Moving up the stack with Stacktician

Amogh Vasekar
amoghvk@gmail.com

LINUX FOUNDATION

Agenda

- Introduction
- History
- Architecture and Implementation
- Current State
- Enhancements

Agenda

- Introduction
  - History
  - Architecture and Implementation
  - Current State
  - Enhancements

Introduction

What is Amazon CloudFormation?

- Create and manage collection of related AWS resources
- Typically, the resources will form your whole application “stack”

In essence, “Infrastructure-as-code”

- JSON file
- Describe resources
- Describe dependencies
- Readable, reusable, reviewable

Sample JSON template

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a
  beautiful website or blog. This template installs a single-instance WordPress deployment using a local MySQL database to store the data.
  It demonstrates using the AWS CloudFormation bootstrap scripts to install packages and files at instance launch time. **WARNING** This
  template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "255",
      "AllowedPattern" : "[\\x20-\\x7E]*",
      "ConstraintDescription" : "can contain only ASCII characters."
    },

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro","m1.small","m1.medium","m1.large","m1.xlarge","m2.xlarge","m2.2xlarge","m2.4xlarge","m3.xlarge","m3.
      2xlarge","c1.medium","c1.xlarge","cc1.4xlarge","cc2.8xlarge","cg1.4xlarge"],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },
  "Resources" : {

    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0"},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "SSHLocation"}}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" : [ "WebServer", "PublicDnsName" ]}, "/wordpress"]] },
      "Description" : "WordPress Website"
    }
  }
}
```

Usefulness

- Single click application deployment for multiple users
- Transparent, reusable
- Ease of resource management and dependencies
- Atomic deployments
- Thus, migrate new business applications to “cloud”

stack*tician

Simply put, CloudFormation for CloudStack ☺

Web interface for template and stack management

cloud STACK COLLABORATION CONFERENCE

LINUX FOUNDATION

Agenda

- Introduction
- **History**
- Architecture and Implementation
- Current State
- Enhancements

History

- Started by @chiradeep
- Introduced at CCC-13 as a prototype
- Execute AWS CloudFormation templates on CloudStack *AS IS*
- Problem – low fidelity between AWS and CloudStack resource types (more on this later)

Agenda

- Introduction
- History
- **Architecture and Implementation**
- Current State
- Enhancements

Architecture

Two components

StackMate

- Execution engine for running the templates and interfacing with CloudStack

Stacktician

- Web interface
- Embeds StackMate underneath

StackMate in 2 mins

- Treat template = stack definition as workflow
- Execute workflow by making appropriate CloudStack API calls
- Core engine for executing templates, does the heavy lifting
- Command line tool

Stacktician in 2 mins

- Web UI for StackMate
- Backed by database for persistence
- Uses Ruby meta-programming to interface with StackMate

Where we were last time…

- Resolve complex resource dependencies
- Intrinsic functions
- Wait Conditions
- Create simple resources like AWS::Instance and AWS::SecurityGroup using StackMate

AWS templates on CloudStack

- Simple resources work well
- Too many differences in more complex resources like VPC and networks
  - No route tables
- Semantic differences between APIs for same resources
  - Different types of values for parameters
- Dependency resolution for CloudStack using AWS templates became too complex
  - Sometimes, two AWS APIs act as a single one in CloudStack
  - CloudStack networking renders AWS options like DhcpOptions unnecessary
  - Too many NoOp resources used as work-around for above

CloudStack Namespace

- Instead create CloudStack specific namespace!
- No more mapping of resources from AWS namespace to CloudStack space
- Adds the ability to validate API parameters
- Every CloudStack resource becomes a “participant” in the workflow execution
- Essentially need to call create and delete APIs

CloudStack Namespace

- Problems
  - New resources added frequently (e.g tags, UCS, Nicira)
  - APIs evolve to support new parameters etc.
  - Too many CloudStack resources to maintain (~51)

- Solution
  - Use API discovery
  - Auto generate code
  - Use API information to validate all required parameters exist / format of parameters

StackMate participants

- Bootstrap StackMate
- Publish CloudStack version specific gem

```
Amoghs:stackmate amoghvasekar$ rake -T
rake generate:docs[apis]          # Generate JSON Docs
rake generate:participants[apis]  # Generate participants for CloudStack
Amoghs:stackmate amoghvasekar$ rake generate:participants utils/apis_user.json
/Users/amoghvasekar/.rvm/rubies/ruby-2.0.0-p247/bin/ruby utils/apis.rb utils/apis_user.json
Amoghs:stackmate amoghvasekar$
```

```
Amoghs:stackmate amoghvasekar$ ls lib/stackmate/participants/
cloudstack.rb                              cloudstack_iso.rb                          cloudstack_securitygroupegress.rb          cloudstack_vmsnapshot.rb
cloudstack_affinitygroup.rb                cloudstack_lbhealthcheckpolicy.rb          cloudstack_securitygroupingress.rb         cloudstack_volume.rb
cloudstack_autoscalepolicy.rb              cloudstack_lbstickinesspolicy.rb           cloudstack_snapshot.rb                     cloudstack_volumeops.rb
cloudstack_autoscalevmgroup.rb             cloudstack_loadbalancer.rb                 cloudstack_snapshotpolicy.rb               cloudstack_vpc.rb
cloudstack_autoscalevmprofile.rb           cloudstack_loadbalancerrule.rb             cloudstack_sshkeypair.rb                   cloudstack_vpnconnection.rb
cloudstack_condition.rb                    cloudstack_network.rb                      cloudstack_staticnat.rb                    cloudstack_vpncustomergateway.rb
cloudstack_egressfirewallrule.rb           cloudstack_networkacl.rb                   cloudstack_staticroute.rb                  cloudstack_vpngateway.rb
cloudstack_firewallrule.rb                 cloudstack_networkacllist.rb               cloudstack_tags.rb                         cloudstack_vpnuser.rb
cloudstack_globalloadbalancerrule.rb       cloudstack_nictovirtualmachine.rb          cloudstack_template.rb                     common.rb
cloudstack_instancegroup.rb                cloudstack_portforwardingrule.rb           cloudstack_togloballoadbalancerrule.rb     stacknest.rb
cloudstack_ipaddress.rb                    cloudstack_project.rb                      cloudstack_toloadbalancerrule.rb
cloudstack_ipforwardingrule.rb             cloudstack_remoteaccessvpn.rb              cloudstack_virtualmachine.rb
cloudstack_iptonic.rb                      cloudstack_securitygroup.rb                cloudstack_virtualmachineops.rb
```

Errors and rollback

- Want stack creation to be atomic
- But, provide an option to user to disable rollback (e.g. for debugging)
  - Catch resource creation errors, and initiate rollback
  - Execute deletion in reverse order of dependencies
  - Skip any resources not yet created
  - If error while deleting, notify user (log for StackMate, UI status for Stacktician)
  - Stacktician injects additional code in StackMate for DB management

Metadata server

- Resources, like VMs, can launch additional initialization scripts upon configuration
  - E.g. – Launch a script via “userdata” parameter
- Typically, script needs additional metadata to work with
- Metadata is specific to a resource

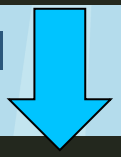
```
"UserData" : { "Fn::Base64" : { "Fn::Join" : ["", [
      "#!/bin/bash -v\n",
      "yum update -y aws-cfn-bootstrap\n",


      "# Helper function\n",
      "function error_exit\n",
      "{\n",
      "  /opt/aws/bin/cfn-signal -e 0 -r \"$1\" '", { "Ref" : "WaitHandle" }, "'\n",
      "  exit 1\n",
      "}\n",

      "# Install Apache Web Server, MySQL, PHP and Drupal\n",
      "/opt/aws/bin/cfn-init -s ", { "Ref" : "StackMate::StackId" }, " -r WebServer ",
        " || error_exit 'Failed to run cfn-init'\n",

      "# Setup MySQL root password and create a user\n",
      "mysqladmin -u root password '", { "Ref" : "DBRootPassword" }, "' || error_exit 'Failed to initialize root password'\n",
      "mysql -u root --password='", { "Ref" : "DBRootPassword" }, "' < /tmp/setup.mysql || error_exit 'Failed to create database
      user'\n",
```

Request metadata

Metadata server

```
"files" : {
  "/tmp/setup.mysql" : {
    "content" : { "Fn::Join" : ["", [
      "CREATE DATABASE ", { "Ref" : "DBName" }, ";\n",
      "CREATE USER '", { "Ref" : "DBUsername" }, "'@'localhost' IDENTIFIED BY '", { "Ref" : "DBPassword" }, "';\n",
      "GRANT ALL ON ", { "Ref" : "DBName" }, ".* TO '", { "Ref" : "DBUsername" }, "'@'localhost';\n",
      "FLUSH PRIVILEGES;\n"
      ]]},
    "mode"  : "000644",
    "owner" : "root",
    "group" : "root"
  }
},
```

And expose retrieval as API



```
"files" : {
  "/tmp/setup.mysql" : {
    "content" : "
      "CREATE DATABASE cloudstack;\nCREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'dbpass';\n
       GRANT ALL ON cloudstack.* TO 'dbuser'@'localhost';\nFLUSH PRIVILEGES;\n",
    "mode"  : "000644",
    "owner" : "root",
    "group" : "root"
  }
},
```

Metadata Server

- StackMate

In memory metadata, embeds a Sinatra server for waitcondition and metadata

- Stacktician

Rails serves metadata, persisted in DB

(Special thanks to Simon Waterhouse for integrating cfn-init scripts and creating scripts for Windows)

Stacktician Scaling Improvements

- Ruby MRI has global interpreter lock. Eventually, allowed for only one stack execution thread to run at a time
  - Use JRuby container, run in JRuby compatible server
- Async wait for stack : one thread per stack launch would overwhelm server
  - GetBack gem for managing thread pool

Delete Stack

Stacktician

- Resource IDs in database
- Execute delete APIs in reverse order of dependency creation
- Not fully tested yet

StackMate

- No DB, no IDs
- Use resource tags to figure out resources, and delete
- In progress

Improvements

StackMate

- Better logging, log level control
- Light-weight CloudStack client for making API calls

Stacktician

- Supports REST-based API for stack operations, querying and manipulation
- Command line tools for all the above operations
- sm-create-stack, sm-create-template etc

Agenda

- Introduction
- History
- Architecture and Implementation
- **Current State**
- Enhancements

Demo

Agenda

- Introduction
- History
- Architecture and Implementation
- Current State
- **Enhancements**

Future enhancements

Better plugin architecture

- Allow creation of custom type of resources
- Interface with other types of Clouds – GCE, AWS etc.
- Allow for scenarios like resource A on CloudStack, B on AWS within single stack
- Currently only in StackMate, and very rudimentary. Enforces some basic constraints, uses tag “stackmate_participant = true” for discovery
  - Bad plugins rejected at run-time
- Needs more work to make robust

Future enhancements

- Nested stacks
  - Treat stack as first class object. Template can contain another stack definition as a resource
  - Works in StackMate, disabled for lack of good implementation
- Support conditional functions in template (e.g Fn:And, Fn:Or, Fn:If)
- Support update stack
  - Semantics to be well defined

Q & A

Questions? Thoughts? Comments? Suggestions?

https://github.com/stackmate

Feel free to log issues – more the better!

Big thanks to Chiradeep for all the help