zhian3191870

Bootstrap Juju on top of openstack

Pre-requisites：

Assume you have Juju (version: 2.0.2-xenial-amd64) and openstack installed properly

The latest stable version of Juju is 2.2.2, you can bootstrap Juju 2.2.2 on top of openstack properly, however it seems that Juju 2.2.2 could not be used to depoy clearwater(vIMS)
If you use apt-get to install juju, which version will be installed depends on your ubuntu series, on 14.04,you get version 1.25 installed by default

ADD NEW CLOUD FOR OUR OWN OPENSTACK PRIVATE CLOUD

create cloud defination file in yaml format

```
vim myopenstack.yaml
```

An example cloud defination file ;

```
clouds:
    myopenstack:
      type: openstack
      auth-types: [userpass]
      regions:
        regionOne:
          endpoint: http://192.168.37.12:5000/v2.0
```

Juju add cloud

```
juju add-cloud myopenstack myopenstack.yaml
```

Adds credentials for our cloud

create new credential file in yaml format

An example credential file looks like as follows

```
credentials:
  myopenstack:
    myopenstack:
      auth-type: userpass
      password: eevNrAGtXWKduWD2G2f6uNmWN
      tenant-name: admin
      username: admin
```

then add credential for your own cloud

```
juju add-credential myopenstack -f mycredential.yaml
```

INSTALL HTTP SERVER TO PROVIDE IMAGE MEDATATA

```
sudo apt install nginx
```

while some error occurs,

```
Unmet dependencies. Try 'apt-get -f install' with no packages (or specify a
solution)
```

use following commands to fix

```
sudo apt --fix-broken install
sudo apt-get update
sudo apt-get upgrade
```

GENERATE IMAGE METADATA&TOOLS FOR UBUNTU IMAGE IN OUR OPENSTACK

you should have uploaded these images to openstack before doing this

```
mkdir mt  #mt respresents for metadata :)

sudo juju metadata generate-image -s trusty -i 5e43a21c-2a96-42ea-b218-
ba88adef96f3 -r regionOne -d mt -u http://192.168.37.12:5000/v2.0
```

-s series

-i image id in your openstack

-r region

-d which directory to save your metadata

-u endpoint url

generate tools

```
sudo juju metadata generate-tools -d mt
```

copy metadata file to our web server root directory

```
sudo cp -a mt/tools/ /var/www/html/
sudo cp -a mt/images/ /var/www/html/
```

make these files accessible, we just use

```
sudo chmod -R 777 /var/www/html/
```

for convenience

make sure our web server is running

```
ps -ef | grep nginx
```

you should see nginx working process

BOOTSTRAP JUJU CONTROLLER

(Juju version: 2.0.2-xenial-amd64)

```
juju --debug --show-log  bootstrap myopenstack myopenstack --config image-
metadata-url=http://192.168.22.7/images --config network=test_hou --config
use-floating-ip=True
```

Here is the trickest part that take us some time to figure out,add proper DNS for juju controller when it's ready. (you can see it in openstack dashboard,ready means you can ssh to juju controller)
Two ways to add DNS address for juju controller, one is add DNS record manually, ssh to juju controller with the private key file generated by Juju which will be injected into juju

controller machine while creating instance.The other way is pretty much easier,add proper DNS server address for your network specified to Juju controller
If lucky, we have done it, next open Juju gui

```
juju gui
juju gui --show-credentials  # show username and password to login into juju dashboard
```

Restrictions：

Juju 1.25 was not going to work because of known issue hadn't been fixed :

Unable to bootstrap with --metadata-source, more details here https://bugs.launchpad.net/juju-core/+bug/1591488
Uploading image metadata files to swift container and then exposing a service on openstack was not recommended because of you may need to make sure your authentication with keystone works properly
Assume you have Juju (version: 2.0.2-xenial-amd64) and openstack installed properly

The latest stable version of Juju is 2.2.2, you can bootstrap Juju 2.2.2 on top of opentack properly, however it seems that Juju 2.2.2 could not be used to depoy clearwater(vIMS)
If you use apt-get to install juju, which version will be installed depends on your ubuntu series, on 14.04,you get version 1.25 installed by default

ADD NEW CLOUD FOR OUR OWN OPENSTACK PRIVATE CLOUD

create cloud defination file in yaml format

```
vim myopenstack.yaml
```

An example cloud defination file ;

```
clouds:
    myopenstack:
      type: openstack
      auth-types: [userpass]
      regions:
        regionOne:
          endpoint: http://192.168.37.12:5000/v2.0
```

Juju add cloud

```
juju add-cloud myopenstack myopenstack.yaml
```

Adds credentials for our cloud

create new credential file in yaml format

An example credential file looks like as follows

```
credentials:
  myopenstack:
    myopenstack:
      auth-type: userpass
      password: eevNrAGtXWKduWD2G2f6uNmWN
      tenant-name: admin
      username: admin
```

then add credential for your own cloud

```
juju add-credential mycredential.yaml
```

INSTALL HTTP SERVER TO PROVIDE IMAGE MEDATATA

```
sudo apt install nginx
```

while some error occurs,

```
Unmet dependencies. Try 'apt-get -f install' with no packages (or specify a
solution)
```

use following commands to fix

```
sudo apt --fix-broken install
sudo apt-get update
sudo apt-get upgrade
```

GENERATE IMAGE METADATA&TOOLS FOR UBUNTU IMAGE IN OUR OPENSTACK

you should have uploaded these images to openstack before doing this

```
mkdir mt  #mt respresents for metadata :)

sudo juju metadata generate-image -s trusty -i 5e43a21c-2a96-42ea-b218-
ba88adef96f3 -r regionOne -d mt -u http://192.168.37.12:5000/v2.0
```

-s series

-i image id in your openstack

-r region

-d which directory to save your metadata

-u endpoint url

generate tools

```
sudo juju metadata generate-tools -d mt
```

copy metadata file to our web server root directory

```
sudo cp -a mt/tools/ /var/www/html/
sudo cp -a mt/images/ /var/www/html/
```

make these files accessible, we just use

```
sudo chmod -R 777 /var/www/html/
```

for convenience

make sure our web server is running

```
ps -ef | grep nginx
```

you should see nginx working process

BOOTSTRAP JUJU CONTROLLER

(Juju version: 2.0.2-xenial-amd64)

ADD NEW CLOUD FOR OUR OWN OPENSTACK

create cloud defination file in yaml format

NOTE: a file in yaml format doesn't support any tabs

```
vim myopenstack.yaml
```

An example cloud defination file ;

```
clouds:
    myopenstack:
      type: openstack
      auth-types: [userpass]
      regions:
        regionOne:
          endpoint: http://192.168.37.12:5000/v2.0
```

Juju add cloud

```
juju add-cloud myopenstack myopenstack.yaml
```

Adds credentials for our cloud

create new credential file in yaml format

An example credential file looks like as follows

```
credentials:
  myopenstack:
    myopenstack:
       auth-type: userpass
       password: eevNrAGtXWKduWD2G2f6uNmWN
       tenant-name: admin
       username: admin
```

add credential file for your own cloud using following command

```
juju add-credential mycredential.yaml
```

INSTALL HTTP SERVER TO PROVIDE IMAGE MEDATATA URL

```
sudo apt install nginx
```

while some error occurs,

```
Unmet dependencies. Try 'apt-get -f install' with no packages (or specify a
solution)
```

use following commands to fix

```
sudo apt --fix-broken install
sudo apt-get update
sudo apt-get upgrade
```

GENERATE IMAGE METADATA&TOOLS FOR UBUNTU IMAGE IN OUR OPENSTACK

```
mkdir mt  #mt respresents for metadata

sudo juju metadata generate-image -s trusty -i 5e43a21c-2a96-42ea-b218-
ba88adef96f3 -r regionOne -d mt -u http://192.168.37.12:5000/v2.0
```

generate tools

```
sudo juju metadata generate-tools -d mt
```

copy metadata file to our web server root directory

```
sudo cp -a mt/tools/ /var/www/html/
sudo cp -a mt/images/ /var/www/html/
```

make these file accessible, we just use

```
sudo chmod -R 777 /var/www/html/
```

for convenience

make sure our web server is running

```
ps -ef | grep nginx
```

you should see nginx working process

BOOTSTRAP JUJU CONTROLLER

```
juju --debug --show-log  bootstrap myopenstack myopenstack --config image-
metadata-url=http://192.168.22.7/images --config network=test_hou --config
use-floating-ip=True
```

If lucky, we have done it, next open Juju gui

```
juju gui
juju gui --show-credentials  # show username and password to login into juju
dashboard
```

Restrictions：

Juju 1.25 was not going to work because of known issue hadn't been fixed :

Unable to bootstrap with --metadata-source, more details here
https://bugs.launchpad.net/juju-core/+bug/1591488
Uploading image metadata files to swift container and then exposing a service on openstack was not recommended because of you may need to make sure your authentication with keystone works properly