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

1 INTRODUÇÃO

Intro

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 para exploração e experimentação por parte dos alunos.

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 feita 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.

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.

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

2.1 Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2) faz parte do modelo 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 .. descrição? Tabela?
  - M4
  - M3
- Otimizadas para computação
  - C4

  - C3

- Otimizadas para memória
  - R3
- GPU
  - G2
- Otimizadas para armazenamento
  - I2
  - D2

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 Amazon Virtual Private Cloud

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.

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 infraesturura 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.

2.4 ??? 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 (SLAUGHTER, 2014) - 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 (CLUSTER, 2011) - 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.

- Gjovik University College - Erik Hjelmas - Bláblabla

3 DESENVOLVIMENTO

Resuminho das sessões

3.1 Ambiente de rede

A metodologia de desenvolvimento do projeto foi incremental, primeiramente focando em criar um ambiente base onde as intâncias 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 intâncias 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 intância 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 seguites:

- Packages - Provê suporte aos gerênciadores de instalação apt, msi, python, rpm, rubygems e yum.
- Files - Suporte para criação de arquivos e diretóríos, 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 instãncias, 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.

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

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

```
"AWS::CloudFormation::Init" : {
  "config" : {
```

```
    "files" : {
      "/etc/hosts" : {
        "content" : {
          "Fn::Join" : ["\n", [
          "127.0.0.1   localhost   localhost",
          {"Fn::FindInMap":["IpAddressConfig", "Node01", "IP" ]},"  node01
    node01",
          {"Fn::FindInMap":["IpAddressConfig", "Node02", "IP" ]},"  node02
    node02",
          {"Fn::FindInMap":["IpAddressConfig", "Node03", "IP" ]},"  node03
    node03",
          {"Fn::FindInMap":["IpAddressConfig", "Node04", "IP" ]},"  node04
    node04",
          {"Fn::FindInMap":["IpAddressConfig", "Node05", "IP" ]},"  node05
    node05"]]
        }
      }
    }
  }
}
```

3.2.2 Instalação de Pacotes

A imagem utilizada nas VMs já possui repositórios yum configurados, diponibilizando 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.

```
"AWS::CloudFormation::Init" : {
  "config" : {
    "commands" : {
      "exportOpenMPIBin" : {
        "command" : "echo \"export PATH=/usr/lib64/openmpi/bin:$PATH\" >> /
    home/ec2-user/.bashrc"
      },
      "exportOpenMPILib" : {
        "command" : "echo \"export LD_LIBRARY_PATH=/usr/lib64/openmpi/lib:
    $LD_LIBRARY_PATH\" >> /home/ec2-user/.bashrc"
      }
    }
  }
}
```

4 RESULTADOS

Funciona assim ó:

5 CONCLUSÃO

É isso champs.

REFERÊNCIAS

AMAZON CloudFormation. https://aws.amazon.com/pt/cloudformation/, acessado em Março de 2016.

AMAZON Elastic Compute Cloud. https://aws.amazon.com/pt/ec2/instance-types/, acessado em Abril de 2016.

AMAZON Machine Images. 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. https://aws.amazon.com/pt/education/awseducate/, acessado em Março de 2016.

CLUSTER, S. **StarCluster**. http://star.mit.edu/cluster/docs/latest/overview.html, acessoado em Abril de 2016.

EC2 Instance Pricing. https://aws.amazon.com/pt/ec2/pricing/, acessado em Abril de 2016.

SLAUGHTER, E. **CS-149 Setup**. https://bitbucket.org/elliottslaughter/cs149-setup, acessado em Março de 2016.