

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**AUTOMATIZAÇÃO DO GERENCIAMENTO E
IMPLANTAÇÃO DE INSTÂNCIAS AWS EC2
PARA ENSINO DE COMPUTAÇÃO
DISTRIBUÍDA**

TRABALHO DE GRADUAÇÃO

Cezar Augusto Contini Bernardi

Santa Maria, RS, Brasil

2016

AUTOMATIZAÇÃO DO GERENCIAMENTO E IMPLANTAÇÃO DE INSTÂNCIAS AWS EC2 PARA ENSINO DE COMPUTAÇÃO DISTRIBUÍDA

Cezar Augusto Contini Bernardi

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da
Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para
a obtenção do grau de

Bacharel em Ciência da Computação

Orientador: Prof. Dr. João Vicente Ferreira Lima

**396
Santa Maria, RS, Brasil**

2016

AGRADECIMENTOS

Valeu gurizada

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

AUTOMATIZAÇÃO DO GERENCIAMENTO E IMPLANTAÇÃO DE INSTÂNCIAS AWS EC2 PARA ENSINO DE COMPUTAÇÃO DISTRIBUÍDA

AUTOR: CEZAR AUGUSTO CONTINI BERNARDI

ORIENTADOR: JOÃO VICENTE FERREIRA LIMA

Local da Defesa e Data: Santa Maria, dia de mês de 2016.

Resuminho

Palavras-chave: Cloud Computing. Computação Distribuída. Orquestração. Ensino.

ABSTRACT

Undergraduate Final Work
Undergraduate Program in Computer Science
Federal University of Santa Maria

DEPLOYMENT AND MANAGEMENT AUTOMATION OF AWS EC2 INSTANCES FOR DISTRIBUTED COMPUTING TUTORSHIP

AUTHOR: CEZAR AUGUSTO CONTINI BERNARDI

ADVISOR: JOÃO VICENTE FERREIRA LIMA

Defense Place and Date: Santa Maria, May diath, 2016.

Abstract

Keywords: Cloud Computing. Orchestration, Tutorship, Distributed Computing.

SUMÁRIO

LISTA DE FIGURAS	7
1 INTRODUÇÃO.....	8
1.1 Objetivos.....	8
1.2 Justificativa	8
1.3 Organização do texto.....	9
2 FUNDAMENTOS E REVISÃO DE LITERATURA.....	10
2.1 Computação em Nuvem	10
2.2 Amazon Web Services	11
2.2.1 Amazon Elastic Compute Cloud - EC2	11
2.2.2 Amazon Virtual Private Cloud - VPC	14
2.2.3 Amazon CloudFormation	14
2.2.4 Amazon Simple Storage Service - S3	14
2.3 Casos de Uso	14
2.4 ??? tchê	16
2.5 Trabalhos Relacionados	16
3 DESENVOLVIMENTO.....	17
3.1 Ambiente de rede	17
3.2 Instâncias	18
3.2.1 Acesso SSH	18
3.2.2 Instalação de Pacotes	19
3.2.2.1 OpenMPI	19
4 RESULTADOS	21
5 CONCLUSÃO	22
REFERÊNCIAS	23

LISTA DE FIGURAS

2.1	<i>Infraestrutura Geral da Amazon Web Services (INFRAESTRUTURA GLO- BAL, 2016)</i>	11
2.2	<i>Modelo de VPC com subredes, grupos de segurança, ACL, tabelas de rote- amento, internet gateway e VPG (TONELLI, 2013)</i>	15

1 INTRODUÇÃO

Os serviços na nuvem, também conhecidos como cloud computing, tem expandido muito ultimamente, proporcionando diversas oportunidades de aplicação, nem sempre de fácil uso. Esses serviços podem ser classificados em três principais categorias, sendo elas Infraestrutura como um Serviço (IaaS), Plataforma como um Serviço (PaaS) e Software como um Serviço (SaaS) (SOUSA et al., 2010).

Destre estes modelos, o IaaS, aqui utilizado, garante muita flexibilidade para se adequar a diferentes requisitos de ensino, se tornando uma ferramenta muito útil para permitir maior contato com ambientes realistas por parte dos alunos. Ou também acesso a recursos computacionais inalcançáveis de outra maneira, como clusters.

Porém, a utilização de tais recursos requer esforço de gerência para implantação e uso sem demais problemas. O que este projeto visa é a automatização dessa etapa, facilitando o acesso a esses recursos dentro da universidade. Diversas instituições já vem implementando suas próprias soluções de orquestração, como Stanford University e Massachusetts Institute of Technology.

1.1 Objetivos

Desenvolver um modelo de infraestrutura para Amazon Web Services focado no processo de ensino de programação paralela e distribuída, de forma que possa ser facilmente adaptado a mudanças conforme desejar o educador.

1.2 Justificativa

O ensino de computação distribuída será beneficiado significativamente, abrindo oportunidades de exploração e experimentação de MPI para programação paralela ou RMI/RPC para programação distribuída em um ambiente realista. Um ponto importante é que cada aluno teria acesso a esse ambiente para si só e poderia experimentar mais e compreender melhor o funcionamentos dessas técnicas de programação.

Também facilitará o trabalho do educador, o qual terá um ambiente totalmente controlado e replicável para repassar o funcionamento das ferramentas a serem ensinadas.

1.3 Organização do texto

Este trabalho está organizado da seguinte forma: O capítulo 2 apresenta fundamentação, ferramentas e trabalhos relacionados que fazem parte do tema e da proposta de solução do trabalho.

O capítulo 3 detalha a implementação do trabalho, apresentando o processo de desenvolvimento da solução e como as ferramentas apresentadas no capítulo 2 foram utilizadas.

No capítulo 4 são apresentados os resultados do trabalho: Como utilizar a ferramenta e o que o ambiente implantado disponibiliza para desenvolvimento. E por fim, no capítulo 5, apresentam-se as considerações finais e conclusões do trabalho.

2 FUNDAMENTOS E REVISÃO DE LITERATURA

Para a realização deste trabalho foi necessário escolher uma ferramenta de IaaS como plataforma de desenvolvimento, que suportasse escalabilidade e orquestração. A plataforma escolhida foi a Amazon Web Services (AMAZON WEB SERVICES, 2016), por preencher os requisitos, além de disponibilizar concessões a alunos e educadores.

2.1 Computação em Nuvem

Dentre os modelos de cloud, o PaaS oferece abstração de diversas camadas, como sistema operacional, rede e infraestrutura, deixando-as prontas para o uso no desenvolvimento de aplicativos a serem disponibilizados na nuvem. Com isso o desenvolvedor pode focar mais no desenvolvimento. Um exemplo notável de tal serviço é o Google App Engine.

De forma semelhante, o SaaS disponibiliza aplicativos prontos para uso, novamente citando o Google como exemplo: Google Apps For Work (Drive, Gmail, Docs, entre outros).

Porém, este projeto visa o uso de IaaS, que foca em dar liberdade ao usuário montar a sua própria infraestrutura, gerenciando suas próprias máquinas virtuais, rede, sistema operacional e ambiente de desenvolvimento. A plataforma de IaaS da Amazon foi escolhida por ter infraestrutura física própria e prover acesso a educadores e alunos (AMAZON WEB SERVICES EDUCATE, 2016), além de escalabilidade e ferramentas que aprimoram a replicabilidade do trabalho. Para replicabilidade, será feito o uso do Amazon CloudFormation (AMAZON CLOUDFORMATION, 2016), o qual recebe um arquivo JSON descrevendo a pilha a ser inicializada e a implanta. A pilha a ser implantada utilizará os serviços de Elastic Compute Cloud (EC2)(AMAZON ELASTIC COMPUTE CLOUD, 2016) e Virtual Private Cloud(AMAZON VIRTUAL PRIVATE CLOUD, 2016) para melhor organização e acesso.

O uso de computação em nuvem também possibilita maior foco no ensino do que em gerenciamento de sistemas por parte dos professores (MIRCEA; ANDREESCU, 2011), aumentando a qualidade final do curso e da instituição em geral. Outro benefício gerado pelo uso de sistemas em cloud é a redução de custos para a instituição, reduzindo a necessidade de servidores locais (TOUT; SVERDLIK; LAWVER, 2009).

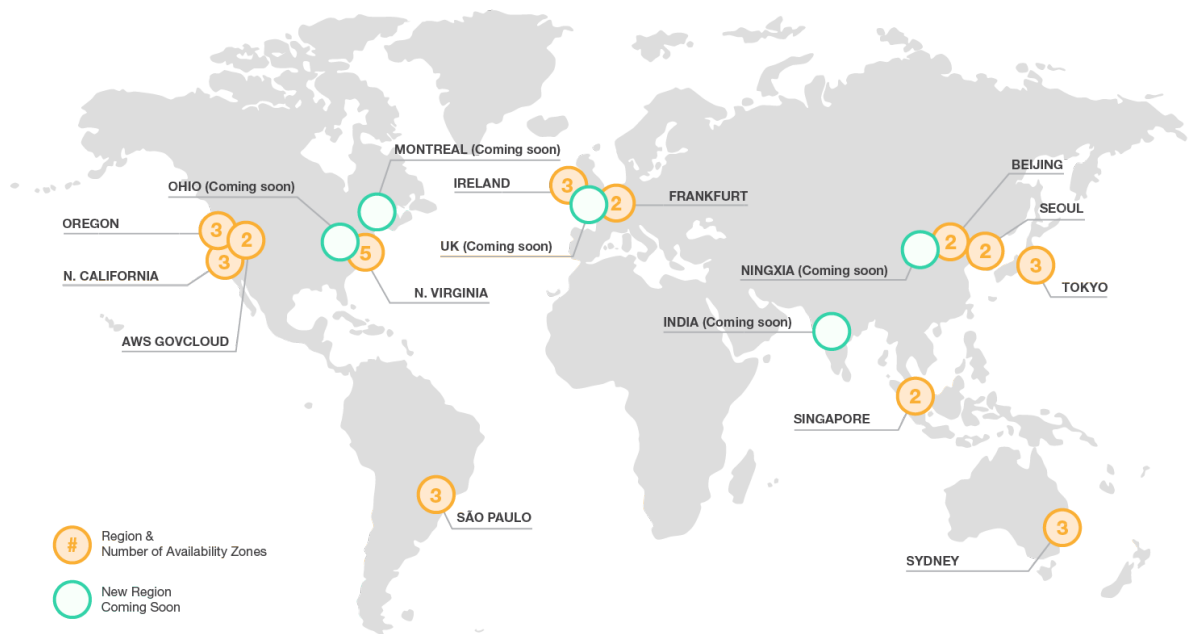


Figura 2.1: *Infraestrutura Geral da Amazon Web Services (INFRAESTRUTURA GLOBAL, 2016)*

2.2 Amazon Web Services

É uma plataforma de IaaS que disponibiliza um amplo conjunto de serviços. Esses serviços são naturalmente integrados, possibilitando a elaboração de infraestruturas adequadas a necessidade com grande facilidade.

Conta com diversas regiões pelo mundo onde sua infraestrutura está instalada (Figura 2.1), abrangendo o acesso com baixa taxa de atraso.

Assim como a maioria dos serviços disponíveis para nuvem, o AWS tem diferentes valores associados aos seus recursos, incluindo uma nível de uso gratuito, com limitações (EC2 INSTANCE PRICING, 2016).

2.2.1 Amazon Elastic Compute Cloud - EC2

Amazon Elastic Compute Cloud (EC2) faz parte da solução de IaaS da Amazon, provendo recursos de computação em nuvem. O serviço de EC2 conta com diversos tipos de instâncias que melhor se adequam a diferentes propósitos dentro da computação. Esses tipos são:

- Uso Geral

- T2 - São instâncias com menor custo. Tem capacidade de intermitência, contando com processadore Intel Xeon com até 3.3 GHz em frequência de turbo, com 1 ou 2 vCPUs com 0.5 a 8 GB de RAM. Seus principais casos de uso são ambientes de desenvolvimento, servidores de compilação, repositórios de código, sites e aplicações web de baixo tráfego, microsserviços. experimentos iniciais de produtos e pequenos bancos de dados. Utiliza Amazon EBS (Elastic Block Store) como dispositivo de armazenamento.
 - M4 - Tem hardware mais robusto que as instâncias T2, utilizando processadores Intel Xeon® E5-2676 v3 2,4 GHz (arquitetura Haswell), com 2 a 40 vCPUs e 8 a 160 GB de RAM. Seu armazenamento em EBS é otimizado, com taxas de transferência de 450 a 4000 Mbps.
 - M3 - Contam com processadores Intel Xeon E5-2670 v2 (Ivy Bridge) ou Xeon E5-2670 (Sandy Bridge) 2.6 GHz, com 1 a 8 vCPUs, 3.75 a 30 GB de RAM e armazenamento em SSD, seu diferencial das demais instâncias dessa categoria. Seus casos de uso mais indicados são bancos de dados de pequeno e médio porte, tarefas de processamento de dados que exigem memória adicional, grupos de armazenamento em cache e servidores de backend para SAP, Microsoft SharePoint, computação em cluster e outras aplicações empresariais.
- Otimizadas para computação
 - C4 - Utilizam processadore Intel Xeon E5-2666 v3 (Haswell) de alta frequência otimizados especificamente para o EC2, com 2 a 36 vCPUs e 3.75 a 60 GB de RAM. O armazenamento EBS é otimizado, com taxas de transferência entre 500 e 4000 Mbps. Esta instância, assim como a C3 tem suporte aprimorado a rede, aumentando a quantidade de pacotes por segundo que podem ser enviados e recebidos.
 - C3 - Com processadores Intel Xeon E5-2680 v2 (Ivy Bridge) de 2 a 32 vCPUs, 3,75 a 60 GB de RAM e armazenamento em SSD, seus casos de uso principais são frotas de frontend de alto desempenho, servidores da web, processamento em lotes, dados analíticos distribuídos, aplicativos científicos e de engenharia de alto desempenho, veiculação de anúncios, jogos MMO e codificação de vídeo.
 - Otimizadas para memória

- R3 - O foco deste tipo de instância é a memória RAM, contando com valores de 15.25 a 244 GB, tem também armazenamento em SSD, tornando-a ideais para bancos de dados de alto desempenho, caches de memória distribuídos, análises em memória, montagem e análise de genomas, implementações maiores de SAP, Microsoft SharePoint e outros aplicativos empresariais.
- GPU
 - G2 - Conta com 1 ou 4 GPUs NVidia, com 1.536 núcleos CUDA e 4 GB de VRAM. Seu propósito é streaming de aplicações 3D, aprendizagem de máquina, codificação de vídeo e outras cargas de trabalho de gráficos ou computação de GPU no lado do servidor. Para isso conta com processador Intel Xeon E5-2670 (Sandy Bridge) com 8 ou 32 vCPUs, 15 ou 60 GB de RAM e armazenamento em SSD.
- Otimizadas para armazenamento
 - I2 - Com foco em E/S, possui de 1 a 8 SSDs com 800 GB cada, processadores Intel Xeon E5-2670 v2 (Ivy Bridge) com 4 a 32 vCPUs e 30.5 a 244 GB de RAM. O suporte a rede também é aprimorado. Seus casos de uso incluem bancos de dados NoSQL, como o Cassandra e o MongoDB, bancos de dados transacionais escaláveis, armazém de dados, Hadoop e sistemas de arquivo em cluster.
 - D2 - Instâncias de armazenamento denso, contam com processadores Intel Xeon E5-2676v3 (Haswell) com 4 a 36 vCPUs, 30.5 a 244 GB de RAM e de 3 a 20 HDDs de 2 TB. Essas instâncias são aplicáveis a armazéns de dados para processamento paralelo massivo (MPP), computação distribuída de MapReduce e Hadoop, sistemas de arquivo distribuídos, sistemas de arquivos de rede e aplicações de processamento de logs ou dados.

Cada tipo e sub-tipo de instância tem um valor de utilização por hora diferente, crescendo consideravelmente ao se utilizar recursos melhores.

Além dos tipos de instância, estão disponíveis, na forma de AMIs (AMAZON MACHINE IMAGES, 2016), diversos tipos de imagens de máquinas virtuais, baseados em diferentes sistemas operacionais. Cada uma das AMIs tem características que as tornam melhores para determinadas tarefas computacionais, geralmente diferenciadas por recursos pré-instalados e configurados.

2.2.2 Amazon Virtual Private Cloud - VPC

Um recurso do AWS feito para possibilitar a divisão de outros recursos, principalmente EC2, em subnets. Com isso, é possível definir um ponto único de acesso aos demais recursos, aumentando a organização, escalabilidade e segurança da infraestrutura.

O VPC permite a personalização e divisão de endereços IP, levando a possibilidade de separação ou agrupamento de serviços. Também permite a comunicação com serviços em diferentes VPCs.

Também suporta a implantação de grupos de segurança, tabelas de roteamento e listas de controle de acesso, possibilitando a formação de uma infraestrutura robusta de rede, conforme Figura 2.2.

2.2.3 Amazon CloudFormation

Oferece um modo simples de implantar e gerenciar um grupo de recursos do AWS, de forma previsível e replicável. O uso dessa ferramenta pode ser feita de duas formas principais, sendo elas:

- Declaração estruturada em JSON Através de um arquivo de texto estruturado é possível descrever toda a infraestrutura a ser criada, com parâmetros variáveis ou estáticos para cada recurso da pilha.
- CloudFormation Designer Uma interface gráfica que disponibiliza o drag-and-drop dos recursos afim de facilitar ainda mais a criação de uma pilha de recursos AWS.

2.2.4 Amazon Simple Storage Service - S3

Um serviço de armazenamento da Amazon, baseado em buckets, oferece uma interface web service para armazenar e recuperar dados.

Esse serviço está bem acoplado aos demais serviços da Amazon, possibilitando o acesso via instâncias EC2, por exemplo, centralizando o armazenamento de forma confiável.

2.3 Casos de Uso

- System Administration, Gjovik University College por Erik Hjelmås (HJELMAS, 2015)
 - Faz o uso de OpenStack, disponibilizando um ambiente exclusivo a cada aluno para ex-

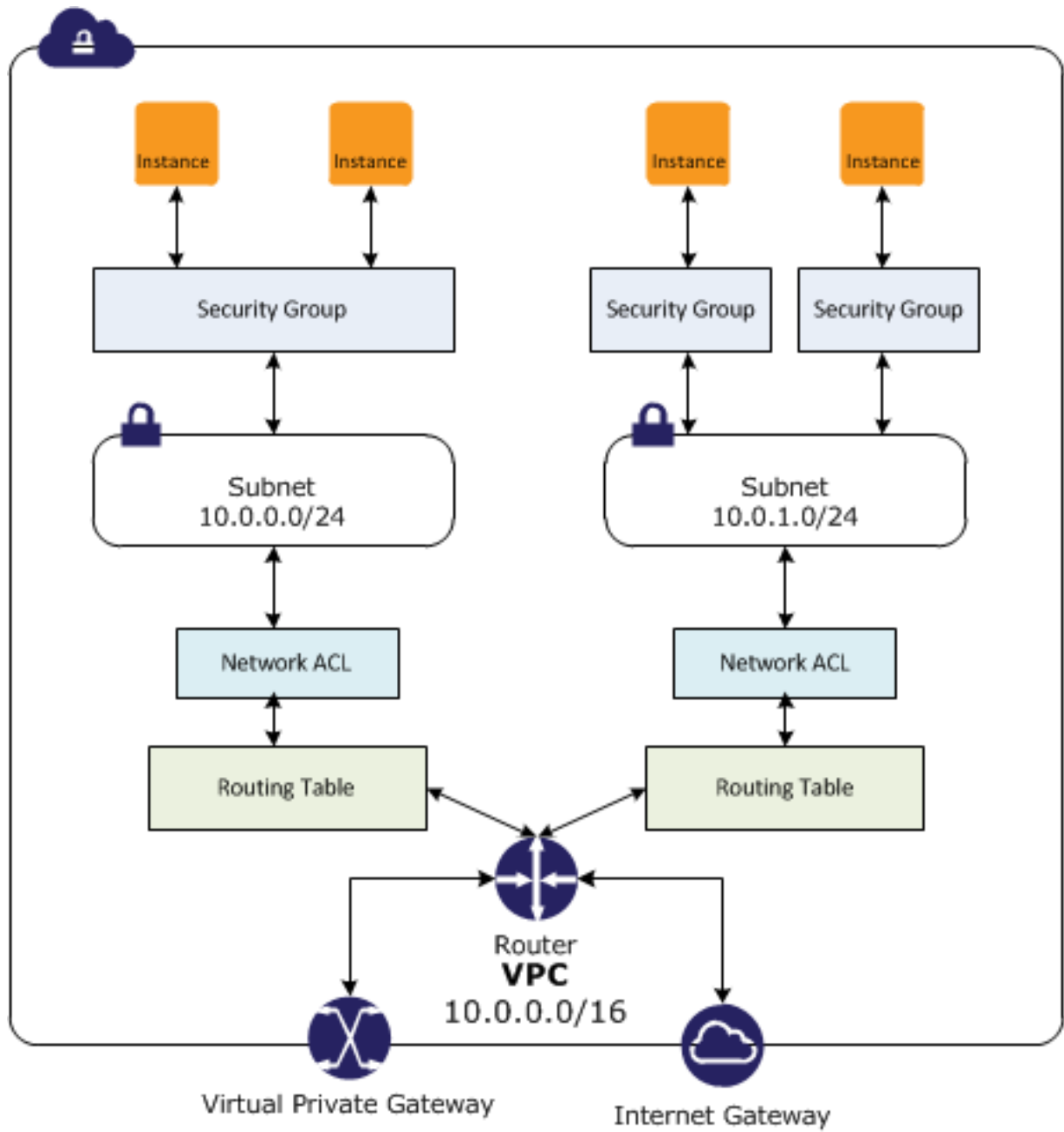


Figura 2.2: *Modelo de VPC com subredes, grupos de segurança, ACL, tabelas de roteamento, internet gateway e VPG (TONELLI, 2013)*

perimentação. O ambiente provido pelo professor conta com algumas VMs de diferentes sistemas operacionais para que o aluno as configure, experimente e aprenda o conteúdo.

- Parallel Computing, Stanford University por Alex Aiken e Kunle Olukotun (SLAUGHTER, 2014) - Utiliza diversas infraestruturas AWS para realização de tarefas em threads, STM, MapReduce (Hadoop), OpenMP e GPGPU (CUDA).
- Introduction to Modeling and Simulation, Massachusetts Institute of Technology por Markus J. Buehler e Jeffrey C. Grossman (TECHNOLOGY, 2011a) - Dado o caráter naturalmente intensivo em computação da matéria, foi implementado um cluster em cloud que acomodasse o uso por parte de todos os alunos.

2.4 ??? tchê

E agora?

2.5 Trabalhos Relacionados

Existem outras ferramentas para facilitar a implantação de ambientes em EC2. Cada um desses projetos tem um foco diferente dos demais no que se refere a orquestração do ambiente, e normalmente requerem o uso de mais ferramentas auxiliares para o pleno funcionamento.

- Stanford Elliot Slaughter - Baseado em um script Python para automatizar a implantação de diferentes tipos de clusters. Cada um desses tipos contém as ferramentas e ambiente preparado para um conteúdo específico dentre threads, STM, MapReduce (Hadoop), OpenMP e GPGPU (CUDA). A forma como o ambiente é implantada, e quais ferramentas instaladas, pode ser alterada por meio de modificações no script e nos arquivos de configuração.
- StarCluster (TECHNOLOGY, 2011b) - Provê uma interface em CLI para criação, configuração, gerência e acesso a clusters na Amazon EC2. Este projeto automatiza a implantação em si, mas não os passos de configuração do ambiente e de ferramentas disponíveis para desenvolvimento.

3 DESENVOLVIMENTO

Foi realizada uma pesquisa de opinião dos educadores. Tal pesquisa foi feita com o intuito de averiguar a aceitação em introduzir um novo método no processo de ensino. Além disso, foi necessário averiguar quais ferramentas, frameworks, bibliotecas e aplicativos são necessários para o andamento do conteúdo a ser ensinado.

3.1 Ambiente de rede

A metodologia de desenvolvimento do projeto foi incremental, primeiramente focando em criar um ambiente base onde as instâncias serão posicionadas. Esse ambiente base foi feito em volta de um VPC. Primeiramente, estabeleceu-se uma rede para endereços privados sob a qual todas as instâncias estarão alocadas. Decidiu-se pelo uso da rede 192.168.0.0/24 por ser um espaço de rede comumente utilizado para o propósito de redes privadas, além de fornecer endereços suficientes para a andamento do projeto.

Esta rede foi subdividida em duas, sendo as subredes 192.168.0.0/28 para uso das instâncias de computação e 192.168.240.0/28 para a instâncias com acesso externo. A instância colocada sob a subrede pública além de funcionar como *bastion* de acesso a rede privada com as instâncias de computação, também funciona como NAT (Networking Address Translation) para citadas instâncias. Tal medida foi tomada pois facilita o controle sobre endereçamento e organização dentro do VPC.

/* TODO: Regiões e AMIs? */

Como boa prática de segurança, foram associados grupos de segurança as subredes, tornando disponível o acesso a redes externas somente pela instância NAT, também apenas permitindo o acesso SSH via esta e demais nodos da subrede. Além dos grupos de segurança, tabelas de roteamento foram implantadas para direcionamento do tráfego de rede aos respectivos gateways. Do ponto de vista da subrede pública, esse gateway é um recurso especial da Amazon que permite o acesso a internet, chamado Internet Gateway, e do ponto de vista da subrede privada o gateway é a própria instância NAT.

/* TODO: Figuras do código? *Imagem representativa do ambiente? */

3.2 Instâncias

Para realização do projeto foram utilizadas instâncias do tipo *t2.micro*, as quais se encaixam no plano de uso livre da AWS. Esse tipo pode ser alterado no momento da implantação do modelo, com a ressalva de que os servidores AWS da América do Sul não suportam o tipo *g2*, com GPUs. Porém, essas instâncias podem ser utilizadas se a região escolhida for outra, como *us-east*.

Para configuração de arquivos e instalação de pacotes necessários foram utilizados as ferramentas disponibilizadas pelo recurso Init do CloudFormation. Dentre as opções disponíveis, foram utilizadas as seguintes:

- Packages - Provê suporte aos gerenciadores de instalação *apt*, *msi*, *python*, *rpm*, *rubygems* e *yum*.
- Files - Suporte para criação de arquivos e diretórios, assim como seus conteúdos.
- Commands - Disponibiliza a execução de comandos *bash*, no caso de instâncias Linux.

Essas opção são sempre executadas pelo CloudFormation na ordem descrita, evitando a edição de pacotes ou arquivos ainda inexistentes.

3.2.1 Acesso SSH

O primeiro desafio encontrado nesta etapa foi a simplificação da configuração do acesso SSH entre as instâncias, necessário a execução de algoritmos distribuídos com MPI. Por padrão, instâncias EC2 já têm a chave pública referente ao *key pair* configurado previamente a aplicação da pilha, porém, a chave privada está apenas disponível em arquivo PPK e é utilizada da máquina do usuário para acesso as instância pública. Isto não permite que as instâncias tenham acesso entre si via SSH, então foi necessário requerer a chave privada como *string* no momento de implantação, salvá-las nas VMs e reorganizar a estrutura do arquivo com um script python, pois a quebras de linha são perdidas e são necessárias para a correta interpretação da chave.

```

1 import re
2 input = open("/home/ec2-user/.ssh/id_rsa.in").read()
3 bgnEnd = re.findall('-----[A-Z\s]+-----', input)
4 content = re.findall('-\s(.+)\s-', input)
5 lines = content[0].split(" ")
6 outputFile = open("/home/ec2-user/.ssh/id_rsa", "w")
7 outputFile.write(bgnEnd[0] + '\n')
8 [outputFile.write(line + '\n') for line in lines]
```

```

9 outputFile.write(bgnEnd[1])
10 outputFile.close()

```

Outra modificação aplicada foi a edição do arquivo `/etc/hosts` para adicionar *alias* aos endereços das instâncias da subrede.

```

1 "AWS::CloudFormation::Init" : {
2   "config" : {
3     "files" : {
4       "/etc/hosts" : {
5         "content" : {
6           "Fn::Join" : ["\n", [
7             "127.0.0.1    localhost    localhost",
8             {"Fn::FindInMap":["IpAddressConfig", "Node01", "IP"]}, " node01",
9             {"Fn::FindInMap":["IpAddressConfig", "Node02", "IP"]}, " node02",
10            {"Fn::FindInMap":["IpAddressConfig", "Node03", "IP"]}, " node03",
11            {"Fn::FindInMap":["IpAddressConfig", "Node04", "IP"]}, " node04",
12            {"Fn::FindInMap":["IpAddressConfig", "Node05", "IP"]}, " node05",
13            "node05"]],
14         }
15       }
16     }
17 }

```

3.2.2 Instalação de Pacotes

A imagem utilizada nas VMs já possui repositórios yum configurados, disponibilizando a instalação dos pacotes `openmpi`, `gcc` e `g++` /*até agora*/.

3.2.2.1 OpenMPI

A configuração desse pacote começa com a exportação de variáveis de ambiente, indicando os *wrappers* de compilação e execução. Para isso foi utilizado o recurso de execução de comandos dos `Init`.

```

1 "AWS::CloudFormation::Init" : {
2   "config" : {
3     "commands" : {
4       "exportOpenMPIBin" : {
5         "command" : "echo \"export PATH=/usr/lib64/openmpi/bin:$PATH\" >> /
6         home/ec2-user/.bashrc"
7       },
8       "exportOpenMPILib" : {
9         "command" : "echo \"export LD_LIBRARY_PATH=/usr/lib64/openmpi/lib:
10         $LD_LIBRARY_PATH\" >> /home/ec2-user/.bashrc"
11       }
12     }
13 }

```

```
10     }  
11   }  
12 }
```

4 RESULTADOS

Funciona assim ó:

5 CONCLUSÃO

É isso champs.

REFERÊNCIAS

- AMAZON CloudFormation. Disponível em: <<https://aws.amazon.com/pt/cloudformation/>>, acessado em Março de 2016.
- AMAZON Elastic Compute Cloud. Disponível em: <<https://aws.amazon.com/pt/ec2/instance-types/>>, acessado em Abril de 2016.
- AMAZON Machine Images. Disponível em: <<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>>, acessado em Abril de 2016.
- AMAZON Virtual Private Cloud. Disponível em: <<https://aws.amazon.com/vpc/>>, acessado em Abril de 2016.
- AMAZON Web Services Educate. Disponível em: <<https://aws.amazon.com/pt/education/awseducate/>>, acessado em Março de 2016.
- AMAZON Web Services. Disponível em: <<https://aws.amazon.com/pt/>>, acessado em Março de 2016.
- EC2 Instance Pricing. Disponível em: <<https://aws.amazon.com/pt/ec2/pricing/>>, acessado em Abril de 2016.
- HJELMAS, E. **Lecture notes in System Administration**. [S.l.]: Gjovik University College, 2015.
- INFRAESTRUTURA Global. Disponível em: <<https://aws.amazon.com/pt/about-aws/global-infrastructure/>>, acessado em Março de 2016.
- MIRCEA, M.; ANDREESCU, A. I. Using cloud computing in higher education: a strategy to improve agility in the current financial crisis. **Communications of the IBIMA**, [S.l.], 2011.
- SLAUGHTER, E. **CS-149 Setup**. Disponível em: <<https://bitbucket.org/elliottslaughter/cs149-setup>>, acessado em Março de 2016.
- SOUSA, F. R. et al. Gerenciamento de dados em nuvem: conceitos, sistemas e desafios. **Topicos em sistemas colaborativos, interativos, multimidia, web e bancos de dados, Sociedade Brasileira de Computacao**, [S.l.], p.105, 2010.

TECHNOLOGY, M. I. of. **StarCluster at MIT**. Disponível em: <<http://star.mit.edu/cluster/about.html>>, acessado em Abril de 2016.

TECHNOLOGY, M. I. of. **StarCluster**. Disponível em: <<http://star.mit.edu/cluster/docs/latest/overview.html>>, acessado em Abril de 2016.

TONELLI, R. **AWS: introduction to virtual private cloud security**. Disponível em: <<http://blog.celingest.com/en/2013/04/19/aws-virtual-private-cloud-vpc-security/>>, acessado em Maio de 2016.

TOUT, S.; SVERDLIK, W.; LAWVER, G. Cloud computing and its security in higher education. **Proc ISECON, v26 (Washington DC)**, [S.l.], v.2314, 2009.