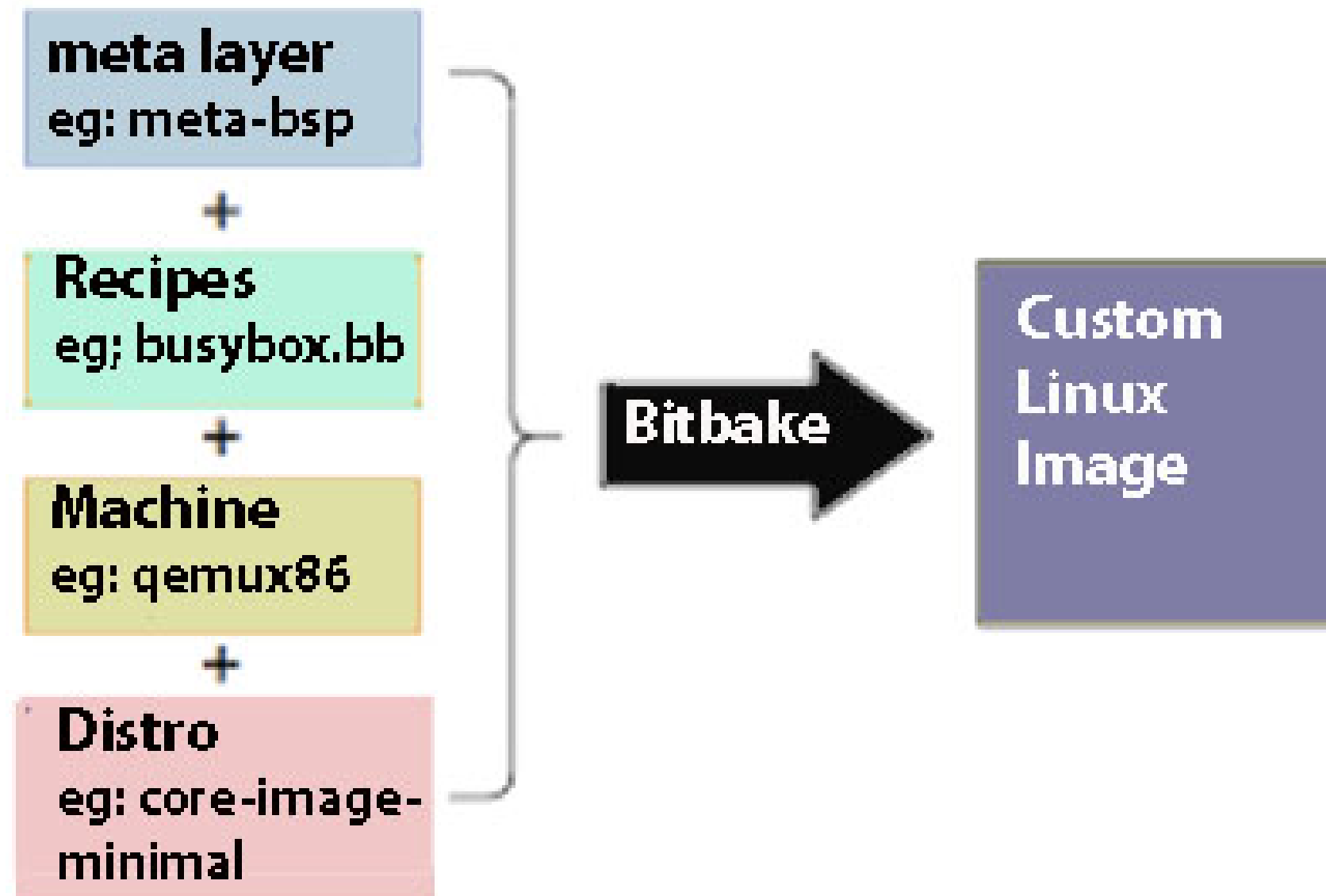


PRESENTED BY:
Mohamed Saied

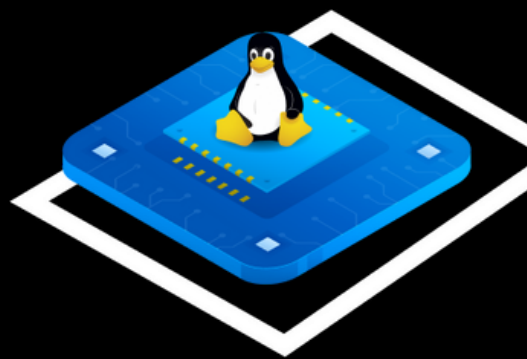
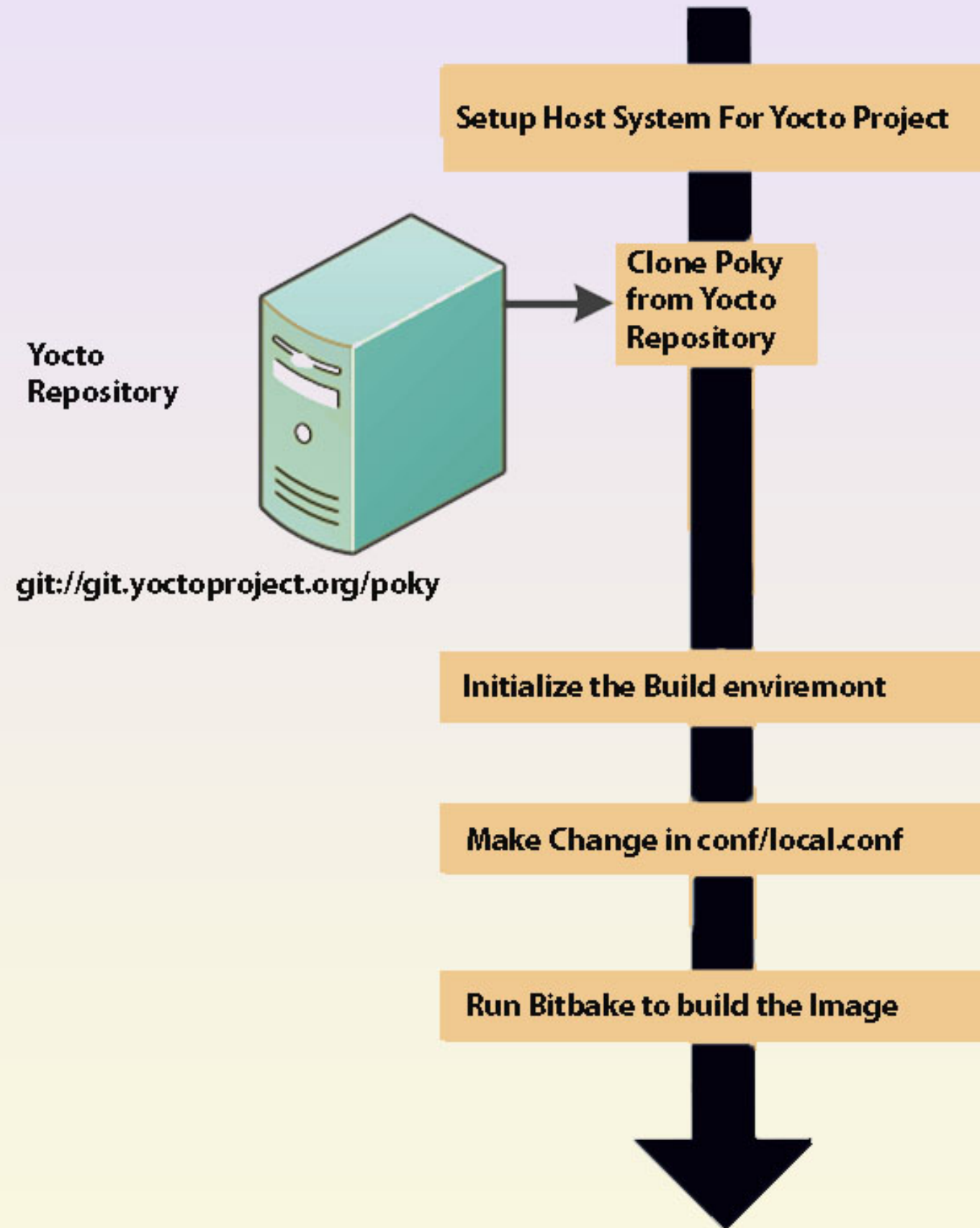
Bitbake demystified



Yocto Custom Image



Yocto Image build procedure



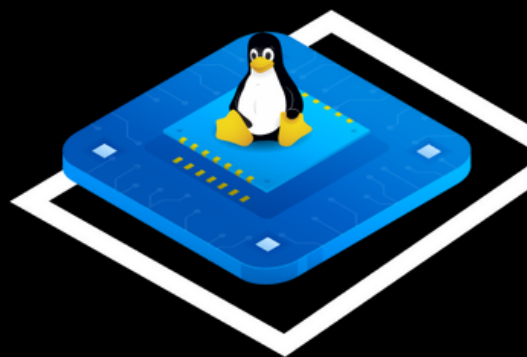
BitBake Basic Commands:

bitbake <target>:

Initiates the build process for the specified target, which can be an image, package, or recipe.

bitbake -c <task> <target>:

Executes a specific task for the given target. For example, bitbake -c clean <target> cleans the build artifacts.



Layer and Configuration Management:

bitbake-layers show-layers:

Displays a list of layers currently added to the build configuration.

bitbake-layers add-layer <path_to_layer>:

Adds a new layer to the build configuration.

bitbake-layers remove-layer <layer_name>:

Removes a layer from the build configuration.

bitbake-layers show-append:

Displays the list of append files used in the build.



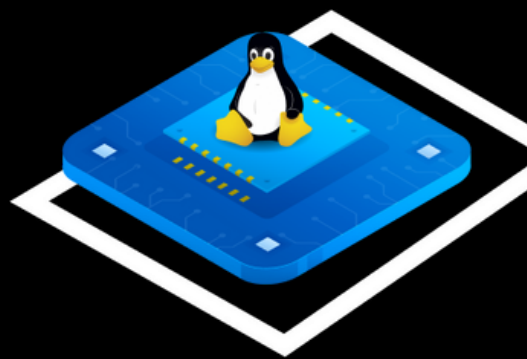
Cleaning:

bitbake -c clean <target>:

Cleans the build artifacts for the specified target.

bitbake -c cleansstate <target>:

Cleans the shared state cache for the specified target.



Metadata and Configuration:

bitbake -e <target> | less:

Displays the metadata/environment variables for the specified target.

bitbake -s:

Shows available recipes and their versions.

Package Management:

bitbake -g <target>:

Generates a task dependency graph for the specified target.

bitbake -k <target>:

Continues with the build despite encountering errors for the specified target.



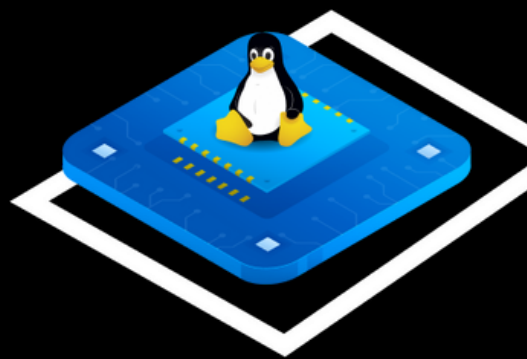
Miscellaneous:

bitbake -h | less:

Displays the help page for BitBake commands.

bitbake -D <target>:

Enables debug logging for the specified target.



Steps to create a new meta layer

```
$ cd poky
$ source oe-init-build-env
# Now, build folder would your current working directory.

$ bitbake-layers create-layer ../meta-tutorial
# This command creates the meta-tutorial layer inside the poky directory.

# Add this layer into bblayer.conf file
$ bitbake-layers add-layer ../meta-tutorial

# Display all layer present in the bblayer.conf file.
$ bitbake-layers show-layers
```

```
tutorial@adda:~/yocto/poky/build$
tutorial@adda:~/yocto/poky/build$ bitbake-layers create-layer ../meta-tutorial
NOTE: Starting bitbake server...
Add your new layer with 'bitbake-layers add-layer ../meta-tutorial'
tutorial@adda:~/yocto/poky/build$ bitbake-layers add-layer ../meta-tutorial
NOTE: Starting bitbake server...
tutorial@adda:~/yocto/poky/build$ bitbake-layers show-layers
NOTE: Starting bitbake server...
layer                                path                                priority
=====
meta                                /home/tutorial/yocto/poky/meta      5
meta-poky                           /home/tutorial/yocto/poky/meta-poky 5
meta-yocto-bsp                       /home/tutorial/yocto/poky/meta-yocto-bsp 5
meta-tutorial                       /home/tutorial/yocto/poky/meta-tutorial 6
tutorial@adda:~/yocto/poky/build$
```



Write the simple hello recipe file

```
DESCRIPTION = "Simple helloworld application"
LICENSE = "MIT"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"

SRC_URI = "file://hello.c"

S = "${WORKDIR}"

do_compile() {
    ${CC} hello.c ${LDFLAGS} -o hello
}

do_install() {
    install -d ${D}${bindir}
    install -m 0755 hello ${D}${bindir}
}
```

Reference

<https://docs.yoctoproject.org/dev-manual/new-recipe.html>



Directory tree

```
tutorial@adda:~/yocto/poky/meta-tutorial$ tree
```

```
.
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
├── recipes-example
│   ├── example
│   │   └── example_0.1.bb
│   └── hello
│       ├── files
│       │   └── hello.c
│       └── hello_1.0.bb
```

```
5 directories, 6 files
```

```
tutorial@adda:~/yocto/poky/meta-tutorial$
```



Select machine configuration and Add hello package to rootsfs

We are building an image for machine QEMUARM so we have to add this machine in conf/local.conf file.

```
#By default, this machine selection is enabled.  
MACHINE ??= "qemuarm"  
  
#We need to add the hello software package to the target image  
IMAGE_INSTALL_append = " hello"
```

Run bitbake to build the minimal boot image for QEMU.

```
$bitbake core-minimal-image
```

Run Qemu

```
$runqemu qemuarm
```

