**Objective:** Create a CloudFormation template to deploy an S3 bucket with versioning enabled.

**Steps:**

1. **Create a YAML or JSON Template:** Decide whether you want to work with YAML or JSON for your CloudFormation template. YAML tends to be more human-readable.
2. **Define the Resources:** Define the resources you want to create. In this case, it's an S3 bucket with versioning enabled.
3. **Add Parameters (Optional):** If you want to make your template more flexible, you can add parameters. For example, you might want to allow users to specify the bucket name.
4. **Write the Template:** Here's an example YAML template to create an S3 bucket with versioning enabled:

yamlCopy code

AWSTemplateFormatVersion: '2010-09-09' Resources: MyS3Bucket: Type: AWS::S3::Bucket Properties: BucketName: my-bucket-name VersioningConfiguration: Status: Enabled

1. **Save the Template:** Save your template to a file with a **.yaml** or **.json** extension.
2. **Deploy the Stack:** Using the AWS Management Console, AWS CLI, or SDKs, deploy your CloudFormation stack using the template you created.

If you're using the AWS CLI, you can use the following command:

bashCopy code

aws cloudformation create-stack --stack-name my-s3-stack --template-body file://path/to/your/template.yaml

1. **Verify:** After the stack is created, verify that the S3 bucket was created with versioning enabled by checking the AWS Management Console or using AWS CLI commands like **aws s3api get-bucket-versioning**.
2. **Clean Up:** Once you're done experimenting, don't forget to delete the CloudFormation stack to avoid incurring any unnecessary costs.

That's it! This exercise should give you a good starting point for working with AWS CloudFormation. As you become more familiar with CloudFormation, you can explore more advanced features and templates.

Top of Form

Bottom of Form