

# OpenStack Heat

## OpenShift Autoscaling on OpenStack Heat

Steven Dake (sdake@redhat.com)

Twitter: steven\_dake

CloudOpen 2013- Monday, September 16, 2013

# Heat Mission



To explicitly model the relationships between OpenStack resources of all kinds; and to harness those models, expressed in forms accessible to both humans and machines, to manage infrastructure resources throughout the life-cycle of applications.

# Agenda



- HOT Format Introduction
- OpenStack Heat architecture
- Autoscaling Introduction
- OpenShift Autoscaling Workflow
- Future of Autoscaling in OpenStack
- Conclusion

# HOT Format



- CloudFormation refactored
- Parameters
- Resources
- Outputs
- Full specification:  
[http://docs.openstack.org/developer/heat/template\\_guide/hot\\_spec.html](http://docs.openstack.org/developer/heat/template_guide/hot_spec.html)

# HOT Format - Parameters



- User defined parameters passed into template from CLI or GUI
- Parameters include type, description, default value, hidden, and constraints

```
parameters:  
  InstanceType:  
    type: string  
    description: Instance type to create  
    default: m1.small  
    hidden: False  
    constraints:  
      - allowed_values {m1.tiny, m1.small, m1.large}
```

# HOT Format - Resources



- Resources for Heat to Orchestrate
- Consists of Type, Properties, DependsOn
- Resources produce global attributes

```
resources:  
  MyInstance:  
    type: OS::Nova::Server  
    properties:  
      KeyName: { get_param: KeyName }  
      ImageId: { get_param: ImageId }  
      InstanceType: { get_param: InstanceType }
```

# Hot Format - Outputs



- Displayed via CLI/GUI to identify important information of template
- Includes a description and value field

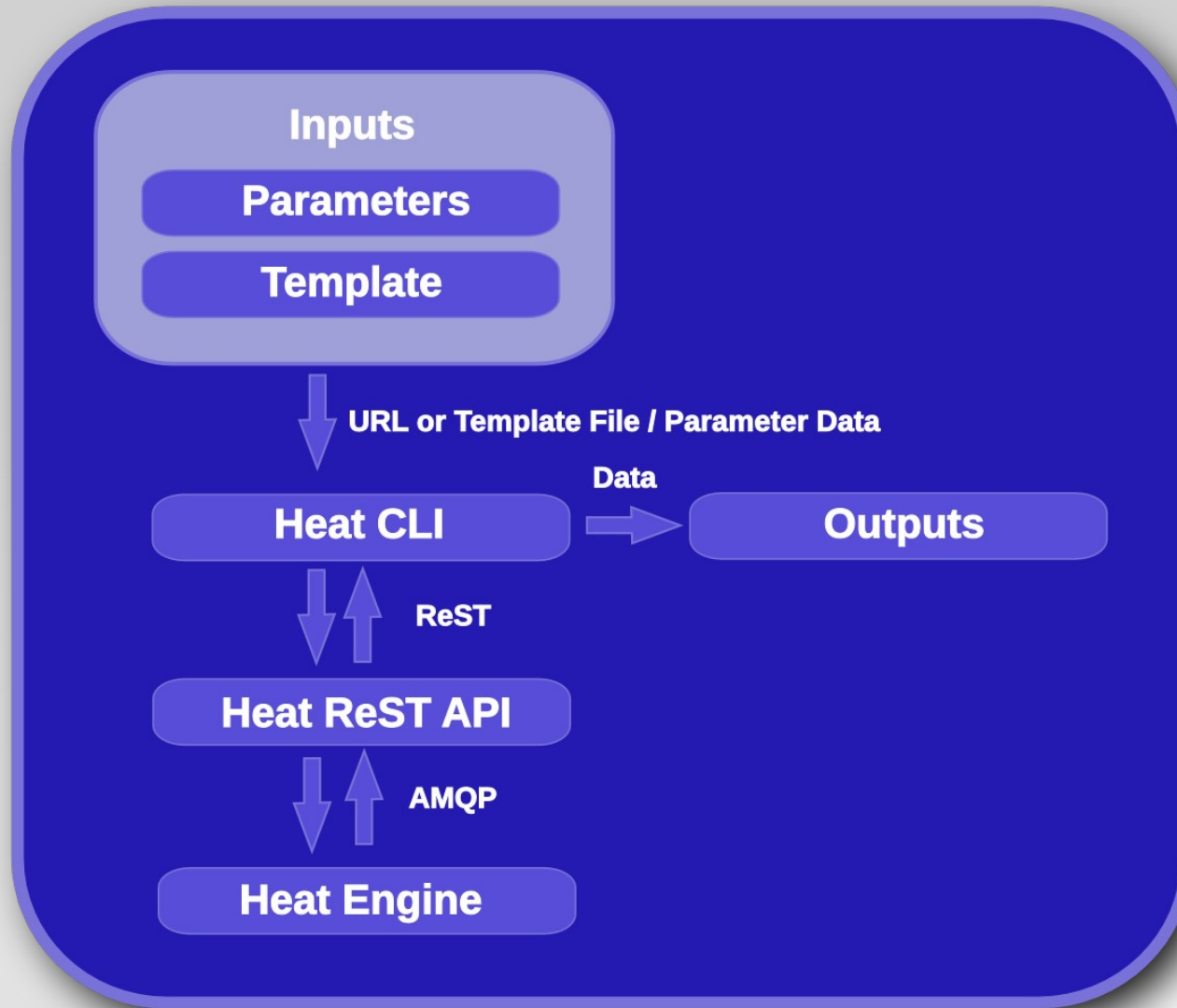
```
outputs:
```

```
  InstanceIP:
```

```
    description: The IP address of the instance
```

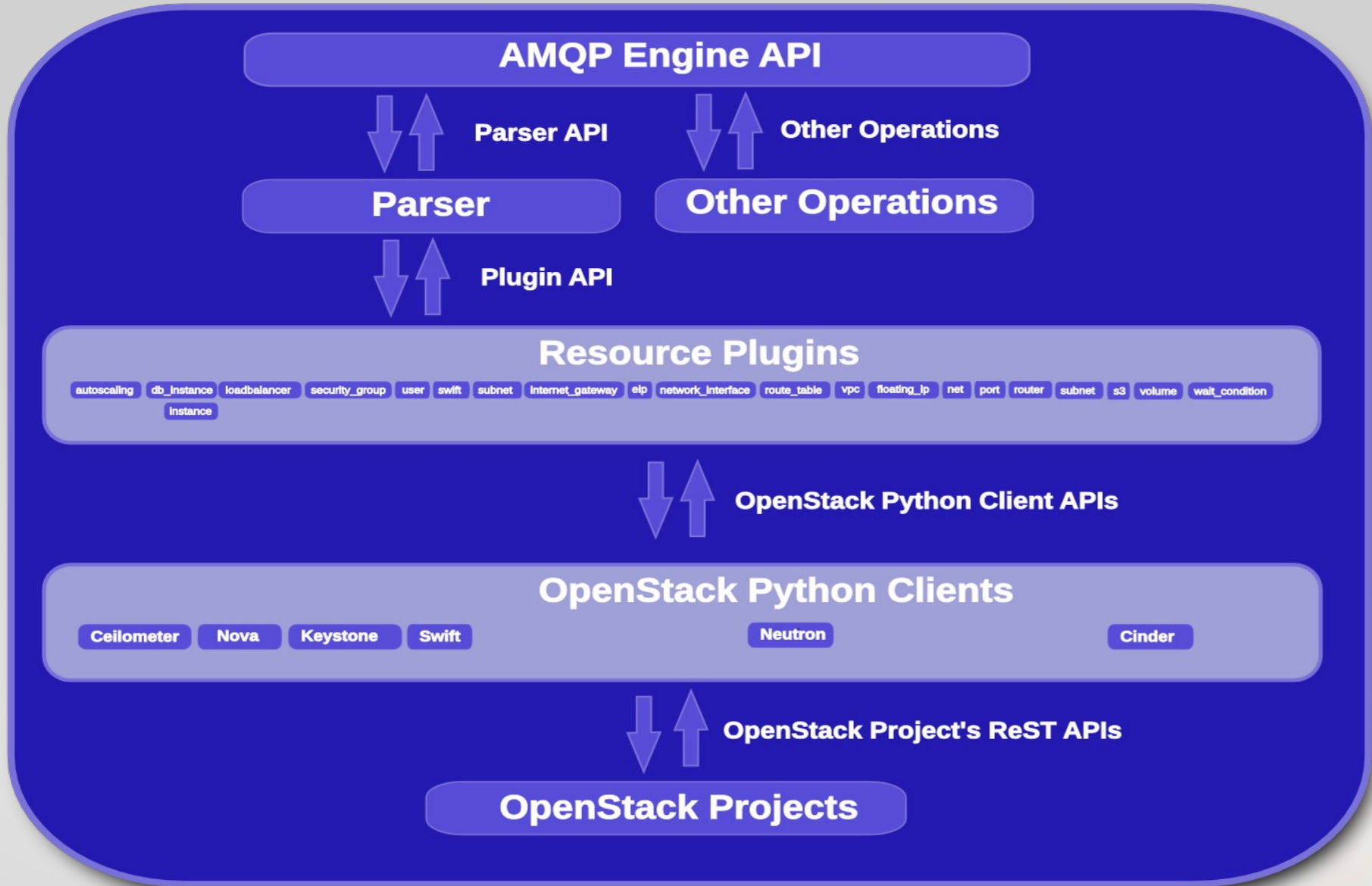
```
    value: {get_attr: [MyInstance, PublicIP] }
```

# OpenStack Heat Architecture





# OpenStack Heat Engine Architecture



# Autoscaling Introduction



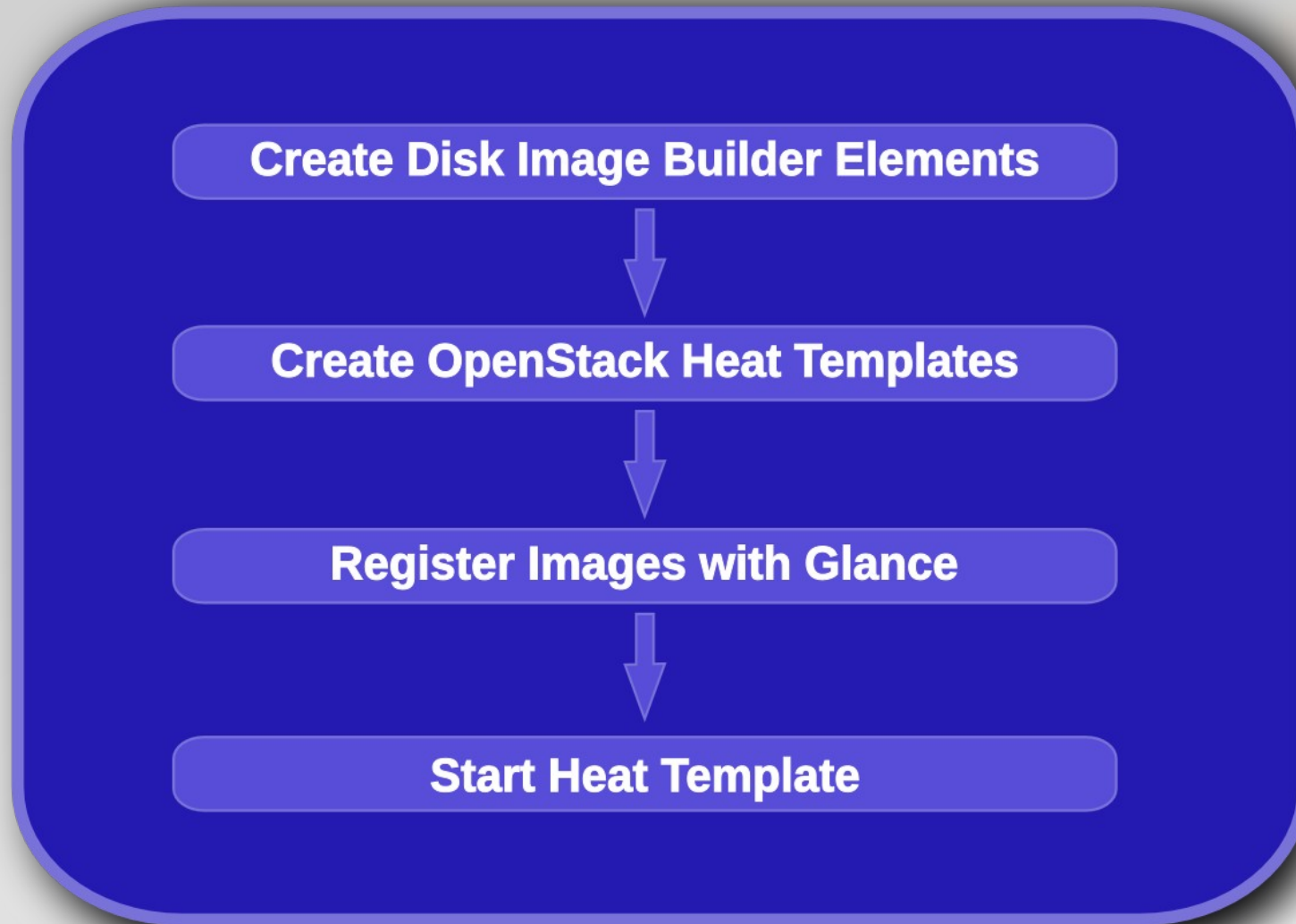
- Metrics or user events drive scaling
- Metrics can include CPU utilization, memory utilization, many more as well as custom dimensions
- Dynamically add and reduce OS::Nova::Server resources to meet demand
- Front end Neutron LBAAS or Heat provided HA-Proxy Load Balancer distributes load to server resources

# Autoscaling Workflow - Internal View



- User instantiates template with Heat's CLI
- Heat registers with Ceilometer for callbacks on Alarm events
- Ceilometer tells Heat about Alarm events and Heat scales a Group based upon a Policy decision to scale up or down
- OS::Nova::Server instances can also call Alarms internally

# OpenShift on OpenStack Autoscaling Workflow



<http://github.com/openstack/heat-templates>

# OpenShit Autoscaling Workflow

## Step 1: Create DIB elements

### Elements directory structure



```
elements/openshift-origin-broker:
```

```
-rw-rw-r--. 1 sdake sdake   37 Jun  2 12:14 element-deps
drwxrwxr-x. 2 sdake sdake 4096 Jun  2 12:14 install.d
-rw-rw-r--. 1 sdake sdake  176 Jun  2 12:14 README.md
```

```
elements/openshift-origin-broker/install.d:
```

```
-rwxrwxr-x. 1 sdake sdake 1598 Jun  2 12:14 30-openshift-origin-broker
```

```
elements/openshift-origin-node:
```

```
-rw-rw-r--. 1 sdake sdake   37 Jun  2 12:14 element-deps
drwxrwxr-x. 2 sdake sdake 4096 Jun  2 12:14 install.d
-rw-rw-r--. 1 sdake sdake  172 Jun  2 12:14 README.md
```

```
elements/openshift-origin-node/install.d:
```

```
-rwxrwxr-x. 1 sdake sdake 1610 Jun  2 12:14 30-openshift-origin-node
```

```
elements/openshift-origin-repos:
```

```
-rw-rw-r--. 1 sdake sdake   23 Jun  2 12:14 element-deps
drwxrwxr-x. 2 sdake sdake 4096 Jun  2 12:14 pre-install.d
-rw-rw-r--. 1 sdake sdake  176 Jun  2 12:14 README.md
```

```
elements/openshift-origin-repos/pre-install.d:
```

```
-rwxrwxr-x. 1 sdake sdake 286 Jun  2 12:14 29-puppetlabs-release
-rwxrwxr-x. 1 sdake sdake 648 Jun  2 12:14 30-openshift-origin-repos
```

# OpenShift Autoscaling Workflow

## Step 1: Create DIB Elements

### Building the broker image



#### Part 1: Parse Dependencies

```
[sdake@freedom openshift-origin-broker]$ more element-deps  
openshift-origin-repos
```

#### Part 2: Load Dependencies

```
[sdake@freedom openshift-origin-repos]$ ls -l pre-install.d  
-rwxrwxr-x. 1 sdake sdake 286 Jun  2 12:14 29-puppetlabs-release  
-rwxrwxr-x. 1 sdake sdake 648 Jun  2 12:14 30-openshift-origin-repos
```

#### Part 3: Configure Broker

```
[sdake@freedom openshift-origin-broker]$ ls -l install.d  
-rwxrwxr-x. 1 sdake sdake 1598 Jun  2 12:14 30-openshift-origin-broker
```

# OpenShift Autoscaling Workflow

## Step 1: Create DIB elements

### Contents of 30-openshift-origin-broker



```
[sdake@freedom install.d]$ more 30-openshift-origin-broker
#!/bin/bash

set -uex

install-packages \
    openshift-origin-broker \
    rubygem-openshift-origin-msg-broker-mcollective \
    rubygem-openshift-origin-dns-nsupdate \
    rubygem-openshift-origin-dns-bind \
    rubygem-openshift-origin-controller \
    openshift-origin-broker-util \
    rubygem-passenger \
    mod_passenger \
    openssh \
    rubygem-openshift-origin-auth-mongo \
    rubygem-openshift-origin-remote-user \
    rubygem-openshift-origin-console \
    openshift-origin-console \
    mongodb \
    mongodb-server \
    bind \
    bind-utils \
    ntpdate \
    policycoreutils \
    mcollective \
    httpd \
    openssh-server \
    rhc \
    activemq \
    activemq-client \
    git \
    puppet \
    ruby \
    ruby-devel \
    ruby-irb \
    ruby-libs \
    tar \
    yum-plugin-priorities \
    mysql-devel \
    mongodb-devel \
    system-config-firewall-base \
    rubygem-execjs \
    rubygem-uglifyer \
    rubygem-listen \
    rubygem-sass \
    rubygem-sass-rails \
    autogen-libopts \
    ntp \
    rubygem-coffee-script-source \
    rubygem-coffee-script \
    rubygem-coffee-rails \
    rubygem-idn \
    rubygem-addressable \
    rubygem-crack \
    rubygem-webmock \
    rubygem-fakefs \
    rubygem-chunky_png \
    rubygem-hpricot \
    rubygem-haml \
    rubygem-fssm \
    rubygem-compass \
    rubygem-compass-rails \
    rubygem-mongo \
    rubygem-jquery-rails \
    rubygem-openshift-origin-dns-avahi \
    rubygem-ref \
    rubygem-therubyracer

sed --in-place -e \
    s/Type=oneshot/"Type=oneshot\nTimeoutSec=0"/
/lib/systemd/system/cloud-final.service
```



# OpenShift Autoscaling Workflow

## Step 2: Create Heat Template - Policy



```
resources:
  OpenshiftUser:
    Type: AWS::IAM::User
  OpenshiftOriginKeys:
    Type: AWS::IAM::AccessKey
  Properties:
    UserName:
      Ref: OpenshiftUser
  OpenshiftOriginNodeGroup:
    Type: AWS::AutoScaling::AutoScalingGroup
    DependsOn: BrokerWaitCondition
    Properties:
      AvailabilityZones: []
      LaunchConfigurationName:
        Ref: NodeLaunchConfig
      MinSize:
        Ref: NodeCountMinimum
      MaxSize:
        Ref: NodeCountMaximum
      LoadBalancerNames: []
  OpenshiftOriginScaleUpPolicy:
    Type: AWS::AutoScaling::ScalingPolicy
    Properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName:
        Ref: OpenshiftOriginNodeGroup
      Cooldown: '120'
      ScalingAdjustment: '1'
  OpenshiftOriginScaleDownPolicy:
    Type: AWS::AutoScaling::ScalingPolicy
    Properties:
      AdjustmentType: ChangeInCapacity
      AutoScalingGroupName:
        Ref: OpenshiftOriginNodeGroup
      Cooldown: '60'
      ScalingAdjustment: '-1'
```

Alarm  
Policy  
Group





# OpenShift Autoscaling Workflow

## Step 2: Create Heat Template - Alarms



### NodeScaleUp:

Type: **AWS::CloudWatch::Alarm**

Properties:

AlarmDescription: Scale-up if event received from broker

MetricName: Heartbeat

Namespace: system/linux

Statistic: SampleCount

Period: '60'

EvaluationPeriods: '1'

Threshold: '0'

AlarmActions: [{Ref: [OpenshiftOriginScaleUpPolicy](#)}]

Dimensions:

- Name: AutoScalingGroupName

Value:

Ref: [OpenshiftOriginNodeGroup](#)

ComparisonOperator: GreaterThanThreshold

### NodeScaleDown:

Type: **AWS::CloudWatch::Alarm**

Properties:

AlarmDescription: Scale-down if event received from broker

MetricName: Heartbeat

Namespace: system/linux

Statistic: SampleCount

Period: '60'

EvaluationPeriods: '1'

Threshold: '0'

AlarmActions: [{Ref: [OpenshiftOriginScaleDownPolicy](#)}]

Dimensions:

- Name: AutoScalingGroupName

Value:

Ref: [OpenshiftOriginNodeGroup](#)

ComparisonOperator: GreaterThanThreshold

Alarm  
Policy  
Group



# OpenShift Autoscaling Workflow

## Step 2: Create Heat Template

### Optionally Trigger Alarms



UserData commands:

```
cat << EOF > /etc/heat/notify-scale-up
#!/bin/bash
/opt/aws/bin/cfn-push-stats --credential-file /etc/heat/heat-credentials --heartbeat --watch {Ref:
NodeScaleUp}
EOF
chmod 0700 /etc/heat/notify-scale-up
cat << EOF > /etc/heat/notify-scale-down
#!/bin/bash
/opt/aws/bin/cfn-push-stats --credential-file /etc/heat/heat-credentials --heartbeat --watch{Ref:
NodeScaleDown}
```

Alarm  
Policy  
Group

# OpenShift Autoscaling Workflow

## Step 3: Register images with glance



```
[sdake@freedom heat-templates] glance image-create  
--name=openshift-origin-broker --disk-format=qcow2  
--container-format=bare < openshift-origin-broker.qcow2
```

```
[sdake@freedom heat-templates] glance image-create  
--name=openshift-origin-node --disk-format=qcow2  
--container-format=bare < openshift-origin-node.qcow2
```

# Autoscaling Lifecycle Example

## Step 4: Launch Heat Template



```
[sdake@freedom heat-templates] heat create  
-template-file OpenShiftAutoScaling.yaml --parameters  
"KeyName=sdake;Prefix=broked.org;NodeCountMaximum=20"
```

# Future of Autoscaling in OpenStack



- Scale other resources beyond OS::Nova::Server
- Rackspace considering adding API model for Autoscaling based upon Otter
- Autoscaling available today in high quality format in Heat

# Conclusion



- Entering OpenStack Integrated status in November 2013
  - Active code base
    - 3048 commits as of September 2013
    - 56 contributors
  - Cross Project functionality with OpenStack projects Keystone, Nova, Neutron, Cinder, Ceilometer, Swift, Glance, Horizon, TripleO and Tempest
- HOT holds significant promise for future de-facto standard orchestration DSL
- OpenStack Heat provides application autoscaling today with a stable workflow model
- OpenShift on OpenStack in progress in the community