Quick Start Guide

Network and Edge Reference System Architectures - On Premises Edge AI Box

Develop and verify edge analytics services for On Prem Edge AI Box using BMRA on the Intel® Core™ processor.

Authors

Abhijit Sinha

Zhifang Long

Alex Lam

Introduction

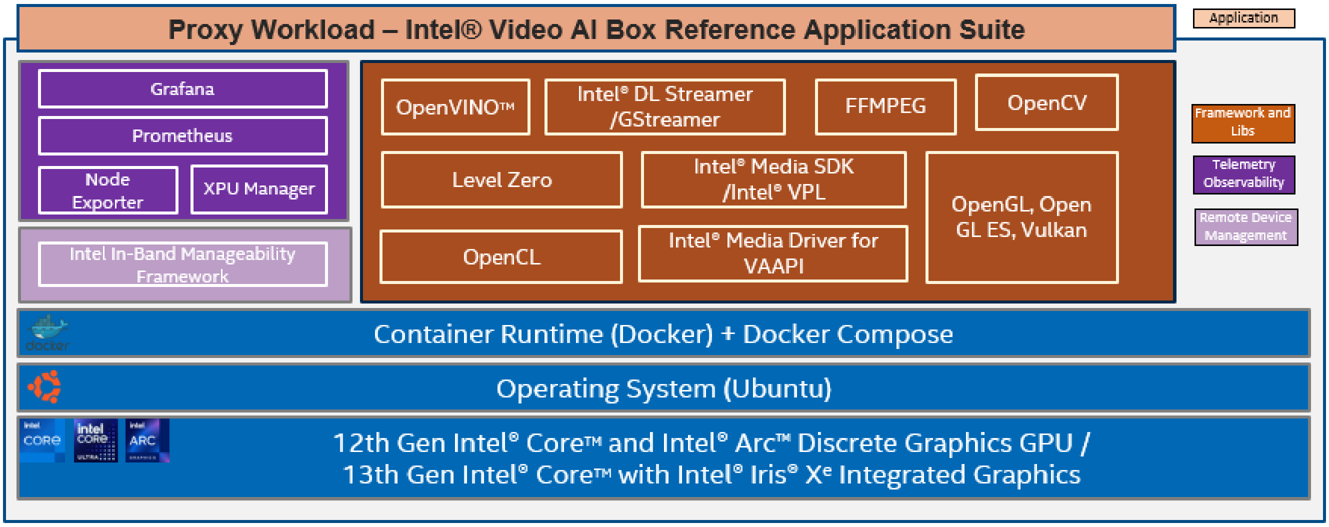
The Reference System Architectures (Reference System[1]) are forward-looking template solutions for fast automated software installation and deployment.

This document is a quick start guide to configure and deploy **Edge AI Box** underlying software requirements using the **Container Bare Metal Reference System Architecture (BMRA)** on **Intel® Core™ processors** with either **Intel® Arc™ Discrete Graphics GPU** or **Intel® Iris® Xᵉ Integrated Graphics** platform.

The Reference System is deployed using the **On Prem Edge AI Box Configuration Profile** with **optimized configuration for edge video analytics workloads in a single box** in real time for lightweight edge devices. Video Analytics is enabled by OpenVINO™ toolkit and a choice of OpenCV or Intel® Deep Learning Streamer (Intel® DL Streamer) as AI-based media analytics frameworks. The platform is accelerated by Intel® Arc™ Discrete Graphics GPU or Intel® Iris® Xᵉ Integrated Graphics, as shown in Figure 1.

On Prem Edge AI Box Architecture

Figure 1 shows the architecture diagram of the On Prem Edge AI Box Profile where media analytics frameworks OpenCV and Intel® DL Streamer are containerized and work alongside a Video Analytics base library container including OpenVINO™ toolkit and media accelerators, and drivers. The provided container suite is used for microservice-based system architectures.



Proxy Workload – Intel® Video AI Box Reference Application Suite
Application
Grafana
Prometheus
Node Exporter
XPU Manager
Intel In-Band Manageability Framework
OpenVINO™
Intel® DL Streamer /GStreamer
FFMPEG
OpenCV
Level Zero
Intel® Media SDK /Intel® VPL
OpenGL, Open GL ES, Vulkan
OpenCL
Intel® Media Driver for VAAPI
Framework and Libs
Telemetry Observability
Remote Device Management
Container Runtime (Docker) + Docker Compose
docker
Operating System (Ubuntu)
12th Gen Intel® Core™ and Intel® Arc™ Discrete Graphics GPU /
13th Gen Intel® Core™ with Intel® Iris® Xᵉ Integrated Graphics
intel CORE
intel CORE
intel ARC

Figure 1: Architecture of Edge AI Box deployment using BMRA `on_prem_aibox` Profile

[1] In this document, "Reference System" refers to the Network and Edge Reference System Architecture.

Hardware BOM

Following is the list of the hardware components that are required for setting up Reference Systems:

Ansible Host	Laptop or server running a UNIX base distribution
Target Node	1x 11th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics; OR 1x 12th Gen Intel® Core™ desktop processor with Intel® Arc™ Discrete Graphics GPU; OR 1x 12th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics; OR 1x 12th Gen Intel® Core™ processor for IOT Edge with Intel® Iris® X^e Integrated Graphics; OR 1x 13th Gen Intel® Core™ mobile processor with Intel® Iris® X^e Integrated Graphics; OR 1x Intel® Core™ Ultra processor with integrated Intel® Arc™ GPU and Intel® AI Boost
Discrete GPU	Intel® Arc™ A380 Graphics
BIOS	Use the default BIOS settings (The user may need to disable secure boot to install the out of tree (OOT) drivers)

Software BOM

Following is the list of the software components that are required for setting up Reference Systems:

High Level Media Frameworks	Intel® DL Streamer, GStreamer, OpenCV, FFmpeg
Inference Frameworks	OpenVINO™ toolkit
Media and Video Acceleration	Intel® Media SDK/Intel® Video Processing Library (Intel® VPL), Intel® Media Driver for VAAPI, Libva
Graphics and Compute Acceleration	Intel GPU driver and OpenGL, OpenCL, Level Zero runtime
AI Acceleration	Intel® AI Boost driver and runtime
Observability	XPU Manager, Node Exporter, Prometheus, Grafana
Remote Device Management	Intel® In-Band Manageability Framework
Container Runtime	Docker, Docker-compose
OS	Ubuntu 22.04.2 Desktop (Ubuntu default kernel5.19, or Linux Kernel Overlay for Intel® Core™ Ultra processor)

For more details on software versions for the **On Prem Edge AI Box Profile**, refer to Chapter 4 of BMRA User Guide listed in the Reference Documentation section.

Getting Started

Prerequisites

Before starting the deployment, perform the following steps:

- A fresh OS installation is expected on the controller and target nodes to avoid a conflict between the RA deployment process with the existing software packages. To deploy RA on the existing OS, ensure that there is no prior Docker or Kubernetes* (K8s) installations on the server(s).
- The target nodes hostname must be in lowercase, numerals, and hyphen ' – '.
  - For example: wrk-8 is acceptable; wrk_8, WRK8, Wrk^8 are not accepted as hostnames.
- The target node must be Network Time Protocol (NTP) synced, i.e., the correct date and time must be set.
- The BIOS on the target node is set as per the recommended settings.

Deployment Setup

Ansible playbooks are used to install the Bare Metal (BMRA), which sets up the infrastructure for an On Prem Edge AI Box. Figure 2 shows the deployment model for Edge AI Box infrastructure using BMRA.

The target device starts with Ubuntu 22.04.2 Desktop only, acting as both Ansible host and target, and it ends with the deployed infrastructure using the `on_prem_aibox` Reference System profile.

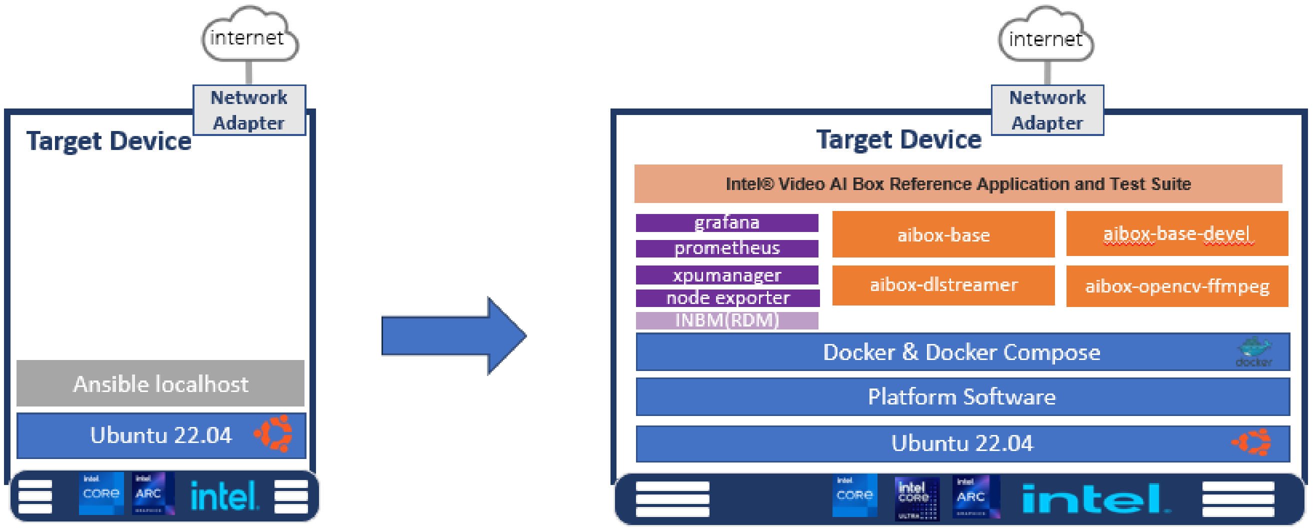


Figure 2: BMRA deployment setup for Edge AI Box

Installation Flow for RA Deployment

Ansible playbooks are used to install the Bare Metal (BMRA), which sets up the infrastructure for an On Prem Edge AI Box.

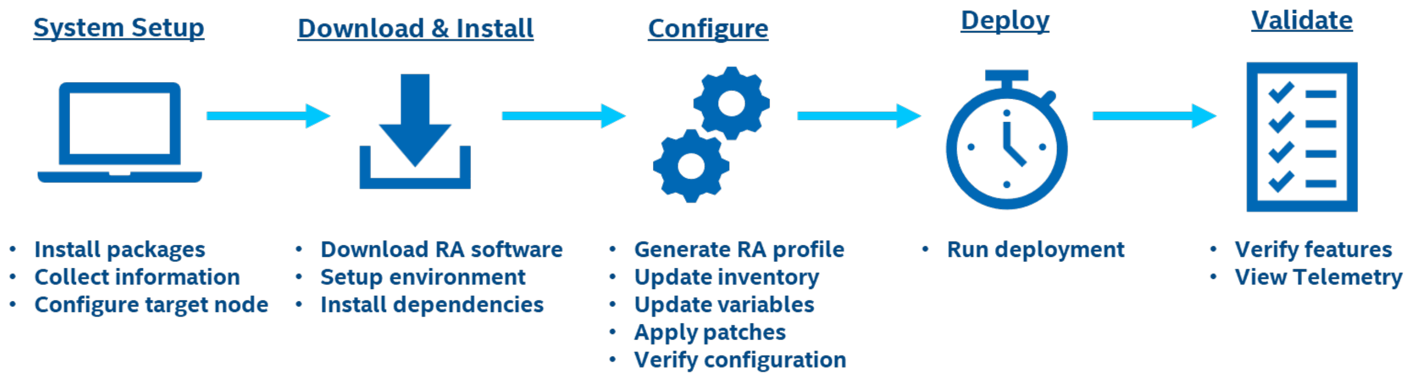


Figure 3: RA Deployment Flow

Step 1 – Set Up the System

The Edge AI Box is deployed on a single target host running Ubuntu OS. The deployment is on a localhost bare-metal environment (known as target host) and there is no need for a separate Ansible host for this deployment.

Target Host

System Setup

Install necessary packages (some might already be installed):

```
# sudo apt update
# sudo apt install -y python3 python3-pip openssh-client git build-essential
# pip3 install -upgrade pip
```

Step 2 – Download and Install

Target Host

Download & Install

1. Download the source code from the GitHub repository for the Reference System server:

```
# git clone https://github.com/intel/container-experience-kits/
# cd container-experience-kits
# git checkout v24.01
```

2. Set up Python* virtual environment and install dependencies:

```
# python3 -m venv venv
# source venv/bin/activate
# pip3 install -r requirements.txt
```

3. Install Ansible dependencies for the Reference System:

```
# ansible-galaxy install -r collections/requirements.yml
```

4. If the target device is an Intel® Core™ Ultra processor, then first download the related NDA packages.

4.1 Download Ubuntu with Kernel Overlay for Intel® Core™ Ultra processor – Software Packages (Software Kit: 781820) and Intel® AI Boost driver, then put kernel, audio firmware, and Intel® AI Boost driver in /tmp/ folder as shown below:

```
tmp
├── driver
│   └── vpu-linux-drivers-ubuntu2204-release-*.tar.gz
├── firmware
│   └── intel
│       ├── sof-ace-tplg
│       │   ├── sof-mtl-es83x6-ssp1-hdmi-ssp02.tplg
│       │   └── sof-mtl-rt711.tplg
│       └── sof-ipc4
│           └── mtl
│               └── sof-mtl.ri
└── linux-kernel-overlay
    ├── linux-headers-*-mainline-tracking-*_amd64.deb
    ├── linux-image-*-mainline-tracking-*_amd64.deb
    └── linux-libc-dev_*_amd64.deb
```

Note: The exact package version or name may be changed in different releases, so use * in the above filename as an example.

4.2 For localhost deployment, you must manually install the overlay kernel.

```
# sudo dpkg -i /temp/linux-overlay-kernel/*.deb
```

Modify the grub option to boot from the kernel, and then reboot to the kernel before following the Ansible deployment.

```
# sudo vim /etc/default/grub
```

GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux * -mainline-tracking- *"

Note: Change above * to match the kernel version you installed.

```
# sudo update-grub
# sudo reboot
```

For remote deployment (Ansible Host and Target Node are not on the same machine), there is no need to do the kernel installation manually.

Step 3 – Configure

Configure

The **On Prem AI Box** configuration profile (`on_prem_aibox`) is used for this deployment.

Target Host

1. Generate the configuration files:

```
# export PROFILE=on_prem_aibox  make examples ARCH=core
# cp examples/k8s/${PROFILE}/inventory.ini .
```

Note:
1) The AI Box is deployed on the target (localhost) so the *inventory.ini* file does not need updates.
2) If the target device is Intel® Core Ultra ™ processor, the above ARCH parameter in the make command line should be specified as :

```
  make examples ARCH=ultra
```

2. Copy `group_vars` and `host_vars` directories to the project root directory:

```
# cp -r examples/k8s/${PROFILE}/group_vars examples/k8s/${PROFILE}/host_vars .
```

3. Update the `host_vars` filename with the target machine's hostname:

```
# mv host_vars/node1.yml host_vars/localhost.yml
```

4. If the server is behind a proxy, update *group_vars/all.yml* by updating and uncommenting the lines for `http_proxy`, https_proxy, and `additional_no_proxy`.

```
## Proxy configuration ##
http_proxy: "http://proxy.example.com:port"
https_proxy: "http://proxy.example.com:port"
additional_no_proxy: ".example.com,mirror_ip"
```

If the target device is Intel® Core™ Ultra processor and you store the NDA packages in different locations in the above step 2, update the below paths:

```
update_kernel: true
nda_kernel_path: "/tmp/linux-kernel-overlay"
nda_firmware_path: "/tmp/firmware"
nda_driver_path: "/tmp/driver"
```

5. Set "intel_inband_manageability_enabled" to true and configure "intel_inband_manageability_mode" as "cloud" or "inbc"(local mode). For this document, we will be using the “inbc” mode:

```
intel_base_container_enabled: true
intel_inband_manageability_enabled: true   # Supported values for mode are 'inbc', 'cloud'.

# If local inbc option is chosen, then provisioning will be performed automatically,
otherwise provisioning should be run manually using the provision-tc command.
# For more information, refer to: https://github.com/intel/intel-inb-
manageability/blob/develop/docs/In-Band%20Manageability%20Installation%20Guide%20Ubuntu.md

intel_inband_manageability_mode: 'inbc'
```

6. Apply required patches for Kubespray (even though we do not install Kubernetes, it is needed for compatibility with other Ansible scripts):

```
# ansible-playbook -i inventory.ini playbooks/k8s/patch_kubespray.yml
```

7. (Recommended) You can check the dependencies of components enabled in `group_vars` and `host_vars` with the package dependency checker:

```
# ansible-playbook -i inventory.ini playbooks/preflight.yml
```

8. (Optional) Verify that Ansible can connect to the target server by running the following command and checking the output generated in the *all_system_facts.txt* file:

```
# ansible -i inventory.ini -m setup all > all_system_facts.txt
```

Step 4 – Deploy

Target Host

Deploy

Now the BMRA `on_prem_aibox` configuration profile can be deployed on the bare metal system by using the following command:

```
# ansible-playbook -i inventory.ini -b -K playbooks/on_prem_aibox.yml
```

Step 5 – Validate Video Analytics

Target Host

Validate

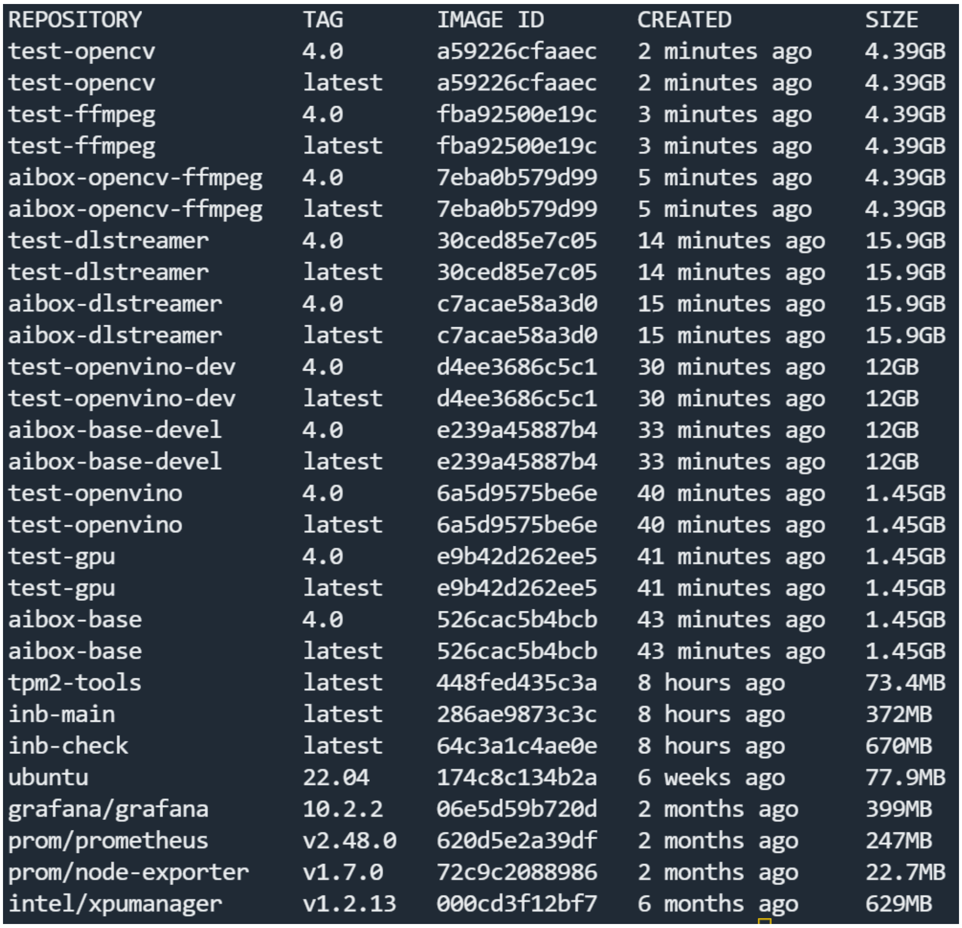
1. After the successful deployment of the `on_prem_aibox` profile, the base container-related Docker files and scripts are generated in the following location.

```
# ls /opt/intel/base_container/
      dockerfile/          # Base container Dockerfiles and build scripts
      test/                # Test container Dockerfiles, build scripts, and test
scripts
```

2. You can use the build and test scripts to build and test the base containers. Following is an example to build and test the `dlstreamer` base container. The test uses Intel® DL Streamer to detect cars in an input video.

```
# cd /opt/intel/base_container/dockerfile
# ./build_base.sh
# ./build_dlstreamer.sh

# cd /opt/intel/base_container/test
#./test_dlstreamer.sh
```



```
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
test-opencv            4.0       a59226cfaaec   2 minutes ago    4.39GB
test-opencv            latest    a59226cfaaec   2 minutes ago    4.39GB
test-ffmpeg            4.0       fba92500e19c   3 minutes ago    4.39GB
test-ffmpeg            latest    fba92500e19c   3 minutes ago    4.39GB
aibox-opencv-ffmpeg    4.0       7eba0b579d99   5 minutes ago    4.39GB
aibox-opencv-ffmpeg    latest    7eba0b579d99   5 minutes ago    4.39GB
test-dlstreamer        4.0       30ced85e7c05   14 minutes ago   15.9GB
test-dlstreamer        latest    30ced85e7c05   14 minutes ago   15.9GB
aibox-dlstreamer       4.0       c7acae58a3d0   15 minutes ago   15.9GB
aibox-dlstreamer       latest    c7acae58a3d0   15 minutes ago   15.9GB
test-openvino-dev      4.0       d4ee3686c5c1   30 minutes ago   12GB
test-openvino-dev      latest    d4ee3686c5c1   30 minutes ago   12GB
aibox-base-devel       4.0       e239a45887b4   33 minutes ago   12GB
aibox-base-devel       latest    e239a45887b4   33 minutes ago   12GB
test-openvino          4.0       6a5d9575be6e   40 minutes ago   1.45GB
test-openvino          latest    6a5d9575be6e   40 minutes ago   1.45GB
test-gpu               4.0       e9b42d262ee5   41 minutes ago   1.45GB
test-gpu               latest    e9b42d262ee5   41 minutes ago   1.45GB
aibox-base             4.0       526cac5b4bcb   43 minutes ago   1.45GB
aibox-base             latest    526cac5b4bcb   43 minutes ago   1.45GB
tpm2-tools             latest    448fed435c3a   8 hours ago      73.4MB
inb-main               latest    286ae9873c3c   8 hours ago      372MB
inb-check              latest    64c3a1c4ae0e   8 hours ago      670MB
ubuntu                 22.04     174c8c134b2a   6 weeks ago      77.9MB
grafana/grafana        10.2.2    06e5d59b720d   2 months ago     399MB
prom/prometheus        v2.48.0   620d5e2a39df   2 months ago     247MB
prom/node-exporter     v1.7.0    72c9c2088986   2 months ago     22.7MB
intel/xpumanager       v1.2.13   000cd3f12bf7   6 months ago     629MB
```

3. On completion of the test, the results can be checked. If the test is successful, you see `PASSED` in the test result file.

```
# cd ~/nep_validator_data/
# cat test_dlstreamer_result_aibox-dlstreamer
```

4. On successful test completion, the output video can be seen marked with rectangle bounding boxes and object labels in the `videos` directory:

```
# ls ~/nep_validator_data/videos
output_person-vehicle-bike-detection-2004.mp4
```

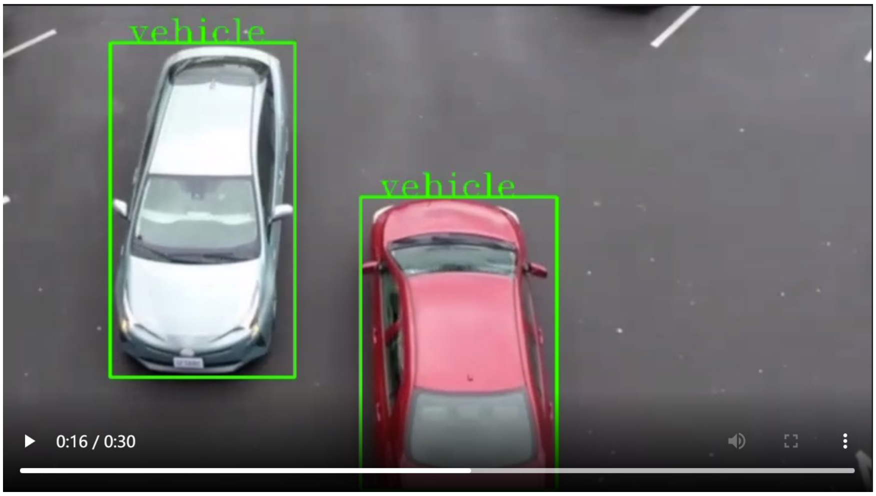


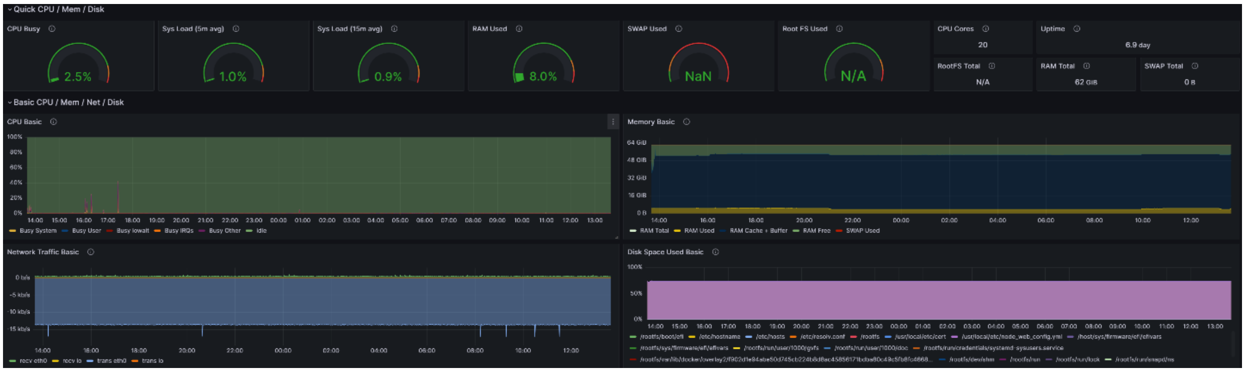
Figure 3: Edge AI Box test results with rectangle bounding boxes and object labels over the videos

Additional feature verification tests for the `on_prem_aibox` configuration profile can be found here.

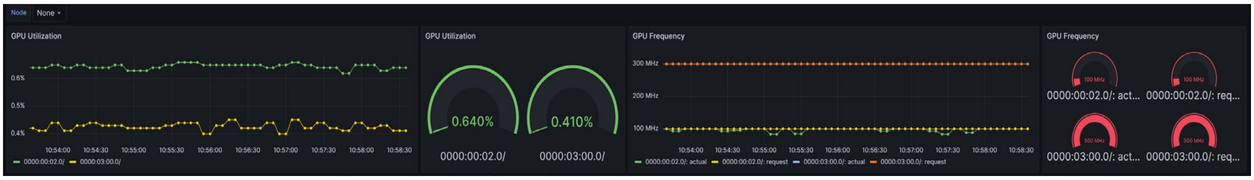
Step 6 – Validate Telemetry

1. After the successful deployment of the `on_prem_aibox` profile, telemetry related services are started automatically. User can use browser to open built-in dashboards to view the telemetry graphs.
   - If user uses the device with local display, they can directly login to https://127.0.0.1:3000/ with local browser.
   - If user uses the device remotely via ssh, they can forward port 3000 to a local PC by using the below command, then login to https://127.0.0.1:3000 with local browser.
     - "ssh -L 3000:127.0.0.1:3000 user@aibox_device_ip"

   The default username and password are admin/admin. The first login will ask the user to change the password.
2. After login, click left top "menu" button, navigate to "Home -> Dashboards -> General" to view the available dashboards.
   - Double-click node-exporter dashboard to see CPU and OS telemetries.



   - Double-click xpumanager dashboard to see GPU telemetries.



Note: The GPU telemetries from xpumanager does not work on the Intel® Core™ Ultra processor, and will be supported in the future releases.

Step 7 – Validate Intel® AI Boost on Intel® Core™ Ultra Processor

1. Use the below command to check whether the Intel® AI Boost driver is installed correctly.

```
$ lsmod | grep vpu
```

The below output is shown when the VPU driver module is loaded.

```
intel_vpu                 245760   0
drm_shmem_helper           24576   1 intel_vpu
drm                       659456   12
intel_vpu,drm_kms_helper,drm_shmem_helper,drm_display_helper,drm_buddy,i915,ttm
```

2. Use the below command to run the Intel® AI Boost test and check the result.

```
$ cd /opt/intel/vpu-linux-driver/bin
$ ./vpu-benchmark-test -m /usr/local/share/vpu/validation/models/20230620_vpu-models-por-
ir_v11_ov_2023.0.0-10926-b4452d56304/resnet-50-pytorch/onnx/FP16-INT8/resnet-50-pytorch.xml
```

The below output is shown when the test is PASSED, and the throughput FPS is shown.

```
…
[Step 11/11] Dumping statistics report
[ INFO ] Execution Devices: [ 3720 ]
[ INFO ] Count:                     40536 iterations
[ INFO ] Duration:                  60006.09 ms
[ INFO ] Latency:
[ INFO ]    Median:                 5.90 ms
[ INFO ]    Average:                5.92 ms
[ INFO ]    Min:                    3.55 ms
[ INFO ]    Max:                    13.61 ms
[ INFO ] Throughput:                675.53 FPS

real    1m1.455s
user    0m8.133s
sys     0m1.151s
+ set +x
Run benchmark_app on /usr/local/share/vpu/validation/models/20230620_vpu-models-por-
ir_v11_ov_2023.0.0-10926-b4452d56304/resnet-50-pytorch/onnx/FP16-INT8/resnet-50-pytorch.xml
with success
Test PASSED
```

Step 8 – Validate Remote Device Management

The RDM features are required to trigger a remote reboot, update the system packages by apt package, and update the container image for pulling a public Docker image from the default hub. The deployment is based on "on_prem_aibox" profile with "intel_inband_manageability_enabled" enabled, and the supported modes are "cloud" and "inbc".

8.1. Steps to verify RDM with inbc mode

1. Make sure the on_prem_aibox deployment is passed in the previous step.
2. Run inbc cmd to pull a Docker image.

```
# inbc aota --app docker --command pull --version 1.0 --containertag nginx:latest
```

3. The expected output is shown below:

```
…
containertag nginx:latest
    INFO:Connected to MQTT broker: localhost on port: 8883
    Subscribe to: manageability/response
    Subscribe to: manageability/event
    INFO:Publishing message: <?xml version="1.0" encoding="utf-
8"?><manifest><type>ota</type><ota><header><type>aota</type><repo>remote</repo></header><ty
pe><aota><cmd>pull</cmd><app>docker</app><deviceReboot>no</deviceReboot><version>1.0</versi
on><containerTag>nginx:latest</containerTag></aota></type></ota></manifest> on topic:
manageability/request/install with retain: False
    INBC command-line utility tool
    |INFO:Message received: Command: install_check passed. Message: Install check passed.
on topic: manageability/event
    -INFO:Message received: {"status": 200, "message": "COMMAND SUCCESSFUL"} on topic:
manageability/response AOTA Command Execution is Completed
    INFO:INBC code: 0
    INFO:Disconnected from MQTT broker
```

4. Confirm that the NGINX image exists on the system:

```
# docker images |grep nginx
nginx    latest     a8758716bb6a    2 months ago     187MB
```

Reference Documentation

The *Network and Edge Bare Metal Reference System Architecture User Guide* provides information and a full set of installation instructions for a BMRA.

The *Network and Edge Reference System Architectures Portfolio User Manual* provides additional information for the Reference Architectures including a complete list of reference documents.

The *Edge AI Box* website provides more information for the sample test cases and usage for version 3.1.

Other collaterals, including technical guides and solution briefs that explain in detail the technologies enabled in the Reference Architectures are available in the following location: Network & Edge Platform Experience Kits.

Document Revision History

REVISION	DATE	DESCRIPTION
001	September 2023	Initial release.
002	October 2023	Updated BMRA version to 23.10 and added telemetry services to Edge AI Box.
003	January 2024	Updated BMRA version to 24.01 and added MTL hardware support with AI Box Remote Device Management.

No product or component can be absolutely secure.

Intel technologies may require enabled hardware, software, or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

0124/DN/WIT/PDF 788715-003US