



**Hewlett Packard
Enterprise**

Minimal First Machine in the DC

Preventing snowflakes

Andy McGuigan Simon McCartney

Background

- Responsible for building & maintaining small clouds to facilitate dev & test in the Helion OpenStack (HOS) platform group
- Dozens of 10-node environments
 - No supporting infrastructure in each environment
 - We treat each environment as an independent DC
 - SSH & IPMI (HP iLO) access only
- Rebuilding clouds on a daily basis
 - New HOS builds
 - Cloud model changes: more compute, LVM/Ceph, VSA etc
 - Damaged installs
- **Need to be able to build a cloud with zero human interaction**

The Challenge

- Helion OpenStack releases are self-contained ISOs
 - Boot from ISO to install Lifecycle Manager (first machine in your cloud)
- hLinux, HP's Debian Jessie based distro
 - Traditional Debian/Ubuntu console installer, asks for network details etc
- Cobbler/DHCP/PXE to bare-metal install rest of cloud nodes
- Ansible to install OpenStack, see <https://github.com/hpe-helion-os>
- **How do we automate that first machine in the data centre?**

ISO-BUILDER

An Ansible role to customize ISO images

- Takes an existing ISO image
- Injects in templated config files (Debian pre-seed config)
 - Network config: DHCP or static, NTP etc
 - SSH authorized_keys
 - Custom package install
 - Disk layout
 - Custom apt mirrors
- **Zero Keypress ISO images**
- Uses mkisofs to build new ISO image
 - Legacy BIOS & UEFI support
- Support for Debian, hLinux, Ubuntu, DBAN
- <https://github.com/simonmcc/iso-builder>

ISO-BUILDER

Static Network Config Example

```
d-i netcfg/disable_autoconfig boolean true
```

```
d-i netcfg/get_ipaddress string {{ iso_ipaddress }}
```

```
d-i netcfg/get_netmask string {{ iso_netmask }}
```

```
d-i netcfg/get_gateway string {{ iso_gateway }}
```

```
d-i netcfg/get_nameservers string {{ iso_nameserver }}
```

```
d-i netcfg/confirm_static boolean true
```

ISO-BUILDER

Using the iso-builder role in a playbook

```
---
```

```
- name: Build Custom ISO
  hosts: all
  gather_facts: false
  sudo: true
```

```
vars:
  iso_distro: ubuntu
  iso_version: "14.04.3"
  iso_distro_flavor: server
  iso_host: "releases.ubuntu.com"
  iso_basename: "ubuntu-{{ iso_version }}-{{ iso_distro_flavor }}-amd64"
  iso_url: "http://{{ iso_host }}/{{ iso_version }}/{{ iso_basename }}.iso"
  iso_static_network: true
  iso_ipaddress: "{{ ansible_ssh_host }}"
  iso_netmask: "255.255.255.0"
  iso_gateway: "{{ mgmt_gateway }}"
```

```
roles:
  - { role: iso-builder }
```

ISO-BUILDER

Boot from IPMI virtual media, in Ansible

We use *python-hpilo* and *ansible-provisioning* to manipulate physical servers

- <https://pypi.python.org/pypi/python-hpilo>
- <https://github.com/ansible-provisioning/ansible-provisioning>

```
- name: HP iLO | Power Off
  local_action: command hpilo_cli "{{ ilo_ip }}" \
                set_host_power host_power=false

- name: HP iLO | Boot of Virtual Media (once only)
  local_action: hpilo_boot host="{{ ilo_ip }}" \
                  media='cdrom' image="{{ ilo_iso_url }}" \
                  state='boot_once' force='yes'
```

ISO-BUILDER

Demo Video!

ISO-BUILDER

Development & Testing

- Vagrant based workflow, enabling OSX based development of Linux ISO images
- ‘Disposable’ vagrant box used to build ISO images

```
vagrant up iso-builder-ubuntu
```

```
vagrant provision iso-builder-ubuntu
```

- Test box configured to boot off ISO

```
vagrant up boot-from-iso
```

Recap

- Even your first machine doesn't have to be a snowflake
- Automated ISO customization & generation
- Use existing IPMI/iLO tooling to trigger (re-)installs
- Use custom DBAN ISO to destroy data
- Vagrant for development & testing



**Hewlett Packard
Enterprise**

Thank you

@amcguign
@simonmcc