UDACITY

Return to Classroom

Deploy a High-Availability Web App using CloudFormation

REVIEW

CODE REVIEW 2

HISTORY

Meets Specifications

Congratulations !!

You did an excellent work considering it was your first attempt. Some of the points that I liked about your submission :

- *You have kept a decent number of parameters which makes the stacks reusable. Having too many or too less parameters can make it complex or not reusable.* 👏
- *You have correctly displayed the* `LoadBalancer DNS URL` *in the output section* 👏
- *Logical separation of the stacks* 👏

This project introduced to you the concept of **Infrastructure as a Code (IaaC)** - which is one of the critical concepts when it comes to creating and configuring the infrastructure on a Cloud Platform.
I really liked your **your overall coding of the infrastructure - parameterizing the stack and showing the relevant information using the Outputs section** 👏

You completed one of the important milestones in your journey as Cloud DevOps student. All the hard work did pay off and you have passed with flying colors. 🌈

I hope this project gave some good practical learning. Take a small break and prepare for the next battle.

Keep it Up !! Wishing you Best of Luck for the journey ahead :)
We look forward to receiving your future project submissions soon .

Additional Resources 💡:

Here are some of the resources which would good to go through and would be useful for next project :

- *This following article explains CI/CD in detail - Continuous Integration Continuous Deployment (CI/CD)*
- *Official Ansible Documentation - A Configuration management tool .*

PS: If you have any doubts regarding any of the concept, feel free to search or post a question on Knowledge Portal where many of the fellow students and mentors may have faced the same situation before and would have provided the appropriate steps to resolve it.

Have a Good Day and Stay Safe ✌

Keep Learning and Stay Udacious

The Basics

The more the better, but an exaggerated number of parameters can be messy (say, 10 or more). 1 or 0 is definitely lacking.

> *Great work setting the appropriate number of parameters (not too few and not too many) for your CloudFormation stacks ✓*

Although Parameter section is totally optional, it is recommended to have some parameters in your template because it helps you in the following ways:

- *Enables you to input custom values to your template each time you create or update a stack*
- *Helps in avoiding frequent changes to the code itself*

Additional Reading Resource 💡

- *Here is the - Official AWS Documentation on Parameters section of a CloudFormation Template*

This is the mandatory section of the script, we are looking for a LoadBalancer, Launch Configuration, AutoScaling group a health check, security groups and a Listener and Target Group.

> ***All of the following resources are including in the script ✔***
>
> - *LoadBalancer ✓*
> - *Launch Configuration ✓*
> - *AutoScaling Group ✓*
> - *Security Groups ✓*
> - *Listener and Target Group ✓*

This is optional, but it would be nice to have a URL here with the Load Balancer DNS Name and “http” in front of it .

> ***Output section is present ✓***
>
> *It's great to see that you not only gave the LoadBalancer DNS name as output, but also appended "http" in front of it* 👏🏻

Usually the variables declared in the Output section are used for the following purposes:

- *Import into other stacks (to create cross-stack references - what we did in this project)*
- *Return in response (to describe stack calls)*
- *View on the AWS CloudFormation console. (what we did in this project)*

Additional Reading Resource 💡

- *Here is the - Official AWS Documentation on Output section of a CloudFormation Template*

If the student provides a URL to verify his work is running properly, it will be a page that says "it works! Udagram, Udacity"

1. The LoadBalancer DNS URL is provided ✓

2. The application is working perfectly ✓

```
WebAppLaunchConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash
        apt-get update -y
        apt-get install apache2 -y
        systemctl start apache2.service
        cd /var/www/html
        echo "Udacity Demo Web Server Up and Running!" > index.html
    ImageId: ami-0729e439b6769d6ab
    SecurityGroups:
    - Ref: WebServerSecGroup
    InstanceType: t3.medium
    BlockDeviceMappings:
    - DeviceName: "/dev/sdk"
      Ebs:
        VolumeSize: '10'
```

NOTE

We strongly recommend you to `delete the stack` once this rubric point has been passed by a reviewer `to avoid recurring charges`

Load Balancer

The auto-scaling group needs to have a property that associates it with a target group. The Load Balancer will have a Listener rule associated with the same target group

1. The AutoScaling has a property that associates it with a target group ✓

2. Load Balancer has a Listener rule associated with the same target group✓

Port 80 should be used in Security groups, health checks and listeners associated with the load balancer

Port 80 is used in the following places:

- *Load Balancer Security Group ✓*
- *Listeners ✓*
- *Health Checks of Target Groups✓*

Auto-Scaling

Students should be using PRIV-NET (private subnets) for their auto-scaling instances

Suggestion 💡

It would have bene good if you had provided the `Architecture Diagram` *for your infrastructure too*

The machine should have 10 GB or more of disk and should be a t3.small or better.

1. Machine has 10GB or more disk size ✓

2. Instance type is t3.small or better. Good to see that you have used t3.medium ✓

```
WebAppLaunchConfig:
  Type: AWS::AutoScaling::LaunchConfiguration
  Properties:
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash
        apt-get update -y
        apt-get install apache2 -y
        systemctl start apache2.service
        cd /var/www/html
        echo "Udacity Demo Web Server Up and Running!" > index.html
    ImageId: ami-0729e439b6769d6ab
    SecurityGroups:
    - Ref: WebServerSecGroup
    InstanceType: t3.medium
    BlockDeviceMappings:
    - DeviceName: "/dev/sdk"
      Ebs:
        VolumeSize: '10'
```

There shouldn't be a 'keyname' property in the launch config

> ***`keyname` property has been removed from the LaunchConfiguration*** ✓
>
> *We removed the `keyname` property because we didn't want any outside access to our instance*

Additional Reading Resource 💡

- LaunchConfiguration is a template which defines the desired configurations of an EC2 instance. Once we define this, it is used by the AutoScaling Group when it has to launch new EC2 instances.
- *Here is the - Official AWS Documentation on Launch configurations*

Bonus

Any values in the output section are a bonus

As mentioned previously, the output section displays the LoadBalancer DNS name correctly and we were able to access the application as well ✓

Any resource of type AWS::EC2::Instance, optional, but nice to have.

This part is optional and was not implemented ✓

DOWNLOAD PROJECT

2 CODE REVIEW COMMENTS ›

RETURN TO PATH

Rate this project

START