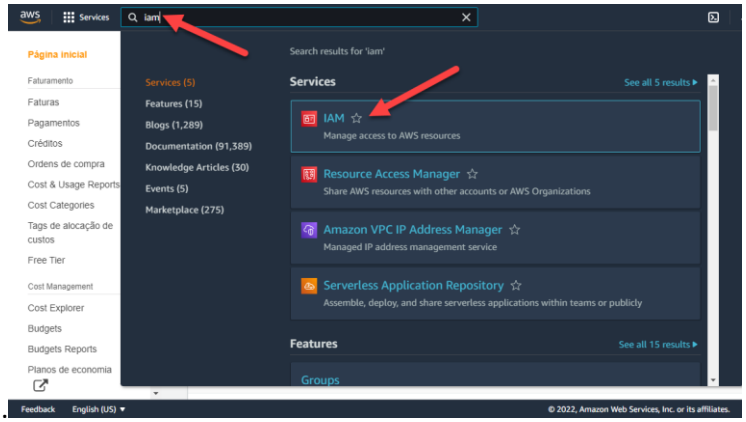
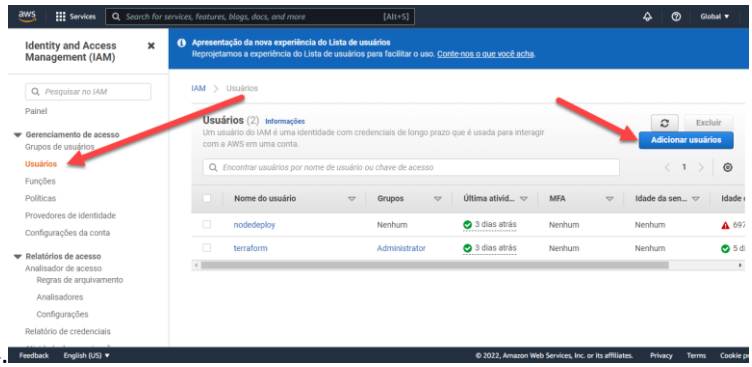
Implantação em Amazon Web Services (AWS) utilizado recurso EC2 de planta virtual utilizando node-red e supervisório utilizando Scada-LTS.

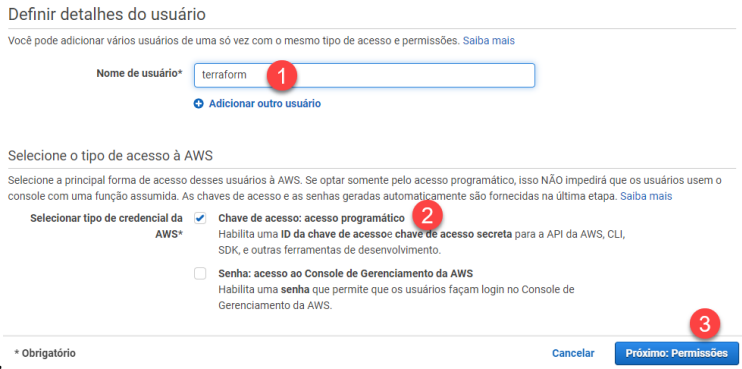
1. Pré-requisitos
   1.1. Tem conta no Amazon Web Services (AWS): https://aws.amazon.com
   1.2. Conhecimento básico em informática, git e execução de comandos shell.
2. Clonar o repositório no GitHub: https://github.com/rlmariz/virtual-lab-deploy.git
3. Após clonar o repositório a pasta de trabalho será “virtual-lab-deploy\aws”, todos os arquivos salvos e comandos executados devem ser nessa pasta.
4. Criar conta de acesso para o terraform no AWS
   4.1. Acesse o console da AWS e faça o login com sua conta e pesquise pelo produto IAM (Identity and Access Management).



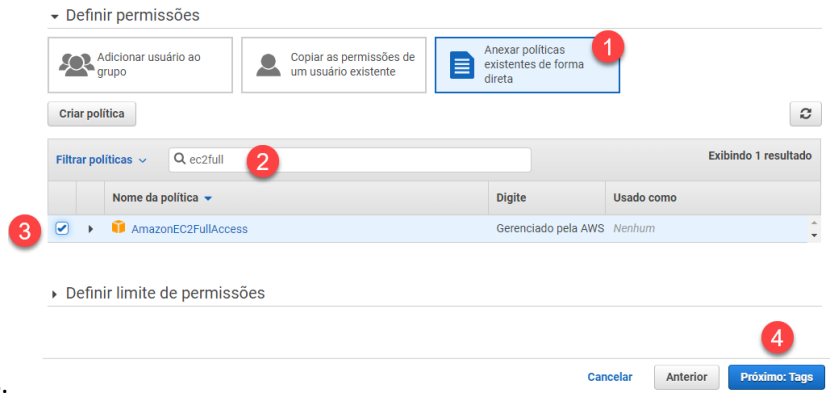
   4.2.
   4.3. Iremos criar um usuário para que o terraform possa interagir com a AWS, clique em USERS e em seguida em ADD USER.



   4.4.
   4.5. Definir detalhes do usuário.

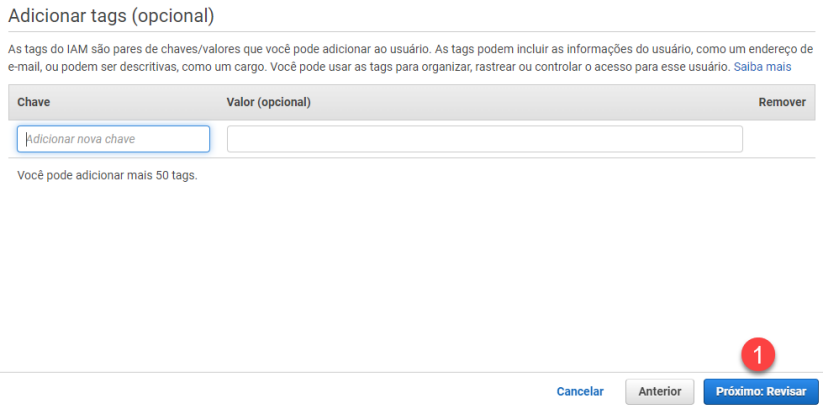


   4.6.
   4.7. Adicione a política AmazonEC2FullAccess ao usuário, o que dará permissão total ao usuário apenas a recursos da EC2, e clique em Next.



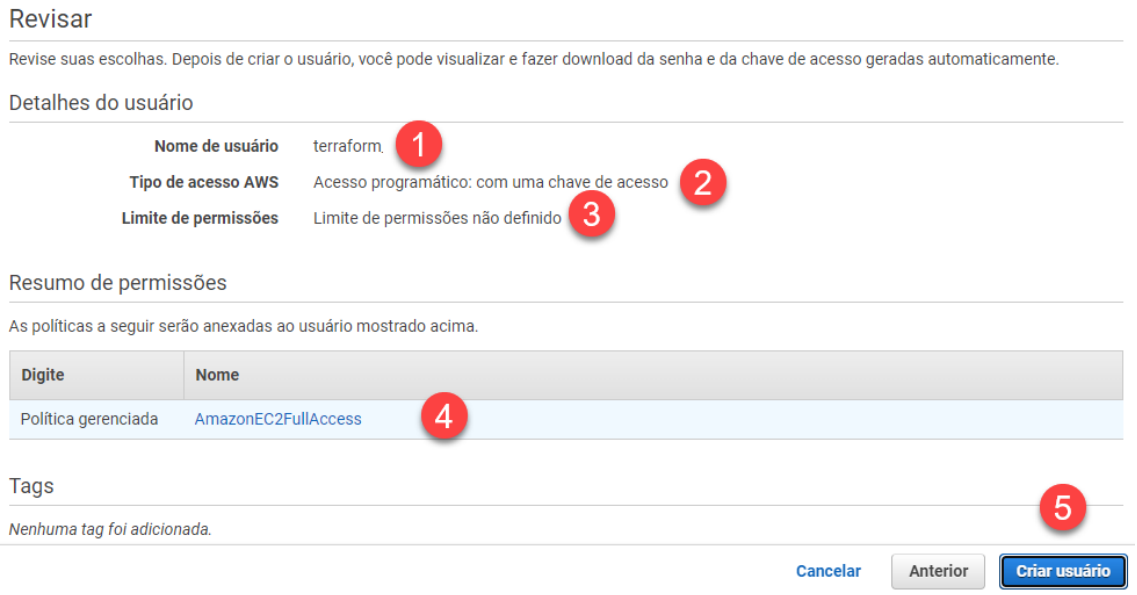
4.8.

4.9. Tags são utilizadas para adicionar informações relevantes ao usuario, clique em Next.



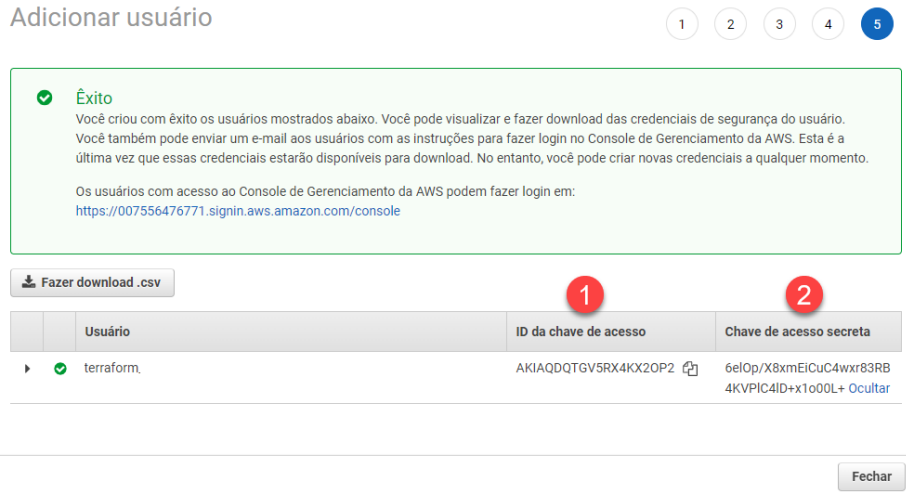
4.10.

4.11. Verifique os dados e clique em Create user.



4.12.

4.13. Clique em show e copie o Access key ID e Secret access key

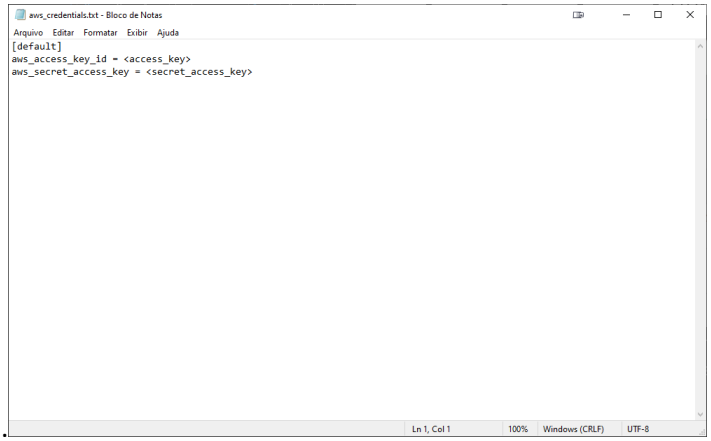


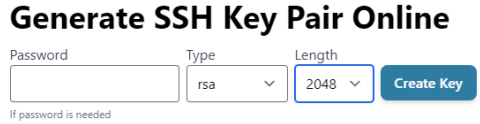
4.14.

Fechar

4.15. O usuário criado e chave de acesso não devem ser compartilhados, uma vez que quem tiver acesso a estes dados terá controle sobre os recursos adicionados como política, por questões de segurança este usuário não existe mais em minha conta.

5. Editar arquivo ***aws_credentials.txt*** e adicionar a chave de acesso e a chave secreta substituindo os valores <access_key> e <secret_access_key>.

5.1. 

6. Acessar o site e gerar par de chaves rsa que será utilizado para conexão ssh.
   6.1. Acesso o website https://www.wpoven.com/tools/create-ssh-key#
   6.2. Configure o type como rsa, length 2048, password deixe em branco e clique em create key.
   6.3. 

   6.4. Fazer download do ***Private Key*** e salvar arquivo com nome ***aws.key***.
   6.5. Fazer download do ***Public Key*** e salvar arquivo com nome ***aws.pub.key***.
7. Instalar o Terraform
   7.1. Acesse o site https://www.terraform.io/downloads
   7.2. Siga as instruções de acordo o sistema operacional que está utilizando
   7.3. Ao fazer o download do executável de preferência coloque na pasta de trabalho para facilitar sua utilização
   7.4. Pode ser feito o teste para verificar se está tudo ok executando no prompt de comandos: terraform –version
   7.5. Vai exibir a versão instalada e a plataforma: Terraform v1.1.7 on windows_amd64
8. Caso seja necessário pode se editar o arquivo variables.tf e fazer os ajustes necessários

8.1.
```
variable "region" {
  description = "Define what region the instance will be deployed"
  default     = "us-east-1"
}

variable "name" {
  description = "Name of the Application"
  default     = "virtuallab"
}

variable "env" {
  description = "Environment of the Application"
  default     = "prod"
}

variable "instance_type" {
  description = "AWS Instance type defines the hardware configuration of the machine"
  default     = "t2.micro"
}
```

9. Executar o comando terraform init

```
PS C:\temp\virtual-lab-deploy\aws> .\terraform.exe init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.5.0...
- Installed hashicorp/aws v4.5.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\temp\virtual-lab-deploy\aws>
```

9.1.

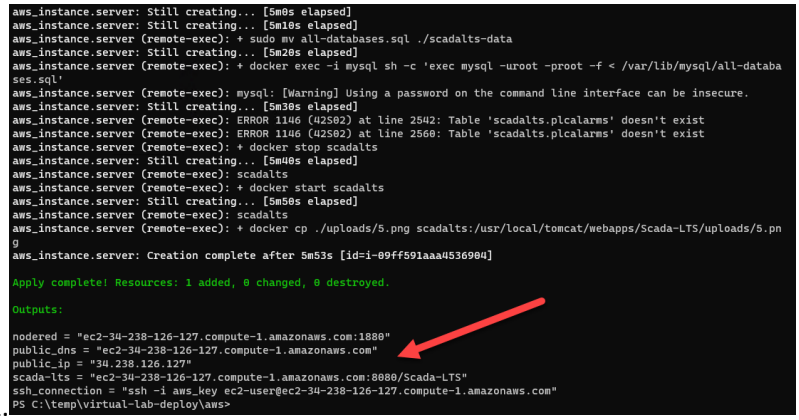
10. O próximo passo é criar infraestrutura e subir aplicação, vamos executar o comando: terraform apply
11. Vai ser exibido o plano de trabalho e estando tudo ok basta digitar “yes” e dar enter.

```
            + ipv6_cidr_blocks = []
            + prefix_list_ids  = []
            + protocol         = "tcp"
            + security_groups  = []
            + self             = false
            + to_port          = 22
          },
      ]
    + name                   = "allow_ssh"
    + name_prefix            = (known after apply)
    + owner_id               = (known after apply)
    + revoke_rules_on_delete = false
    + tags_all               = (known after apply)
    + vpc_id                 = (known after apply)
  }

Plan: 3 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + nodered        = (known after apply)
  + public_dns     = (known after apply)
  + public_ip      = (known after apply)
  + scada-lts      = (known after apply)
  + ssh_connection = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

11.1.

12. O processo leva certa de 6 minutos para ser implementado e ao final será exibido os endpoints para acesso ao supervisório, node-red e caso necessário o comando para acesso via ssh.

12.1.



13. O usuário e senha para acesso ao supervisório é admin/admin.
14. É muito importante desalocar os recursos após finalizar sua utilização para não ter custos extras, para fazer isso basta executar o comando: terraform destroy e confirmar com yes, ao final os recursos serão desalocado.

14.1.