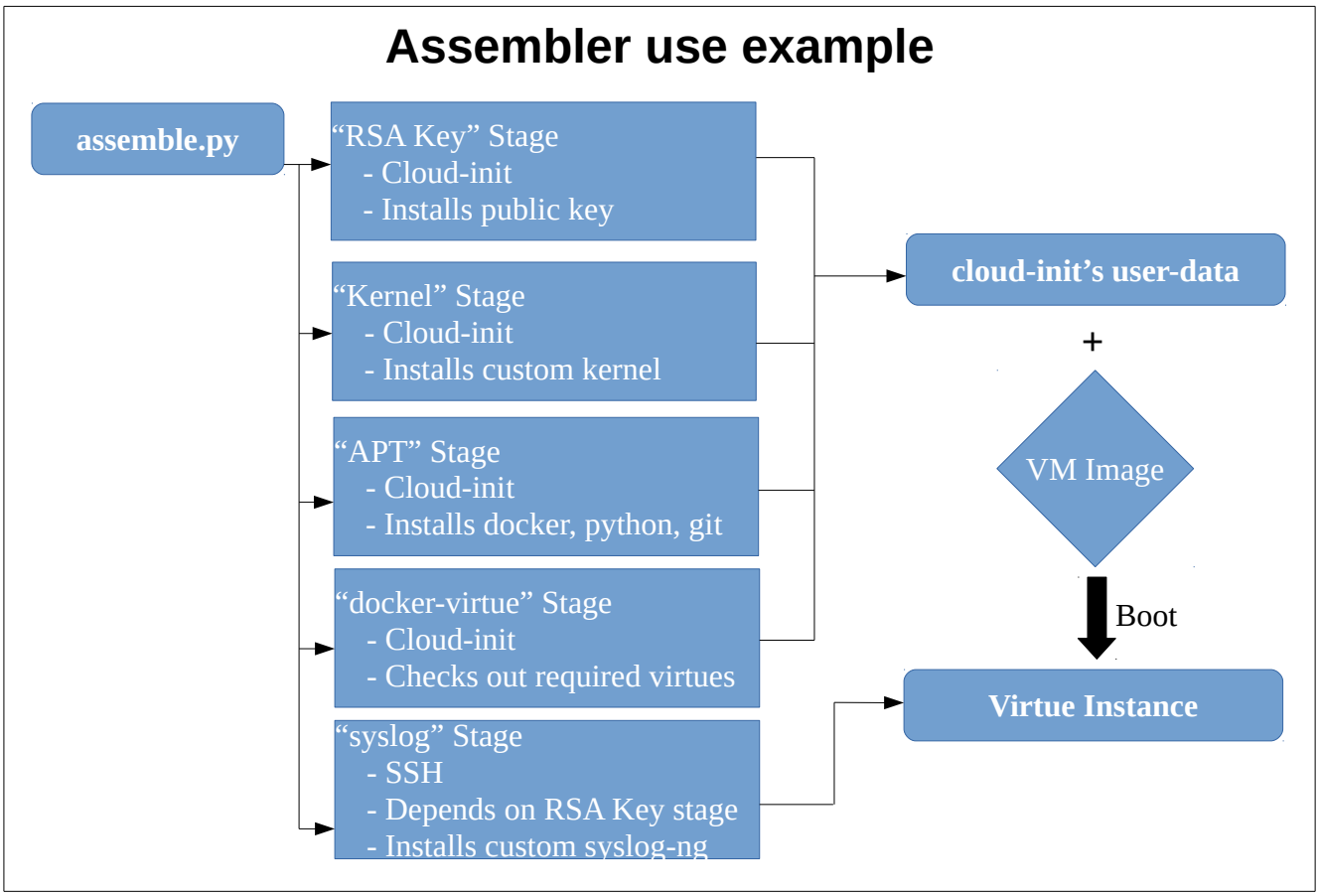
Virtue Assembler Design Doc

The goal of Virtue Assembler is to take a compatible VM image and install all required Virtue components into it. This is including, but not limited to – custom kernel, syslog-ng with custom modules, docker, a series of docker containers developed for this program, and authorized public keys.

The process is multi-staged, and allows for addition of more stages as well as stage dependency. The stages can use either the cloud-init method, such that new configuration of the VM will be stored in a data file and applied to the VM on boot. Or the SSH method, such that a running VM will be configured on the fly using an ssh connection. Both methods have its ups and downs. Cloud-init does not rely on the network, but can not transfer large files. SSH relies on the network, but can transfer large files.

Upon running the assembler, cloud-init files will be generated. Once a VM is up and running with the completed cloud-init config, the SSH stages are executed.



Stage design

Each stage needs to be able to tell its name, its type (cloud-init or ssh) and if it depends on other stages. Each stage is a python class that inherits either from CIStage or SSHStage classes with defined NAME and DEPENDS variables.

CIStage provides an interface to easily configure cloud-init's modules such as write files, or add users.

Overall design

The entry point (assember.py) needs to collect all required data, then run all of the CIStage based stages, wait for the resulting VM to appear available over the SSH connection, wait for VM's cloud-init scripts to finish, and run all of the SSHStage based stages.