Xen Shell - Feature # 208: Allow customers to provide their own cloud-init data for cloud-init installs such as

Status:	New	**Priority:**	Normal
Author:	admin	**Category:**	
Created:	2022-04-29	**Assigned to:**	
Updated:	2022-05-03	**Due date:**	
Subject:	Allow customers to provide their own cloud-init data for cloud-init installs such as Ubuntu 22.04		

Description:

The current implementation of cloud-init installs at bitfolk simply asks the customer for the minimum required information (hostname and password) and then proceeds with an automated install.

It would be possible to allow customers to provide their own cloud-init data so they could have a repeatable and automated customised install.

h2. cloud-init data

This data consists of three files in YAML format. At the moment these are stored on the Xen Shell side as template files and the things between @[%@ and @%]@ markers are directives in that templating language.

After the files are templated they are shoved into a VFAT disk image and attached to the customer's VM as it boots. The cloud-init package finds the filesystem by label (@CIDATA@) and looks inside it for its instructions.

h3. @meta-data@

This just provides a unique identirier within BitFolk's cloud for this VM. As such this probably doesn't need to be user-supplied.

h3. @network-config@

The VM's network configuration in "netplan":https://netplan.io/ format.

Example template:

<pre>
version: 2
ethernets:
  eth0:
    dhcp4: false
    dhcp6: false
    addresses: [[% ip_list %]]
    gateway4: '85.119.80.1'
    gateway6: '[% ip6_gateway %]'
    nameservers:
      addresses: ['85.119.80.232','85.119.80.233','2001:ba8:1f1:f205::53','2001:ba8:1f1:f206::53']
</pre>

h3. @user-data@

The main customisation of the operating system takes place here. Very little of what is done here is actually essential, they're just niceties for running on BitFolk's network.

Example template:

```
<pre>
users:
   - name: 'ubuntu'
     lock_passwd: false
     shell: '/bin/bash'
     groups: 'sudo'
[% IF ssh_public_keys.size -%]
     ssh_authorized_keys:
[%  FOREACH item IN ssh_public_keys -%]
       - '[% item %]'
[%  END -%]
[% END -%]

chpasswd:
   list:
      - 'ubuntu:[% crypted_passwd %]'
   expire: false

timezone: 'Etc/UTC'

bootcmd:
   - [ cloud-init-per, instance, wipefs-swap, wipefs, -a, /dev/xvdb ]

disk_setup:
   /dev/xvdb:
      table_type: 'mbr'
      layout:
         - [ 100, 82 ]
      overwrite: true

fs_setup:
   - label: SWAP
     filesystem: swap
     device: '/dev/xvdb'
     partition: 1

mounts:
   - [ 'LABEL=SWAP', 'none', 'swap', 'sw', '0', '0' ]

write_files:
   - path: '/root/bitfolk_post_install.sh'
     permissions: '0755'
     content: |
       #!/bin/sh

       echo "$0 Starting BitFolk cloud-init post-install scriptâ€¦"

       echo "$0: Enabling swap"
       swapon -a

       echo "$0: Disabling online discardâ€¦"
       sed -i 's/discard,//' /etc/fstab
```

```
    echo "$0: Removing EFI boot mountpointâ€¦"
    sed -i '/^LABEL=UEFI/d' /etc/fstab

    echo "$0: Disabling time-based fsckâ€¦"
    for DEV in $(blkid -s TYPE -t TYPE=ext2 -o device) \
            $(blkid -s TYPE -t TYPE=ext3 -o device) \
            $(blkid -s TYPE -t TYPE=ext4 -o device); do
      echo "    $DEV"
      tune2fs -i 0 "$DEV"
    done

    echo "$0: Removing /etc/default/grub.d/50-cloudimg-settings.cfgâ€¦"
    rm -f /etc/default/grub.d/50-cloudimg-settings.cfg

    echo "$0: Unhiding GRUB boot menuâ€¦"
    sed -i -e 's/GRUB_TIMEOUT_STYLE=hidden/GRUB_TIMEOUT_STYLE=menu/' /etc/default/grub

    echo "$0: 5 second GRUB timeout.."
    sed -i -e 's/GRUB_TIMEOUT=0/GRUB_TIMEOUT=5/' /etc/default/grub

    echo "$0: Removing 'quiet splash' from kernel cmdlineâ€¦"
                                        sed     -i     -e     's/GRUB_CMDLINE_LINUX_DEFAULT="quiet
splash"/GRUB_CMDLINE_LINUX_DEFAULT="console=hvc0"/' /etc/default/grub
    update-grub

    echo "$0: Fixing up /etc/apt/sources.list for apt-cacherâ€¦"
                  sed  -i  -e  's#http://\(security.ubuntu.com/\)#http://apt-cacher.lon.bitfolk.com/ubuntu/\1#'
/etc/apt/sources.list
                sed  -i  -e  's#http://\(archive.ubuntu.com/\)#http://apt-cacher.lon.bitfolk.com/ubuntu/gb.\1#'
/etc/apt/sources.list
    apt-get update
    echo "$0: Done."

# Hostname is set here as "fqdn:" happens on every boot.
runcmd:
  - [ hostnamectl, hostname, [% fqdn %] ]
  - /root/bitfolk_post_install.sh
</pre>
```

The SSH public keys come from those stored in the Panel.

h2. Interested in this feature?

If you're interested in seeing this happen, I would first suggest that you vote this issue up and follow it.

Next could you think about how you would like to get your @network-config@ and @user-data@ in to the Xen Shell?

Your @user-data@ could be pointed to by URL, but in that case the @network-config@ would have to be stock in order to get the network set up.

The Xen Shell could be told to download a copy of these files by URL.

I suppose copy and paste is an option, but that seems rather unweidly and error-prone, especially for a file format that is so sensitive to indentation (YAML).

The maximum size of all cloud-init data is 16KiB.

History