Introducing self-service to researchers

Bart Vanbrabrant
About me

- Did a PhD at KU Leuven
- CTO of Inmanta
  - KU Leuven spin-off
  - Develops an orchestration tool
  - Open source at https://github.com/inmanta/inmanta
- Research fellow at KU Leuven
- Was here in 2009
Research environment

- Large research group
  - ~ 80 researchers
  - Research on software engineering, security, networks and systems, …
Research environment

- Single person silos
- “Tiger teams”
- Large project teams
Resources for experiments

- Based on smaller teams and projects
- Very limited for the single person silo
- Labs with equipment: mostly desktops out of warranty
About my PhD

A Framework for Integrated Configuration Management of Distributed Systems
About my PhD

A Framework for Integrated Configuration Management of Distributed Systems

Een raamwerk voor geïntegreerd configuratiebeheer van gedistribueerde systemen
About my PhD

A Framework for Integrated Configuration Management of Distributed Systems
Single person silo
Single person silo

ME
Networking lab
Networking lab
Specs

- PII 400 mhz
- <= 256 ram
- 4 and 8 GB disks
- 1x 100M NIC
Optiplex GX100
Specs

- PII and PIII: 400 -> 650 Mhz
- 512M ram
- 8 GB disks
- 2x 100M NIC
Stacks
Lots of them
Optiplex GX1

2008
Optiplex GX1

2011
Desktop hoarding

- New desktop every 4 years
- Desktops self-managed
  - Linux
  - Windows
  - *BSD
Desktop hoarding

- New desktop every 4 years
- Desktops self-managed
  - Linux
  - Windows
  - *BSD
- $10/4*80 = 200$ missing desktops?
Desktop hoarding

- New desktop every 4 years
- Desktops self-managed
  - Linux
  - Windows
  - *BSD
- $10/4*80 = 200$ missing desktops?

Where are they?
Shadow-IT

- Multiple old desktops under desk
- Sharing desktop stacks in offices
- You can use mine if I can use yours
- ...

26
Shadow-IT

- No pool of hardware to use
- No backups
- No updates
- No idea what is actually available
- All machines have a public IP!
Cause

- Silos
- Not just Devs vs Ops
  - Central IT
  - Local IT
  - Research group IT
  - Researchers
- Different goals
- Communications through tickets, email, …
No silo any more

- Researcher with operational skills
- Collaborate in projects, using other research as case study
- Java and Application servers
Tipping point

- Use a J2EE application as case study
Tipping point

- Use a J2EE application as case study
- Deploy Jboss on machines with **256M** ram
Tipping point

- Use a J2EE application as case study
- Deploy Jboss on machines with 256M ram
- In 2011
SO YOU ARE TELLING ME TO DEPLOY JBOSS 7

ON AN OPTIPLEX GX100?
Managed servers

- Buy one or more beefy servers
- Use for a single research project (2 to 5 years)
- Peak usage
- In DC (“computerzaal”) by local IT staff
- Ubuntu LTS
Managed servers

- Buy new server in may 2014
  - New OS: 14.04

- Buy new server in march 2014
  - 2 year old OS: 12.04
More recent software?

- Researcher: I need a newer version of X
- IT staff: There is no update in Ubuntu repo
- Researcher: But I really need it!
- IT staff: No worry, we installed GCC and increased your quota. But please, put it in a NoBackup folder
Network zones

- DMZ
- Intranet
- Lab
- Experiments
Virtualization

- Managed system
- Root access on virtual machines
- No solution for networking and storage
Self service

- Setup a pool of hardware
- Isolate users from each other
- Users and teams can create servers, networks, change firewall rules, use multiple zones, offer services to other users, …
Do we already have our private Cloud?

Just Finished!

PRIVATE CLOUD CENTER

CEO

CIO

DATA CENTER
Public cloud

- Easy: credit card and ready
- Bounded to their AUP
- Harder to predict cost
- Data not on premise
- Not so cheap
Private cloud

- Harder to set up and operate
- Up-front cost and hosting cost
- Better integration with own infrastructure
- Fixed size
OpenStack

- Start with one server with OpenStack Folsom in 2012
- Added two servers two months later
- Add two servers
- Migrated local storage to shared Ceph storage
- Add three servers
- Upgraded from 1G to 4x1G
- Added NetApp NFS archive
- Add SSD caching layer in Ceph
- Migrated to 2x10G
- Add eight server servers
- **Same interface for users!**
Current setup

- 2x controller
- 1x mgmt server
- 6x performance nodes
  - CPU pinning, NUMA awareness, local storage
  - 2x8 Cores + 64GB ram
- 6x general purpose
  - 128GB ram
  - 2x14 cores
  - Ceph storage
  - SSD + SAS disks
- 2x10G
- NetApp filer with 48TB storage
Virtual infrastructure

- Three different uplinks (network zones)
- Different routing and floating IP model
- Each user gets their own sandbox with quotas
- Team projects get their own project
- Security groups by default closed
- Routes from and to all user networks within private address range
- Access with SSH or OpenVPN
Management
entity Project:  
"
" A personal sandbox project in OpenStack  
: param name The username and projectname to create  
: param email The email address of the user  
: param subnet_id The third octet of the project  
: param public_network Create a network in the public zone  
: param lab_network Create a network in the lab zone (they get their own router)  
: param exp_network Create a network in the experimental zone  
"

string name  
string email  
number subnet_id  
bool public_network  
bool lab_network  
bool exp_network

end  
index Project(name)  
index Project(subnet_id)  

Project personal_projects [0:] -- [1] neutron::Network lab_uplink  
Project personal_projects [0:] -- [1] neutron::Subnet lab_routing_subnet  
Project admin_projects [0:] -- [1] keystone::Project admin_project  
Project personal_projects [0:] -- [1] keystone::Project project  
Project personal_projects [0:] -- [1] vm::IaaS iaas  
Project personal_project_public_router [0:] -- [1] neutron::Router public_router  
Project personal_project_exp_router [0:] -- [1] neutron::Router exp_router
User setup

**implementation** personalProject **for** Project:

```
# define the user stuff
self.project = keystone::Project(iaas=self.iaas, name=name, enabled=true,
                                 description="Personal project of {{ name }} ({{ email }}))")
user = keystone::User(iaas=self.iaas, name=name, email=email, enabled=true,
                      password=password)

keystone::Role(project=self.project, user=user, role="Member")
keystone::Role(project=self.project, user=user, role="heat_stack_owner")
keystone::Role(project=self.project, user=keystone::User[name="admin"], role="admin")

keystone::Role(project=keystone::Project[name="openvpn"], user=user, role="Member")

std::User(host=self.iaas.access_vm, name=name, shell="/bin/bash")
```

end
Public network zone

implementation publicNet for Project:
  # define network and subnet
  net = neutron::Network(name="{{ name }}_net", project=self.project, self.iaas=iaas)

  subnet = neutron::Subnet(project=self.project, network=net, router=self.public_router,
    dhcp=true, name="{{ name }}_net",
    network_address="172.16.{{ subnet_id }}.0/24",
    iaas=self.iaas, allocation_start="172.16.{{subnet_id}}.10",
    allocation_end="172.16.{{subnet_id}}.254")

end
Lab network zone

implementation labNet for Project:

# Create a dedicated router and network for this demo environment
router = neutron::Router(name="router-{{name}}", iaas=self.iaas, project=self.project,
ext_gateway=self.lab_uplink)

net = neutron::Network(name="{{name}}_lab", project=self.project, iaas=self.iaas)
subnet = neutron::Subnet(project=self.project, network=net, router=router, dhcp=true,
        name="{{name}}_lab", network_address="172.17.{{subnet_id}}.0/24",
iaas=self.iaas, allocation_start="172.17.{{subnet_id}}.10",
allocation_end="172.17.{{subnet_id}}.254")

router_port = neutron::RouterPort(project=self.admin_project, iaas=self.iaas,
        name="{{name}}_lab_routing",
        address="172.18.0.{{subnet_id}}",
        subnet=self.lab_routing_subnet,
router=router, purged=false)

router.routes = neutron::Route(destination="172.16.0.0/16", nexthop="172.18.0.254")
router.routes = neutron::Route(destination="172.17.0.0/16", nexthop="172.18.0.254")
router.routes = neutron::Route(destination="172.19.0.0/16", nexthop="172.18.0.253")
neutron::Route(destination="172.17.{{subnet_id}}.0/24", nexthop="172.18.0.{{subnet_id}}",
router=self.public_router)
end
Experimental network zone

implementation expNet for Project:
  # define network and subnet
  net = neutron::Network(name="{{ name }}_exp", project=self.project, iaas=self.iaas)
  subnet = neutron::Subnet(project=self.project, network=net, router=self.exp_router,
                          dhcp=true, name="{{ name }}_exp",
                          network_address="172.19.{{ subnet_id }}.0/24",
                          iaas=self.iaas, allocation_start="172.19.{{subnet_id}}.10",
                          allocation_end="172.19.{{subnet_id}}.254")
end

implement Project using personalProject
implement Project using labNet when lab_network
implement Project using publicNet when public_network
implement Project using expNet when exp_network
Lessons learned
Self-service

- Getting money to change things is hard
- Got support from local IT
- Start small
- Documentation
- Demo!
- Training
- Pro-bono services
Elasticity

- (private) Cloud should be elastic
- Resource have to seem infinite
- But do enforce quotas
- Capacity management and planning
- Capacity problems encourages hoarding
Resource accounting

- Quotes help wasting resources
- Abandoned virtual machines
- 50 useful VM’s vs one VM trashing for weeks?
- Working on better monitoring
- Anomaly detection
- Public shaming
Contact

- bart.vanbrabant at inmanta.com
- @bartvanbrabant
- https://bart.vanbrabant.eu
- https://inmanta.com
- https://github.com/inmanta/inmanta
- @inmanta_com