Yash Mutatkar **Date: 4 Nov 2024**

Automating Infrastructure Deployment with AWS CloudFormation

In this guide, I will walk you through my process of deploying, updating, and deleting AWS resources using CloudFormation. The focus will be on automating the provisioning of an S3 bucket and an EC2 instance while using parameters to customize the resource configurations.

Prerequisites

Before getting started, I made sure I had the following:

- **AWS CLI**: The AWS Command Line Interface was installed and configured with the necessary IAM permissions to access CloudFormation, S3, and EC2.
- **CloudFormation Template**: I prepared a YAML file that defines the infrastructure resources I wanted to create—specifically, an S3 bucket and an EC2 instance.

Steps

1. Deploying the Initial CloudFormation Stack

To kick things off, I deployed my CloudFormation stack. This stack included an S3 bucket and an EC2 instance, with the EC2 instance type specified through parameters. I ran the following command:

```
aws cloudformation create-stack --stack-name yash-stack --template-body
file://cloudformation.yaml --parameters
ParameterKey=InstanceType,ParameterValue=t2.micro
```

- **--stack-name**: I named my stack `yash-stack` for easy identification.
- **--template-body**: This option pointed to my local CloudFormation template file (`cloudformation.yaml`), which described the resources I wanted to create.
- **--parameters**: I set the `InstanceType` parameter to `t2.micro`, overriding the default values in the template.

Once I executed the command, I received an output with the `StackId`, confirming that CloudFormation had initiated the stack creation.

2. Verifying Stack Deployment

To check the status of my stack creation, I used the following command:

```
aws cloudformation describe-stacks --stack-name yash-stack
```

I looked for `StackStatus: CREATE_COMPLETE` in the output to confirm that the deployment was successful. If there were any errors, the output provided details that helped me troubleshoot.

3. Updating the CloudFormation Stack

After successfully deploying the initial stack, I wanted to make some changes. For example, I decided to add a tag to the EC2 instance to identify it as an "Updated EC2 Instance."

Once I modified the `cloudformation.yaml` file, I ran the `update-stack` command:

```
aws cloudformation update-stack --stack-name yash-stack --template-body
file://cloudformation.yaml --parameters
ParameterKey=InstanceType,ParameterValue=t2.micro
```

- **--update-stack**: This command applies the changes from my modified template to the existing stack.
- **--parameters**: I re-specified parameters to confirm or adjust the configuration settings as needed.

To monitor the progress of the update, I used the `describe-stacks` command again. I looked for `StackStatus: UPDATE_COMPLETE` to ensure that the updates were applied successfully. If any issues arose, I checked the specific event logs with `describe-stack-events`.

4. Deleting the Stack

Finally, to clean up the resources I had created, I decided to delete the stack. I used the following command:

```
aws cloudformation delete-stack --stack-name yash-stack
```

This command removed the stack and all resources created within it.

To verify the deletion, I ran the `describe-stacks` command once more. Once the stack deletion was complete, `yash-stack` no longer appeared in the output, confirming that all resources were cleaned up.