FleSSR Nagios Cloud Probes

This project allows nagios to monitor the abilities of a cloud using EoverI's Zeel/i Abstraction Layer

Motivation

Private Clouds are collections of software with unpredicable interactions; with no configuration changes a cloud may fail or become unresponsive. It is important to have a proactive system to alert cloud administrators to issues in their cloud - ideally before they have a major impact on users.

The highest-level test for a cloud is to determine if a user can request its resources and do some work with them and this is the challenge these probes attempt to solve.

Resources

A Eucalyptus cloud has 3 *primary resources*:

- Instances
- Volumes
- Snapshots

In addition to these primary resources, an additional resource which may be limited (based on the implementation) is the *Security Group*.

Approach

It makes sense, then, to perform basic service tests, to attempt to provision some *resources*.

Monitoring interval and intensity (how frequently resources are provisioned and how many resources are provisioned at a time) must be decided as a separate concern. There is a trade off between the accuracy of a monitoring report and the load that monitoring places on the subject cloud--frequent and intense monitoring gives a clear picture of cloud state but may interfering cloud use.

The monitoring probes first try to answer the following questions:

- Is the Cloud Provider responsive?
- Can unlimited resources be allocated on the provider (RSA key generation)

If these tests pass then some more intensive probes execute to answer the following questions:

- Can a Volume be provisioned?
    - Can a Snapshot be created?
- Can an Instance be provisioned?
- Can a Security Group be created?
    - Does the network allow traffic to Instances?

The probes have configuration files to allow the user to specify:

- The Instance Type and Image to provision
- The SSH user for connetivity tests
- The Volume size for disk tests

The Probe Package

The Probe Package is a collection of **probe specifications** and binaries which allows the monitoring of a single Cloud Provider. Multiple Probe Packages may be deployed on a single machine, allowing one nagios instance to monitor many clouds simultaneously.

It is necessary to configure the probe package for each cloud they must monitor. The key requirements are:

- Zeel/i Cloud Provider Implementation
- Cloud Provider credentials

Additionally, box provisioning probes should be customised to supply:

- AMI/EMI and Instance Type for provisioning tests

Directory structure

The top-level directories are as follows:

- **bin** holds binaries used by the probe package
- **config** holds the core configuration for the probe package
- **lib** holds the libraries necessary to run the probes
- **probe** holds autogenerated probe launch scripts
- **spec** holds probe specifications (and probe configuration)
- **example** holds example nagios configuration files

Installation

Cloud Provider credentials

Zeel/i Standalone Provider
This monitoring package includes a standalone version of Zeel/i to minimise the number of moving parts in the monitoring system

Once the probes are configured the cloud provider credentials should be configured. The **config/harness.config** file contains the class of the Zeel/i Managed Provider and the credentials for an account on that provider (for EC2 API compliant clouds, the accessKey and secretAccessKey).

config/harness.config

```
provider.class=com.eoveri.flessr.nagios.test.provider.TestableOxfordBoxProvider
provider.accessKey=someAccessKey
provider.secretAccessKey=someSecretAccessKey
```

Generating nagios probe scripts

Autogenerated scripts and absolute paths
The scripts generated by this process include absolute file paths. If you move the monitoring package be sure to re-run this stage

To use the default probe specifications as shipped, once the probe package is installed simply run

```
./bin/generate-probes.sh
```

This will generate a number of probe scripts (based on the probe specifications in the **spec** directory) in the **probe** directory. Each of these probe scripts will, when run, test some cloud functionality and then return the status in the format nagios requires.

Consult the **example** directory for how to run the probes from nagios.

Customising Instance probes

The Probe Package ships with two probes which provision Instances: **provision-box** and **test-box-connectivity**. Their configuration file can be found in **spec/(name)/test.config**. The configuration files contain documentation describing what each configuration field means.

spec/provision-box/test.config

```
# The Instance Type Name and Eucalyptus Machine Image Id to provision.
# The test will fail if one of these instances cannot be provisioned
# due to insufficient capacity or because the Image does not exist
# NOTE: Eucalyptus EMIs are case sensitive

instanceType=m1.small
ami=emi-095E1140
```

Customising Disk probes

Volume sizes
For testing, strike a balance between what is a useful size to confirm is available and the load it will place on the cloud: many provider implementations zero volumes on allocate, so this test may spend a lot of time waiting for a volume to be zeroed (and then immediately delete it)

The Probe Package ships with two probes which provision Disks: **provision-disk** and **snapshot-disk**. Their configuration, found in **spec/(name)/test.config**, specifies the volume size to provision.

Adding probes

It is possible to create custom probe specifications - for instance, to have a probe for provisioning m1.small instances as well as a probe for provisioning m1.xlarge instances. Each directory in the **spec** directory is translated into a probe script, each of which can be run independently.