



A large, abstract cluster of white and light gray geometric shapes, resembling a 3D wireframe or a collection of facets, occupies the left side of the frame. The shapes are various sizes and angles, creating a sense of depth and complexity.

CloudFormation

What is CloudFormation

- ❖ Declarative way of outlining your AWS Infrastructure, for many resources(most of them are supported)
- ❖ No modification, reupload is only option
- ❖ For example within your template you can provide your resources as below and created in same order:
 - ❖ I want a security group
 - ❖ 2 instances
 - ❖ 2 elastic Ips.

Benefits

- ❖ Infrastructure as a code
 - ❖ No manual creation, which is excellent for control
 - ❖ Template code can be version controlled like using Git.
 - ❖ Changes to infrastructure are reviewed through code.
- ❖ Cost
 - ❖ All resources created are tagged using an identifier so can easily see how much they cost.
 - ❖ You can estimate the costs of resources using the template
 - ❖ Savings strategy: In Dev, we can automate deletion of templates at a scheduled time and reschedule at a different time for creation.
 - ❖ Cost of Cloudformation is free but not the resources it creates.

Benefits (continued..)

- ❖ Productivity
 - ❖ Automated generation of Diagram of your templates.
 - ❖ Declarative programming (no need to worry about finding out ordering and orchestration)
- ❖ Separation of concern: create many stacks for many apps, and many layers.Ex:
 - ❖ VPC stacks.
 - ❖ Network stacks
 - ❖ App stacks

Environment setup

Windows 10 installation (https://www.youtube.com/watch?v=DM2DS_952s4)

VSCode (<https://code.visualstudio.com/>)

Extension: CloudFormation Linter

- ◊ Description: AWS CloudFormation template Linter
- ◊ Author: kddejong
- ◊ URI: kddejong.vscode-cfn-lint
- ◊ Marketplace: [CloudFormation Linter - Visual Studio Marketplace](#)

Extra settings for plugin: pip install cfn-lint pydot

Alt.Environment setup

Atom editor <https://atom.io>

unzip and add that to applications folder in mac OS.

On mac: packages: help - search - install/update packages

on windows: file-settings-packages -install

atom-cform -> auto completion of cloud formation and other code snippets

atom-cform-yaml -> for yaml

atom-cfn-lint → for syntax highlighting and errors/warnings

new ~ demo.json

start<enter> ->create cf skeleton

Introductory Example

- ❖ Simple EC2 instance
- ❖ Security group to ec2 instance
- ❖ Updating the templates will show the change history in change sets.

<https://codeshare.io/K8bMBK>

Resources:

MyInstance:

Type: AWS::EC2::Instance

Properties:

AvailabilityZone: us-east-1a

ImageId: ami-0742b4e673072066f

InstanceType: t2.micro

Update Behavior

- ❖ Update resources based on what you submit and stack's current template, which method to use depends on which property you update for a resource
 - ❖ Update with no interruption
 - ❖ Without disrupting resources & without changing physical ID
 - ❖ Ex: updating the IAM instance profile (IAMInstanceProfile) of an EC2 instance
 - ❖ Update with some interruption
 - ❖ Ex: updating the ec2 instance type. Say from t2.micro to t2.large
 - ❖ Replacement
 - ❖ Recreating the resource with new physical ID
 - ❖ Creates new resource, changes references from other resources and deletes the old resource

Sample: Updating a S3 bucket

- ❖ Two cases:
 - ❖ Updates with no interruption (adding AccessControl)
 - ❖ Replacement Updates (updating the name of the bucket)

Template components

- ❖ AWSTemplateFormatVersion: identifies the capabilities of the template
- ❖ Description: comments about the template
- ❖ Transform: Specifies one or more Macros that are used to process the template
- ❖ Metadata
- ❖ Resources: your AWS resources declared in the template(MANDATORY)
- ❖ Parameters: the dynamic inputs for your template
- ❖ Mappings: the static variables for your template
- ❖ Outputs: References to what has been created
- ❖ Conditionals: List of conditions to perform resource creation
- ❖ Rules: validate a parameter(s) during stack creation/update
- ❖ Template helpers: References and functions

Parameters

- ❖ A way to provide inputs to your Cloudformation template
- ❖ Helpful for reusability of templates and dynamic inputs that can't be determined ahead of time.
- ❖ Can be cross-validated using Rules.

Parameter Settings

- ❖ Parameters can be controlled by all these settings:
 - ❖ Type: (<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>)
 - ❖ String
 - ❖ Number
 - ❖ CommaDelimitedList
 - ❖ List<Number>
 - ❖ AWS-Specific Parameter (to help catch invalid values – match against existing values in the AWS account)
 - ❖ List<AWS-Specific-Parameter>
 - ❖ SSM Parameter (get parameter value from SSM Parameter store)

Parameter Settings(continued)

- ❖ Parameters can be controlled by all these settings (continued):
 - ❖ Description
 - ❖ ConstraintDescription (String)
 - ❖ Min/MaxLength
 - ❖ Min/MaxValue
 - ❖ Default
 - ❖ AllowedValues(array)
 - ❖ AllowedPattern(regex)
 - ❖ NoEcho (Boolean)

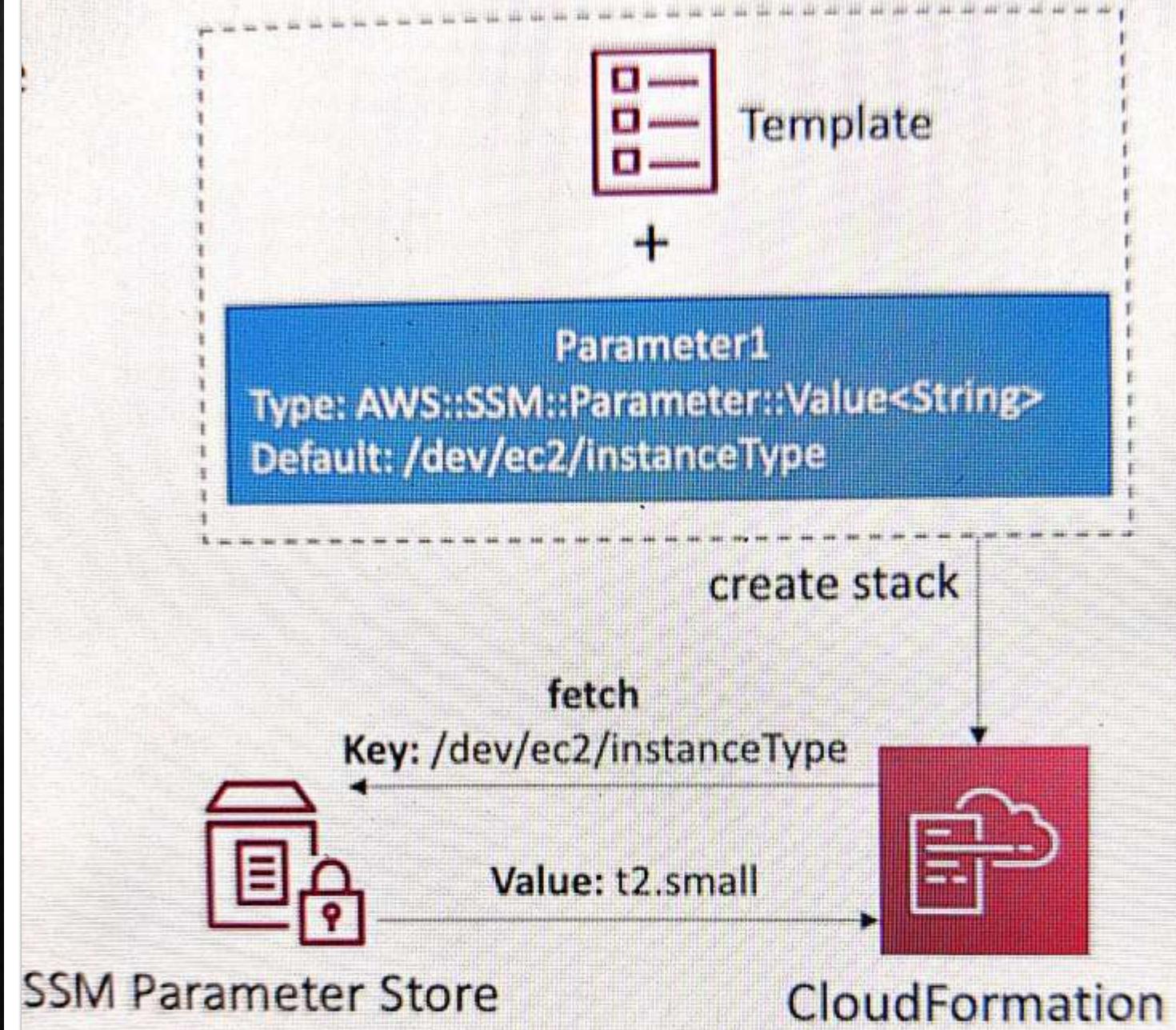
How to Reference a Parameter

- ❖ The Fn::Ref function can be used to reference parameters.
- ❖ The shorthand for this in YAML is !Ref
- ❖ This function can also reference other elements within the template
- ❖ Parameters can be used anywhere in a template, except:
 - ❖ AWSTemplateFormatVersion
 - ❖ Description
 - ❖ Transform
 - ❖ Mappings

Template Options

- ❖ Common parameters to any CloudFormation template(seen in webconsole stack options)
 - ❖ Tags
 - ❖ Permissions
 - ❖ Notification Options
 - ❖ Timeout
 - ❖ Rollback Failure
 - ❖ Rollback Configuration
 - ❖ Stack Policy
 - ❖ Termination Protection
 - ❖ Quick-create Link

How SSM Parameter works



One more sample

- ❖ Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

Parameter starting with /aws are public parameters provided by AWS, so select under public parameters of AWS Systems Manager, Parameter store.

SSM Parameter Type

- ❖ Type: AWS::SSM::Parameter::Value<String>
Default: /dev/ec2/instanceType
- ❖ Reference parameters in Systems Manager Parameter Store
- ❖ Specify SSM parameter key as the value
- ❖ CF always fetches the latest value, you can't specify parameter version
- ❖ CF doesn't store Secure String values
- ❖ Validation done on SSM parameter keys, but not values
- ❖ Supported SSM Parameter Types:
 - ❖ AWS::SSM::Parameter::Name Value<String> Value<List<String>> Value<AWS-Specific-Parameter>
 - ❖ AWS::SSM::Parameter::Value<List<AWS-Specific-Parameter>>

Deployment

- ❖ Manual using designer or code editor and upload them
- ❖ Automated using aws cli or using continuos delivery tool

Cost estimate of template

- ❖ Estimate cost link in the Review step while deploying through web console

AWS resources

- ❖ Reference of resources and types:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-template-resource-type-ref.html>

- ❖ EC2 instance :

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-instance.html>