1. Strengthen your Yocto deployments with Autobuilder2 CI tool

Marco Cavallini, KOAN
https://koansoftware.com
Who am I

- Founded **KOAN** on 1996
- Working with software for industrial automation until 1999
- Linux embedded developer since 2000
- Openembedded board member since 2009
- Yocto Project participant since 2010
- Yocto Project Advocate since 2012

- Software development and consulting
- BSP creation
- Device driver kernel development
- Open embedded and Yocto Project training
Agenda

- What is a Continuous Integration (CI) system
- Autobuilder2 History
- Buildbot, the foundations
- Buildbot mechanics
- Buildbot installation
- Autobuilder2 installation
- Autobuilder2 configuration
- Autobuilder2 usage (as-is)
- Autobuilder2 customization
- Autobuilder2 usage for CI on a single machine
What is Autobuilder

• Autobuilder is a project based on Buildbot

• Buildbot is a Python open-source application used to build, test, and release a wide variety of software.

• Autobuilder and Buildbot are licensed under GPLv2
CI overview

• Typical CI on a single machine
CI overview

• Typical CI on a distributed system
Our goal

- Speed Yocto builds by populating premirrors with Autobuilder2
Autobuilder overview

yocto-autobuilder2

yocto-autobuilder-helper

Buildbot
The Continuous Integration Framework
Autobuilder history *

• **Creation of Autobuilder**
  • The autobuilder started life as something OpenedHand used for testing Poky linux.

• **Yocto-autobuilder**
  • It became "yocto-autobuilder" under Beth's stewardship in December 2012 when it was totally re-implemented.

• **Autobuilder2**
  • In February 2018 it was rewritten again, in particular to move from the long obsolete "buildbot eight" codebase to the "buildbot nine" one but also to fix many long running issues and get back to using an upstream codebase.

* Thanks to Richard Purdie who provided these information
People behind Autobuilder *

• **Project Autobuilder**
  • Richard Purdie, Elizabeth Flanagan, Joshua Lock as well as contributions from Tracy Graydon, Anibal Limón and Bill Randle.

• **Project Autobuilder2**
  • Richard Purdie and Joshua Lock.
  • Michael Halstead is the project sysadmin who maintains the infrastructure it all runs on top of.

*Thanks to Richard Purdie who provided these information*
Buildbot, a CI framework for Python
Buildbot basics

- **Buildbot is a job scheduling system**
  - it queues jobs, executes the jobs when the required resources are available, and reports the results
Buildbot basics

• Workers are typically run on separate machines
Buildbot basics
What happens inside the master

![Diagram showing the Yocto Project workflow](image)

- **VCS repos**: Polls for changes in repositories.
- **Buildfactory**: Starts the build process.
- **Worker nodes**: Worker 1, Worker 2, ..., Worker n.
- **Builder nodes**: Builder 1, Builder 2, ..., Builder n.
- **Scheduler**: Triggers builds based on changes or scheduled events.
- **Direct commands**: Can be used for specific builds.
- **Status/results**: Sends status and results back to the scheduler.

*Artwork by Mauro Salvini*
Buildbot installation

• On a native system
  • Probably the fastest solution

• In a Python sandbox
  • Isolates it from the host system
  • Using *pip*

• In a Docker container
  • Isolates it from the host system

https://docs.buildbot.net/2.4.0/full.html
Buildbot installation (in a Python sandbox) [1/3]

- **Create a sandbox**
  
  ```
  mkdir abot-sandbox
  cd abot-sandbox

  python3 -m venv sandbox
  source sandbox/bin/activate
  ```

- **Install master**
  
  ```
  pip install --upgrade pip
  pip install 'buildbot[bundle]'
  ```

- **Install worker**
  
  ```
  pip install --upgrade pip
  pip install buildbot-worker
  ```
Buildbot installation (in a Python sandbox) [2/3]

• Create the master

```
buildbot create-master master
mv master/master.cfg.sample master/master.cfg
```

• Create the worker

```
pip install setuptools-trial
buildbot-worker create-worker worker localhost example-worker pass
```
Buildbot installation (in a Python sandbox) [3/3]

- Content of the sandbox

(sandbox) koan@amonra:~/abot-sandbox$ tree -L 2
.
├── master
│   ├── buildbot.tac
│   │   └── master.cfg
│   │       → c['workers'] = [worker.Worker("example-worker", "pass")]
│   ├── state.sqlite
│   └── sandbox
│       ├── bin
│       │   └── include
│       │       └── lib
│       │           └── lib64 -> lib
│       │       └── pip-selfcheck.json
│       │   └── pyvenv.cfg
│       └── share
│           └── worker
│                   ├── buildbot.tac
│                   │   → workername = 'example-worker'
│                   └── info

Buildbot execution

• Execution of master
  ```
  buildbot start master
  ```

• Control of the build system (using your browser)
  ```
  http://localhost:8010/
  ```
Buildbot

• To be continued in a dedicated session...
• Now let’s have a look at Autobuilder
Autobuilder2
# Buildbot vs. Autobuilder2 lexicon

<table>
<thead>
<tr>
<th>Buildbot</th>
<th>Autobuilder2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>Controller</td>
</tr>
<tr>
<td>Worker</td>
<td>Worker</td>
</tr>
</tbody>
</table>
Autobuilder2 installation

- After you created the sandbox
- Create the master and worker directories

```
bUILdBOT create-master -r yocto-controller

bUILdBOT-worker create-worker -r /
   --umask=0o22 yocto-worker localhost example-worker pass
```

- `yocto-controller` is the directory for master
- `--umask` sets the proper permissions
- `yocto-worker` is the directory for worker
- `localhost` is the network address of the master
- `example-worker` is the name of the worker
- `pass` is the password (master.cfg)
Autobuilder2 installation

• Clone yocto-autobuilder2

```bash
cd yocto-controller
git clone https://git.yoctoproject.org/git/yocto-autobuilder2 yoctoabb
ln -rs yoctoabb/master.cfg master.cfg
```

• `yoctoabb` is the mandatory Autobuilder2 directory name

• Clone yocto-autobuilder-helper

```bash
cd..
git clone https://git.yoctoproject.org/git/yocto-autobuilder-helper
```
Autobuilder2 tree

- Content of the Autobuilder2 sandbox

```bash
(sandbox) koan@amonra:~/ab2-sandbox$ tree -L 3
.
  ├── autobuilder
  │    ├── git
  │    │    └── sandbox
  │    │         └── yocto-autobuilder-helper
  │    │         └── yocto-controller
  │    │             └── yoctoabb
  │    │                     └── master.cfg
  │    └── yocto-worker
  │        └── buildbot.tac
```

* Simplified tree list
Autobuilder2 installation

• Complete installation instructions

http://git.yoctoproject.org/cgit.cgi/yocto-autobuilder2/tree/README-Guide.md

Autobuilder2

Autobuilder default configuration
Autobuilder2 configuration

- Ab2 default configuration
  - The default configuration of Ab2 uses a lot of workers to generate images for several MACHINES

```python
# List of workers in the cluster
workers_ubuntu = ['ubuntu1904-ty-1', 'ubuntu1804-ty-1', 'ubuntu1804-ty-2', 'ubuntu1804-ty-3', 'ubuntu1604-ty-1']
workers_centos = ['centos7-ty-1', 'centos7-ty-2', 'centos7-ty-3', 'centos7-ty-4']
workers_fedora = ['fedora29-ty-1', 'fedora30-ty-1', 'fedora30-ty-2']
workers_debian = ['debian8-ty-1', 'debian9-ty-2', 'debian10-ty-1', 'debian10-ty-2', 'debian10-ty-3']
workers.opensuse = ['tumbleweed-ty-1', 'tumbleweed-ty-2', 'tumbleweed-ty-3', 'opensuse151-ty-1', 'opensuse150-ty-1']
```
Autobuilder2 official website

https://autobuilder.yoctoproject.org/typhoon/
Autobuilder2 website navigation
### Autobuilder2 website navigation

<table>
<thead>
<tr>
<th>Builder Name</th>
<th>Builds</th>
<th>Tags</th>
<th>Workers</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-full</td>
<td>431</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>a-quick</td>
<td>744</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>beaglebone</td>
<td>1180</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>beaglebone-alt</td>
<td>52</td>
<td>49</td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>bringup</td>
<td></td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>build-appliance</td>
<td>1379</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>buildperf-centos7</td>
<td>1415</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>buildperf-ubuntu1804</td>
<td>1499</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>buildtools</td>
<td>1390</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
<tr>
<td>check-layer</td>
<td>1140</td>
<td></td>
<td>1 2 3 4 5 6 9 10 11 13 15 16 17 22 23 24 25 26 27 29 30 31</td>
</tr>
</tbody>
</table>
## Autobuilder2 - builds

<table>
<thead>
<tr>
<th>Device</th>
<th>Build Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>beaglebone</td>
<td>1160, 1159, 1158</td>
</tr>
<tr>
<td>genericx86</td>
<td>1148, 1147, 1146, 1145</td>
</tr>
<tr>
<td>genericx86-64</td>
<td>1154, 1153, 1152, 1151</td>
</tr>
</tbody>
</table>
# Autobuilder2 - workers

<table>
<thead>
<tr>
<th>Device</th>
<th>1160</th>
<th>1159</th>
<th>1158</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>29</th>
<th>30</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>beaglebone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beaglebone-alt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bringup</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **beaglebone**: 1160, 1159, 1158
- **beaglebone-alt**: 52, 51, 50, 49
- **bringup**: 1, 2, 3, 4, 5, 6
Autobuilder2 – worker details

<table>
<thead>
<tr>
<th>State</th>
<th>Masters</th>
<th>WorkerName</th>
<th>Recent Builds</th>
<th>Admin</th>
<th>Host</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ubuntu1804-ty-1</td>
<td></td>
<td>qemux64-64-al</td>
<td>Michael Halstead</td>
<td>Ubuntu</td>
<td>2019.07.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>pkgman-rpm-non-rpm/1144</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>no-x11/1149</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>meta-mingw/1147</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>non-gpl3/1144</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>meta-mingw/1145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>musl-qemu64/1147</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Builds:

<table>
<thead>
<tr>
<th>Builder</th>
<th>#</th>
<th>Started At</th>
<th>Duration</th>
<th>Owners</th>
<th>Worker</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>qemux64-64-al</td>
<td>52</td>
<td>5 hours ago</td>
<td>2 hours</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
<tr>
<td>pkgman-rpm-non-rpm</td>
<td>1144</td>
<td>5 hours ago</td>
<td>an hour</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
<tr>
<td>no-x11</td>
<td>1148</td>
<td>13 hours ago</td>
<td>9 minutes</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
<tr>
<td>meta-mingw</td>
<td>1147</td>
<td>13 hours ago</td>
<td>12 minutes</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
<tr>
<td>non-gpl3</td>
<td>1144</td>
<td>a day ago</td>
<td>20 minutes</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
<tr>
<td>meta-mingw</td>
<td>1145</td>
<td>2 days ago</td>
<td>13 minutes</td>
<td></td>
<td></td>
<td>build successful</td>
</tr>
</tbody>
</table>
# Autobuilder2 – build details

## Build summary

![Highlighted step](image)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Time elapsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>worker_preparation</td>
<td>5 s worker ready</td>
</tr>
<tr>
<td>1 C</td>
<td>Clobber build dir</td>
<td>2 s '/home/pokybuild/yocto-autobuilder-helper/janitor/clobberdir /home/pokybuild/yocto-worker/beaglebone/'</td>
</tr>
<tr>
<td>2 C</td>
<td>Fetch yocto-autobuilder-helper</td>
<td>3 s update</td>
</tr>
<tr>
<td>3</td>
<td>SetProperties</td>
<td>1 s Properties Set</td>
</tr>
<tr>
<td>4 C</td>
<td>Write main layerinfo.json</td>
<td>1 s 'printf '{ ...'</td>
</tr>
<tr>
<td>5</td>
<td>Unpack shared repositories</td>
<td>5 s '/home/pokybuild/yocto-worker/beaglebone/yocto-autobuilder-helper/scripts/shared-repo-unpack /home/pokybuild/yocto-worker/beaglebone/layerinfo.json ...'</td>
</tr>
<tr>
<td>6 C</td>
<td>Set build revision</td>
<td>1 s property 'yp_build_revision' set</td>
</tr>
<tr>
<td>7 C</td>
<td>Set build branch</td>
<td>1 s property 'yp_build_branch' set</td>
</tr>
<tr>
<td>8 C</td>
<td>run-config</td>
<td>2:56:56 '/home/pokybuild/yocto-worker/beaglebone/yocto-autobuilder-helper/scripts/run-config beaglebone ...'</td>
</tr>
</tbody>
</table>
Autobuilder2 – build details

<table>
<thead>
<tr>
<th>Run</th>
<th>Description</th>
<th>Lines</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>run-config</td>
<td>145</td>
<td>view all 145 lines</td>
</tr>
<tr>
<td></td>
<td>stdio</td>
<td></td>
<td>view all 145 lines</td>
</tr>
<tr>
<td></td>
<td>step1b</td>
<td>24868</td>
<td>view all 24868 lines</td>
</tr>
<tr>
<td></td>
<td>step1c</td>
<td>74</td>
<td>view all 74 lines</td>
</tr>
<tr>
<td></td>
<td>step2b</td>
<td>1764</td>
<td>view all 1764 lines</td>
</tr>
<tr>
<td></td>
<td>warnings</td>
<td>0</td>
<td>view all 0 lines</td>
</tr>
<tr>
<td></td>
<td>errors</td>
<td>0</td>
<td>view all 0 lines</td>
</tr>
</tbody>
</table>
## Autobuilder2 – build details

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running task 9589 of 9611</td>
<td>Started</td>
<td>Execute task 9589 of 9611</td>
</tr>
<tr>
<td>Running task 9681 of 9611</td>
<td>Succeeded</td>
<td>Execute task 9681 of 9611</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Started</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
<tr>
<td>Core image sato-sdk-pptest-1.0-r0</td>
<td>Succeeded</td>
<td>Build core image sato-sdk-pptest-1.0-r0</td>
</tr>
</tbody>
</table>

Tasks Summary: Attempted 9611 tasks of which 1001 didn’t need to be rerun and all succeeded.
Autobuilder2 – build details

About this buildbot running for Yocto Autobuilder: typhoon

- Python version: 3.5.3
- Buildbot version: 2.3.1
- Twisted version: 19.2.1

Configuration

buildbot-www is configured using
  auth
  user
  buildbotURL https://autobuilder.yoctoproject.org/typhoon/
  avatar_methods
  port 8010
  multiMaster false
  authz
  plugins
  logfileName http.log
  titleURL https://autobuilder.yoctoproject.org/typhoon/
  title Yocto Autobuilder: typhoon
  versions (["Python","3.5.3"],["Buildbot","2.3.1"],["Twisted","19.2.1"])

Yocto Project | The Linux Foundation
Autobuilder custom ‘lighter’ configuration
Autobuilder2 reduced configuration

• Reduce complexity
  • The goal is to setup a configuration for an image for a single MACHINE only
  • This will help you strengthen the deployments thanks to recurring builds, typically nightly
Autobuilder2 reduced configuration

- Files to be modified

```
  .
  ├── yocto-autobuilder-helper
  │   └── config.json
  ├── yocto-controller
  │   └── yoctoabb
  │       ├── builders.py
  │       ├── config.py
  │       └── master.cfg
  │           └── schedulers.py
  └── yocto-worker
      └── buildbot.tac
```
Autobuilder2 configuration

- In `yocto-autobuilder-helper`
  - Edit `yocto-autobuilder-helper/config.json`

```json
"BASE_HOMEDIR" : "/home/koan/ab2-sandbox",
"BASE_SHAREDDIR" : "${BASE_HOMEDIR}/autobuilder",
"BASE_PUBLISHDIR" : "${BASE_HOMEDIR}/downloads",
```

- In `yocto-controller`
  - Edit `yocto-controller/yoctoabb/master.cfg`

```python
    c['title'] = "KOAN lite Yocto Autobuilder"
    c['titleURL'] = "http://localhost:8010/"
    c['buildbotURL'] = "http://localhost:8010/"
```
Autobuilder2 configuration

- In yocto-controller (again)
  - Edit `yocto-controller/yoctoabb/config.py`

```python
workers_koan = ["example-worker"]
workers = workers_koan
all_workers = workers,
```

```python
sharedrepodir = "/home/koan/ab2-sandbox/repos"
publish_dest = "/home/koan/ab2-sandbox/pub"
```

- Specify the helper directory

```python
repos = {
    "yocto-autobuilder-helper":
    ["file:///home/koan/ab2-sandbox/yocto-autobuilder-helper",
     "master"],
```
Autobuilder2 ‘lite’ customized

http://localhost:8010/
Autobuilder2 ‘lite’ customized
Autobuilder2 to speed up Yocto build

• Share the artefacts with Yocto
  • Edit `local.conf`
  • Share downloads

```bash
PREMIRRORS_prepend = "\n  git://.*/.* file:///home/koan/ab2-sandbox/autobuilder/current_sources/ \n  ftp://.*/.* file:///home/koan/ab2-sandbox/autobuilder/current_sources/ \n  http://.*/.* file:///home/koan/ab2-sandbox/autobuilder/current_sources/ \n  https://.*/.* file:///home/koan/ab2-sandbox/autobuilder/current_sources/ \n"
```

• Share `SSTATE`

```bash
SSTATE_MIRRORS = "file:///.* \n  file:///home/koan/ab2-sandbox/autobuilder/pub/sstate(PATH"
```
Autobuilder2 to speed up Yocto build
Questions?

https://yoctoproject.org

https://koansoftware.com