AWS CLI CloudFormation

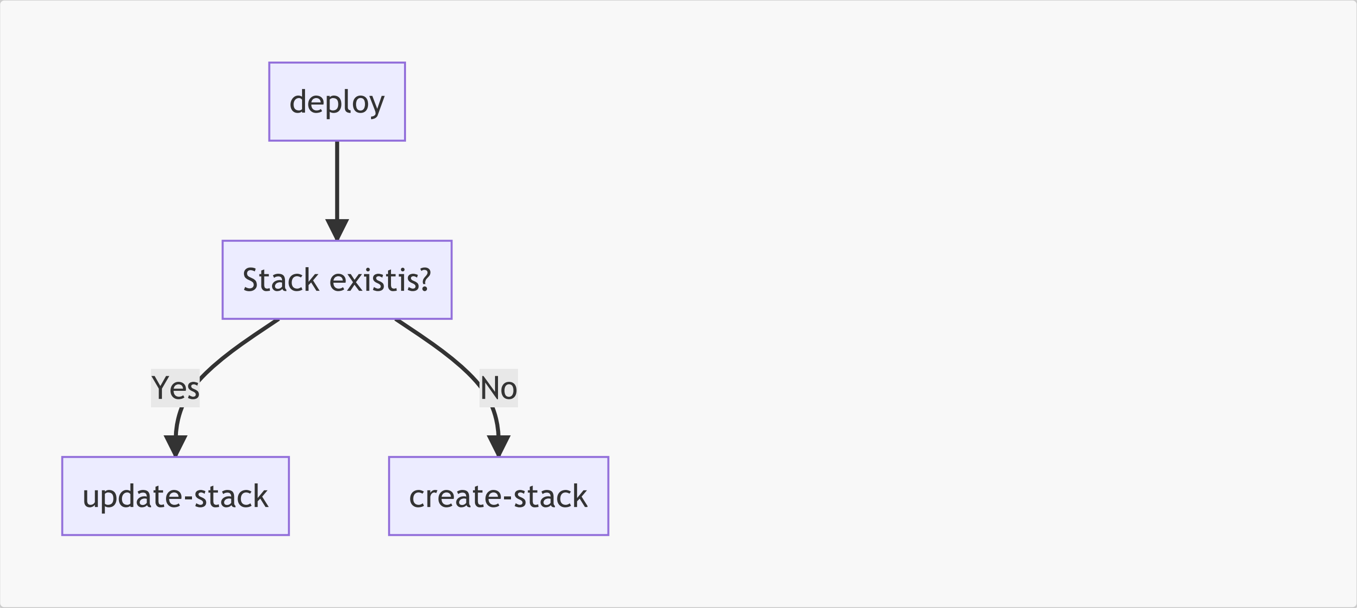
Describing and Listing Your Stacks

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE

aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query
StackSummaries[0]

aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query
StackSummaries[0].[StackName,StackStatus]

aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query
StackSummaries[0] --output yaml

aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query
StackSummaries[0] --output json

aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE --query
StackSummaries[0].[StackName,StackStatus] --output json|text|tab
```

Creating a Stack

```
aws cloudformation create-stack --stack-name myteststack --template-body
file:///home/testuser/mytemplate.yml --parameters
ParameterKey=Parm1,ParameterValue=test1 ParameterKey=Parm2,ParameterValue=test2

aws cloudformation create-stack --stack-name s3-01c-cli-stack --template-body
'file://01c-S3.yml' --parameters ParameterKey=BucketNameParam,ParameterValue=test-
cf-cli-bucket ParameterKey=EnvironmentName,ParameterValue=test
```

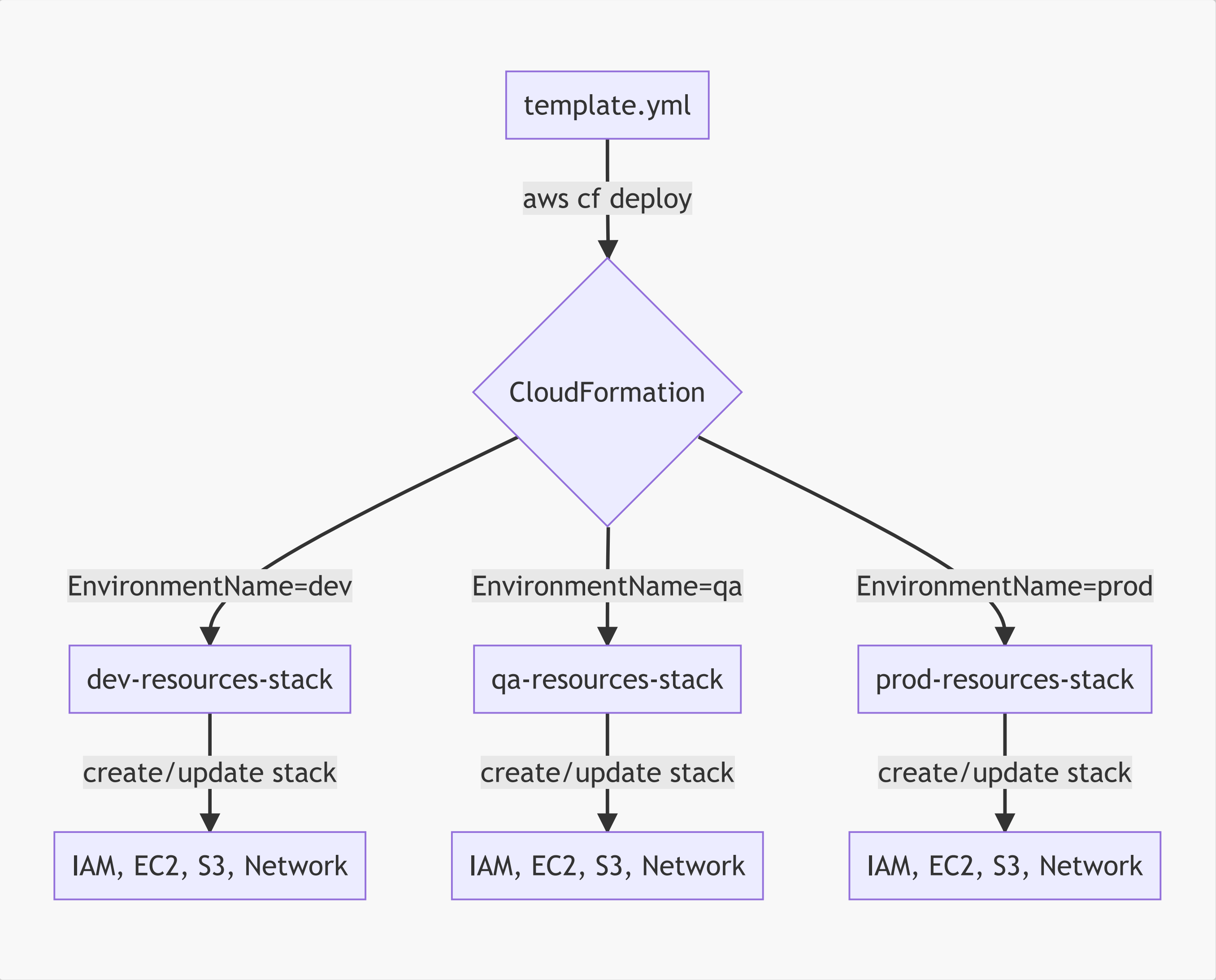
Deploy a Stack using CLI



- `deploy` : check if stack exists, if stack exists -> update the stack, if stack is not present, it will create stack.

```
aws cloudformation deploy --template-file cf-template.yaml --stack-name
mystackname --capabilities CAPABILITY_IAM

aws cloudformation deploy --stack-name s3-01c-cli-git-stack-deploy --template-file
'01c-S3.yml' --parameter-overrides BucketNameParam=test-deploy-cf-cli-git-bucket
EnvironmentName=test

aws cloudformation deploy --template-file /path_to_template/template.yml --stack-
name my-new-stack --parameter-overrides Key1=Value1 Key2=Value2 --tags Key1=Value1
Key2=Value2

# Stack creation
aws cloudformation deploy --stack-name testclicheck-s3-01c-cli-git-stack-deploy --
template-file '01c-S3.yml' --parameter-overrides BucketNameParam=test-deploy-cf-
cli-git-bucket EnvironmentName=testclicheck

OR

aws cloudformation deploy --template-file 04a-IAM-EC2-Role.yml --stack-name dev-
iam-ec2-stack --parameter-overrides KeyName=test-cf-key EnvironmentName=dev --
capabilities CAPABILITY_NAMED_IAM

# Validate the stack in Console
# Make slight modifications to the above mentioned same template file.
# Run the above deploy command again.
# Here, CF cli deploy command will check for stack with same name and update
existing stack with the new template file.

# Here in above command, there is only one environment is created with Environment
name as dev.
# Same Template will be used to create another environment using qa prefix.
aws cloudformation deploy --template-file 04a-IAM-EC2-Role.yml --stack-name qa-
```

```
iam-ec2-stack --parameter-overrides KeyName=test-cf-key EnvironmentName=qa --
capabilities CAPABILITY_NAMED_IAM
```



Viewing Stack Event History

```
aws cloudformation describe-stack-events --stack-name s3-01c-cli-stack
```

Listing Resources

```
aws cloudformation list-stack-resources --stack-name s3-01c-cli-stack
aws cloudformation list-stack-resources --stack-name VPC-03a-stack --query
StackResourceSummaries[*].[PhysicalResourceId,ResourceType]
```

Validating a Template

- In S3

```
aws cloudformation validate-template --template-url
https://s3.amazonaws.com/cloudformation-templates-us-east-1/S3_Bucket.template
```

- In Local

```
aws cloudformation validate-template --template-body file://sampletemplate.yml

# Using a loop to validate all templates in a directory
for i in $(ls | grep -i '.yml'); do echo "$i"; aws cloudformation validate-
template --template-body file://./$i; done;
```

Deleting a Stack

```
aws cloudformation delete-stack --stack-name prod1-ec2-stack
```

Stack Drift

Detect CloudFormation Stack Drift

- Enter the stack name and stack drift id.

```
aws cloudformation detect-stack-drift --stack-name <CF_STACK_NAME>

aws cloudformation detect-stack-drift --stack-name VPC-03a-stack
# {
#     "StackDriftDetectionId": "ed933f30-469e-11ec-a4e0-0a0575ffedcb"
# }
```

Describe CloudFormation Stack Drift

```
aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-
id <STACK_DRIFT_ID>

aws cloudformation describe-stack-drift-detection-status --stack-drift-detection-
id ed933f30-469e-11ec-a4e0-0a0575ffedcb
```

View Stack Resources Drift

- When the stack drift detection operation is complete, use the describe-stack-resource-drifts command to review the results, including actual and expected property values for resources that have drifted.

```
aws cloudformation describe-stack-resource-drifts --stack-name VPC-03a-stackVPC-03a-stack
```

CF CLI Script Scenario

- Write a shell script to detect stack drift for all stacks present in your aws account. Schedule this script in cron to get list of drifted stack daily at 10:00am. Output should be in csv file with this name `DD-MM-YYYY_CF_DRIFT_LIST.csv`
- Headers for CSV File can be:
  - Stack Name
  - Stack Status
  - CreatedTime
  - DriftStatus