

Robot basic architecture based on SSES Blueprint

Installation guide

V2.0 11/09/2022

Table of contens

1 Introduction

The guide covers the installation details which are related to Robot basic architecture based on SSES Blueprint.

This guide covers detailed information of the various types of deployments, detailed steps and what are the various components it will install. In addition, the guide provides information on hardware requirements, prerequisite software and minimum hardware requirements. On successful deployment, IoT Gateway、PC/Server for robot control、Multi Sensor Module will be installed.

2 License

Apache License 2.0

2.1 How to use this document

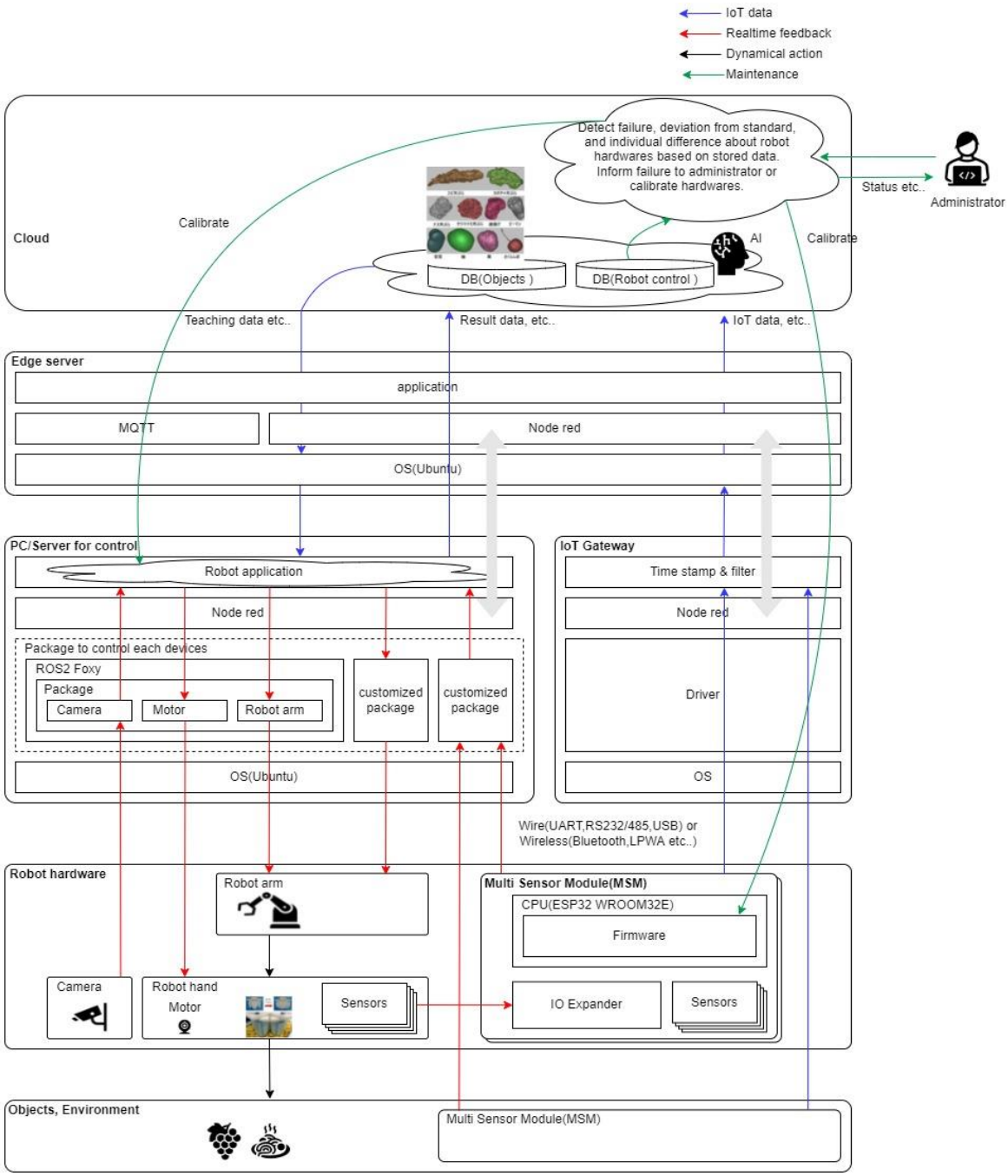
The document includes details of prerequisites, pre-installation and installation steps.The prerequisites and pre-installation software and hardware should be ready before executing the installation steps.

2.2 Deployment Architecture

The Deployment Architecture consists of the following nodes

- IoT Gateway
- PC/Server for robot control
- Multi Sensor Module

Note: Other Nodes in the figure will be released from the next major release.


IoT data
Realtime feedback
Dynamical action
Maintenance
Cloud
Detect failure, deviation from standard, and individual difference about robot hardwares based on stored data. Inform failure to administrator or calibrate hardwares.
Status etc..
Administrator
Calibrate
AI
Calibrate
DB(Objects)
DB(Robot control)
Teaching data etc..
Result data, etc..
IoT data, etc..
Edge server
application
MQTT
Node red
OS(Ubuntu)
PC/Server for control
Robot application
Node red
Package to control each devices
ROS2 Foxy
Package
Camera
Motor
Robot arm
customized package
customized package
OS(Ubuntu)
IoT Gateway
Time stamp & filter
Node red
Driver
OS
Wire(UART,RS232/485,USB) or Wireless(Bluetooth,LPWA etc..)
Robot hardware
Robot arm
Camera
Robot hand
Motor
Sensors
Multi Sensor Module(MSM)
CPU(ESP32 WROOM32E)
Firmware
IO Expander
Sensors
Objects, Environment
Multi Sensor Module(MSM)

3 Pre-Installation Requirements

3.1 Hardware Requirements

The number of hardware required depends mainly on the use case and the size of the enterprise. A use case may include one or more IoT Gateways, one or more PC/Server for robot controls, and one or more Multi Sensor Modules.

3.1.1 Minimum Hardware Requirements

N/A

3.1.2 Recommended Hardware Requirements

IoT Gateway

Raspberry Pi 4 Model B / 4GB が推奨されます

CPU	Broadcom BCM2711
Architecture	ARM
RAM	4GB
Disk	16GB

PC/Server for robot control

CPU	4 core
Architecture	x64_AMD
RAM	4GB
Disk	500GB

3.2 Software Prerequisites

・The IoT Gateway has raspbian pre-installed.

・PC/Server for robot control has Ubuntu pre-installed.

3.3 Database Prerequisites

N/A

3.4 Other Installation Requirements

N/A

4 Installation High-Level Overview

4.1 Bare Metal Deployment Guide

4.1.1 Install Bare Metal Jump Host

4.1.1.1 IoT Gateway

4.1.1.1.1 Modify the system configuration

4.1.1.1.1.1 Update the OS to the latest version.

```
$ sudo apt update
$ sudo apt upgrade
```

####### 4.1.1.1.1.1.1 Troubleshooting

If ”sudo apt update” command fail because of the large difference between the machine time and the actual time, you follow the steps in Chapter Setting_time_server first.

4.1.1.1.1.2 Display setting

During normal operation, the system operates with only the main unit and no display/keyboard/mouse is connected. It is basically a remote connection. When the power is turned on without the display connected, make the following settings so that the screen is displayed properly via VNC connection.
Edit /boot/config.txt to enable the following configuration items and remove the leading.

```
#hdmi_force_hotplug=1
```

4.1.1.1.1.3 Starting Node-RED automatically

Configure Node-RED to start automatically when power is turned on.Run the following command to allow Node-RED to start automatically when power is turned on.

```
$ sudo systemctl enable nodered.service
```

To stop the automatic startup when the power is turned on, execute the following command.

```
$ sudo systemctl disable nodered.service
```

The following is the default file that will be automatically started when power is turned on. Please create as needed.
~/.node-red/flow_$(hostname).json
Example: If hostname is raspberrypi,
~/.node-red/flow_raspberrypi.json .

4.1.1.1.1.3.1 Troubleshooting

If node-red is not installed, install it with the following command

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
https://nodered.jp/docs/getting-started/raspberrypi

$ sudo systemctl start nodered.service
```

4.1.1.1.1.4 Allowing VNC and SSH

Maintenance of the gateway will be done via the network. Therefore, we need to allow access to VNC and SSH. There are two ways to allow access: using the OS GUI or from the terminal screen.
Warning: Wireless LAN devices are used for other purposes in a later step, so connect with a wired LAN device.

4.1.1.1.1.4.1 Using GUI

Click on the raspberry symbol in the upper left corner of the screen to bring up the menu. Hover your cursor over "Settings" at the bottom of the menu. This will bring up the lower menu on the right. Select "Raspberry Pi Settings" from the menu and click on it.

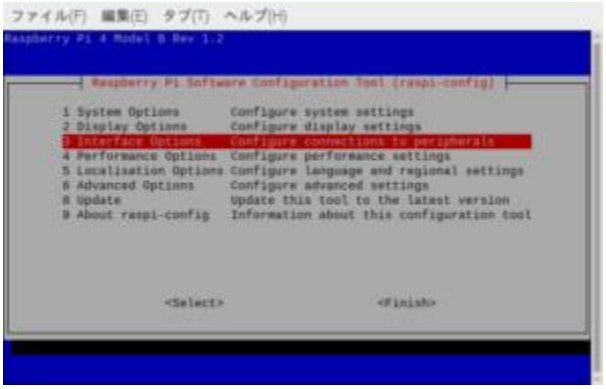
This will bring up the Raspberry Pi configuration screen as shown on the left. Select the "Interface" tab, select the SSH and VNC enable buttons, and press OK to complete the configuration.

4.1.1.1.1.4.2 Using terminal

To do this from a terminal screen, open a terminal screen and enter the following command

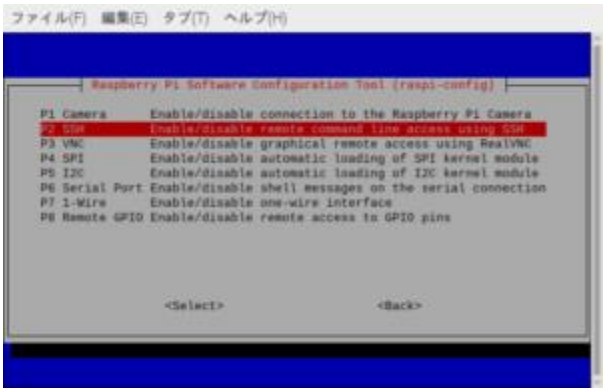
```
$ sudo raspi-config
```

This will bring up the following configuration screen.



Use the cursor keys to move the red band to "3 Interface Options", and press the Enter key to move to the next screen.
To move to <Select> and <Finish>, press the Tab key to move the red band down. To exit, use the left and right cursor keys to move the red band to <Finish>, and press the Enter key.
You will be returned to the original terminal screen.

On the next screen, you will see the Enable/Disable menu for SSH and VNC, so use the cursor keys to move the red band there as well, and press Enter.
To move to <Select> and <Back> below, press the Tab key, and the red strip will move down. To return to the previous screen, use the left and right cursor keys to move the red band to <Back>, and press the Enter key. You will return to the upper selection menu.



You will then see a screen asking, "Do you want to allow? screen will be displayed.
The figure shows the SSH permission screen.
The "Yes" mark is red, so press the Enter key. The screen will change to "Permission granted". To select "No", use the cursor

keys to move to the right. To select "No", move to the right with the cursor keys. The red band will move to "No".
To select "No," move the cursor key to the right. You will be taken back to the initial settings screen, so please allow VNC in the same way.

4.1.1.1.1.5 Setting time server

Set the time server of the upper network to synchronize the gateway with the time.

How to change the NTP server of the gateway
This section describes how to change the NTP server of the gateway. Please check the URL you want to connect to as an NTP server beforehand. As an example, we will access the address (ntp.nict.jp) of the NTP service provided by the National Institute of Information and Communications Technology (NICT), which determines and maintains Japan's standard time.

4.1.1.1.1.5.1 Modify setting file

Write the address of the NTP service in the timesyncd.conf file.
First, back up the original file by copying it.

```
$ cd /etc/systemd
$ sudo cp timesyncd.conf timesyncd.conf.bak
```

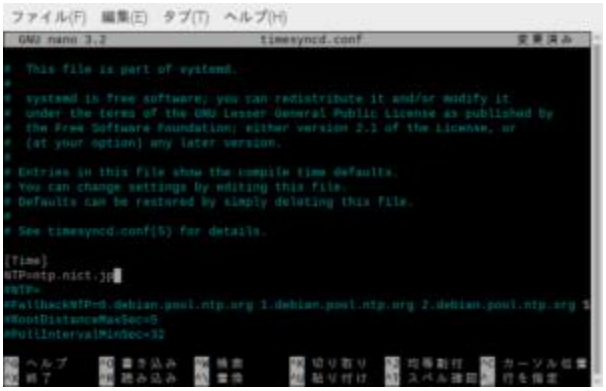
Modify file through editor.

```
$ sudo nano timesyncd.conf
```

Afrer [Time], write following

```
NTP=ntp.nict.jp
```

If there are multiple connections, put them in a row with a space as a separator. If it fails to connect to the first one, it will try to connect to the second one, and if it fails to connect to the second one, it will try to connect to the third one, and so on. It will not go beyond the first connection, so it is recommended to arrange them in the order you want to connect.



4.1.1.1.1.5.2 Reload the daemon and restart the service.

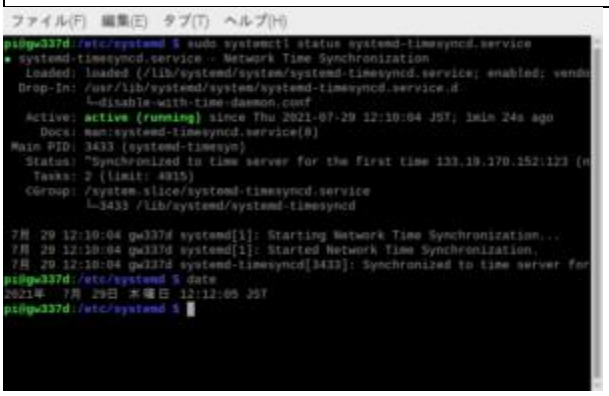
Execute the command to enable NTP. Next, reload the daemon, and finally restart the timesyncd service to complete the configuration.

```
$ sudo timedatectl set-ntp true
$ sudo systemctl daemon-reload
$ sudo systemctl restart systemd-timesyncd.service
```

4.1.1.1.1.5.3 Confirmation that the service is started and that the time is correct.

Confirm that the service is Active : active (running) by pressing the start confirmation command.

```
$ sudo systemctl status systemd-timesyncd.service
```



Type the following command to make sure that the OS time and the NTP service time match.

```
$ date
```

You can check the NTP service of the National Institute of Information and Communications Technology (NICT), which we used as an example, by accessing the following address.
https://www.nict.go.jp/JST/JST5.html

4.1.1.1.2 Installing software

Install some of the application software needed to build the gateway.

4.1.1.1.2.1 Installing GTKTerm

Install GTKTerm, a serial port terminal software that can be used with Raspberry Pi OS and Linux systems such as Ubuntu. This is used to check the USB connection and Bluetooth connection of the sensor module.

```
$ sudo apt install gtkterm
```

4.1.1.1.2.2 Installing hostapd

Install hostapd, a software that provides services for wireless LAN access points. It is used to realize AP mode (Access Point Mode), which is used for the initial configuration of the gateway.

```
$ sudo apt install hostapd
```

4.1.1.1.2.3 Installing dnsmasq

Install dnsmasq, a software that provides DNS/DHCP server functions for small-scale networks. Assign an IP address using the DHCP server function in AP mode, which is used for initial configuration of the gateway.

```
$ sudo apt install dnsmasq
```

4.1.1.1.2.4 Installing blueman

Install Blueman, which is a Bluetooth manager tool that allows you to make Bluetooth connections with sensor modules via GUI.

```
$ sudo apt install blueman
```

4.1.1.1.3 Connecting sensor module

Refer to exhibit material.

4.1.1.1.4 Setting mDNS(multicast Domain Name System)

Use a DNS service to map IP addresses to hostnames. This is not necessary for networks with static IP. Just specify the IP address directly.

If your Raspberry Pi 4 has been configured with a gateway up to this point, the DNS server should be up and running. This will confirm that the service is up and running.

```
$ systemctl status avahi-daemon.service

Active : active (running)
```

4.1.1.1.4.1 Creating /etc/avahi/services/ssh.service

Create a new /etc/avahi/services/ssh.service and include the following information.

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">

<service-group>
  <name replace-wildcards="yes">%h</name>

  <service>
    <type>_ssh._tcp</type>
    <port>22</port>
  </service>
</service-group>
```

4.1.1.1.4.2 Modifying hostname

The name in the DNS service will be $(hostname).local. Normally, the hostname of the Raspberry Pi is raspberrypi, so if you leave the hostname unchanged, DNS will assign the name raspberrypi.local to the IP address. If you want to change the name to something else, you need to change the hostname.

If you do not want to change the hostname, please jump to "Restart DNS Service".

In order to change the hostname, you need to modify two files.

Modifying /etc/hostname

Use an editor to change /etc/hostname. Before the change, it is

only listed as raspberrypi.

Adding /etc/hosts
Use an editor to change /etc/hosts.

```
172.0.1.1		raspberrypi
```

Adding following.

```
172.0.1.1		<NewHostName>
```

<NewHostNam> should contain the hostname you wish to change.

4.1.1.1.4.3 Restarting DNS service

Restart DNS service.

```
$ sudo /etc/init.d/avahi-daemon restart
```

You can now access Node-RED at http://$(hostname).local:1880/.

4.1.1.1.5 Installing the standard flow of Node-RED

The following is simple flow to receive data from MSM and insert documents to mongodb.

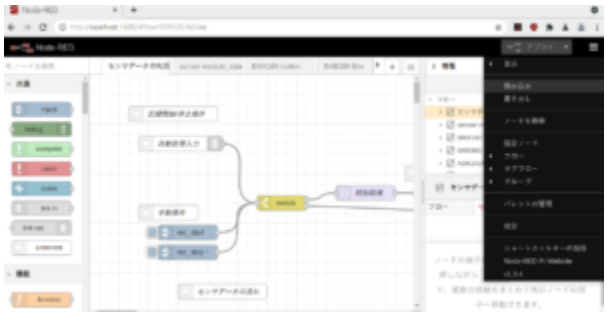
PDH_Flow.json

You need to change serialport name, Mongo URL,ID and password in the flow depends on your environment.

```
{
    "id": "84e623793b0ba795",
    "type": "serial-port",
    "serialport": "/dev/rfcomm1",
    "serialbaud": "1000000",
    "databits": "8",
    "parity": "none",
    "stopbits": "1",
    "waitfor": "",
    "dtr": "none",
    "rts": "none",
    "cts": "none",
    "dsr": "none",
    "newline": "¥¥n",
    "bin": "false",
    "out": "char",
    "addchar": "",
    "responsetimeout": "10000"
},
{
```

```
        "id": "f9365a9a34f2501d",
        "type": "mongodb",
        "hostname": "127.0.0.1",
        "topology": "direct",
        "connectOptions": "",
        "port": "27017",
        "db": "<Your Mongo URL>",
        "name": "",
        "credentials": {
            "user": "<Your mongo id>",
            "password": "<Your mongo password>"
        }
    }
```

Select "Load" from the three-line menu in the upper right corner.

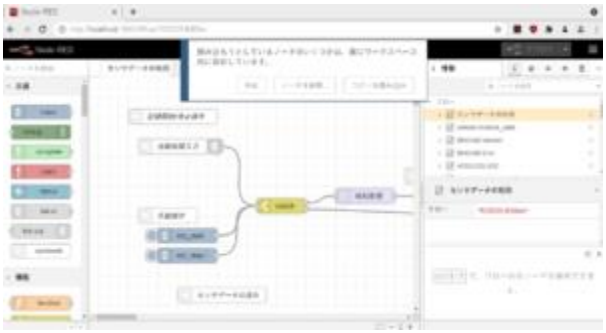


The Import Flow window will open. Click "Select File to Import".

When the file selection window opens, select the flow (.json file) that you just downloaded and click the "Open" button.

The flow you are trying to load is now displayed.
In this state, click the "Load" button.

If you load a flow from a previous flow that still exists, you may get the message "Some of the nodes you are trying to load already exist in the workspace.
In this case, please complete the loading as "Load a copy" and edit it to your desired process in the flow editing screen.



・If you want to return to the initial state completely, please delete all the flows before loading them.

4.1.1.1.6 Setting password of Node-RED

When accessing the Node-RED flow screen, enter http://<IP ADD>:1880. In this case, the password is not managed by default. Follow the steps below to restrict the password.

4.1.1.1.6.1 Outline

The user name and password can be configured by writing them in ~/.node-red/settings.js.

Passwords are hashed using the bcrypt algorithm for added security, so they must be hashed beforehand using a separate command.

4.1.1.1.6.2 Hashing password

Hashing a password is done with the following command.

When you enter the command, you will be prompted for a Password, enter it and hit the return key, and the hashed password will be returned to the screen. Copy this password and paste it into the setting.js file.

```
$ node-red admin hash-pw [RET]
Password:XXXXXX[RET]
$2a$08$9tHg3MHtmRDRiBlCOL.EiOR0qB3vNjP1eFF2VYouDWrrd
OzjwrHdy
```

4.1.1.1.6.3 Modifying settings.js

Use an editor to open the ~/.node-red/settings.js file.

In the middle of the file, there is a description of the user name and password as shown below. Remove the commented-out "//" at the beginning of the line, change the description to the user name and the hashed password output above, and save the Save the file.

<Before>

```
//adminAuth: {
// type: "credentials",
// users: [{
```

```
// username: "admin",
// password:
"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogEN
OMcxWV9DN.",
// permissions: "*"
// }]
//},
```

<After>

```
adminAuth: {
type: "credentials",
users: [{
username: "passwd",
password:
"$2a$08$9tHg3MHtmRDRiBlCOL.EiOR0qB3vNjP1eFF2VYouDWrr
dOzjwrHdy",
permissions: "*"
}]
},
```

After saving the file, restart Node-RED and your username and password will be valid.

```
$ sudo systemctl restart nodered
```

4.1.1.2 PC/Server for robot control

4.1.1.2.1 Installing GTKTerm

See Gateway GTKTerm installation instructions.

4.1.1.2.2 Installing Node-RED.

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

4.1.1.2.3 Autostart setting and service start

```
$ sudo systemctl start nodered.service
$ sudo systemctl status nodered.service
```

4.1.1.2.4 Installing ASR

To download the install script.

```
$ wget https://raw.githubusercontent.com/signalogic/SigSRF_SDK/master/rar_packages/autoInstall_SigSRF_SDK_2022v7.sh -O- | tr -d '¥r' > autoInstall_SigSRF_SDK_2022v7.sh
```

```
$ wget https://github.com/signalogic/SigSRF_SDK/raw/master/rar_packages/Signalogic_sw_host_SigSRF_SDK_distroNN_date.rar
```

where "distroNN" is the Linux distro and version and "date" is the package date. To avoid entering the distro release version and .rar date, you can use one of the following wildcard format wget commands:

```
$ wget -r -l7 --cut-dirs 6 -nH -N -A "*SDK_Ubuntu*.rar" --content-disposition -R "*return_to*" -erobots=off https://github.com/signalogic/SigSRF_SDK/tree/master/rar_packages/
```

For the ASR version of the SDK, the following commands can be used:

```
$ wget -r -l7 --cut-dirs 6 -nH -N -A "*SDK_ASR_Ubuntu*.rar" --content-disposition -R "*return_to*" -erobots=off https://github.com/signalogic/SigSRF_SDK/tree/master/rar_packages/
```


Running the install script requires being logged in as root or as a user with sudo root privilege.

```
$ chmod 755 *.sh
```

To run the install script enter:

```
$ sudo ./autoInstall_SigSRF_SDK_2022v7.sh
```

The script will then prompt as follows:

```
1) Host
2) VM
Please select target for SigSRF software install [1-2]:
```

Select 1.

The script will next prompt for an install option:

```
1) Install EdgeStream and SigSRF Software
2) Install EdgeStream and SigSRF Software with ASR Option
3) Install EdgeStream and SigSRF Software with coCPU Option
4) Uninstall EdgeStream and SigSRF Software
5) Check / Verify EdgeStream and SigSRF Software Install
6) Exit
Please select install operation to perform [1-6]:
```

Select 2

The script will prompt for an install path:

```
Enter the path for EdgeStream and SigSRF software installation:
```

Click Enter without no path.

4.1.2 Creating a Node Inventory File

N/A

4.1.3 Creating the Settings Files

N/A

4.1.4 Running

N/A

4.2 Virtual Deployment Guide

4.2.1 Standard Deployment Overview

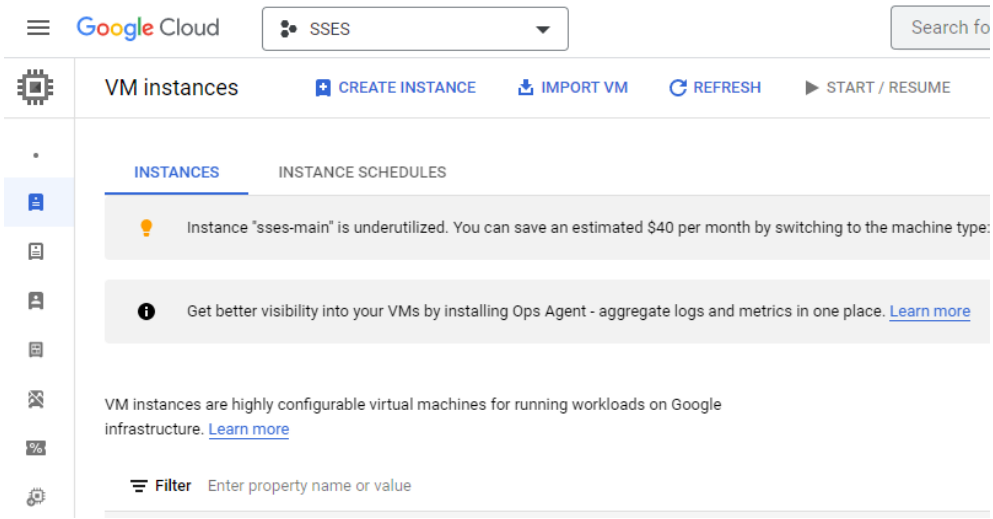
Build a Cloud/Edge Cloud environment.

As an example, we describe how to build using Google Compute Engine (GCE) of Google Cloud.

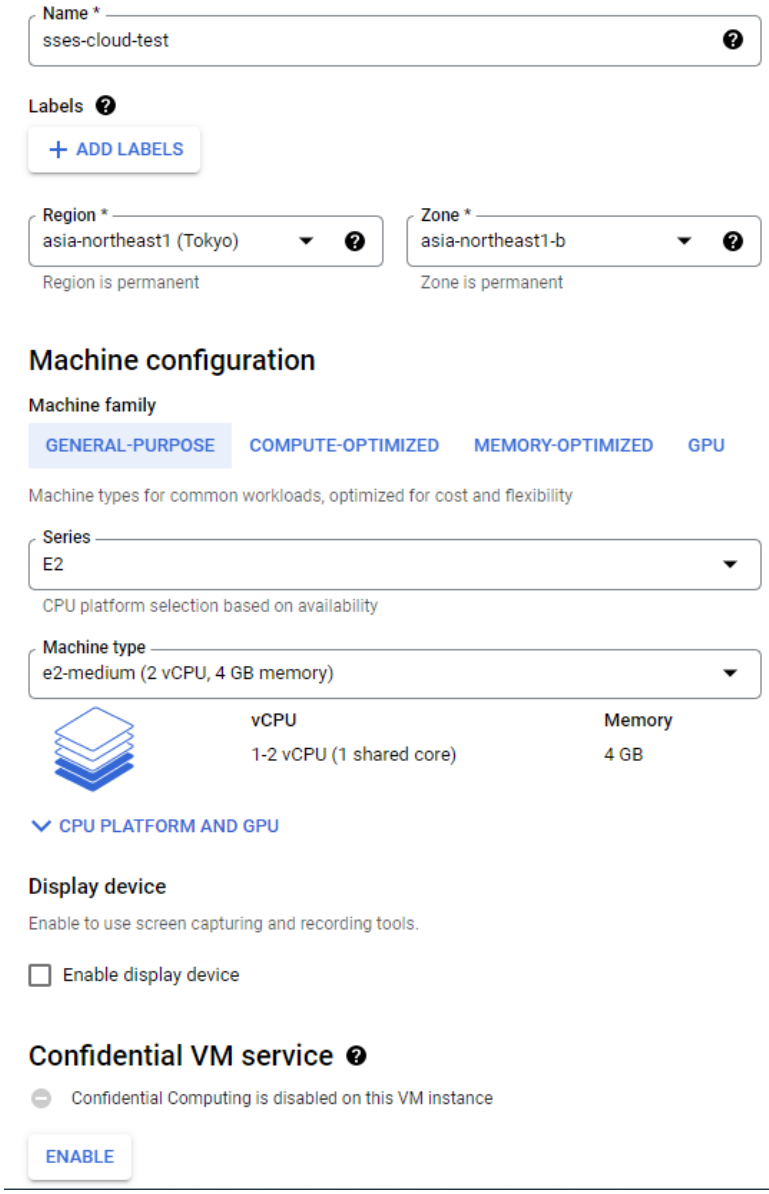
4.2.1.1 Setting up a VM with GCE

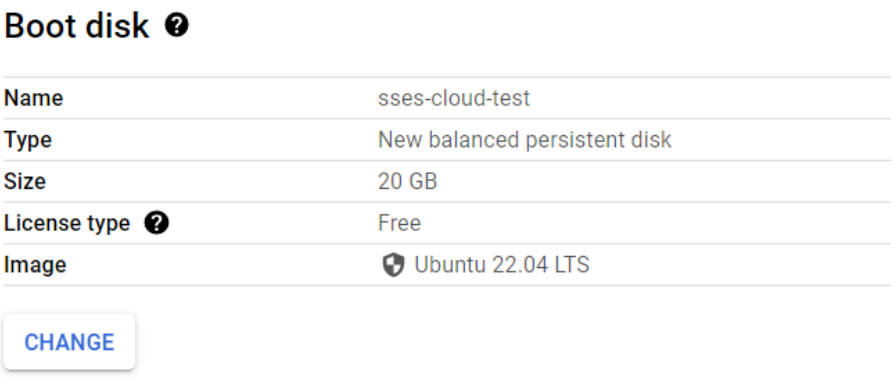
4.2.1.1.1 Deployment GCE

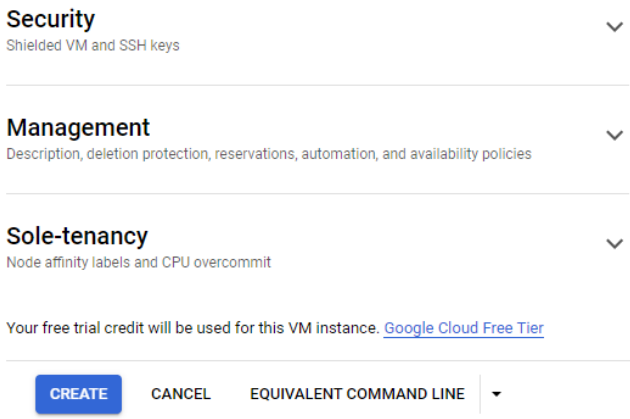
Click CREATE INSTANCE on the GCE operation screen



Enter the appropriate Name, Boot disk (OS, etc.), Machine configuration, and click CREATE at the bottom of the scroll.





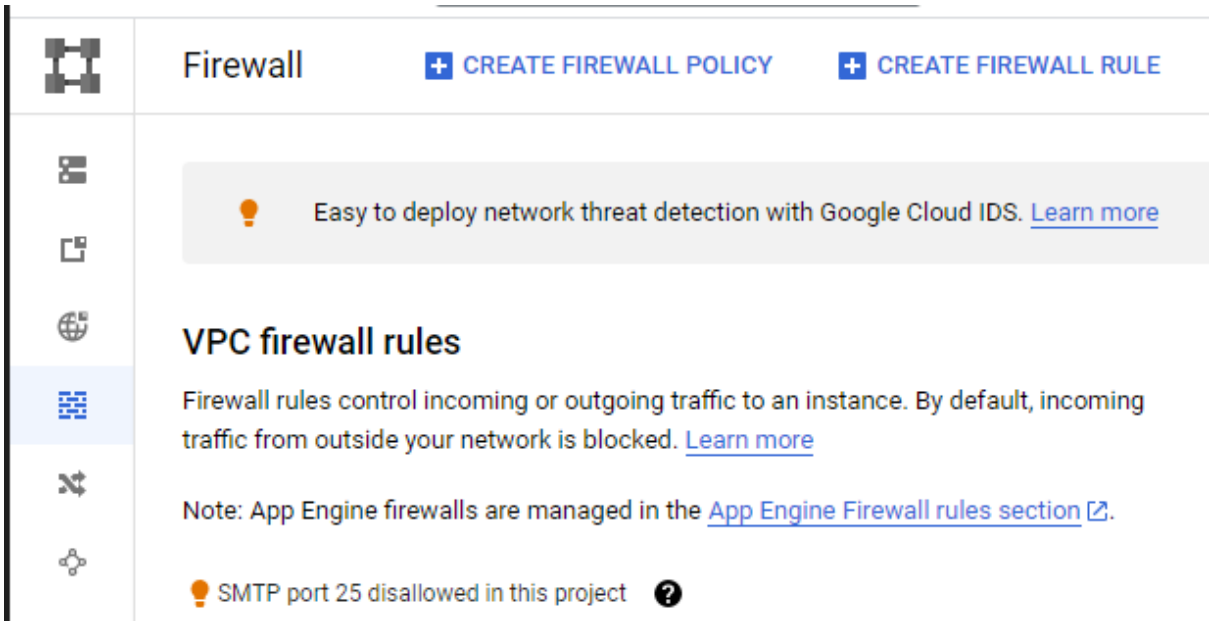


Confirm that the VM has been created

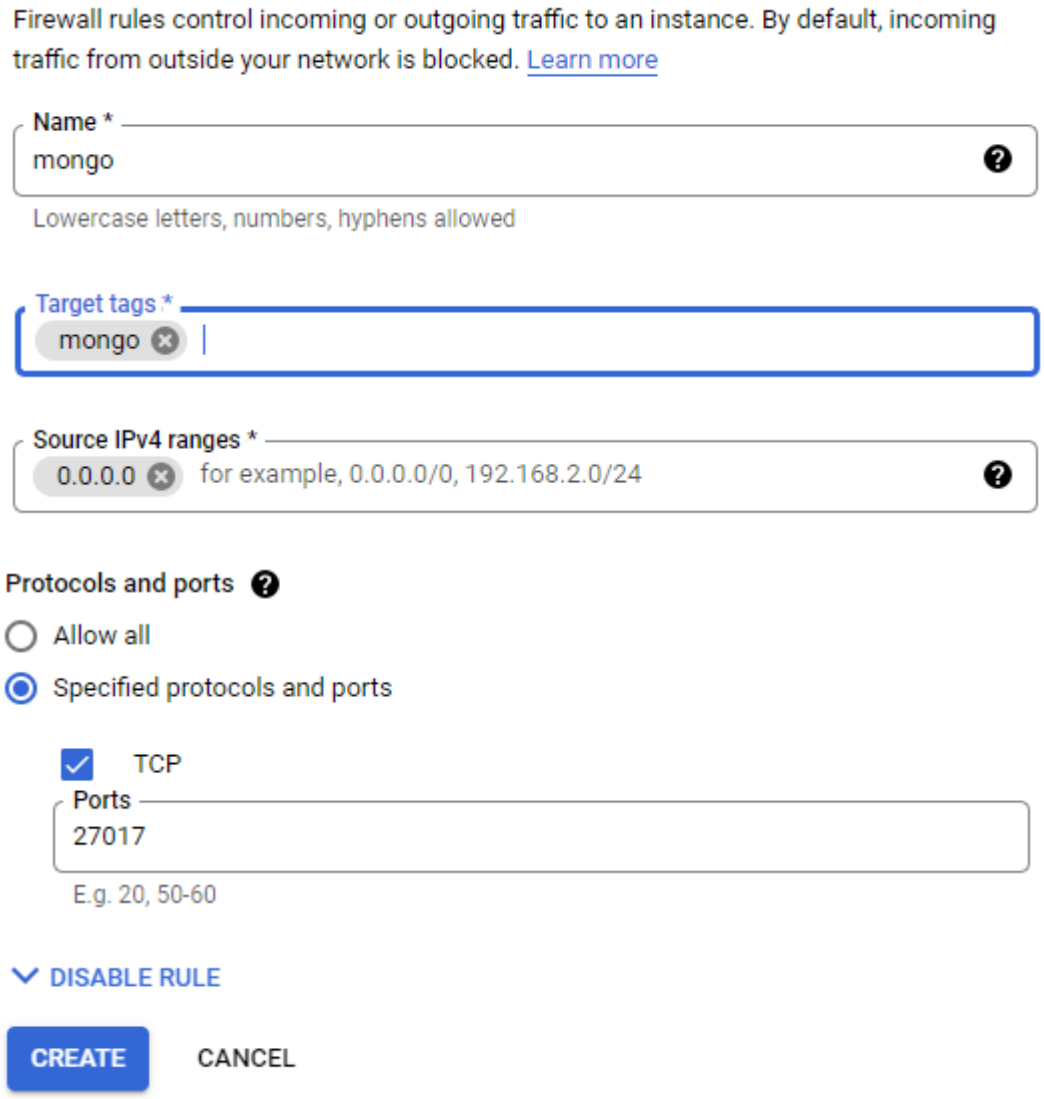
4.2.1.1.2 Setting firewall of GCE

Open moongo DB default port 27017

Click CREATE FIREWALL RULE.

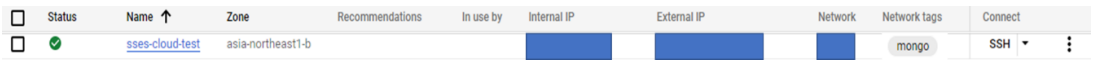


Set the appropriate Name, Target tags, Source IPv4 ranges, Ports, and click CREATE.



Verify that the rule has been created

Set the tag you set to the VM.



4.2.1.2 Installaing MongoDB

Import the public key used by the package management system. The operation should respond with an OK.

```
$ wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc |
sudo apt-key add -
```

Create the /etc/apt/sources.list.d/mongodb-org-6.0.list file for Ubuntu 20.04 (Focal).

```
$ echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
```

Reload local package database.

```
$ sudo apt-get update
```

Install Openssl.

```
$ echo "deb http://security.ubuntu.com/ubuntu focal-security main" | sudo tee /etc/apt/sources.list.d/focal-security.list
$ sudo apt-get update
$ sudo apt-get install libssl1.1
```

Install the MongoDB packages.

```
$ sudo apt-get install -y mongodb-org=6.0.2 mongodb-mongosh=6.0.2
```

Configure automatic startup of MongoDB

```
$ sudo systemctl enable mongod
```

Start MongoDB

```
$ sudo systemctl start mongod
```

4.2.2 Snapshot Deployment Overview

N/A

4.2.3 Special Requirements for Virtual Deployments

Install Jump Host

N/A

Verifying the Setup - VMs

N/A

4.3 Upstream Deployment Guide

We will release next.

Upstream Deployment Key Features

N/A

Special Requirements for Upstream Deployments

N/A

Scenarios and Deploy Settings for Upstream Deployments

N/A

Including Upstream Patches with Deployment

N/A

Running

N/A

Interacting with Containerized Overcloud

N/A

5. Verifying the Setup as defined the Akraino validation feature project plus any additional testing specific to the blue print

   Refer to test document.

6. Developer Guide and Troubleshooting

   Utilization of Images

   N/A

   Post-deployment Configuration

   N/A

   Debugging Failures

   N/A

   Reporting a Bug

   N/A

7. Uninstall Guide
8. Troubleshooting

   Error Message Guide

9. Maintenance

   Blue Print Package Maintenance

   - Software maintenance
   - Hardware maintenance

   Blue Print Deployment Maintenance

10. Frequently Asked Questions

11. License

Component	Software name	Version	License
PC/Server for control	Ubuntu	18.04	GPL
	Node red	1.3.4	Apache License 2.0
	Robot arm package		Apache License 2.0
	gtkterm		GPL 3.0
	SigSRF_SDK(ASR)		Github SigSRF License 1.0
IoT gateway	Raspbian	Raspbian GNU/Linux 10 (buster)	BSD-3-Clause,MIT,GPL
	Node red	1.3.4	Apache License 2.0
	Node.js	10.24.0	MIT
	node-red-dashboard	2.30.0	Apache License 2.0
	node-red-contribute-moment	4.0.0	Apache License 2.0
	gtkterm	0.99.7+git9d63182-2	GPL 3.0
	hostapd	2:2.7+git20190128+0c1e29f-6+deb10u3	GPL 2.0
	dnsmasq	2.80-1+rpt1+deb10u1	GPL 2.0 or 3.0
	blueman	2.0.8-1+deb10u1	GPL 3.0
Cloud/Edge Cloud	Ubuntu	22.04	GPL
	mongo	6.0.2	SSPL 1.0
MSM	arduino-esp32		LGPL 2.1

	Arduino IDE	1.8.15	AGPL 3.0
	Arduino ESP32	2.0.2	LGPL 2.1
	esp-idf	4.4	Apache License 2.0
	TaskScheduler	3.4.0	BSD-3-Clause
	Arduino_JSON	0.1.0	LGPL 2.1
	Adafruit Unified Sensor	1.1.4	Apache License 2.0
	Adafruit GFX Library	1.10.13	BSD-2-Clause
	Adafruit BusIO	1.11.0	BSD-3-Clause
	Adafruit SSD1306	2.5.1	BSD
	Adafruit_BME680	2.0.1	BSD
	Adafruit_MCP4725	2.0.0	BSD-3-Clause
	Adafruit_ADS1X15	2.3.0	BSD-3-Clause
	Adafruit_MCP9600	2.0.0	MIT

	Adafruit_MCP2300 8	2.1.0	BSD
	Adafruit_MPR121	1.1.1	MIT
	DFRobot_BMX160	-	MIT
	RPR-0521RS	-	MIT
	U8g2lib	2.31.2	BSD-2-Clause
	preMSM firmware		Apache License 2.0

12. References

SIP: https://sip-sses.net/

Building gateway: https://sip-sses.net/gateway/

13. Definitions, acronyms and abbreviations

- SIP: The Cross-ministerial Strategic Innovation Promotion Program
  SIP is a national program led by the Council for Science, Technology and Innovation （CSTI） of the Japanese Government with interdisciplinary management to realize scientific and technological innovation in our country.
  About SIP | SIP | JST
- SSES: Sensor-rich Soft End effector System
  立命館大学 SSES Platform - Ritsumeikan Univ. (sip-sses.net)

14. Revision history

Version	Date	Editor	Contents
0.1	02/08/2022	Inoue	Draft version
1.0	02/10/2022	Inoue	Review completed and published as first edition
1.1	03/04/2022	Inoue	Minor modifications to procedures
1.2	03/23/2022	Inoue	・Delete “4.1.1.1.4 transition between AP mode and CL mode” ・Modify Ubuntu version of PC/Server for control
2.0	11/09/2022	Inoue	Modify for Akraino Release 7