cloud-init

Cloud and Virtualization Workshop

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Rationale

- Clouds usually give you a "pre-installed" OS image to clone
- But you may need to customize it before you can login
  - Create users and passwords and/or upload SSH public keys
  - Set the hostname
  - You don't want these things hard-coded in the OS image!
- Need to grow the partition table/filesystem to virtual disk size
- You may want to automate further changes
  - e.g. install a set of packages that you always need, or config files
- **Cloud-init** provides a standardized mechanism for this

What is cloud-init?

- A software package which is run *early on* in the boot process
- Written in Python
- Adopted by all the major (Linux) OS vendors
- Picks up configuration information from the cloud environment
- Runs *modules* which perform configuration actions
- Documentation is improving

Configuration information

- Built-in "data sources" for many different cloud vendors
  - e.g. EC2 data source: makes HTTP request to 169.254.169.254
  - auto-detects which data source to use
- There is a "Nocloud" data source for standalone environments
  - Reads a virtual floppy disk or CD-ROM drive
  - Can be vfat (MS-DOS) or iso9660 (CD-ROM) filesystem
- Fetches various configurations: all are optional
  - *metadata* (instance-data)
  - network-config
  - user-data, vendor-data

metadata → instance-data

- Metadata: information supplied by the Cloud itself about the VM
- Cloud-specific
  - Could just be an instance ID
  - Clouds often provide additional info
    - e.g. OS image, region, instance name, tags, …
- Cloud-init collects metadata from the various clouds and structures it into a standardized form called "instance-data"
- You can query it to control later actions

network-config

- As you'd imagine: specifies how to configure the network
  - IP addresses, gateway, DNS server, DNS search…
- Typically in cloud is DHCP on primary network interface
- Self-hosting: you can configure static IP

```
version: 2
ethernets:
  eth0:
    addresses:
      - 100.64.0.99/22
    routes:
      - to: default
        via: 100.64.0.1
    nameservers:
      search: [ws.nsrc.org]
      addresses: [100.64.0.1]
```

user-data

- This is the main part that controls instance initialization
- There are different formats of user-data
  - https://cloudinit.readthedocs.io/en/latest/explanation/format.html
- At simplest, it's a shebang line plus a script

```
#!/bin/sh
echo "Hello World" > /var/tmp/output.txt
apt-get update
apt-get -y install apache2
```

cloud-config

- The most commonly seen form of user-data
- Configures modules to be run by cloud-init. YAML format.
- Must start with **#cloud-config**

```
#cloud-config
fqdn: srv1.example.com
users:
  - name: sysadm
    passwd: $6$XqBb4pf3$rTN75u32r30VDbY252DwLLJ0rAuxI…
    shell: /bin/bash
write_files:
  - path: /etc/sudoers.d/10-sysadm
    content: |
      sysadm ALL=(ALL:ALL) NOPASSWD: ALL
```

Other formats of user-data

- Less commonly seen, see the documentation if interested
  - MIME-multipart, cloud-config-archive, jinja2 templates…
- Note: older OS images may have older versions of cloud-init, which don't have all the latest features

vendor-data

- vendor-data is just like user-data, but intended to carry a set of defaults which can be added to or overridden by user-data
- For example, vendor-data can set a common set of users or security policies
- vendor-data and user-data are merged (see docs for details)
  - user-data takes preference
  - user-data can disable any sections of vendor-data it doesn't want

When does it run?

- Some parts of cloud-init run on first boot only, some on every boot
  - "per-instance" and "per-boot" modules
  - If this matters to you, read the documentation!
- *network-config is done on every boot*
- If you manually change the network config, then you need to disable cloud-init from overwriting your changes

```
echo "network: {config: disabled}" > \
  /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

Checking cloud-init success/fail

- From CLI

```
cloud-init status
cloud-init status --format json
cat /var/log/cloud-init-output.log   # also sent to console
```

- Note: cloud-init may still be running at the time you login

```
cloud-init status --wait [--format json]
```

- For very detailed debugging

```
cat /var/log/cloud-init.log
```

Preparing template images

- You might make a template by booting a VM, customizing it by hand (or script), shutting down, and then cloning from it
- You want to make sure cloud-init "forgets" that it has already booted, and runs again when the clone boots

```
cloud-init clean --logs --machine-id --seed -c all
# then shutdown, e.g. halt -p
```

- You might still need to remove some things by hand (e.g. logs)
  - Note that older versions of cloud-init clean didn't remove sshd host keys or network configuration

cloud-init labs