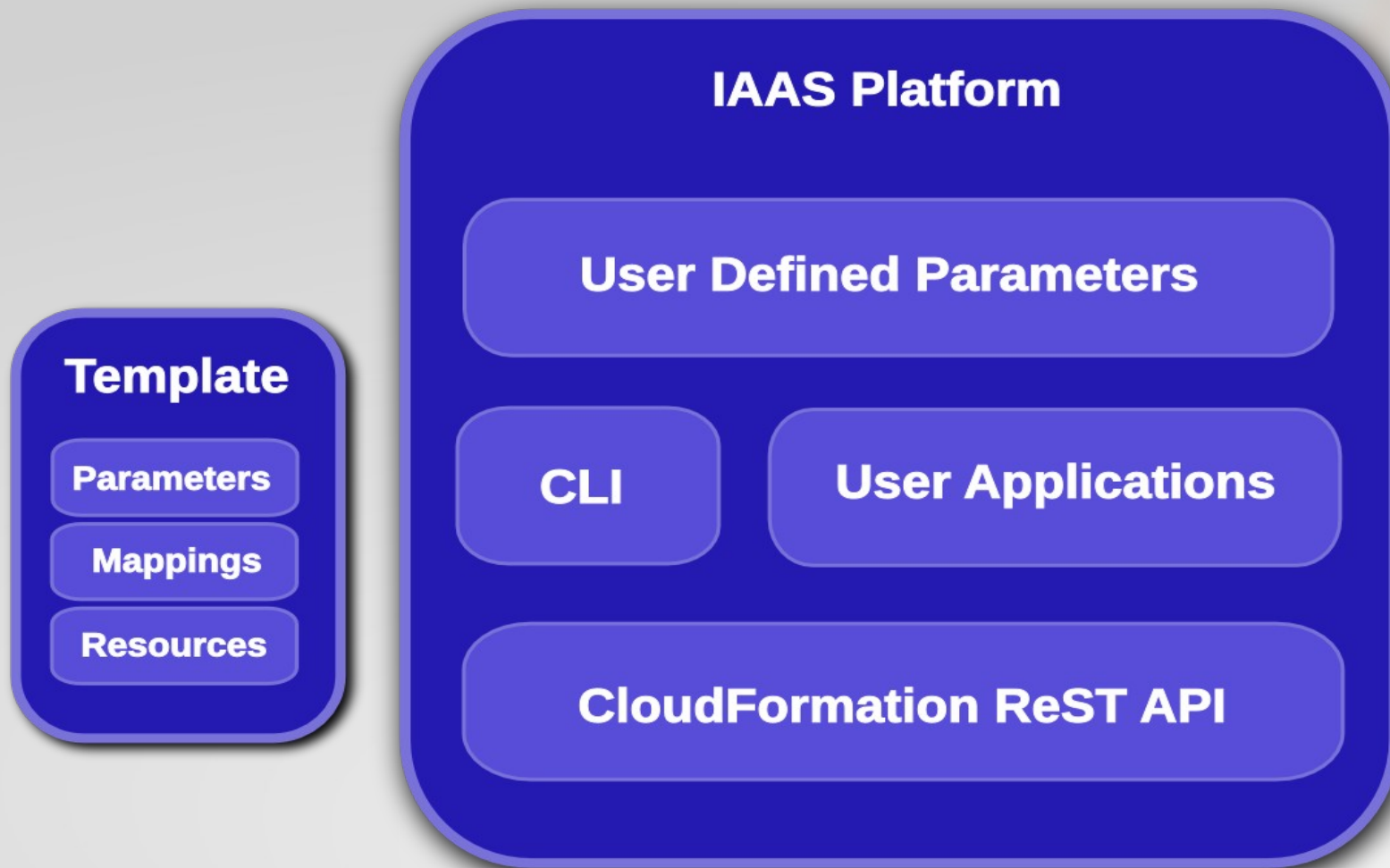


Heat API

OpenStack AWS CloudFormation Orchestration

Steven Dake (sdake@redhat.com)
CloudOpen - Thursday, August 30, 2012

CloudFormation Overview



CloudFormation API



CloudFormation API

Template

Parameters

Mappings

Resources

Life Cycle Operations
Create, Delete, Update

Introspection Operations
List, Describe, EventsList

CloudFormation Template Parameters



Definition:

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

Using the Parameter:

```
{ "Ref" : "InstanceType" }
```

CloudFormation Template Mappings



Definition:

```
"Mappings": {  
  "DistroArch2Inst": {  
    "F16" : { "32" : "F16-i386-cfntools", "64" : "F16-x86_64-cfntools" },  
    "F17" : { "32" : "F17-i386-cfntools", "64" : "F17-x86_64-cfntools" },  
    "U10" : { "32" : "U10-i386-cfntools", "64" : "U10-x86_64-cfntools" }  
  }  
}
```

Using the Mapping:

```
"ImageId": {  
  "Fn::FindInMap" : [  
    "DistroArch2Inst", { "Ref" : "Distribution" }, { "Ref" : "Arch" }  
  ]  
}
```

CloudFormation Template Resources



```
Resources {  
  "WikiDatabase": {  
    "Type" : "AWS::EC2::Instance",  
    .. bunch of stuff ...  
  },  
  
  "DatabaseIPAddress" : {  
    "Type" : "AWS::EC2::EIP"  
  },  
  
  "DatabaseIPAssoc" : {  
    "Type" : "AWS::EC2::EIPAssociation",  
    "Properties" : {  
      "InstanceId" : { "Ref" : "WikiDatabase" },  
      "EIP" : { "Ref" : "DatabaseIPAddress" }  
    }  
  }  
}
```

Heat Overview



Heat Overview

Heat API Component

RPC (RabbitMQ)

Heat Engine Component

Heat Metadata Component

OpenStack IAAS Services

Heat Image Contents



Instance Image

cloud-init

cfn-tools

Distribution JEOS
Just Enough Operation System

Fedora 16, Fedora 17, Ubuntu 10

Heat Image Creation



Image Creation

heat-jeos

OZ tdl

Oz

Upstream Repo

Glance

Heat Horizontal Scaling



Heat Horizontal Scaling

API Load Balancer

Heat API Component

Consistent Hash of Stack Name

RPC (RabbitMQ)

Heat Engine Component

Heat Metadata Component

Metadata Load Balancer

OpenStack IAAS Services

Heat Resource Types



Parameters

Type
Default
Allowedvalues
AllowedPattern
MaxLength

MaxValue
Minvalue
Description
ConstraintDescription

Other

Fn::Base64
Fn::FindInMap
Fn::GetAtt
Fn::Join
Ref

AWS::Region
AWS::StackName
DependsOn
MetaData

Resources

AWS::EC2::Volume
AWS::EC2::CustomerGateway
AWS::EC2::DhcpOption
AWS::EC2::InternetGateway
AWS::EC2::NetworkAcl
AWS::EC2::NetworkAclEntry
AWS::EC2::Route
AWS::EC2::RouteTable
AWS::EC2::Subnet
AWS::EC2::SubnetNetworkAclAssociation
AWS::EC2::SubnetRouteTableAssociation
AWS::EC2::VPC
AWS::EC2::VPCDhcpOptionsAssociation
AWS::EC2::VPCGatewayAttachment
AWS::EC2::VPNConnection
AWS::EC2::VPNGateway

AWS::AutoScaling::AutoScalingGroup
AWS::AutoScaling::LaunchConfiguraiton
AWS::AutoScaling::ScalingPolicy
AWS::AutoScaling::Trigger
AWS::CloudFormation::Authentication
AWS::CloudFormatoin::Stack
AWS::CloudFormation::WaitCondition
AWS::CloudFormation::WaitConditionHandle
AWS::CloudWatch::Alarm
AWS::EC2::VolumeAttachment
AWS::EC2::EIP
AWS::EC2::EIPAssociation
AWS::EC2::Instance
AWS::EC2::SecurityGroup
AWS::EC2::SecurityGroupIngress
AWS::IAM::AccessKey
AWS::IAM::Group
AWS::IAM::Policy
AWS::IAM::User
AWS::RDS::DBInstance

Heat Resource Types



Parameters

Type
Default
Allowedvalues
AllowedPattern
MaxLength

MaxValue
Minvalue
Description
ConstraintDescription

Other

Fn::Base64
Fn::FindInMap
Fn::GetAtt
Fn::Join
Ref

AWS::Region
AWS::StackName
DependsOn
MetaData

Resources

AWS::EC2::Volume
AWS::EC2::CustomerGateway

EC2::DhcpOption
EC2::InternetGateway
EC2::NetworkAcl
EC2::NetworkAclEntry
EC2::Route
EC2::RouteTable
EC2::Subnet

AWS::EC2::VPNGateway

AWS::AutoScaling::AutoScalingGroup
AWS::AutoScaling::LaunchConfiguraiton

AWS::AutoScaling::ScalingPolicy
AWS::AutoScaling::Trigger
AWS::CloudFormation::Authentication
AWS::CloudFormation::Stack
AWS::CloudFormation::WaitCondition
AWS::CloudFormation::WaitConditionHandl
AWS::CloudWatch::AlarmAWS::EC2::Volum
AWS::EC2::EIP
AWS::EC2::EIPAssociation
AWS::EC2::Instance
AWS::EC2::SecurityGroup

Demonstration of Nested Stacks



Normal Stack Database Instance Definition:

42 lines

```
"Resources": {
  "DatabaseServer": {
    "Type": "AWS::EC2::Instance",

    ... bunch more stuff ...

    "UserData" :
      "#!/bin/bash -v\n",
      "/opt/aws/bin/cfn-init\n",
      "# Setup MySQL\n",
      "mysqladmin -u root password '",
      { "Ref" : "DBRootPassword" }, "'\n",
      "cat << EOF | mysql -u root\n",
      "--password='", { "Ref" : "DBRootPassword" },\n",
      "'\n",
      "CREATE DATABASE ", { "Ref" :
        "DBName" }, "; \n",
      "GRANT ALL PRIVILEGES ON ",
      { "Ref" : "DBName" }, ". * TO \"", { "Ref" :
        "DBUsername" }, "\"@\"localhost\" \n",
      "IDENTIFIED BY \"", { "Ref" :
        "DBPassword" }, "\"; \n",
      "FLUSH PRIVILEGES; \n",
      "EXIT \n",
      "EOF \n"
  }
}
```

Using a RDS Nested Stack Definition:

13 lines

```
"Resources" : {
  "DatabaseServer": {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName" : { "Ref" : "DBName" },
      "Engine" : "MySQL",
      "MasterUsername" : { "Ref" : "DBUsername" },
      "DBInstanceClass" : { "Ref" : "DBClass" },
      "DBSecurityGroups" : [],
      "AllocatedStorage" : { "Ref" : "DBStorage" },
      "MasterUserPassword": { "Ref" : "DBPassword" }
    }
  }
}
```


Demonstration of High Availability



```
"Resources": {
  "WebServerRestartPolicy": {
    "Type": "HEAT::HA::Restarter"
    "Properties": {
      "InstanceId": { "Ref": "WikiServer" }
    }
  },
  "HttpFailureAlarm": {
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
      "AlarmDescription": "Restart the
      WikiDatabase if httpd fails > 3 times
      In 5 minutes",
      "MetricName": "ServiceFailure",
      "Namespace": "system/linux",
      "Statistic": "SampleCount",
      "Period": "300",
      "EvaluationPeriods": "1",
      "Threshold": "2",
      "AlarmActions": [ { "Ref":
"WebServerRestartPolicy" } ],
      "ComparisonOperator":
"GreaterThanThreshold"
    }
  },
  "WebServer": {
    "Type": "AWS::EC2::Instance",
    "Metadata": {
      "AWS::CloudFormation::Init": {
        "config": {
          "files": {
```

```
"/etc/cfn/notify-on-httpd-restarted" : {
  "content" : { "Fn::Join" : [ "", [
    "#!/bin/sh\n",
    "/opt/aws/bin/cfn-push-stats --watch ",
    { "Ref": "HttpFailureAlarm" },
    " --service-failure\n"
  ] ] },
  },
  "/tmp/cfn-hup-crontab.txt" : {
    "content" : { "Fn::Join" : [ "", [
      "MAIL=\"\"\n",
      "\n",
      "* * * * * /opt/aws/bin/cfn-hup -f\n"
    ] ] },
  },
  "/etc/cfn/hooks.conf" : {
    "content": { "Fn::Join" : [ "", [
      "[cfn-http-restarted]\n",
      "triggers=service.restarted\n",
      "path=Resources.WebServer.Metadata\n",
      "action=/etc/cfn/notify-on-httpd-
restarted\n",
      "runas=root\n"
    ] ] },
  },
  }
}
... more instance stuff ...
}}
```

Demonstration of Autoscaling



```
"Resources": {
  "WebServerGroup": {
    "Type": "AWS::AutoScaling::AutoScalingGroup",
    "Properties": {
      "AvailabilityZones": { "Fn::GetAZs" : "" },
      "LaunchConfigurationName": { "Ref" : "LaunchConfig" },
      "MinSize": "1",
      "MaxSize": "3",
      "LoadBalancerNames": [ { "Ref" : "ElasticLoadBalancer" } ]
    },
  },
  "WebServerScaleUpPolicy": {
    "Type": "AWS::AutoScaling::ScalingPolicy",
    "Properties": {
      "AdjustmentType": "ChangeInCapacity",
      "AutoScalingGroupName": { "Ref" : "WebServerGroup" },
      "Cooldown": "60",
      "ScalingAdjustment": "1"
    },
  },
  "WebServerScaleDownPolicy": {
    "Type": "AWS::AutoScaling::ScalingPolicy",
    "Properties": {
      "AdjustmentType": "ChangeInCapacity",
      "AutoScalingGroupName": { "Ref" : "WebServerGroup" },
      "Cooldown": "60",
      "ScalingAdjustment": "-1"
    },
  },
  "MEMAlarmHigh": {
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
      "AlarmDescription": "Scale-up",
      "MetricName": "MemoryUtilization",
      "Namespace": "system/linux",
      "Statistic": "Average",
      "Period": "60",
      "EvaluationPeriods": "1",
      "Threshold": "50",
      "AlarmActions": [ { "Ref": "WebServerScaleUpPolicy" } ],
      "Dimensions": [
        {
          "Name": "AutoScalingGroupName",
          "Value": { "Ref": "WebServerGroup" }
        }
      ],
      "ComparisonOperator": "GreaterThanThreshold"
    },
  },
}
```

In Closing



- Users and developers wanted!
 - Connect with the community via IRC on #heat@freenode
 - Check out the repository:
<https://github.com/heat-api>
 - Read the Documentation:
<https://github.com/heat-api/heat/wiki>
- Heat simple but powerful method for orchestrating OpenStack environments
- Heat provides near parity with CloudFormation template and API for OpenStack