Run KVM CI Tests on Avocado

Step 1: Install Avocado-Framework and Avocado-VT Plugin

Install via python-pip:

```
pip install --user avocado-framework
pip install --user avocado-framework-plugin-vt
```

They will be installed into ~/.local and the avocado binary file can be found in ~/.local/bin.

So you may need to add ~/.local/bin to PATH environment variable:

```
export PATH=~/.local/bin:$PATH
```

Step 2: Bootstrap Avocado-VT

Try to run the avocado vt-bootstrap command:

```
avocado vt-bootstrap --vt-type qemu --vt-skip-verifiy-download-assets --yes-to-all
```

This command will choose the qemu test backend and skip over verifying and downloading the JeOS.qcow2.xz. If you want to specify other guest, you can add **--vt-guest-os $GUEST_OS**. For example:

```
avocado vt-bootstrap --vt-type qemu --vt-guest-os Fedora.27.aarch64 --yes-to-all
```

To show the available guests:

```
avocado list --vt-list-guests
```

The configuration files about the guest os can be found in ~/.local/lib/python2.7/site-packages/usr/share/avocado-plugins-vt/shared/cfg/guest-os.

Step 3: Run Virt Tests

3.1: Create Guest Image

- Firstly install Fedora.27.aarch64.qcow2 via iso image
  Download Fedora-Server-dvd-aarch64-27-1.6.iso to local avocado-vt data directory (/var/lib/avocado/data/avocado-vt/isos/linux or ~/avocado/data/avocado-vt/iso/linux).

Then run the following command:

```
avocado run --vt-type qemu --vt-guest-os Fedora.27.aarch64 \
            --vt-only-filter 'arm64-pci' --vt-extra-params 'netdst=br0' \
            unattended_install.cdrom.extra_cdrom_ks.default_install.aio_native
```

If everything is ok, you would see the output like the following format:

```
JOB ID: <id>
JOB LOG: /home/<user>/avocado/job-results/job-YYYY-MM-DDTHH.mm-xxxxx/job.log
(1/1) io-github-autotest-qemu.unattended_install.cdrom.extra_cdrom_ks.default_install.aio_native: PASS (578.05
s)
```

More log files could be found in /home/<user>/avocado/job-results/job-YYYY-MM-DDTHH.mm-xxxxx/test-results, such as guest's serial logs.

3.2: Write Avocado-VT Tests

See http://avocado-vt.readthedocs.io/en/latest/WritingTests/WritingSimpleTests.html.

3.3: Run Tests with Cartesian Configuration

We can run various tests by using Cartesian Configuration, such as test-qemu-ubuntu-compute.cfg.

Then simply run one command:

```
avocado run --vt-config ./test-qemu-ubuntu-compute.cfg
```

Step 4: One Command to Run All Test Suilts

Firstly clone the auto_setup_ci git repo.

```
git clone https://github.com/zhengxiang93/auto_setup_ci.git
```

1. Run ./check_dependencies.sh to check the required tools and dependencies.

2. Run ./install_avocado.sh to install Avocado framework and Avocado-VT plugin.

3. Run ./run_tests.sh to start Avocado virt tests which is simply defined by test-qemu-ubuntu-compute.cfg.

Core/Virtualization/HowTo/TestsOnAvocado (last modified 2018-06-13 14:03:28)