Automation in the Cloud – Using Open Source Software

Michael Eichenberger
CEO stepping stone GmbH
Open Cloud Day
11th of June 2013

Agenda

- About us
- Why Cloud Management and Automation?
- Necessary Software Components
- Virtual Machine Life Cycle
- Creation of Virtual Machines
- Example Puppet Configuration
- Example Virtual Machine Creation
- Questions

About us

- Founded 9 years ago
- Clear focus on Cloud Computing
  - Software as a Service (SaaS)
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
- With emphasis on Managed Services
  - Backup, Maintenance, Monitoring and Standby Duty (all Services Platform-independent)
- Providing Solutions for Partners and SME's

Why Cloud Management and Automation?

- An increasing numbers of servers need to be operated easily and efficiently.
- New servers with pre-configured Software-Stacks need to be deployed in mere minutes.
- Deployed servers need to be updated automatically.
- Servers need to be automatically monitored.

Necessary Software Components

- Virtualization Environment (Cloud)
  - stoney cloud
- Configuration Management
  - Puppet
- Monitoring
  - Zabbix
- Glue (Provisioning)
  - Perl

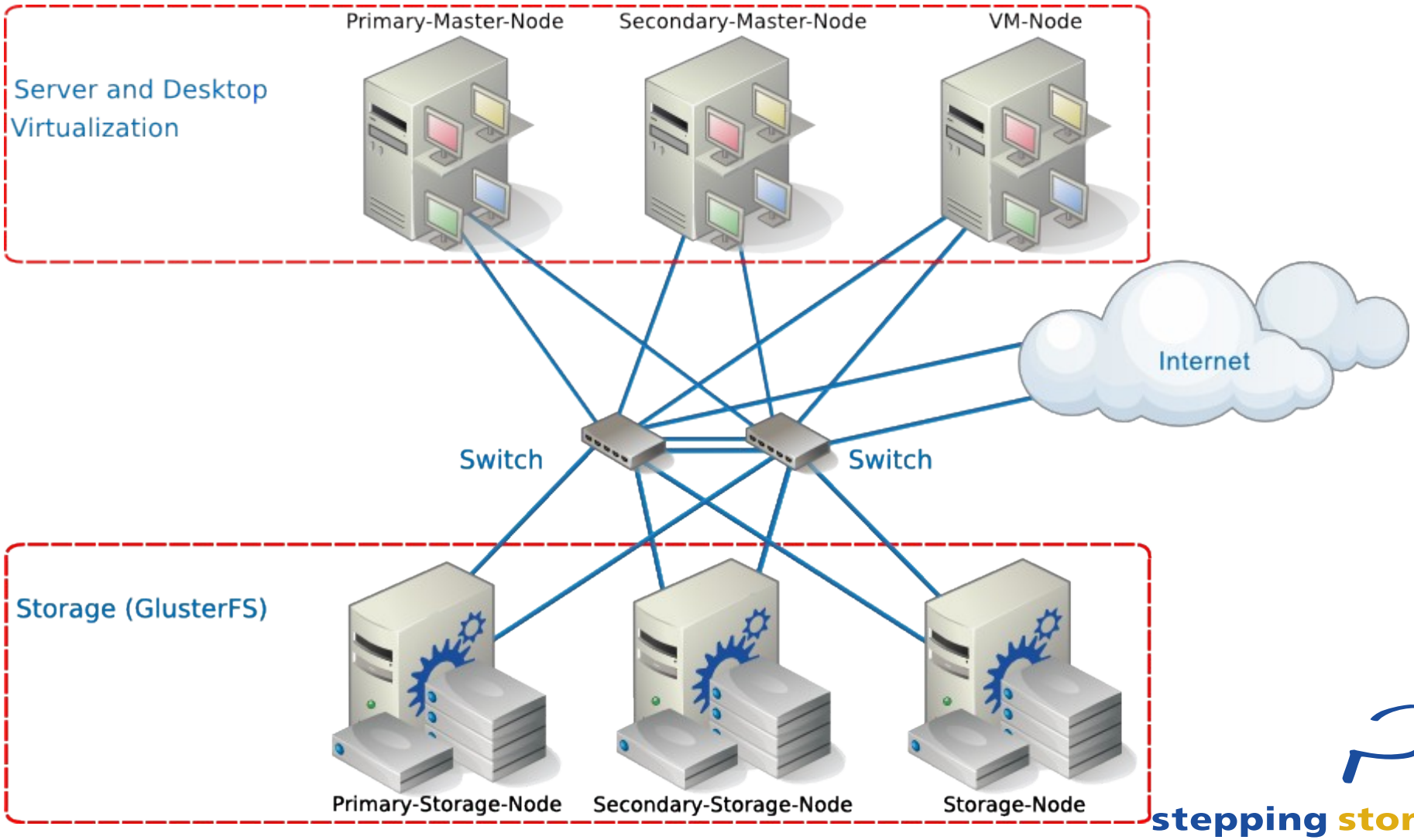
Why stoney cloud?

- Open Source Cloud Management solution with service providers as the target audience.
- Turnkey solution to build public and private clouds.
- Runs unmodified Windows and Linux servers as well as desktops on commodity hardware.
- Scales horizontally (VM- and Storage-Nodes).
- User friendly web interface to manage every aspect of your cloud.
- “Lean Provisioning”
- Expandable.

Lean Provisioning (block streaming)

- Creation of a virtual disk image with a backing store file and the subsequent "streaming" (or "pulling") of all blocks from the master disk.
- The virtual machine can be started as soon as the virtual disk image exists, even though the data isn't yet duplicated.

Overview stoney cloud



stepping stone

Why Puppet?

- Manages the configuration of Unix-like and Microsoft Windows nodes.
- Has a custom declarative language to describe system configurations.
- Has a resource abstraction layer, which enables administrators to describe the configuration in high-level terms.
- Can store node information externally with the help of External Node Classifiers (ENC).

External Node Classifier (ENC)

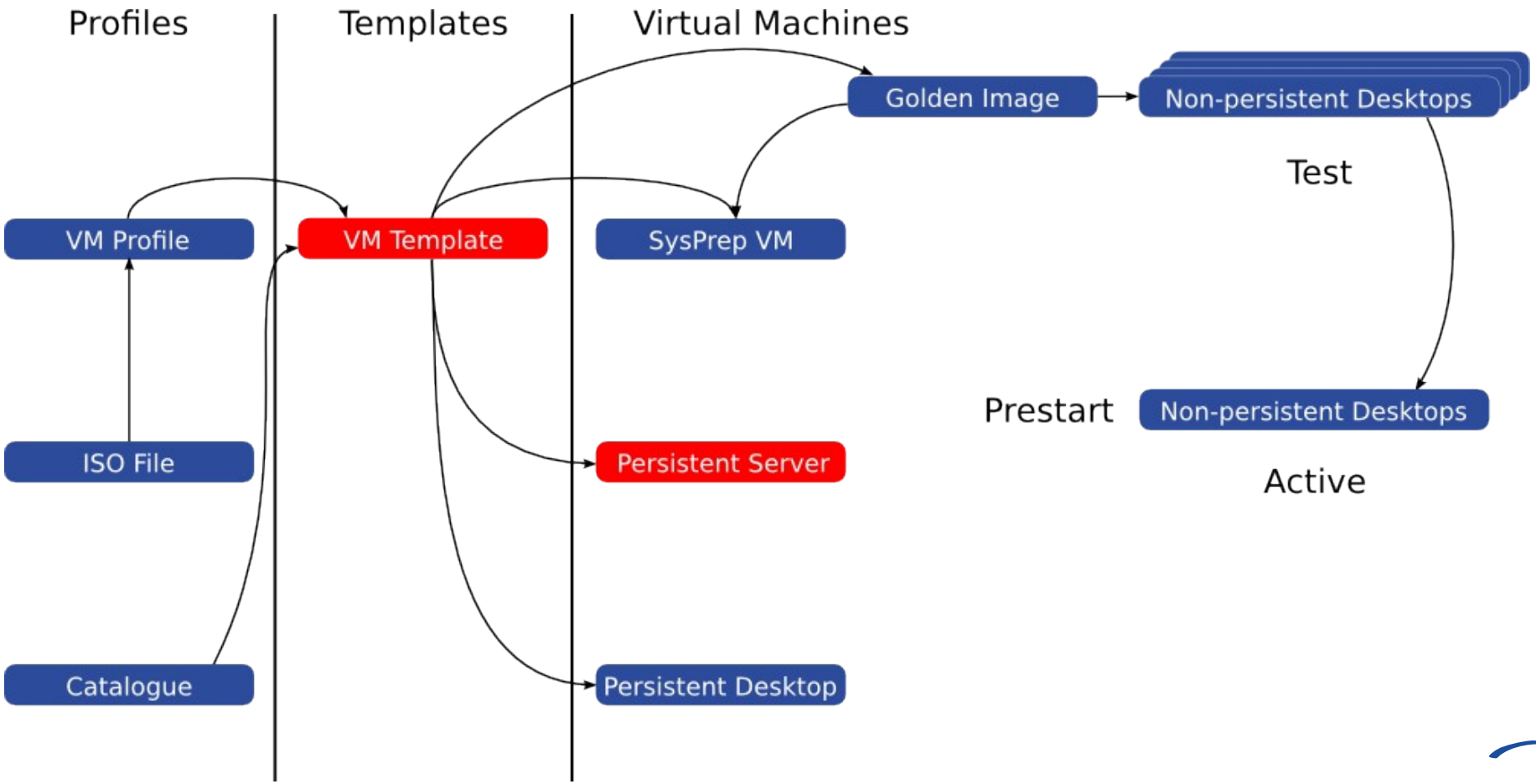
- External Node Classifiers are easy to write.
- They return classes, parameters and an environment based on a fully qualified domain name (fqdn) lookup.
- Parametrized classes are an important feature that the native LDAP ENC does not support.
- You ENC can support multiple LDAP leafs.

Why Zabbix?

- Supports Unix-like and Microsoft Windows operating systems.
- SNMP, IPMI, database monitoring, web-monitoring, ...
- Extensible with client and server scripts.
- Proxy servers (monitor restricted networks).
- Multi-tenant capable web interface.
- Has a JSON-RPC API for configuration, administration and monitoring.

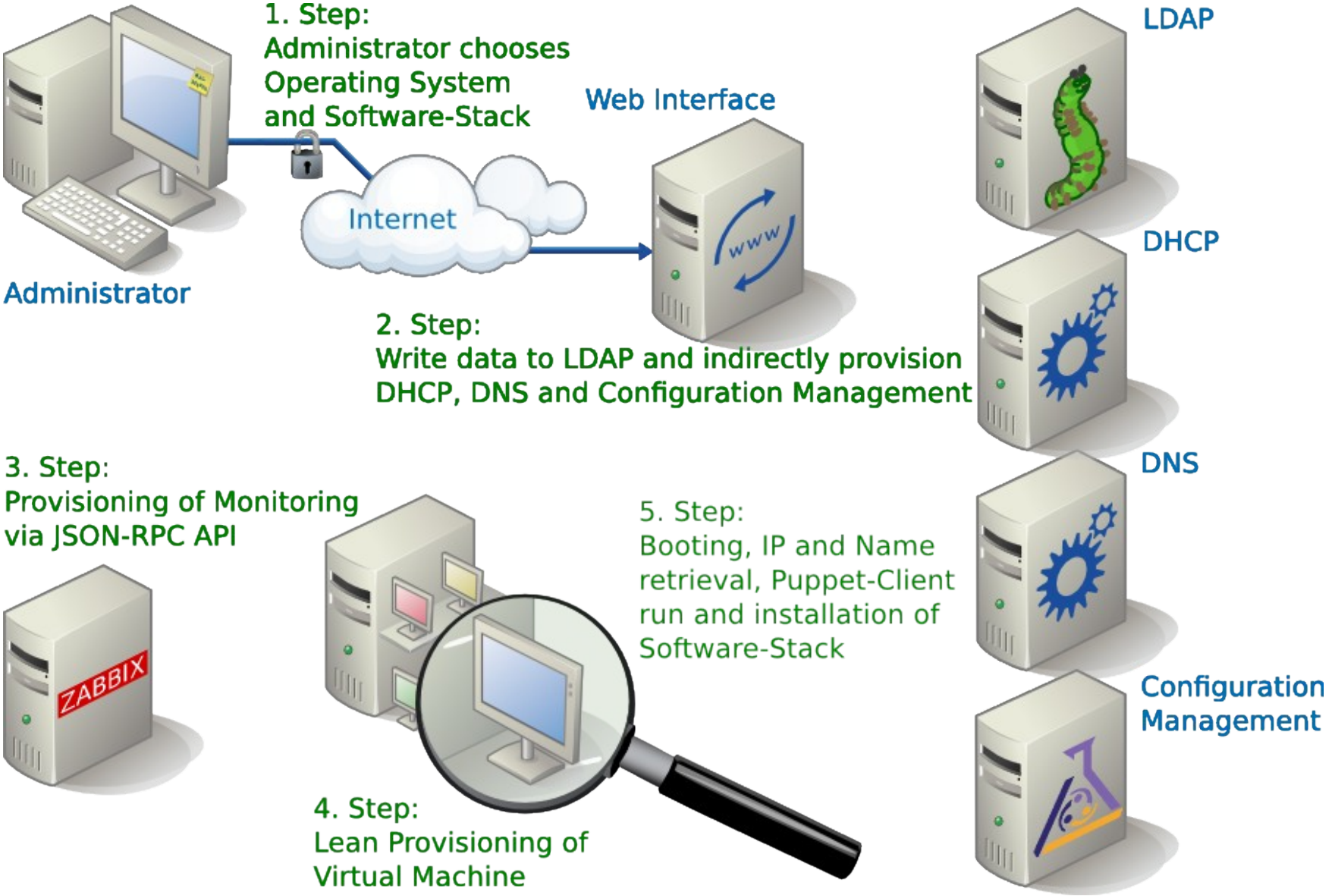
stepping stone

Virtual Machine Life Cycle



stepping stone

Creation of Virtual Machines



Example Puppet Configuration File System Structure

```
/etc/puppet/environments/
    production/
    testing/
    development/
        manifests/
        hieradata/
        modules/
            files/
            manifests/
                init.pp
                params.pp
                site.pp
                package.pp
            templates/
```

Example Puppet Configuration manifests/init.pp

```
class django {
    include stdlib
    class { 'django::package': }
}
```

Example Puppet Configuration manifests/package.pp

```
class django::package {
    include django::params
    package { $django::params::required_packages:
        ensure => present,
    }
}
```

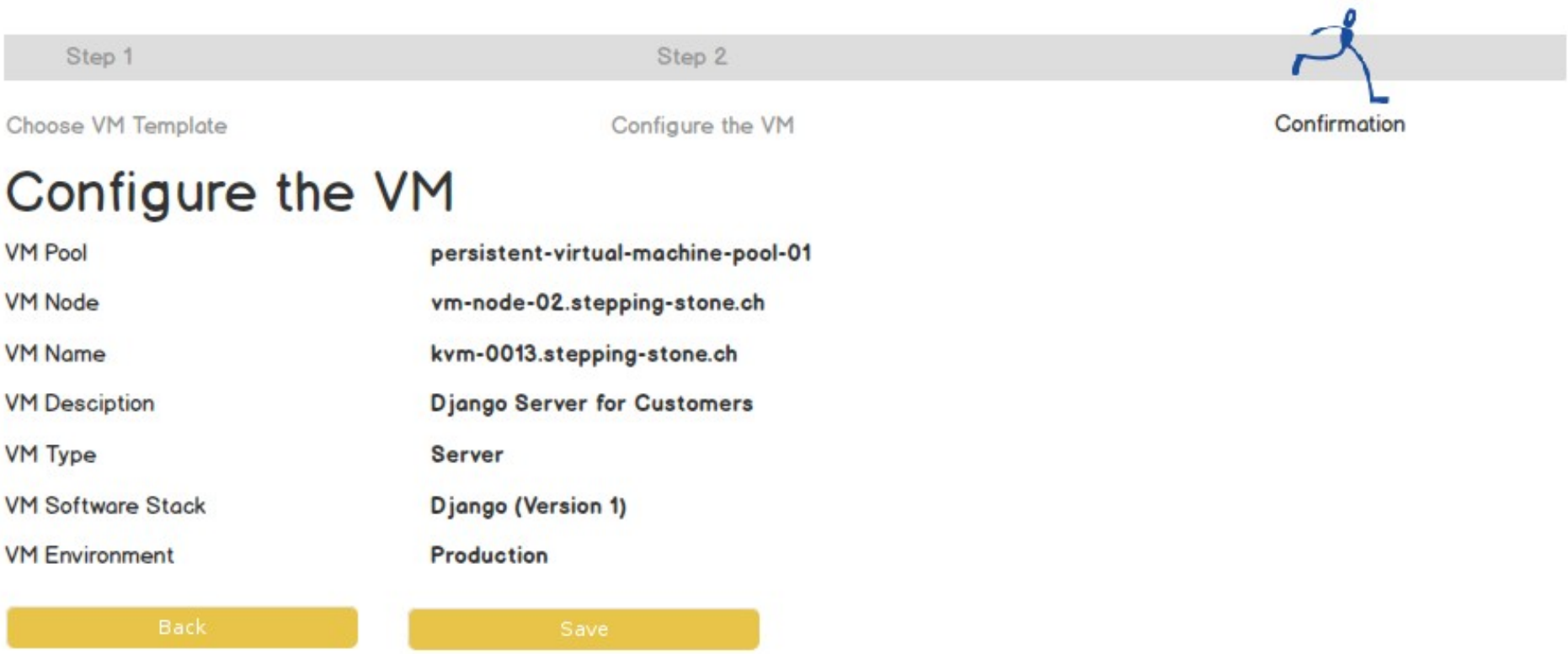
Puppet manifests/params.pp

```
class django::params {
    case $::operatingsystem {
        /Gentoo/: {
            $required_packages = ["django"]
        }
        default: {
            err("django::params not defined for os ${::operatingsystem}")
        }
    }
}
```

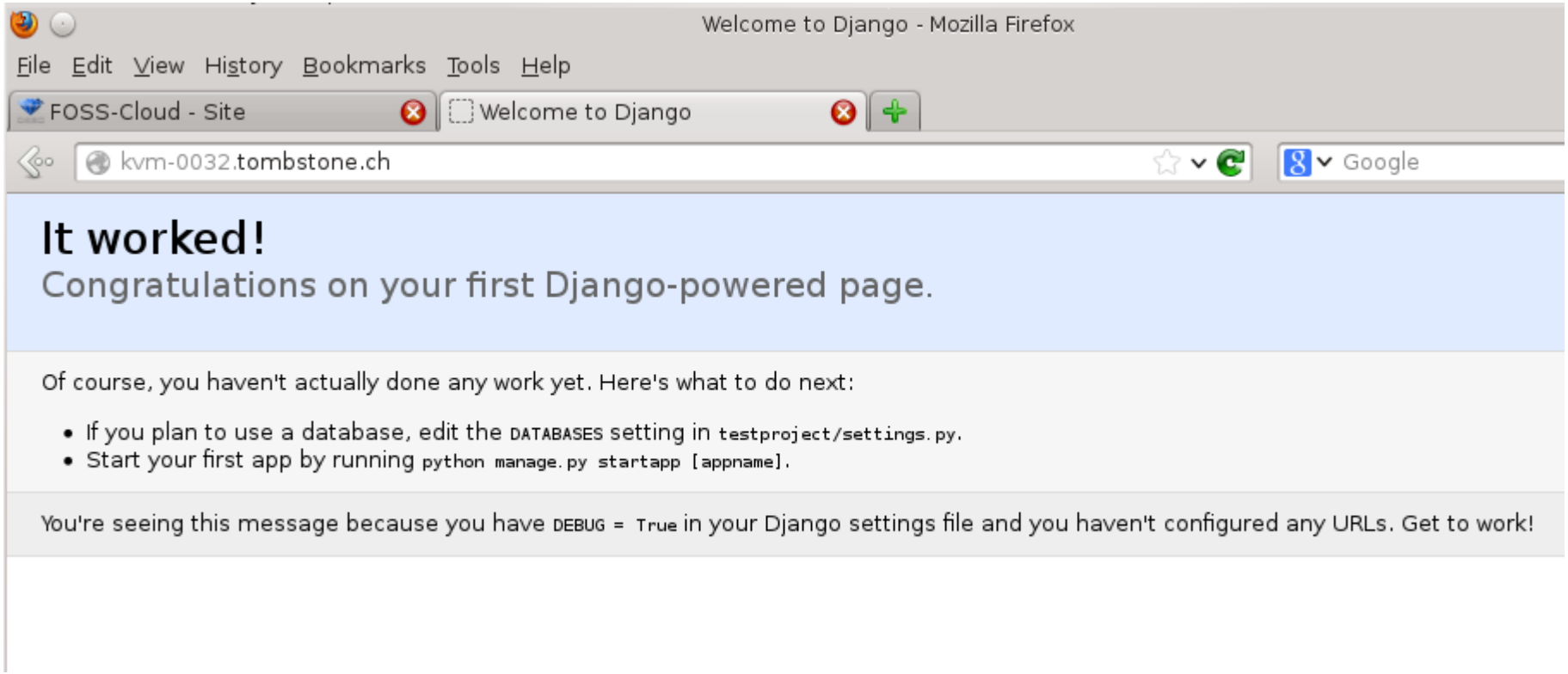
Example Puppet Configuration manifests/site.pp

```
define django::site(
    $user  = 'nobody',
    $group = 'nobody',
    $chdir = undef)
{
    Class['django'] -> django::site[$name]
    file { "/var/lib/$name":
        ensure  => directory,
        owner   => $user,
        group   => $group,
        mode    => '0640',
    }
    exec { "django-admin.py startproject $name /var/lib/$name":
        creates => "/var/lib/$name/manage.py",
        path    => [ "/usr/bin" ],
        user    => $user,
        group   => $group,
        require => File["/var/lib/$name"],
    }
```

Example Virtual Machine Creation with Django Framework



Example Virtual Machine Creation with Django Framework



stepping stone

Questions?

Links

- http://www.stoney-cloud.org/
- https://puppetlabs.com/
- http://www.isc.org/downloads/dhcp/
- https://www.powerdns.com/
- http://www.zabbix.com/
- http://www.stepping-stone.ch/

Contact

stepping stone GmbH
Neufeldstrasse 9
CH-3012 Bern

Phone: +41 31 332 53 63
www.stepping-stone.ch
info@stepping-stone.ch