Samba 4 in the cloud

Jelmer Vernooij

Samba Team

10 mei 2012

Agenda

Introduction

Running the Samba tests in the cloud

Samba 4 Juju

Introduction

Why the cloud?

The Cloud

- ▶ "Computing/storage as a service"
- ▶ External computing resources
- ▶ On-demand
- ▶ Scale as necessary

Amazon Cloud

- Public cloud
    - Usable by anyone
    - `aws.amazon.com`
- Couple of data centers all over the world
- Management API

Amazon Web Services

- ▶ REST (HTTP) API for Cloud management
- ▶ Operations
    - ▶ Managing instances
    - ▶ Managing images
    - ▶ Managing security

euca2ools

euca2ools are a set of tools that use the AWS API

- euca-create-instance
- euca-describe-instances
- euca-terminate-instances
- euca-start-instances
- euca-stop-instances

python-boto

python-boto is a Python library for accessing the AWS API.

Example

```
>>> import boto
>>> ec2_conn = boto.connect_ec2(
    AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY)
>>> images = ec2_conn.get_all_images(
    image_ids=['ami-b111f4d8'])
>>> images[0]
Image:ami-b111f4d8
>>> reservation = images[0].run(1,1, 'yourKeyPair')
>>> reservation
Reservation:r-c2dc2fab
```

Build farm issues

- ▶ Not much control over build farm machines
  - ▶ Troubleshooting when machine breaks
  - ▶ Software that is installed
- ▶ Limited resources
  - ▶ Some machines only build some commits, are much slower

Test Scenarios

1. Fire up a bunch of instances with particular images
2. Run testsuite
3. Tear down or suspend image

Different images:

- ▶ Various operating systems / versions
- ▶ With different packages installed

On-demand:

- ▶ Always build on all supported systems every X minutes
- ▶ Build on each after each commit

Thoughts

- ▶ Shouldn't be an excuse for making the testsuite slow
- + more reliable than existing build farm
- + more control than existing build farm
- + can run against more servers
- - more expensive
- - trusting Amazon

Private Cloud

- ▶ Cloud "in the basement"
    - ▶ OpenStack
    - ▶ Eucalyptus
    - ▶ On a single heavy-duty machine (LXC)
    - ▶ . . .
- ▶ ... not really relevant for Samba?

Juju

What is juju?

- ▶ "Service Orchestration"
- ▶ Coupled with provisioning
- ▶ Complimentary to config management (chef, puppet, ...)
- ▶ Uses service descriptions ("charms") to deploy services, usually one per VM
- ▶ (often) takes care of management of extra services
- ▶ Originally Ubuntu-based, but not really Ubuntu-specific

Charms

- ▶ Usually run in their own system (VM, LXC container, ...)
- ▶ Similar to packages
- ▶ Simple yaml metadata
- ▶ A few scripts that run at the right time
    - ▶ install-time
    - ▶ upgrade
    - ▶ removal
    - ▶ config changes
    - ▶ relationship changes

Charm — Example

— Demo —

Charms — relationships

Samba 4 in the cloud

Jelmer Vernooij

Introduction

Running the
Samba tests in the
cloud

Samba 4 Juju

- ► Each charm
  - ► Provides certain interfaces
  - ► Consumes certain interfaces
- ► *wordpress* uses the *database* interface provided by *mysql*

Charms — scaling

- juju deploy mysql wiki-slave-db
- juju add-relation wiki-slave-db:slave wiki-db:master
- juju add-relation demo-wiki:slave-db wiki-slave-db:db

Example — deployment

- ▶ juju set demo realm="SambaXP"
- ▶ juju deploy samba4 –config=my-domain.yaml mydomain

Deployment options

- ▶ Amazon EC2
- ▶ Local environment
- ▶ Private cloud
    - ▶ OpenStack
    - ▶ Eucalyptus
- ▶ Bare metal (MaaS)

A charm for Samba 4

- Focused on just DC
- Simple wrapper around "samba4" Debian package

Current Status

- ▶ Basic charm works
    - ▶ Automatically provisions
    - ▶ Allows configuring domain/realm
    - ▶ Based on daily Ubuntu package of Samba4
- ▶ GPLv3+ licensed
- ▶ Unfinished as of yet:
    - ▶ Only basic setup
    - ▶ No file sharing
    - ▶ No replication setup

More information

- ▶ *#juju* on *irc.freenode.net*
- ▶ http://juju.ubuntu.com/
- ▶ Existing charm:
  lp:~jelmer/ubuntu/charm/samba4/trunk