

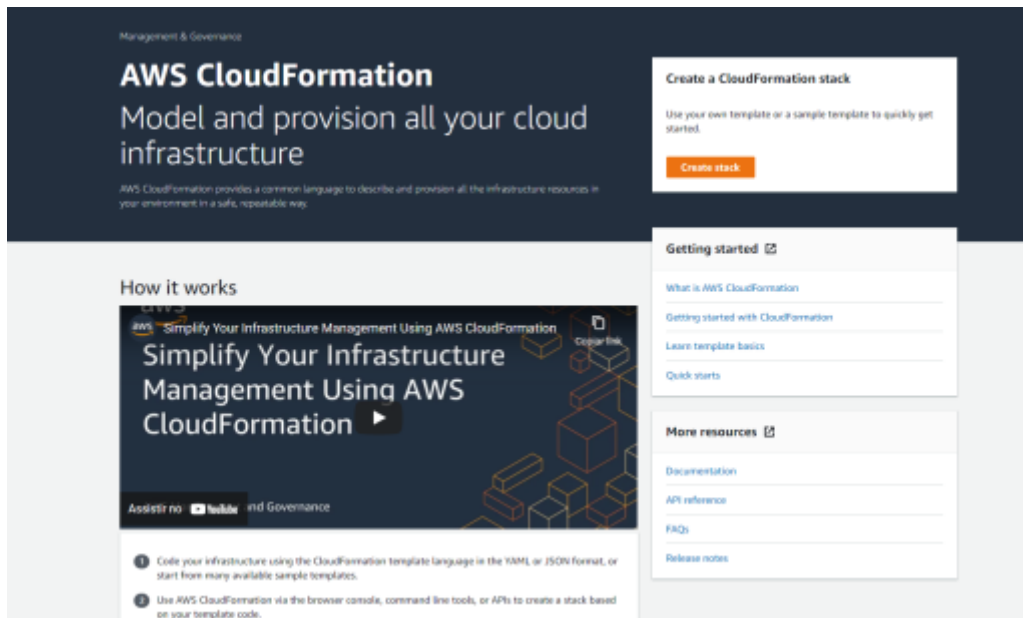
7 - CRIANDO UM SERVIDOR EC2 COM CLOUDFORMATION

Vamos criar o primeiro código para cloud formation.

- Vamos criar um servidor

- ec2 linux
- endereçamento público de ip
- t2.micro (free)
- Amazon linux

- Vamos alterar a região para OHIO, temos que mudar para essa região pois iremos utilizar no script parâmetros que se utilizarmos em outra região irá dar um erro (depois veremos esse erro).



Benefits and features

Stacks

A stack is a collection of AWS resources that you can manage as a single unit. All the resources in a stack are defined by the stack's AWS CloudFormation template

[View stacks](#)


StackSets

StackSets enables you to create, update, or delete stacks across multiple accounts and regions with a single operation

[View stacksets](#)

Change sets

Change sets allow you to preview how proposed changes to a stack might impact your running resources, making changes to your stack only when you decide

[Learn more](#) 

Drift detection


Run drift detection to identify configuration changes between your live resources and the template. Drifts will be detected on stacks and resources

[Learn more](#) 

Resource types

Over 500 resource types are supported by CloudFormation, covering over 100 AWS services

Template linting and policy-as-code

Check resource properties and values you describe in your templates against the [AWS CloudFormation resource specification](#) . Validate your templates for policy compliance against rules you create.

- Vamos criar um stack

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready
 ☐ Use a sample template
 ☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL
 ☐ Upload a template file

Amazon S3 URL

 Amazon S3 template URL
S3 URL: Will be generated when URL is provided

- Vamos criar o arquivo no editor para utilizar como template.

```

server-ec2.yml U X
C: > Users > GABI > PycharmProjects > UDEMY > CERTIFICAÇÃO AMAZON AWS SOLUTION ARCHITE
1  ---
2  Resources:
3    WebServers:
4      Type: AWS::EC2::Instance
5      Properties:
6        AvailabilityZone: us-east-2a
7        ImageId: ami-02bcbb802e03574ba
8        InstanceType: t2.micro
9

```

- Vamos entender como o código funciona;

Resources -> quais serão os recursos que estaremos aplicando no código.

1 - Damos um nome para o recurso (webservers) vamos criar servidores web, nesse local colocamos o nome que quisermos. Estamos falando que vamos criar um grupo que se chama webserver.

2 - Dentro desse grupo o TIPO(TYPE) será AWS máquinas EC2 instâncias.

- Precisa sempre ser separado por 2 pontos (:)

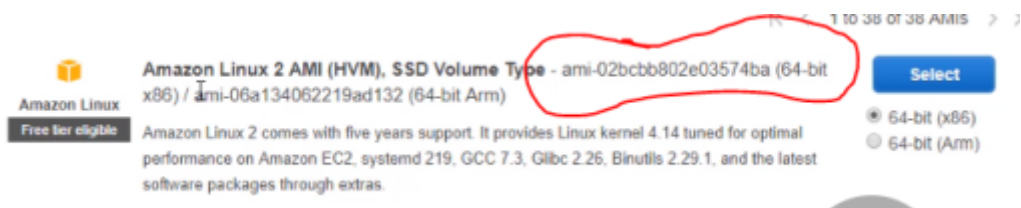
3 - Quais são as propriedades dessa máquina (properties):

- AVAILABILITYZONE _> Ela irá estar no data center (us-east-2a). Por isso colocamos OHIO no console da amazon.

- Se utilizarmos outra região, teremos que trocar isso.

Código	Nome
us-east-1	Leste dos EUA (Norte da Virgínia)
us-east-2	Leste dos EUA (Ohio)
us-west-1	Oeste dos EUA (Norte da Califórnia)
us-west-2	Oeste dos EUA (Oregon)
ca-central-1	Canadá (Central)
eu-central-1	UE (Frankfurt)
eu-west-1	UE (Irlanda)
eu-west-2	UE (Londres)
eu-west-3	UE (Paris)
eu-north-1	UE (Estocolmo)
ap-northeast-1	Ásia-Pacífico (Tóquio)
ap-northeast-2	Ásia-Pacífico (Seul)
ap-northeast-3	Ásia-Pacífico (Osaka – Local)
ap-southeast-1	Ásia-Pacífico (Cingapura)
ap-southeast-2	Ásia-Pacífico (Sydney)
ap-south-1	Ásia Pacífico (Mumbai)
sa-east-1	América do Sul (São Paulo)

- IMAGEID -> A imagem do servidor que queremos utilizar



- InstanceType -> Qual o tipo de instancia que iremos utilizar
instances - virtual servers that run aplciations.

-- APOS A CRIAÇÃO DO ARQUIVO , IREMOS FAZER O UPLOAD DELE NA ABA DE
CLOUDFORMATION CREATE STACK.

Create stack

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL ☒ Upload a template file

Upload a template file
Choose file server-ec2.yml
JSON or YAML formatted file

S3 URL: <https://s3.us-east-2.amazonaws.com/cf-templates-132e84tlcab3-us-east-2/2021157J39-server-ec2.yml> [View in Designer](#)

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

- Depois damos um nome = webserver-stack

- Temos agora uma infinidade de configurações, mas nesse laboratorio não precisamos alterar nada.

Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

Key	Value	Remove

[Add tag](#)

Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional
Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name [Remove](#)

Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

► **Stack policy**
Defines the resources that you want to protect from unintentional updates during a stack update.

► **Rollback configuration**
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

Review webserver-stack

Step 1: Specify template Edit

Template

Template URL
https://s3-us-east-2.amazonaws.com/cf-templates-132e84tlicab3-us-east-2/2021157j39-server-ec2.yml

Stack description
-

[Estimate cost](#)

Step 2: Specify stack details Edit

Parameters (0)

Key	Value
No parameters	

There are no parameters defined in your template

Step 3: Configure stack options Edit

Tags (0)

- Após isso clicamos em create stack

CloudFormation > Stacks > webserver-stack

webserver-stack Delete Update Stack actions Create stack

Stack info **Events** Resources Outputs Parameters Template Change sets

Events (1)

Timestamp	Logical ID	Status	Status reason
2021-06-06 12:50:31 UTC-0500	webserver-stack	CREATE_IN_PROGRESS	User initiated

- Ao irmos para o console EC2 vemos que a maquina esta sendo criada.

Welcome to the new Instances experience! We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the new EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

Instances (1) Filter instances Connect Instance state Actions Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	elastic IP
-	i-5a8fb17203aaf12	Running	t2.micro	Initializing	No alarms	us-east-2a	ec2-5-129-210-18.us-e...	5.129.210.10	-

- Quando voltamos a tela do cloudformation e atualizamos vemos que o sistema ja foi inicializado

webserver-stack

[Delete](#)[Update](#)[Stack actions](#)[Create stack](#)[Stack info](#)[Events](#)[Resources](#)[Outputs](#)[Parameters](#)[Template](#)[Change sets](#)

Events (5)



Timestamp	Logical ID	Status	Status reason
2021-06-06 12:30:52 UTC-0900	webserver-stack	CREATE_COMPLETE	-
2021-06-06 12:30:51 UTC-0900	WebServers	CREATE_COMPLETE	-
2021-06-06 12:30:56 UTC-0900	WebServers	CREATE_IN_PROGRESS	Resource creation initiated
2021-06-06 12:30:54 UTC-0900	WebServers	CREATE_IN_PROGRESS	-
2021-06-06 12:30:51 UTC-0900	webserver-stack	CREATE_IN_PROGRESS	User initiated