Building KVM virtual machines quickly with virt-builder and cloud-init

Akron Linux Users Group

Michael Meffie

January 6, 2021

Objectives

Show how to quickly create local kvm guests with pre-built images.

- vagrant
- cloud-init
- virt-builder

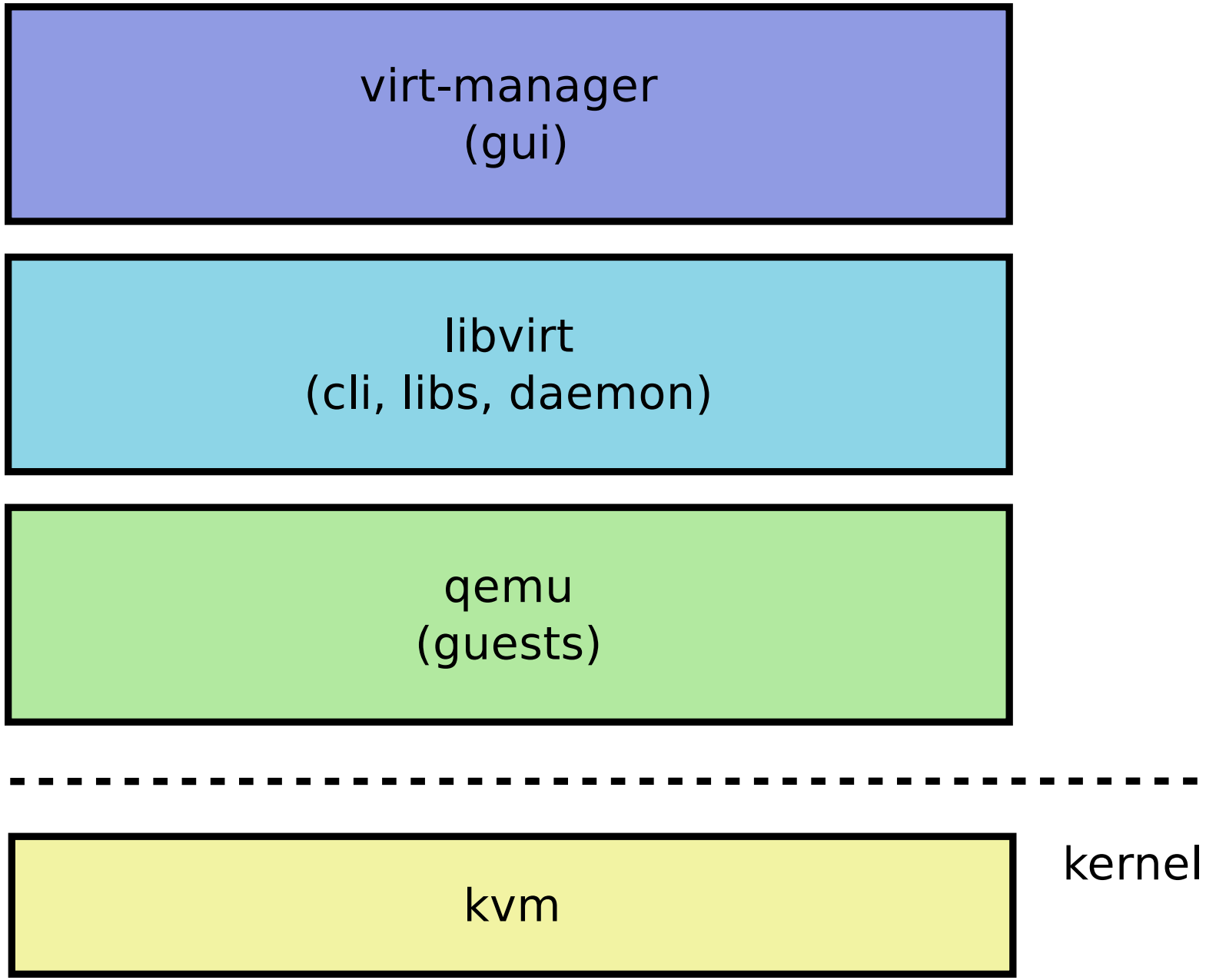
Hardware Virtualization Support

- Processor support required and must be enabled.
- See `/proc/cpuinfo`
- Intel: `vmx` flag
- AMD: `svm` flag
- Also, be sure virtualization is enabled in BIOS!

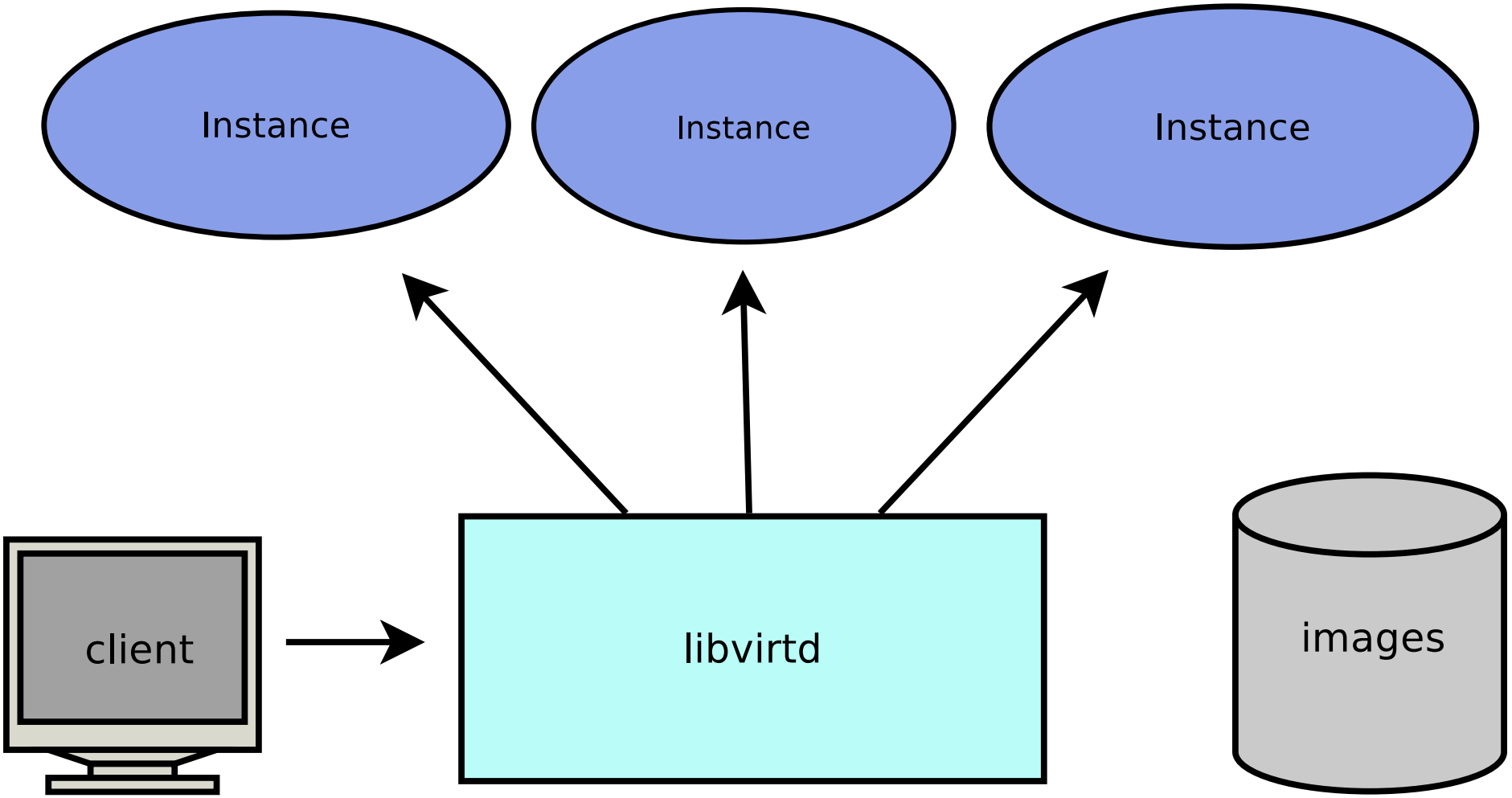
Example:

```
$ grep vmx /proc/cpuinfo
```

KVM Software Stack



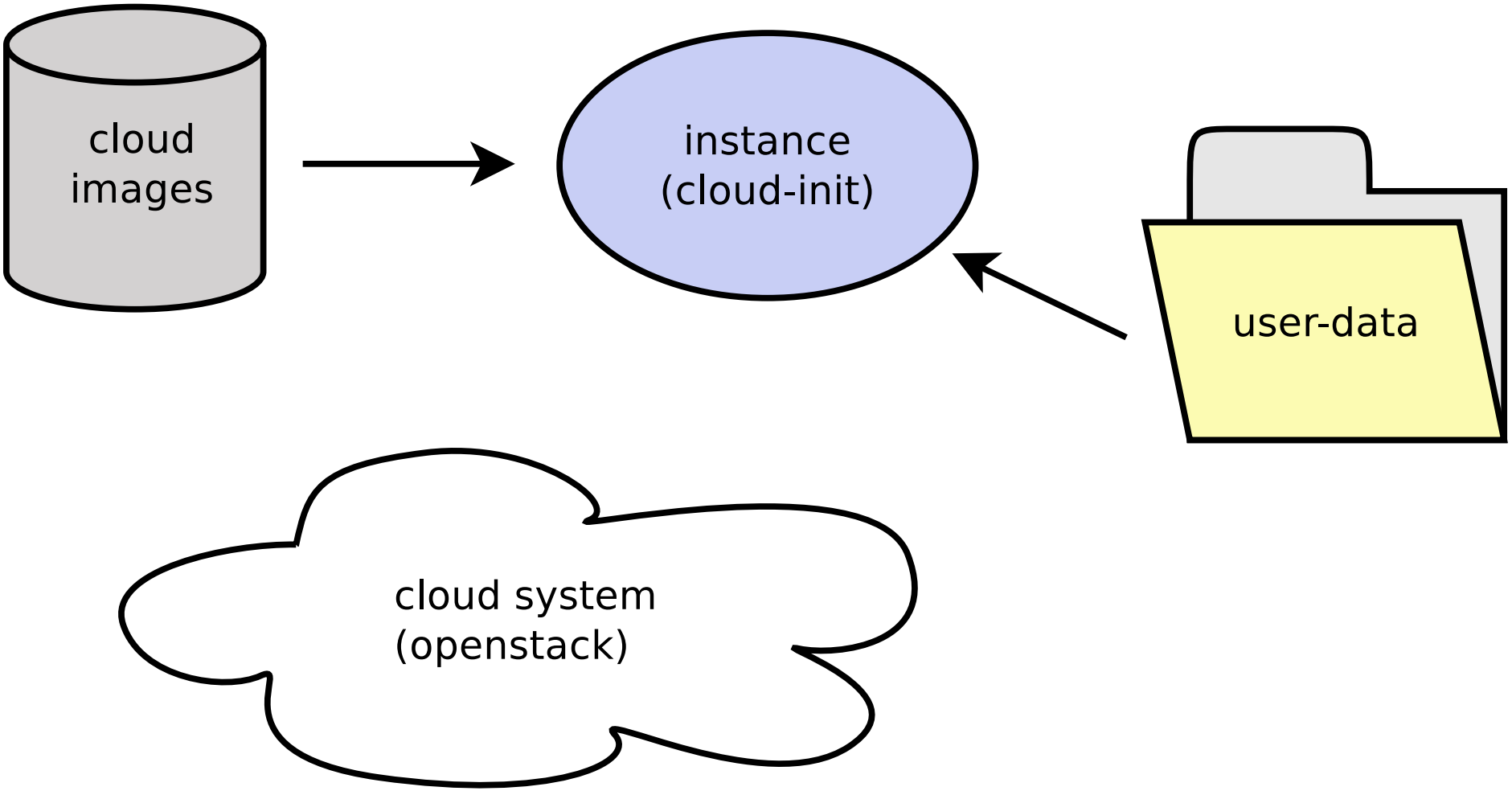
Libvirtd



Vagrant

- HashCorp tool for spinning up guests.
- Supports a variety of providers
- Linux provider is experimental, but works
- Search for libvirt provider boxes
- Do not install vagrant with apt or yum!
- Ruby based

cloud-init


cloud
images
instance
(cloud-init)
user-data
cloud system
(openstack)

cloud-init

- Widely used for depoying images on cloud providers
- Many images available online
- Boots quickly
- No root passwords in cloud-init images
- Data is injected on first boot using link-local addresses
- NoCloud option available for regulare libvirt/kvm setup

cloud-init

- Download a cloud-init enabled image
- Create a virtual cdrom disk with the cloud-init meta-data
- Create a guest with the disk attached
- cloud-init will use the data from attached disk
- See kvm-install-vm shell script

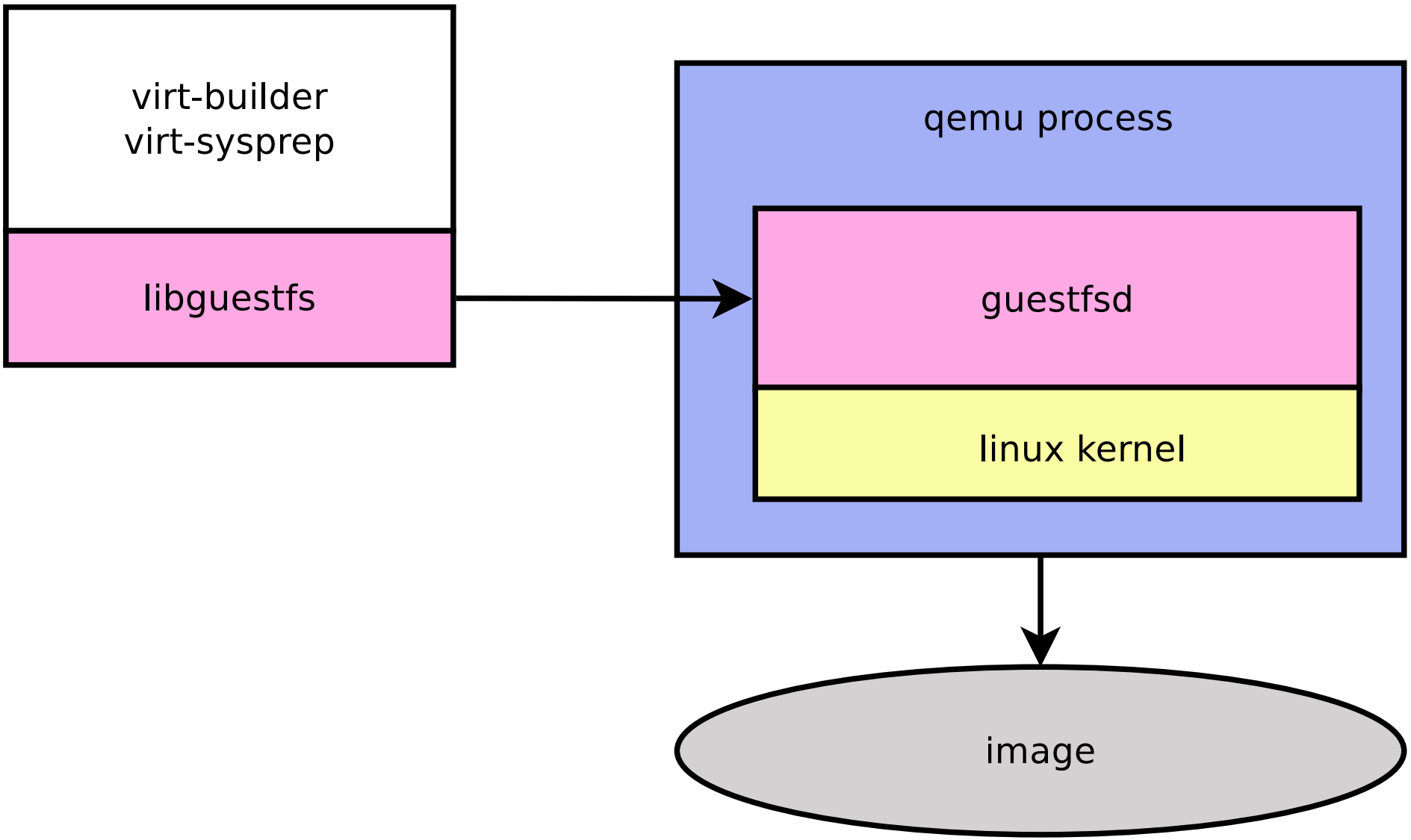
cloud-init

```
$ cat meta-data
instance-id: iid-local01;
local-hostname: cloudimg;

$ cat user-data
#cloud-config
password: passw0rd
chpasswd: { expire: False }
ssh_pwauth: True

$ genisoimage -output seed.iso
              -volid cidata
              -joliet
              -rock user-data meta-data
```

virt-builder

- libguestfs tool suite
- virt-builder - download and customize image
- virt-sysprep - edit images clones
- virt-up - homebrew virt-builder/virt-sysprep wrapper

virt-builder



Demos

- kvm-install-vm
- virt-up