




AWS INFRA BUILDER

Uma linguagem desenvolvida para
auxiliar a criação de infraestrutura
de *Terraform* para **AWS**.



MOTIVAÇÃO

1.

Cloud está nos tópicos que
mais gosto de trabalhar e
onde pretendo me
desenvolver

Melhorar meu
conhecimento sobre as
tecnologias

2.

3.

Uma linguagem que facilite
a introdução de **novos**
Engenheiros a Cloud com
Terraform

CARACTERÍSTICAS

1. As variáveis de chaves privadas da conta da AWS devem ser declaradas como "**KEY_ID**" e "**SECRET_KEY**".
2. A linguagem trabalha majoritariamente com declaração de constantes. As variáveis aparecem na execução dos *loops* com **for**. Os valores podem ser **dígitos** ou **strings** (iniciadas e finalizadas com ["]).
3. Declaração de funções não permitem criar funções genéricas, mas configurar previamente algumas variáveis necessárias ou opcionais de entre uma de duas funções possíveis atualmente: **InstanceBuilder** e **SecurityGroupBuilder**. Essa predefinição é salva em um identificador.
4. A aplicação de estruturas condicionais se dá à criação de **alertas** no *CloudWatch* da **AWS**, onde a comparação é feita entre métricas disponíveis para uma instância e valores de *treshold*.

CARACTERÍSTICAS

5. Toda a execução de funções, *loops* e condicionais ocorrem dentro dos limites de "**START_BUILD**" e "**END_BUILD**".
6. Os *loops* ocorrem no espaço dos números naturais, começando no valor antes dos [:] e indo até o valor depois. Varia de 1 em 1, e os valores indicados devem ser **inteiros**.
7. A linguagem, ao ser compilada, cria a infraestrutura **Terraform** indicada, mas não a executa. Tanto o ***init*** quanto o ***apply*** devem ser executados pelo usuário dentro da pasta gerada após compilar.

EXEMPLOS

```
KEY_ID is "[YOUR_KEY_ID]"
SECRET_KEY is "[YOUR_SECRET_KEY]"
```





```
general_instance_description is "General Description"
```

```
function InstanceBuilder as instance_builder_type1 with
    instance_type as "t2.medium"
end
```

```
START_BUILD
    perform SecurityGroupBuilder with
        sg_name as "Teste SG" and
        sg_description as general_instance_description . " for SecGroup " and
        ingress_port as 22 and
        ingress_protocol as "tcp"
    end
    perform instance_builder_type1 with
        instance_name as "Teste 1"
    end
    for i in 1:(2*1) perform InstanceBuilder with
        instance_name as "Teste 2 (" . (i + 4) . ")"
    end
END_BUILD
```

Depois do *apply*...

INSTÂNCIAS

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	Public
<input type="checkbox"/>	Teste 2 (5)	i-03973e89c382f0626	Running 	t2.micro	 2/2 checks passed	No alarms +	us-east-1b	ec2-52
<input type="checkbox"/>	Teste 1	i-0ff753d0c487c92d7	Running 	t2.medium	 2/2 checks passed	No alarms +	us-east-1b	ec2-54

SECURITY GROUPS

<input type="checkbox"/>	-	sg-001e163a48e3a34ed	Teste SG	vpc-00936471ada724ab0 	General Description fo...	285415892922
--------------------------	---	--------------------------------------	----------	---	---------------------------	--------------

1.

Exemplo com **Instâncias**,
Security Groups, declaração
de função e *loop*

EXEMPLOS

```
KEY_ID is "[YOUR_KEY_ID]"
SECRET_KEY is "[YOUR_SECRET_KEY]"

treshhold_limit is 90
instance_1_id is "i-092990761db1e7ca1"

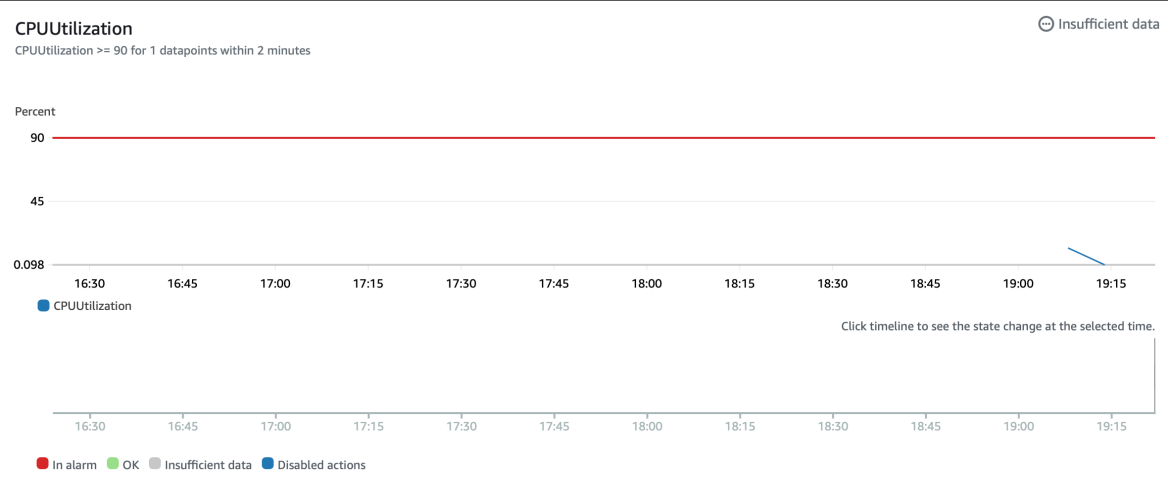
START_BUILD
  on CPUUtilization >= treshhold_limit in instance_1_id alert
END_BUILD
```

2.

Exemplo com **Alerta, condicional**, infraestrutura **já existente**

Depois do *apply...*

ALARME E CPUUTILIZATION



DESCRIÇÃO DO ALARME

Name alarm1	State ☹ Insufficient data	Namespace AWS/EC2	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold CPUUtilization >= 90 for 1 datapoints within 2 minutes	Metric name CPUUtilization	Missing data treatment Treat missing data as missing
Description No description	Last change 2023-06-12 19:21:59	Instanceld i-092990761db1e7ca1	Percentiles with low samples evaluate
	Actions No actions	Instance name Teste 2 (5)	ARN arn:aws:cloudwatch:us-east-1:285415892922:alarm:alarm1
		Statistic Average	
		Period 120	

CURIOSIDADES

1.

A linguagem inicialmente não foi idealizada como um auxílio para construção de infraestrutura para **Terraform**, mas conversando com os professores e outros alunos, achei uma motivação pertinente e um espaço de atuação adequado para valorizar o que foi produzido.

2.

O CloudWatch quando configurado é um sistema de alertas muito interessante e pode ser utilizado junto de outras ferramentas de segurança, como **Zabbix** ou **Wazuh**, para aprimorar o monitoramento da infraestrutura como um todo. Pode ser também aplicado para sistemas de **Load Balancer** e **Auto Scalling**, tanto pela dashboard quanto por *Terraform*.

3.

A linguagem foi criada com o intuito de ser um MVP para a ideia de traduzir o código *Terraform* para uma linguagem mais **reconhecível** pelos alunos. Por ser criada pensando nesse cenário, **expandi-la** para se adequar a outras estruturas e receber outros argumentos é trivial.