**CLOUDFORMATION**

Parameter types enable CloudFormation to validate inputs earlier in the stack creation process.

CloudFormation currently supports the following parameter types:

String – A literal string

Number – An integer or float

List<Number> – An array of integers or floats

CommaDelimitedList – An array of literal strings that are separated by commas

AWS::EC2::KeyPair::KeyName – An Amazon EC2 key pair name

AWS::EC2::SecurityGroup::Id – A security group ID

AWS::EC2::Subnet::Id – A subnet ID

AWS::EC2::VPC::Id – A VPC ID

List<AWS::EC2::VPC::Id> – An array of VPC IDs

List<AWS::EC2::SecurityGroup::Id> – An array of security group IDs

List<AWS::EC2::Subnet::Id> – An array of subnet IDs

Conditions cannot be used within the Parameters section. After you define all your conditions, you can associate them with resources and resource properties only in the Resources and Outputs sections of a template.

**Stack B, then Stack C, then Stack A**

All of the imports must be removed before you can delete the exporting stack or modify the output value. In this case, you must delete either Stack B or Stack C, then you delete Stack A.

Q: What happens when one of the resources in a stack cannot be created successfully?

By default, the “automatic rollback on error” feature is enabled. This will direct CloudFormation to only create or update all resources in your stack if all individual operations succeed. If they do not, CloudFormation reverts the stack to the last known stable configuration. This is useful when, for example, you accidentally exceed your default limit of Elastic IP addresses, or you don’t have access to an EC2 AMI that you’re trying to run. This feature enables you to rely on the fact that stacks are created either fully or not at all, which simplifies system administration and layered solutions built on top of CloudFormation.

Q: Can stack creation wait for my application to start up?

Yes. One of the options CloudFormation provides is a *WaitCondition* resource that acts as a barrier, blocking the creation of other resources until a completion signal is received from an external source such as your application or management system. Other options include creating custom logic with AWS Lambda functions.

Q: Can I save my data when a stack is deleted?

Yes. CloudFormation allows you to define deletion policies for resources in the template. You can specify that snapshots be created for Amazon EBS volumes or Amazon RDS database instances before they are deleted. You can also specify that a resource should be preserved and not deleted when the stack is deleted. This is useful for preserving Amazon S3 buckets when the stack is deleted.

Q: Can I update my stack after it has been created?

Yes. You can use CloudFormation to modify and update the resources in your existing stacks in a controlled and predictable way. By using templates to manage your stack changes, you have the ability to apply version control to your AWS infrastructure just as you do with the software running on it.

Q: Can I manage resources created outside of CloudFormation?

Yes! With Resource Import, you can bring an existing resource into AWS CloudFormation management using [resource import](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resource-import.html).

Q: Are there limits to the size of description fields?

Template, Parameter, Output, and Resource description fields are limited to 4096 characters.

Q: Are there limits to the number of parameters or outputs in a template?

You can include up to 60 parameters and 60 outputs in a template.

Q: Are there limits to the number of resources that can be created in a stack?

Currently, you can create up to 200 resources per stack. Creating smaller templates and stacks and modularizing your application across multiple stacks is a best practice to minimize blast radius for your resource changes, and to troubleshoot issues with multiple resource dependencies faster, since smaller groups of resources will have less complex dependencies than larger groups.