ARCHITECTURE OF A CLOUD SERVICE USING PYTHON TECHNOLOGIES

ABRAHAM MARTIN
@ABRAHAM_MARTINCO
MANAGED WEB SERVICE

• Born to solve a problem around university
• Servers under desks
• Security problems
MANAGED WEB SERVICE

- Managed:
  - Software/OS maintained by us
  - Web hosting capabilities (PHP, CGIs, MySQL…)
  - No backups worries
  - Dedicated resources (v2)
MANAGED WEB SERVICE

- v1
  - Solaris 7, Apache 1.3, PHP 4.3, MySQL 4.1…
  - home-grown system involving chroot and loop back mounts

- v2
  - Updated Software (Solaris 10, Apache 2, PHP5, MySQL, perl…)
  - Solaris Zones
MANAGED WEB SERVICE

- v2
  - Database driven (scripts launched)
  - NIS and NFS server
  - Replicated but manual failover
  - ZFS
  - vhosts, aliases…
  - Manual process (or executing scripts) but not available for end users
MANAGED WEB SERVICE

- v2
  - > 200 users
  - > 400 websites
MANAGED WEB SERVICE

- Falcon
- Plone based
- >200 sites
MANAGED WEB SERVICE

- v3
- Restart
- Complete Isolation, dedicated VMs
- No root access
- Managed and maintained by “us” but still offering same (and more) options
- Web panel to delegate users some power
MANAGED WEB SERVICE

• v3

• Debian 8 (AMP by default)

• Other apache mods available (e.g. mod_wsgi)

• List of system packages available to install

• Authorisation, vhost, dns, tls, backups, password reset, and power management given to the user

• Fully automated processes based on a web panel.
Messages:

- No billing details are available, please add them.

- Production server
  IPv4: 131.111.58.246
  IPv6: 2001:630:212:8::8c:246
  hostname: mws06767.mws3.csx.cam.ac.uk
  hostkey fingerprint:

- Test server
  IPv4: 172.28.18.246
  hostname: mws-47746.mws3.csx.private.cam.ac.uk
  hostkey fingerprint:
Managed Web Service administration site

Server settings

- Web sites
- System packages
- Unix Groups
- Change database root password
- The server is currently ON
- If it does not respond you can do a hard reset by clicking here
- Restore backup

The Managed Web Service is provided by the University Information Services.
MANAGED WEB SERVICE

- v3
- Test server (for testing upgrades, changes, etc)
- Clone options
VM ARCHITECTURE

• Dedicated Managed VMs
• VMWare solution
• vSphere control panel + APIs
• ESXi servers
• External backup server
• No replicated
VM ARCHITECTURE

• Flow

• Django web panel receives request from authenticated user

• A hostname and IPs (4&6) are allocated

• VM API to create a new VM

• VM API to install OS (Callback when VM ready)

• Ansible is executed
ANSIBLE

• Application Deployment + Configuration Management + Continuous Delivery

• Inventory of targets (dynamic or static)

• Roles (DB server, Web server, etc)

• A target can have more than one role

• Playbook: Targets and roles
ANSIBLE PLAYBOOK

---
#mwsclients.yml; playbook for MWS client machines
- hosts: mwsclients
gather_facts: no
roles:
  - common
  - mwscommon
  - metrics_service
  - mwsclient

• For each role:
  • tasks (yaml), templates (jinja2), scripts, handlers, vars
- name: update software
  apt:
    upgrade: dist
    update_cache: yes
  tags: upgrades

- name: install base software
  apt:
    state: present
    name: {{item}}
  with_items:
    # Base MWS software
    - openssh-server
    - apache2
    - libapache2-mod-ucam-webauth
    - libapache2-mod-php5
    - mysql-server
    - php5
    - php5-gd
    - php5-mysql
    - php5-mcrypt
    - git
    # Software for interactive users
    - screen
    - emacs
    - vim-gtk
  tags: base_software

- name: static network configuration
  template:
    dest: /etc/network/interfaces
    src: interfaces.j2
  notify: reboot
# mwsclient/handlers/main.yml - handlers file for the mwsclient role

- name: reload Apache
  service: name=apache2 state=reloaded

- name: restart autofs
  service: name=autofs state=restarted

- name: reboot
  command: shutdown -r -t 1
MANAGED WEB SERVICE

- Authentication
  - Raven (potentially Shibboleth/SAML2)
  - Custom auth backend
  - Webauth
Authorised users and groups

You can authorise other users as administrators or ssh-only users. Administrators will have access to all the features of the control panel and can connect to the server via SSH. SSH-only users will only have access to the server via SSH. They won't be able to access the web panel.

You can search for other users using the text field below by typing their name or CRSid. It will let you autocomplete by selecting their entry from the drop down list.

You can also authorise users or administrators using Lookup groups.

Administrators:

Dr Abraham Martin-Campillo (amc203)

SSH-only users:

Search for user

Administrators lookup groups:

Search for groups

SSH-only lookup groups:

Search for groups

The list of users authorised using Lookup groups are refreshed every 24 hours. If you want to refresh it now, you can use the following button.

Force update >
AUTHORISATION (LDAP-ish BASED)

• *nix users:

  • User is installed in the VM (Using Ansible)
  
  • UID (important for shared file storage) taken from Jackdaw (User central database)
  
  • Periodic task to refresh installed users (in VMs) authorised via LDAP groups
  
  • SSH public key uploaded to the web panel
ARCHITECTURE

Raven
Lookup

Cron tasks

Authentication
Authorisation
Django Control Panel

Celery async tasks

VM API
IPReg API
Ansible
Mailing
Auth

VM guests (Xen or VMware)

IPReg
Jackdaw

Bes API
Billing API

Bes++
Finance

VM server (Xen or VMware)
IP REGISTER API

- Preallocated IP addresses
- cam.ac.uk domains aliases available for users (API)
- Service/Host addresses
- SSHFP records and DNSSEC
The authenticity of host 'test.dev.mws3.csx.cam.ac.uk (131.111.8.73)' can't be established.
Are you sure you want to continue connecting (yes/no)?
CENTRAL INVENTORY

- Bes++ (django)
- JSON file with information about all hosts:
  - Location, IP, hostname, VM properties
- Pull consumed
API COMMUNICATION TYPES

- REST / non REST HTTPS APIs
- SSH APIs
- JSON / non JSON
- Callbacks
ASYNC TASKS

- Some API calls
- Background processes
- Cron jobs
- Celery
- Redis
@shared_task
(base=TaskWithFailure,
default_retry_delay=5*60, max_retries=288)
# Retry each 5 minutes for 24 hours
def foo(param):
    var
class TaskWithFailure(Task):
    abstract = True

    def on_failure(self, exc, task_id, args, kwargs, einfo):
        LOGGER.error("An error happened")
CELERYBEAT_SCHEDULE = {
    'cronjob1': {
        'task': 'apimws.task1',
        'schedule': timedelta(hours=1, minutes=30),
        'args': ()
    },
}

MANAGED WEB SERVICE

- More features (all Ansible driven)
  
  - Change DB root passwd
  
  - Create vhosts
  
  - Aliases
  
  - TLS Certs
  
  - Install some system packages
  
  - Backups (Snapshots)
VM ARCHITECTURE (1)
VM ARCHITECTURE (2)

Network Conf: IP + Mask + Gateway

Web Server: Apache

Storage: no replicated / replicated

DRBD: Primary -> Sync -> Secondary

Services

Resource Manager: Pacemaker

Messaging: CoroSync

Prod VMWare VM 1

Prod VMWare VM 2

Test VMWare VM
VM ARCHITECTURE (3)

VM Pool: Xen VM, Xen VM, Xen VM

Xen VBD: Virtual block devices

DRBD: Primary → Sync → Secondary

Resource Manager: Pacemaker

Messaging: CoroSync

Xenserver 1, Xenserver 2, Dummy VM

domU (Service) dom0
MANAGED WEB SERVICE

- Deployment of Xen servers
- Three-node cluster
- Nodes on different location
- Live migration
- Deployed using Ansible
- Different service (API)
- name: django collect static files
  sudo: yes
  sudo_user: www-data
  django_manage: command=collectstatic app_path={{install_web_dir}}/
                  settings={{django_name}}.production_settings

- name: disable apache default site
  command: a2dissite default
            removes=/etc/apache2/sites-enabled/000-default.conf

- name: enable django site
  command: a2ensite {{django_name}}
            creates=/etc/apache2/sites-enabled/{{django_name}}

- name: install celeryd config file
  template: src=celeryd.j2
             dest=/etc/default/celeryd
  notify:
     restart celery
# mwsserver/handlers/main.yml - handlers for the mws server

- name: restart apache
  service: name=apache2 state=restarted

- name: restart celery
  service: name={{item}} state=restarted
  with_items:
    - celeryd
    - celerybeat
SECURITY

• No root passwords, only keys

• Separation of privileges (different users)
  • pre-generation of host keys
  • userv services

• TLS certs
“The HTTP/2 specification itself won’t require the use of TLS, even though many (or possibly all) browsers will do so for the new protocol.”

-MARK NOTTINGHAM
CHAIR OF THE IETF HTTP WORKING GROUP
SSL Report: test.dev.mws3.csx.cam.ac.uk (131.111.8.73)
Assessed on: Tue, 21 Jul 2015 20:23:19 UTC | Clear cache

Summary

Overall Rating

Certificate 100
Protocol Support 95
Key Exchange 90
Cipher Strength 90

Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports TLS_FALLBACK_SCSV to prevent protocol downgrade attacks.

This server supports HTTP Strict Transport Security with long duration. Grade set to A+. MORE INFO »
METRICS AND LOGGING

• statsd & collectd
• cluster AMQP message brokers
• cluster carbon/graphite (storage)

Grafana
METRICS AND LOGGING