

Lab: Cuttlefish Getting Started

1. Make sure virtualization with KVM is available.

```
grep -c -w "vmx|svm" /proc/cpuinfo
```

This should return a non-zero value. If running on a cloud machine, this may take cloud-vendor-specific steps to enable.

2. Download, build, and install the host debian package:

```
sudo apt install -y git devscripts config-package-dev debhelper-compat golang
git clone https://github.com/google/android-cuttlefish
cd android-cuttlefish
debuild -i -us -uc -b -d
sudo dpkg -i ../cuttlefish-common_*_64.deb || sudo apt-get install -f
sudo usermod -aG kvm,cvdnetwork,render $USER
sudo reboot
```

The reboot will trigger installing additional kernel modules and applying udev rules.

3. Go to <http://ci.android.com/>
4. Enter a branch name. Start with `aosp-master` if you don't know what you're looking for
5. Navigate to `aosp_cf_x86_64_phone` and click on `userdebug` for the latest build

For ARM, use branch `aosp-master-throttled-copped` and device target `aosp_cf_arm64_only_phone-userdebug`

6. Click on `Artifacts`
7. Scroll down to the OTA images. These packages look like `aosp_cf_x86_64_phone-img-xxxxxx.zip` -- it will always have `img` in the name. Download this file
8. Scroll down to `cvd-host_package.tar.gz` . You should always download a host package from the same build as your images.
9. On your local system, combine the packages:

```
mkdir cf
cd cf
tar xvf /path/to/cvd-host_package.tar.gz
unzip /path/to/aosp_cf_x86_64_phone-img-xxxxxx.zip
```

10. Launch cuttlefish with:

```
$ HOME=$PWD ./bin/launch_cvd
```

You can use `adb` to debug it, just like a physical device:

```
$ ./bin/adb -e shell
```

Cuttlefish: WebRTC Streaming

WebRTC streaming allows users to remotely control their Cuttlefish virtual devices from their browsers, without having to install any other software in the client machine. Other advantages of WebRTC streaming are:

- More efficient encoding than VNC
- In-browser ADB
- Extensible protocol (camera stream, microphone, sensor data are all possible over WebRTC)

Using WebRTC

To use WebRTC and interact with your Cuttlefish device in your browser, do the following:

To launch your devices using WebRTC, add the `--start_webrtc=true` flag to your `launch_cvd` invocation.

```
launch_cvd --start_webrtc=true
```

To see a list of all the devices you can connect to, point your browser to <https://localhost:8443>.

Port usage

Besides TCP:8443, WebRTC uses other ports to establish a connection and run. These ports must be allowed on the firewall if connecting from a different machine than where Cuttlefish is being executed. The list of ports needed are:

- TCP:15550
- UDP:15550
- UDP:15551
- UDP:15552
- UDP:15553
- UDP:15554
- UDP:15555
- UDP:15556
- UDP:15557
- UDP:15558

You will need to stop the virtual device within the same directory as you used to launch the device.

```
$ HOME=$PWD ./bin/stop_cvd`
```