmetro e configurações de grupo de segurança.
- **Nó de cache (*cache node*):** é o menor bloco de construção de uma implantação de ElastiCache. É um tamanho fixo da memória RAM conectado à rede. Cada nó de cache executa uma instância do serviço *memcached* e tem porta DNS e nome próprios. São suportados vários tipos de nós de cache, cada um com diferentes quantidades de memória.
- **Identificador de cluster de cache (*cache cluster identifier*):** é um identificador fornecido pelo cliente para um cluster de cache. Este identificador especifica um determinado cluster de cache ao interagir com os comandos e API. O identificador do cluster de cache deve ser exclusivo para o cliente em uma região AWS.

- **Grupo de segurança de cache (*cache security group*):** o ElastiCache permite controlar o acesso aos clusters de cache usando grupos de segurança de cache. Um grupo de segurança de cache age como um *firewall*, um controle de acesso de rede ao cluster de cache. Por padrão, acesso de rede é desativado para os clusters de cache. Se você quer que seus aplicativos accessem o cluster de cache, você deve habilitar explicitamente o acesso de *hosts* em grupos de segurança específicos do EC2. Uma vez configuradas as regras de entrada, as mesmas regras se aplicam a todos os clusters de cache associados a esse grupo de segurança.
- **Grupo de parâmetros de cache (*cache parameter group*):** o ElastiCache permite controlar os parâmetros de tempo de execução de nós de cache usando grupos de parâmetro de cache. Um grupo de parâmetros de cache representa uma combinação de valores específicos para cada um dos parâmetros passados para o *memcached* durante a inicialização; portanto, determina o processo de comportamento em tempo de execução *memcached* em cada nó de cache. Os valores de parâmetro em um *cache parameter group* específico aplicam-se a todos os nós de cache em todos os clusters de cache associados a esse grupo.
- **Gestão da versão do mecanismo de cache (*cache engine version management*):** é possível controlar quando o software compatível com o protocolo de *memcached* deve ser atualizado para novas versões suportadas pelo ElastiCache. Isso permite manter a compatibilidade com versões específicas do *memcached*, testar novas versões com seu aplicativo antes da implantação em produção e realizar upgrades de versão em seus próprios termos e prazos.
- **Janela de manutenção (*maintenance window*):** cada cluster de cache tem uma janela de manutenção semanal, durante a qual as alterações de sistema são aplicadas. Se não for especificada uma preferência quando da criação do cluster de cache, o ElastiCache atribui uma janela de manutenção de trinta minutos em um dia da semana selecionado aleatoriamente.
- **Métricas CloudWatch (*CW metrics*):** o ElastiCache fornece métricas que permitem a monitoração dos clusters. Essas métricas podem ser acessadas através do CloudWatch.

#### **8.4.3. Utilização**

Um processo convencional de uso do ElastiCache pode ser descrito em alguns passos:

- Use o console de gerenciamento AWS ou as APIs do ElastiCache para iniciar um cluster de cache selecionando um nome de cluster de

cache, tipo de nó de cache e os números de nós que melhor atendem às necessidades.

- Conecte-se aos nós de cache usando clientes *memcached* favoritos ou linguagem de programação. Como o ElastiCache é compatível com *memcached*, a Amazon reforça que o código e a maioria dos clientes devem funcionar sem sofrer modificação.
- Obtenha estatísticas de monitoramento detalhadas dos nós de cache sem cobranças adicionais via CloudWatch.
- Se, em algum momento, for necessária capacidade adicional, pode-se adicionar mais nós de cache ao cluster de cache clicando algumas vezes no console ou por meio de uma chamada de API simples.
- Pague somente pelos recursos consumidos com base nas horas do nó de cache usadas.

#### 8.4.4. Recursos

Alguns dos recursos do ElastiCache são:

- **Parâmetros pré-configurados:** os nós do ElastiCache são pré-configurados com um razoável conjunto de parâmetros e configurações apropriados para o tipo de nó selecionado. Um *memcached* cluster pode ser iniciado rapidamente e conectado ao aplicativo em minutos, sem necessidade de configuração adicional. Se for necessário controle adicional, é possível fazê-lo através de grupos de parâmetro de cache.
- **Detecção automática de falha e recuperação:** o ElastiCache monitora o status dos clusters de cache e substitui automaticamente os nós de cache no caso de um particionamento de rede, uma falha de software ou uma falha de hardware de *host*. Os nós de cache substituídos são projetados para adotarem o mesmo nome DNS que os nós com falha de cache, evitando a necessidade de uma atualização das listas de ponto de acesso por parte do cliente.
- **Monitoramento detalhado e métricas:** o ElastiCache fornece métricas detalhadas baseadas no CloudWatch para implantações *memcached* sem custo adicional. O console AWS pode ser usado para visualizar as principais métricas operacionais para os nós de cache, incluindo a utilização da capacidade de memória/computação, número de acertos ao cache, erros de cache e número de conexões de cache.
- **Correção automática de software:** o ElastiCache atualiza o software de cache com as correções mais recentes. Pode-se controlar quando e se os clusters de cache serão corrigidos por meio do gerenciamento de versão do mecanismo de cache.

- **Botão de ação scaling:** com apenas alguns cliques no console AWS ou com uma simples chamada de API, é possível expandir os recursos de memória em minutos adicionando ou excluindo nós de cache.

#### 8.4.5. Nós de cache

O ElastiCache atualmente aceita os seguintes tipos de nó de cache:

- Padrão.
  - **Nó pequeno de cache (*cache.m1.small*):** 1,3 GB de memória, um ECU (um núcleo virtual com um ECU), plataforma de 64 bits, capacidade de I/O moderada.
  - **Nó grande de cache (*cache.m1.large*):** 7,1 GB de memória, quatro ECUs (dois núcleos virtuais com dois ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
  - **Nó extragrande de cache (*cache.m1.xlarge*):** 14,6 GB de memória, oito ECUs (quatro núcleos virtuais com dois ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
- Mais memória.
  - **Nó extragrande de mais memória de cache (*cache.m2.xlarge*):** 16,7 GB de memória, 6,5 ECUs (dois núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
  - **Nó duplo extragrande de mais memória de cache (*cache.m2.2xlarge*):** 33,8 GB de memória, treze ECUs (quatro núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
  - **Nó quádruplo extragrande com mais memória de cache (*cache.m2.4xlarge*):** 68 GB de memória, 26 ECUs (oito núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de I/O.
- Aprimorada.
  - **Nó de cache extragrande (*cache.m3.xlarge*):** 14,6 GB de memória, treze ECUs (quatro núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, capacidade de E/S moderada.
  - **Nó de cache duplo extragrande (*cache.m3.2xlarge*):** 29,6 GB de memória, 26 ECUs (oito núcleos virtuais com 3,25 ECUs cada), plataforma de 64 bits, alta capacidade de E/S.Mais CPU
  - **Nó extragrande com CPU de alta performance de cache (*cache.m6.6xlarge*):** 6,6 GB de memória, 26 ECUs (oito núcleos virtuais, cada um com unidades de processamento EC2 2.5),

plataforma de 64 bits, alta capacidade de I/O.

Cada tipo de nó de cache lista a memória disponível para o *memcached* após levar em consideração a sobrecarga do software do sistema. É importante reforçar que o ElastiCache está disponível na região América do Sul (São Paulo).

A **Figura 8-17** ilustra as opções.

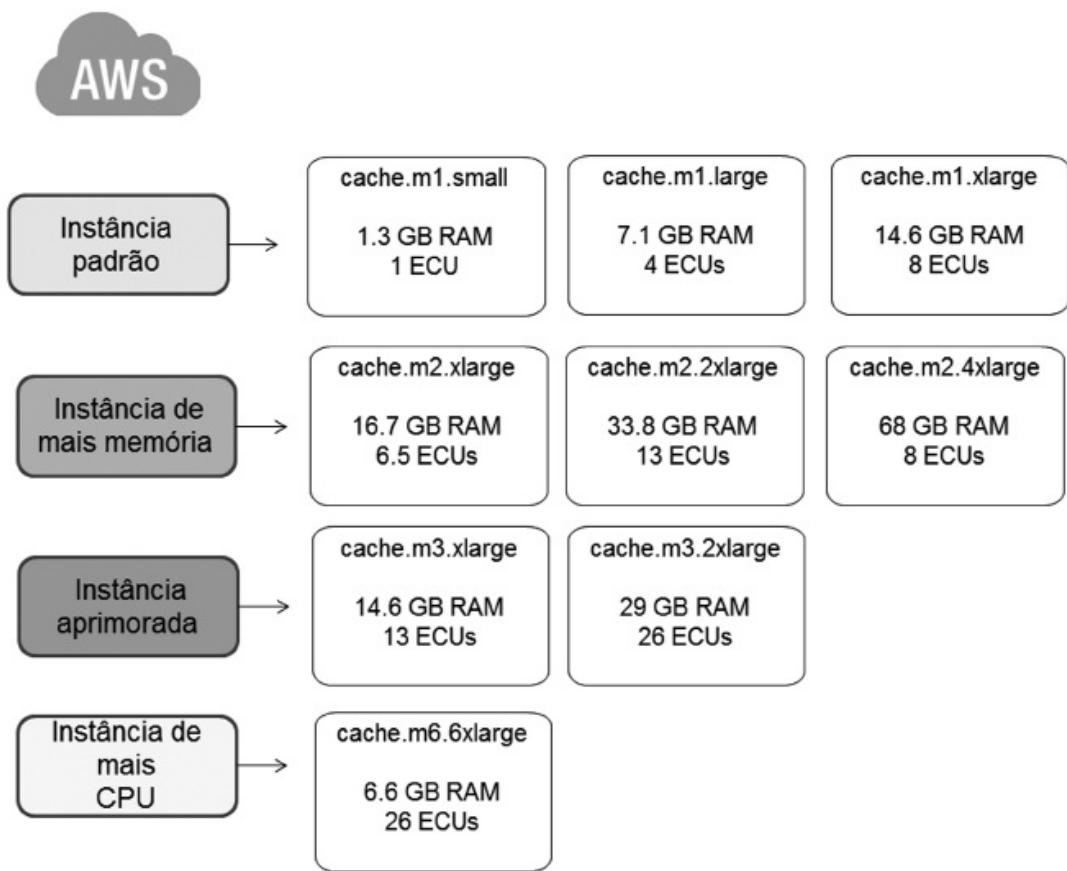


Figura 8-17 ElastiCache: tipos de nós de cache

- **Nós de cache reservados:** permitem que seja feito um único pagamento inicial para um nó de cache e se reserve o nó de cache para um período de um ou três anos a taxas significativamente mais baixas. Nós de cache reservados estão disponíveis em três variedades – utilização pesada, utilização média e utilização leve – que permitem otimizar custos do ElastiCache com base na utilização esperada.

As ferramentas de linha de comando, a API ou o Console de Gerenciamento AWS podem ser utilizados para listar e comprar as ofertas de nó de cache reservado disponíveis. Os três tipos de ofertas de nó de cache reservado baseiam-se na duração e classe do nó de cache.

A **Figura 8-18** ilustra a caixa de diálogo no console de gerenciamento AWS que permite adquirir os nós de cache reservados.

**Purchase Reserved Cache Nodes**

Select from the options below, then enter the number of Cache Nodes you wish to reserve with this order. When you are done, click the **Continue** button.

**Product Description:** memcached

**Cache Node Type:** cache.m1.small

**Term:** 1 year

**Offering Type:** Light Utilization

**Reserved Cache Node ID:** optional... (optional)

<b>One-time Payment (per cache node):</b> \$69.00	<b>Usage Charges:</b> \$0.045 (hourly)
<b>Number of Cache Nodes:</b> 1	Charges for your usage will appear on your monthly bill.
<b>Total One-time Payment: (Due Now):</b> \$69.00	

**Continue**

Figura 8-18 Purchase reserved cache nodes

A AWS introduziu um programa livre de uso por tempo limitado para o ElastiCache. A AWS pretende com isso conquistar novos clientes para o ElastiCache. Clientes elegíveis serão capazes de executar um pequeno nó de cache por sessenta dias. Este programa pode ser usado para desenvolver novos aplicativos, testar aplicativos existentes ou simplesmente obter experiência prática com o ElastiCache. O ElastiCache pode “rodar” dentro de uma VPC.

O recurso Auto Discovery permite que aplicativos automaticamente e de forma transparente adicionem ou removam nós de cache dos clusters. As aplicações podem reagir mais rapidamente às mudanças no cluster.

## 8.5. Referências bibliográficas

Amazon Web Services. **Amazon Relational Database Services: getting starter guide.** API Version 2011-04-01, 2011.

<http://aws.typepad.com/>

<http://aws.amazon.com/pt/rds/>

<http://aws.amazon.com/pt/rds/faqs/>

<http://aws.amazon.com/pt/dynamodb/>

<http://aws.amazon.com/pt/dynamodb/faqs/>

Varia, Jinesh & Papo, Jose. **The Total Cost of (Non) Ownership of a NoSQL Database Cloud Service**. Amazon Web Services, 2012.

# 9. Gerenciamento

## 9.1. Introdução

Serviços de gerenciamento também são fornecidos pela AWS.

O web service CloudWatch (CW) responde pelo gerenciamento de recursos, fornecendo métricas para monitoramento de boa parte dos serviços AWS e ainda permite a geração de alertas.

A AWS publica o status global de funcionamento dos web services no SHD (*Service Health Dashboard*). O status de um serviço específico pode ser obtido e monitorado em <http://status.aws.amazon.com/>. O capítulo 2 tratou disto.

Neste capítulo será visto o web service CloudWatch.

## 9.2. Gerenciamento tradicional *versus* gerenciamento na AWS

O DATACENTER consiste de dois núcleos principais: a infraestrutura (também conhecida como Tecnologia de Operação – TO) e a TI (Tecnologia da Informação). À medida que o DATACENTER evolui esses dois núcleos são cada vez mais integrados e o gerenciamento também.

Gerenciamento do DATACENTER sempre foi uma questão complexa. Existem inúmeras ferramentas de gerenciamento normalmente fornecidas pelos fabricantes para cada uma das partes do DATACENTER. O status atual é fazer o gerenciamento através de um framework/ferramenta que possibilite integrar as áreas de TO e TI. Surgiu então o DCIM (*Datacenter Infrastructure Management*).

A proposta do DCIM é possibilitar fazer o gerenciamento de todo o DATACENTER em um console único. Esta ainda é uma tarefa árdua. A proposta do DCIM é manter uma fonte com detalhes de consumo dos dispositivos de TI envolvidos na topologia do DATACENTER, otimizar os recursos utilizados na energização e no resfriamento do DATACENTER e poder ser até utilizado no planejamento da capacidade quando integrado a outras ferramentas de software.

O estágio atual do DCIM é realizar a integração entre ferramentas de empresas de TI como BMC, CA, HP, IBM, Microsoft e VMware e companhias de infraestrutura (TO) como Emerson e Schneider.

Em resumo, a proposta do DCIM é ser um único sistema para

diagnosticar toda a infraestrutura de TI e as instalações. A **Figura 9-1** ilustra a utilização do DCIM.

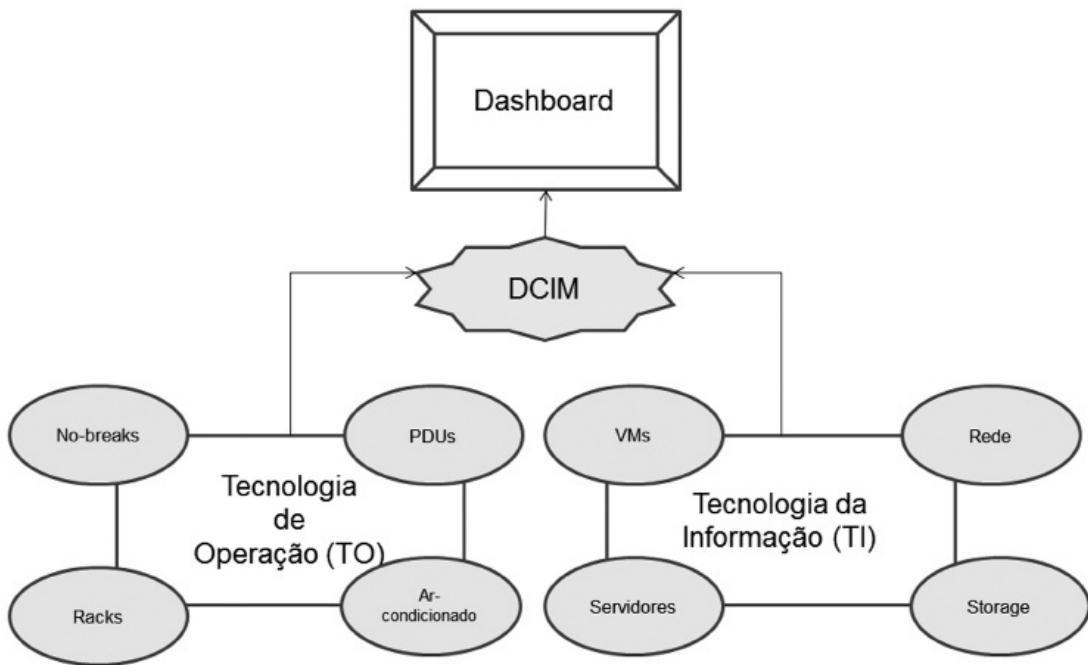


Figura 9-1 Gerenciamento tradicional com DCIM

O gerenciamento na AWS é fornecido como um web service. Existe uma opção de gerenciamento básico gratuito e outra opção mais detalhada, que é paga. O CloudWatch permite gerenciar cada DATACENTER construído com a arquitetura AWS.

O gerenciamento da operação (TO) é realizado pela própria AWS. A posição sobre o status dos serviços nos vários DATACENTERS da Amazon é fornecida pelo SHD.

A **Figura 9-2** ilustra as duas opções da AWS.

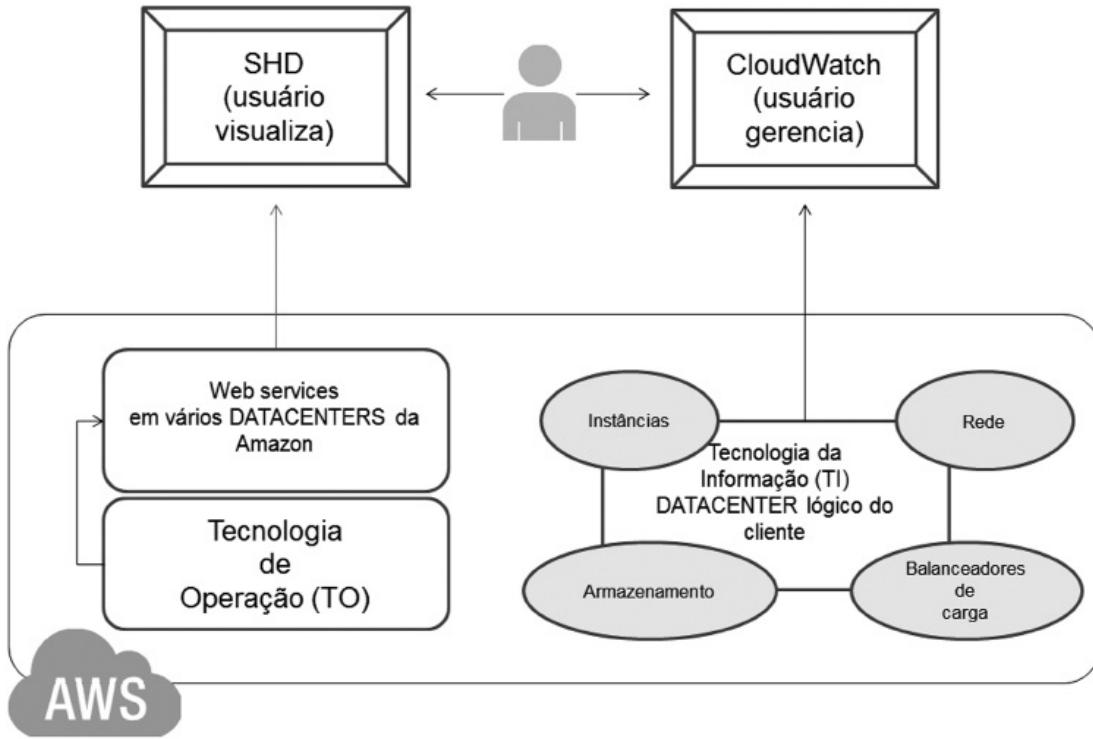


Figura 9-2 Gerenciamento com a AWS

### 9.3. CloudWatch

O CloudWatch é um web service que permite monitorar, gerenciar e publicar diversas métricas relativas ao funcionamento da AWS, bem como configurar ações de alarme baseadas em dados dessas métricas.

O CW permite verificar a utilização de recursos, desempenho operacional e padrões de demanda em geral, incluindo métricas como utilização de CPU, leitura e gravação em disco e tráfego de rede.

Os alarmes produzidos pelo CW ajudam na implementação de decisão de forma mais fácil, permitindo enviar notificações ou automaticamente fazer alterações nos recursos monitorados, com base em regras predefinidas.

O CW é comumente utilizado para manter aplicativos e serviços, funcionando com eficiência. A Amazon sugere que você pode usá-lo para descobrir, por exemplo, que seu site funciona melhor quando o tráfego de rede em suas instâncias EC2 permanece abaixo de certo limiar. Pode-se criar também um procedimento automatizado para garantir o número certo de instâncias para uma determinada quantidade de tráfego. O CW pode diagnosticar problemas, olhando para o desempenho do sistema antes e após a ocorrência de um problema. É importante ressaltar que o CW não agrupa dados entre regiões.

A **Figura 9-3** associa o CW a um repositório de métricas. Um produto AWS – tal como o EC2 – coloca as métricas no repositório, recupera estatísticas e gera alarmes com base nessas métricas. Se uma empresa coloca

suas próprias métricas personalizadas no repositório, poderá recuperar estatísticas baseadas em métricas personalizadas. Os alarmes geram notificações SNS e alimentam o serviço de *Auto Scaling*, a ser visto no capítulo 10.

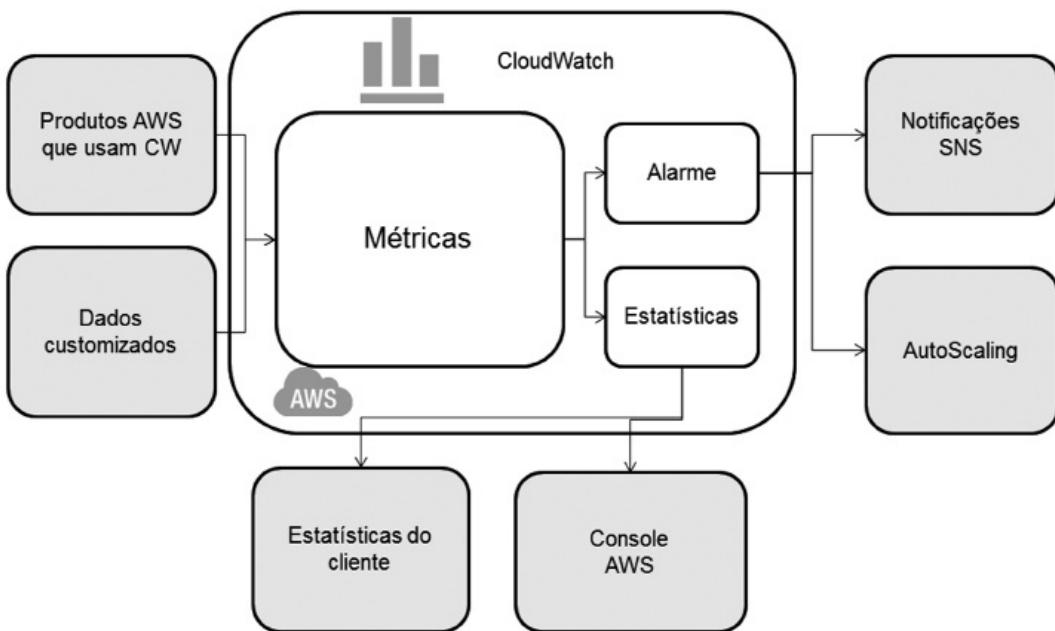


Figura 9-3 Arquitetura do CloudWatch

Dados de métricas para qualquer instância EC2 podem ser recuperados até duas semanas a partir do início do monitoramento. Após duas semanas, os dados de métricas para uma instância EC2 não estarão disponíveis se o gerenciamento tiver sido desabilitado para aquela instância EC2. Se for necessário arquivar métricas por mais de duas semanas, isto pode ser feito ao chamar o comando *mon-get-stats* e armazenar os resultados no S3.

Para usar o CW, basta selecionar as instâncias EC2 que devem ser monitoradas. É possível fornecer dados de métrica oriundos do próprio negócio ou dos aplicativos. O CW começará agregando e armazenando dados de monitoramento que podem ser acessados usando as APIs do web service ou ferramentas de linha de comando.

O CW permite monitorar recursos da AWS e de aplicativos que os clientes executam na AWS. Desenvolvedores e administradores de sistema podem utilizá-lo para coletar e monitorar métricas, obter *insights* e reagir imediatamente a uma situação real para manter seus aplicativos e negócios funcionando. Com o CloudWatch, obtém-se visibilidade integral do sistema, da utilização de recursos, do desempenho de aplicativos e do status operacional.

### 9.3.1. Conceitos

Alguns conceitos mostrados a seguir são fundamentais para o entendimento do papel do CloudWatch.

- **Métrica:** é um conceito fundamental para o CW e representa um conjunto de *datapoints* ordenados no tempo.
- **Namespace:** é um container conceitual para a métrica. Métricas no espaço de *namespaces* são isoladas umas das outras, para que métricas de aplicativos diferentes não sejam agregadas por engano nas mesmas estatísticas.
- **Dimensão:** é um par de nome/valor que ajuda a identificar com exclusividade uma métrica. Cada métrica tem características específicas que a descrevem, pode-se pensar em dimensões como categorias para essas características. As dimensões ajudam a criar uma estrutura conceitual para o plano de estatísticas. Como dimensões fazem parte do identificador exclusivo de uma métrica, sempre que for adicionado um par nome/valor exclusivo a uma métrica, uma nova métrica está sendo criada.
- **Time stamp:** retorna o carimbo do tempo a partir de um par de nome/valor. Cada *datapoint* deve ser marcado com um *time stamp*, que pode ser de até duas semanas no passado e até um dia no futuro. Se não for providenciado o *time stamp*, o CW o cria com base no tempo em que o dado foi recebido.
- **Unidade:** representa a unidade da estatística de medida. Por exemplo, a unidade para a métrica *NetworkIn* do EC2 é bytes porque *NetworkIn* controla o número de bytes que o EC2 recebe de uma instância em todas as interfaces de rede. As seguintes unidades são suportadas pelo CW:
  - *Seconds*
  - *Bytes*
  - *Bits*
  - *Percent*
  - *Count*
  - *Bytes/seconds*
  - *Bits/seconds*
  - *Count/second*
  - *None (default)*

▪ **Estatísticas:** são agregações de dados métricos durante um período especificado de tempo. O CW fornece estatísticas com base em *datapoints* métricos definidos pelo

usuário ou por produtos AWS. As agregações são feitas usando *namespace*, nome da métrica, dimensões e unidades de *datapoints* medidos, no período de tempo especificado. Possíveis estatísticas são:

- *Minimum*
- *Maximum*
- *Sum*
- *Average*
- *SampleCount*

• **Período:** é o

comprimento de tempo

associado com uma estatística CW específica. Cada estatística representa uma agregação de dados de métricas coletadas para um período específico de tempo. No CW pode-se ajustar como os dados são agregados, variando o comprimento do período. Pode-se utilizar um período curto como um minuto (sessenta segundos) ou duas semanas (1.209.600 segundos).

- **Agregação:** o CW agrupa estatísticas de acordo com o período de duração especificado em chamadas do tipo *GetMetricStatistics*.
- **Alarmes:** o CW é especialmente útil porque ajuda a tomar decisões e ações imediatas, automáticas, com base em dados. Alarmes podem automaticamente iniciar ações em seu nome, com base nos parâmetros especificados.

### 9.3.2. Distribuição das principais métricas

As principais métricas do CloudWatch são:

- **Monitoramento básico de instâncias EC2:** sete métricas pré-selecionadas a uma frequência de cinco minutos, grátis.
- **Monitoramento detalhado de instâncias EC2:** sete métricas pré-selecionadas a uma frequência de um minuto, por um custo adicional.
- **Volumes EBS:** oito métricas pré-selecionadas a uma frequência de cinco minutos, grátis.
- **Elastic Load Balancers:** quatro métricas pré-selecionadas a uma frequência de um minuto, grátis.
- **Instâncias de banco de dados RDS:** treze métricas pré-selecionadas a uma frequência de um minuto, grátis.
- **Buscas SQS:** sete métricas pré-selecionadas a uma frequência de

cinco minutos, grátis.

- **Tópicos SNS:** quatro métricas pré-selecionadas a uma frequência de cinco minutos, grátis.
- **Nós do ElastiCache:** 29 métricas pré-selecionadas a uma frequência de um minuto, grátis.
- **Tabelas do DynamoDB:** sete métricas pré-selecionadas a uma frequência de cinco minutos, grátis.
- **AWS Storage Gateway:** onze métricas de gateway pré-selecionadas e cinco métricas de volume de armazenamento pré-selecionadas a uma frequência de cinco minutos, grátis.
- **Fluxos de trabalho do *Elastic MapReduce*:** 23 métricas pré-selecionadas a uma frequência de cinco minutos, inteiramente grátis.
- **Grupos do *Auto Scaling*:** sete métricas pré-selecionadas a uma frequência de um minuto, opcional e cobradas pelo preço padrão.
- **Estimativa de cobrança da fatura:** pode-se optar por habilitar as métricas para monitorar cobranças da AWS. O número de métricas depende dos produtos e serviços da AWS utilizados.

Essas métricas não estão disponíveis em todas as regiões.

### 9.3.3. Interfaces

O CW pode ser acessado usando várias interfaces diferentes – pelo console de gerenciamento do AWS, fazendo download e instalando a interface de linha de comando (*Command Line Interface* – CLI), ou criando uma solicitação de consulta com a API QUERY.

A **Figura 9-4** ilustra a página principal do CloudWatch cujo acesso é via console de gerenciamento AWS.

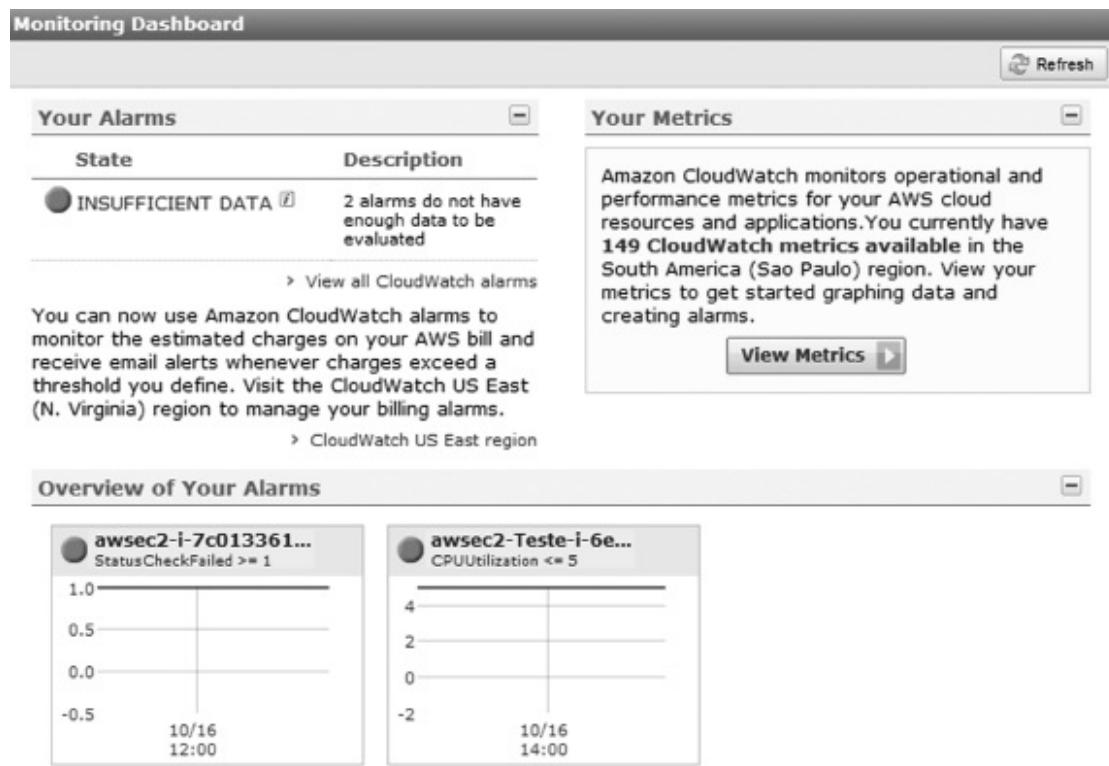


Figura 9-4 Tela inicial do CloudWatch

### 9.3.4. Utilização das métricas

Se existe um registro para o serviço da AWS, já se está automaticamente registrado para o CW. Todas as instâncias do EC2 são automaticamente habilitadas para o monitoramento básico (serviço gratuito).

O CW também coleta métricas automaticamente para volumes EBS, *Elastic Load Balancers* e instâncias de banco de dados RDS gratuitamente.

### 9.3.5. Exibindo métricas e criando alarmes

#### 9.3.5.1. Exibindo métricas

É possível visualizar gráficos das métricas clicando na aba “Metrics” no painel de navegação do CW ou em “View Metrics”, na opção “Your Metrics”. A primeira opção teria a seguinte sequência:

- No painel de navegação, clique em “Metrics”.
- Role para baixo até a métrica desejada.
- Clique na métrica.

A **Figura 9-5** mostra o gráfico para a métrica *CPUUtilization*, agregada pelo ID da imagem do EC2.

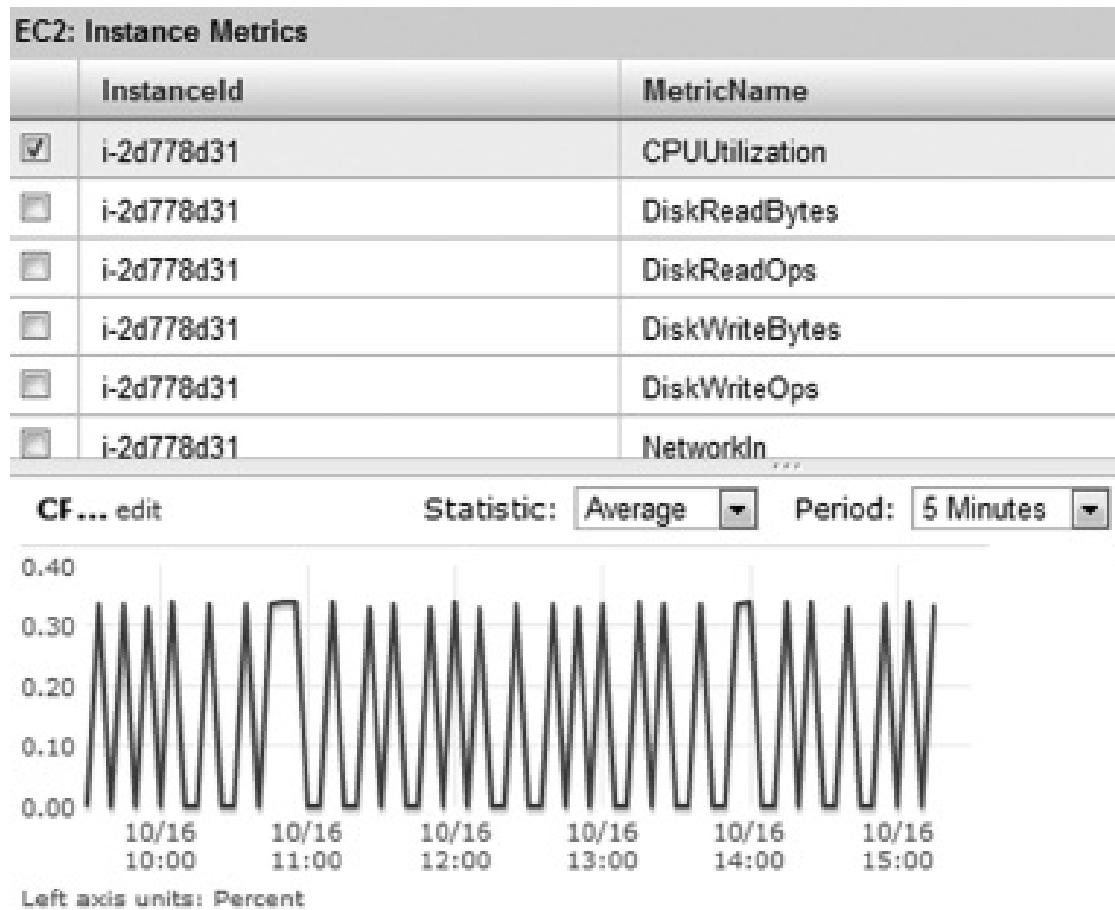


Figura 9-5 Exibindo suas métricas

### 9.3.5.2. Criando alarmes

É possível criar um alarme na página “Your Alarms”.

Para criar um alarme a sequência seria:

- Abra o console CW em <https://console.aws.amazon.com/cloudwatch/>.
- No painel de navegação, clique em “Alarms”.
- Clique em “Create Alarm”. A página do *wizard* para a criação do alarme abre.

A **Figura 9-6** ilustra a caixa de diálogo “Create Alarm Wizard”.

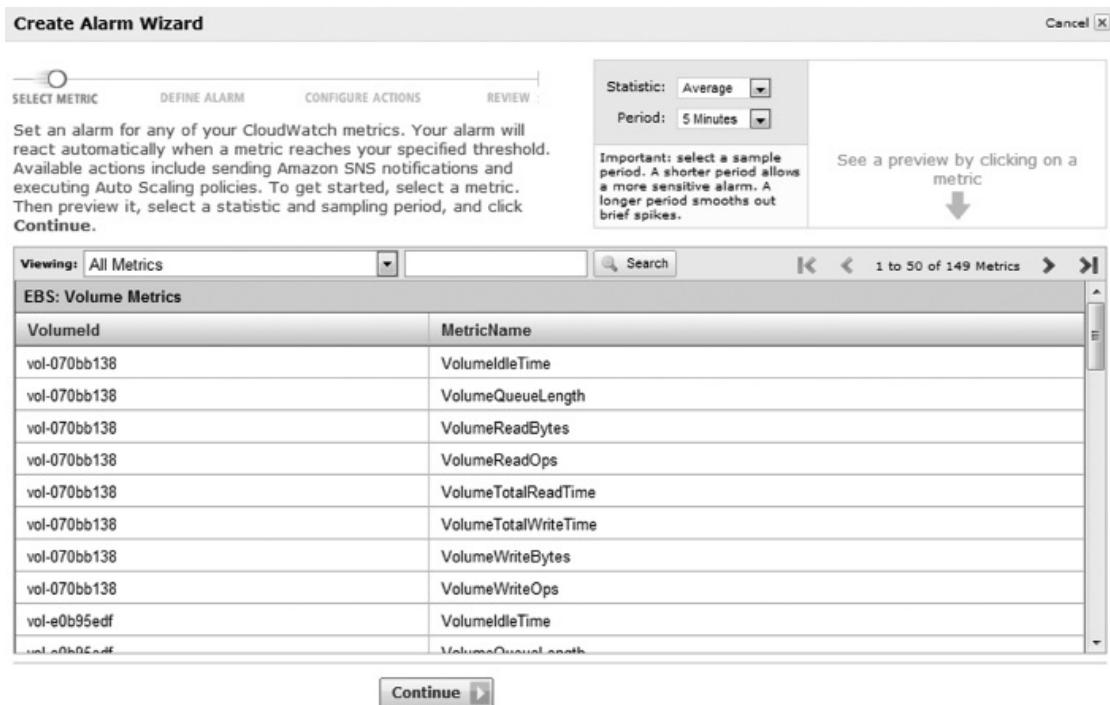


Figura 9-6 Create Alarm Wizard

### 9.3.6. Métricas

O CW oferece monitoração básica para vários produtos da AWS. Monitoramento básico significa que um serviço envia *datapoints* para o CW a cada cinco minutos. Monitoramento detalhado significa que um serviço envia *datapoints* para o CW a cada minuto.

O CW fornece monitoramento básico para os seguintes serviços:

- O EC2 começa a enviar estatísticas básicas automaticamente quando se inicia qualquer instância (incluindo AMIs pagas). Pode-se optar por realizar um monitoramento detalhado.
- O EBS começa a enviar estatísticas básicas automaticamente quando se monta um volume.
- O ELB começa a enviar estatísticas quando se passa a usá-lo.
- O EMR começa a enviar estatísticas básicas automaticamente quando se cria um fluxo de trabalho.
- O *Auto Scaling* começa a enviar estatísticas quando se passa a usá-lo. Pode-se optar por realizar um monitoramento detalhado.
- O RDS começa a enviar estatísticas específicas quando se passa a usá-lo.
- O DynamoDB passa a enviar estatísticas básicas automaticamente quando se começa a usá-lo.

#### 9.3.6.1. Monitoração de estatísticas agregadas

O CW permite monitorar estatísticas agregadas. Um cenário específico poderia mostrar como usar o console de gerenciamento AWS, a API *GetMetricStatistics* ou o comando *get-seg-stats* para obter o uso médio da CPU para todas as instâncias do EC2 em uma determinada conta. O CW retornaria estatísticas para todas as dimensões no *namespace* AWS/EC2.

Para exibir a utilização média da CPU para todas as instâncias do EC2:

- Abra o console CW em <https://console.aws.amazon.com/cloudwatch/>.
- No painel de navegação, clique em “Metrics”.
- Selecione “EC2: Aggregated across Instance” na lista que aparece. Surgem as métricas disponíveis para todas as instâncias.
- Selecione a linha que contém o *CPUUtilization*. É exibido um gráfico mostrando *CPUUtilization* para todas as instâncias do EC2.

#### 9.3.6.2. Monitoração de estatísticas específicas

O CW permite monitorar estatísticas específicas. Um cenário específico seria o de descrever como usar o console de gerenciamento do AWS, o comando *get-seg-stats* ou a API *GetMetricStatistics* para determinar a máxima utilização de CPU de uma instância específica do EC2.

Para exibir a utilização média da CPU para uma instância específica:

- Abra o console CW em <https://console.aws.amazon.com/cloudwatch/>.
- No painel de navegação, clique em “Metrics”.
- Selecione “EC2: Instance Metrics” na lista que aparece. Surgem as métricas específicas para instâncias individuais.
- Selecione a linha que contém a *CPUUtilization*. É exibido um gráfico mostrando *CPUUtilization* para uma instância específica do EC2.

#### 9.3.6.3. Descrição das métricas EC2

- ***CPUUtilization***: porcentagem de unidades de computação EC2 alocadas que estão atualmente em uso na instância. Esta métrica identifica o poder de processamento necessário para executar um aplicativo em uma instância selecionada. Unidades: *Percent*.
- ***DiskReadOps***: operações de leitura de todos os discos efêmeros disponíveis para a instância. Esta métrica identifica a taxa na qual um aplicativo lê um disco. Esta métrica pode ser usada para determinar a velocidade com que um aplicativo lê dados de um disco rígido. Unidades: *Count*.
- ***DiskWriteOps***: operações de gravação para todos os discos

efêmeros disponíveis para a instância. Esta métrica identifica a taxa na qual um aplicativo grava em um disco rígido. Esta métrica pode ser usada para determinar a velocidade com que um aplicativo salva os dados em um disco rígido. Unidades: *Count*.

- **DiskReadBytes:** bytes lidos de todos os discos efêmeros disponíveis para a instância. Essa métrica é utilizada para determinar o volume de dados que o aplicativo lê do disco rígido da instância. Esta métrica pode ser usada para determinar a velocidade do aplicativo. Unidades: *Bytes*.
- **DiskWriteBytes:** bytes gravados de todos os discos efêmeros disponíveis para a instância. Essa métrica é utilizada para determinar o volume de dados que o aplicativo grava para o disco rígido da instância. Esta métrica pode ser usada para determinar a velocidade do aplicativo. Unidades: *Bytes*.
- **NetworkIn:** número de bytes recebidos em todas as interfaces de rede por instância. Esta métrica identifica o volume de tráfego de rede de entrada para um aplicativo em uma única instância. Unidades: *Bytes*.
- **NetworkOut:** número de bytes enviados em todas as interfaces de rede por instância. Esta métrica identifica o volume de tráfego de rede de saída para um aplicativo em uma única instância. Unidades: *Bytes*.

#### 9.3.6.4. Monitoramento básico e avançado do EC2

Para instâncias EC2, o monitoramento básico do CW coleta e registra métricas de utilização de CPU, transferência de dados e atividade de uso de disco de cada instância EC2 em uma frequência de cinco minutos.

O monitoramento básico já está habilitado automaticamente para todas as instâncias do EC2, e essas métricas podem ser acessadas no console do EC2 ou no console do CW.

O monitoramento detalhado do CW mostra essas mesmas métricas em intervalos de um minuto e também habilita a agregação de dados pela AMI ID do EC2 e pelo tipo de instância. Para habilitá-lo deve-se:

- Entrar no console de gerenciamento da AWS.
- No console do EC2, clique no botão “Launch Instance”.
- Selecionar uma AMI para iniciar uma instância, selecionar o par de chaves e configurar o *firewall*.
- Clicar na caixa de seleção “Enable CloudWatch detailed monitoring for the instance” (ativar monitoramento detalhado do CloudWatch desta instância).

- As próximas etapas são idênticas às descritas no capítulo 5 para o lançamento de instâncias EC2. A **Figura 9-7** ilustra esta operação.

**Request Instances Wizard**

CHOOSE AN AMI    INSTANCE DETAILS    CREATE KEY PAIR    CONFIGURE FIREWALL    REVIEW

**Number of Instances:** 1      **Availability Zone:** No Preference

**Advanced Instance Options**

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

**Kernel ID:**       **RAM Disk ID:**

**Monitoring:**  Enable CloudWatch detailed monitoring for this instance  
(additional charges will apply)

**User Data:**  
 as text  
 as file  
 base64 encoded

**Termination Protection:**  Prevention against accidental termination.      **Shutdown Behavior:**

**IAM Role:**  If enabled, you will not be able to terminate the instances using the API or the Console until Termination Protection has been disabled.

---

< Back      **Continue** >

Figura 9-7 Detailed Monitoring

Também é possível ativar o monitoramento detalhado de uma instância já em operação ao clicar com o botão direito do mouse na instância na console do EC2 e selecionar “Enabled Detailed Monitoring”. A **Figura 9-8** ilustra esta opção.

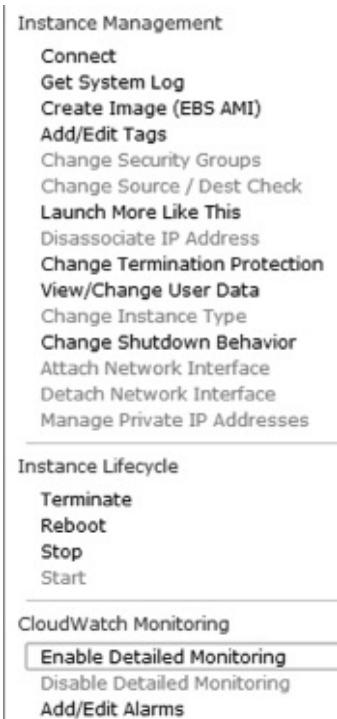


Figura 9-8 Enable detailed monitoring

### 9.3.6.5. Monitoramento de métricas personalizadas

O CloudWatch também pode monitorar métricas geradas pelos aplicativos que são executados utilizando recursos da AWS. Através de uma simples chamada PUT API, pode-se imediatamente enviar e armazenar qualquer métrica que seja importante para o desempenho do negócio ou para o desempenho operacional do aplicativo. É possível enviar a quantidade de dados de métricas personalizadas necessárias de acordo com as necessidades.

As mesmas funcionalidades do CW estarão disponíveis em uma frequência de até um minuto para dados personalizados, incluindo estatísticas, gráficos e alarmes.

### 9.3.7. Alarmes

Pode-se criar um alarme do CW que envia uma mensagem ao SNS quando a métrica muda de estado. Um alarme escuta uma única métrica ao longo de um período de tempo especificado e executa uma ou mais ações com base no valor da métrica em relação a um determinado limite ao longo de um período de tempo.

A ação é uma notificação enviada ao SNS ou a uma política de *Auto*

*Scaling*. Alarmes invocam ações para alterações sustentadas de estado. Alarmes não são invocados simplesmente porque eles estão em um estado particular; o estado deve ter sido mudado e deve ser mantido por um número específico de períodos.

Depois que um alarme invoca uma ação devido a uma alteração no estado, o seu comportamento subsequente depende do tipo de ação associada com o alarme. Para notificações de política baseada em *Auto Scaling*, o alarme continua a chamar a ação para cada período que permanece no novo estado. Para notificações de SNS, nenhuma ação adicional é invocada.

#### 9.3.7.1. Estados dos alarmes

Um alarme tem três possíveis estados:

- **OK**: a métrica está dentro do limite definido.
- **ALARM**: a métrica está fora do limite definido.
- **INSUFFICIENT\_DATA**: o alarme foi iniciado, a métrica não está disponível ou não há dados suficientes disponíveis para a métrica determinar o estado de alarme.

A **Figura 9-9** ilustra um resultado gráfico unidades *versus* períodos de tempo de uma determinada métrica para demonstrar o funcionamento do alarme. O limite (*threshold*) de alarme é definido como 3 e a ruptura mínima é de três períodos. Ou seja, o alarme chama a ação somente quando o limite é rompido por três períodos consecutivos. Na figura, isso acontece do terceiro ao quinto período de tempo, e o estado do alarme é definido como "ALARM". No período 6, o valor cai abaixo do limite e o estado reverte para "OK". Mais tarde, durante o nono período de tempo, o limite é rompido novamente, mas não para três períodos consecutivos. Consequentemente, o estado do alarme permanece "OK".

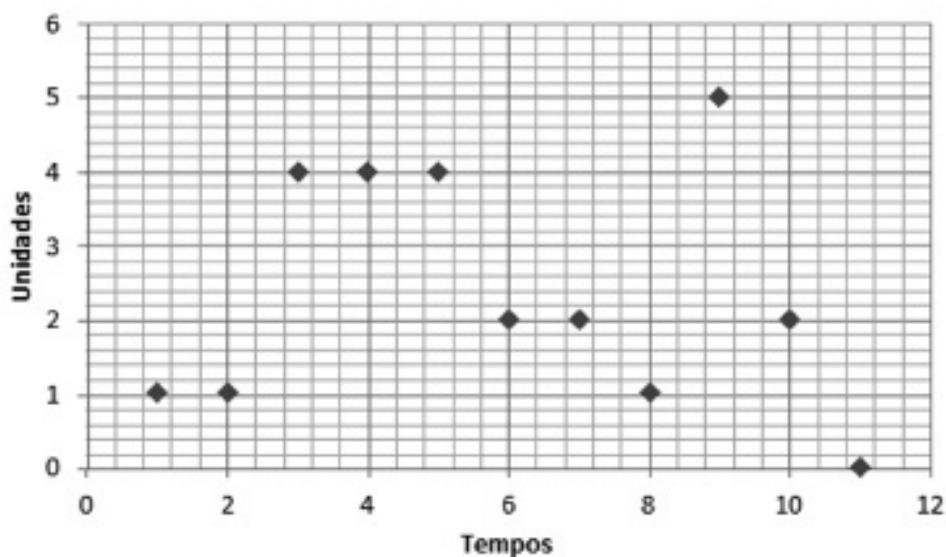


Figura 9-9 Unidades *versus* períodos de tempo

Pode-se criar um alarme já no lançamento da instância, conforme ilustra a **Figura 9-10**.

**Create Alarm for i-0fe62313** Cancel

You can use CloudWatch alarms to be notified automatically whenever any status check for this instance fails.  
To create an alarm, first choose whom to notify and then define when the notification should be sent.

**Send a notification to:**   create topic

---

**Whenever:**

**Is:** Failing

**For at least:**  consecutive period(s) of

---

**Name this alarm:**

---

Figura 9-10 Create alarm

### 9.3.7.2. Características comuns dos alarmes

- Pode-se criar até quatrocentos alarmes por conta da AWS. Para criar ou atualizar um alarme, utiliza-se a API *PutMetricAlarm*.
- Pode-se listar qualquer ou todos os alarmes configurados atualmente e listar os alarmes em um estado particular usando a API *DescribeAlarms*. Pode-se ainda filtrar a lista por intervalo de tempo.
- Pode-se desativar e ativar alarmes usando as APIs *DisableAlarmActions* e *EnableAlarmActions*.
- Pode-se testar um alarme definindo-o para qualquer estado usando a API *SetAlarmState*. Esta mudança de estado temporário dura apenas até a próxima comparação de alarme.
- Pode-se ver o histórico do alarme usando a API *DescribeAlarmHistory*. O CW preserva o histórico do alarme por duas semanas. Cada transição de estado é marcada com um *time stamp* único.

### 9.3.7.3. Configurar SNS

Pode-se também configurar o e-mail para alarmes dos mais diversos ou mesmo para notificação SNS. Para o exemplo será criado um alarme para uso da CPU acima de 70%. A **Figura 9-11** define o alarme.

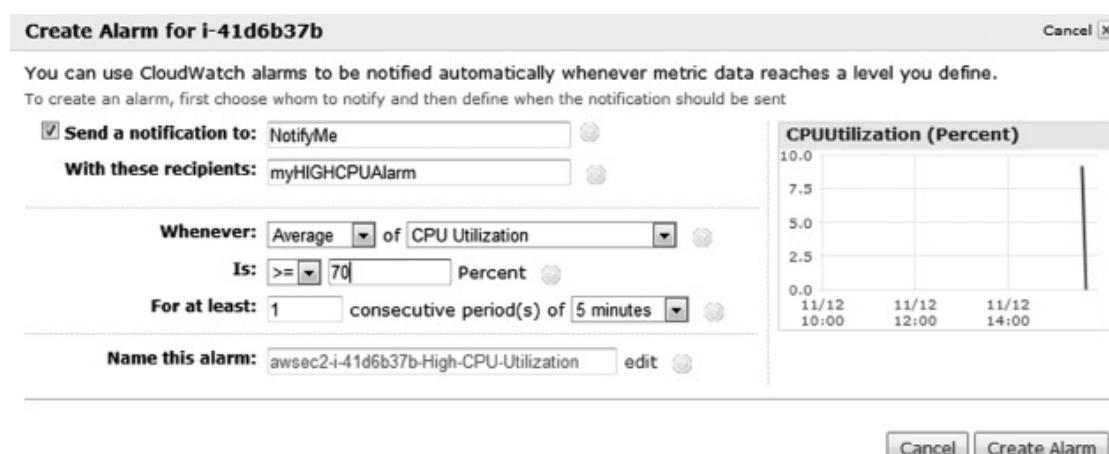


Figura 9-11 Define alarm

Pode-se criar um tópico SNS no passo “Configure Actions to Create Alarm Wizard” e cadastrar um e-mail para receber o alarme. A **Figura 9-12** ilustra a criação do tópico SNS.

Defina o tópico, por exemplo, *MyHIGHCPUAlarm* e cadastre o e-mail que receberá o alerta.

When Alarm state is	Take action	Action details
ALARM	Send Notification	Topic: myHIGHCPUAlarm Email(s): manoel@manoelveras.com.br <input type="button" value="ADD ACTION"/>
A topic is a communication channel that can be reused across Send Notification actions. Please enter a new topic name and a list of comma-separated email addresses.		

Figura 9-12 Configure action

### 9.3.8. Monitoramento e alarmes das taxas de cobrança

Antes de criar um alarme para obter os custos estimados, deve-se habilitar o monitoramento. Quando o monitoramento for ativado, será possível acompanhar as taxas estimadas. Isso cria dados de métrica que podem ser utilizados para criar um alarme de faturamento. Depois de habilitar a métrica de faturamento não é permitido desativar o recolhimento dos dados, mas é possível excluir os alarmes que foram criados.

Para habilitar o monitoramento dos encargos estimados:

- Vá ao site da Amazon Web Services em <http://aws.amazon.com>.
- Clique em “My Account” e, em seguida, clique em “Account Activity”.
- No espaço fornecido, digite seu nome de usuário e senha e clique em “Sign in using our secure server”.
- No âmbito de atividade da conta, no box “Monitor your estimate charges”, clique em “Enable Now”.

Os alarmes podem ser definidos usando o console CW ou a interface de linha de comando (CLI).

A **Figura 9-13** ilustra a caixa de diálogo para criação do alarme de faturamento.



Figura 9-13 Alarm details for AWS billing estimated charges

Pode-se criar um alarme de faturamento na página “Account Activity” ou utilizando o console do CW. Quando o monitoramento da taxa estimada é ativado pela primeira vez, demora cerca de quinze minutos antes de se poder ver os dados de faturamento.

Na **Figura 9-14** definem-se os principais parâmetros para a criação do

alarme no console de gerenciamento. O exemplo cria um alarme que vai enviar uma mensagem de e-mail quando as taxas estimadas de uso do AWS excederem US\$ 50.

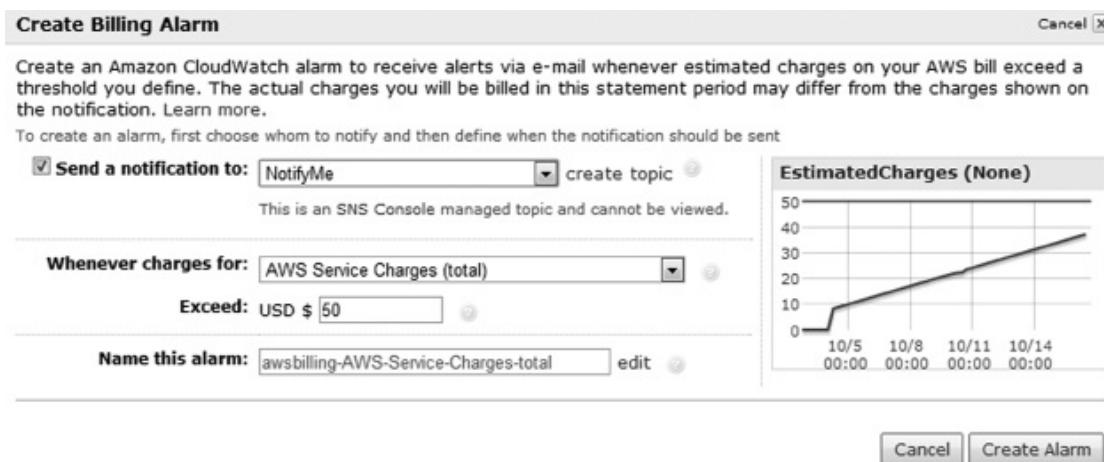


Figura 9-14 Create billing alarm

## 9.4. Referências bibliográficas

**Amazon CloudWatch Developer Guide.** API Version 2010-08-01.

**Amazon Web Services. Auto Scaling.** API version 2011-01-01, 2011.

**Amazon Web Services. Elastic Load Balancing:** gsg. API version 2011-11-15, 2011.

## ***PARTE III – ASPECTOS AVANÇADOS***

---

# 10. Arquitetura

## 10.1. Introdução

Este capítulo trata das melhores práticas para o desenho da arquitetura. Trata de mostrar como construir a nova arquitetura em um ambiente de nuvem. Como referência para o capítulo utilizou-se o artigo do Jinesh Varia “Projetando para a nuvem: Práticas recomendadas”, traduzido para o português e publicado pela AWS.

## 10.2. Conceito

Arquitetura empresarial (*enterprise architecture*) consiste na lógica organizacional dos processos de negócio e da TI, refletindo os requisitos de integração e padronização do modelo operacional da organização. Ela é formada pela arquitetura do negócio e da TI. A arquitetura de TI, por sua vez, é formada por três blocos: aplicação, dados e tecnologia.

Neste capítulo o foco é a arquitetura da infraestrutura, um subconjunto da arquitetura tecnológica, mesmo que em alguns casos sejam feitas considerações sobre a arquitetura do software (aplicação).

A **Figura 10-1** ilustra essas relações.

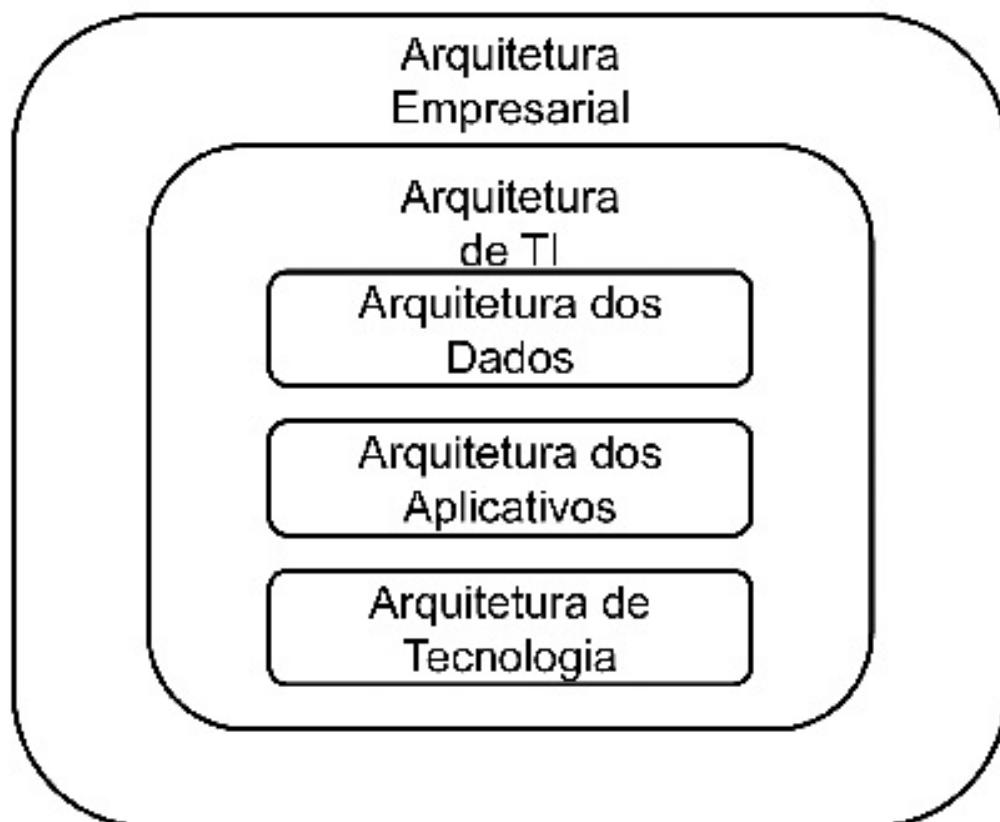


Figura 10-1 Arquiteturas

A arquitetura da infraestrutura trata de ordenar os componentes da infraestrutura, otimizar as relações entre os componentes e otimizar suas propriedades para obter maiores benefícios para o negócio.

A **Figura 10-2** ilustra simbolicamente a arquitetura da infraestrutura.

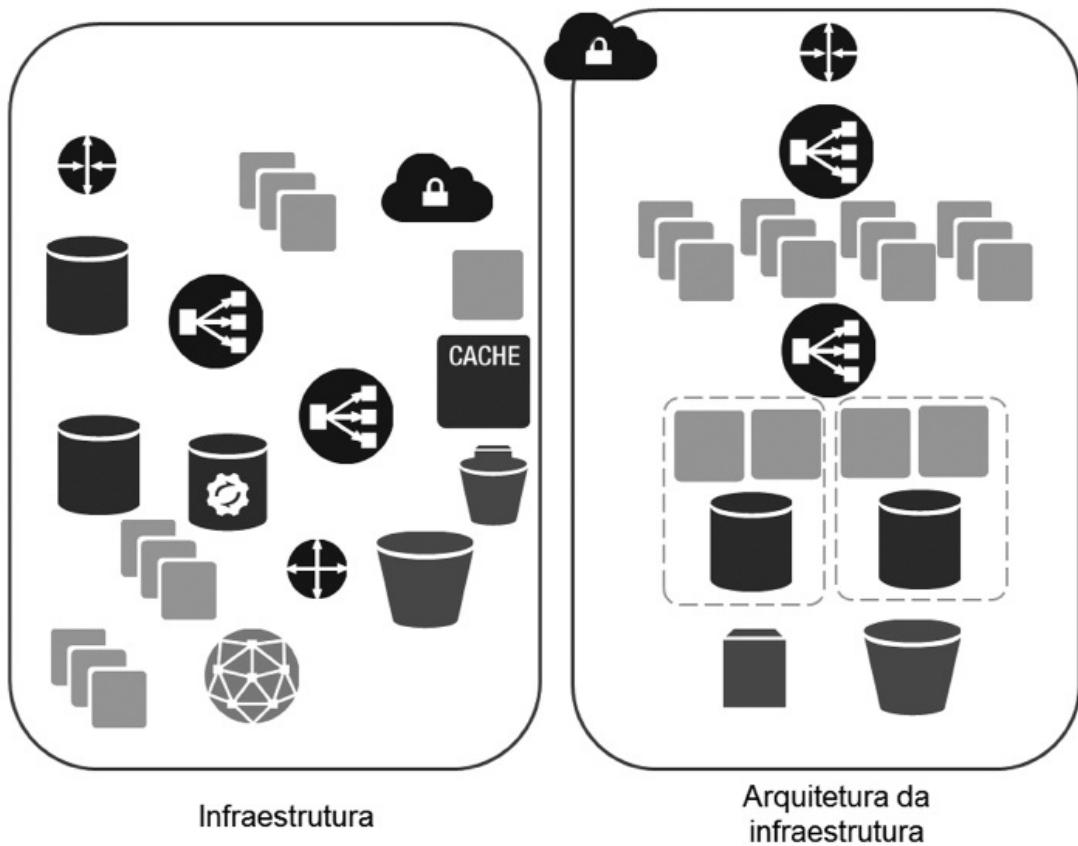


Figura 10-2 Infraestrutura e arquitetura da infraestrutura

Uma peça-chave da arquitetura é a sua capacidade de ser elástica. Neste capítulo apresenta-se o *Auto Scaling*, uma espécie de funcionalidade do EC2 que consiste em um conjunto de ferramentas que provê escalabilidade automática para instâncias EC2.

### 10.3. Arquitetura tradicional *versus* arquitetura AWS

Os aplicativos foram projetados tradicionalmente para utilizar uma infraestrutura de hardware fixa, rígida e pré-provisionada. Foi sempre assim. As empresas nunca tiveram a necessidade de provisionar e instalar servidores diariamente, como agora. Os negócios atualmente demandam agilidade, e a TI precisa prover sistemas que permitam que a empresa se reconfigure de forma rápida em função das mudanças do ambiente. O tempo de provisionamento até a aquisição de novos recursos era demasiadamente elevado, os arquitetos de software nunca investiram tempo e recursos na otimização do hardware e simplesmente aceitavam esta situação. A noção de “elasticidade” dentro de

uma arquitetura foi esquecida porque a ideia de ter novos recursos diariamente não era tarefa trivial [VARIA, 2011].

O DATACENTER é a peça-chave da infraestrutura de TI. Além da TI, considera-se também parte do DATACENTER as instalações físicas, incluindo energização e resfriamento, conforme mostrado no capítulo 2.

O provisionamento do DATACENTER sempre foi feito no limite, todos os aspectos de construção, infraestrutura e a própria infraestrutura de TI eram provisionados no máximo da capacidade. O provisionamento do DATACENTER para cima era uma forma de fugir das dificuldades em realizar o planejamento da capacidade (*capacity planning*). Isto trazia uma perda de eficiência e um custo extremamente elevado para a operação como um todo.

A modularidade foi introduzida recentemente tanto para os aspectos da construção como para os componentes da solução para melhorar estes aspectos. De fato, novos projetos que utilizam o conceito de modularidade são muito mais eficientes e sustentáveis do que projetos de dez anos atrás. Ainda assim, está longe de ser uma solução ideal.

A **Figura 10-3** ilustra a ideia de modularidade. Observe que a modularidade vale tanto para a infraestrutura (TO) como para a TI.

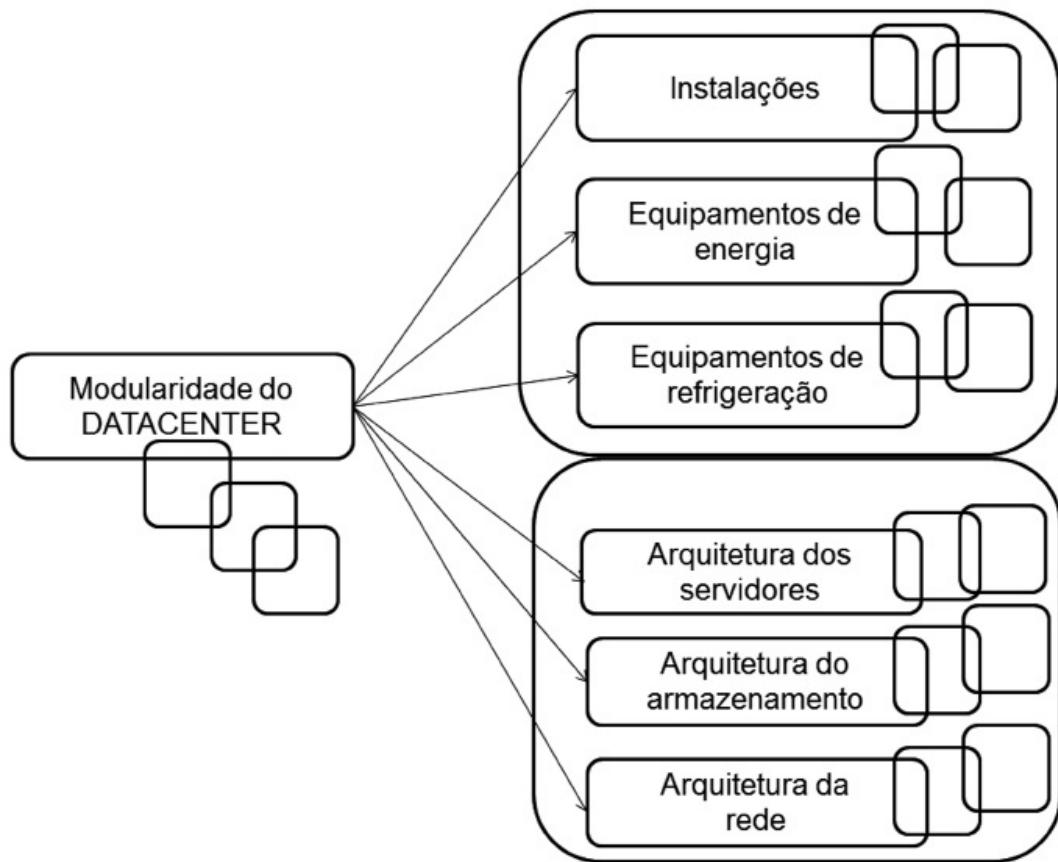


Figura 10-3 Modularidade do DATACENTER

A **Figura 10-4** ilustra o desenho da infraestrutura de TI do DATACENTER convencional. Esta é um arquitetura de infraestrutura de três camadas muito

comumente encontrada em DATACENTERS empresariais e em provedores de serviços de internet.

Os componentes mostrados na figura são projetados para serem utilizados de forma convencional. Qualquer aumento não planejado de capacidade implica em novas aquisições, atualizações de contratos de licenciamento de software, atualizações de versões, etc.

Servidores web podem ser adquiridos com maior capacidade, de forma a atender *requests* em dias de pico. Mas quase sempre esta estratégia resulta em capacidade ociosa. Servidores de aplicação também precisam ser dimensionados considerando os picos de utilização, o que não é tarefa trivial. Servidores de cache otimizam o uso dos servidores de banco de dados e fazem com que boa parte das requisições sejam atendidas em cache. Dados são armazenados na unidade de storage e posteriormente salvos em unidades de fita, de acordo com política preestabelecida. Balanceadores de carga podem ser utilizados na camada web e na camada de aplicação, conforme sugerido.

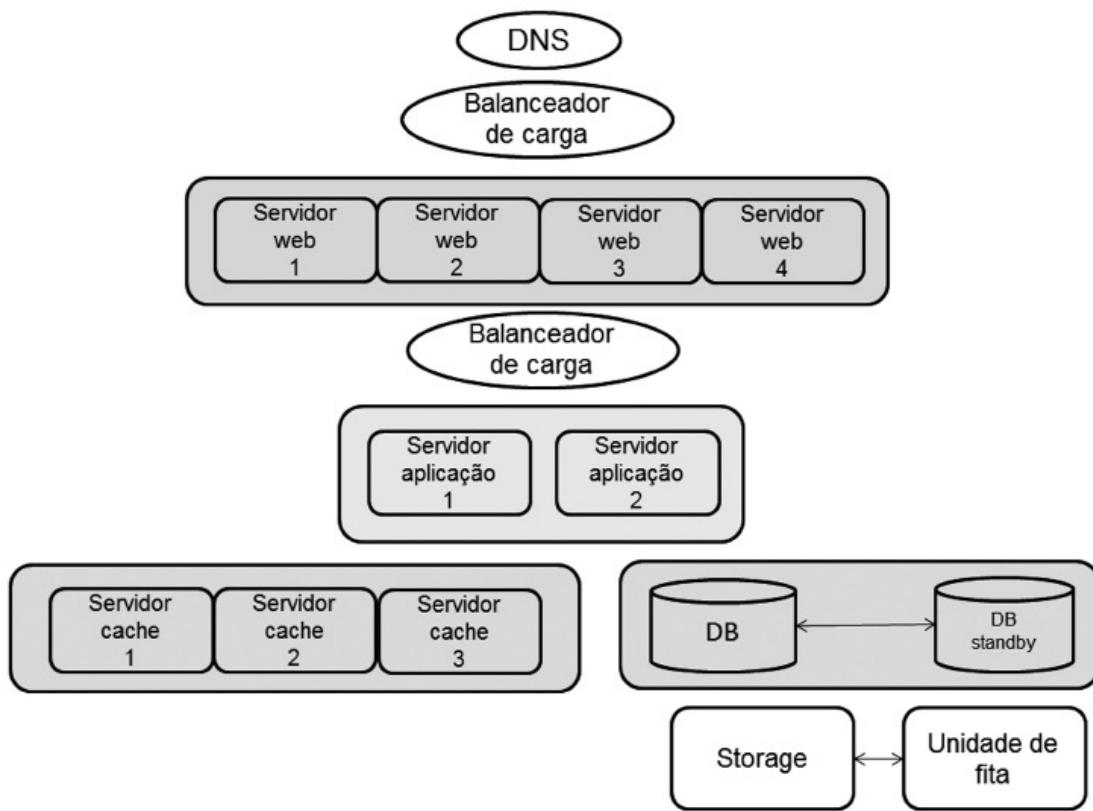


Figura 10-4 Desenho da infraestrutura de TI do DATACENTER

Esta arquitetura suporta o modelo de três camadas utilizado para construção de uma aplicação. Ele é baseado em uma camada de apresentação (*presentation tier* – pt), uma camada de regras de negócio (*business tier* – bt) e uma camada de dados (*data tier* – dt). A **Figura 10-5** ilustra o modelo de três camadas para a aplicação.

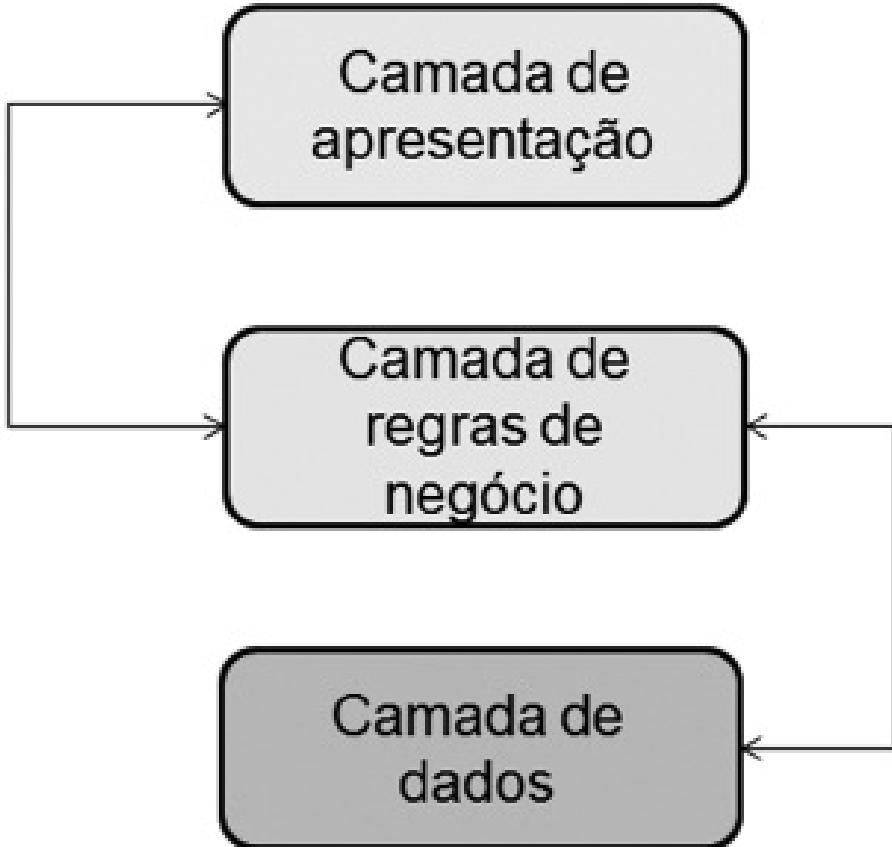


Figura 10-5 Modelo de aplicação de três camadas

Com a nuvem, a forma de construir a infraestrutura de TI e os aplicativos muda. As infraestruturas do DATACENTER envolvendo instalações, energia, refrigeração passam a ser de responsabilidade do provedor de serviços de nuvem. Cloud computing simplifica o processo de aquisição dos recursos necessários; não há nenhuma necessidade de fazer pedidos de hardware e software antes do tempo e de manter o hardware e o software guardados para uma nova necessidade. Em vez disso, os arquitetos de nuvem podem solicitar o que precisam poucas horas antes da necessidade, aproveitando a escala e o tempo de resposta rápida da nuvem. A ideia também se aplica para liberar os recursos desnecessários ou subutilizados quando não mais se precisam deles.

Se você não pode abraçar a mudança e implementar a elasticidade na sua arquitetura de aplicativo, você será incapaz de aproveitar a nuvem ao máximo. Assim como um arquiteto de nuvem, você deve raciocinar de maneira criativa, pensando de que maneira você pode implementar a elasticidade no seu aplicativo e otimizar o uso de recursos e, consequentemente, a conta no final do mês [VARIA, 2011].

Projetar arquiteturas inteligentes para nuvens elásticas, para que a infraestrutura seja executada somente quando necessária, é uma arte em si. A elasticidade deve ser um dos requisitos do projeto arquitetônico ou mesmo

uma propriedade do sistema.

Algumas perguntas se fazem necessárias:

- Quais os componentes ou que camadas em minha arquitetura de aplicativo podem se tornar elásticas?
- O que será necessário para tornar esse componente elástico?
- Qual será o impacto da implementação da elasticidade na arquitetura geral do sistema?

Varia adverte que quando você decidir mover os seus aplicativos para a nuvem e tentar mapear as suas especificações de sistema para aquelas disponíveis na nuvem, você vai notar que a nuvem pode não ter a especificação exata do recurso que você tem no local. Continuando, o autor citado alerta que você deve compreender que a nuvem fornece recursos abstratos, e eles se tornam poderosos quando você os combina com o modelo de provisionamento *on demand*. Você não deve ficar com receio e nem constrangido quando estiver utilizando recursos de nuvem, porque é importante compreender que, mesmo que você não tenha uma réplica exata do seu DATACENTER no ambiente da nuvem, você poderá obter outras vantagens para compensar essa necessidade.

A **Figura 10-6** ilustra o DATACENTER construído com recursos da AWS. Ele pode ser projetado, construído, utilizado e desligado quando necessário. Em poucas horas pode ser posto no ar. Imaginou a diferença quando comparado ao modelo convencional?

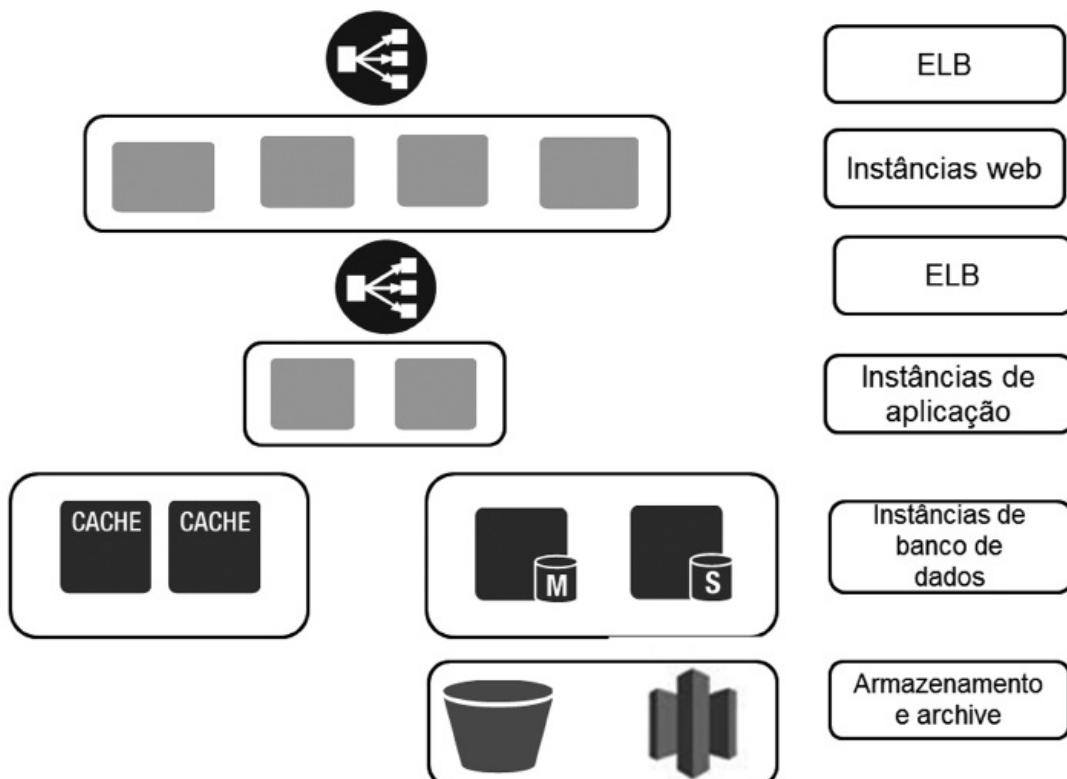


Figura 10-6 Desenho de DATACENTER AWS

O artigo “Building Scalable Applications in the Cloud”, da RightScale, dá uma série de dicas sobre como dimensionar o DATACENTER AWS mesmo sem utilizar o recurso de *Auto Scaling*, visto mais à frente. Principais sugestões do artigo:

- Utilizar instâncias *m1.large* que conseguem atender cinco mil *requests* por segundo. Um cluster com ELB, ou mesmo o HAProxy com quatro instâncias web, suporta vinte mil *requests* por segundo. O artigo sugere sempre utilizar pelo menos duas instâncias em balanceamento de carga em zonas de disponibilidade diferentes para redundância. No início do ciclo de vida da aplicação pode-se utilizar servidores web e de aplicação rodando na mesma instância, para efeito de redução de custos. Depois é possível remover o servidor de aplicação desta instância. Também pode-se iniciar com uma instância *small* ao invés de utilizar uma instância *large*.
- Servidores de aplicação podem ser colocados também aos pares em diferentes zonas de disponibilidade e devem ser configurados com alertas (*CPU Idle, Free Memory, System Load*) para que mantenham a escalabilidade. A recomendação é fazer o *scale up* de forma agressiva e o *scale down* de forma conservadora.
- Avaliar se a arquitetura da aplicação pode se beneficiar de uma camada de cache. Aplicações intensivas em leitura podem se aproveitar de uma estrutura de cache para otimizar o desempenho. Servidores de cache são intensivos em memória. O artigo sugere também que instâncias de cache estejam em zonas de disponibilidade separadas. Valores de *time-to-live* (TTL) também devem ser definidos. Aplicações com pequena demanda de cache podem utilizar soluções *memcached* residentes na mesma instância da aplicação.
- O artigo recomenda que a camada de banco de dados deve utilizar um ou mais bancos de dados *slave*. Reforça-se que bancos de dados *slave* devem residir em diferentes zonas de disponibilidade e podem ser promovidos usando scripts pré-configurados. O artigo aconselha que as instâncias de banco de dados utilizem volumes EBS, de preferência em uma configuração *striped*. Deve-se também realizar *snapshots* periódicos do banco de dados.
- A escalabilidade vertical pode ser feita facilmente em bancos de dados MySQL, conforme mencionado no capítulo 8. A escalabilidade horizontal pode ser alcançada com a utilização de múltiplos bancos de dados *slave*. A arquitetura para réplica de leitura foi vista no capítulo 8. Pode-se utilizar também a opção MySQL Proxy para implementação do mecanismo de escalabilidade horizontal. Neste

caso, o MySQL Proxy Client deve ser instalado nos servidores de aplicação.

## 10.4. Melhores práticas com AWS

Jinesh Varia sugere algumas melhores práticas para a arquitetura do DATACENTER na nuvem que reproduzo aqui.

Tabela 10-1 Melhores práticas com AWS

Melhor prática	Contexto
1. Prepare-se para falhas e nada falhará	Infraestrutura e aplicação
2. Separe componentes da aplicação	Aplicação
3. Implemente a elasticidade	Infraestrutura e aplicação
4. Pense paralelo	Aplicação
5. Ponha os dados no lugar certo	Infraestrutura
6. Utilize práticas recomendadas de segurança	Infraestrutura e aplicação

### 10.4.1. Prepare-se para falhas e nada falhará

#### Contexto: infraestrutura e aplicação

**Regra de ouro:** seja um pessimista ao projetar arquiteturas na nuvem; presuma que tudo irá falhar. Em outras palavras, sempre projete, implemente e implante arquiteturas considerando a falha.

Se for aceito que as coisas falham ao longo do tempo e for incorporado esse pensamento na arquitetura, poderão ser criados mecanismos para lidar com a falha antes que ela aconteça.

Tolerância a falhas pode ser caracterizada como a capacidade de um sistema de permanecer em operação mesmo que alguns dos componentes utilizados para construí-los venham a falhar. Vale para a infraestrutura e para o aplicativo.

No caso da infraestrutura, algumas questões precisam ser levantadas [VARIA, 2011]:

O que acontecerá se um nó em seu sistema falhar? Como você reconhece esta falha? Como eu faço para substituir o nó? Para quais tipos de cenários eu tenho que me planejar? Quais são meus pontos únicos de falha?

Se um balanceador de carga está disposto na frente de um pool de servidores de aplicativos, o que acontecerá se o balanceador de carga falhar? Se não houver mestre e escravos na arquitetura, o que acontecerá se o nó mestre falhar? Como é que ocorre o *failover* e como é que um novo escravo é instanciado e colocado em sincronia com o mestre?

Da mesma forma que projetar para falhas de hardware, deve-se projetar para falhas de software. No caso do aplicativo, algumas questões precisam ser levantadas:

O que acontecerá com o meu aplicativo se os seus serviços dependentes mudarem suas interfaces? E se o serviço de *downstream* expirar ou retornar uma exceção? E se as chaves de cache crescerem além do limite de memória de uma instância?

Existem práticas sugeridas pela AWS que ajudam no projeto:

1. **Enfrente um *failover* com tranquilidade usando EIPs:** é possível rapidamente remapear e fazer um *failover* para outro conjunto de servidores, para que o seu tráfego seja direcionado aos novos servidores. Isso funciona muito bem quando você deseja fazer atualizações para versões mais novas ou em caso de falhas de hardware (foi visto no capítulo 5). Endereços do tipo EIP podem ser mapeados para qualquer instância, independentemente da região. Os endereços são associados com uma conta. Podem ser “desatachados” de uma instância falha e mapeados para uma nova instância em um curto espaço de tempo. Essas operações podem ser feitas via uma chamada de API, linha de comando ou pelo console de gerenciamento AWS. A **Figura 10-7** ilustra o uso de EIPs em uma arquitetura de DATACENTER. O IP 1.1.1.2 utilizado por uma instância web com problema é aproveitado instantaneamente por outra instância web.

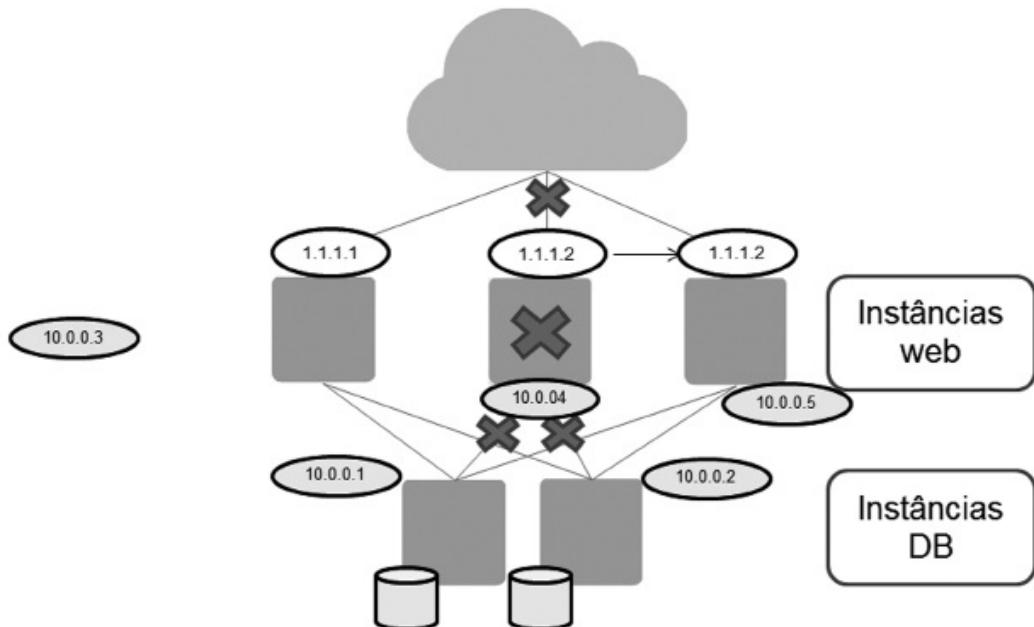


Figura 10-7 IP elástico (EIP) em operação

2. **Utilize várias zonas de disponibilidade:** zonas de disponibilidade são conceitualmente semelhantes aos DATACENTERS, conforme visto no capítulo 3. Ao implantar sua arquitetura em várias zonas de disponibilidade, você pode garantir alta disponibilidade. Utilize a funcionalidade *Multi-AZ* do RDS para replicar automaticamente atualizações de banco de dados em várias zonas.
3. **Mantenha uma AMI para que seja possível restaurar ambientes em uma zona de disponibilidade diferente:** construir aplicações tolerantes a falha é construir uma biblioteca de AMIs. Instâncias que utilizam a mesma AMI e que podem substituir uma instância com problema podem ser lançadas a qualquer momento através de uma chamada de API, linha de comando ou pelo console de gerenciamento. A **Figura 10-8** ilustra a operação que permite lançar uma instância idêntica à instância ativa feita pelo console de gerenciamento.

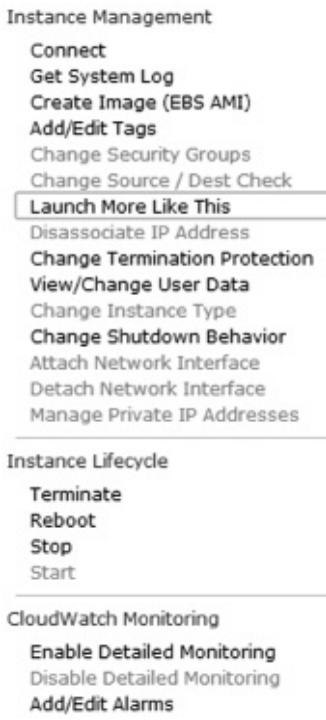


Figura 10-8 Lançando uma mesma instância EC2

- 4. Utilize o CloudWatch para monitoração e para tomar as medidas apropriadas em caso de degradação de desempenho ou falha de hardware:** configure um grupo de *Auto Scaling* para manter um tamanho fixo do cluster ELB, para que este substitua as instâncias EC2 com problemas por novas instâncias. A definição de regras para a operação do *Auto Scaling* é uma prática interessante para aplicações tolerantes a falha. Pode-se definir, por exemplo, que devem existir no mínimo duas instâncias (estratégia N+1) de uma AMI disponível. Quando usada em conjunto com a opção ELB, cada instância pode ter uma fração da carga de entrada. Os aspectos essenciais do gerenciamento foram tratados no capítulo 9.
- 5. Utilize o EBS e configure *snapshots* incrementais que sejam automaticamente enviados para o S3 e os dados sejam mantidos independentes de suas instâncias:** volumes EBS persistem independentemente do ciclo de vida da instância. EBSs guardam dados de forma redundante e, portanto, são mais duráveis do que discos rígidos convencionais. Se a instância falha, o volume EBS pode ser “atachado” a uma nova instância de forma simples. *Snapshots* EBS também podem ser criados de forma simples e podem ser utilizados para criar novos volumes que são uma réplica do volume original. O armazenamento na AWS foi visto no capítulo 6.
- 6. Utilize o RDS como sistema de banco de dados e defina o período de retenção para os backups, para que sejam realizados backups automatizados:** o capítulo 9 trata do banco de dados RDS.

- 7. Distribua o tráfego com o ELB:** distribua tráfego para suas aplicações através de várias instâncias EC2. Qualquer *request* enviado para o servidor DNS é delegado a um pool de instâncias EC2. O capítulo 8 trata do ELB.
- 8. Reserve capacidade na nuvem AWS:** esta é parte da estratégia de construir aplicações tolerantes a falha, pois, como se sabe, esses recursos são finitos. A opção e os preços de reservar a capacidade são tratados no capítulo 4.

#### **10.4.2. Separe componentes da aplicação**

##### **Contexto: aplicação**

A nuvem reforça o princípio SOA de design de que quanto mais separados estiverem os componentes do sistema, muito mais e de melhor maneira eles se dimensionarão e facilitarão a elasticidade. Lógico que aplicações já construídas e que não utilizaram este princípio representam um desafio para a elasticidade.

A chave é construir componentes que não sejam tão dependentes uns dos outros, para que, se um componente for desativado (falhar), estiver suspenso (não responder) ou permanecer ocupado (lento para responder) por algum motivo, os outros componentes do sistema sejam construídos, a fim de continuar a trabalhar como se nenhuma falha estivesse acontecendo. Basicamente, o acoplamento mais solto isola as várias camadas e componentes do aplicativo para que cada componente interaja de forma assíncrona com os outros e os trate como uma “caixa preta”.

Perguntas necessárias: qual componente de negócios ou recurso poderia ser isolado do aplicativo monolítico atual e ser executado de forma autônoma separadamente? E, em seguida, como posso adicionar mais instâncias desse componente sem quebrar meu sistema atual e ao mesmo tempo atender a mais usuários? Quanto esforço será necessário para encapsular o componente, para que ele possa interagir com outros componentes de forma assíncrona?

Jinesh Varia ressalta que a separação dos componentes, a construção de sistemas assíncronos e o dimensionamento horizontal tornam-se muito importantes no contexto da nuvem. Isso não somente permitirá ampliar o sistema adicionando mais instâncias do mesmo componente, mas também permitirá criar modelos híbridos inovadores nos quais alguns componentes continuam a ser executados no local, enquanto outros componentes podem tirar proveito do dimensionamento em nuvem e usar a nuvem para computação adicional e largura de banda. Dessa forma, com o mínimo esforço, pode-se direcionar “excesso” de tráfego para a nuvem através da implementação de balanceamento de carga inteligente.



A **Figura 10-9** ilustra a forma usual (a) e a ideia de desacoplar componentes de software utilizando filas (b).

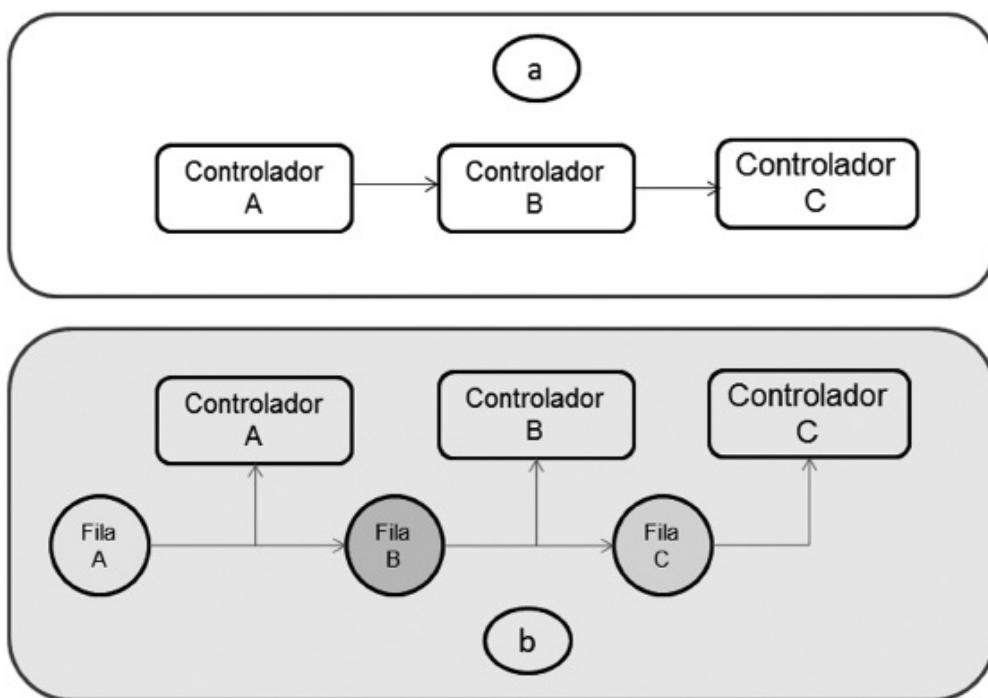


Figura 10-9 Desacoplamento entre componentes utilizando filas

Táticas específicas da AWS para implementar essa prática recomendada:

1. Utilizar o Amazon SQS para isolar componentes.
2. Utilizar o Amazon SQS como armazenamento em *buffers* entre componentes.
3. Projete cada componente a fim de que este exponha uma interface de serviço e seja responsável por sua própria capacidade de expansão em todas as dimensões adequadas e para que interaja com outros componentes de forma assíncrona.
4. Inclua a construção lógica de um componente em uma AMI para que ele possa ser implantado com mais frequência.
5. Deixe seus aplicativos o mais sem monitoração de estado possível. Armazene o estado de sessão fora do componente.

#### 10.4.3. Implemente a elasticidade

##### Contexto: infraestrutura e aplicação

A nuvem traz a ideia de elasticidade para os aplicativos. A elasticidade pode ser implementada de três maneiras:

- **Dimensionamento proativo cíclico:** dimensionamento periódico que

ocorre em intervalo fixo (diário, semanal, mensal ou trimestralmente).

- **Dimensionamento proativo baseado em evento:** executa o dimensionamento apenas quando se espera uma grande onda de solicitações de tráfego devido a um evento de negócios agendado (novo lançamento de produto, campanhas de marketing).
- **Auto Scaling baseado em demanda:** usando um serviço de monitoramento, o sistema pode enviar disparadores que tomam medidas apropriadas para expansão ou redução do poder computacional com base em métricas.

Para implementar a “elasticidade”, é preciso primeiro automatizar o processo de implantação e simplificar a configuração e o processo de compilação do aplicativo. Isso assegurará que o sistema pode ser expandido sem qualquer intervenção humana. Isso resultará em benefícios de custo imediato, à medida que a utilização é aumentada ao garantir que recursos estejam alinhados com a demanda, em vez de estarem sendo executados potencialmente em servidores que são subutilizados.

Táticas sugeridas pela AWS para automatizar a infraestrutura:

1. Defina grupos de *Auto Scaling* para diferentes clusters usando o recurso disponível no EC2.
2. Monitore métricas de sistema (CPU, memória, I/O de disco, I/O de rede) usando o CloudWatch e tome as medidas apropriadas (iniciar novas AMIs dinamicamente usando o serviço de *Auto Scaling*) ou envie notificações.
3. Armazene e recupere informações de configuração de máquina dinamicamente.
4. Crie um processo de compilação que insira as últimas versões em um *bucket* no Amazon S3. Faça o download da versão mais recente de um aplicativo durante a inicialização do sistema.
5. Invista na construção de ferramentas de gerenciamento de recursos (*scripts* automatizados, imagens pré-configuradas) ou use ferramentas de gerenciamento de configuração inteligente de código aberto.
6. Inclua o sistema operacional e suas dependências de software em uma AMI para que seja mais fácil gerenciar e manter. Passe parâmetros ou arquivos de configuração no momento da inicialização e recupere dados do usuário e metadados de instância após a inicialização.
7. Reduza a agregação e o tempo de inicialização ao iniciar a partir de volumes do EBS e anexar vários volumes do EBS a uma instância. Crie *snapshots* de volumes comuns e compartilhe *snapshots* entre

contas, sempre que possível.

8. Componentes do aplicativo não devem presumir integridade ou localização do hardware no qual estão sendo executados. Por exemplo, anexe dinamicamente o endereço IP de um novo nó ao cluster. Faça um *failover* automaticamente e inicie um novo clone em caso de falha.

#### 10.4.4. Pense paralelo

##### Contexto: aplicação.

A nuvem faz a paralelização sem esforço. É aconselhável não apenas implementar a paralelização sempre que possível, mas também automatizá-la, pois a nuvem permite que você crie um processo repetitivo facilmente.

Quando se trata de acesso a dados (recuperação e armazenamento), a nuvem é projetada para manipular operações massivamente paralelas. Para atingir o máximo desempenho e taxa de transferência, deve-se aproveitar o pedido de paralelização. Ao realizar o *multi-threading* de suas solicitações usando *threads* simultâneos você poderá armazenar ou buscar os dados mais rapidamente do que solicitá-los em sequência. Assim, sempre que possível, os processos de um aplicativo em nuvem devem ser feitos de maneira segura para o *thread* através de uma filosofia de não compartilhamento e de aproveitamento de *multi-threading*.

Quando se trata de processamento ou de solicitações em execução na nuvem, torna-se ainda mais importante aproveitar a paralelização. Uma prática geralmente recomendada, no caso de um aplicativo web, é distribuir as solicitações de entrada em vários servidores web assíncronos usando o balanceador de carga ELB. No caso de aplicativo de processamento em lote, é possível definir nós que podem gerar vários nós de trabalho secundário nessa tarefa de processos em paralelo.

Táticas específicas sugeridas pela AWS para paralelização:

1. Realize o lançamento de vários *threads* do S3.
2. Realize o lançamento de vários *threads* de solicitações GET e BATCH PUT do DynamoDB.
3. Crie um JobFlow usando o serviço Amazon Elastic MapReduce para cada um dos seus processos diários de lote (indexação, análise de logs, etc.), que calcularão o trabalho em paralelo e economizarão tempo.

4. Use o serviço ELB e distribua a carga entre vários servidores web dinamicamente.

#### 10.4.5. Ponha os dados no lugar certo

##### Contexto: infraestrutura

Em geral é uma boa prática manter os dados dinâmicos mais próximos da computação e os dados estáticos mais próximos do usuário final. Na nuvem, essa prática recomendada é ainda mais relevante e importante porque muitas vezes você tem que lidar com latências de internet. Além disso, na nuvem, paga-se pela largura de banda (dentro e fora da nuvem) por gigabyte de transferência de dados, e o custo pode aumentar muito rapidamente. Se uma grande quantidade de dados que precisa ser processada reside fora da nuvem, pode ser mais barato e mais rápido “enviar” e transferir os dados para a nuvem primeiro e, em seguida, fazer a computação.

Táticas específicas sugeridas pela AWS para implementar essa prática recomendada:

- Insira dados provenientes de discos rígidos na nuvem AWS usando o serviço de Import/Export.
- Utilize a mesma zona de disponibilidade para lançar um cluster de máquinas.
- Crie uma distribuição de *bucket* do S3 e deixe o conteúdo dos caches do CloudFront desse *bucket* em todos os pontos de presença ao redor do mundo.

#### 10.4.6. Utilize práticas recomendadas de segurança

##### Contexto: aplicação e infraestrutura

Em um ambiente *multi-tenant*, arquitetos de nuvem frequentemente demonstram preocupações com a segurança. O capítulo 13 relata as práticas bem aceitas para a segurança.

### 10.5. Elasticidade com *Auto Scaling*

#### 10.5.1. Introdução

O *Auto Scaling* permite expandir ou reduzir a capacidade do EC2 automaticamente, de acordo com as condições predefinidas. Com o *Auto Scaling*, garante-se que o número de instâncias do EC2 usadas se redimensione facilmente durante picos de demanda para manter o desempenho e diminua automaticamente durante quedas de demanda, para minimizar custos. Pode-se dizer que o *Auto Scaling* é o instrumento principal

utilizado pela AWS para a obtenção da elasticidade.

O *Auto Scaling* é especialmente útil para aplicativos que sofrem variabilidade de uso por hora, dia ou semana e normalmente é ativado por alarmes estabelecidos pelo CloudWatch, sendo disponibilizado sem custo adicional além das taxas do CloudWatch.

Roberto Silva, no seu blog “Planeta Tecnologia”, reforça que, antes da criação dos alarmes, é necessário lembrar que o monitoramento padrão do CloudWatch que se dá em intervalos de cinco minutos pode não servir. O monitoramento detalhado, que se dá em intervalos de um minuto, é mais apropriado. Este pode ser habilitado para a instância principal.

Ao selecionar uma instância para ser encerrada quando uma condição de escala for atendida, o *Auto Scaling* tenta preservar instâncias com a configuração de inicialização. É possível definir uma instância específica para encerramento usando a API *TerminateInstanceInAutoScalingGroup*.

O *Auto Scaling* não permitirá excluir o grupo *Auto Scaling* se ele contiver instâncias do EC2 em execução. É necessário esvaziar o grupo *Auto Scaling* ao definir seu tamanho para zero usando o comando específico na linha de comando. Será possível excluir com segurança o grupo *Auto Scaling* assim que ele estiver vazio.

Os principais recursos do *Auto Scaling* podem ser resumidos a seguir:

- Dimensiona horizontalmente as instâncias do EC2 com facilidade e automaticamente quando a demanda aumentar.
- Livra-se de instâncias desnecessárias do EC2 automaticamente e economiza dinheiro quando a demanda diminui.
- Dimensiona dinamicamente o cluster ELB com base em métricas do CloudWatch ou de forma previsível de acordo com sua própria programação.
- Recebe notificações via SNS baseadas em alertas do CloudWatch para iniciar ações do *Auto Scaling* ou quando o *Auto Scaling* completar uma ação.

O *Auto Scaling* pode ser usado em instâncias EC2, incluindo aquelas dentro da VPC ou instâncias que formam o *High Performance Computing*.

O *Auto Scaling* foi projetado para ajudar a tornar mais fácil usar o EC2, ajudando a reduzir a carga operacional de implantação e manutenção de aplicativos. O *Auto Scaling* monitora a integridade de cada instância EC2 que lança. Se qualquer instância termina inesperadamente, o *Auto Scaling* detecta o problema e inicia uma instância de substituição. Esse recurso ajuda a manter um número fixo de instâncias EC2.

Em um cenário comum do EC2, várias cópias de um aplicativo são

executadas simultaneamente para cobrir o volume de tráfego do cliente. Os clientes vão ver apenas uma URL para o aplicativo, mas por trás dessa URL existem diversas instâncias idênticas EC2 atendendo as solicitações de cada cliente. Essas instâncias EC2 são classificadas em grupos de *Auto Scaling*. Este é o conceito central do serviço. Grupos de *Auto Scaling* são definidos com um número mínimo e máximo de instâncias do EC2. Este tipo de grupo tem gatilhos que aumentam ou diminuem o tamanho com base na utilização da CPU média para todo o grupo.

### 10.5.2. Conceitos

- **Grupo de *Auto Scaling* (*Auto Scaling group*)**: um grupo de *Auto Scaling* é uma representação de várias instâncias do EC2 que compartilham características semelhantes e que são tratadas como um agrupamento lógico para fins de gerenciamento e dimensionamento de instância. Por exemplo, se um único aplicativo opera em várias instâncias, você pode querer aumentar ou diminuir o número de instâncias nesse grupo para melhorar o desempenho do aplicativo. Você pode usar o grupo *Auto Scaling* para aumentar automaticamente o número de instâncias ou manter um número fixo de instâncias. Um grupo de *Auto Scaling* pode conter instâncias EC2 que existem em uma ou mais zonas de disponibilidade.
- **Verificação de integridade (*health check*)**: uma verificação de integridade é uma chamada para verificar o status da saúde de cada instância de um grupo de *Auto Scaling*. Se o relatório informa a degradação de desempenho, o *Auto Scaling* encerra a instância e lança outra para tomar o seu lugar. Isso garante que o grupo *Auto Scaling* está consistente e operando normalmente.
- **Configuração de lançamento (*launch configuration*)**: uma configuração de lançamento captura os parâmetros necessários para criar novas instâncias do EC2. Você pode anexar apenas uma configuração de lançamento para um grupo de *Auto Scaling* por vez. Quando você anexa uma configuração de lançamento para seu grupo de *Auto Scaling*, quaisquer novas instâncias serão iniciadas usando os novos parâmetros de configuração. Instâncias existentes não são afetadas. Quando o *Auto Scaling* precisa de escala para baixo, ele termina primeiros instâncias que têm uma configuração de lançamento mais antiga.
- **Gatilho (*trigger*)**: um gatilho é um conceito que combina duas características AWS: um alarme CW (configurado para assistir uma métrica CW específica) e uma política de *Auto Scaling* que descreve o que deve acontecer quando é ultrapassado o limiar do alarme. Na maioria dos casos, serão necessários dois gatilhos, um para ampliar e outro para reduzir o grupo de instâncias.

- **Política:** é um conjunto de instruções para o *Auto Scaling* que informa ao serviço como responder a mensagens de alarme CloudWatch. Você pode configurar um alarme CW para enviar uma mensagem para *Auto Scaling* sempre que uma métrica específica alcançar um valor de disparo. Quando o alarme envia a mensagem, o *Auto Scaling* executa a política associada a um grupo *Auto Scaling* para dimensionar o grupo para cima ou para baixo.
- **Alarme:** um alarme do CW é um objeto que cuida de uma única métrica. Um alarme pode mudar de estado, dependendo do valor da métrica. Quando um alarme muda de estado, ele executa uma ou mais ações. Para criar um alarme, use a ação *PutMetricAlarm* para especificar a métrica, os valores de limite para a métrica, o número de períodos de avaliação e, opcionalmente, uma ou mais ações de SNS para executar quando o alarme altera o estado.
- **Scheduled Update:** uma *scheduled update* é uma chamada para o *Auto Scaling* que está programada para um tempo futuro. Atualmente, as atualizações são suportadas somente para *min-max* e capacidade desejada.
- **Scaling activity:** é um processo que implementa uma alteração no grupo de *Auto Scaling*, como, por exemplo, mudar o tamanho do grupo. Também pode ser um processo para substituir uma instância, ou para executar outras operações de longa duração suportadas pelo serviço.
- **Auto Scaling Instance Termination:** termina instâncias EC2 tanto em resposta a chamadas específicas para a ação *TerminateInstanceInAutoScalingGroup* como também em resposta a outras atividades de dimensionamento.
- **Cooldown:** é o período de tempo após o *Auto Scaling* iniciar uma atividade de escala durante o qual nenhuma outra atividade pode ter lugar. Um período *cooldown* permite que o efeito de uma atividade de escala se torne visível nas métricas que originalmente disparou a atividade. Este período é configurável e define a hora do sistema para executar e ajustar novas atividades (como a escala e de expansão) que afetam a capacidade.
- **Suspendable Process e Resume Process:** pode-se querer parar processos automatizados de dimensionamento em certos grupos para executar operações manuais ou desligar a automação em situações de emergência. Para satisfazer estas necessidades o *Auto Scaling* oferece duas ações: *SuspendProcesses* e *ResumeProcesses*. Você pode suspender processos de dimensionamento a qualquer momento. Quando estiver pronto, você pode retomar todos os processos suspensos.

### 10.5.3. Utilização comum

Uma sequência para utilização comum do *Auto Scaling* é sugerida a seguir.

- Fazer o download das ferramentas de linha de comando do *Auto Scaling* e das ferramentas de linha de comando do CW no site da AWS.
- Utilizar o comando *as-create-launch-config* para criar uma configuração inicial para o grupo *Auto Scaling*. Uma configuração inicial captura os parâmetros necessários para iniciar novas instâncias do EC2.
- Utilizar o comando *as-create-auto-scaling-group* para criar um grupo *Auto Scaling*. Um grupo *Auto Scaling* é uma coleção de instâncias EC2 para as quais se estabelecem certas condições de escala.
- Utilizar o comando *as-put-scaling-policy* para descrever cada ação de escala que se quer estabelecer. Por exemplo, é possível criar uma política que acrescenta instâncias do EC2 e outra que remove instâncias EC2.
- Utilizar o comando *mon-put-metric-alarm* para criar um alarme para a condição sob a qual se deseja acrescentar ou remover instâncias EC2 e para especificar a política do *Auto Scaling* que se deseja que o alarme execute quando a condição se apresentar. É possível definir alarmes com base em qualquer métrica que o CW coletar. Exemplos de métricas sobre as quais é possível estabelecer condições de alarme incluem utilização média de CPU, atividade de rede ou utilização de disco.
- O *Auto Scaling* monitora as condições e, quando essas se apresentarem, executará a ação de escala correspondente anteriormente definida. As taxas de cobrança apropriadas do CloudWatch serão aplicadas.

### 10.5.4. Características

O *Auto Scaling* oferece vários recursos que ajudam a economizar tempo e dinheiro.

- **Capacidade elástica:** adicionar capacidade computacional quando o uso dos aplicativos sobe e removê-la quando o uso cai automaticamente.
- **Facilidade de usar:** gerenciar instâncias espalhadas por uma ou várias zonas de disponibilidade como uma única entidade coletiva, usando ferramentas simples de linha de comando ou por meio de programação através de uma API de serviços web fácil de usar.

- **Poupar custo:** encerrar instâncias subutilizadas automaticamente e lançar novas instâncias quando você precisar delas, sem a necessidade de intervenção manual.
- **Redundância geográfica e escalabilidade:** distribuir, escalar e balancear aplicações automaticamente ao longo de várias zonas de disponibilidade dentro de uma região.
- **Manutenção mais fácil:** substituir automaticamente instâncias insalubres ou perdidas baseadas em alarmes predefinidos e limites.
- **Programar as ações:** definir o cronograma de ações de dimensionamento para tempos futuros e datas quando se espera precisar de mais ou menos capacidade.

## 10.5.5. Tipos de dimensionamento

### 10.5.5.1. Dimensionamento manual

Dimensionamento manual é o modo mais básico para dimensionar recursos. Envie uma chamada de API ou use a interface de linha de comando (CLI) para iniciar ou encerrar uma instância EC2. Só é necessário especificar a alteração na capacidade desejada. O *Auto Scaling* gerencia o processo.

### 10.5.5.2. Dimensionamento por agendamento

Às vezes sabe-se exatamente quando se vai precisar aumentar ou diminuir o número de instâncias no seu grupo simplesmente porque essa necessidade possui um cronograma previsível. O *Auto Scaling* com dimensionamento por agendamento significa que as ações de dimensionamento são executadas automaticamente, em função da data e hora.

### 10.5.5.3. Dimensionamento por política

Uma maneira avançada de escalar recursos é fazer o dimensionamento por política. Por exemplo, pode-se criar uma política que amplia o pool de instâncias sempre que a taxa de utilização de CPU média ficar acima de 90% por quinze minutos.

## 10.5.6. Ferramentas

Atualmente existem duas ferramentas que podem ser utilizadas para trabalhar com o *Auto Scaling*. É possível usar as ferramentas de linha de comando (CLI) ou usar a API QUERY. As ferramentas de linha de comando devem ser instaladas no computador que acessa recursos EC2 na nuvem AWS. As ferramentas do *Auto Scaling* funcionam em qualquer máquina conectada à internet, e os sistemas operacionais suportados são Windows, Linux e Mac.

A AWS fornece o “Auto Scaling Command Line Tool”, que pode ser

baixado do site AWS e possibilita o uso do *Auto Scaling*. A **Figura 10-10** ilustra esta opção.

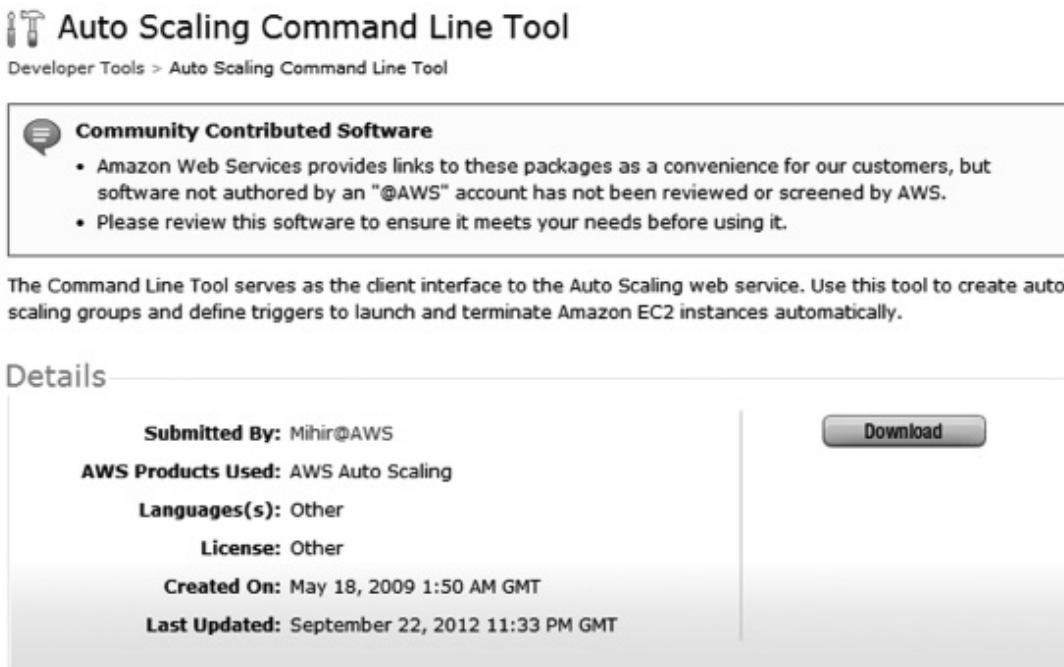


Figura 10-10 Auto Scaling Command Line Tool

### 10.5.7. Funcionamento

Quando se configura o *Auto Scaling*, diz-se ao serviço como fazer uma de três coisas:

- Manter os níveis atuais de instância (verificação de integridade).
- Criar mais instâncias (escala para cima).
- Excluir instâncias atuais (escala para baixo).

Pode-se manter os níveis atuais de instância com base no estado de saúde das instâncias do grupo *Auto Scaling* em questão. Pode-se escalar para cima ou para baixo o grupo com base em uma política ou uma ação programada.

Um uso comum do *Auto Scaling* é manter os níveis atuais de instância conduzindo verificações do estado de saúde das instâncias.

Uma política de escala diz ao *Auto Scaling* para executar uma ação de escala quando o valor de uma determinada métrica cruza um limite determinado para cima ou para baixo.

Uma ação programada diz ao *Auto Scaling* para executar uma ação de escala em determinado momento. Por exemplo, se existe agenda para promoção de produto e se espera maior tráfego na época de lançamento, pode-se agendar mais instâncias para acomodar o tráfego na data de lançamento. Provavelmente será necessário agendar uma ação

correspondente para diminuir instâncias quando o tráfego diminuir.

É possível combinar verificações de integridade, dimensionamento de políticas e ações agendadas. Por exemplo, pode ser necessário aumentar e diminuir as instâncias com base na utilização de CPU durante as datas de alto tráfego, mas também garantir que instâncias insalubres sejam finalizadas e substituições sejam lançadas.

A **Figura 10-11** ilustra um diagrama esquemático de como utilizar instâncias EC2,平衡adores de carga ELB, *Auto Scaling* e CW. Uma solicitação HTTP é balanceada através de uma coleção de instâncias EC2. O CloudWatch captura e armazena dados sobre o desempenho das instâncias. Estes dados são usados pelo *Auto Scaling* para regular o número de instâncias EC2 na coleção.

O *Auto Scaling* utiliza *status checks* para atuar nas instâncias EC2 do grupo.

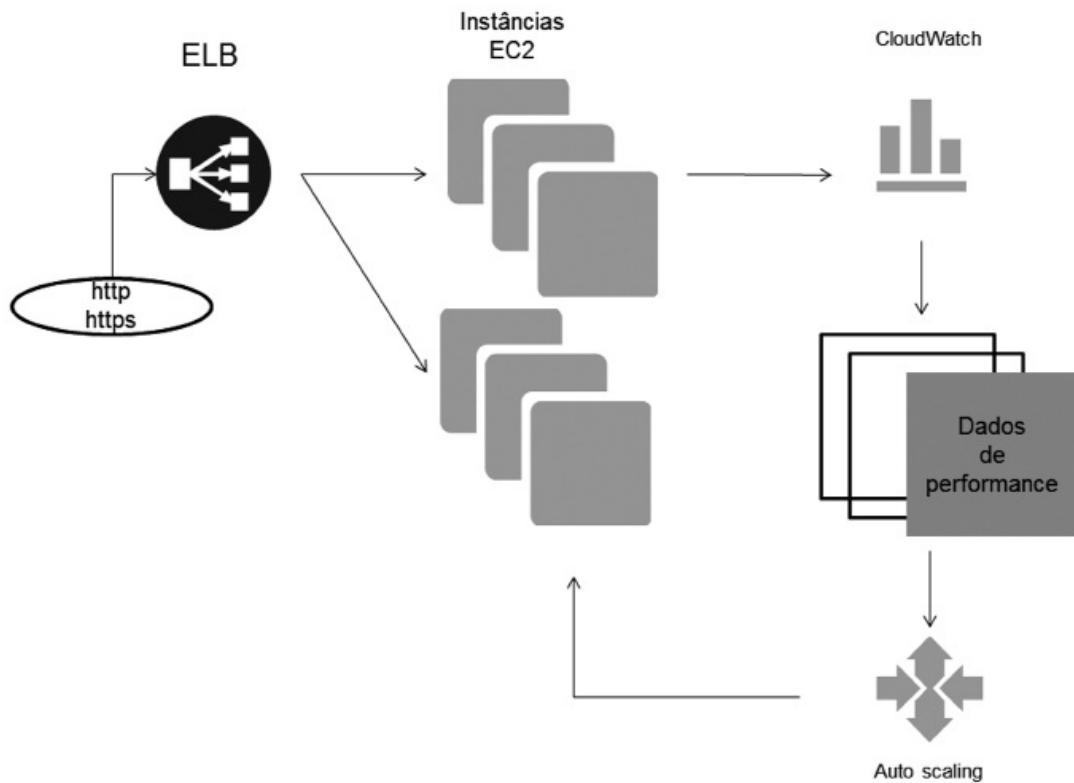


Figura 10-11 Uso do EC2, *Auto Scaling* e CloudWatch em conjunto

O ElasticWolf (<http://www.elasticwolf.com/>) é um aplicativo cliente para gerenciar recursos de nuvem AWS, incluindo o *Auto Scaling*, com uma interface gráfica fácil de usar. Este projeto foi inspirado por uma ferramenta mais velha chamada ElasticFox, mas não tem nenhum código em comum. O ElasticWolf adiciona novos recursos, incluindo suporte para a região de AWS GovCloud, melhor suporte a VPC e outras melhorias.

## 10.6. Aplicações web

## 10.6.1. Introdução

Manter a hospedagem de um site web altamente disponível e escalável pode ser uma tarefa complexa e cara. Tradicionais arquiteturas web escaláveis não são necessárias só para implementar soluções complexas para garantir altos níveis de confiabilidade, mas também são necessárias para uma previsão exata de tráfego para fornecer um alto nível de serviço ao cliente. Conseguir disponibilidade e escalabilidade na forma tradicional pode implicar em altos custos.

A AWS fornece uma saída para este problema. O artigo “Web Application Hosting in the AWS Cloud: Best Practices”, de Matt Davis, publicado pela AWS, retrata esta situação e a possível solução com a AWS.

## 10.6.2. Aplicações web tradicionais e aplicações web com AWS

A arquitetura de um *web hosting* tradicional é construída em torno de um modelo comum de três camadas, conforme já visto. Esta arquitetura já foi projetada para expansão adicionando *hosts* para as camadas. Ela também possui recursos internos para *failover*, otimizando a disponibilidade. Esta arquitetura não está aprimorada para lidar com picos de tráfego não previstos. Uma consequência ruim do provisionamento lento associado a um modelo de hospedagem tradicional é a incapacidade de reagir a tempo para os picos de tráfego inesperados.

A arquitetura *web hosting* é facilmente transportável para os serviços de nuvem AWS, com apenas um pequeno número de modificações.

No modelo tradicional de hospedagem, servidores precisam ser provisionados para lidar com a capacidade de pico e os ciclos não usados são desperdiçados fora dos períodos de pico. Aplicações web com AWS podem alavancar o pedido de provisionamento de servidores adicionais, o que permite que a capacidade e os custos sigam a demanda de tráfego.

Diferenças importantes entre a versão tradicional e a versão AWS:

- Não existe mais necessidade de utilização de servidores físicos no modelo AWS.
- Os *firewalls* existem em todos os lugares no modelo EC2.
- Existe a disponibilidade de utilizar vários DATACENTERS.
- As instâncias EC2 devem ser consideradas efêmeras e dinâmicas. As aplicações não devem assumir que o *host* está sempre disponível e que os locais onde os dados estão não podem falhar.

O artigo “The Total Cost of (Non) Ownership of Web Applications in the Cloud” demonstra as vantagens da solução web baseada na plataforma AWS em termos de custo total de propriedade (*Total Cost of Ownership* – TCO). O artigo reforça que sites pouco previsíveis em termos de demanda são os que

apresentam maior vantagem para uso da arquitetura AWS.

## 10.7. Aplicações empresariais

### 10.7.1. Introdução

Empresas de todos os tamanhos estão implementando aplicativos empresariais na nuvem para simplificar a gestão de infraestrutura, melhorar o *time to market* e baixar os custos. A economia em custos de propriedade inerente a um ambiente de provedor de nuvem em comparação aos custos de um DATACENTER empresarial pode ser significativa<sup>[8]</sup>.

### 10.7.2. Aplicações tradicionais e aplicações na AWS

Aplicações empresariais são complexas e pedem ambientes com demandas diversas de infraestrutura. Prever a infraestrutura para esses ambientes é uma verdadeira aposta. Quase sempre as ferramentas de *sizing* consideram perfis de usuário com base em informações nem sempre muito confiáveis e fazem diversas aproximações antes de soltar o resultado do dimensionamento da infraestrutura. Os projetistas sabem disso, e a forma de proteger o projeto é dimensionar a capacidade dos componentes para cima. Quase sempre o projeto é superdimensionado para reduzir riscos.

Em muitos casos, logo após o projeto, a empresa em questão precisa reavaliar sua infraestrutura devido a uma nova aquisição ou mesmo devido à realização de um *merge*, situações muito comuns nos dias de hoje. Em certas situações, justificar a ampliação de uma infraestrutura de forma rápida pode ser um problema. Como convencer os outros “CxOs” que a infraestrutura precisa ser ampliada já de imediato?

A TBR<sup>[9]</sup> concluiu um estudo global que reforça que empresas no mundo inteiro enxergam a utilização da nuvem para estes ambientes como forma de otimizar e estender estas funções. Os *drivers* para a adoção do modelo de nuvem no ambiente de desenvolvimento e qualidade, segundo a pesquisa, são:

- Permitir avaliar alternativas para novos serviços de negócio.
- Atualização da TI: envelhecimento da infraestrutura atual que retarda as operações de negócio.
- Aumento da capacidade e do desempenho da infraestrutura sem investir ou fazer manutenção na infraestrutura tradicional.

Boa parte dos recursos de uma solução de ERP (*Enterprise Resource Planning*) vai para a infraestrutura destes ambientes e de produção.

Fazer o correto *sizing* da arquitetura do SAP, o principal software do tipo ERP, é um fator primordial para uma implementação bem-sucedida deste tipo de projeto. Este *sizing* deve considerar os aspectos de disponibilidade e

performance (SLA), de um lado, e, do outro, o risco e o custo envolvidos. Quanto melhor os SLAs, menor o risco e maior o custo do projeto.

O SAP requer, além da instância de produção (PRD), instâncias adicionais para controle de qualidade (QAS), desenvolvimento (DEV) e, em certas instalações, o treinamento (TRAIN). Observe que diversas bases de dados convivem ao mesmo tempo.

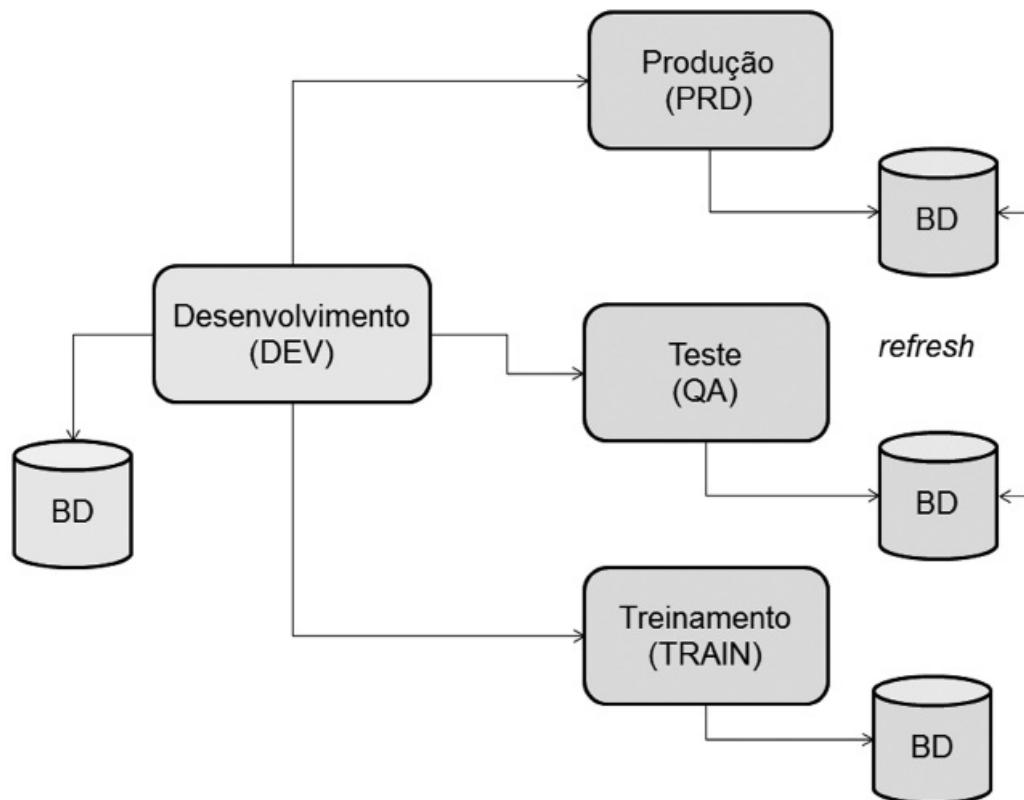


Figura 10-12 Ambientes SAP

Esses ambientes são conhecidos como *landscapes* (cenários) SAP. O que torna ainda mais complexo o *sizing* do SAP é que, para cada aplicativo (R/3, BI, CRM, Web AS), o SAP requer um cenário próprio. Também em muitos casos os cenários do SAP consistem de cópias adicionais dos dados de produção e de testes para outras funções corporativas como BI (*Business Intelligence*) e CRM (*Customer Relationship Management*), como ilustra a **Figura 10-13**.

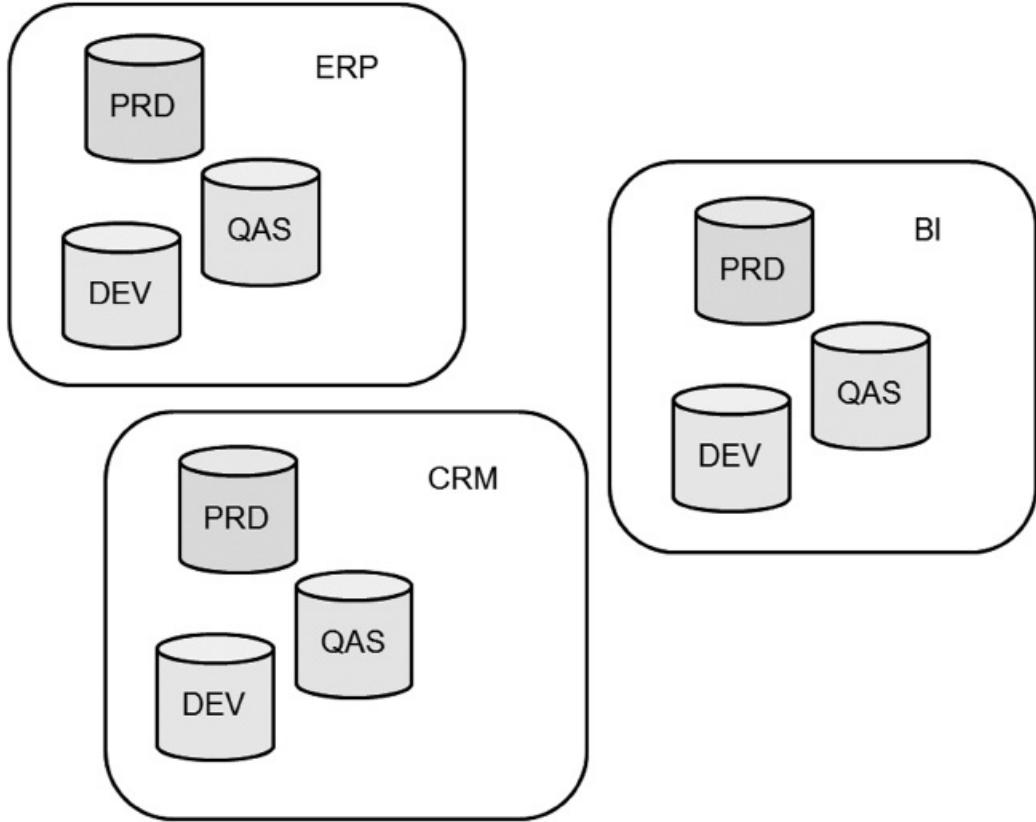


Figura 10-13 Aplicativos e ambientes SAP

Evidentemente, o dimensionamento da infraestrutura para o SAP não é trivial. E se a infraestrutura fosse adquirida mediante a demanda?

A **Figura 10-14** ilustra uma possível migração do SAP R/3 para a AWS feita em fases. A primeira fase pode migrar os ambientes de qualidade e desenvolvimento. Na segunda fase pode-se avaliar e resolver se vale a pena fazer a migração do ambiente de produção. Neste caso, aspectos da migração evidenciados no próximo item, como a dependência entre aplicações e a manutenção da consistência entre as bases de dados, devem ser considerados.

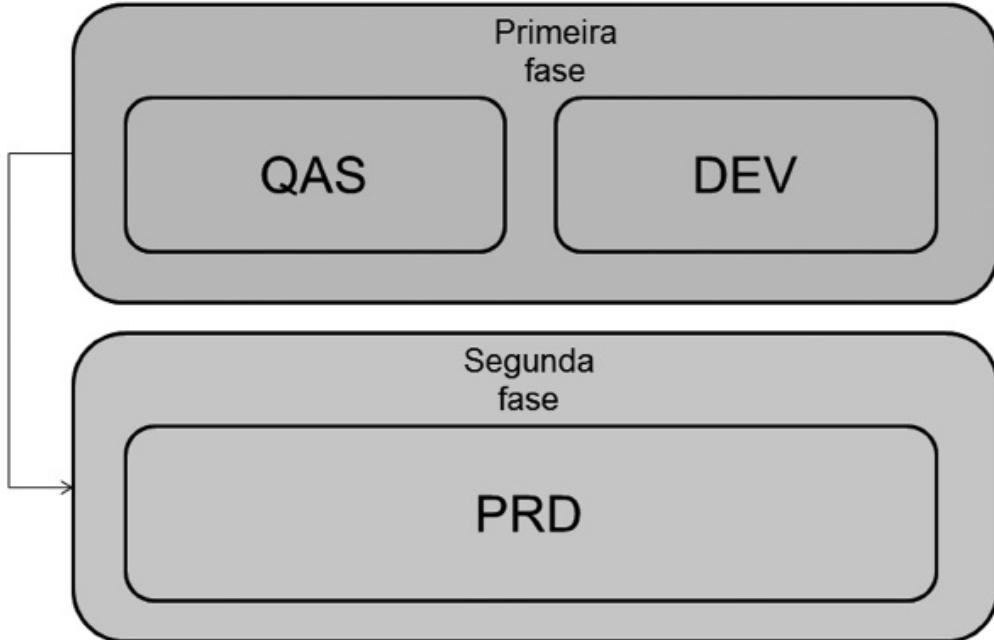


Figura 10-14 Migração para a AWS

O artigo “TCO Study for SAP on Amazon Web Services (AWS)”, publicado pela VMS em maio de 2012, aponta as vantagens em TCO da solução SAP baseada na AWS.

O referido artigo investigou os benefícios da utilização de uma arquitetura Cloud Computing para ambientes SAP. Foram comparados os custos de propriedade de três ambientes distintos para SAP com a solução baseada na AWS.

Para isto, o artigo comparou os ambientes ERP (*on-premises*), ambiente terceirizado de ERP e SAP All-in-One (*on-premises*) com o AWS. O artigo utilizou o modelo de TCO da SAP e o VMS Banchmarkerbase.

As conclusões do estudo refletem a redução do TCO total para o ambiente AWS da seguinte forma:

- ERP (*on-premises*, trezentos usuários): 17%.
- ERP (outsourcing, trezentos usuários): 15%.
- SAP All-in-One (*on-premises*, cem usuários): 22%.

Os principais componentes do TCO total são os custos de licenciamento de software, as taxas de manutenção e a operação (que inclui o gerenciamento de incidente e problema). Todos esses não são afetados pela AWS. O artigo permite concluir que o TCO de cinco anos para os sistemas SAP executados na AWS será reduzido entre 15% e 22% sobre sistemas idênticos que são implantados no local ou terceirizados.

As soluções SAP são certificadas para uso em produção no AWS, incluindo SAP Business Suite, SAP Hana One, SAP Business All-in-One e SAP

Business Objects. Para saber mais sobre o funcionamento do SAP na AWS visite <http://aws.amazon.com/sap>.

A AWS recentemente apresentou uma opção para ambientes SAP HANA. O SAP HANA é um banco de dados na memória que combina a funcionalidade de processamento de lógica de aplicativo na memória, processamento analítico de dados e processamento de dados transacional. O SAP HANA remove os limites da arquitetura de banco de dados tradicional que impõem restrições no desenvolvimento de aplicativos para suportar o negócio em tempo real. O SAP HANA ONE, a versão inicial do SAP HANA, foi disponibilizado no AWS Marketplace.

A AWS oferece uma plataforma de infraestrutura de nuvem segura e confiável, permitindo que as empresas implementem e executem de forma rápida aplicativos empresariais na nuvem. Segundo a Amazon, os clientes estão usando a AWS para executar todos os tipos de aplicativos empresariais, de pequenas soluções departamentais a aplicativos de missão crítica que automatizam os processos empresariais de toda a empresa. Com serviços de banco de dados, rede, armazenamento e computação *on demand*, departamentos de TI podem construir ambientes seguros na nuvem da AWS para executar aplicativos empresariais, incluindo também software certificado da Oracle, Microsoft e IBM.

Os links a seguir apontam para informações sobre a utilização das principais aplicações na nuvem AWS:

- Oracle: <http://aws.amazon.com/pt/enterprise-applications/oracle/>
- SAP: <http://aws.amazon.com/pt/enterprise-applications/sap/>
- Microsoft: <http://aws.amazon.com/pt/enterprise-applications/windows-server/>
- IBM: <http://aws.amazon.com/pt/solutions/global-solution-providers/ibm/>

O artigo “Amazon.com leverages the AWS Cloud for Database Backups”, publicado pela Amazon Web Services em setembro de 2012, mostra a migração da estrutura tradicional de backup da Amazon baseada no Oracle Recovery Manager (RMAN) para o backup baseado no S3 da AWS.

A AWS também disponibilizou uma arquitetura de referência para o Microsoft SharePoint Server. O artigo “Microsoft SharePointServer on AWS: Reference Architecture”, publicado pela Amazon Web Services em fevereiro de 2012, trata de explicar a migração do ambiente SharePoint para ambientes típicos de intranet e de internet para ambientes baseados na AWS. O cenário internet apresenta como diferenças do cenário intranet a presença da zona desmilitarizada (DMZ) para *firewall* e *threat management* e o fato de os controladores Active Directory residirem dentro da fazenda de servidores, e

não associados ao ambiente de usuário, como no caso do ambiente intranet.

Teste e desenvolvimento são fases importantes do ciclo de desenvolvimento de software. Nessas fases são aplicadas diversas técnicas e práticas visando produzir um software de qualidade.

Cada uma dessas fases requisita diferentes ambientes de infraestrutura de TI, que oneram o projeto como um todo. Além disso, esses ambientes são muitas vezes construídos de forma isolada e não permitem o compartilhamento de recursos.

A TBR[10] concluiu um estudo global que reforça que empresas no mundo inteiro enxergam a utilização da nuvem para esses ambientes como uma forma de otimizar e estender tais funções. Os drivers para a adoção do modelo de nuvem no ambiente de desenvolvimento e qualidade, segundo a pesquisa, são:

- Permitir avaliar alternativas para novos serviços de negócio.
- Atualização da TI: envelhecimento da infraestrutura atual, que retarda as operações de negócio.
- Aumento da capacidade e do desempenho da infraestrutura sem investir ou fazer manutenção na infraestrutura tradicional.

A AWS sugere que um ambiente baseado na nuvem permite obter a granularidade necessária para cada projeto e a consequente otimização do uso dos recursos.

O artigo “Development and Test on Amazon Web Services”, publicado pela Amazon em novembro de 2012, ilustra as várias possibilidades de uso da AWS nessas fases, incluindo sugestões para o armazenamento do repositório do código-fonte, ferramentas de gerenciamento de projetos, disposição de instâncias *builder* e opções de teste utilizando instâncias *spot*. Vale a pena conferir.

### 10.7.3. Checklists operacionais

O artigo “Operational Checklists”, de Steven Morad, publicado pela Amazon e escrito em fevereiro de 2012, elenca dois principais checklists que devem ser feitos antes de pôr a aplicação empresarial em produção. Esses checklists são melhores práticas que aumentam a chance de sucesso quando da realização do *deployment* de uma aplicação na nuvem AWS.

- **Checklist de operações básicas:** cobre questões técnicas relevantes que empresas devem considerar quando pensam em adotar serviços AWS. Envolve a utilização de grupos de segurança, uso do CNAME para mapear o nome DNS para o ELB, endereçamento IP dinâmico, utilização do EBS como unidade de armazenamento local, utilização do IAM para usuários específicos pertencentes a uma conta e teste

de aplicações antes de entrar em produção.

- **Checklist de operações empresariais:** cobre questões de revisão importantes referenciadas por melhores práticas que a empresa deve considerar. Envolve aspectos de gerenciamento do faturamento, acesso, segurança, disponibilidade, recuperação de desastres, monitoramento, gestão da mudança, controle de versões, etc.

## 10.8. Migração para a AWS

### 10.8.1. Introdução

Este item é baseado no artigo “Migrating your Existing Applications to the AWS Cloud: A phase-driven approach to Cloud Migration”, escrito por Jinesh Varia e publicado pela Amazon em outubro de 2010. Neste artigo o autor discute passos, técnicas e metodologias para migrar uma aplicação para a nuvem AWS. O artigo ajuda a pensar a forma de construir uma estratégia de migração de aplicativos para a nuvem.

### 10.8.2. Possíveis estratégias

Varia reforça que desenvolvedores e arquitetos que buscam construir novas aplicações para a nuvem podem simplesmente projetar os componentes, os processos e o fluxo de trabalho para a aplicação, empregar as APIs da nuvem de sua escolha e alavancar as últimas melhores práticas para o projeto, o desenvolvimento, o teste e a implantação da aplicação. Na escolha para implantar suas soluções em uma infraestrutura baseada em nuvem, como a AWS, eles podem tirar proveito imediato de escalabilidade instantânea e elasticidade, redução do esforço operacional, provisionamento *on demand* e automação.

O caso mais complexo é o de migrar aplicações existentes para a nuvem. O que se busca neste caso é aproveitar as vantagens do desenvolvimento na nuvem para aplicações já construídas. Um desafio.

Um dos principais diferenciais dos serviços da AWS é a sua flexibilidade. Ela dá às empresas a liberdade de utilizar modelos de programação, linguagens, sistemas operacionais e bancos de dados que já estão sendo utilizados na versão anterior à nuvem. Como resultado, muitas organizações estão movendo aplicações existentes para a nuvem hoje.

Também é verdade que alguns aplicativos, atualmente implantados em DATACENTERS empresariais, não apresentam ganhos concretos com a mudança para a nuvem. Esses aplicativos podem continuar a rodar localmente.

Jinesh Varia alega que existem diversos ativos dentro de uma organização que podem ser movidos para a nuvem, hoje, com o mínimo esforço.

O passo a passo, em uma abordagem orientada por fase, ajudará a identificar projetos ideais para a migração, construir o apoio necessário dentro da organização e migrar aplicativos com confiança.

Muitas organizações estão utilizando uma abordagem incremental na migração para a nuvem. É muito importante entender que, como qualquer migração, relacionada com a nuvem ou não, há um tempo para que os custos envolvidos, bem como as resistências entre os membros da equipe da TI (e também de fora), sejam absorvidos.

O autor do artigo citado sugere começar por construir apoio organizacional e por evangelizar e treinar. Concentrar-se no ROI de longo prazo, bem como em fatores tangíveis e intangíveis, e estar a par dos últimos desenvolvimentos na nuvem para que se possa tirar o máximo proveito dos seus benefícios.

Rodar aplicações na nuvem AWS pode reduzir custos de infraestrutura, aumentar a agilidade dos negócios e remover o “trabalho pesado” da TI de dentro da empresa.

A migração bem-sucedida em grande parte depende de três coisas: a complexidade da arquitetura do aplicativo; quão flexível (pouco acoplada) é a aplicação e quanto esforço se está disposto a colocar na migração. O artigo reforça que quando os clientes seguem a abordagem passo a passo (foco do artigo citado) e investem tempo e recursos para a construção de projetos de prova de conceito, eles enxergam claramente o enorme potencial da AWS e são capazes de alavancar seus pontos fortes rapidamente.

Assim, as estratégias para a nuvem podem ser divididas em dois grandes blocos:

- Se forem novas aplicações, o projeto já deve ser pensado para estar pronto para a nuvem. Parece simples, mas não é. Boa parte dos desenvolvedores ainda não sabe como projetar aplicativos para a nuvem, principalmente levando em consideração uma nova forma de adquirir infraestrutura.
- Se forem aplicações antigas, pode-se utilizar a abordagem de migração planejada por fases.

A **Figura 10-15** ilustra estas opções.

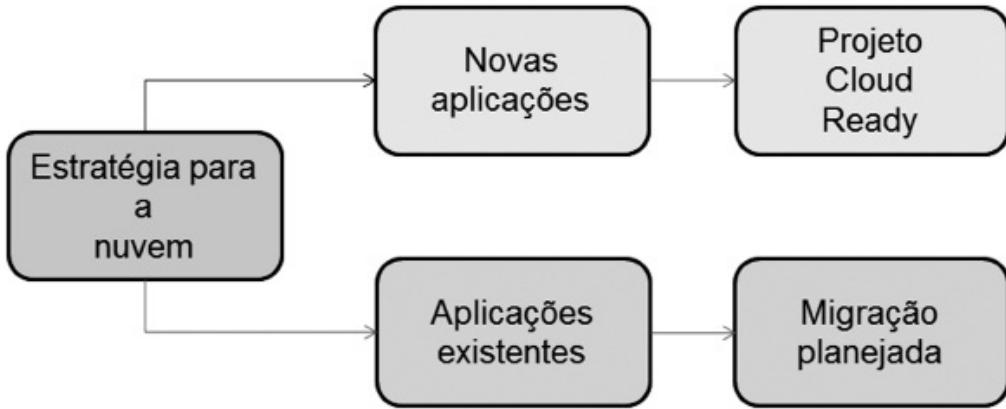


Figura 10-15 Estratégias para a nuvem

### 10.8.3. Fases sugeridas

Jinesh Varia sugere seis fases para todo o processo, descritas a seguir:

- **Fase 1:** sugere a construção de um *business case* para a migração para a nuvem. As aplicações que devem ser migradas para a nuvem são identificadas e os esforços para realizar esta migração são estimados. Nesta fase são definidos os critérios de sucesso para o projeto de migração e as ferramentas do ambiente de TI que poderão ser reutilizadas. Também verifica-se a forma de licenciamento dos produtos a serem migrados. Deve-se nesta fase criar uma árvore de dependência entre aplicações e uma classificação que permita saber a diferença entre elas em termos de criticidade.

A Figura 10-16 ilustra uma espécie de árvore com a dependência entre aplicações.

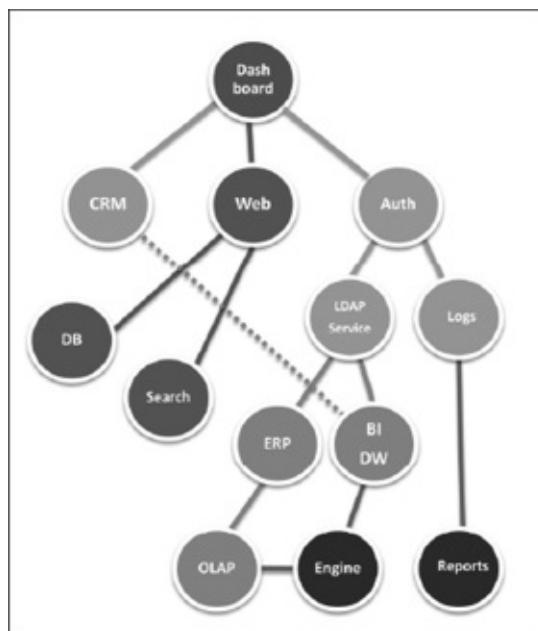


Figura 10-16 Dependências entre aplicações

São também identificados os candidatos certos para a migração, como sugere a Figura 10-17.

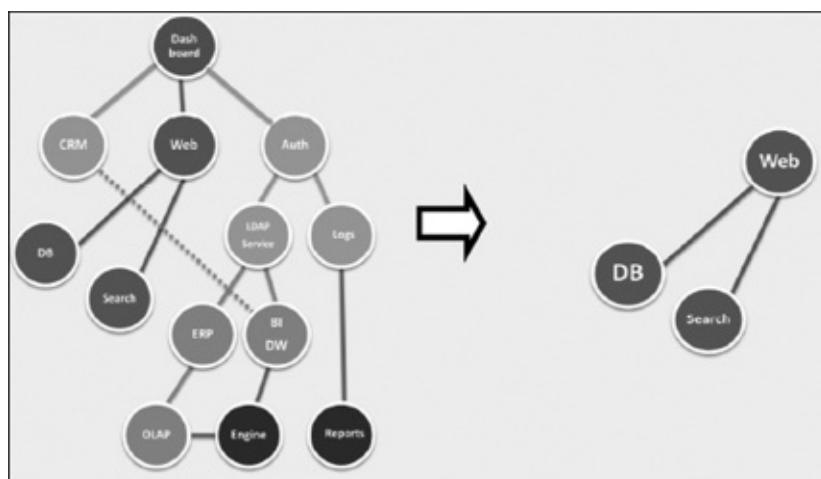


Figura 10-17 Identificação dos candidatos para a migração

- **Fase 2:** trata da prova de conceito. Normalmente nesta fase uma aplicação miniatura é desenvolvida com dados que simulem a aplicação real. É um piloto.
- **Fase 3:** deve-se escolher a opção de armazenamento dos dados considerando os diversos *trade-offs* entre as soluções existentes. São consideradas as dimensões de custo, durabilidade, disponibilidade, latência, desempenho, acessibilidade e outras. Os dados serão movidos para a nuvem e são consideradas as condições de realização de backup e restore para situações críticas.

- **Fase 4:** trata da migração da aplicação. A questão central é como migrar a aplicação ou parte dela sem parar o negócio. Aqui também a estratégia de backup e restore deve estar bem calibrada. O autor sugere duas alternativas para a migração da aplicação: ou faz do tipo *forklift*, que sugere migrar a aplicação inteira sem grandes mudanças ou poucas modificações no código para a nuvem, ou faz uma migração híbrida, que consiste em migrar partes da aplicação para a nuvem e deixar partes da aplicação rodando localmente. A migração híbrida reduz o risco e permite pensar a migração em partes definidas no tempo.
- **Fase 5:** prega a realização de testes e a confirmação de que tudo está funcionando como planejado ou próximo a isto. Deve-se nesta fase gastar tempo para encontrar novos benefícios do uso da nuvem. A estratégia para recuperação de desastres e para a segurança deve ser pensada aqui.
- **Fase 6:** trata da otimização do uso da nuvem para questões relacionadas a custos, como a utilização de instâncias reservadas, por exemplo. Aspectos de utilização da elasticidade e aumento da eficiência devem ser considerados também.

As fases sugeridas são ilustradas na **Figura 10-18**. Manter a ordem das fases em uma migração não é necessariamente importante, segundo o autor.

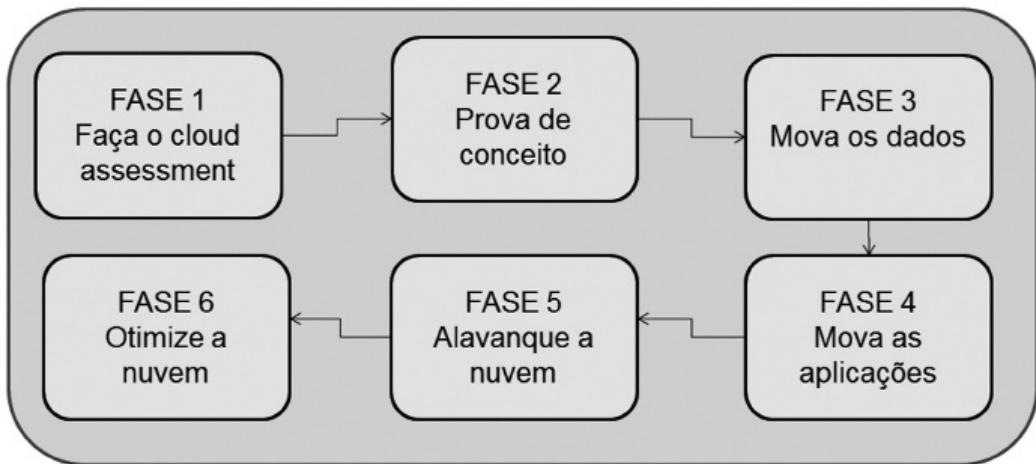


Figura 10-18 Fases sugeridas por um processo de migração

## 10.9. Referências bibliográficas

Amazon Web Services. **AWS Web Application Host for Linux**. Amazon Web Services, 2011.

Amazon Web Services. **AWS Web Application Host for Microsoft Windows**. Amazon Web Services, 2011.

Amazon Web Services. **Development and Test on Amazon Web Services**. 2012.

Amazon Web Services. **Microsoft SharePoint Server on AWS: Reference Architecture**. 2012.

Amazon Web Services. **Static Website Hosting**. 2011.

Amazon.com leverages the AWS Cloud for Database Backups. 2012.

<http://aws.amazon.com>

<http://aws.amazon.com/pt/architecture/>

<http://aws.typepad.com>

<http://planetatecnologia.com/tutorial-do-auto-scaling-com-exemplos-praticos>

Tavis, Matt. **Web Application Hosting in the AWS Cloud**. Amazon Web Services, May 2010.

Tavis, Matt. **Web Application Hosting in the AWS Cloud**. Amazon Web Services, May 2010.

TBR. **Serviços de Nuvem IBM**: Como a IBM está aumentando o valor agregado do ambiente de desenvolvimento na nuvem para os clientes.

TBR, 2011.

Varia, Jinesh. **Architecting for the Cloud: Best Practices.** Amazon Web Services, Jan. 2011.

Varia, Jinesh. **Migration your Existing Application to the AWS Cloud.** Amazon Web Services, Oct. 2010.

Varia, Jinesh. **The Total Cost of (Non) Ownership of Web Applications in the Cloud.** Amazon Web Services, Aug. 2012.

VMS. **TCO Study for SAP on Amazon Web Services (AWS).** 2012.

# 11. Governança

## 11.1. Introdução

O objetivo da governança de TI é a criação de valor para os stakeholders, obtendo os benefícios para a empresa com recursos e riscos de TI otimizados (baseado no CobIT 5).

A norma ISO que trata a governança da TI é a 38500:2008. Existem ainda a ISO 20000, relativa ao gerenciamento de serviços de TI, e a ISO 27001, com foco em sistemas de gerenciamento da segurança.

A governança da TI especifica direitos de decisão e estrutura de responsabilidades para encorajar comportamentos desejáveis no uso da TI (WEILL & ROSS).

Direitos de decisão tratam de estabelecer quem de fato deve tomar decisão referente às principais questões da TI, incluindo os princípios de TI, arquitetura empresarial, infraestrutura, necessidades comerciais e priorização dos investimentos em TI.

Estruturas de responsabilidade definem as responsabilidades relacionadas à TI.

Em uma situação onde parte ou toda a infraestrutura de TI está na mão de terceiros, como é o caso da adoção parcial ou total da nuvem AWS, é necessário entender a nova estrutura de responsabilidades que passa a ser compartilhada com o provedor.

Conformidade (*compliance*) também é um aspecto-chave da governança da TI, e ela existe para facilitar o atendimento de padrões ditados por normas e regras. Em TI, gestão da conformidade é utilizada de forma mais abrangente e se refere ao atendimento de padrões ditados por normas e regras e também aos esforços para evitar desvios chamados de controles gerenciados. A conformidade e o controle com o uso da nuvem também passam a ser compartilhados.

A **Figura 11-1** ilustra aspectos envolvidos com a governança da TI.

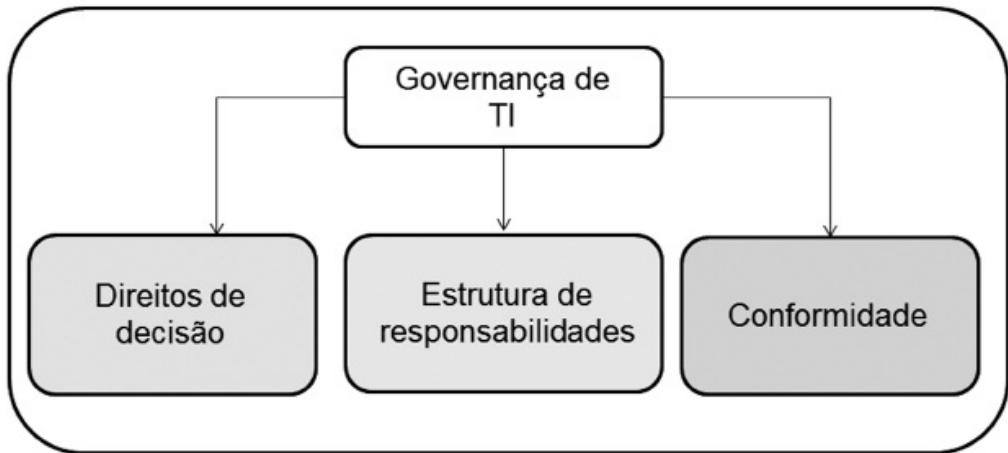


Figura 11-1 Governança da TI

Já os acordos de nível de serviço (*Service Level Agreement – SLA*), utilizados aqui no contexto da governança, refletem os aspectos formais da relação entre provedores de TI e os clientes.

A Amazon reforça que os clientes devem manter uma governança adequada ao longo de todo o ambiente de TI, independentemente da utilização de serviços de nuvem. Boas práticas de governança de TI incluem compreender os objetivos de conformidade, os requisitos exigidos e a criação de um ambiente de controle que possibilite atender a esses objetivos e requisitos.

A IBM sinaliza que os serviços de nuvem podem representar um nível adicional de risco para o negócio e que organizações precisam estabelecer um relacionamento de confiança com provedores de serviço e compreender o risco envolvido com a operação. Este relacionamento é conhecido como *trust-but-verify*, ou seja, é de confiança porém observado.

Este capítulo trata da governança da TI considerando o ambiente da nuvem AWS e seus impactos.

## 11.2. Governança tradicional e governança com AWS

### 11.2.1. Direitos de decisão e estrutura de responsabilidades

Uma parte importante da governança de TI é definir a estrutura de decisão para TI. Weill e Ross (2006) criaram a matriz de arranjos de governança de TI que permite sistematizar as decisões de TI considerando quais as principais decisões a serem tomadas (colunas) e quem as tomam (linhas).

As cinco principais decisões de TI são:

- **Princípios de TI:** esclarece o papel de negócio da TI. Trata das declarações de alto nível sobre como a TI é e como deve ser utilizada

no negócio.

- **Arquitetura empresarial:** define os requisitos de integração e padronização dos processos e sustenta o modelo operacional da organização. Trata da organização lógica de dados, de aplicações e infraestrutura, definidas a partir de um conjunto de políticas, relacionamentos e opções técnicas adotadas para obter a padronização e a integração técnicas e de negócio desejadas.
- **Infraestrutura de TI:** determina os serviços de entrega e de suporte da TI. Trata dos serviços de TI, coordenados de maneira centralizada e compartilhada, que fornecem a base para a capacidade de TI da empresa.
- **Necessidade de aplicações de negócio:** especifica as necessidades de aplicações, quer sejam adquiridas em formas, de pacote ou desenvolvidas internamente.
- **Investimentos e priorização de TI:** trata da escolha de que iniciativas financeirar e quanto gastar. Trata de decisões sobre quanto e onde investir em TI, incluindo a aprovação de projetos e as técnicas de justificativa.

Na matriz de arranjos de governança de TI, os títulos das linhas listam um conjunto de arquétipos que identificam as pessoas envolvidas nas decisões de TI:

- **Monarquia de negócio:** os altos gerentes.
- **Monarquia de TI:** os especialistas em TI.
- **Feudalismo:** cada unidade de negócio toma decisões independentes.
- **Federalismo:** combinação entre o centro corporativo e as unidades de negócio, com ou sem o envolvimento do pessoal de TI.
- **Duopólio de TI:** o grupo de TI e algum outro grupo.
- **Anarquia:** tomada de decisões individual ou por pequenos grupos de modo isolado.

Esses arquétipos descrevem os arranjos decisórios que os autores encontraram em uma pesquisa com as maiores empresas americanas. O desafio sugerido pelo modelo é determinar quem deve ter a responsabilidade por tomar e contribuir com cada tipo de decisão de governança de TI. As decisões de utilizar a nuvem devem obrigatoriamente obedecer à matriz de governança de TI.

As organizações possuem diferentes arranjos decisórios que muitas vezes não são explícitos. Esses arranjos estão intimamente ligados aos mecanismos de controle da informação e a sua cultura e, em muitas situações, para o bem da organização, precisam ser mudados. A mudança só é

realmente efetivada com o apoio e o envolvimento das lideranças do primeiro time de executivos que estão fora da organização de TI.

Weill e Ross (2006) observaram que, na maioria das empresas onde a governança de TI é considerada de boa qualidade, as decisões referentes a princípios de TI e priorização de investimentos são baseadas em arranjos do tipo duopólio, onde as decisões pertencem ao grupo de TI e a algum outro grupo. As decisões relativas à infraestrutura e arquitetura de TI na maioria dos casos utilizavam um arranjo do tipo monarquia de TI.

O que acontece com as empresas que possuem uma governança de TI efetiva? Segundo Weill e Ross (2006), têm lucros até 20% maiores do que empresas que buscam estratégias similares.

No caso de adoção da nuvem, pode-se assumir que esta decisão está ligada diretamente às decisões de arquitetura empresarial (relacionada diretamente com a arquitetura de TI) e infraestrutura de TI. Decisões compartilhadas podem ser uma saída para decisões de adoção da nuvem. Também a arquitetura empresarial influencia e é influenciada pela infraestrutura de TI. É mais comum que a arquitetura empresarial que sustenta o modelo operacional e, logicamente, a estratégia influencie a infraestrutura de TI.

A decisão compartilhada permite tirar o peso da decisão sobre o diretor de TI, ao mesmo tempo em que força a participação dos outros executivos na decisão, fazendo com que eles saibam como a empresa funcionará neste novo modelo. Como a mudança tem diversas implicações, melhor que pessoas de negócio e de TI assumam suas responsabilidades neste novo contexto. Isto é só uma sugestão e, para adotá-la, deve-se observar a realidade da organização que planeja ir para a nuvem.

A **Figura 11-2** ilustra a matriz de governança encontrada em empresas consideradas de boa governança e a sugestão de alterar o padrão de decisão com a presença da nuvem AWS. Decisões de arquitetura e infraestrutura podem ser compartilhadas com decisores do negócio para melhor governança.

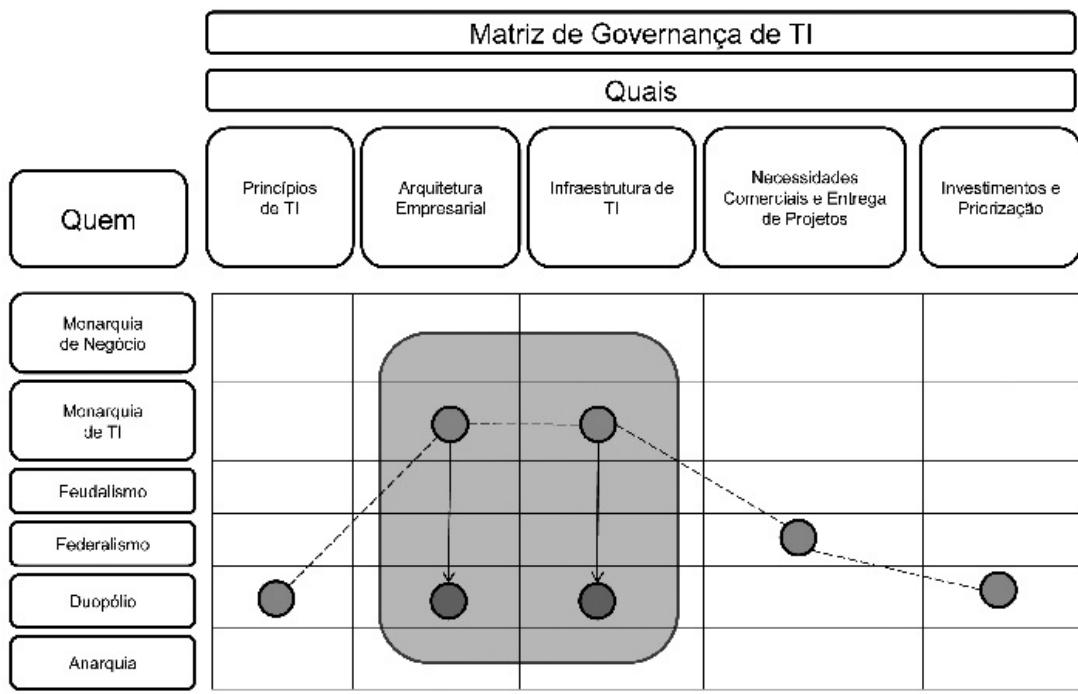


Figura 11-2 Matriz de arranjos de governança de TI

A **Figura 11-3** ilustra a ideia de responsabilidade compartilhada na nuvem em função do modelo adotado (no diagrama, a parte clara é responsabilidade do cliente e a parte escura é responsabilidade do provedor). Nos extremos, a responsabilidade é do cliente ou do provedor de serviços. Mas, no caso do IaaS, a responsabilidade é compartilhada e boa parte dela é do cliente.

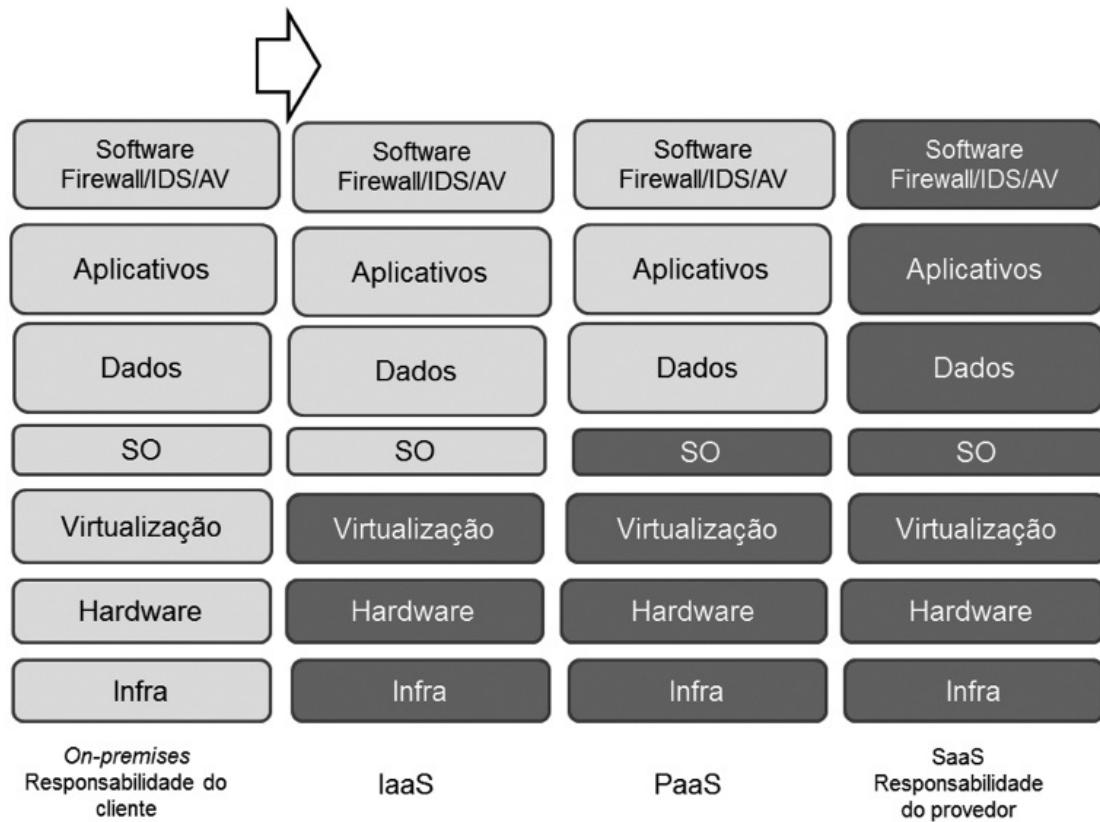


Figura 11-3 Responsabilidade compartilhada na nuvem

No caso da AWS, um caso típico de IaaS, o cliente assume a gestão e a responsabilidade pelo sistema operacional convidado (inclusive atualizações e patches de segurança) e por qualquer outro aplicativo instalado, bem como pela configuração do grupo de segurança, configurações da rede e gerenciamento da conta.

A **Tabela 11-1** ilustra a divisão de responsabilidades no modelo AWS. Como pode ser visto, trata-se de um modelo de responsabilidade compartilhada.

**Tabela 11-1 Responsabilidade compartilhada na AWS**

AWS	Cliente
<b>Instalações</b>	Sistema operacional
<b>Segurança física</b>	Aplicação
<b>Segurança da infraestrutura</b>	Grupos de segurança ( <i>firewall</i> )
<b>Segurança do hardware</b>	Configuração da rede
<b>Virtualização da infraestrutura</b>	Gerenciamento da conta

O propósito da responsabilidade compartilhada é aumentar a segurança

e/ou atender aos requisitos de conformidade, aproveitando as tecnologias disponibilizadas pelo próprio AWS. A natureza desta responsabilidade compartilhada também deve fornecer a flexibilidade e o controle que permitem a implantação de soluções que atendam aos requisitos de certificação específicos do setor que a empresa cliente atua.

### **11.2.2. Gestão da conformidade e controles gerenciados**

A base para o tratamento deste assunto foi obtida no artigo “Risco e Conformidade da Amazon Web Services”, publicado pela AWS em maio de 2011. Neste artigo a AWS fornece informações sobre seu próprio programa de gestão de conformidade, para que os clientes possam incorporar controles da AWS na sua estrutura de governança.

Conformidade trata do atendimento de especificações prometidas a terceiros.

O modelo de responsabilidade compartilhada entre o cliente e a AWS também se estende aos controles de TI. Assim como a responsabilidade para operar o ambiente de TI é compartilhada entre a AWS e os seus clientes, o mesmo acontece com o gerenciamento, com a operação e com a verificação compartilhada de controles de TI.

Verifica-se assim que a auditoria para a maioria das camadas e de controles acima dos controles físicos continua a ser de responsabilidade do cliente.

A AWS utiliza o modelo de auditoria SAS 70. Este modelo permite que provedores de serviço demonstrem o nível de segurança dos seus processos e ambientes de controle em um formato padrão de dois tipos de relatórios:

- **Relatório tipo I:** reporta o nível de controle do prestador de serviço em determinada data.
- **Relatório tipo II:** inclui, também, testes detalhados de auditoria sobre a efetividade dos processos do prestador de serviço, com cobertura mínima de seis meses.

As áreas de cobertura do SAS 70 são: ambiente de controle interno, desenvolvimento e manutenção de sistemas, segurança lógica, acesso físico, área de operações, plano de contingência e procedimentos de recuperação, controle de entrada de dados, controle de saída de dados e controle de processamento.

Os controles lógicos e do acesso físico definidos pela AWS são documentados no relatório tipo II SAS 70 e estão disponíveis para análise por equipes de auditoria e conformidade do cliente.

A certificação ISO 27001, que garante que a AWS implementou um sistema para gerência da segurança da informação de acordo com os padrões

de sistemas de gerência ISO, está disponível para análise dos auditores.

Se um cliente processa informações financeiras na nuvem AWS, pode ser necessário comprovar que alguns sistemas AWS estejam de acordo com os requisitos da Sarbanes-Oxley (SOX). Os auditores do cliente devem fazer sua própria determinação sobre a aplicabilidade da SOX. A Amazon reforça que, como a maioria dos controles de acesso lógico é gerenciada pelo cliente, o cliente está mais bem posicionado para determinar se as suas atividades de controle satisfazem as normas pertinentes. Se auditores SOX solicitarem informações específicas sobre controles físicos da AWS, eles podem fazer referência ao relatório SAS 70 tipo II disponibilizado pela AWS, que detalha os controles fornecidos.

A Amazon pode ajudar a diminuir a preocupação do cliente em relação aos controles operacionais gerenciando os controles associados à infraestrutura física implantada no ambiente AWS que anteriormente tenham sido gerenciados pelo cliente. Os clientes podem então utilizar as certificações e os atestados de terceiros recebidos pela AWS disponíveis para realizar procedimentos de avaliação e verificação de controle conforme necessário.

A gestão da conformidade pode incluir a seguinte abordagem sugerida pela Amazon:

- Revisar as informações disponíveis na AWS juntamente com outras informações para entender o máximo possível sobre o ambiente de TI da AWS e, em seguida, documentar todos os requisitos de conformidade.
- Projetar e implementar os objetivos de controle para atender aos requisitos de conformidade corporativa.
- Identificar e documentar controles pertencentes a terceiros.
- Verificar se os objetivos de controle são atendidos, se os controles principais foram projetados com eficiência e se apresentam bom funcionamento.

Os clientes AWS podem e precisam identificar os principais controles gerenciados pela AWS. Para este fim, a AWS publica uma ampla variedade de controles de TI específicos em seu relatório SAS 70 tipo II. Os controles AWS podem ser considerados eficientes para muitos fins de conformidade, incluindo auditorias Sarbanes-Oxley (SOX), seção 404, demonstrativo financeiro. Utilizar relatórios SAS 70 tipo II é também geralmente permitido por outros organismos de certificação externos.

Se um cliente AWS requer que um amplo conjunto de objetivos de controle seja atendido, uma avaliação das certificações obtidas pela AWS deve ser realizada. Conforme dito pela própria Amazon, a AWS fornece informações sobre seu programa de conformidade para permitir que os clientes

incorporem controles AWS em sua estrutura de governança. Essas informações podem ajudar os clientes a documentar uma estrutura de governança e de controle com a AWS, incluída como uma parte importante desta estrutura.

A AWS gerencia um ambiente de controle abrangente que inclui políticas, processos e atividades de controle que utilizam diversos aspectos do ambiente de controle geral da Amazon. Esse ambiente de controle está em vigor para a entrega segura de ofertas de serviços da AWS. O ambiente de controle coletivo abrange pessoas, processos e tecnologia necessários para estabelecer e manter um ambiente que suporte a operação da AWS.

Como política essencial, a AWS utiliza a política de menor privilégio, onde o acesso é negado a todos por padrão.

A AWS está em conformidade com várias certificações e atestados de terceiros, descritos na **Tabela 11-2**.

**Tabela 11-2 Certificações e atestados AWS**

Certificações Descrição	
<b>SAS 70 SOC 1 Tipo II</b>	A AWS executa e produz um relatório sobre a eficácia do funcionamento desses controles junto com um parecer de auditor independente.
<b>PCI DSS Nível 1</b>	A AWS está em conformidade com o padrão de segurança de dados PCI como um provedor de serviços de host compartilhado.
<b>ISO 27001</b>	A AWS obteve a certificação ISO 27001 para o seu sistema de gestão de segurança da informação (ISMS – <i>Information Security Management System</i> ) que abrange a infraestrutura, DATACENTERS e serviços.
<b>FISMA</b>	A AWS permite que os clientes da agência de governo alcancem e mantenham a conformidade com a gestão de segurança de informação Federal (FISMA – <i>Federal Information Security Management Act</i> ). A AWS foi certificada e acreditada para operar em nível FISMA baixo.

### 11.3. SLAs tradicionais e SLAs com a AWS

Os clientes devem examinar cuidadosamente os serviços que escolherem, assim como os acordos de nível de serviços envolvidos, a integração desses serviços ao seu ambiente de TI corporativo e as leis e regulamentos aplicáveis.

Um acordo de nível de serviço (*Service Level Agreement* – SLA) é um acordo firmado entre a área de TI e o cliente interno. O SLA descreve o serviço de TI, suas metas de nível de serviço, além de papéis e responsabilidades das partes envolvidas no acordo.

O SLA tem a forma de um documento que deve ser acordado entre os

requisitantes ou interessados em um determinado serviço de TI e o responsável pelos serviços de TI da organização e deve ser revisado periodicamente para certificar-se de que continua adequado ao atendimento das necessidades de negócio da organização.

É comum também referir-se ao contrato firmado entre a organização e o fornecedor externo de TI como um SLA. Para a ITIL (*Information Technology Infrastructure Library*), a biblioteca mais utilizada de melhores práticas para os serviços de infraestrutura de TI, este documento, de aspecto mais formal, na verdade é o Contrato de Apoio (CA), que também envolve a definição de níveis de serviço, mas é mais complexo por envolver um fornecedor externo e por aspectos jurídicos precisarem ser considerados. Neste livro consideramos os dois tipos de documento como uma forma de SLA.

Como o SLA envolve a definição de níveis mínimos de serviço que são esperados pelo cliente de TI, é comum o uso de indicadores que permitam a mensuração quantitativa da qualidade do serviço recebido. Alguns indicadores comumente utilizados como métricas de SLA são o desempenho e a disponibilidade. A **Tabela 11-3** ilustra os componentes típicos de um SLA.

**Tabela 11-3 Componentes de um SLA**

Componentes Descrição	
Parâmetros de nível de Serviço	Descreve uma propriedade observável do serviço que possui valor mensurável.
Métricas	São definições das propriedades dos serviços medidas por um provedor de serviços ou computadas de outras métricas e constantes.
Função	A função especifica como computar um valor de uma métrica dos valores de outras métricas e constantes.
Diretivas de medição	Especificação de como medir as métricas.

Outro conceito importante é o do gerenciamento do nível de serviço (*Service Level Management – SLM*). O SLM envolve a negociação, o acordo e a apropriada documentação de níveis de serviço que atendam às necessidades do negócio, permitindo a entrega de serviços de TI com a qualidade esperada. No contexto do SLM, o SLA tem fundamental importância, pois é nele que estarão definidos, aceitos e formalizados os níveis de serviço esperados pelo cliente de TI.

Os três principais tipos de SLAs para serviços de nuvem são SLA de infraestrutura, SLA de plataforma e SLA de aplicação, conforme ilustra a **Figura 11-4**.

O SLA de infraestrutura deve garantir a disponibilidade da infraestrutura.

Este SLA envolve normalmente servidores, energia, conectividade da rede, etc. O SLA de plataforma trata de acordos de serviço referentes a uma plataforma de desenvolvimento na nuvem. O SLA de aplicação envolve a aplicação específica. A **Tabela 11-4** ilustra elementos-chave de um SLA de infraestrutura.

**Tabela 11-4 Elementos-chave de um SLA de infraestrutura**

Elementos-chave	Descrição
<b>Disponibilidade do hardware</b>	99% de uptime por mês.
<b>Disponibilidade da rede do DATACENTER</b>	99,99% de uptime por mês.
<b>Disponibilidade do backbone</b>	99,999% de uptime por mês.
<b>Garantia de latência interna</b>	Não pode exceder 60 msecs em intervalos de medida de cinco minutos.
<b>Crédito de serviço por indisponibilidade</b>	Devolução de crédito de serviço para o período de <i>downtime</i> .
<b>Notificação de “outage” garantida</b>	Notificação do cliente em no máximo uma hora do <i>downtime</i> .

A **Tabela 11-5** ilustra elementos-chave de um SLA de aplicação.

**Tabela 11-5 Elementos-chave de um SLA de aplicação**

Elementos-chave	Descrição
<b>Resposta do site web</b>	3,5 segundos por solicitação do usuário.
<b>Latência do servidor de banco de dados</b>	0,5 segundo por “query”.
<b>Latência do servidor web</b>	0,2 segundo por solicitação do usuário.

Os acordos de nível de serviço para plataforma ainda estão em fase inicial, e mesmo os acordos de nível de serviço para infraestrutura, mais comuns, são muitas vezes inadequados.

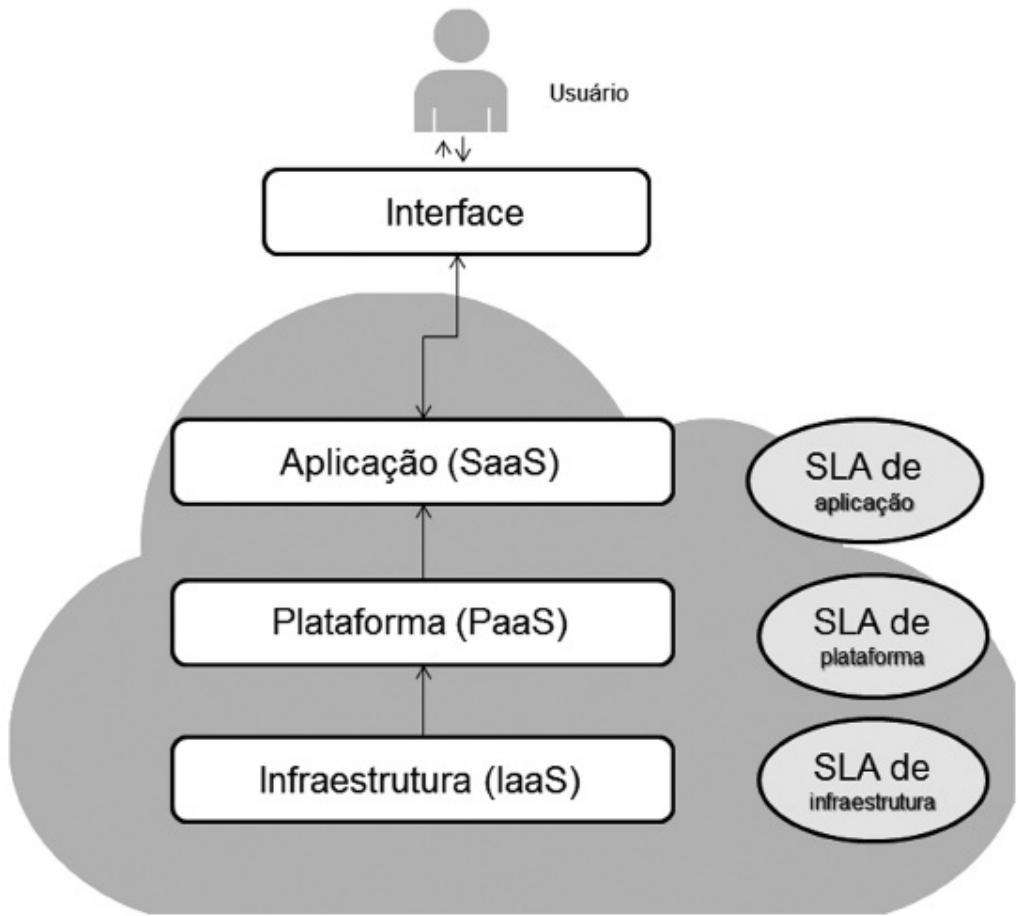


Figura 11-4 SLAs para diferentes modelos de cloud computing

O SLA também pode ser negociado entre o provedor de nuvem e o cliente. Neste caso, o SLA possui um ciclo de vida composto pelas fases descritas a seguir:

- **Definição:** normalmente é proposta pelo provedor de serviços e define a oferta de serviços e os correspondentes SLAs usando templates padrões. Evidentemente, esses SLAs podem ser customizados mediante um ajuste das condições de contrato com o provedor.
- **Publicação e descoberta:** é uma fase em que os provedores anunciam suas ofertas através de um catálogo normalmente exposto na mídia, e os clientes devem ter a habilidade de encontrar a oferta adequada. Os serviços encontrados podem ser negociados posteriormente.
- **Negociação:** envolve negociar o possível contrato, os serviços e os SLAs. Para contratos simples, esta fase é quase automática. Para contratos complexos esta fase é vital. No caso de provedores de nuvem de infraestrutura, aplicações precisarão ser avaliadas antes de serem colocadas na nova infraestrutura.
- **Operacionalização:** trata de monitorar os SLAs, medir os parâmetros e compará-los com os valores assumidos em contrato – logicamente,

com apoio de um software. A operacionalização também envolve a ação de capturar e registrar indicadores/métricas dos SLAs para conformidade e também trata da correção quando o SLA acordado não atende.

- **Descomissionamento:** envolve terminar as atividades realizadas relativas a um determinado SLA quando o contrato chega ao fim.

No caso da AWS, os acordos de nível de serviço se restringem à infraestrutura. Para os principais serviços existem contratos específicos. O SLA da aplicação dependerá diretamente do desenho da arquitetura da infraestrutura utilizada e dos serviços fornecidos pela AWS. O desenho da arquitetura da infraestrutura é de responsabilidade do cliente e pode inclusive ter o apoio de consultores da AWS.

As descrições dos acordos de nível de serviço para os web services EC2, S3, CloudFront e Route 53 podem ser encontradas em:

- <http://aws.amazon.com/pt/ec2-sla/>
- <http://aws.amazon.com/pt/s3-sla/>
- <http://aws.amazon.com/pt/cloudfront/sla/>
- <http://aws.amazon.com/pt/route53/sla/>

A seguir descreve-se o contrato de nível de serviço para o EC2 da forma que foi redigido pela Amazon.

### **11.3.1. Contrato de nível de serviços do EC2**

Data de vigência: 23 de outubro de 2008.

O presente contrato de nível de prestação de serviços da plataforma EC2 (“SLA”) consiste em um procedimento que rege o uso da Elastic Compute Cloud (EC2) (“Amazon EC2”) de acordo com os termos do contrato do cliente AWS da Amazon (o “Contrato do Cliente AWS”), celebrado entre a Amazon Web Services, LLC (a “AWS”, “nós” e correlatos de “nós”) e os usuários dos serviços da AWS (“você” ou “o cliente”). O presente SLA aplica-se separadamente a cada conta que se utiliza da Amazon EC2. Salvo disposição em contrário neste instrumento, o presente SLA está sujeito aos termos do contrato da AWS, e as expressões grafadas com letras iniciais maiúsculas assumem os significados a elas atribuídos no Contrato da AWS. Reservamo-nos o direito de mudar os termos do presente SLA de acordo com o Contrato da AWS.

#### **11.3.1.1. Compromisso de prestação de serviços**

A AWS empreenderá esforços comercialmente razoáveis para tornar a plataforma EC2 disponível com base em Porcentagem de Tempo de Atividade Anual (definida adiante) no nível de, no mínimo, 99,95% durante cada Ano de

Serviços. Na hipótese do EC2 não cumprir o compromisso expresso pela Porcentagem de Tempo de Atividade Anual, o cliente terá o direito de receber o Crédito de Serviço correspondente, conforme descrito adiante.

Definições:

- “Ano de Serviços” corresponde aos 365 dias precedentes à data de uma reivindicação relacionada ao SLA.
- “Crédito de Serviço” significa um crédito em dólares, calculado na forma estabelecida adiante neste instrumento, crédito este que a AWS poderá creditar, como devolução, a uma conta do EC2 para tanto qualificada.
- “Indisponível” significa que todas as instâncias operacionais do cliente não têm conectividade externa durante um período de cinco minutos, estando ele impossibilitado de usar instâncias substitutivas.
- “Período Elegível para Crédito” equivale a um único mês e refere-se ao ciclo de cobrança mensal no qual ocorreu o mais recente evento de Região Indisponível incluído em reivindicação baseada no SLA.
- “Porcentagem de Tempo de Atividade Anual” é calculada subtraindo-se de 100% a porcentagem de períodos de cinco minutos, durante o Ano de Serviços, nos quais o Amazon EC2 esteve no estado de “Região Indisponível”. Se o cliente estiver utilizando o Amazon EC2 durante período inferior a 365 dias, o Ano de Serviços que lhe corresponde será, ainda assim, considerado como os 365 dias precedentes; no entanto, os dias anteriores a seu uso dos serviços serão considerados como de 100% de Disponibilidade na Região. Os períodos de inatividade operacional que ocorrerem antes de uma reivindicação bem-sucedida de Crédito de Serviço não poderão ser usados para efeito de reivindicações futuras. As medições da Porcentagem de Tempo de Atividade Anual excluem períodos de inatividade direta ou indiretamente decorrentes de qualquer Exclusão Relativa ao SLA do Amazon EC2 (definida adiante).
- “Região Indisponível” e “Indisponibilidade na Região” são expressões que indicam que mais de uma zona de disponibilidade em que o cliente estiver executando uma instância, dentro da mesma região, encontra-se “Indisponível” para ele.

#### **11.3.1.2. Compromissos de serviço e créditos de serviço**

Se a Porcentagem de Tempo de Atividade Anual de um cliente decair para menos de 99,95% no Ano de Serviços, o referido cliente se qualificará para o recebimento de um Crédito de Serviço equivalente a 10% de sua conta (excluindo os pagamentos efetuados uma única vez em razão de instâncias reservadas) no Período Elegível para Crédito. Para apresentar uma

reivindicação, o cliente não precisará esperar 365 dias contados da data em que começou a usar os serviços ou 365 dias contados da última reivindicação bem-sucedida. Permite-se a um cliente apresentar uma reivindicação em qualquer momento em que sua Porcentagem de Tempo de Atividade Anual no transcurso de 365 dias decair a nível inferior a 99,95%.

Aplicaremos quaisquer Créditos de Serviço unicamente contra futuros pagamentos relativos ao Amazon EC2 que se façam devidos pelo cliente de outra forma; ressalvado que à AWS será permitido atribuir Créditos de Serviço ao cartão de crédito que o cliente usou para pagar os serviços do Amazon EC2 no ciclo de cobrança durante o qual ocorreu o erro. Os Créditos de Serviço não darão ao cliente direito a reembolso ou a qualquer outro pagamento por parte da AWS. A AWS aplicará e emitirá o Crédito de Serviço somente se a quantia relativa ao crédito no ciclo de cobrança mensal aplicável for superior a US\$ 1,00. Os Créditos de Serviço não podem ser transferidos ou atribuídos a nenhuma outra conta. Salvo disposição em contrário no Contrato da AWS, a única e exclusiva reparação de um evento de indisponibilidade ou inoperacionalidade do Amazon EC2 ou de outra falha da AWS em fornecer os serviços do Amazon EC2 consiste no recebimento de um Crédito de Serviço (se assim qualificado) de acordo com os termos do presente SLA ou de acordo com o término do uso do Amazon EC2 pelo cliente.

### **11.3.1.3. Solicitação de crédito e procedimentos de pagamento**

Para receber um Crédito de Serviço, o cliente deve apresentar uma solicitação neste sentido enviando uma mensagem eletrônica endereçada a [aws-sla-request@amazon.com](mailto:aws-sla-request@amazon.com). Para efeito de sua qualificação, a solicitação de crédito deve: (i) incluir o número da conta do cliente no campo reservado ao assunto da mensagem (encontra-se o número da conta na parte superior da página *AWS Account Activity*); (ii) incluir, no corpo da mensagem, as datas e os horários de cada incidente de Região Indisponível que o cliente afirma ter ocorrido, mencionando os IDS das instâncias que estavam sendo executadas e que foram afetadas no momento da ocorrência do incidente; (iii) incluir os registros de solicitação ao servidor do cliente, os quais documentam os erros e corroboram a alegada interrupção de operacionalidade (quaisquer dados confidenciais ou sigilosos contidos nesses logs devem ser retirados ou substituídos por asteriscos); e (iv) fazer que a mensagem seja recebida por nós no prazo de trinta dias úteis a contar do último incidente relatado na reivindicação feita com base no SLA. Se a Porcentagem de Tempo de Atividade Anual da referida solicitação for confirmada pela AWS e ficar abaixo do nível de 99,95% no respectivo Ano de Serviços, providenciaremos o Crédito de Serviço em favor do cliente no prazo de um ciclo de cobrança subsequente ao mês em que ocorreu a correspondente solicitação. A omissão em providenciar a solicitação e fornecer as demais informações exigidas desqualificará o cliente quanto ao recebimento do Crédito de Serviço.

#### **11.3.1.4. Exclusões relativas ao SLA do EC2**

O Compromisso de Prestação de Serviços não se aplica às circunstâncias de indisponibilidade, interrupção ou término do EC2 ou a quaisquer outras questões relacionadas ao desempenho do EC2 que (i) resultem de uma interrupção como a descrita na Cláusula 6.1 do Contrato da AWS; (ii) forem causadas por fatores que fujam ao cabível controle da AWS, inclusive casos de força maior ou de acesso à internet e problemas correlatos que ocorram além do ponto de demarcação do Amazon EC2; (iii) resultem de quaisquer atos ou omissões do cliente ou de terceiros; (iv) resultem do equipamento, software ou outras tecnologias que o cliente usar e/ou do equipamento, software ou outras tecnologias de terceiros (à parte do equipamento de terceiros sob o controle direto da AWS); (v) resultem de falhas de instâncias individuais não atribuíveis à Indisponibilidade na Região; ou (vi) decorram da interrupção ou término pela AWS do direito conferido ao cliente de usar a plataforma do Amazon EC2 de acordo com o Contrato da AWS (em conjunto, as “Exclusões Relativas ao SLA do Amazon EC2”). Se a disponibilidade sofrer o impacto causado por outros fatores diferentes daqueles explicitamente relacionados neste acordo, a AWS, a seu exclusivo critério, emitirá um Crédito de Serviço levando em consideração tais fatores.

### **11.4. Questões específicas de conformidade**

Existem questões específicas de conformidade da AWS que são tratadas no artigo “Risco e Conformidade da Amazon Web Services”, citado anteriormente. As principais questões são resumidas aqui.

- **Propriedade sobre os controles físicos.** A AWS controla os componentes físicos da TI. O cliente possui e controla todo o resto, incluindo o controle sobre pontos de conexão e transmissão da rede. Para ajudar os clientes a compreenderem melhor os controles em vigor e como efetivamente eles operam, a Amazon sugere a consulta do relatório SAS 70 tipo II, bem como controles de segurança física detalhada e controles ambientais. Os clientes AWS que assinaram um acordo de confidencialidade com a AWS podem solicitar uma cópia do relatório SAS 70 tipo II.
- **Localização dos dados.** Os clientes AWS devem determinar a região física em que seus dados e seus servidores deverão estar localizados. A replicação de dados contidos em objetos no S3 é feita dentro do cluster regional em que os dados são armazenados e não são replicados para outros clusters de DATACENTERS em outras regiões. A AWS tem controle sobre a localização de dados, para que os dados permaneçam no local especificado pelo cliente.
- **Propriedade sobre os dados.** Os clientes AWS mantêm o controle e a propriedade sobre os seus dados. A Amazon reforça que a AWS não

mede esforços para proteger a privacidade dos clientes e se mantém atenta ao determinar as solicitações legais com as quais deve estar em conformidade.

- **Isolamento dos dados.** Segundo a Amazon, todos os dados armazenados pela AWS em nome dos clientes tem recursos de isolamento seguro e controle de capacidades. O S3 fornece controles de acesso de dados avançados.
- **Portabilidade dos dados.** A AWS permite que os clientes movam os dados para dentro e para fora da AWS conforme necessário.
- **Durabilidade dos dados.** O S3 oferece aos clientes uma infraestrutura de armazenamento durável. Os objetos são armazenados redundantemente no S3 em diversas instalações do S3 em uma região. Uma vez armazenados, o S3 mantém a durabilidade dos objetos ao detectar e reparar rapidamente qualquer redundância perdida. O S3 também verifica regularmente a integridade dos dados armazenados usando recursos sofisticados.
- **Acesso de terceiros e clientes aos DATACENTERS.** A AWS mantém um controle restrito de acesso aos DATACENTERS, mesmo para funcionários internos. O acesso de terceiros aos DATACENTERS AWS não é concedido, exceto quando for explicitamente aprovado pelo gerente responsável do DATACENTER, conforme as políticas de acesso da AWS. A Amazon sugere aos clientes consultar o relatório SAS 70 tipo II para verificar controles específicos referentes ao acesso físico, à autorização de acesso ao DATACENTER e a outros controles relacionados. As visitas de clientes aos DATACENTERS da AWS não são autorizados pela Amazon.
- **Acesso privilegiado.** A AWS fornece controles específicos SAS 70 para abordar a ameaça de acesso privilegiado inadequado, a certificação pública e as iniciativas de conformidade. Todas as certificações e os atestados de terceiros avaliam o acesso lógico e os controles preventivo e de detecção. Além disso, as avaliações periódicas de risco concentram-se em como o acesso privilegiado é controlado e monitorado.
- **Suporte ao gerenciamento de vulnerabilidades.** A AWS é responsável pela correção dos sistemas que fornecem suporte à disponibilização dos serviços ao cliente, tais como o *hypervisor* e os serviços de rede. Isso é feito como exigido pela política da AWS e em conformidade com a ISO 27001, NIST e os requisitos PCI. Os clientes controlam seus próprios sistemas operacionais convidados, software e aplicações, portanto, são responsáveis por seus próprios sistemas de aplicação de correções (patches). Aspectos da aplicação de patches serão vistos no capítulo 13.

- **Suporte a criptografia.** A AWS permite que os clientes usem seus próprios mecanismos de criptografia para quase todos os serviços, incluindo o armazenamento S3 e o processamento EC2. As sessões VPC também são criptografadas. Os clientes também podem usar tecnologias de criptografia de terceiros. Aspectos da criptografia serão tratados no capítulo 12.
- **Paralisações.** A AWS não exige que os sistemas sejam paralisados para executar a manutenção regular e a aplicação de correções de sistema. A manutenção da AWS e a aplicação de correções de sistema geralmente não afetam os clientes. A manutenção das instâncias lançadas é controlada pelo cliente.
- **Backup e Restore.** A AWS permite que os clientes façam seus backups em fitas usando seu próprio provedor de serviço de backup de fita. O backup em fita e o restore associado NÃO são serviços prestados pela AWS.

## 11.5. Referências bibliográficas

Amazon Web Services. **Risco e Conformidade da Amazon Web Services.** Maio 2011.

Cloud Security Alliance. **Security Guidance for Critical Areas of focus in Cloud Computing.** 2011.

Cockcroft, Adrian. **Ops, DevOps and PaaS (NoOps) at Netflix.** Artigo disponível na internet.

IBM. **Segurança e ampla disponibilidade em ambientes de computação em nuvem.** 2011.

ISACA. **CobIT 5: A Business Framework for the Governance and Management of Enterprise IT.** 2012.

Varia, Jinesh. **Projetando para a nuvem:** práticas recomendadas. Janeiro de 2010. Última atualização: janeiro de 2011.

Weill, Peter & Ross, Jeanne W. **Governança de Tecnologia da Informação.** M. Books, 2006.

# 12. Segurança

## 12.1. Introdução

A segurança da informação é um aspecto crucial para adoção da nuvem. A Amazon reforça que, em um ambiente *multi-tenant* e virtualizado, arquitetos de nuvem muitas vezes mostram-se preocupados com a segurança e com o risco. Eles estão certos: a adoção da nuvem representa um nível adicional de risco para o negócio.

A ISO 27001:2005 é o padrão para sistema de gestão da segurança da informação (*Information Security Management System – ISMS*) e foi publicada em outubro de 2005 pela ISO e pela IEC. Esta norma foi elaborada para estabelecer, implementar, operar, monitorar, revisar, manter e melhorar um ISMS. A adoção de um ISMS deve ser uma decisão estratégica para a organização. A especificação e implementação do ISMS de uma empresa são influenciadas pelas suas necessidades e objetivos, exigências de segurança, os processos empregados e o tamanho e a estrutura da organização.

A Amazon sugere também que a segurança deve ser tratada em grandes domínios. Por exemplo, no caso da segurança do aplicativo, as camadas de VIRTUALIZAÇÃO e de rede devem ser utilizadas para dar segurança à camada do aplicativo.

Reforça-se que, no caso de serviços IaaS (como a AWS), parte da segurança é de responsabilidade do provedor e parte é do cliente.

Este capítulo trata dos diversos aspectos que envolvem a segurança da informação na AWS. Aspectos de governança foram tratados no capítulo 11.

## 12.2. Conceitos

Segurança da informação é a prática de assegurar que os recursos que geram, armazenam ou proliferam as informações sejam protegidos contra quebra de confidencialidade, comprometimento da integridade e contra a indisponibilidade de acesso a tais recursos [Diógenes, Mauser, 2011].

### 12.2.1. Pilares

Os pilares da segurança da informação são ilustrados na **Figura 12-1**.

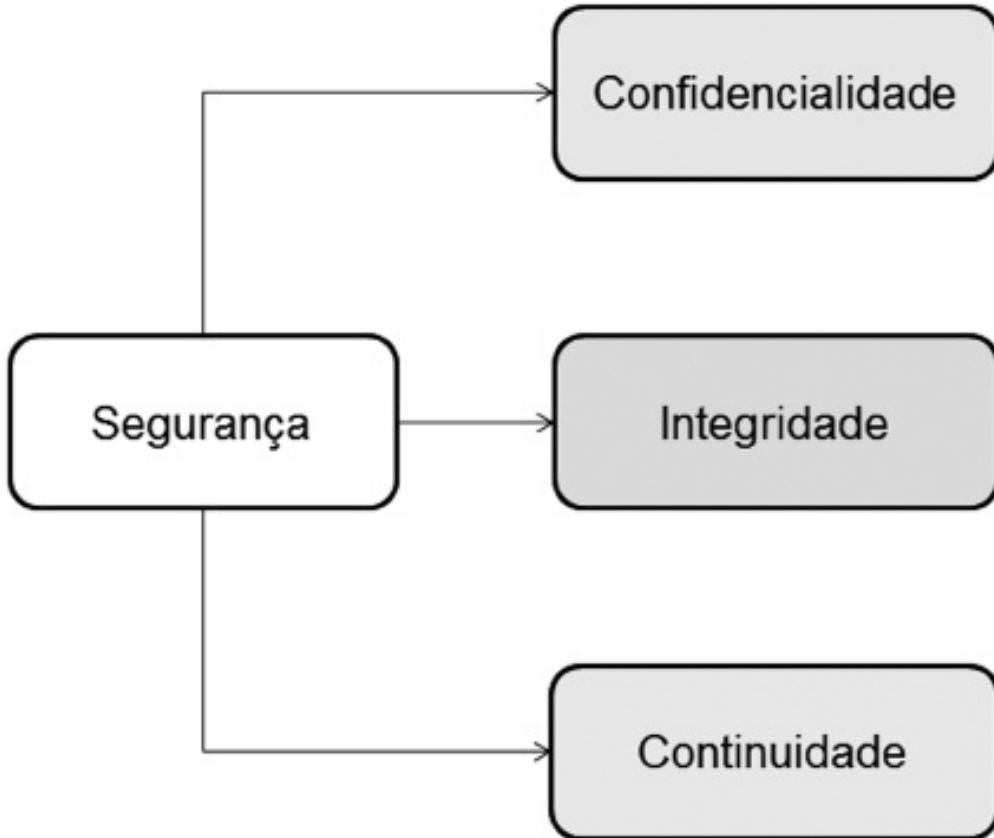


Figura 12-1 Pilares da segurança da informação

Explicando:

- Confidencialidade trata da prevenção de vazamento de informação para usuários e ou sistemas não autorizados.
- Integridade trata de preservar o dado na forma íntegra, ou seja, sem sofrer modificações não autorizadas.
- Disponibilidade trata de disponibilizar a informação quando se necessita.

Este capítulo trata das técnicas utilizadas para obtenção da confidencialidade e da integridade na nuvem AWS. A disponibilidade será abordada no capítulo 13.

## 12.2.2. Criptografia

Criptografia é a ciência de escrever em cífras ou em códigos, de forma a permitir que somente o destinatário a decifre e compreenda.

A criptografia permite transformar a informação em algo ilegível enquanto está sendo transmitida ou armazenada, fazendo com que usuários não autorizados não possam acessá-la. Assim, a criptografia tem a ver com dados transmitidos e armazenados.

Os sistemas de criptografia permitem fornecer confidencialidade,

integridade e não repúdio (não permite repudiar a culpa, caso seja realmente culpado). Os algoritmos utilizados pelos sistemas de criptografia vão garantir mais ou menos confidencialidade e integridade.

Os modelos de criptografia dividem-se em:

- **Criptografia simétrica (algoritmo de chave secreta)**, que utiliza apenas uma chave, tanto para criptografar como para decriptar.
- **Criptografia assimétrica (algoritmo de chave pública)**, que utiliza duas chaves: uma pública e uma privada. A chave pública pode ser distribuída abertamente e a chave privada é mantida secreta.

A AWS utiliza criptografia assimétrica como algoritmo principal e possui uma infraestrutura de chave pública (*Public Key Infrastructure – PKI*) interna com controle dos certificados emitidos e dos que precisam ser revogados.

### 12.2.3. Atualizações de segurança

Um aspecto importante é a aplicação das atualizações de segurança, para que haja resistência a ataques. Essas atualizações ocorrem para cada sistema operacional e são conhecidas como patches de segurança, fazendo parte do gerenciamento de configuração.

Ao longo do tempo, erros no software são descobertos e exigem patches de correção. A Amazon sugere que se garantam as seguintes diretrizes básicas para maximizar a segurança:

- Fazer o download de patches do site do fornecedor e atualizar as AMIs.
- Reimplantar instâncias AMIs novas e testar os aplicativos para garantir que os patches não corrompam nada.
- Certificar-se de que a AMI atualizada está implantada em todas as instâncias.
- Investir em scripts de teste para executar verificações de segurança periodicamente e automatizar o processo.
- Verificar se o software de terceiros está definido para as configurações mais seguras.
- Nunca execute os processos com acesso root ou administrador, a menos que seja extremamente necessário.

No caso da nuvem AWS, esta responsabilidade é dividida. Quando se trata de atualizações ligadas ao sistema virtualizado (*hypervisor*) esta responsabilidade é da AWS, mas quando se trata de atualizações do sistema operacional convidado esta responsabilidade é do cliente – devendo-se neste caso definir uma forma de fazer as atualizações com segurança. O cliente pode optar por instalar as atualizações automaticamente, fazer download das

atualizações e escolher quando instalá-las, ou pode ser apenas notificado sobre as atualizações.

Segundo a Amazon, as alterações de emergência, não rotineiras, e outras alterações de configuração da infraestrutura existente são autorizadas, conectadas, testadas, aprovadas e documentadas em conformidade com as normas do setor para sistemas similares. Atualizações de infraestrutura da AWS são feitas para minimizar qualquer impacto sobre o cliente e seu uso dos serviços. A AWS reforça que se comunicará com os clientes via e-mail ou através do console de status de serviço AWS (<http://status.aws.amazon.com/>) quando a utilização do serviço se tornar suscetível a situações negativas.

O processo de gerenciamento de mudança da AWS é projetado para evitar interrupções de serviço não intencionais e para manter a integridade do serviço. As alterações implantadas nos ambientes de produção são:

- **Revisadas:** processo de revisão trata dos aspectos técnicos de uma mudança.
- **Testadas:** os testes devem garantir que as mudanças, quando aplicadas, se comportem conforme o esperado e não impactem negativamente o desempenho.
- **Aprovadas:** alterações devem ser aprovadas como parte de um sistema de controle para que haja compreensão sobre o impacto nos negócios.

## 12.3. Segurança tradicional *versus* segurança na AWS

Uma solução de gestão de segurança tradicional envolve:

- **Serviços de firewall e VPN.** Serviços capazes de regular o tráfego de dados entre as redes da empresa e impedir a transmissão e recepção de tráfego nocivo ou não autorizado de uma rede para outra. Os equipamentos devem, por exemplo, implementar tecnologias de filtro de pacotes utilizando mecanismos de verificação de tráfego segundo tabela de estado de conexões.
- **Serviços de prevenção de intrusão.** Serviços providos por equipamentos capazes de identificar, prevenir e bloquear tentativas de intrusão e atividades maliciosas de rede entre os diversos segmentos de rede da empresa, incluindo o acesso à internet, a redes do tipo MPLS e até de redes de contingência baseadas em VPN.
- **Serviços de proxy/cache com filtro de conteúdo web.** Serviços responsáveis pela liberação e pelo bloqueio de acessos feitos pelos

usuários da rede corporativa a websites e assemelhados, conforme política de acesso à internet. Esses serviços deverão, ainda, implementar tecnologias de detecção e bloqueio de intrusão por meio de assinaturas e por análise de comportamento, com topologia IPS *in-line* em modo *pass-through/failover*. Deverão ser capazes de interromper tráfego de rede que tenha potencial para causar danos às informações ou ainda o consumo desnecessário de recursos de rede.

- **Serviços de SMTP antisspam.** Serviços que se referem à solução de bloqueio de e-mails não solicitados pelos usuários, capazes de impactar a produtividade de seus colaboradores e degradar o desempenho dos sistemas e redes corporativas, além de potencialmente comprometer a segurança das informações por eles custodiadas.
- **Serviços de firewall de aplicação.** São responsáveis por monitorar e bloquear entrada, saída, solicitação de acesso e chamadas de sistema a aplicações disponibilizadas em servidores web, segundo diretrizes de segurança definidas para a empresa. Tais serviços deverão ser prestados com uso de elementos capazes de operar na camada de aplicação como um proxy, de modo a inspecionar conteúdo do tráfego de aplicações e bloquear tentativas de intrusão, vírus, exploração de vulnerabilidades e comunicações mal formatadas.
- **Serviços de consolidação e correlacionamento de eventos.** Serviços responsáveis por coletar, armazenar, processar, monitorar e correlacionar logs de ativos e servidores de rede, bem como da própria solução de segurança fornecida, de modo a executar ações reativas e proativas, como envio de notificações e alertas aos administradores da rede da empresa.
- **Serviços de gestão de vulnerabilidades.** Estes serviços devem ser capazes de detectar, inventariar e avaliar vulnerabilidades encontradas nos sistemas e recursos da empresa e na solução de segurança utilizada, especialmente quanto ao impacto no ambiente computacional e ao risco inerente à segurança das informações custodiadas. Deverá englobar instalação de agentes e validação de conformidade por meio de monitoração periódica e por demanda.
- **Gestão da segurança da informação.** É um aspecto-chave de qualquer instalação empresarial. Muitas vezes, na solução tradicional este serviço é terceirizado com base em acordos de nível de serviço.

Boa parte desses serviços são realizados pela própria AWS, incluindo os custos e o gerenciamento.

A AWS tem responsabilidade sobre as camadas de infraestrutura, hardware e virtualização, incluindo serviços responsáveis por gerenciar remotamente e administrar equipamentos e softwares componentes da solução de segurança fornecida, envolvendo identificação de eventos que podem comprometer a segurança dos serviços de TI, manutenção da infraestrutura de segurança atualizada, mapeamento e execução de processos de resposta a incidentes de segurança, suporte à solução de segurança, avaliação periódica de configurações, entre outros, sob regime 24x7 (24 horas por dia, sete dias por semana).

A segurança das camadas, incluindo sistema operacional convidado, dados, aplicativos e softwares de rede, são de responsabilidade do cliente. A AWS oferece inúmeras ferramentas que permitem que o cliente otimize a segurança nessas camadas.

## 12.4. Segurança em domínios na AWS

### 12.4.1. Contexto

A AWS utiliza estrutura e políticas de segurança de informação baseadas no CobIT (*Control Objectives for Information and Related Technology*) e na ISO/IEC 27001.

- CobIT é um modelo de estrutura de controle interno orientado para o entendimento e o gerenciamento de riscos associados ao uso da TI. O CobIT possui uma estrutura para controle interno que utiliza padrões aceitos mundialmente como de melhor prática.
- ISO/IEC 27001 é um padrão para sistema de gestão da segurança da informação (SGSI) publicado em outubro de 2005 pelo International Organization for Standardization e pelo International Electrotechnical Commission. Esta norma foi elaborada para estabelecer, implementar, operar, monitorar, revisar, manter e melhorar um SGSI.

A AWS também verifica aspectos de vulnerabilidades, e eventualmente o cliente pode conduzir um teste em suas instâncias (desde que não viole a política de uso aceitável da AWS). A AWS também implementou internamente um programa de segurança da informação formal projetado para proteger a confidencialidade, integridade e disponibilidade de sistemas e dados dos clientes.

A Amazon reforça que segurança deve ser prioridade em um ambiente *multi-tenant* e virtualizado. O princípio básico é que a segurança deve ser implementada em cada nível da arquitetura de TI.

O modelo em domínios mostrado na **Figura 12-2** e baseado no framework de segurança da IBM ilustra os domínios envolvidos com a segurança. A camada “Acesso” foi vista no capítulo 3 e será reforçada aqui.

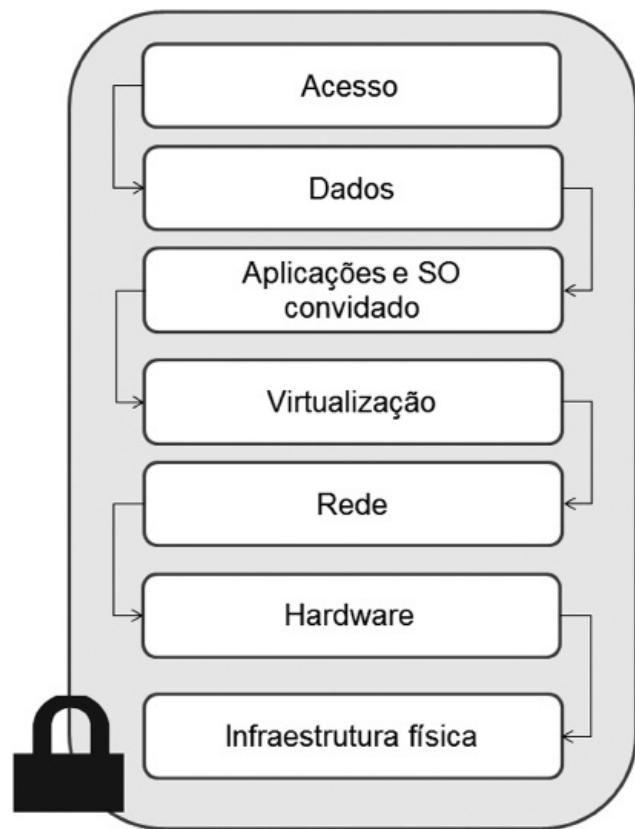


Figura 12-2 Framework de segurança

A segurança da infraestrutura, do hardware e da camada de virtualização é tratada pela AWS, o que pode ser um benefício adicional do uso da nuvem, considerando que esses dois aspectos saem da responsabilidade do cliente. A segurança dos softwares específicos de rede e a segurança em nível do aplicativo é de responsabilidade do cliente, e é natural que este considere a opção de implementar as melhores práticas aplicáveis. A segurança da conta e dos usuários relacionados à conta é de responsabilidade do cliente também.

## 12.4.2. Segurança de acesso

A AWS oferece três tipos de credenciais de segurança (conforme visto no capítulo 3):

- **Credenciais de acesso:** chaves de acesso, certificados X.509 e pares de chave.
- **Credenciais de conexão:** endereço de e-mail, senha e dispositivos AWS *Multi-Factor Authentication*.
- **Identificadores de conta:** ID da conta da AWS e ID de usuário canônico.

### 12.4.2.1. Segurança com IAM

O IAM é a peça central de segurança da conta. Ele permite que o cliente crie múltiplos usuários e gerencie permissões para cada um a partir de sua conta AWS.

Um usuário é uma identidade (dentro de uma conta da AWS do cliente) com credenciais de segurança exclusivas que podem ser usadas para acessar os serviços AWS. O IAM elimina a necessidade de compartilhar senhas ou chaves de acesso e facilita a ativação e a desativação do acesso do usuário, conforme apropriado. Maiores detalhes sobre o uso do IAM podem ser encontrados no capítulo 3.

O IAM permite que o cliente implemente as melhores práticas de segurança, ao atribuir credenciais exclusivas para cada usuário dentro da conta AWS e conceder somente as permissões que os usuários precisam a fim de acessar os recursos da AWS necessários para a realização do seu trabalho. Como padrão, o IAM é seguro; os novos usuários não têm de acessar os recursos da AWS até que permissões sejam explicitamente concedidas.

O IAM é integrado nativamente com a maioria dos serviços AWS. Nenhuma API de serviço foi alterada para oferecer suporte ao IAM, e aplicações e ferramentas baseadas em APIs de serviço da AWS continuarão a funcionar quando o IAM estiver em uso.

Como boa prática, a Amazon reforça que deve-se minimizar o uso das

credenciais de conta AWS tanto quanto for possível quando estiver interagindo com serviços AWS e aproveitar as credenciais de usuário IAM para acessar recursos e serviços AWS.

#### **12.4.2.2. Segurança com MFA**

A MFA é uma camada adicional de segurança que oferece um melhor controle das configurações de conta da AWS. Ao habilitar esse recurso de credencial de conexão, o cliente necessitará fornecer um código de uso único de seis dígitos, além das suas credenciais de nome de usuário padrão e de senha antes que o acesso seja concedido às suas configurações da conta AWS ou aos recursos e serviços da AWS. O cliente obterá esse código de uso único a partir de um dispositivo de autenticação, que estará em sua posse. Isso é chamado de autenticação *multi-gateway* de internet porque dois fatores são verificados antes de o acesso ser concedido: os clientes precisarão fornecer seu nome de usuário (e-mail da Amazon, no caso de uma conta AWS) e senha (o primeiro “fator”: algo que você sabe), e o código preciso de seu dispositivo de autenticação (o segundo “fator”: algo que você tem).

Os clientes podem ativar dispositivos MFA para suas contas da AWS, bem como para os usuários criados em contas da AWS com o IAM.

#### **12.4.2.3. Segurança com rotação de chaves de acesso e certificados**

Pelos mesmos motivos que tornam importante a alteração de senhas frequentemente, a AWS recomenda que os clientes façam regularmente a rotação das suas chaves de acesso e certificados. Para permitir que o cliente possa fazer isso sem um possível impacto na disponibilidade dos seus aplicativos, a AWS é compatível com várias chaves de acesso e certificados simultâneos. Com esse recurso, os clientes podem fazer a rotação das chaves e certificados dentro e fora de operação de modo regular, sem qualquer tempo de inatividade para o aplicativo.

A rotação de chaves e certificados pode ajudar a diminuir os riscos de perda ou comprometimento de certificados ou chaves de acesso. As APIs do IAM da AWS permitem que o cliente faça a rotação das chaves de acesso da sua conta AWS, bem como para usuários criados sob a sua conta AWS usando o IAM.

### **12.4.3. Segurança dos dados**

#### **12.4.3.1. Dados em trânsito e dados em repouso**

A equipe AWS reforça que dados em trânsito e em repouso devem ser protegidos.

##### **Proteja dados em trânsito**

Se for necessário trocar informações confidenciais entre um navegador e

um servidor web, a Amazon recomenda configurar o SSL em sua instância de servidor. Será necessário um certificado de uma autoridade de certificação externa. A chave pública incluída no certificado autentica o servidor para o navegador e serve como base para criar a chave de sessão compartilhada usada para criptografar os dados em ambas as direções.

O time AWS sugere que seja criada uma VPC. Isso permite o uso de recursos isolados dentro da nuvem AWS e a conexão destes recursos diretamente ao DATACENTER corporativo usando o padrão IPSec para conexões criptografadas.

### **Proteja dados em repouso**

Se existe preocupação com o armazenamento de dados confidenciais na nuvem, você deve criptografar os dados (arquivos individuais) antes de enviá-los para a nuvem.

No EC2, a criptografia de arquivo depende do sistema operacional.

Não importa qual sistema operacional ou tecnologia você escolha, a criptografia de dados em repouso apresenta um desafio: gerenciar as chaves usadas para criptografar os dados. Se você perder as chaves, perderá seus dados para sempre, e se as suas chaves forem comprometidas, os dados podem estar em risco. Portanto, certifique-se de estudar os recursos de gerenciamento de chaves de qualquer produto que você escolha e estabeleça um procedimento que minimize o risco de perda das chaves.

As instâncias do EC2 executando Linux podem montar volumes de EBS usando sistemas de arquivos criptografados. Independentemente de qual abordagem você escolha, a criptografia de arquivos e volumes no EC2 ajuda a proteger os arquivos e os dados de log para que somente os usuários e processos no servidor possam ver os dados decriptografados.

As instâncias do EC2 que executam o Windows, por exemplo, podem usar o recurso interno EFS (*Encrypting File System*), que irá lidar com a criptografia e a decriptografia de arquivos e pastas automaticamente e tornar o processo transparente para os usuários.

No entanto, apesar do nome, o EFS não criptografa o sistema de arquivos inteiro; em vez disso, ele criptografa arquivos individuais. Se for necessário ter um volume completamente criptografado, a Amazon sugere considerar o uso do TrueCrypt5, que tem o código aberto e se integra muito bem com volumes formatados pelo NTFS do EBS.

#### **12.4.3.2. Segurança dos dados no S3**

Um usuário autenticado pode listar as chaves e criar ou substituir os objetos em um *bucket* do S3 somente se tiver sido concedida a permissão de ler e escrever em uma ACL no nível do *bucket* ou através de permissões

concedidas a ele pelo IAM. ACLs do *bucket* e do objeto são independentes e o objeto não herda a ACL de um *bucket*.

O S3 fornece uma infraestrutura de armazenamento altamente durável projetada para armazenamento de dados de missão crítica e para dados primários. Objetos são armazenados redundantemente em múltiplos dispositivos através de várias instalações de uma região específica. Para ajudar a garantir durabilidade de dados, as operações PUT e PUT Object Copy do S3 sincronicamente armazenam dados em múltiplas instalações antes de retornar um êxito. Uma vez armazenado, o S3 mantém a durabilidade dos objetos, rapidamente detectando e reparando qualquer redundância perdida.

O S3 também verifica periodicamente a integridade dos dados armazenados usando as somas de verificação. Se o S3 detectar corrupção de dados, estes são reparados usando dados redundantes. Além disso, o S3 calcula as somas de verificação em todo o tráfego de rede para detectar corrupção de pacotes de dados ao armazenar ou recuperar dados.

O armazenamento padrão do S3:

- Utiliza acordo de nível de serviço.
- É projetado para fornecer disponibilidade de 99,99% e durabilidade de 99,99999999% de objetos ao longo de um determinado ano.
- É projetado para sustentar a perda simultânea de dados em duas instalações.

## **Encriptação de dados**

A encriptação de dados oferece maior segurança para os dados de objeto armazenados no *bucket* do S3. Pode-se criptografar dados no cliente e enviar dados criptografados para o S3. Nesse caso, o cliente gerencia o processo e as chaves de criptografia e ferramentas relacionadas. Opcionalmente, é possível usar o recurso de criptografia do servidor: o S3 criptografa os dados do objeto antes de salvá-lo em discos nos DATACENTERS e descriptografa quando os objetos forem acessados, liberando o cliente das tarefas de gerenciamento da criptografia, das chaves de criptografia e das ferramentas relacionadas.

## **RRS (Reduce Redundant Storage)**

Esta opção de armazenamento no S3 permite aos clientes reduzirem custos para armazenamento de dados não críticos, com níveis de redundância mais baixos do que o armazenamento padrão do S3. A opção RRS armazena objetos em vários dispositivos em diversas instalações, fornecendo quatrocentas vezes a durabilidade de uma unidade de disco típica, mas não

replica a mesma quantidade de objetos que o armazenamento padrão do S3 e, portanto, é mais econômico. O armazenamento de redundância reduzida (RRS) é respaldado pelo acordo de nível de serviço do S3 e projetado para fornecer 99,99% de durabilidade e 99,99% de disponibilidade de objetos em um determinado ano. Esse nível de durabilidade corresponde a uma perda anual esperada média de 0,01% de objetos e sustenta a perda de dados em uma única instalação. A **Tabela 12-1** compara o S3 com o RRS em termos de disponibilidade e durabilidade.

**Tabela 12-1 SLA de S3 e RRS**

	S3	RRS
<b>Disponibilidade</b>	99,99%	99,99%
<b>Durabilidade</b>	99,99999999%	99,99%
<b>Importante</b>	Projetado para suportar a perda de dados concorrentes em duas instalações	Projetado para suportar a perda de dados em uma única instalação

### Controle de versão

Controle de versão é um meio de manter diversas variantes de um objeto no mesmo *bucket*. Em um balde, por exemplo, você pode ter dois objetos com a mesma chave, mas versões diferentes, como photo.gif (versão 111111) e photo.gif (versão 121212). É possível habilitar o controle de versão para impedir que objetos sejam excluídos ou substituídos por engano, ou para que seja possível recuperar versões anteriores deles (objetos de arquivo).

*Buckets* podem estar em um dos três estados: sem versões (o padrão), controle de versões ativado ou versionamento suspenso. Uma vez habilitada a versão em um *bucket*, ele nunca pode retornar a um estado sem versão. Pode-se, no entanto, suspender o controle de versão em qualquer *bucket*.

#### 12.4.4. Segurança da aplicação

A segurança da aplicação é uma questão crucial para a segurança na nuvem AWS e é de responsabilidade do cliente. A segurança já deve ser considerada quando do desenvolvimento da aplicação. Implicações de segurança na navegação web, utilização de scripts e riscos associados ao vazamento de informações impostos pelos cookies devem ser cuidadosamente pensados. Está fora do escopo deste livro tratar deste aspecto em profundidade.

#### 12.4.5. Segurança do sistema operacional convidado

Instâncias virtuais são controladas pelo cliente. Os clientes têm acesso completo à raiz ou ao controle administrativo sobre aplicativos, serviços e

contas. A AWS não tem quaisquer direitos de acesso para instâncias de cliente e não pode efetuar login no sistema operacional convidado.

A AWS recomenda um conjunto básico de melhores práticas de segurança, recomendadas para incluir a desativação do acesso somente por senha para seus *hosts* e a utilização de alguma forma de autenticação *multi-factor* para obter acesso às suas instâncias (ou a um mínimo acesso SSH versão 2 com base em certificado). Além disso, os clientes devem empregar um mecanismo de escalonamento de privilégio com registro em uma base por usuário.

## 12.4.6. Segurança da virtualização

### 12.4.6.1. Segurança do EC2

Cada instância do EC2 é protegida por um ou mais grupos de segurança, nomeando conjuntos de regras que especificam em qual ingresso (ou seja, entrada) o tráfego de rede deve ser entregue à sua instância. Portas TCP e UDP podem ser especificadas, códigos e tipos ICMP e endereços de origem. Os grupos de segurança oferecem proteção básica com um *firewall* para instâncias em execução.

Outra forma de restringir o tráfego de entrada é configurar *firewall* baseado em software para as instâncias. As instâncias do Windows podem usar o *firewall* integrado. As instâncias do Linux podem usar o *netfilter* e o *iptables*.

A AWS estabeleceu procedimentos e políticas formais para delinear as normas mínimas de acesso lógico aos *hosts* da plataforma e da infraestrutura AWS. A AWS exige que funcionários com necessidade de acesso aos dados dos clientes passem por uma detalhada verificação de antecedentes (conforme o permitido por lei) proporcional ao seu cargo e ao nível de acesso a dados. As políticas também identificam as responsabilidades funcionais para a administração de acesso lógico e de segurança.

### 12.4.6.2. Níveis de segurança no EC2

A segurança no EC2 é fornecida em vários níveis: no nível do sistema operacional (SO) do sistema *host*, no nível do sistema operacional da instância virtual ou sistema operacional convidado, no nível do *firewall* e em chamadas de API assinadas. Cada um desses itens amplia os recursos de segurança dos outros. O objetivo é oferecer proteção para que dados contidos no EC2 não sejam interceptados por sistemas ou usuários não autorizados e possam eles mesmos fornecer instâncias do EC2 que sejam tão seguras quanto possível, sem sacrificar a flexibilidade de configuração que os clientes exigem.

A **Figura 12-3** ilustra a localização do grupo de segurança.

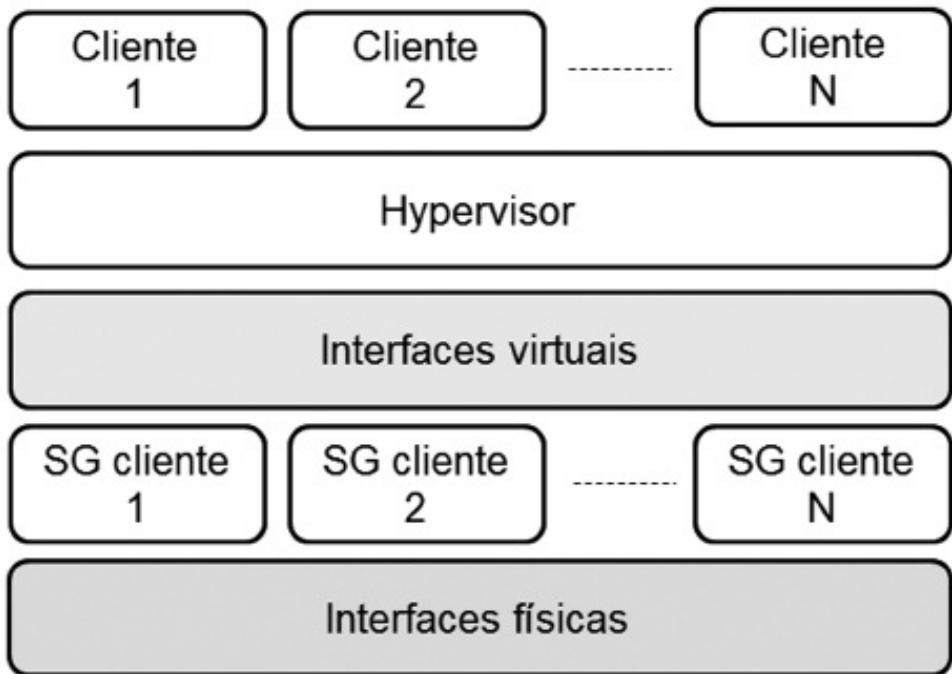


Figura 12-3 Localização do grupo de segurança

- **Sistema operacional de *hosts* de administração:** os administradores com necessidade de acessar o plano de gerenciamento são solicitados a usar uma autenticação *multi-gateway* para obter acesso aos *hosts* de uso específico de administração. Esses *hosts* administrativos são sistemas especificamente concebidos, construídos, configurados e reforçados para proteger o plano de gestão da nuvem. Esse acesso é registrado e auditado. Quando um funcionário não tem mais necessidade de acessar o plano de gestão, os privilégios e o acesso a esses *hosts* e sistemas pertinentes são revogados.
- **Grupo de segurança:** o EC2 fornece uma solução de grupo de segurança (*firewall*) completa; este *firewall* de entrada obrigatório é configurado no modo padrão de negar tudo, e os clientes do EC2 devem explicitamente abrir as portas necessárias para permitir o tráfego de entrada. O tráfego pode ser restrito por protocolo, por porta de serviço, bem como por endereço IP de origem (bloco IP ou roteamento sem classe entre domínios CIDR). O *firewall* pode ser configurado em grupos, permitindo que classes diferentes de instâncias tenham regras diferentes. Considere, por exemplo, o caso de um aplicativo web tradicional de três níveis. O grupo para os servidores web teria a porta 80 (HTTP) e/ou a porta 443 (HTTPS) aberta para a internet. O grupo para os servidores de aplicativos teria a porta 8000 (específico do aplicativo) acessível somente para o grupo de servidor web. O grupo para os servidores de banco de dados teria a porta 3306 (MySQL) aberta apenas para o grupo de

servidor de aplicativo. Todos os três grupos permitiriam o acesso administrativo na porta 22 (SSH), mas apenas a partir da rede corporativa do cliente. Aplicativos altamente seguros podem ser implantados usando esse mecanismo. A **Figura 12-4** ilustra a solução AWS.

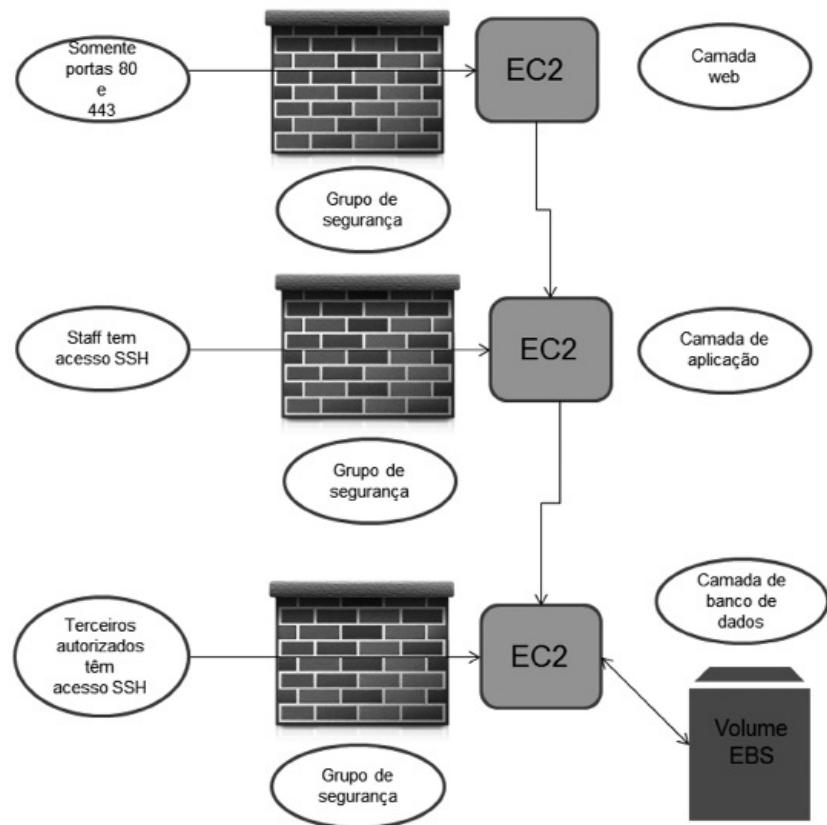


Figura 12-4 Solução de firewall do EC2

O *firewall* não é controlado através de SO convidado; em vez disso, exige o certificado X.509 do cliente e chave para autorizar alterações, acrescentando assim uma camada extra de segurança.

A AWS pode conceder acesso granular para diferentes funções administrativas sobre as instâncias e o *firewall*, portanto, que permita ao cliente implementar segurança adicional através da separação de funções. O nível de segurança proporcionado pelo *firewall* é uma função de quais portas são abertas pelo cliente e para que finalidade e duração. O estado padrão é negar todo o tráfego de entrada, e os clientes devem planejar cuidadosamente o que será aberto quando criarem e protegerem os seus aplicativos. O gerenciamento de tráfego instruído e o design de segurança ainda são necessários em uma base por instância.

A AWS ainda incentiva os clientes a aplicarem filtros adicionais por instância com *firewalls* baseados em *host* como *IPtables* ou o *firewall* do Windows e até utilizar VPNs. Isto pode restringir tanto o tráfego de entrada quanto o de saída em cada instância. Chamadas de API para iniciar e encerrar instâncias, alterar parâmetros de *firewall* e executar outras funções são assinadas pela chave de acesso secreto do cliente da Amazon, que poderia ser qualquer chave de acesso secreto das contas AWS ou a chave de acesso secreto de um usuário criado com o IAM. Sem o acesso à chave secreta do cliente, as chamadas de API do EC2 não podem ser realizadas em seu nome. Além disso, as chamadas de API podem ser criptografadas com o SSL para manter a confidencialidade. A Amazon sempre recomenda o uso de pontos de acesso de API protegidos por SSL.

O IAM da AWS também permite que um cliente possa controlar quais APIs criadas por um usuário no IAM terão permissão para chamar.

#### 12.4.6.3. Segurança do hypervisor

O EC2 utiliza atualmente uma versão personalizada do *hypervisor* Xen. Visto que os convidados paravirtualizados dependem do *hypervisor* para fornecer suporte a operações que normalmente requerem um acesso privilegiado, o SO convidado não tem acesso elevado à CPU. Qualquer CPU x86 fornece quatro modos de privilégio separados: 0-3, chamados de anéis, sendo o anel 0 o mais privilegiado e o 3 o menos privilegiado. O SO *host* é executado em anel 0. No entanto, em vez de ser executado no anel 0 como a maioria dos sistemas operacionais, o SO convidado é executado em um anel 1, menos privilegiado, e os aplicativos são executados em um ainda menos privilegiado anel 3. Esta virtualização explícita dos recursos físicos leva a uma separação clara entre convidado e *hypervisor*, resultando na separação de segurança adicional entre os dois.

Diferentes instâncias em execução na mesma máquina física são isoladasumas das outras através do Xen. A Amazon é ativa participante da

comunidade Xen, o que garante o emprego dos últimos desenvolvimentos. Além disso, o *firewall* AWS reside na camada *hypervisor*, entre a interface de rede física e a interface da instância virtual. Todos os pacotes devem passar por esta camada; assim, uma instância vizinha não tem mais acesso a essa instância do que qualquer outro *host* na internet e pode ser tratada como se estivesse em *hosts* físicos separados. A memória RAM é separada com mecanismos similares.

Instâncias do cliente não têm acesso a dispositivos de disco, mas são apresentadas como discos virtuais. A camada de virtualização de propriedade de disco da AWS automaticamente redefine cada bloco de armazenamento utilizado pelo cliente, garantindo que os dados de um cliente nunca sejam involuntariamente expostos a outro. A AWS recomenda que os clientes protejam seus dados através de meios adequados. Uma solução comum é executar um arquivo criptografado no topo do dispositivo de disco virtual.

#### 12.4.6.4. Segurança do EBS

O acesso ao volume EBS é restrito à conta da AWS que criou o volume e aos usuários sob a conta da AWS criada com o IAM, se o usuário tiver concedido acesso às operações EBS, negando assim a permissão para todas as outras contas e usuários da AWS para exibir ou acessar o volume. No entanto, um cliente pode criar *snapshots*.

#### 12.4.7. Segurança da rede

##### 12.4.7.1. Ataques

A AWS fornece proteção contra problemas de segurança de rede, e o cliente pode implementar mais proteção ainda. A seguir são descritos ataques comuns de rede e como a AWS cuida da segurança.

- **DDos.** O Ataque Distribuído de Negação de Serviço (*Distributed Denial of Service* – DDoS) é uma distribuição sincronizada de requisições tendo um determinado endereço IP como alvo. A AWS reforça que utiliza técnicas proprietárias de redução de DDoS. Além disso, as redes da AWS têm hospedagem múltipla através de vários provedores para alcançar a diversidade de acesso à internet.
- **MITM.** O ataque a intermediários (*Man in the Middle* – MITM) caracteriza-se por escutar o envio de dados no meio e capturar estes dados. O MITM requer um software para ser executado no *host* que vai fazer o ataque de forma que este possa interceptar as ações e alterá-las antes de chegar ao destino. Todas as APIs da AWS estão disponíveis através de pontos de acesso protegidos por SSL que fornecem autenticação de servidor. As AMIs do EC2 automaticamente geram novos certificados de *host* SSH na primeira inicialização e os registram no console da instância. Os clientes, em seguida, podem usar as APIs seguras para chamar o console e

acessar os certificados de *host* antes de fazer o login na instância pela primeira vez. Os clientes são incentivados a usar o SSL para todas as suas interações com a AWS.

- ***Spoofing*.** O *spoofing* acontece no nível IP, na arquitetura TCP/IP. Este tipo de ataque troca o IP de origem no cabeçalho IP para parecer que o pacote está vindo de uma origem diferente. As instâncias EC2 não podem enviar tráfego de rede falsificado. A infraestrutura de *firewall* baseada em *host* controlada pela AWS não permitirá que uma instância envie tráfego com um endereço IP de origem diferente.
- ***Packet sniffing por outros clientes*.** Uma instância virtual que esteja sendo executada em modo promíscuo não pode receber ou “farejar” o tráfego que se destina a uma instância virtual diferente. Mesmo que os clientes possam colocar suas interfaces em modo promíscuo, o *hypervisor* não disponibilizará nenhum tráfego que não seja endereçado a eles. Mesmo duas instâncias virtuais que são pertencentes ao mesmo cliente localizado no mesmo *host* físico não podem escutar tráfego umas das outras. Ataques como envenenamento de cache ARP não funcionam no EC2 e na VPC. Enquanto o EC2 não oferece proteção ampla contra a tentativa mal-intencionada de um cliente de ver dados de outro, todos os clientes devem criptografar o tráfego sensível como uma prática padrão.

#### 12.4.7.2. Teste de penetração e teste de vulnerabilidade

A realização de um teste de penetração (*Pen Test*) deve ser comunicada à AWS para efeito de aprovação. O *Pen Test* simula uma tentativa de invasão no ambiente do cliente dentro da AWS e é conduzido por um *host* de internet fora da AWS.

O teste de vulnerabilidade é mais simples e se caracteriza pela execução de um software que identifica possíveis vulnerabilidades. Também deve ser aprovado pela AWS. Neste caso são realizadas varreduras de portas e de vulnerabilidade.

Varreduras de portas não autorizadas pelos clientes do EC2 são uma violação da política de uso aceitável da AWS. Quando uma varredura de porta não autorizada é detectada ela é interrompida e bloqueada. As varreduras de portas de instâncias do EC2 são geralmente ineficazes, porque, por padrão, todas as portas de entrada nas instâncias do EC2 estão fechadas e só são abertas pelo cliente. O gerenciamento rigoroso dos grupos de segurança pode atenuar ainda mais a ameaça de varreduras de portas. Se o cliente configura o grupo de segurança para permitir o tráfego de qualquer fonte para uma porta específica, essa porta específica ficará vulnerável a uma varredura de portas e pode ser descoberta facilmente.

Varreduras de vulnerabilidade identificam no *host* de destino atualizações de segurança faltantes e sinalizam que vulnerabilidades podem ser exploradas por terceiros.

A aprovação avançada para esses tipos de varreduras pode ser iniciada pelo envio de uma solicitação através do site da Amazon.

#### 12.4.7.3. Segurança na VPC

A segurança dentro da VPC começa com o próprio conceito de VPC e se estende para incluir grupos de segurança, listas de controle de acesso de rede (ACLs), roteamento e gateways externos. Cada um desses itens é complementar e permite fornecer uma rede isolada e segura, que pode ser estendida por meio da habilitação seletiva do acesso à internet ou conectividade privada para outra rede.

A VPC é uma rede isolada dentro da nuvem. Por padrão, não possui conectividade externa estando isolada das outras, podendo assim utilizar sobreposição de endereços IP. Pode criar e anexar um gateway de internet ou um gateway VPN para estabelecer a conectividade externa. Os controles na VPC são mostrados na **Figura 12-5** e resumem os seus componentes de segurança:

- **Instâncias:** isolamento de instâncias e proteção contra detecção.
- **Security Group (SG):** ativa a filtragem do tráfego de entrada e saída de uma instância.
- **Subnets:** ataques tradicionais são bloqueados, como MAC *spoofing* e ARP *spoofing*.
- **ACLs de rede:** filtros de tráfego sem monitoração do estado que se aplica a todo o tráfego de entrada e saída dentro de uma *subnet*.
- **Tabelas de rotas:** todo o tráfego de saída é processado pela tabela de roteamento para determinar o destino.
- **Gateway VPN:** permite conectividade privada entre a VPC e outra rede. O tráfego dentro do gateway VPN é isolado de outros gateways VPN. Os clientes podem se conectar ao gateway VPN utilizando um gateway de cliente com conexões baseadas em chaves pré-compartilhadas e endereços IP dos dispositivos de gateway dos clientes.
- **Gateway de internet:** pode ser ligado diretamente à VPC para permitir ligação direta com o S3, outros serviços e a internet. Cada instância que deseja esse acesso deve ter um IP elástico ou rotear o tráfego por instância NAT.

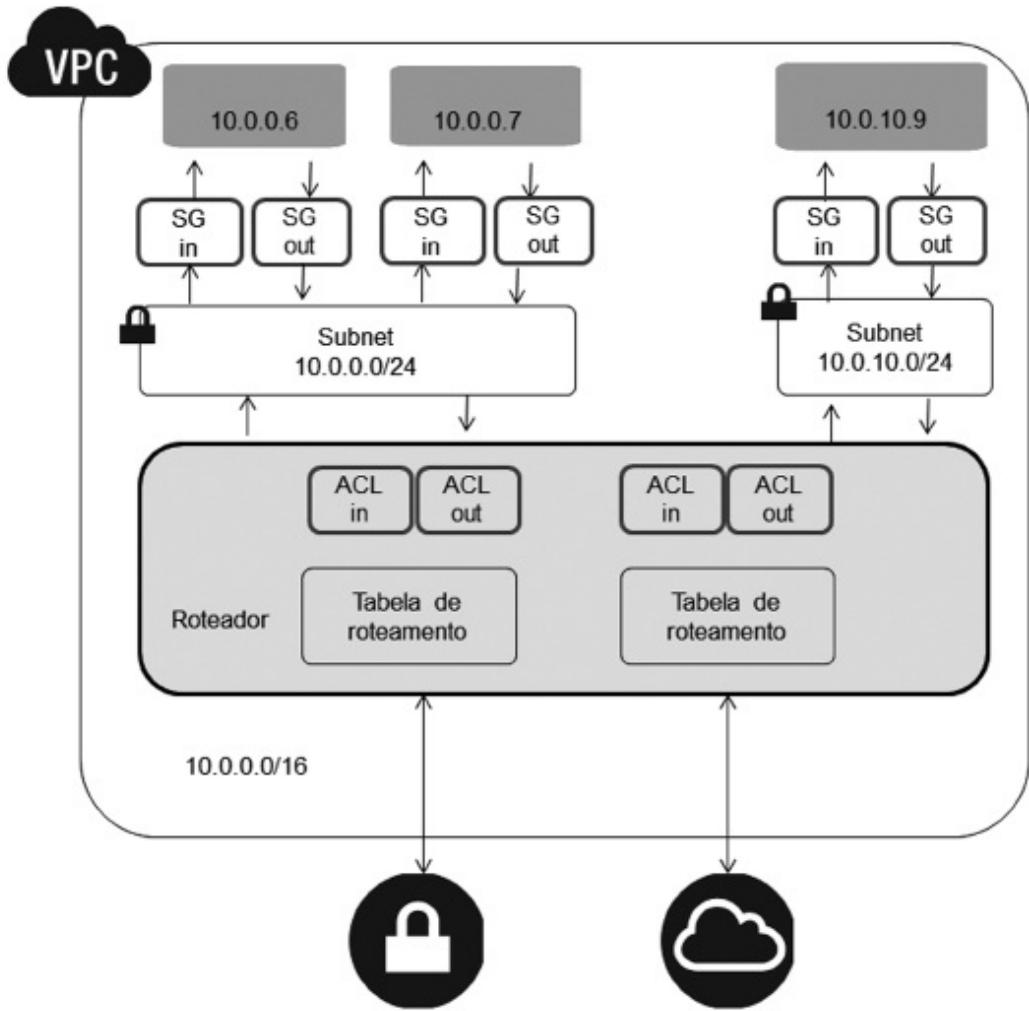


Figura 12-5 Resumo dos recursos de segurança na VPC

Um aspecto importante é entender o papel dos grupos de segurança e ACLs de rede na AWS. A **Tabela 12-2** ilustra as principais diferenças entre eles.

**Tabela 12-2 Grupos de segurança e ACLs de rede**

Grupos de segurança	ACL de rede
Operam no nível da instância.	Operam no nível da <i>subnet</i> .
Especificam-se regras ALLOW, para entrada e saída.	Especificam-se regras ALLOW e DENY, para entrada e saída.
São <i>stateful</i> .	São <i>stateless</i> .
É necessário especificar o grupo de segurança para ser aplicável a uma instância.	É aplicado automaticamente a todas as instâncias de uma <i>subnet</i> .

## 12.4.8. Segurança do hardware

A segurança do hardware é de responsabilidade da AWS. Esta segurança envolve diversos aspectos do hardware, incluindo segurança da BIOS, não

reprogramação desta e desativação de unidades de armazenamento de disco quando atingem o final da vida útil. Os procedimentos da AWS incluem um processo de desativação projetado para impedir que os dados do cliente sejam expostos a pessoas não autorizadas. A AWS usa técnicas detalhadas no DoD 5220.22-M ou NIST 800-88 para destruir dados como parte do processo de desativação. Se um dispositivo de hardware é incapaz de ser desativado usando esses procedimentos, ele será inutilizado ou fisicamente destruído em conformidade com as práticas padrão do setor.

#### 12.4.9. Segurança física

A segurança física trata do controle do acesso físico aos DATACENTERS. Os recursos computacionais não devem ser acessados fisicamente por todos os funcionários e devem diferenciar visitantes e funcionários. Na segurança física utiliza-se o conceito de camadas de mais ou menos segurança, e um funcionário pode até ter que se identificar mais de uma vez.

- Segundo a Amazon, os DATACENTERS da AWS são de última geração, utilizando abordagens inovadoras de arquitetura e de engenharia e incluindo detecção de incêndio e supressão, fornecimento de energia com sistemas redundantes, controle climático e gerenciamento. **Detecção e supressão de incêndio.** Equipamentos automáticos de detecção e supressão de fogo são instalados para reduzir o risco. O sistema de detecção de incêndio utiliza sensores de detecção de fumaça em todos os ambientes do DATACENTER, espaços de infraestrutura elétrica e mecânica, *chillers* e salas de equipamento gerador. Essas áreas são protegidas por sistemas de incêndio de tubos úmidos, interbloqueados duplos ou sistemas gasosos de aspersão.
- **Energia.** Os sistemas de energia elétrica do DATACENTER são projetados para serem totalmente redundantes e passíveis de manutenção sem impacto para as operações, 24 horas por dia e sete dias por semana. As unidades de alimentação de energia ininterrupta (*no-breaks*) fornecem energia de apoio no caso de uma falha elétrica para cargas críticas e essenciais da empresa. Os DATACENTERS usam geradores para fornecer energia para toda a instalação.
- **Clima e temperatura.** O controle climático mantém uma temperatura operacional constante para servidores e outras unidades, o que impede o superaquecimento e reduz a possibilidade de interrupções do serviço. Os DATACENTERS mantêm condições de temperatura e umidade em níveis ideais. Colaboradores e sistemas monitoram e controlam a temperatura e a umidade em níveis adequados.
- **Gerenciamento.** A AWS monitora sistemas elétricos, mecânicos e de manutenção de funções vitais, para que qualquer problema seja

imediatamente identificado. A manutenção preventiva é executada para manter a operacionalidade contínua dos equipamentos.

## 12.5. Segurança da plataforma Microsoft na AWS

O artigo “Securing the Microsoft Platform on Amazon Web Services”, escrito por Tom Stickle e publicado pela AWS em agosto de 2012, traz uma série de dicas sobre como tratar da segurança Microsoft na plataforma AWS. Aqui se reforçam os aspectos de segurança para AD, WSUS e Remote Desktop Gateway.

### 12.5.1. Active Directory

O Active Directory (AD) é a peça central da rede Microsoft. Ele facilita a autenticação, a autorização e a resolução de nomes de servidores na infraestrutura de rede. Controladores de domínio do AD podem ser executados dentro da AWS e serem replicados a partir de controladores de domínio local usando a conexão VPN-VPC. Embora seja certamente possível conectar-se diretamente aos controladores de domínio corporativo através da conexão VPN-VPC, replicar esses serviços proporciona melhor desempenho e é a abordagem recomendada. A Amazon recomenda replicar os controladores de domínio em zonas de disponibilidade diferentes para obter alta disponibilidade.

A Figura 12-6 ilustra a configuração citada.

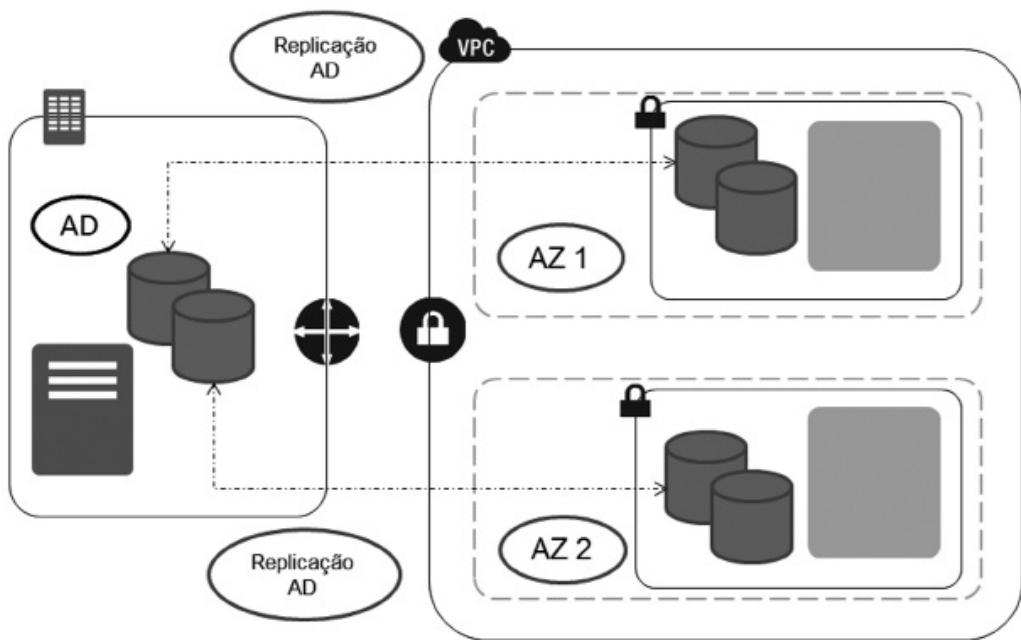


Figura 12-6 AD na AWS

### 12.5.2. WSUS

Manter patches de segurança atualizados é uma parte importante de proteção à infraestrutura. O Windows Server Update Services (WSUS) é o mecanismo geralmente usado para implantar patches nos servidores. Existe sempre a opção de apontar os servidores para um servidor WSUS que está sendo executado no DATACENTER. O fator-chave na decisão sobre isso, porém, é realmente a quantidade de infraestrutura que precisa ser atualizada e seu potencial para saturar a largura de banda da conexão VPN-VPC ou do AWS Direct Connect.

Instalar um servidor Windows Server Update Services (WSUS) na VPC pode minimizar os requisitos de largura de banda na conexão VPN-VPC, sendo possível gerenciar a implantação de atualizações para o ambiente a partir do VPC.

### 12.5.3. Remote Desktop Gateway

Em vez de administrar cada máquina, administram-se conexões via proxy usando o Remote Desktop Gateway (RD Gateway), função do Windows Server 2008 R2. Este serviço fornece uma camada adicional de segurança e oferece controles que podem atenuar o acesso relacionado à administração de ameaças aos servidores. O RD Gateway consolida os logs de auditoria para acesso administrativo e oferece mecanismos de autenticação forte baseados na autenticação mútua do SSL e na capacidade de aplicar diretivas de autorização.

O RD Gateway tem outro recurso interessante que pode ser colocado em uma *subnet* pública e, em seguida, configurado para permitir a administração remota através de uma conexão de internet, por meio de uma conexão VPN-VPC ou utilizando o AWS Direct Connect, dependendo das necessidades. As comunicações de RD Gateway são feitas via HTTPS (através da porta 443), o que permite configurar o gateway na borda da rede, se for necessário. O RDG poderá então se comunicar via proxy com os servidores de *back-end* através do protocolo RDP típico, que é transmitido por padrão à porta 3389. A **Figura 12-7** ilustra uma configuração típica que utiliza grupos de segurança para melhoria do isolamento entre as camadas da aplicação.

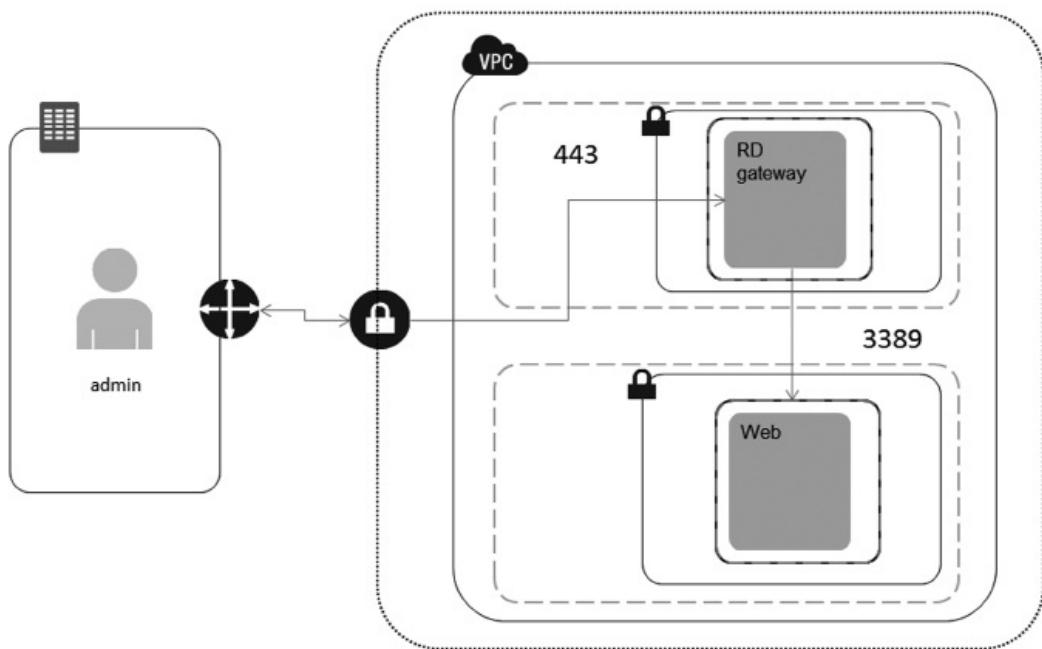


Figura 12-7 Remote Desktop Gateway

## 12.6. Referências bibliográficas

Amazon Web Services. **AWS Security: Best Practices**. Jan. 2011.

Amazon Web Services. **Overview of Security Processes**. May 2011.

Amazon Web Services. **Risk and Compliance**. Dec. 2011.

Amazon Web Services. **Securing the Microsoft Platform on Amazon Web Services**. Aug. 2012.

Diógenes, Yuri; Mauder, Daniel. **Certificação Security +: Da Prática Para o Exame SY0-31**.

<http://aws.amazon.com/pt/security/>

IBM. **Segurança e ampla disponibilidade em ambientes de computação em nuvem**. Jun. 2012.

# 13. Continuidade

## 13.1. Introdução

A ideia da continuidade do negócio é que uma organização tenha a capacidade de se recuperar ou de manter suas atividades no caso de uma interrupção das operações normais do seu negócio. Existe uma relação direta entre continuidade do negócio e recuperação de desastres.

A norma ISO que trata da continuidade do ponto de vista do negócio é a 22301:2012. Ela substitui a norma BS 25999, a primeira norma mundial para gestão de continuidade dos negócios.

Os serviços para recuperação de desastres (*Disaster Recovery – DR*) no ambiente de cloud computing são muito importantes, sendo tratados como aspectos fundamentais da segurança. A AWS oferece uma série de serviços para a plena recuperação de desastres. Este capítulo tem parte baseada no artigo “Usando o Amazon Web Services para recuperação de desastres”, publicado pela Amazon em outubro de 2011.

Neste capítulo serão revistos os conceitos relacionados à recuperação de desastres e aos serviços fornecidos pela AWS.

## 13.2. Fundamentos

### 13.2.1. Continuidade do negócio e recuperação de desastres

Alta disponibilidade e confiabilidade, backup e archive e replicação podem ser consideradas técnicas de recuperação de desastres. A **Figura 13-1** relaciona continuidade do negócio com recuperação de desastres.

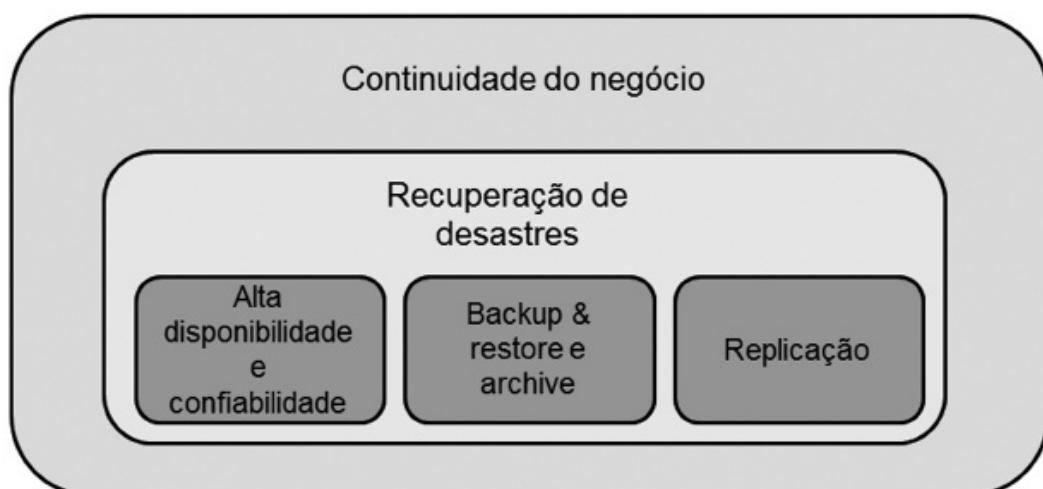


Figura 13-1 Continuidade do negócio

Desastres não se resumem somente a fogo, inundação e outras causas de dano à propriedade; eles também podem resultar de problemas corriqueiros como greves ou mau funcionamento de hardware ou software. E ainda que a restauração do processamento computacional seja um passo importante do processo de recuperação, outros problemas igualmente importantes frequentemente precisam ser resolvidos.

### 13.2.2. RTO e RPO

O acordo de nível de serviço (SLA) para recuperação de desastres normalmente é baseado em duas métricas: objetivo de tempo de recuperação (*Recover Time Objective – RTO*) e objetivo de ponto de recuperação (*Recover Point Objective – RPO*). Os acordos, no caso de recuperação de desastres, devem ser baseados nestas duas métricas, que, por sua vez, dependem do processo, da tecnologia e do pessoal utilizados.

A **Figura 13-2** mostra os indicadores RTO e RPO em termos da ocorrência de um *downtime*.

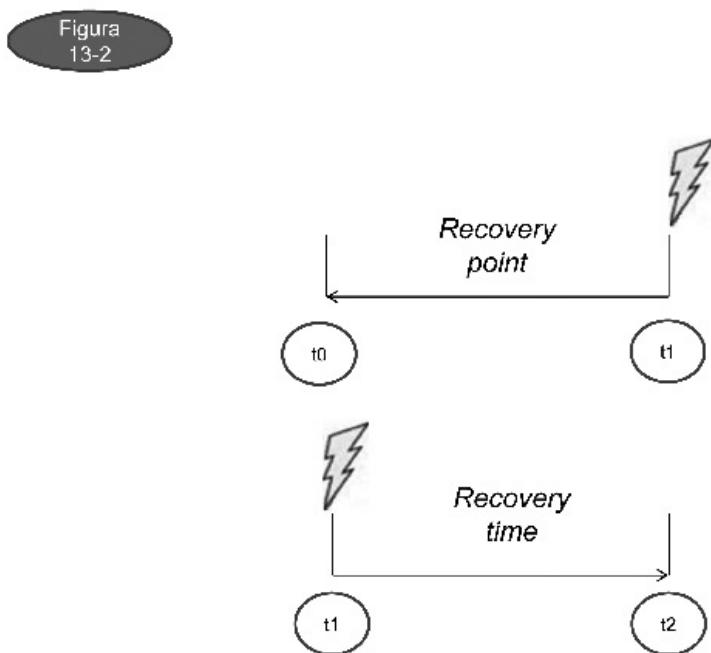


Figura 13-2 RTO e RPO

- **RTO** é o período do tempo e o nível de serviço para um processo de negócios ser restaurado após um desastre (ou interrupção) e para evitar consequências inaceitáveis associadas a uma quebra na continuidade de negócios. RTO pode ser visto como o processo de olhar para frente que estima o tempo que levaria para os negócios se tornarem totalmente operacionais novamente, uma vez que se tenha

uma boa cópia dos dados. Essa é uma medição para tempo de inatividade. Medidas e tecnologias para obtenção de RTO são:

- **Semanas:** restore da fita.
- **Dias:** restore do disco.
- **Horas/Minutos:** migração manual.
- **Segundos:** cluster global.

Por exemplo, se ocorresse um desastre às 12h (meio-dia) e o RTO fosse de oito horas, o processo de DR garantiria a recuperação em nível de serviço aceitável por volta das 20h.

**RPO** descreve a quantidade aceitável de perda de dados medida em tempo. Por exemplo, se o RPO foi de uma hora depois que o sistema foi recuperado, ele conteria todos os dados até um momento determinado que não seria antes das 11h, porque o desastre ocorreu ao meio-dia. RPO pode ser visto como o processo de olhar para trás que mede o quanto será necessário retroceder no tempo para obter uma boa cópia de reinicialização dos dados se algo de ruim acontecer a eles. Por exemplo, suponha-se que hoje seja sexta-feira, 28 de abril, por volta das 9 horas da manhã, e que o último backup dos dados de negócios foi às 9 horas da manhã do dia anterior. Se algo acontecesse aos dados agora, como o sistema parar e os dados se perderem, seria necessário retroceder 24 horas no tempo a fim de obter uma boa cópia reinicializável dos dados. Portanto, nesse caso, o RPO é de 24 horas. Medidas e tecnologias para obtenção de RPO:

- **Semanas:** backup em fita.
- **Dias:** replicação periódica.
- **Horas/minutos:** replicação assíncrona.
- **Segundos:** replicação síncrona.

Uma empresa decide normalmente por um RTO e um RPO aceitáveis com base no impacto financeiro para os negócios, quando os sistemas estão disponíveis. O impacto financeiro é tipicamente avaliado por muitos fatores, como a perda de negócios e danos à sua reputação devido ao tempo de inatividade e a falta de disponibilidade dos sistemas de informação.

Deve-se ressaltar que normalmente as visões de TI e do pessoal de negócios é distinta. O pessoal de negócios quer quase sempre o menor RTO e o menor RPO, e TI precisa explicar que quanto menor estes indicadores maior o preço da solução. Continuidade não é demanda de TI, e sim do negócio.

### 13.2.3. Hot site, warm site e cold site

Outro aspecto importante na recuperação de desastres são os conceitos

de hot site, warm site e cold site.

O hot site é outro site pronto para entrar em operação em caso de desastre. Ele possui praticamente toda a infraestrutura do site principal, com eliminação de algumas redundâncias, e está sempre pronto para entrar em operação.

O warm site é semelhante ao hot site, mas leva mais tempo para que o sistema entre novamente em operação. Diversos sistemas são configurados manualmente neste caso e configurações de software são realizadas por demanda. O warm site é mais barato do que o hot site.

O cold site é um site utilizado na recuperação de um desastre, mas com uma infraestrutura mínima que deverá ser ativada se necessário. Ou seja, o cold site não está pronto para entrar em operação de imediato. Boa parte do hardware e software utilizado precisa ser instalado, e o processo de pôr o sistema em produção pode levar dias. O cold site é mais barato do que o warm site.

### 13.2.4. Alta disponibilidade e confiabilidade

Alta disponibilidade e confiabilidade são conceitos fundamentais e muitas vezes pouco compreendidos.

**Disponibilidade** é a capacidade de serviços, componentes e itens de configuração estarem disponíveis para os usuários conforme estabelecido no SLA.

$$A = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

Onde A é o grau de disponibilidade expresso em porcentagem, MTBF é o “mean time between failures” e o MTTR é o “maximum time to repair”. Se um sistema tem MTBF de 100.000 horas (mais que onze anos) e o MTTR é de uma hora, a disponibilidade (A) é de 99,9999%. Ou seja, em onze anos haverá seis minutos de *downtime*.

A Tabela 13-1 ilustra valores típicos de disponibilidade.

Tabela 13-1 Disponibilidade e *downtime* por ano

Disponibilidade	Downtime por ano
99,9999 (seis noves)	32s
99,999	5m,15s
99,99	52m, 36s
99,9	8h, 46m

**Confiabilidade (reliability)** é uma medida que demonstra a capacidade de trabalhar sem interrupções, de acordo com o SLA acordado. Associa-se este número ao tempo médio entre falhas (MTBF).

MTBF (horas) = (tempo em que o serviço deve ficar disponível em horas — tempo total em que o serviço ficou indisponível em horas) / número de interrupções.

### 13.2.5. Backup e restore

O backup é uma cópia dos dados de produção, criada e retida para o propósito de recuperar dados deletados ou corrompidos.

O backup necessita ser realizado por várias razões, dentre elas:

- Requisitos de negócio.
- Requisitos legais.
- Proteção contra falhas de hardware.
- Proteção contra falha da aplicação.
- Proteção contra erro do usuário.
- Recuperação de desastres.
- Atingimento de níveis de serviço específicos.

A quantidade de perda de dados e o *downtime* que um negócio pode suportar em termos de RTO e RPO definem a estratégia a ser utilizada pelo backup e o restore. A natureza dos dados tem impacto nos tempos de backup e restore.

O backup pode ser classificado da seguinte forma:

- **Backup de imagem ou backup no nível de bloco:** muito rápido. Dificuldade de recuperar arquivos únicos sem acesso aos metadados (dados sobre os dados).
- **Backup no nível de arquivo:** usa o sistema operacional para fazer o backup dos arquivos. Este backup é mais longo. Fácil de recuperar arquivos únicos.
- **Backup no nível da aplicação:** usa APIs para a realização dos backups e é associado à aplicação, tendo prós e contras.

Os tipos de backups utilizados são o **completo** (copia todos os arquivos de um disco para uma unidade de armazenamento, normalmente uma fita), **incremental** (neste caso faz-se o backup só dos dados que foram alterados desde o último backup completo) e **diferencial** (faz backup de todos os dados alterados desde o último backup completo).

O backup normalmente utiliza uma estratégia de organização. A mais comum é o backup avô, pai e filho, onde normalmente o backup avô é anual, o backup pai é mensal e o backup filho é semanal. Outra estratégia é fazer um backup de todas as informações disponíveis no sistema por tempo indeterminado. Evidentemente, a quantidade de mídias neste caso é um fator de grande impacto nos custos da solução.

O backup convencional normalmente é realizado por meio de fitas do tipo LTO (*Line Tape Open*) e/ou discos de baixo custo. Os discos normalmente são mais rápidos e a unidade de fita é mais robusta e com menor preço.

### 13.2.6. Archive

Os archives são cópias principais das informações. As informações arquivadas são valiosas e conservadas para referências futuras e devem ter garantias de autenticidade. Elas tratam informações normalmente em sua forma final e estão sujeitas a pouca ou nenhuma modificação. Como passam a ser a única cópia das informações, devem ser mantidas por períodos de longo prazo (meses, anos ou décadas).

### 13.2.7. Replicação

Replicação é a tecnologia normalmente utilizada para recuperação de desastres de ambientes de TI, mas também pode ser utilizada para replicação de dados dentro do mesmo ambiente físico.

O resultado final da replicação é obter dois conjuntos de dados consistentes e iguais em dois lugares diferentes.

Ao replicar dados para um local remoto, existem alguns fatores a serem considerados.

- **Distância entre os locais:** distâncias maiores normalmente estão sujeitas a mais latência e/ou variação.
- **Largura de banda disponível:** quão amplas e variáveis são as interligações?
- **Taxa de dados exigida pelo aplicativo:** a taxa de dados deve ser menor que a largura de banda disponível.

Existem duas abordagens principais utilizadas na replicação de dados: replicação síncrona e replicação assíncrona.

- Na replicação síncrona os dados são atualizados de forma atômica em vários locais. Isso depende da disponibilidade e do desempenho da rede.
- Na replicação assíncrona os dados não são atualizados de forma atômica em vários locais. Os dados são transferidos conforme o desempenho e a disponibilidade da rede permitem, e o aplicativo

continua a gravar os dados que ainda não podem ser totalmente replicados.

Muitos sistemas de banco de dados oferecem suporte à replicação de dados assíncrona. A réplica do banco de dados pode ser localizada de forma remota e não necessita estar completamente em sincronia com o servidor de banco de dados primário. Isso é aceitável em muitos cenários, por exemplo, como uma fonte de backup ou em casos de utilização de relatórios de leitura.

Na AWS, as zonas de disponibilidade dentro de uma região são bem conectadas, mesmo que fisicamente separadas, o que permite utilizar a replicação síncrona entre instâncias de banco de dados. Por exemplo, quando implantado no modo *Multi-AZ*, o serviço de banco de dados relacional RDS da Amazon usa a replicação síncrona para duplicar dados em uma segunda zona de disponibilidade. Isso garante que os dados não sejam perdidos se a zona primária de disponibilidade se tornar indisponível.

## 13.3. Práticas tradicionais *versus* práticas com a AWS

As práticas de recuperação de desastres podem ser classificadas em práticas tradicionais e práticas baseadas na AWS.

### 13.3.1. Práticas tradicionais

Uma abordagem tradicional de recuperação de desastres envolve diferentes níveis de duplicação fora do local de dados e da infraestrutura. Serviços essenciais aos negócios são criados e mantidos nesta infraestrutura e são testados em intervalos regulares. O ambiente de recuperação de desastres e o ambiente principal devem estar a uma distância física significativa para garantir que o ambiente de recuperação de desastres esteja isolado das falhas que poderiam afetar o site de origem.

A infraestrutura necessária para suportar o ambiente duplicado inclui, mas não se limita a:

- Instalações para abrigar a infraestrutura, incluindo energia e refrigeração.
- Segurança para garantir a proteção física dos ativos.
- Capacidade adequada para dimensionar o ambiente.
- Suporte para reparar, substituir e atualizar a infraestrutura.
- Acordos contratuais com um provedor de serviços (ISP) para fornecer conectividade com a internet.
- Infraestrutura de rede, como *firewalls*, roteadores, switches e平衡adores de carga.
- Suficiente capacidade de servidor para executar todos os serviços de

missão crítica.

Dependendo da criticidade dos serviços, o ambiente duplicado pode ser configurado de forma tolerante a falhas. Isso normalmente envolve a duplicação de toda a infraestrutura listada anteriormente.

### **13.3.2. Práticas com AWS**

Antes de discutir as diferentes práticas de DR utilizando a arquitetura AWS, é importante analisar seus serviços e recursos mais relevantes para a recuperação de desastres.

Na fase de preparação da DR, é essencial considerar a utilização de serviços e recursos que oferecem suporte à migração de dados e ao armazenamento durável, pois estes permitem restaurar dados armazenados na AWS quando ocorrer um desastre.

#### **13.3.2.1. Uso de múltiplas zonas de disponibilidade**

Esta abordagem permite ter os serviços de TI executando em várias zonas de disponibilidade na AWS. A falha em uma zona específica irá redirecionar o tráfego para uma zona diferente que é estável. Esta é uma solução de custo eficaz (em comparação com a segunda abordagem de distribuição de serviços de negócios em várias “regiões de disponibilidade”). No entanto, esta abordagem pode não ser suficiente quando a região cai.

#### **13.3.2.2. Uso de múltiplas regiões**

A segunda abordagem é executar o aplicativo entre várias regiões de disponibilidade. Neste caso, o serviço é hospedado em várias regiões da AWS. É possível ter tráfego geodistribuído e alta disponibilidade através dos continentes. Esta configuração é recomendada para empresas que precisam de um alto nível de escalabilidade, balanceamento de carga e requisitos de acesso de usuário ao redor do mundo. No caso de falha em uma região, o tráfego pode ser redirecionado a outras regiões estáveis.

As regiões AWS são independentes entre si, mas não há diferenças na forma como são acessadas e utilizadas. Isso permite que os clientes criem processos de recuperação de desastres que se estendem por distâncias continentais, sem os desafios ou os custos que isso normalmente acarretaria. Os clientes podem fazer backup de dados e sistemas para duas ou mais regiões AWS, permitindo restauração de serviço mesmo em face de desastres em grande escala. Os clientes AWS podem usar diferentes regiões AWS para servir clientes finais ao redor do mundo com relativamente baixa complexidade para seus processos operacionais.

A **Figura 13-3** ilustra duas situações possíveis de DR com a AWS.

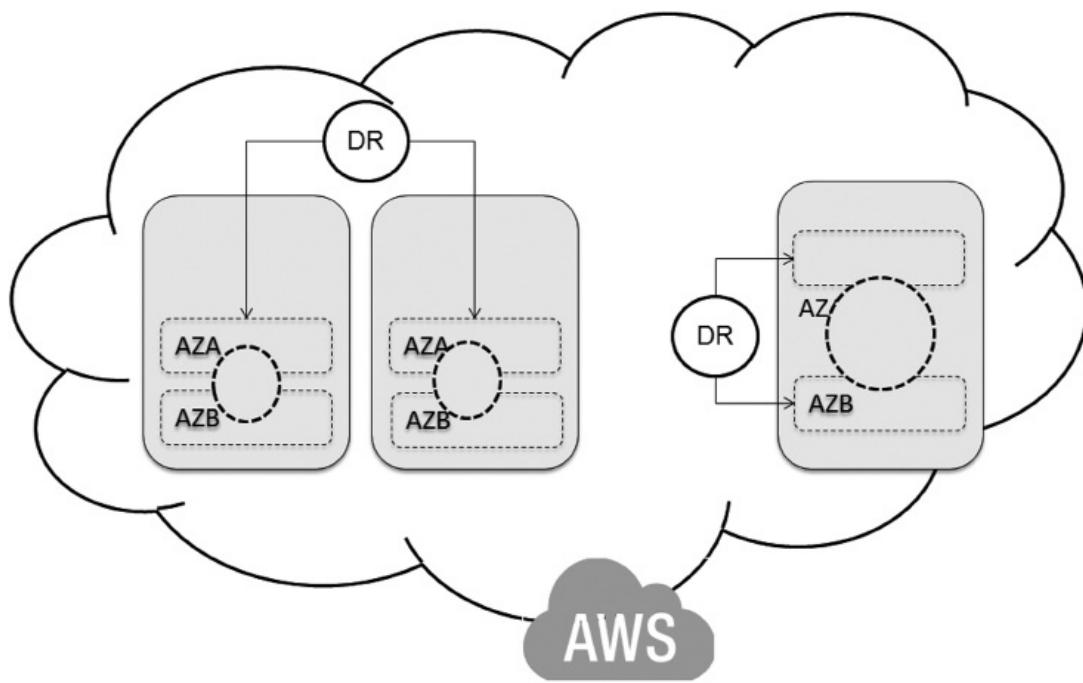


Figura 13-3 DR na AWS

### 13.3.2.3. Uso de recursos essenciais

Ao reagir a um desastre, é essencial delegar rapidamente recursos computacionais para executar o sistema na AWS ou orquestrar o *failover* para recursos já em execução na AWS. As peças de infraestrutura essenciais incluem o DNS, recursos de rede e várias funcionalidades do EC2 descritas a seguir.

#### Computação

O EC2 é um web service que fornece uma capacidade de computação redimensionável na nuvem. Em questão de minutos é possível criar instâncias EC2. No contexto da DR, a capacidade de rapidamente criar máquinas virtuais é essencial. As AMIs são pré-configuradas com sistemas operacionais e algumas delas também podem incluir pilhas de aplicativo. AMIs podem ser configuradas de acordo com a conveniência. Tudo isto foi visto no capítulo 5.

No contexto da DR, é altamente recomendável que uma organização tenha suas próprias AMIs configuradas e identificadas para que possam ser inicializadas como parte do processo de recuperação. Tais AMIs devem ser pré-configuradas com o sistema operacional escolhido, além de conter peças adequadas de pilha de aplicativo.

As instâncias reservadas do EC2, que muitas vezes são usadas para receber um desconto significativo sobre o custo da execução de uma instância EC2, têm outra vantagem particularmente relevante para a DR: as instâncias reservadas ajudam a garantir que a capacidade necessária esteja disponível quando necessário. Como se sabe, a capacidade da nuvem AWS é finita.

O ELB distribui automaticamente o tráfego de entrada dos aplicativos em várias instâncias do EC2. Ele permite atingir uma maior tolerância a falhas nos aplicativos, fornecendo a capacidade de equilíbrio de carga necessária em resposta ao tráfego de entrada dos aplicativos. Assim como é possível pré-alocar endereços IPs elásticos, pode-se pré-alocar o ELB com um nome DNS já conhecido, o que simplifica a execução de plano de DR.

## Armazenamento

O S3 fornece uma infraestrutura durável projetada para armazenamento de dados de missão crítica. Os objetos são armazenados redundantemente em vários dispositivos dentro de uma região.

O EBS pode criar *snapshots* de volumes de dados em um determinado momento. Esses *snapshots* utilizam o S3 como local de armazenagem e podem ser usados como ponto inicial para criação de novos volumes EBS e para proteger dados para durabilidade a longo prazo. Após um volume ser criado, ele pode ser ligado a qualquer instância EC2. Os volumes EBS oferecem armazenamentos fora da instância, que persistem independentemente da vida da instância.

O EBS Snapshot Copy permite copiar *snapshots* de discos EBS entre regiões diferentes na nuvem AWS. Isto facilita o uso de múltiplas regiões AWS para fins de *disaster recovery*.

O armazenamento foi tratado detalhadamente no capítulo 6.

## Rede

Quando se lida com um desastre, é muito provável que seja necessário modificar as configurações de rede, pois está sendo realizado um *failover* para outro local.

O Route 53 é um DNS altamente disponível e escalável, conforme visto anteriormente. Ele foi projetado para dar a desenvolvedores e empresas uma maneira extremamente econômica e confiável de direcionar os usuários finais para aplicativos da internet.

EIP são endereços IP estáticos projetados para computação em nuvem dinâmica. Ao contrário dos tradicionais endereços IP estáticos, os endereços EIP (*Elastic IP*) permitem filtrar a instância ou as falhas da zona de disponibilidade por meio de remapeamento programado dos endereços IP públicos para qualquer instância alocada dentro da conta. Para recuperação de desastres, pré-alocam-se também alguns endereços IP elásticos para os sistemas mais importantes, para que os endereços IP já sejam conhecidos antes que desastres aconteçam. Isso pode simplificar a execução do plano de DR.

A VPC permite aproveitar uma seção privada e isolada da nuvem da AWS, onde podem ser executados recursos AWS em uma rede virtual definida de forma específica. Pode-se ter o controle total sobre o ambiente de rede virtual, incluindo a seleção do próprio intervalo de endereços IP, criação de *subnets* e configuração de tabelas de roteamento e gateways de rede. Isso permitirá criar uma conexão VPN entre o DATACENTER corporativo e a VPC e aproveitar a nuvem AWS como uma extensão do DATACENTER corporativo.

No contexto de DR, pode-se usar a VPC para estender a topologia de rede interna existente para a nuvem. Isso pode ser especialmente apropriado ao recuperar aplicativos corporativos que estejam essencialmente na rede interna.

O capítulo 7 trata da rede VPC.

## Bancos de dados

O serviço de banco de dados relacional RDS facilita a configuração, a operação e o dimensionamento do banco de dados relacional na nuvem. Pode-se utilizar o RDS na fase de preparação para recuperação de desastres para armazenar dados críticos em um banco de dados já em execução e/ou em fase de recuperação.

O capítulo 8 tratou do web service RDS.

### 13.3.2.4. Uso de backup e restore

O artigo “Backup and Recovery Approaches Using Amazon Web Services”, de setembro de 2012, publicado pela Amazon Web Services, esclarece as vantagens operacionais de realizar o backup e o restore na AWS.

Uma instância EC2 do tipo servidor web iniciada na AWS é baseada em uma imagem do tipo AMI e pode ser conectada a volumes existentes de armazenamento do tipo EBS. A AMI normalmente contém o sistema operacional e pode conter algum aplicativo específico. Considerando que as imagens AMIs, ao serem registradas, estão disponíveis para cada conta como *snapshots* EBS (e podem inclusive ser compartilhadas entre contas AWS), pode-se facilmente compartilhar uma AMI entre contas, iniciar uma nova instância baseada na AMI compartilhada e criar uma nova AMI de uma instância que está rodando, o que facilita sobremaneira o backup e o restore da instância EC2. Por sua vez, a AWS mantém no S3 um *bucket* com os diferentes arquivos de configuração. Observe que, no caso da AWS, o backup é feito só dos arquivos de configuração armazenados no S3. O backup na AWS é assim dramaticamente reduzido em termos de tempo (janela de backup) e permite um controle efetivo do ambiente.

No caso de servidores de banco de dados e servidores de arquivos, o procedimento na AWS difere, pois esses tipos de servidores contêm grandes

quantidades de dados que devem ser protegidos. Bancos de dados podem ser replicados assincronamente para outra instância de banco de dados usando um volume EBS. Detalhes sobre o backup do banco de dados RDS foram vistos no capítulo 8.

## 13.4. Cenários AWS

Diversos cenários são possíveis para recuperação de desastres na AWS. O que vai determinar a escolha do cenário é a necessidade do negócio. Possíveis cenários são descritos pela Amazon e replicados a seguir.

A **Figura 13-4** ilustra possíveis cenários e a relação com os parâmetros RPO e RTO. Esses cenários são explicados a seguir.

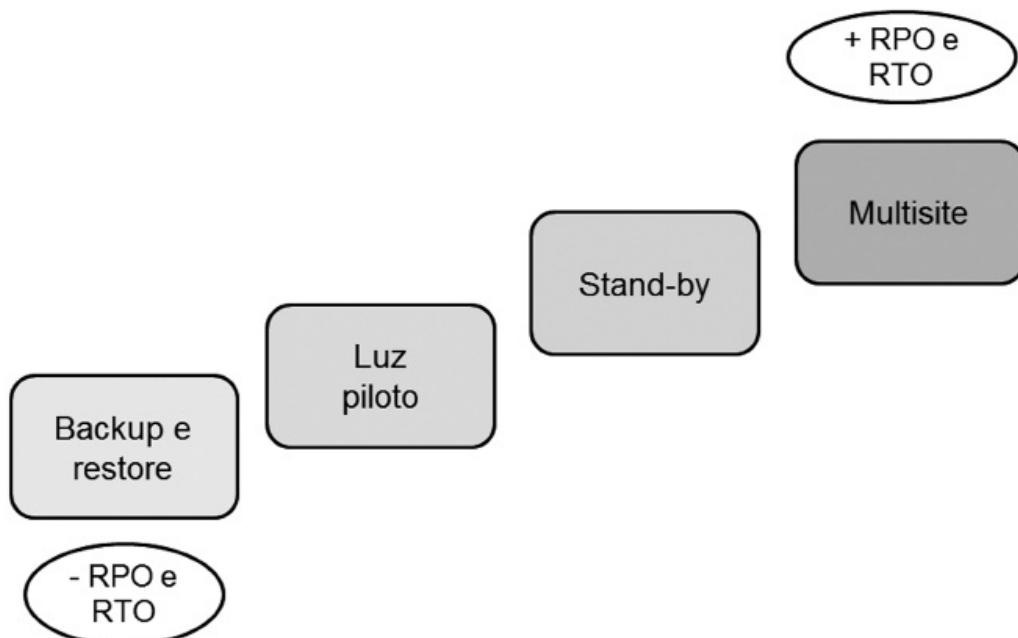


Figura 13-4 Opções de DR na AWS

### 13.4.1. Backup e restore

Em ambientes mais tradicionais, os dados são armazenados em fita e enviados a um local externo regularmente. O tempo de recuperação é mais longo usando esse método. Esta opção com a AWS otimiza esse cenário.

#### 13.4.1.1. Fase de preparação

O backup normalmente envolve o S3. O S3 é um destino interessante para os dados, pois é projetado para fornecer durabilidade de 99,999999999% (onze noves) de objetos ao longo de um determinado ano. A transferência de dados envolvendo o S3 normalmente é feita através da rede e, portanto, está acessível a partir de qualquer local.

É importante ressaltar que existem muitas soluções de backup comerciais e de código aberto que utilizam o S3.

A Figura 13-5 ilustra as opções para o backup e restore no S3.

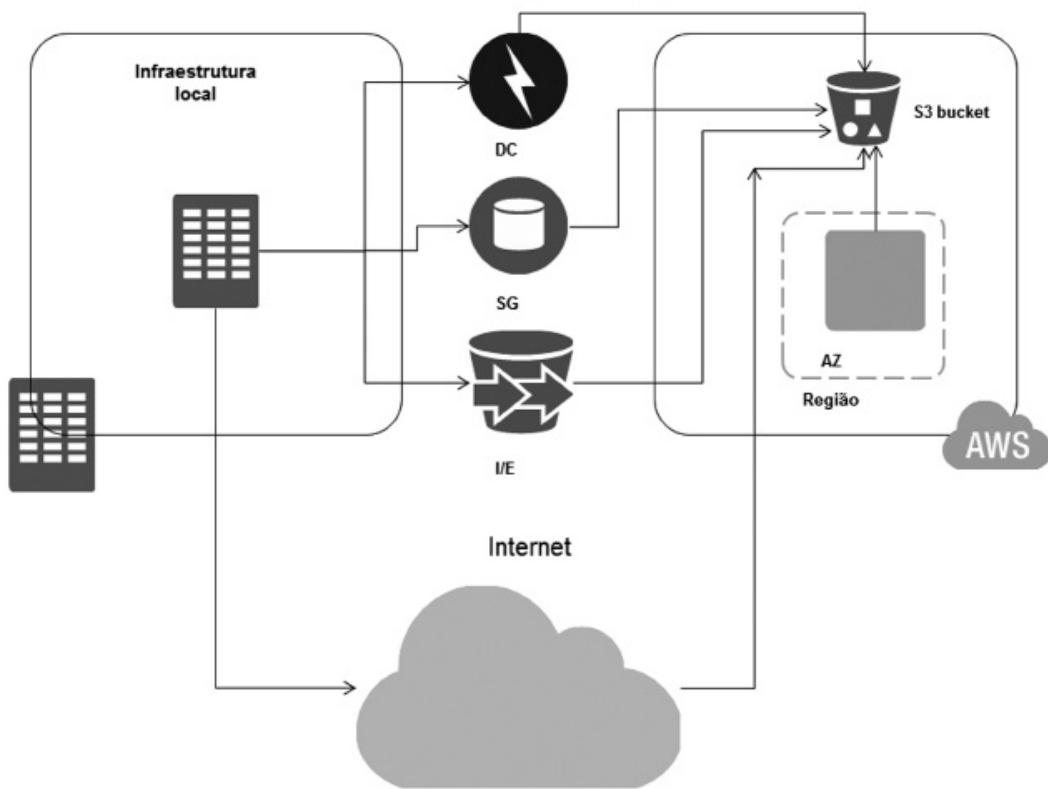


Figura 13-5 Alternativas de backup e restore com AWS

Para sistemas em execução na AWS, os clientes também podem fazer backup no S3. Os *snapshots* de volumes do Elastic Block Store (EBS) e os backups do RDS são armazenados obrigatoriamente no S3. Como alternativa, pode-se copiar arquivos diretamente no S3, ou opta-se por criar arquivos de backup e copiá-los no S3. Existem muitas soluções de backup que armazenam dados de backup no S3, e essas também podem ser usadas a partir do EC2.

As opções Direct Connect, Import/Export e Storage Gateway podem fazer parte de uma solução de backup e serão vistas adiante neste capítulo. Também é possível utilizar diretamente a internet como meio para realização do backup.

#### 13.4.1.2. Fase de recuperação

O backup dos dados é apenas metade da história. A recuperação de dados em um cenário de desastre deve ser testada e alcançada de forma rápida e confiável. Os clientes devem garantir que seus sistemas estejam configurados para retenção adequada de dados, segurança e que tenham sido testados para seus processos de recuperação de dados.

As principais etapas para backup e restauração:

- Selecione uma ferramenta apropriada ou método para fazer backup dos dados na AWS.
- Assegure-se de ter uma política de retenção adequada para esses

dados.

- Certifique-se de que as medidas de segurança adequadas estejam em vigor para esses dados, incluindo as políticas de acesso e criptografia.
- Teste regularmente a recuperação desses dados e a restauração do sistema.

A **Figura 13-6** ilustra um cenário típico de restauração do backup baseado no S3.

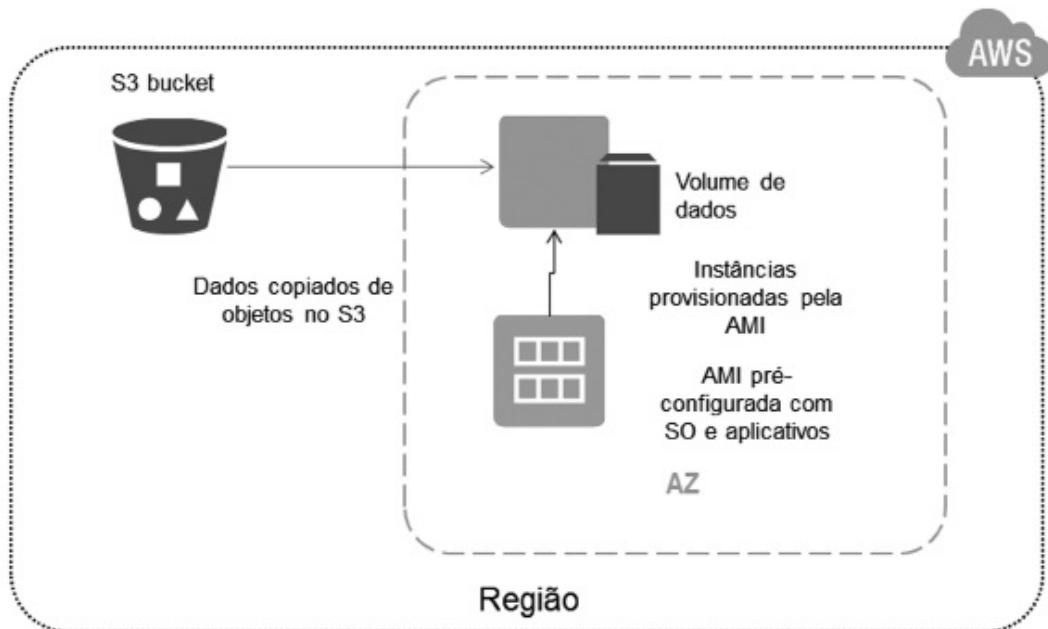


Figura 13-6 Restore de um sistema de backup

### 13.4.2. Luz piloto

A ideia da luz piloto é uma analogia que vem do aquecedor a gás. Em um aquecedor a gás, uma pequena chama que está sempre ligada conduz à ignição de toda a fornalha para aquecer uma casa, conforme necessário.

O cenário de luz piloto é semelhante a um cenário de backup e restauração. No entanto, deve-se assegurar que os elementos fundamentais do sistema já estejam configurados e em execução na AWS (a luz piloto). No momento de executar a recuperação, o núcleo do ambiente de produção seria rapidamente provisionado; com isto, é possível obter menores tempos de recuperação. A luz piloto é uma opção superior em termos de disponibilidade quando comparada a uma solução de backup e restore convencional. O cenário de luz piloto permitirá obter um tempo de recuperação menor do que o cenário “backup e restauração”, pois as peças principais do sistema já estão em execução e são constantemente atualizadas.

Elementos da infraestrutura para a luz piloto propriamente dita

normalmente incluem os servidores de banco de dados, que replicariam os dados para o EC2. Dependendo do sistema, pode haver outros dados críticos fora do banco de dados que precisam ser replicados na AWS. Este é o núcleo fundamental do sistema (a luz piloto) em torno do qual todas as outras peças da infraestrutura podem rapidamente ser configuradas (a fornalha) para restaurar o sistema completo.

Para provisionar o restante da infraestrutura para restaurar serviços críticos de negócios, deve-se ter alguns servidores pré-configurados com Amazon Machine Images (AMIs), prontos para serem iniciados a qualquer momento. Ao iniciar a recuperação, as instâncias dessas AMIs são lançadas rapidamente e encontram sua função dentro da implantação em torno da luz piloto. Do ponto de vista da rede, é possível usar os endereços Elastic IP (que podem ser pré-alocados na fase de preparação para recuperação de desastres) e associá-los às instâncias, ou usar o ELB para distribuir o tráfego para várias instâncias. Em seguida, os registros DNS seriam atualizados para apontarem para a instância do EC2 ou para o ELB usando um CNAME.

Para sistemas menos críticos, pode-se garantir que existam pacotes de instalação e informações de configuração disponíveis sob a forma de um *snapshot* EBS. Isso acelerará a configuração do servidor de aplicativo, pois criam-se rapidamente vários volumes em várias zonas de disponibilidade para anexar a instâncias EC2. Pode-se, em seguida, instalar e configurar adequadamente o ambiente.

Existem ainda algumas tarefas de instalação e configuração para que se possa recuperar totalmente os aplicativos. A AWS permite automatizar o provisionamento e a configuração dos recursos de infraestrutura.

#### **13.4.2.1. Fase de preparação**

A **Figura 13-7** a seguir mostra a fase de preparação, na qual os dados precisam ser regularmente replicados para a luz piloto, o pequeno núcleo em torno do qual o ambiente completo será iniciado na fase de recuperação. Os dados atualizados com menos frequência, como sistemas operacionais e aplicativos, podem ser periodicamente atualizados e armazenados como AMIs.

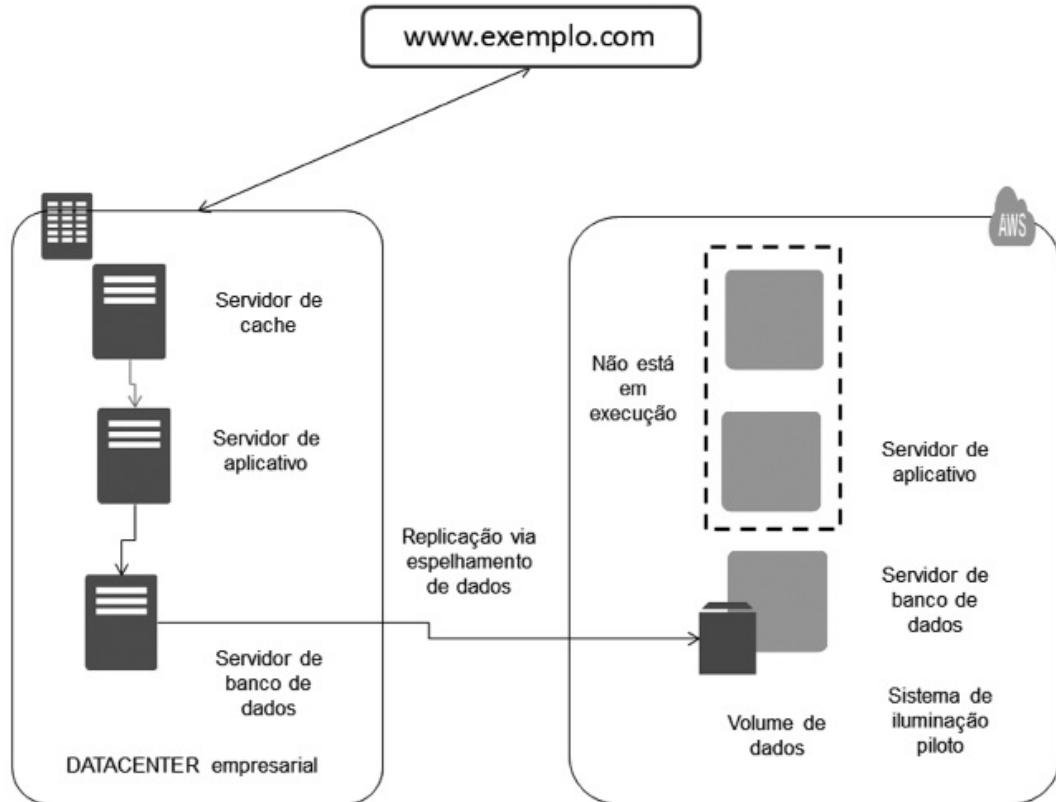


Figura 13-7 Fase de preparação do cenário de luz piloto

Pontos-chave para preparação do cenário de luz piloto:

- Configure instâncias EC2 para replicar ou espelhar dados.
- Verifique os pacotes de software disponíveis na AWS.
- Crie e mantenha Amazon Machine Images (AMI) de servidores-chave para os quais a recuperação rápida seja necessária.
- Execute teste e aplique atualizações de software e de configuração a esses servidores regularmente.
- Considere a possibilidade de automatizar o provisionamento de recursos AWS.

#### 13.4.2.2. Fase de recuperação

Para recuperar o restante do ambiente ao redor da luz piloto, os sistemas devem ser iniciados a partir das AMIs em tipos apropriados de instâncias. Para servidores de dados dinâmicos, é possível redimensioná-los para lidar com volumes de produção, conforme necessário, ou adicionar capacidade de acordo com a demanda.

O dimensionamento horizontal, se possível de ser utilizado, é muitas vezes a forma mais econômica e a abordagem mais simples para adicionar capacidade a um sistema; no entanto, também é possível escolher tipos maiores de instância EC2 e, portanto, fazer o dimensionamento de forma vertical.

Uma vez o ambiente recuperado, deve-se garantir que a redundância seja restaurada o mais rápido possível. Mesmo sendo improvável ocorrer uma falha do ambiente de DR seguida de uma falha do ambiente de produção, é importante estar ciente desse risco. Continue a fazer backups regulares do seu sistema e considere redundância adicional na camada de dados.

A **Figura 13-8** ilustra o cenário de recuperação de luz piloto.

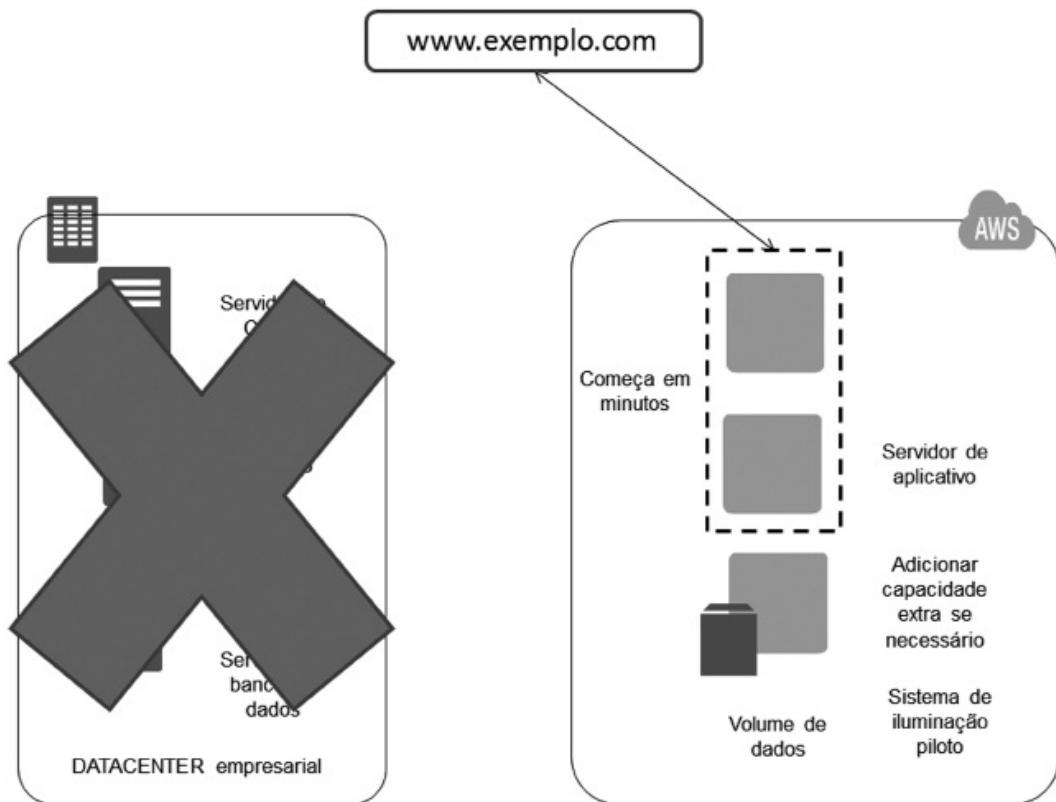


Figura 13-8 Fase de recuperação do cenário de luz piloto

## Pontos-chave para recuperação:

- Inicie instâncias EC2 a partir de suas AMIs personalizadas.
  - Redimensione e/ou dimensione quaisquer banco de dados/instâncias de dados armazenados sempre que necessário.
  - Altere o DNS para apontar para os servidores EC2.
  - Instale e configure todos os sistemas não baseados em AMI, idealmente no modo automatizado.

### **13.4.3. Stand-by**

Uma solução *stand-by* em relação à luz piloto diminui ainda mais o tempo de recuperação, pois, neste caso, alguns serviços estão sempre em execução. Através da identificação de sistemas críticos para os negócios, seria possível duplicar totalmente esses sistemas na AWS e tê-los sempre em execução.

As instâncias podem estar sendo executadas em conjuntos de tamanho mínimo do EC2. Esta solução não é dimensionada para suportar uma carga de produção completa, mas é totalmente funcional. Ela pode ser usada para trabalhos que não sejam de produção, como testes, controle de qualidade, uso interno etc.

No caso de um desastre, o sistema é dimensionado rapidamente para lidar com a carga de produção. Na AWS, isso pode ser feito adicionando mais instâncias ao平衡ador de carga e redimensionando os servidores de pequena capacidade para serem executados em instâncias EC2 maiores. O dimensionamento horizontal, quando possível de ser utilizado, é muitas vezes preferido ao dimensionamento vertical.

### 13.4.3.1. Fase de preparação

O diagrama ilustrado na **Figura 13-9** mostra a fase de preparação para uma solução *stand-by*, em que uma solução no local e uma solução AWS são executadas lado a lado.

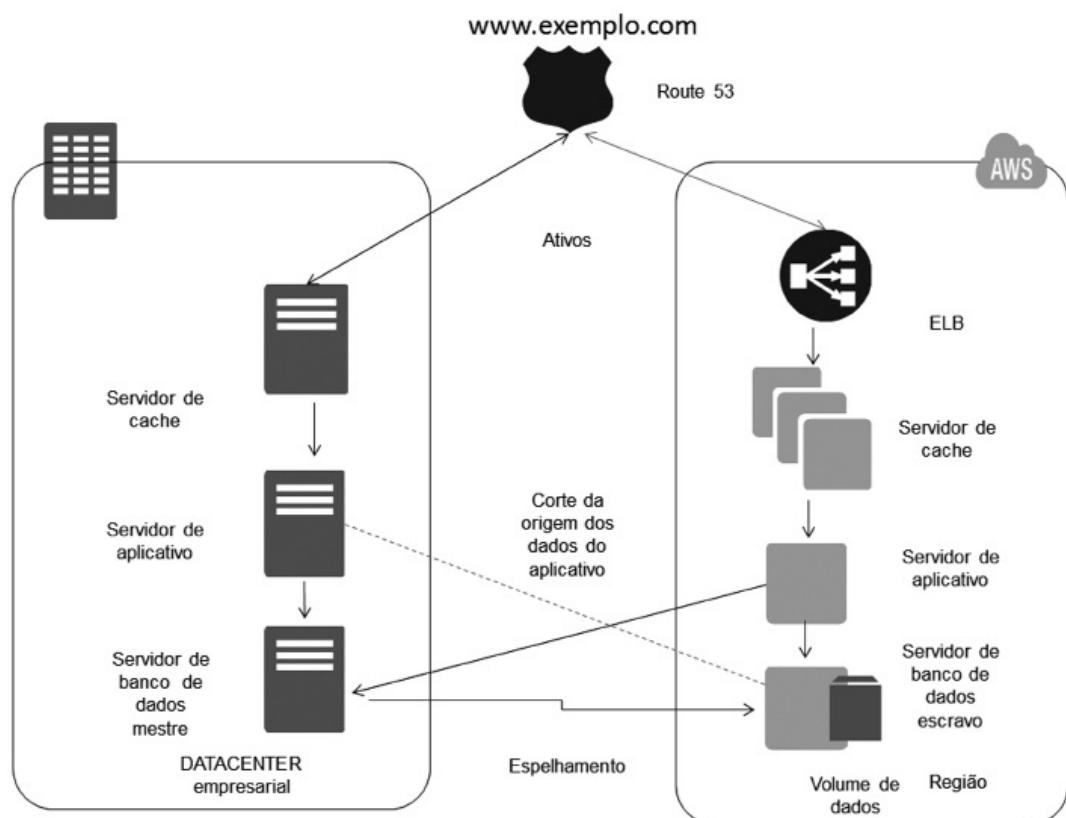


Figura 13-9 Fase de preparação do cenário stand-by

Pontos-chave para preparação:

- Configure instâncias EC2 para replicar ou espelhar dados.
- Crie e mantenha Amazon Machine Images (AMI).
- Execute o aplicativo usando um espaço mínimo de instâncias EC2 ou de infraestrutura AWS.

- Instale patches e atualize arquivos de configuração e de software em conformidade com seu ambiente ao vivo.

### 13.4.3.2. Fase de recuperação

Em caso de falha do sistema de produção, o ambiente *stand-by* será dimensionado para carga de produção e os registros DNS serão alterados para rotear todo o tráfego para a AWS.

A **Figura 13-10** ilustra a recuperação na opção *stand-by*.

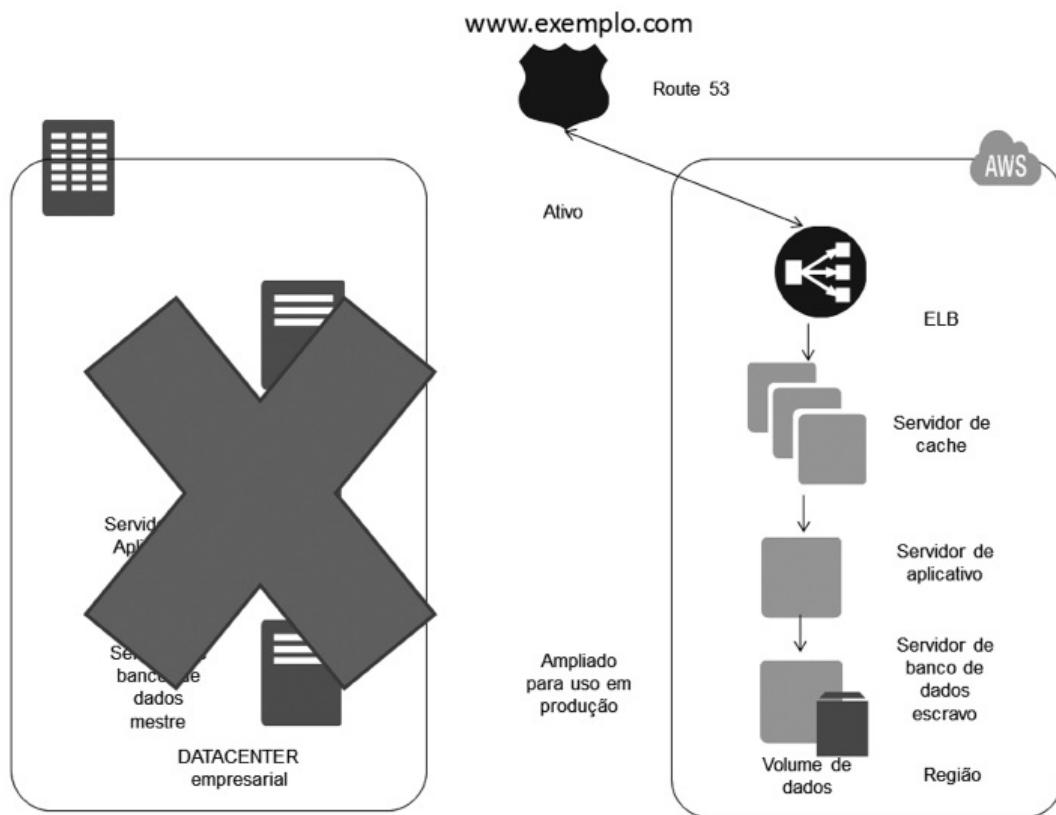


Figura 13-10 Fase de recuperação do cenário stand-by

Pontos-chave para recuperação:

- Inicie aplicativos em tipos maiores de instância EC2 conforme necessário (dimensionamento vertical).
- Aumente o tamanho das frotas EC2 em serviço com o Load Balancer (dimensionamento horizontal).
- Altere os registros DNS para que todo o tráfego seja roteado para o ambiente AWS.
- Considere o uso de *Auto Scaling* para o tamanho ideal do conjunto de instâncias ou para acomodar o aumento de carga.

### 13.4.4. Multisite

Uma solução multisite é executada na AWS e também na infraestrutura existente. O método de replicação de dados utilizado será determinado pelo

objetivo do ponto de recuperação (RPO) escolhido.

Um serviço DNS ponderado, como o Route 53, é utilizado para rotear o tráfego de produção para diferentes sites. Uma proporção de tráfego é roteada para a infraestrutura na AWS e o restante para a infraestrutura local.

Em uma situação de desastre no local, pode-se ajustar a proporção no DNS e enviar todo o tráfego para os servidores AWS. A capacidade do serviço AWS pode ser rapidamente aumentada para lidar com a carga de produção completa. O *Auto Scaling* do EC2 pode ser usado para automatizar esse processo. Pode ser necessária alguma lógica de aplicativo para detectar a falha dos serviços de banco de dados primário e destinar a operação aos serviços de banco de dados secundário.

O custo deste cenário é determinado pela quantidade de tráfego de produção controlado pela AWS em operação normal. Na fase de recuperação, paga-se pelo adicional utilizado e pelo período que o ambiente de DR é necessário em escala completa. Pode-se reduzir ainda mais os custos através da compra de instâncias reservadas aos servidores AWS “sempre em execução”.

#### 13.4.4.1. Fase de preparação

Na **Figura 13-11**, observa-se o uso do DNS para rotear uma parte do tráfego para o site AWS. O aplicativo na AWS pode acessar dados no sistema de produção local. Os dados são replicados ou espelhados para a infraestrutura AWS.

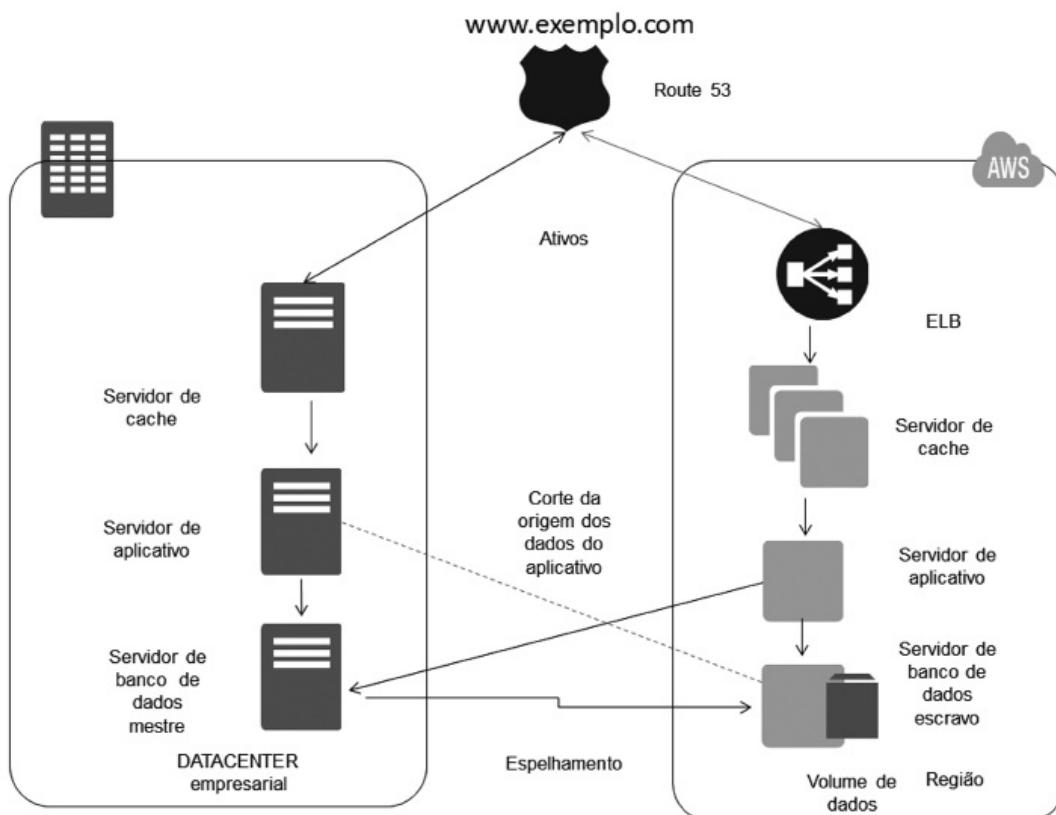


Figura 13-11 Fase de preparação do cenário multisite

Pontos-chave para preparação:

- Configure o ambiente AWS para duplicar o ambiente de produção.
- Configure a importância do DNS ou tecnologia similar para distribuir as solicitações de entrada para ambos os locais.

#### 13.4.4.2. Fase de recuperação

A **Figura 13-12** mostra o que acontece quando ocorrer um desastre no local. O tráfego é redirecionado para a infraestrutura AWS ao se atualizar o DNS.

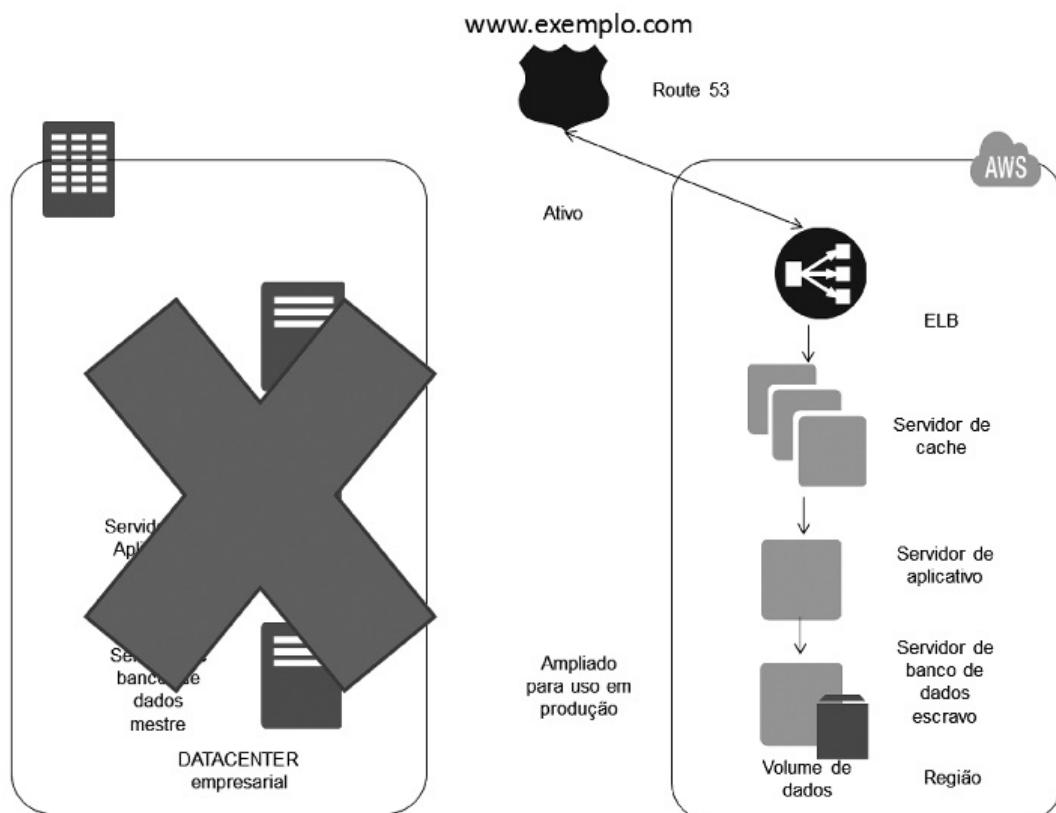


Figura 13-12 Fase de recuperação do cenário multisite

Pontos-chave para recuperação:

- Altere a importância do DNS, para que todas as solicitações sejam enviadas ao site AWS.
- Disponha de uma lógica de aplicativo para *failover* para usar os servidores de banco de dados locais AWS.
- Considere o uso de *Auto Scaling* para automaticamente definir o tamanho ideal do conjunto de instâncias AWS.

É possível aumentar ainda mais a disponibilidade da solução multisite através da concepção de arquiteturas *Multi-AZ*.

### 13.5. PCN e PRD

### 13.5.1. Conceitos

Um plano de continuidade do negócio (PCN) é um plano para a resposta de emergência, operações de backup e recuperação de ativos atingidos por falha ou desastre. Este plano tem como objetivo assegurar a disponibilidade de recursos de sistema críticos, recuperar um ambiente avariado e promover o retorno à sua normalidade.

Normalmente o PCN começa pela realização da análise de impacto no negócio (*Business Impact Analysis – BIA*) e pelo levantamento e priorização dos riscos.

O PCN é baseado em três outros planos descritos a seguir.

- **Plano de gerenciamento de crises (PGC):** este documento tem o propósito de definir as responsabilidades de cada membro das equipes envolvidas com o acionamento da contingência antes, durante e depois da ocorrência do incidente. Além disso, tem que definir os procedimentos a serem executados pela mesma equipe no período de retorno à normalidade. O comportamento da empresa na comunicação do fato à imprensa é um exemplo típico de tratamento dado pelo plano.
- **Plano de continuidade operacional (PCO):** tem o propósito de definir os procedimentos para contingenciamento dos ativos que suportam cada processo de negócio, objetivando reduzir o tempo de indisponibilidade e, consequentemente, os impactos potenciais ao negócio. Orientar as ações diante da queda de uma conexão à internet exemplifica os desafios organizados pelo plano.
- **Plano de recuperação de desastres (PRD):** tem o propósito de definir um plano de recuperação e restauração das funcionalidades dos ativos afetados que suportam os processos de negócio, a fim de restabelecer o ambiente e as condições originais de operação, no menor tempo possível. Uma empresa que utiliza os serviços AWS deve ter o seu próprio PRD. As atividades devem ser realizadas pela empresa e devem envolver recursos internos e recursos e serviços prestados pela própria AWS.

### 13.5.2. Objetivo

Um PCN deve fornecer recursos que permitam aos principais processos de negócio críticos voltarem a funcionar em um tempo aceitável após um desastre ou interrupção não programada. Quase sempre o PCN é desenvolvido considerando aplicações e infraestrutura, o que é um erro. Processos vitais como faturamento devem ser pensados primeiro, depois vêm as aplicações que suportam este processo e os ativos que suportam as aplicações. Ou seja, começa com um processo crítico e termina nos ativos que

suportam este processo, passando pela aplicação.

### **13.5.3. Aspectos importantes do PRD**

Alguns aspectos importantes precisam ser obedecidos para se ter um sólido plano de DR que envolva a AWS.

#### **13.5.3.1. Teste**

Depois que a solução de recuperação de desastres está pronta, ela precisa ser testada. O chamado “dia D” é quando se exerce um *failover* para o ambiente de DR. É preciso garantir que documentação suficiente esteja disponível, para simplificar ao máximo o processo em caso de um evento real ocorrer. Criar um ambiente duplicado para testar cenários durante o chamado dia D é rápido e econômico na AWS, e não é necessário nem tocar no ambiente de produção. Pode-se usar a AWS CloudFormation para implantar ambientes completos na AWS. Para isso se faz uso de um modelo para descrever os recursos AWS e quaisquer dependências ou parâmetros de tempo de execução associados, necessários para criar um ambiente completo.

Diferenciar os testes é fundamental para garantir que se esteja protegido contra uma enorme variedade de diferentes tipos de desastres. São exemplos de cenários de “dia D” comuns:

- Perda de energia para um local ou um conjunto de máquinas.
- Perda de conectividade de ISP para um único local.
- Vírus impactando os principais serviços de negócios que afetam multisites.
- Erro de usuário que causou a perda de dados que requer uma recuperação em um momento determinado.

#### **13.5.3.2. Monitoramento e alertas**

É preciso ter à disposição controle regular e uma monitoração suficiente para alertar quando o ambiente de DR for impactado por falha no servidor, problemas de conectividade e de aplicativo. O CloudWatch fornece métricas para uso dos recursos AWS. Os alertas podem ser configurados com base nos limites definidos em qualquer uma das métricas e, quando necessário, podem ser enviadas mensagens de serviço do Simple Notification Service (SNS) em caso de um comportamento inesperado. Podem ser utilizadas quaisquer soluções de monitoramento na AWS.

Também é possível continuar a usar qualquer ferramenta de monitoramento existente e alertas que a empresa usa para monitorar métricas de instância, bem como estatísticas de sistema operacional convidado e integridade de aplicativos.

#### **13.5.3.3. Backups**

Uma vez tendo alternado para o ambiente de DR, é imprescindível continuar a fazer backups regulares. Conduzir testes de backup e de restauração regularmente é essencial como uma solução de recuperação.

A AWS permite conduzir testes frequentes e econômicos sem a necessidade de infraestrutura de DR para estar “sempre em execução”.

#### **13.5.3.4. Automação**

Pode-se automatizar a implantação de aplicativos em servidores AWS e em servidores locais usando software de gerenciamento ou de orquestração de configuração. Isso permite lidar com aplicativos e alterações de gerenciamento de configuração de aplicativo em ambos os ambientes com facilidade.

O CloudFormation pode ser uma alternativa e funciona em conjunto com várias ferramentas para configurar os serviços de infraestrutura de forma automatizada.

O *Auto Scaling* pode ser usado para garantir que o pool de instâncias seja adequadamente dimensionado para atender à demanda com base em métricas específicas no CloudWatch. Isso significa que, em uma situação de recuperação de desastres, à medida que seu usuário base começa a usar mais o ambiente, a solução pode expandir dinamicamente para atender a essa demanda crescente. Após a finalização do evento e a diminuição potencial da utilização, a solução pode ser dimensionada de volta para um nível mínimo de servidores.

#### **13.5.3.5. Licenciamento de software**

Garantir que o licenciamento esteja correto para o ambiente de recuperação de desastres da AWS é tão importante quanto qualquer outro ambiente de licenciamento. A Amazon fornece os modelos descritos no capítulo 1 como opção para o licenciamento.

“Trazer sua própria licença” é possível para vários componentes de software ou sistemas operacionais. Como alternativa, há uma variedade de software para a qual o custo da licença está incluído na cobrança horária. Isso é conhecido como “licença incluída”.

“Trazer sua própria licença” permite aproveitar os investimentos existentes em software durante um desastre. “Licença incluída” minimiza os custos antecipados com licença para um local de DR que não é utilizado diariamente.

Pode-se proteger o acesso aos recursos no ambiente de DR usando o IAM. Desta maneira, criam-se políticas de segurança baseadas em funções/usuário que dividem as responsabilidades do usuário enquanto trabalham em seu ambiente de DR. O IAM foi tema do capítulo 3.

## **13.6. Referências bibliográficas**

Amazon Web Services. **AWS Direct Connect: Getting Starter Guide.** Versão 2011-08-03.

Amazon Web Services. **AWS Import-Export, Developer Guide.** API Version 2010-06-03.

Amazon Web Services. **AWS Storage Gateway, User Guide.** API Version 2012-04-30.

Amazon Web Services. **Backup and Recovery Approaches using Amazon Web Services.** 2012.

Amazon Web Services. **Storage Options in the AWS Cloud: Use Cases.** Dec. 2010.

Amazon Web Services. **Usando o Amazon Web Services para recuperação de desastres.** Outubro 2011.

## ***PARTE IV – CASO NACIONAL***

---

# 14. Caso Peixe Urbano

## 14.1. Introdução

O Peixe Urbano (<http://www.peixeurbano.com.br>) é uma startup brasileira baseada em um site web cuja finalidade é ajudar pessoas a descobrir e a experimentar produtos, serviços e atividades em diversas cidades existentes no Brasil e em alguns países. Para as empresas locais, parceiras do Peixe Urbano, funciona como uma ferramenta de marketing que permite divulgação além da conquista e fidelização de um grande número de novos clientes. Cada cidade tem o seu próprio site do Peixe Urbano, com ofertas especialmente formatadas para os seus moradores. O Peixe Urbano possui mais de vinte milhões de usuários.

## 14.2. Site

Hoje o Peixe Urbano está presente em mais de oitenta cidades do Brasil, Argentina, México e Chile e é o maior site de compras coletivas de origem latino-americana. Cada uma das cidades onde o Peixe Urbano está presente possui um site específico com ofertas exclusivas. A **Figura 14-1** ilustra o site Peixe Urbano para a cidade do Rio de Janeiro.



Figura 14-1 Peixe Urbano no Rio de Janeiro

## 14.3. Desafios

Alexander Tabor, sócio-fundador e CTO do Peixe Urbano, explica que o site começou suas operações com recursos limitados:

"Nós começamos a empresa do zero e tínhamos muito pouco para investir em infraestrutura. Embora fôssemos uma empresa pequena, já pensávamos no potencial de rápido crescimento caso o serviço fizesse sucesso. Como tal, precisávamos de uma solução que fosse ao mesmo tempo pequena em termos de custos de capital, porém altamente escalável. O Amazon Elastic Compute Cloud (Amazon EC2) era a mais madura das opções disponíveis e atendia a ambos os requisitos."

## 14.4. Benefícios

Quando o Peixe Urbano decidiu lançar seu site na AWS, evitou a despesa de investir em infraestrutura física. Agora, em vez de se preocupar em manter um DATACENTER tradicional, a empresa pode se concentrar em trazer as melhores ofertas diárias para os seus usuários nos diferentes mercados sem

se preocupar com aspectos de escalabilidade. Essa decisão certamente contribuiu para o crescimento do negócio, hoje um dos websites mais populares do país. Alexander Tabor observa:

“O AWS permitiu que lançássemos um site com baixo investimento de capital, o qual evoluiu para um dos sites mais acessados do Brasil, e tudo isso sem ter que alterar a nossa infraestrutura ou arquitetura.”

## 14.5. Startups

O Peixe Urbano pode ser considerado uma startup. O que seria uma startup<sup>[11]</sup>? Uma startup (empresa nascente, para alguns teóricos) é uma empresa nova, até mesmo embrionária ou ainda em fase de constituição, que conta com projetos promissores, ligados ao desenvolvimento de ideias inovadoras. Uma startup envolve risco. São normalmente empreendimentos com baixos custos iniciais e altamente escaláveis, ou seja, possuem uma expectativa de crescimento muito grande quando dão certo. Algumas empresas já solidificadas no mercado e líderes em seus segmentos, como Google e Facebook, também são consideradas startups. Adquirir serviços de TI na forma de cloud computing, diga-se de forma elástica, é aderente ao conceito de startup. Atender de forma imediata a imprevisibilidade da demanda é uma das grandes vantagens do modelo cloud computing.

Os principais conceitos que definem o que seja uma startup são assim descritos:

- Qualquer pequena empresa no período inicial.
- Empresa com custos de manutenção muito baixos, mas que consegue crescer rapidamente e gerar lucros cada vez maiores.
- Grupo de pessoas à procura de um modelo de negócios repetível e escalável, trabalhando em condições de extrema incerteza (melhor).

O terceiro conceito é mais completo e é baseado em:

- Ter um modelo de negócios que é como a startup gera valor – ou seja, como transforma seu trabalho em dinheiro.
- Ser repetível significa ser capaz de entregar o mesmo produto novamente em escala potencialmente ilimitada, sem muitas customizações ou adaptações para cada cliente.
- Ser escalável significa crescer cada vez mais, sem que isso influencie no modelo de negócios. Crescer em receita, mas com custos crescendo bem mais lentamente. Isso fará com que a margem seja cada vez maior, acumulando lucros e gerando cada vez mais riqueza.
- Possuir um cenário de incerteza significa que não há como afirmar se aquela ideia ou projeto de empresa irá realmente dar certo – ou ao

menos se provar sustentáveis.

## 14.6. Ágil

No cenário atual, a economia é digital, pois grande parte dos processos de negócio é amparada pela TI. Assim, as startups precisam ser ágeis e estar continuamente preparadas para as mudanças. Em empresas intensivas em TI (e cada vez é maior o número de empresas que empregam TI intensamente, incluindo as startups), esta questão se torna essencial para a sobrevivência.

O manifesto ágil surge então para rediscutir o processo de desenvolvimento de software e ressalta a importância de valores como foco no cliente, nas mudanças, na agilidade e em equipes multidisciplinares e extremamente capacitadas. Alguns aspectos pregados pelo movimento ágil são reforçados aqui:

- O manifesto não se aplica somente a desenvolvimento de produtos de software, mas, sim, a repensar as empresas amplamente.
- A organização ágil envolve processos e pessoas em todo o ciclo de negócios, em cenários onde as mudanças são uma constante, os prazos cada vez mais curtos e os clientes mais exigentes.
- Ágil implica em ciclos curtos de desenvolvimento com produto sendo constantemente apresentado ao cliente.

## 14.7. Startups enxutas (*lean startups*)

Com o objetivo de permitir que se crie um ambiente propício à inovação, surgiu a fórmula “lean startup”, que consiste na constante busca por um casamento perfeito entre o produto e o cliente e baseia-se essencialmente na ideia de que startups são hipóteses, e que é preciso aplicar “o método científico na identificação da oportunidade de mercado”.

Um ambiente propício a inovação pode ser:

- Um ambiente que permita a realização de melhorias (introduzindo novos recursos).
- Um ambiente que permita a ação de introduzir algo novo (The American Heritage Dictionary).
- Um ambiente que permita traduzir ideias novas em algo tangível com impactos sociais.
- Um ambiente que permita ter uma nova ideia, método ou dispositivo (Merriam-Webster Online).
- Um ambiente que permita a exploração bem-sucedida de novas ideias (Department of Trade and Industry, UK).

- Um ambiente que permita a mudança que cria uma nova dimensão de performance – Peter Drucker (Hesselbein, 2002).
- Um ambiente que permita que uma ideia criativa seja realizada (Harvard Business School Press, 2004).

A startup enxuta tira seu nome da produção enxuta, consagrada pela Toyota. Os princípios da produção enxuta são o aproveitamento do conhecimento de cada funcionário, a redução do tamanho dos lotes, a produção do tipo *just in time*, o controle de estoque e a aceleração do tempo de ciclo. A startup enxuta adota a ideia da produção enxuta para o empreendedorismo.

A startup enxuta pode ser considerada também um conjunto de processos usados por empreendedores para desenvolver produtos e mercados, combinando desenvolvimento ágil de software, desenvolvimento de clientela e plataformas existentes de software.

A iniciativa enxuta defende a criação de protótipos rápidos, projetados para validar suposições de mercado, e a utilização de feedback dos clientes para envolvê-los muito mais rapidamente do que através de práticas de desenvolvimento de software mais tradicionais, como o *waterfall model* (modelo em cascata).

#### **14.7.1. Princípios**

Os princípios da startup enxuta descritos por Eric Ries são:

- Empreendedores estão por toda parte.
- Empreender é administrar.
- Aprendizado validado.
- Construir-medir-aprender.
- Contabilidade para inovação.

O autor reforça a necessidade de administração desses empreendimentos e o aprendizado chamado de validado, que precisa ser demonstrado mediante a melhoria das métricas da startup. Também reforça que a contabilidade precisa utilizar uma abordagem quantitativa que permite monitorar os resultados financeiros.

#### **14.7.2. Ciclo construir-medir-aprender**

A startup é uma catalisadora que transforma ideias em produtos. À medida que os clientes interagem com os produtos, geram feedback e dados.

Eric Ries propõe o ciclo construir-medir-aprender para criação de protótipos empresariais rápidos e projetados para validar suposições de mercado. O ciclo permite utilizar o feedback dos clientes para envolvê-los

rapidamente.

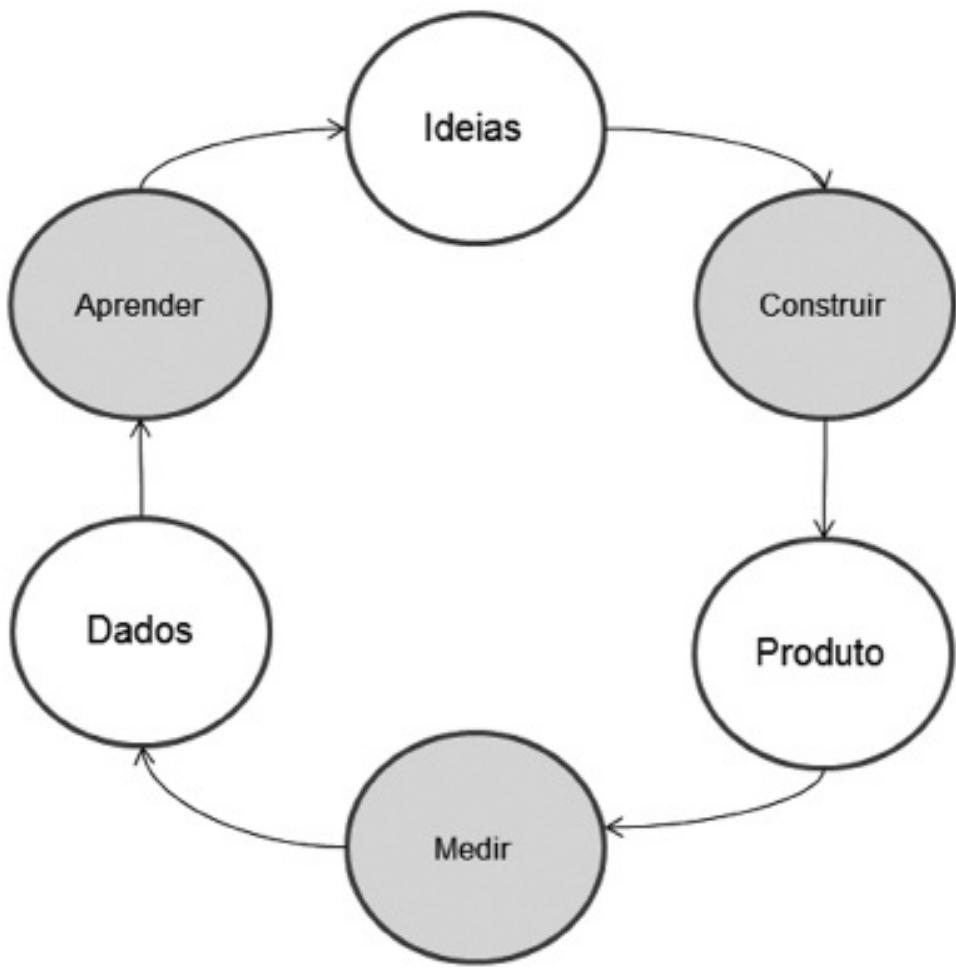


Figura 14-2 Ciclo construir-medir-aprender

#### 14.7.3. Minimum Viable Product (MVP)

Segundo Ries, um produto mínimo viável (MVP) ajuda os empreendedores a começar o processo de aprendizagem o mais rápido possível. No entanto, não é necessariamente o menor produto imaginable; trata-se, apenas, de um produto que permite percorrer o ciclo construir-medir-aprender de feedback com o menor esforço possível.

MVP, em sua essência, é uma estratégia usada para testes quantitativos e rápidos de um produto ou das suas características.

Ajuda a verificar se a hipótese estratégica original é correta ou se é necessário fazer uma grande mudança. A questão é: persevera ou testa uma nova hipótese?

- Se persevera:
  - Deve melhorar o produto.
    - Conseguir mais feedback.
    - Preparar-se para novas versões.

- Se testa uma nova hipótese:
- Se não cresce, deve mudar fundamentalmente a estratégia.
  - Deve decidir entre cortar gastos ou levantar mais dinheiro para o empreendimento.
- Deve ter uma ideia de quantas hipóteses ainda podem ser feitas.

A questão central para a startup é: quanto tempo e energia elas devem investir em infraestrutura de forma antecipada na expectativa de sucesso?

## 14.8. Infraestrutura

A AWS hospeda milhares de sites de empresas com perfil de startups, incluindo o Peixe Urbano. A condição de elasticidade propiciada por uma infraestrutura entregue como um serviço é uma característica aderente à realidade da maioria das startups. A AWS hospeda o site web do Peixe Urbano, assim como vários outros serviços assíncronos.

O Peixe Urbano utiliza linguagem Cº e desenvolveu o site web utilizando o AWS SDK para .NET, além de utilizar a interface de linha de comando da AWS.

O Peixe Urbano executa uma média de 180 instâncias EC2 simultaneamente. A capacidade de personalizar recursos em tempo real permite que a empresa mantenha a estabilidade do site web, mesmo durante picos de uso. Alexander Tabor, CTO do Peixe Urbano, diz: "Temos tido variações nos níveis de tráfego de um dia para o outro de até 1000%, e a infraestrutura foi capaz de suportar sem problemas."

### 14.8.1. Regiões

A AWS permite rodar as instâncias EC2 em múltiplos locais. Os locais da AWS são compostos por regiões e zonas de disponibilidade (*Availability Zones – AZ*), conforme visto no capítulo 2. Regiões consistem de uma ou mais zonas de disponibilidade separadas em áreas geográficas ou países.

O Peixe Urbano utiliza duas regiões diferentes, uma nos Estados Unidos e outra no Brasil, para efeito de aumento de desempenho.

### 14.8.2. Zonas de disponibilidade

As zonas de disponibilidade são zonas independentes onde estão os DATACENTERS. O Peixe Urbano utiliza mais de uma zona de disponibilidade na mesma região, para efeito da disponibilidade do ambiente.

No Peixe Urbano as instâncias EC2 rodam em múltiplas zonas de disponibilidade ao mesmo tempo. São utilizados serviços de balanceamento de carga (*Elastic Load Balancing – ELB*) que se integram a instâncias em

múltiplas zonas de disponibilidade para conseguir alta disponibilidade com baixa intervenção humana.

### **14.8.3. EC2**

O EC2 é um web service que permite ao Peixe Urbano utilizar recursos de processamento de acordo com a demanda de uso do aplicativo por parte dos clientes e pagar pelo uso. Como se sabe, no EC2 paga-se com base no tipo de instância e no uso por hora.

No caso do Peixe Urbano, instâncias EC2 são utilizadas para diversas finalidades, incluindo proxy, cache, servidores web, servidores de aplicação e gerenciadores de banco de dados.

O EC2 proporciona ao aplicativo do Peixe Urbano obter a capacidade de:

- Configurar os requisitos de computação instantaneamente.
- Ajustar a capacidade com base na demanda.

### **14.8.4. ELB**

O ELB faz o balanceamento da carga entre múltiplas instâncias EC2 e até entre instâncias em múltiplas zonas de disponibilidade de forma automática utilizando recursos de gerenciamento da plataforma baseados em métricas e alarmes.

No caso do Peixe Urbano, o ELB é utilizado tanto para os servidores web como para os servidores de aplicação.

### **14.8.5. Auto Scaling**

O *Auto Scaling* permite expandir ou reduzir a capacidade do EC2 automaticamente, de acordo com as condições predefinidas, como visto no capítulo 10. Com o *Auto Scaling*, o Peixe Urbano pode garantir que o número de instâncias EC2 que fazem parte do cluster ELB se redimensione facilmente durante picos de demanda para manter o desempenho e diminua automaticamente durante quedas de demanda, para minimizar custos.

O *Auto Scaling* é o instrumento principal utilizado pela AWS para a obtenção da elasticidade. No caso do Peixe Urbano, existem variações de carga da ordem de 1000%.

### **14.8.6. S3**

O Simple Storage Service (S3) é a forma de armazenamento principal da AWS. Ele armazena objetos distribuídos e foi projetado para armazenamento de dados primários, secundários e de missão crítica através de uma interface web service fácil de usar. No caso do Peixe Urbano, o storage S3 é utilizado como repositório de dados para backup.

### **14.8.7. Gerenciadores de banco de dados**

Na AWS pode-se utilizar uma instância Windows ou Linux EC2 e instalar um gerenciador de banco de dados qualquer aproveitando o benefício da licença BYOL (“traga sua própria licença”) permitida por alguns fornecedores de softwares gerenciadores de banco de dados.

O Peixe Urbano utiliza três bases de dados (SAP e outros sistemas) instaladas em instâncias EC2.

#### 14.8.8. Gerenciamento baseado no IAM

O Identity and Access Management (IAM) permite controlar com segurança o acesso aos serviços e recursos da AWS pelos usuários. Ele permite criar e gerenciar usuários. O IAM oferece maior segurança, flexibilidade e controle ao usar a AWS e é essencialmente um sistema de autorização. O Peixe Urbano utiliza o IAM para controle do acesso dos usuários ao sistema.

### 14.9. Arquitetura

A **Figura 14-3** ilustra a nuvem Amazon e a localização do cloud DATACENTER do Peixe Urbano. Observe que o Peixe Urbano, conforme mencionado, roda em vários DATACENTERS da Amazon AWS. Este é o conceito de cloud DATACENTER.

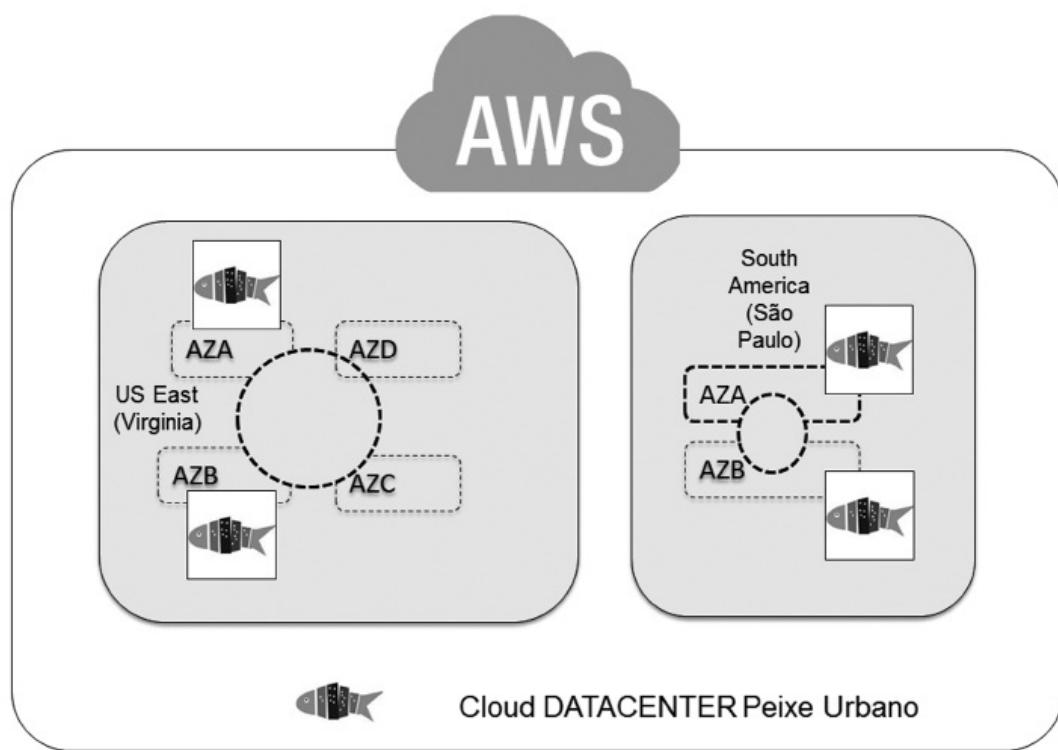


Figura 14-3 Cloud DATACENTER do Peixe Urbano

A **Figura 14-4** ilustra a arquitetura do Peixe Urbano. Alguns detalhes:

- Foi desenvolvida em três camadas (instâncias web em balanceamento de carga, instâncias de aplicação e servidores de

banco de dados) baseadas em instâncias EC2.

- Utiliza o web service S3 para armazenamento de conteúdo estático e para realizar o backup dos dados.
- Utiliza bancos de dados PostgreSQL e MySQL rodando em instâncias EC2.
- Ambientes de produção e desenvolvimento SAP na AWS rodando em instâncias EC2 e utilizando SQL Server Enterprise.
- Contempla um cache nginx baseado em instâncias EC2.
- Tanto o cache do nginx como os servidores de aplicação utilizam平衡adores de carga ELB com escalabilidade automática (*Auto Scaling*).
- O proxy reverso nginx utiliza no mínimo quatro instâncias e o servidor de aplicação, no mínimo cinco instâncias.

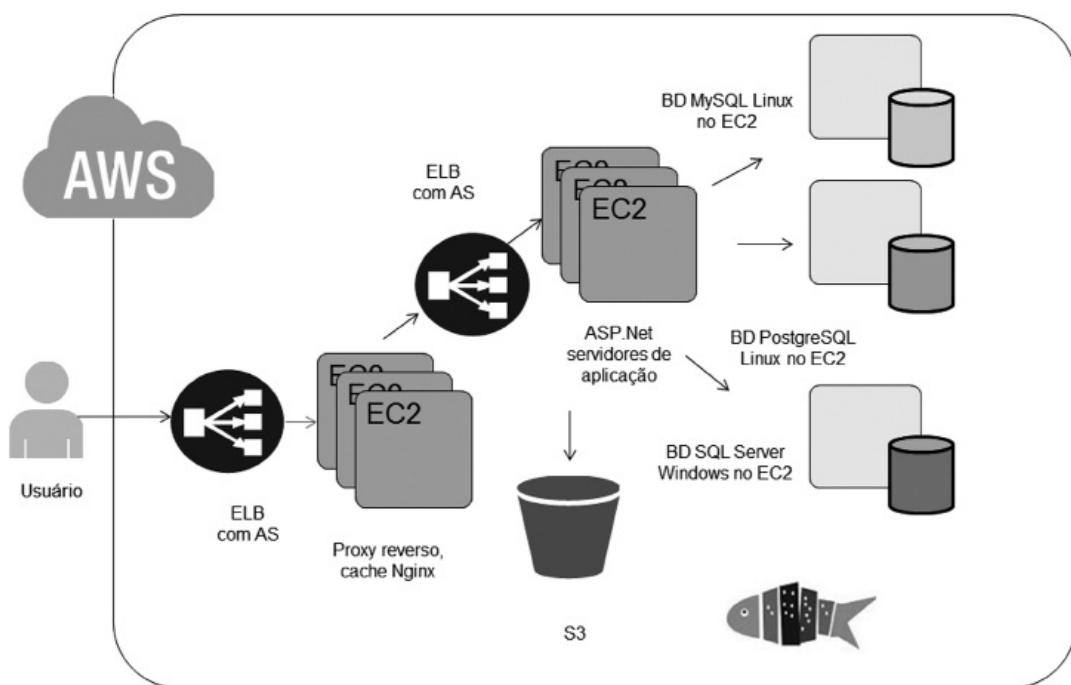


Figura 14-4 Arquitetura AWS para o Peixe Urbano

Constatações importantes relatadas pelos dirigentes do Peixe Urbano:

- O servidor de aplicação aumenta o número de instâncias várias vezes durante o dia.
- O cache do nginx aumenta a velocidade do site para usuários no Brasil e reduz o consumo dos servidores de aplicação.
- O cache do nginx permite configurar URLs parciais e direcioná-las para grupos de servidores diferentes.

A **Figura 14-5** ilustra a arquitetura de três camadas de forma detalhada. Instâncias rodam onde é mais apropriado rodar. Proxy reverso e cache estão

em zonas de disponibilidade (AZs) em São Paulo, e servidores de aplicação e de banco de dados estão em zonas de disponibilidade localizadas nos Estados Unidos.

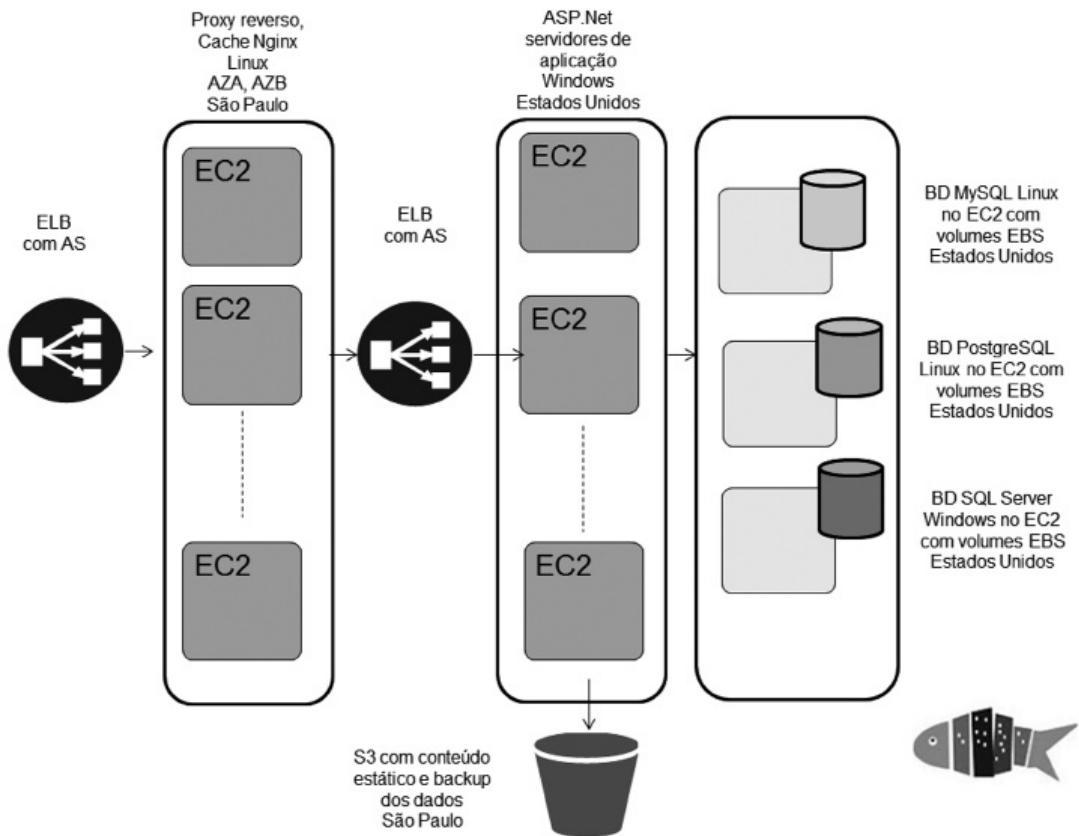


Figura 14-5 Detalhes da arquitetura

## 14.10. Referências bibliográficas

Amazon Web Services. Caso Peixe Urbano.  
<http://aws.amazon.com/pt/solutions/case-studies/peixe-urbano/>.  
Ries, Eric. **A Startup Enxuta**. Leya, 2012.

- [1] *Magic Quadrant for Cloud Infrastructure as a Service e Web Hosting*. Gartner, 2010.
- [2] *The Business Value of Amazon Web Services Accelerates Over Time*, IDC, 2012.
- [3] Ver em <http://aws.typepad.com/brasil/2012/06/amazon-s3-1-trilhao-de-objetos.html>.
- [4] Consultar o artigo *The Business Value of Amazon Web Services Accelerates Over Time*, IDC, 2012.
- [5] "Amazon DATACENTER Size" em: <http://huanliu.wordpress.com/2012/03/13/amazon-data-center-size/>
- [6] U (*rack unit*) é a unidade de medida utilizada para descrever a altura de servidores, switches e outros dispositivos montados em racks de 19 polegadas. Cada "rack unit" equivale a 44.45 mm (1.75").
- [7] *The Digital Universe Decade – Are You Ready?*, IDC, maio de 2010.
- [8] Consulte o artigo "Cloud Computing takes off", Morgan Stanley, 2011.
- [9] Serviços de Nuvem IBM: como a IBM está aumentando o valor agregado do ambiente de desenvolvimento na nuvem para os clientes, TBR, 2011.
- [10] Serviços de Nuvem IBM: como a IBM está aumentando o valor agregado do ambiente de desenvolvimento na nuvem para os clientes, TBR, 2011.
- [11] Uma boa referência é o livro "A Startup Enxuta", de Eric Ries, publicado no Brasil pela Editora Leya em 2012.



Prefácio  
José Papo  
AWS América Latina

MANOEL VERAS

# Arquitetura de Nuvem

AMAZON WEB SERVICES (AWS)

Explica a arquitetura AWS  
Relaciona os modelos IaaS e PaaS com a AWS  
Descreve os principais serviços e produtos da AWS  
Ensina a montar a arquitetura e construir o DATACENTER com recursos da AWS  
Apresenta o caso AWS Peixe Urbano



