# Build Yocto Image for Raspberry Pi

This page provides all the instructions needed to build yocto image for raspberry pi with kanto recipes

Building a Yocto Image for a Raspberry Pi involves several steps, including setting up the environment, download the necessary layers, configuring the build and compiling the image.

## Before you begin

- Install the required packages required for the build on a Ubuntu/Debian system

```
sudo apt-get update
sudo apt install -y gawk wget git-core diffstat unzip texinfo gcc-multilib
build-essential chrpath socat cpio python3 python3-pip python3-pexpect \
xz-utils debianutils iputils-ping libsdl1.2-dev xterm zstd liblz4-tool \
```

## Clone the Yocto/Poky Repository

- Verify the Yocto Version of Kanto availabe in meta-kanto layer, https://github.com/eclipse-kanto/meta-kanto, this example provides information for kirkstone branch.

- Create a Source folder

```
mkdir sources
cd sources
```

- If it is kirkstone branch then, clone poky for kirkstone version in source directory

```
git clone https://github.com/yoctoproject/poky.git
cd poky
git checkout kirkstone
```

Note : Change branch based on meta-kanto layer.

## Add Meta Layers to the source directory

- Based on the yocto version, clone the meta layers in the source directory `meta-raspberrypi` `meta-openembedded` `meta-virtualization` `meta-lts-mixins`
- Clone meta-raspberry pi layer to sources directory

```
cd ..
git clone git://git.yoctoproject.org/meta-raspberrypi
cd meta-raspberrypi
git checkout kirkstone
```

- Clone meta-openembedded layer to sources directory

```
cd ..
git clone https://github.com/openembedded/meta-openembedded.git
cd meta-openembedded
git checkout kirkstone
```

- Clone meta-virtualization layer to sources directory

```
cd ..
git clone https://git.yoctoproject.org/git/meta-virtualization
cd meta-virtualization
git checkout kirkstone
```

- Clone meta-lts-mixins layers to source directory

```
cd ..
git clone https://git.yoctoproject.org/git/meta-lts-mixins
cd meta-lts-mixins
git checkout kirkstone/go
```

Note : The above layer is required to get the updated go version to be added in the yocto build, since kanto requires go version 1.19 and above.

## Add meta-kanto layer to the sources directory

- Clone meta-kanto layer to the sources directory

```
cd ..
git clone https://github.com/eclipse-kanto/meta-kanto.git
```

```
cd meta-kanto
git checkout kirkstone
```

Note : Make sure all the layers cloned are of same yocto version ( kirkstone in this case)

## Create Build Directory

- After cloning all the required meta layers, move out of source directory to create build directory

```
cd ../..
source sources/poky/oe-init-build-env
```

- Run the below command to view the layers present in `bblayers.conf` file

```
bitbake-layers show-layers
```

Note : Resolve any dependendencies if occured while running bitbake command.

## Configure bblayer.conf file

- By Default in the bblayer.conf file some of the layers will be added
- Add all the layers to the bblayers.conf file with below command

```
bitbake-layers add-layer /home/path/to/meta/layer/directory
```

The following layers should be added in the bblayer.conf file

```
meta-raspberrypi
meta-openembedded
meta-virtualization
meta-lts-mixins
meta-openembedded/meta-oe
meta-openembedded/meta-python
meta-opemembedded/meta-networking
meta-openembedded/meta-filesystems
meta-kanto
```

- Example to add layers to bblayers.conf file

while adding layers bitbake might have dependencies, add the dependent layers first. Example, To add meta-kanto layer to bblayer.conf file which is kept at `/home/yocto/sources/meta-kanto`

`bitbake-layers add-layers /home/yocto/sources/meta-kanto`

- After adding all the required layers in bblayer.conf file, verify again by running the below command

```
bitbake-layers show-layers
```

## Configure local.conf file

- Open local.conf file which is placed at the below location in build directory

```
vi conf/local.conf
```

- Change the machine variable in local.conf file to raspberry pi machine

```
MACHINE ??= "raspberrypi4"
```

Note: Check the sources/meta-raspberrypi/conf/machine for the availabe machines for raspberry pi.

- Add required variables in local.conf file as shown and provided in the link below,

**https://github.com/eclipse-kanto/meta-kanto**

```
# Add the required DISTRO_FEATURES
DISTRO_FEATURES:append = " virtualization systemd"

# Configure the kernel modules required to be included
MACHINE_ESSENTIAL_EXTRA_RRECOMMENDS += "kernel-modules"

# System initialization manager setup
VIRTUAL-RUNTIME_init_manager = "systemd"
DISTRO_FEATURES_BACKFILL_CONSIDERED = "sysvinit"
VIRTUAL-RUNTIME_initscripts = "systemd-compat-units"

# Add the Eclipse Kanto components
IMAGE_INSTALL:append = " mosquitto"
IMAGE_INSTALL:append = " suite-connector"
IMAGE_INSTALL:append = " aws-connector"
IMAGE_INSTALL:append = " azure-connector"
```

```
IMAGE_INSTALL:append = " software-updates"
IMAGE_INSTALL:append = " file-upload"
IMAGE_INSTALL:append = " file-backup"
IMAGE_INSTALL:append = " update-manager"
IMAGE_INSTALL:append = " container-management"
IMAGE_INSTALL:append = " local-digital-twins"
```

- Run bitbake `target-name' availabe as shown below,

  ```
  Common targets are:
  core-image-minimal
  core-image-full-cmdline
  core-image-sato
  core-image-weston
  meta-toolchain
  meta-ide-support
  ```

  Note : If any build issues comes up, resolve the issue and run the bitbake `target-name' command again for the build.

## Final Build Image Repository Location

- After the successful build, the image will be availabe at the below location.

```
build/tmp/deploy/images/`machine_name`/
```

## Run & Test the image with QEMU (Quick Emulator)

- If RaspberryPi device is not availabe, the image can be run and tested on QEMU. Make the below changes to run on QEMU.

  In `build/conf/local.conf' file

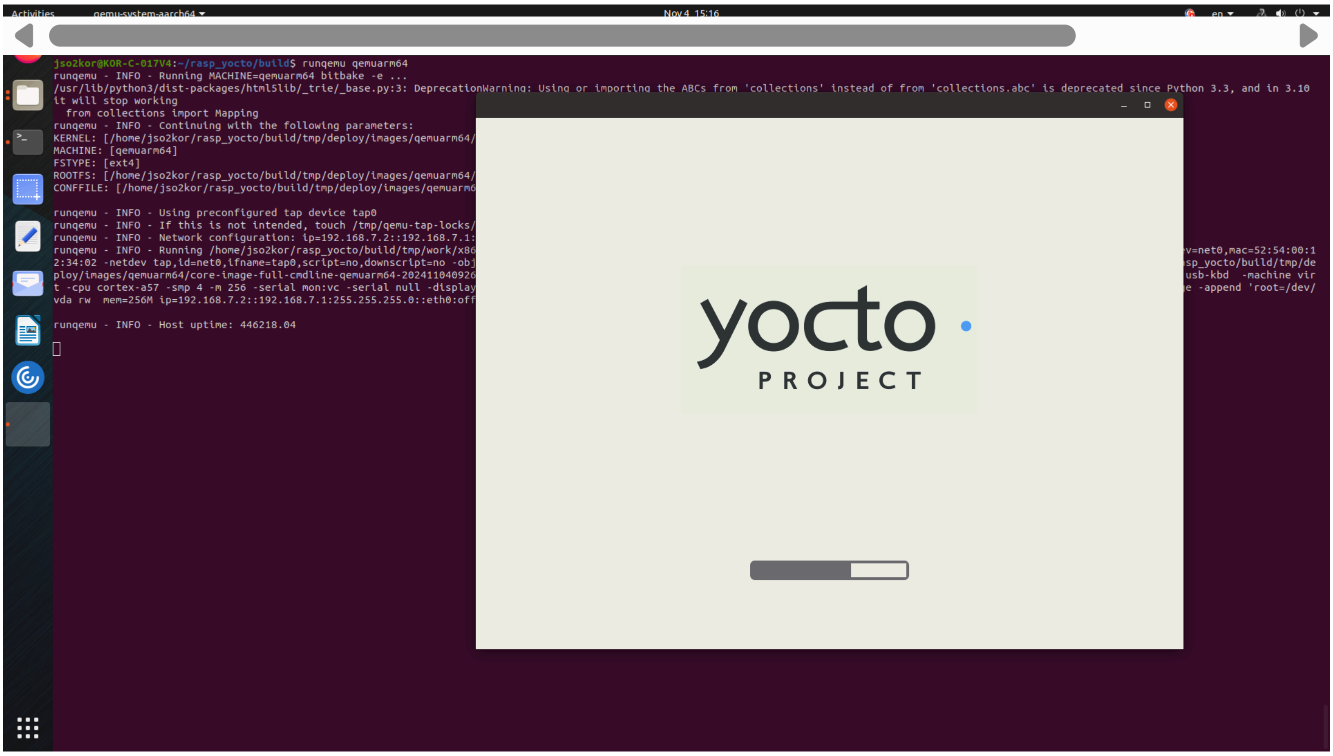
- Change the machine variable name to relevant qemu arch as shown below by changing the `MACHINE` varilabe from raspberrypi4 to 'qemux86_64`

```
# You need to select a specific machine to target the build with. There ar
# of emulated machines available which can boot and run in the QEMU emulat
#
#MACHINE ?= "qemuarm"
#MACHINE ?= "qemuarm64"
#MACHINE ?= "qemux86"
```

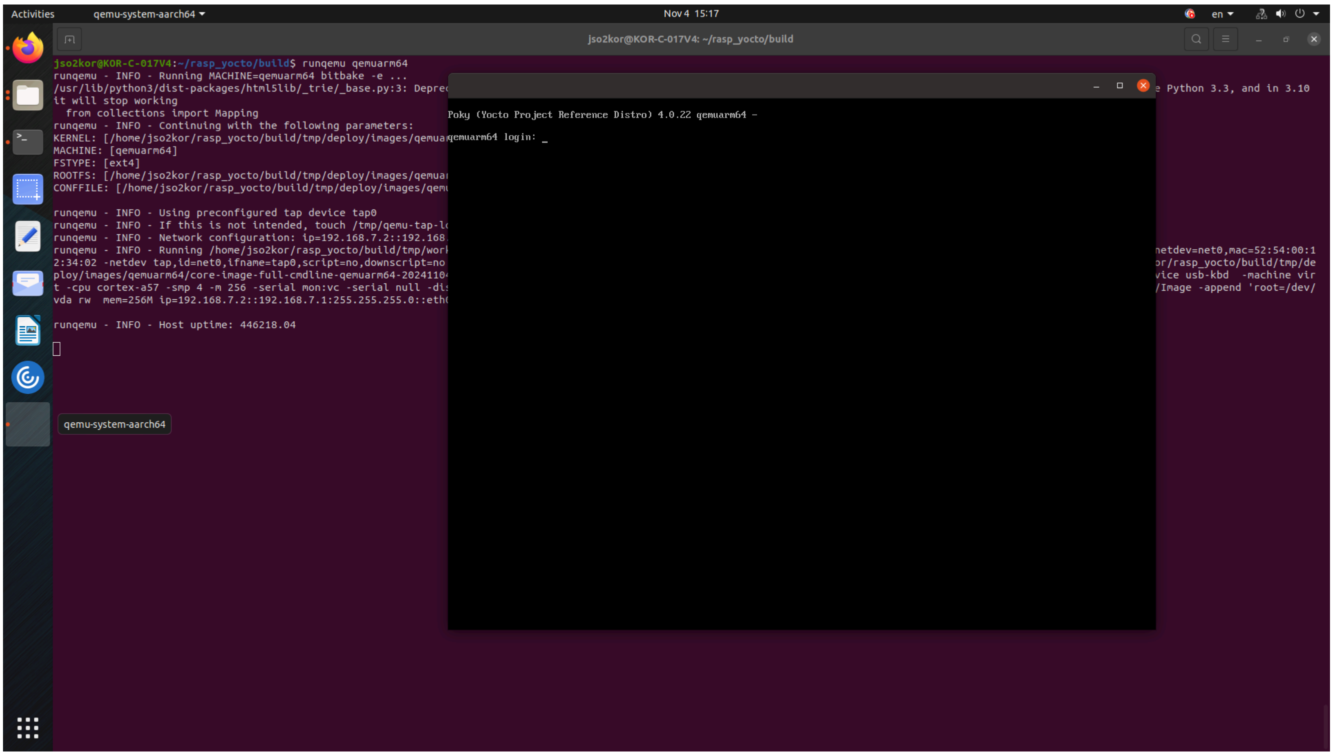
```
#MACHINE ?= "qemux86-64"

# This sets the default machine to be qemux86-64 if no other machine is se
MACHINE ??= "qemuarm64"
```

- Run bitbake `target-name` command by sourcing the `oe-init-build-env` script
- In the same build directory run the below command to run qemu

```
runqemu qemuarm64
```



```
jso2kor@KOR-C-017V4:~/rasp_yocto/build$ runqemu qemuarm64
runqemu - INFO - Running MACHINE=qemuarm64 bitbake -e ...
/usr/lib/python3/dist-packages/html5lib/_trie/_base.py:3: DeprecationWarning: Using or importing the ABCs from 'collections' instead of from 'collections.abc' is deprecated since Python 3.3, and in 3.10
it will stop working
  from collections import Mapping
runqemu - INFO - Continuing with the following parameters:
KERNEL: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuarm64/
MACHINE: [qemuarm64]
FSTYPE: [ext4]
ROOTFS: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuarm64/
CONFFILE: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuarm6

runqemu - INFO - Using preconfigured tap device tap0
runqemu - INFO - If this is not intended, touch /tmp/qemu-tap-locks/
runqemu - INFO - Network configuration: ip=192.168.7.2::192.168.7.1:
runqemu - INFO - Running /home/jso2kor/rasp_yocto/build/tmp/work/x86                v=net0,mac=52:54:00:1
2:34:02 -netdev tap,id=net0,ifname=tap0,script=no,downscript=no -obj                sp_yocto/build/tmp/de
ploy/images/qemuarm64/core-image-full-cmdline-qemuarm64-202411040926                usb-kbd  -machine vir
t -cpu cortex-a57 -smp 4 -m 256 -serial mon:vc -serial null -display                e -append 'root=/dev/
vda rw  mem=256M ip=192.168.7.2::192.168.7.1:255.255.255.0::eth0:off

runqemu - INFO - Host uptime: 446218.04
```



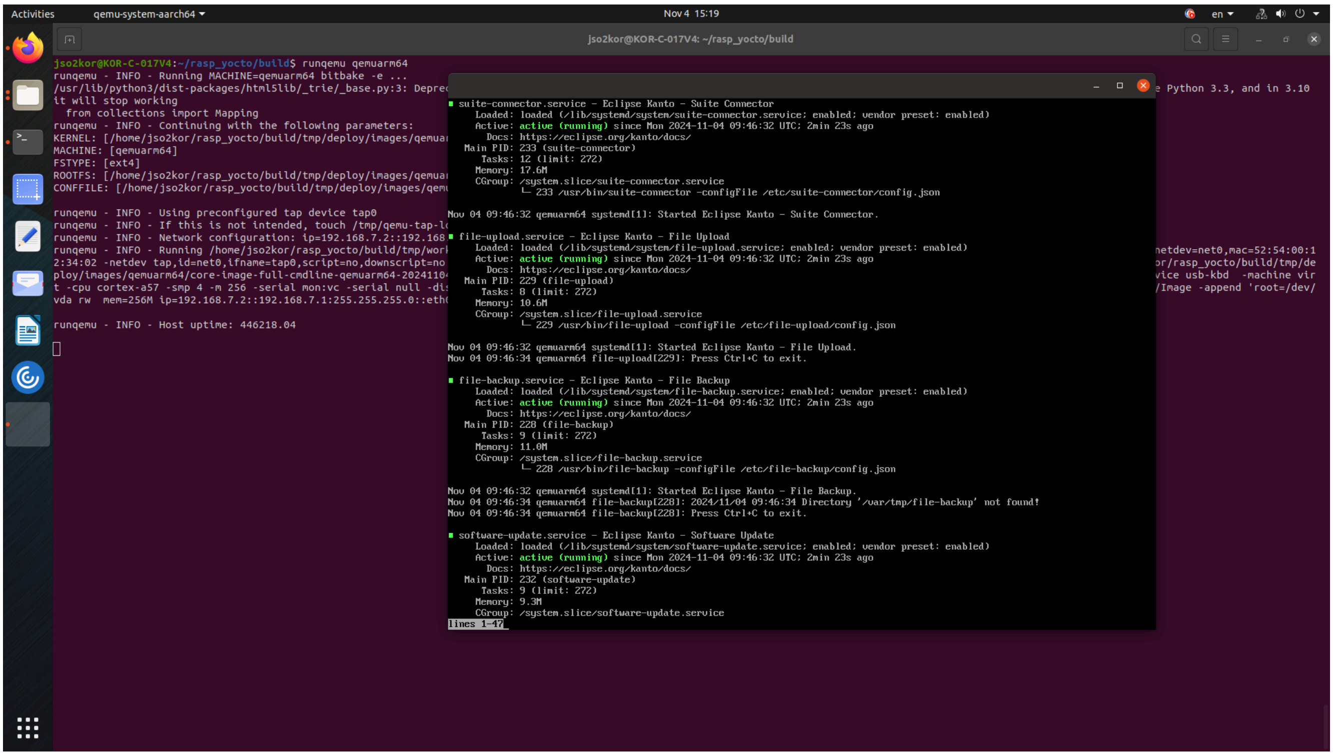
```
jso2kor@KOR-C-017V4:~/rasp_yocto/build$ runqemu qemuarm64
runqemu - INFO - Running MACHINE=qemuarm64 bitbake -e ...
/usr/lib/python3/dist-packages/html5lib/_trie/_base.py:3: Deprec                    e Python 3.3, and in 3.10
it will stop working
  from collections import Mapping
runqemu - INFO - Continuing with the following parameters:
KERNEL: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemua
MACHINE: [qemuarm64]
FSTYPE: [ext4]
ROOTFS: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuai
CONFFILE: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemu

runqemu - INFO - Using preconfigured tap device tap0
runqemu - INFO - If this is not intended, touch /tmp/qemu-tap-lc
runqemu - INFO - Network configuration: ip=192.168.7.2::192.168.
runqemu - INFO - Running /home/jso2kor/rasp_yocto/build/tmp/work                    netdev=net0,mac=52:54:00:1
2:34:02 -netdev tap,id=net0,ifname=tap0,script=no,downscript=no                     or/rasp_yocto/build/tmp/de
ploy/images/qemuarm64/core-image-full-cmdline-qemuarm64-2024110                     vice usb-kbd  -machine vir
t -cpu cortex-a57 -smp 4 -m 256 -serial mon:vc -serial null -di                     /Image -append 'root=/dev/
vda rw  mem=256M ip=192.168.7.2::192.168.7.1:255.255.255.0::eth

runqemu - INFO - Host uptime: 446218.04
```

```
Poky (Yocto Project Reference Distro) 4.0.22 qemuarm64 -
qemuarm64 login: _
```

- The above command will open a window which boots as "YOCTO PROJECT" and it enters to command line window. Enter login as `root', and check for kanto components with the below commands.

  ```
  systemctl status \
  suite-connector.service \
  container-management.service \
  software-update.service \
  file-upload.service \
  file-backup.service \
  system-metrics.service \
  kanto-update-manager.service
  ```

  All listed services must be in an active running state.



```
jso2kor@KOR-C-017V4:~/rasp_yocto/build$ runqemu qemuarm64
runqemu - INFO - Running MACHINE=qemuarm64 bitbake -e ...
/usr/lib/python3/dist-packages/html5lib/_trie/_base.py:3: Deprec                            e Python 3.3, and in 3.10
it will stop working
  from collections import Mapping
runqemu - INFO - Continuing with the following parameters:
KERNEL: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuar
MACHINE: [qemuarm64]
FSTYPE: [ext4]
ROOTFS: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemuar
CONFFILE: [/home/jso2kor/rasp_yocto/build/tmp/deploy/images/qemu

runqemu - INFO - Using preconfigured tap device tap0
runqemu - INFO - If this is not intended, touch /tmp/qemu-tap-lc
runqemu - INFO - Network configuration: ip=192.168.7.2::192.168.
runqemu - INFO - Running /home/jso2kor/rasp_yocto/build/tmp/work                            netdev=net0,mac=52:54:00:1
2:34:02 -netdev tap,id=net0,ifname=tap0,script=no,downscript=no                             or/rasp_yocto/build/tmp/de
ploy/images/qemuarm64/core-image-full-cmdline-qemuarm64-2024110                             vice usb-kbd  -machine vir
t -cpu cortex-a57 -smp 4 -m 256 -serial mon:vc -serial null -di                             /Image -append 'root=/dev/
vda rw  mem=256M ip=192.168.7.2::192.168.7.1:255.255.255.0::eth

runqemu - INFO - Host uptime: 446218.04
```

```
● suite-connector.service - Eclipse Kanto - Suite Connector
     Loaded: loaded (/lib/systemd/system/suite-connector.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-11-04 09:46:32 UTC; 2min 23s ago
       Docs: https://eclipse.org/kanto/docs/
   Main PID: 233 (suite-connector)
      Tasks: 12 (limit: 272)
     Memory: 17.6M
     CGroup: /system.slice/suite-connector.service
             └─ 233 /usr/bin/suite-connector -configFile /etc/suite-connector/config.json

Nov 04 09:46:32 qemuarm64 systemd[1]: Started Eclipse Kanto - Suite Connector.

● file-upload.service - Eclipse Kanto - File Upload
     Loaded: loaded (/lib/systemd/system/file-upload.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-11-04 09:46:32 UTC; 2min 23s ago
       Docs: https://eclipse.org/kanto/docs/
   Main PID: 229 (file-upload)
      Tasks: 8 (limit: 272)
     Memory: 10.6M
     CGroup: /system.slice/file-upload.service
             └─ 229 /usr/bin/file-upload -configFile /etc/file-upload/config.json

Nov 04 09:46:32 qemuarm64 systemd[1]: Started Eclipse Kanto - File Upload.
Nov 04 09:46:34 qemuarm64 file-upload[229]: Press Ctrl+C to exit.

● file-backup.service - Eclipse Kanto - File Backup
     Loaded: loaded (/lib/systemd/system/file-backup.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-11-04 09:46:32 UTC; 2min 23s ago
       Docs: https://eclipse.org/kanto/docs/
   Main PID: 228 (file-backup)
      Tasks: 9 (limit: 272)
     Memory: 11.0M
     CGroup: /system.slice/file-backup.service
             └─ 228 /usr/bin/file-backup -configFile /etc/file-backup/config.json

Nov 04 09:46:32 qemuarm64 systemd[1]: Started Eclipse Kanto - File Backup.
Nov 04 09:46:34 qemuarm64 file-backup[228]: 2024/11/04 09:46:34 Directory '/var/tmp/file-backup' not found!
Nov 04 09:46:34 qemuarm64 file-backup[228]: Press Ctrl+C to exit.

● software-update.service - Eclipse Kanto - Software Update
     Loaded: loaded (/lib/systemd/system/software-update.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-11-04 09:46:32 UTC; 2min 23s ago
       Docs: https://eclipse.org/kanto/docs/
   Main PID: 232 (software-update)
      Tasks: 9 (limit: 272)
     Memory: 9.3M
     CGroup: /system.slice/software-update.service
lines 1-47
```

## Flash the image on Raspberry Pi

- The build image will be availabe at `build/tmp/deploy/images/raspberrypi4/`
- Identify the SD Card Device with the below command to find the device name of your SD card

  ```
  lsblk
  ```

- Flash the image to sd card

```
sudo dd if=/path/to/image.wic of=/dev/sdX bs=4M status=progress
```

- Boot the Raspberry Pi,

  Insert the SD card into your Raspberry Pi and power it on. The Pi should boot from the Yocto image.Login to your device and run the below command to verify the kanto components.

```
systemctl status \
suite-connector.service \
container-management.service \
software-update.service \
file-upload.service \
file-backup.service \
system-metrics.service \
kanto-update-manager.service
```

Last modified November 5, 2024