Cloud-init: The cross-cloud magic sauce.

Scott Moser / Chad Smith
smoser / blackboxsw @
FreeNode #cloud-init
OSS Europe ’18: Edinburgh
What is cloud-init?

Cloud Init is cross platform cloud instance initialization software.

That didn’t help. Maybe tell me what problem it solves?
Why is cloud-init?

Basic instance initialization.
Why is cloud-init?

Ahh. That’s better.
Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Into This
Cloud Instance Makeup

Inputs to a cloud instance.
● Disk Image
● Meta-data
● User-data
● Vendor-data

Each cloud platform provides in different format in a different manner (disk, network, serial device ...) 

Cloud-init commonizes.
Disk Image

Just a container for blocks.
How do you get one?
● Use Existing.
● Make your own.
● Download.

https://download.opensuse.org/repositories/Cloud:/Images:
https://cdimage.debian.org/cdimage/openstack/
https://cloud-images.ubuntu.com/
https://alt.fedoraproject.org/cloud/
Meta-data

● Information provided by the cloud platform
● Examples:
  ○ Hostname
  ○ Authorization information (ssh public keys)
  ○ Network information
Network Configuration

Initially network was simple: dhcp on eth0

Now clouds have multiple nics, multiple IPs, ipv4 and ipv6, bonds, bridges, vlan.

Each cloud describes network differently. Cloud-init renders to OS specific network config:
  • /etc/network/interfaces
  • Netplan
  • Sysconfig
  • Arch, FreeBSD
User-data

Simple

```bash
#!/bin/sh
echo Hi Mom
```

Fancier

```bash
#cloud-config
packages: [pastebinit]
runcmd:
  - echo Hi Mom | tee /run/greeting.log | pastebinit
```

Vendor Data
User-Data Uses

Enable automation and integration.
Specific configuration language for:

<table>
<thead>
<tr>
<th>Add Packages and Upgrade</th>
<th>Configure LXD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Hostname</td>
<td>Add Users and Groups</td>
</tr>
<tr>
<td>Add SSH Keys</td>
<td>Partition Disks</td>
</tr>
<tr>
<td>Run Arbitrary Code</td>
<td>Grow the root Partition</td>
</tr>
<tr>
<td>Start Puppet or Chef</td>
<td>Phone Home</td>
</tr>
<tr>
<td>Timezone / Locale</td>
<td>Mirror Selection</td>
</tr>
</tbody>
</table>
Cloud Init Goals

**Never** need to make a custom image or reboot.
- Execute code at defined points in boot:
  - asap, network-up, “final”

Support custom image via image capture workflow.
- Execute code on boot **once (ever), per-instance**, or per-boot.
- Boot your instance, modify things, snapshot, re-use.

OSes should have to only ship a single image.
Multi-distro Support
Multi-cloud Support
Cloud-init 18.4/18.5 features

- **(18.5) Hotplug network configuration**
  - Automatically apply network config based on metadata/udev events
- **(18.4) Standardized instance metadata on all clouds and distros**
  - `/run/cloud-init/instance-data.json`
- **(18.4) Extended command line tooling**
  - `cloud-id/cloud-init query`: emit specific instance-data attributes. Report canonical cloud name and region
  - `cloud-init analyze`: detailed analysis of boot-time operations
  - `cloud-init status`: block until cloud-init successfully completes
Handoff to existing Chef management

```yaml
#cloud-config
chef:
  install_type: "omnibus"
  omnibus_version: "12.3.0"
  node_name: "your-node-name"
  server_url: "https://chef-server.you"
  run_list:
    - "recipe[apache2]"
    - "role[frontend]"
```
Handoff to existing Puppet management

#cloud-config
puppet:
  conf:
    agent:
      certname: puppet-agent-{{region}}.mydomain
      server: puppetmaster.mydomain
    ca_cert: |
      -----BEGIN CERTIFICATE----- ...

Handoff to existing SaltStack management

```yaml
#cloud-config
salt_minion:
  conf:
    master: salt.example.com
  grains:
    role:
      - web
  public_key: |
    -----BEGIN PUBLIC KEY------ ...
  private_key: |
    -----BEGIN PRIVATE KEY------ ...
```
Templates allow for one #cloud-config

## template: jinja
#cloud-config
{% set HN='oss-' ~ v1.platform ~ '-' ~ v1.region ~ '-' ~ range(9) | random %}
puppet:
  conf:
    agent:
      server: puppetserver.blackboxsw.com
      certname: {{ HN }}
  hostname: {{ HN }}
...
{% if v1.region == 'us-east-2' and v1.cloud_name == 'aws' -%}
echo 'Installing custom proxies ...'
{%- endif %}
Cloud-init at work [demo]

- Standardized Instance-data available from cloud-init
  - [https://asciinema.org/a/206773](https://asciinema.org/a/206773)

- Deploying puppet across clouds and Linux distributions
  - Live demo + asciinema
Thanks, Questions, Contact

Thanks!

Questions?

IRC: Freenode #cloud-init
    smoser, blackboxsw

Email:
    ○ Scott Moser <scott.moser@canonical.com>
    ○ Chad Smith <chad.smith@canonical.com>

Mailing List: cloud-init@lists.launchpad.net