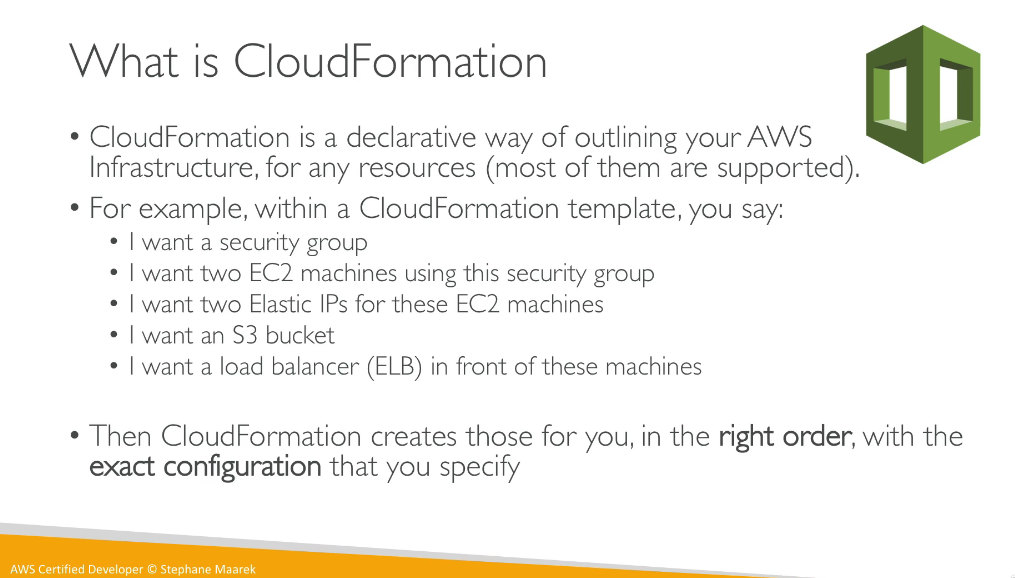
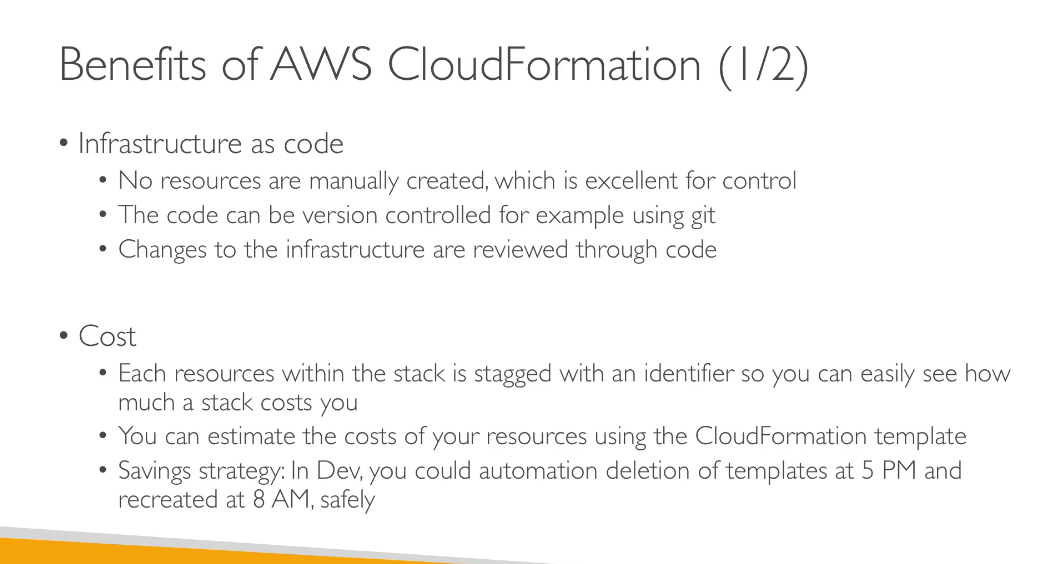
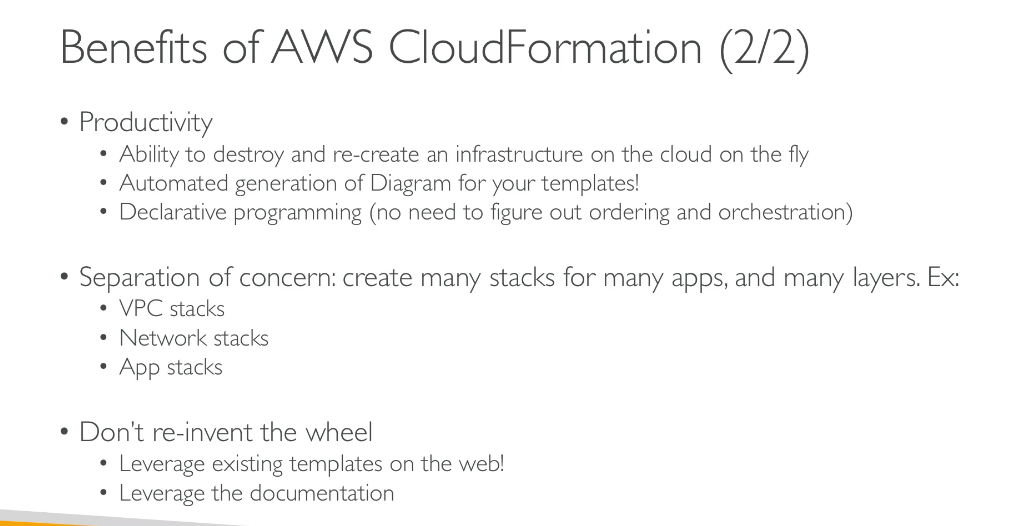
**CLOUDFORMATION**

Cloudformation is a declarative way of outlining your AWS insfrastructure for any resourcs.thus it iis known as infrastructure a scode.



**Benefits of cloudformation**

****

****

**How cloudformation works**

**Graphical user interface, text, application, email

Description automatically generated**

Templates are uploaded in the s3 bucket and then referenced by the cloudformation and every time some updation is to be made the existing cloudformation cannot be edited a new templet is to be generated and the cloudformation will itself check what changes are in between the two and how a new version can be updated.

All the stacks in a cloud formation are identified by names a d adeletinga stack deletes every single artifact that was created by the cloudformation.

**Deplying the cloudformation templates**

**Graphical user interface, text, application, email

Description automatically generated**

**Building blocks**

**Text

Description automatically generated**

**INTRODUCTORY EXAMPLE FOR HANDS ON WITH CLOUDFORMATION**

Some important points to be noted while discussing about the cloudformation

1.While creatinga cloudformation we need to specify a templte which can b uploaded to an s3 and then can be used from there while creating.

2.While creating the stack the cloudformation will automatically know the order in which the resources will be created.

3.When we update a cloudformation it will know whether to delete the existing resource or not or to keep it an djust update it.

4.When w edelete a cloudformation all the infrastructure created by it will be deleted.

**Yaml**

**Graphical user interface, text, application

Description automatically generated**

**Everything in yaml is akey value pair as can be seen.**

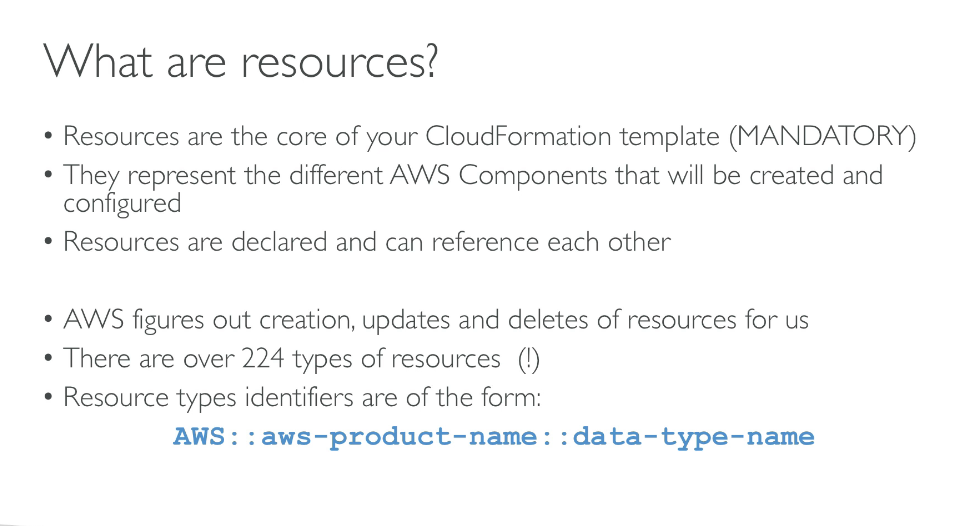
There can be nested objects in yaml as can be seen fro bill-to

There is a support for ararys a s can be seen that the product object is an array and it is denoted by a (-) sign.

There can be multilines strings as well as can be seen in lines: |

So this denotes a multiline string

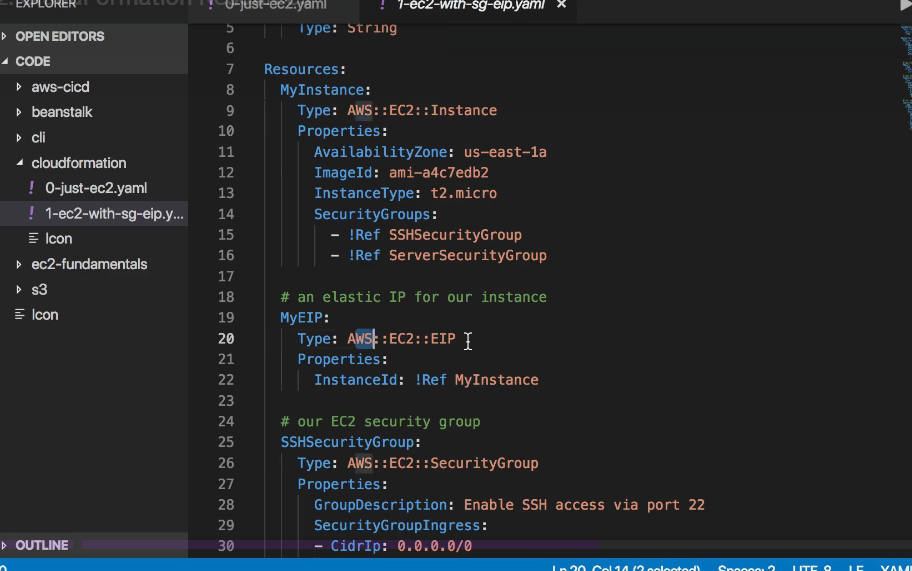
**CLOUDFORMATION RESOURCES**

****

**Graphical user interface, text, application, email

Description automatically generated**

**Aw** has the full documentation for us to write our code.we can reference it and write the code and as in what manner needed.



**WHAT ARE PARAMETERS?**

Parameters are a way to provide inputs to your aws cloudformation template.

They are important to know about if:

You want to reuse your templates across company

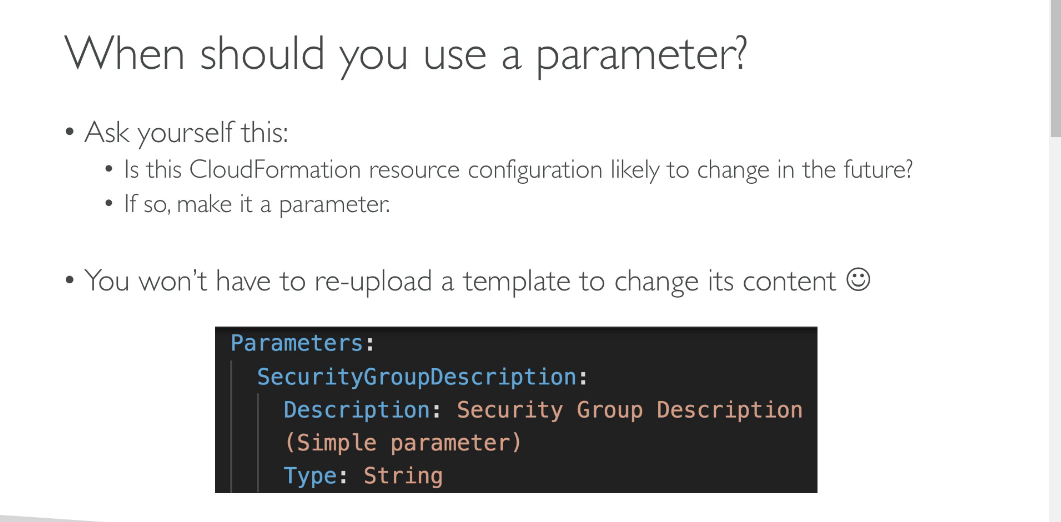
Some inputs cannot be determined ahead of time.

Parameters are extremely powerful,controlled,and can prevent errors from happening in your template.

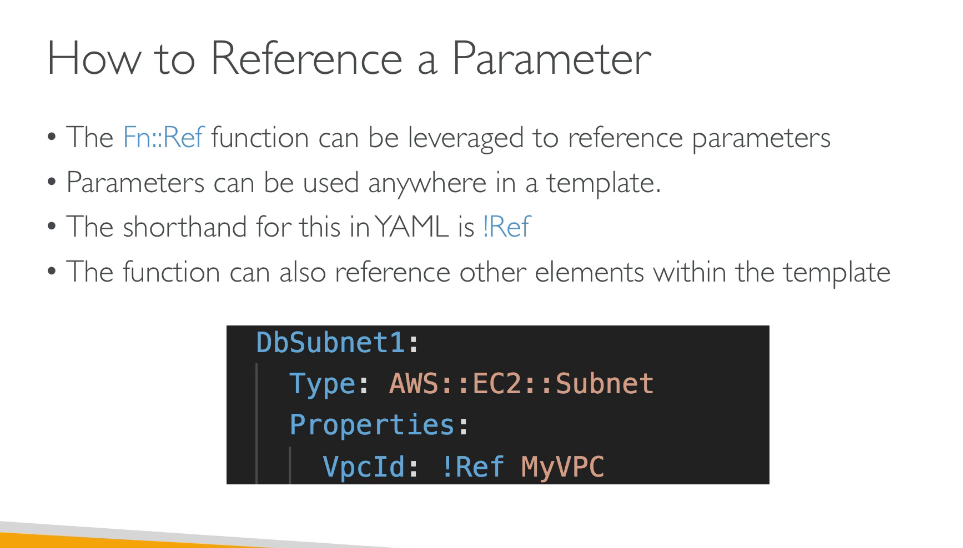
Thanks to types.

We generally need the parameters in case if that configuration is likely to change in the future.and if so make it a parameter.

You wont have to re-upload a template to change its content.

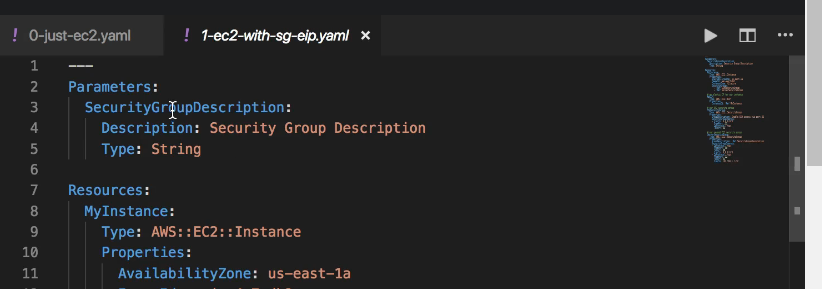






**How it works at the code level?**

Suppose in our code we have a parameetrs section which will define the paramentes.

****

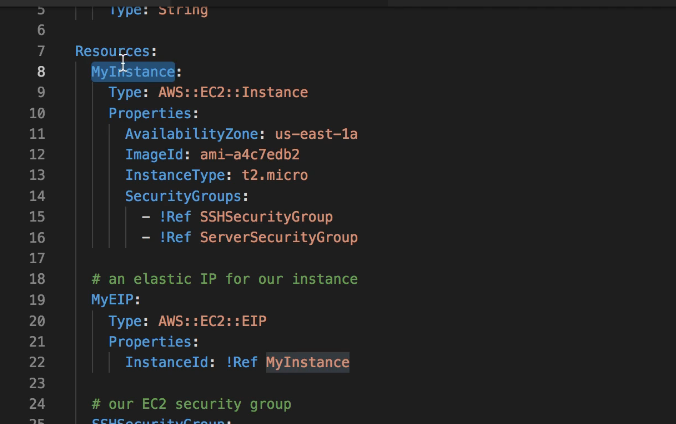
That will be used below as can be seen.

Text

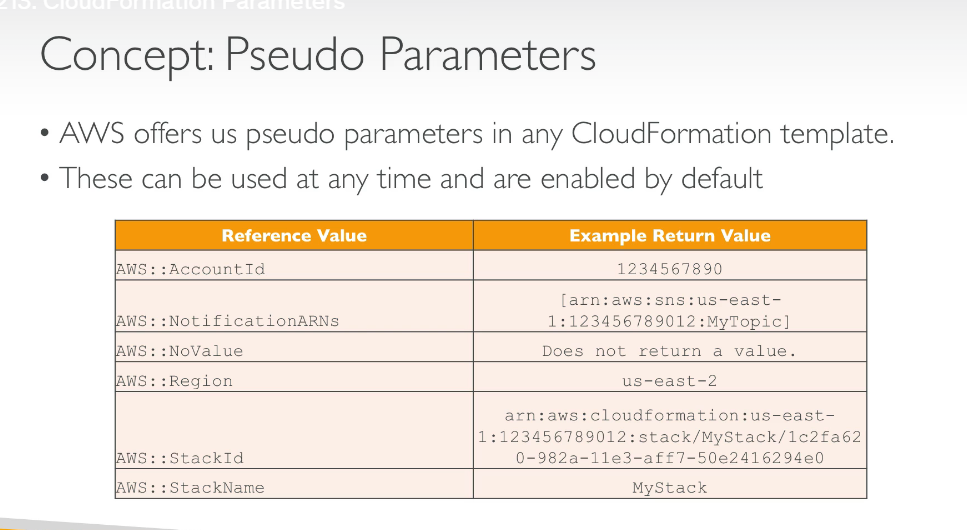
Description automatically generated

Check thelone no 39 which is using that parameter value.

**We can also use the reference as can be seen below:**

****

MyInstance being refferd at line no 22



There is also a concept of pseudo parameters.

AWS offers pseudo paameters in any cloudformation template.these can be used at any time and are enabled by default.

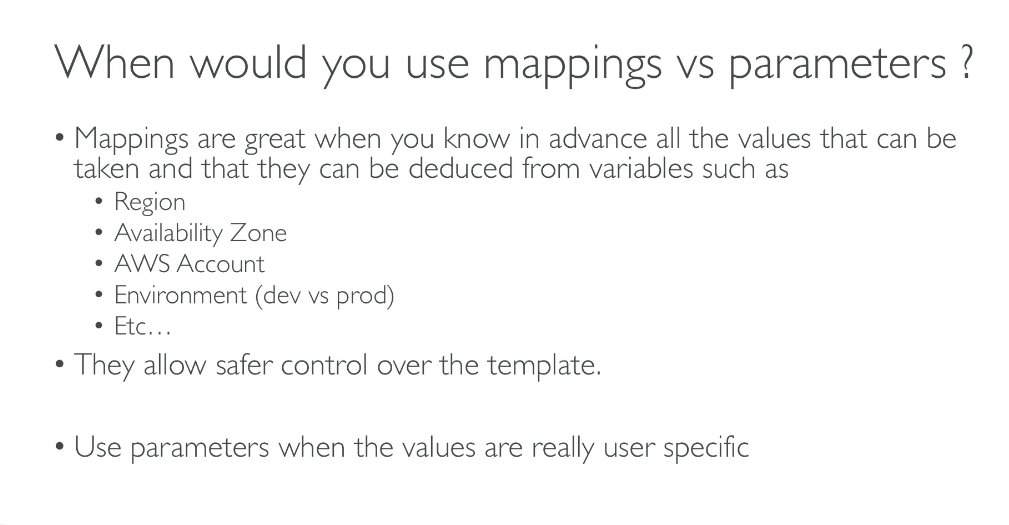
**WHAT ARE MAPPINGS?**

Mappings are fixed variables within your cloudformation template.they are very handy to differentiate between environments(dev vs prod),regions(AWS regions),AMI typesetc.

All the values are hardcoded with the template.

Example can be seen below:



**When would you use mappings vs parameters?  
**

**How do we access the mapping values**

**Graphical user interface, text, application

Description automatically generated**

To take a value of a mapping value we need to specify in the way as shown above.

!FindInMao[RegionMap,!Ref “AWS::Region”<which is a pseudo parameter>,32]

**OUTPUTS**

**Text, letter

Description automatically generated**

**OUTPUTS EXAMPLE**

**Text

Description automatically generated**

We create a ssh security group as a part of one template.We can create an output that references that security group.

So from the above example we can understand that this ssh security group can be exported with the name of sshsecuritygroup.

**How do we import the output that was exported?**

**Text

Description automatically generated**

**CLOUDFORMATION CONDITIONS**

Conditions are used to control the creation of resources or outputs based on a condition.

Conditions can be whatever you want them to be ,but the common conditions can be the environment,aws region,,any parameter value.

Graphical user interface, text, application, email

Description automatically generated

Conditions are used to control the creation of resources or outputa based on a condition.

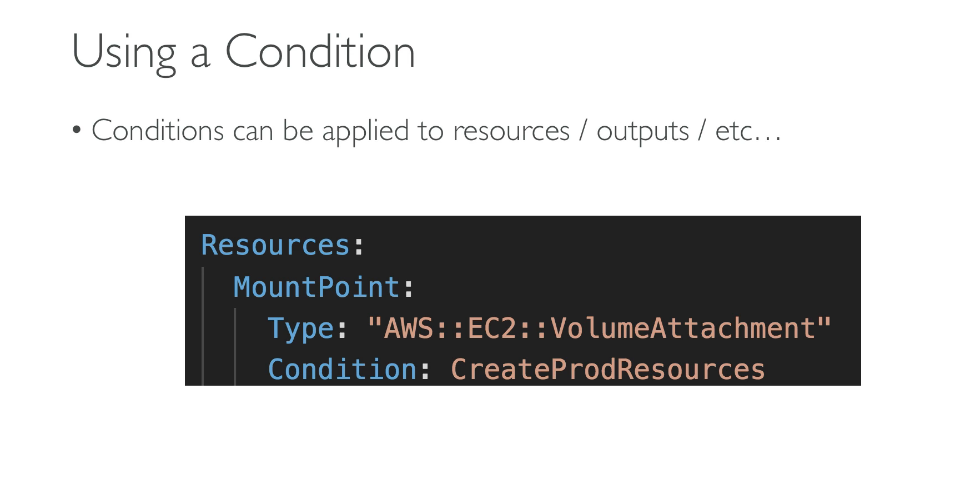
You can set a condition for a particular environment or a region.Each condition can reference another condition,parameter value or mapping.

So a condition can be created a sis shown below:

Graphical user interface, text, application

Description automatically generated

If the above condition comes out to be true then the new resource as shown in below statement will be created.



**MUST KNOW INTRINSIC FUNCTIONSGraphical user interface, text, application, email

Description automatically generated**

**Text

Description automatically generated**



Graphical user interface, application

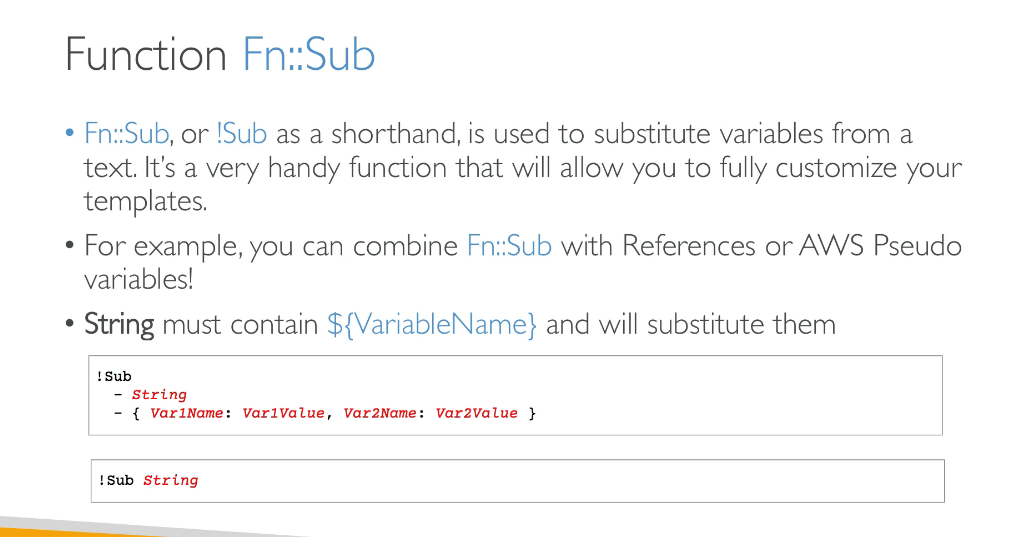
Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

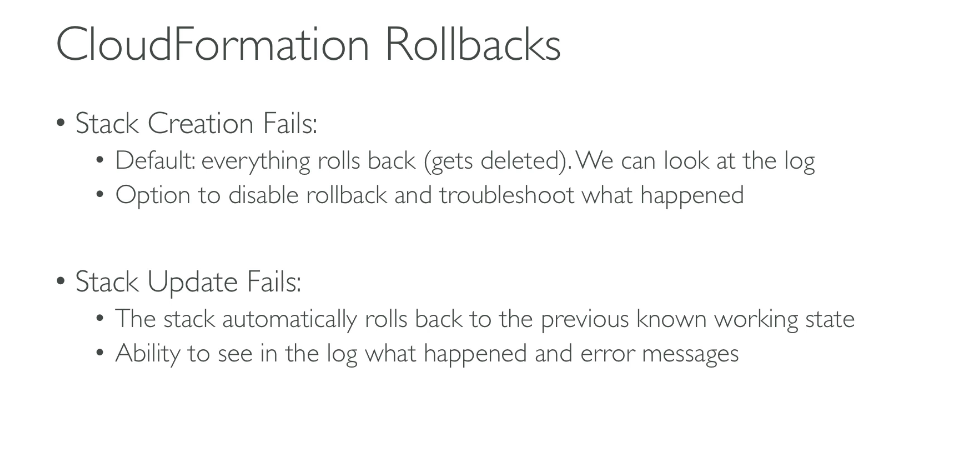
Description automatically generated



**ConditionsGraphical user interface, text, application

Description automatically generated**

**CLOUDFORMATION ROLLBACKS**

****