Cisco LoRa WAN Deployment Guide

A Prescriptive Guide for beginners .

Table of Contents

Getting Started :

Note: The Following indexes has soft links to cisco documents , please use ecopy for the hyperlinks to work, avoid printing.

IR1100 Product Overview ***Installing IR1101 Router*** ***a. Optional - Antenna Selection and Installation*** ***b. Optional - Installing the IRM-1100 Expansion Module*** ***c. Optional - IR1101 Technical Specifications Details***	IR1101 product Guides
IR1800 Product Overview ***Cisco IR1800 Series Platform Features*** ***Installing IR1800 Router*** ***a. Antennas for the IR1800 Series Router*** ***b. Installing Pluggable Interface Modules***	IR1800 Product Guides

Connecting to Router Console via Serial Port ***a. Baud rate settings*** ***b. Mandatory: Drivers to download***	Connecting to Router Serial Console via Serial cable., cable and reference links console cable (Cisco P/N CAB-CONSOLE-USB, 6ft long) **Note: when managing device locally using CLI we tend to use the serial console and connect to router CLI prompt**
Introduction to the Web User Interface ***a. WebUI Dashboard*** ***b. Day 0 Cellular Mode*** ***c. Configuration Notes***	WebUI introduction, managing the device locally and usage of WebUI in day zero cellular mode
Setting Your Computer to Connect to the Router WebUI ***a. Choosing Basic Mode option on WebUI*** ***b. Choosing Advanced Mode WebUI***	Choice of WebUI basic mode and advanced mode
Understanding Licensing options ***a. Cisco Software Licensing*** ***b. Consolidated Packages*** ***c. Network-Essentials*** ***d. Network-Advantage***	Know about licensing options **Note: Network advantage is must for LoRa PIM to work**
Image upgrade on IR1101 ***a. Installing the IOS-XE Software image*** ***b. Understanding ROMMON Images*** ***c. know about Supported File Systems*** ***d. Enabling and Disabling USB access*** ***e. Autogenerated File Directories and Files*** ***f. Flash Storage*** ***g. know about LED Indicators*** ***h. Related Documentation***	Know about image upgrade and other details about device filesystem and enabling and disabling usb, led status and other related topics. **Note: LoRa PIM is supported from 17.10.1 release on wards. It's important to upgrade your box to latest image**
Understanding Application Hosting ***a. Application Hosting*** ***b. Information About Application Hosting*** ***c. Application Hosting on the IR1101 Router*** ***d. How to Configure Application Hosting*** ***e. Installing and Uninstalling Apps*** ***f. Overriding the App Resource Configuration*** ***g. Verifying the Application Hosting Configuration*** ***h. Configuration Examples for Application Hosting***	Detail document about application hosting, how to configure and install and manage the apps using CLI and few examples. **Note: we are supporting packet forwarders as docker app. How to install and manage app on routers is critical step to learn.**
About Local Manager ***a. Overview*** ***b. Cisco IOx Local Manager Pages and Options*** ***c. Cisco IOx Local Manager Workflows***	Using local manager for app deployment and description of various tabs under local manager. Local manger is specifically designed to help manage app on routers **Note: we can manage app life cycle using local manager. Understanding app management via local manager helps.**
Cisco 4G LTE-Advanced Configuration	Document links to the cellular provision on IR1101 and IR1800.

Sample Deployment Work sheet

Router	PIM module 915 | 815	Network server + Docker App	WAN choice
IR1101	P-LPWA-900	Chirp Stack + CPF	GE
IR1800	P-LPWA-800	Thingpark + Actility Packet Forwarder	Cellular – select a proper cellular PIM module

Few LoRa Deployment Considerations

a. **Platform Choice: IR1101 vs IR1800**
   Refer to platform documentation for hardware and technical specification s to see which device meets your requirements.
b. **PIM choice: 915 PIM vs 815 PIM module**
   Currently we support two different pluggable interface modules
   - P-LPWA-900 for 915 MHz
   - P-LPWA-800 for 868 MHz
c. **PIM slot choice: LORA PIM module in Base slot vs Expansion slot on IR1101**
   For IR1101 and existing deployment where the Base PIM slot is being used for cellular, we recommend using expansion slot for lora PIM. For the IR1800 we recommend using which ever PIM slot is available.

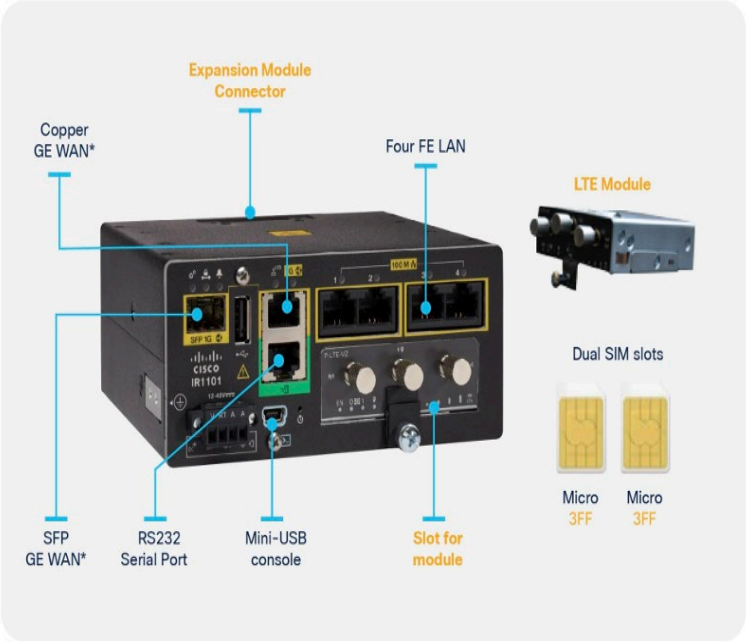
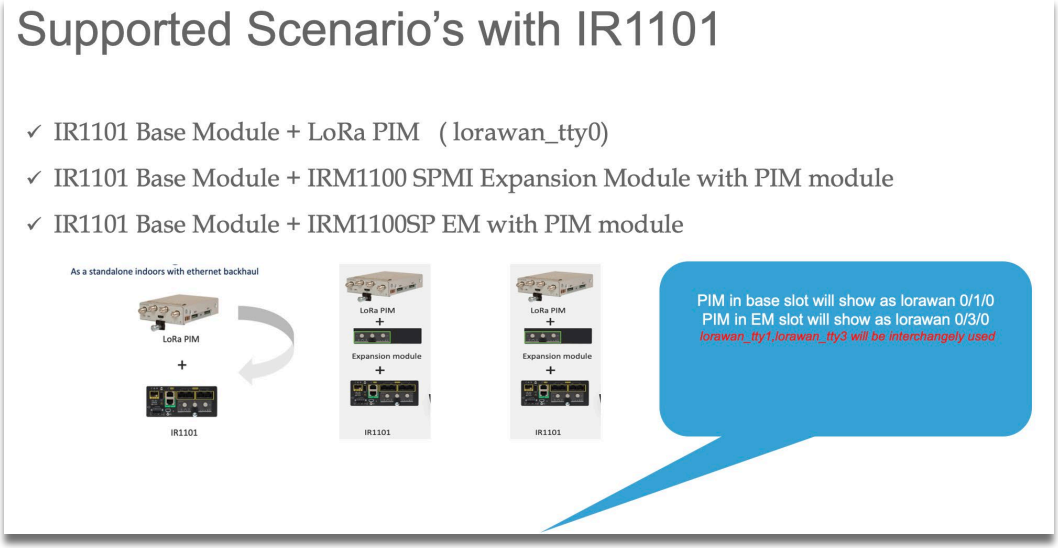


Figure 1.

Cisco IR1101 base platform front view

Figure 2.

Expansion Modules



d. LORA PIM module support in all the IR1800 pids

PIM slots are available on IR1812, IR1831, IR1833, IR1835 , any choice of PIM slot can work across all SKUs.

IR1821-K9	IR1831-K9	IR1833-K9	IR1835-K9
• 1 cellular slot, 1 Wi-Fi slot	• 2 cellular slots, 1 Wi-Fi slot	• 2 cellular slots, 1 Wi-Fi slot	• 2 cellular slots, 1 Wi-Fi slot
• 4 GB memory	• 4 GB memory	• 4 GB memory	• 8 GB memory
• CAN bus	• CAN bus	• CAN bus, PoE/PoE+, ADR GNSS slot, SSD slot	• CAN bus, PoE/PoE+, ADR GNSS slot, SSD slot
			• 4 digital I/O ports, advanced security features, 1 RS232/485 combo port

e. WAN Choice: GE or Cellular

Typical enterprise deployments vs field deployment, a GE 0/0/0 can be used as WAN for enterprise and extended enterprise cases where we have cable reach.

For outdoor deployments we can choose cellular as WAN as well. Please make sure to have a valid Cellular module and working SIM card and proper cellular data plan in place before making the cellular as WAN.

f. **Packet forwarder choice: "Common packet forwarder (CPF)"app vs "Actility packet forwarder" (APF)**

We support two packet forwarders based on your choice of network server we can use common packet forwarder or Actility packet forwarder.

g. **CPF authentication choice: Choice of authentication with CPF deployment**

Common packet forwarder supports various kinds of authentication.

```
IR1101 (config-if-lorawan-cpf)#auth-mode ?
        client-server  client and server authentication mode
        server         authentication mode
        sudi           cisco sudi authentication mode
```

h. **Choice of security with Actility deployment**

Public key and then IPsec tunnel with strong swan server.

i. **Choice of Sensor support class A/B/C**

Currently we support Class A and Class C sensors only. Class B will be supported in the next release on CPF.

j. **Choice of Network server: chirp stack vs Thingpark.**

Open-source network server vs engineered Thingpark solution is a customer choice.

Bringing up Actility Packet forwarder on Routers

Bringing up Cellular WAN interface , please refer to the following guide.

Cisco 4G LTE-Advanced Configuration

Important Notes:

- Actility packet forwarder need some prerequisite setting for the app to ssh to the Router CLI prompt and provision the authentication keys between iox app and the host CLI.
- Use the following on Router config cli /HOST:
  - add a new username:password , which is specific to Actility only.
    - config terminal
    - username actility privilege 15 password 0 actilityPassword
    - exit
  - add/run docker container with the following options (here you can see the default ip addr, user and password but you'll need to change them according to your config):
    - --device /dev/ttyACM0:/dev/ttyACM0
    - --env HOST_IP_ADDR=192.168.42.11
    - --env HOST_USER=actility
    - --env HOST_SETUP_PASSWORD=actilityPassword
  - Once the app is activated and working You'll see that the provisioned password is deleted and new one is created by app it self .Notice that we do not have a password for the actility user anymore (username actility privilege 15).
  - If you want to reinstall the lrr packet forwarder app again in future, you'll have to set again username actility privilege 15 password 0 actilityPassword
  - An APP upgrade won’t erase these credentials.
  - Latest packages include other docker run time options to support the LRR.ini file configuration for TPE – OCP deployments

To configure application hosting, enable IOx and configure a VirtualPortGroup to a Layer 3 data port as described in the following sections.

Step 1: Enable IOx

Perform the following steps to enable access to Cisco IOx Local Manager. IOx Local Manager provides a web-based user interface that you can use to manage, administer, monitor, and troubleshoot apps on the host system, and to perform a variety of related activities

```
1. Enter the following command to enable privileged EXEC mode:
   Device> enable
2. Enter this command to enter global configuration mode:
   Device# configure terminal
3. Enter this command to enable Cisco IOx:
   Device(config)# iox
4. Enter this command to enable the HTTP server on your IPv4 or IPv6
   system:
   Device(config)# ip http server
5. Enter this command to enable a secure HTTP (HTTPS) server:
   Device(config)# ip http secure-server
6. Use the following command to establish a username-based authentication
   system and privilege level. The username privilege level must be
   configured as 15.
   Command format:
   username name privilege level password {0 | 7 | user-
   password } encrypted-password
   Command example:
   Device(config)# username cisco privilege 15 password 0 cisco
7. Enter this command to exit the interface configuration mode and return
   to the privileged EXEC mode:
   Device(config-if)# end
```

Step 2 : Configure a VirtualPortGroup to a Layer 3 Data Port

Multiple Layer 3 data ports can be routed to one or more VirtualPortGroups or containers. A VirtualPortGroup interface is a virtual interface that connects the application hosting network to the IOS routing domain.VirtualPortGroups and Layer 3 data ports must be on different subnets.

To configure a VirtualPortGroup to a Layer 3 data port, follow these steps:

1. Enter the following command to enable privileged EXEC mode. Enter your password if prompted.
   Device> **enable**
2. Enter the following command to enter global configuration mode:
   Device# **configure terminal**
3. Enter the following command to enable IP routing. The ip routing command must be enabled to allow external routing on Layer 3 data ports.
   Device(config)# **ip routing**
4. Use the following command to configure an interface and enter interface configuration mode.
   Command format: interface *type number*
   Command example: Device(config)# **interface gigabitethernet 0/0/0**
5. Enter the following command to place the interface in Layer 3 mode and make it operate more like a router interface than a switch port:
   Device(config-if)# **no switchport**
6. Use the following command to configure an IP address for the interface.
   Command format:ip address *ip-address mask*
   Command example: Device(config)# **ip address dhcp**
7. Enter the following command to exit interface configuration mode and return to global configuration mode:
   Device(config-if)# **exit**
8. Use the following command to configure an interface and enter interface configuration mode.
   Command format:interface *type number*
   Command example: Device(config)# **interface virtualportgroup 0**
9. Use the following command to configure an IP address for the interface.
   Command format:ip address *ip-address mask*
   Command example:Device(config-if)# **ip address 192.168.2.1 255.255.255.0**
10. Enter the following command to exit interface configuration mode and return to privileged EXEC mode:
    Device(config-if)# **end**

Step 3 : Configure Application Networking

Application vNIC interface is the standard Ethernet interface inside the container that connects to the platform data plane for application to send and receive packets.

1. Use the following command to enter global configuration mode, and then enter configuration commands, one per line. Press CTRL-Z when you are finished entering configuration commands.
   Device# configure terminal

2. Use the following command to configure the application and enter the application configuration mode.
   Command format:

Step 4:

```
app-hosting appid app1
Command example:
Device(config)# app-hosting appid app1
        1. Use the app-vnic command to configure the
           application interface and the gateway of the
           application. For example:
           Device(config-app-hosting)# app-vnic gateway0
           virtualportgroup 0 guest-interface 0
        2. Use the guest-ipaddress command to configure the
           application Ethernet interface IP address. For
           example:
           Device(config-app-hosting-gateway0)# guest-ipaddress
           192.168.2.9 netmask 255.255.255.0
        3. Use the app-default-gateway command to configure the
           default gateway for the application. For example:
           Device(config-app-hosting-gateway0)# app-default-
           gateway 192.168.2.1 guest-interface 0
        4. Enter the following command to exit global
           configuration mode and return to privileged EXEC
           configuration mode:
           Device# end
```

Application Lifecycle Management

This section describes how to install and uninstall apps.

Use the following command to enable privileged EXEC mode:
Device> **enable**

Use the following command to install an app from the specified location. The app can be installed from any local storage location such as, flash, bootflash, and usbflasho.

```
Command format:app-hosting install appid application-
name package package-path
Command example:
Device(config)# app-hosting install appid APFAPP package
flash:actility_tar_gz.tar
```

Use the following command to activate the application. This command validates all application resource requests, and if all resources are available, activates the application. If all resources are not available, the activation fails.

Command format:app-hosting activate appid *application-name*
Command example:Device# **app-hosting activate appid APFAPP**

Use the following command to start the application. This command activates the application start-up scripts.

Command format:app-hosting start appid *application-name*
Command example: Device# **app-hosting start appid APFAPP**

Use the following command to stop the application.

```
Command format:app-hosting stop appid application-name
Command example:Device# app-hosting stop appid APFAPP
```

Use the following command to deactivate all resources that are allocated for the application.

```
Command format:app-hosting deactivate appid application-name
Command example: Device# app-hosting deactivate appid APFAPP
```

Use the following command to uninstall the application. This command uninstalls all packaging and images that are stored and removes all changes and updates to the application.

Command format: app-hosting uninstall appid *application-name*
Command example: Device# **app-hosting uninstall appid APFAPP**

Step 5: Verifying the Application Hosting Configuration

This section describes how to verify the application hosting configuration.

```
1. Use the following command to enable privileged EXEC mode:
   Device> enable
2. Use the following command to display the status of all IOx services:
   Device# show iox-service
   Router#show iox-service

   IOx Infrastructure Summary
   ---------------------------
   IOx service (CAF)               Running
   IOx service (HA)                Not Supported
   IOx service (IOxman)            Running
   IOx service (Sec storage)       Running
   Libvirtd 5.5.0                  Running
   Dockerd v19.03.13-ce            Running

3. Use the following command to display detailed information about the
   application:
   Device# show app-hosting detail
4. Use the following command to display the list of applications and
   their statuses:
   Device# show app-hosting list
   App id State
   ---------------------------------------------
   APFAPP RUNNING
5. Use the Console command to connect to the application, as shown in the
   following example:
   iox-ir1101-02# app-hosting app-hosting connect appid APFAPP session

   /home/actility/var/log/lrr
   /var/volatile/log/_LRRLOG # pwd
   /home/actility/var/log/lrr
   /var/volatile/log/_LRRLOG # ls -lrt
   -rw-r--r--    1 root     root           19 Jul  7 0646 SHELL.log
   -rw-r--r--    1 root     support        53 Jul  7 0647 suplog.log
   -rw-r--r--    1 root     support        99 Jul  7 0648 pkiconfig.txt
   -rw-r--r--    1 root     root          430 Jul  7 0720 lrr_startup_service.log
   -rw-r--r--    2 root     root         1620 Jul  7 0721 gwmgr_04.log
   -rw-r--r--    2 root     root         1620 Jul  7 0721 gwmgr.log
   -rw-r--r--    1 root     root         1657 Jul  7 0721 radioparams.txt
   -rw-r--r--    1 root     root         2227 Jul  7 0721 logicchan.txt
   -rw-r--r--    1 root     root         1118 Jul  7 1721 stat.html
   -rw-r--r--    2 root     root        50515 Jul  7 1721 TRACE_04.log
   -rw-r--r--    2 root     root        50515 Jul  7 1721 TRACE.log
   -rw-r--r--    1 root     root           64 Jul  7 1723 lrcstatuslink.txt
   /var/volatile/log/_LRRLOG #
```

Sample Running Config

```
Router#show running-config brief
Building configuration...

Current configuration  7651 bytes
!
! Last configuration change at 072004 UTC Thu Jul 7 2022 by actility
! NVRAM config last updated at 065725 UTC Thu Jul 7 2022 by actility
!
version 17.9
service timestamps debug datetime msec
service timestamps log datetime msec
service call-home
platform qfp utilization monitor load 80
platform hardware throughput level 250M
platform punt-keepalive disable-kernel-core
!
hostname Router
!
boot-start-marker
boot system flashir1101-universalk9.S2C.SSA.bin
boot-end-marker
!
!
aaa new-model
!
!
aaa authentication login default local
aaa authorization exec default local
aaa authorization network FlexVPN_Author local
!
!
aaa session-id common
!
!
!
!
!
!
!
!
!
!
login block-for 60 attempts 3 within 30
login delay 3
login on-success log
ipv6 unicast-routing
!
        !
!
!
!
!
!
subscriber templating
!
!
!
!
!
!
multilink bundle-name authenticated
!
!
!
```

```
!
!
!
!
!
crypto pki trustpoint TP-self-signed-1150468717
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-1150468717
 revocation-check none
 rsakeypair TP-self-signed-1150468717
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
!
crypto pki trustpoint ActilityTP-slrc
 enrollment terminal
 revocation-check none
!
crypto pki trustpoint ActilityTP
 enrollment pkcs12
 revocation-check crl
 rsakeypair ActilityTP
!
crypto pki trustpoint ActilityTP-rrr1
 revocation-check crl
!
!
!
crypto pki certificate map FlexVPN_Cert_Map 1
 subject-name co slrc1_prod-us_actility-tpe-ope
!
crypto pki certificate map FlexVPN_Cert_Map 2
 subject-name co slrc2_prod-us_actility-tpe-ope
!
crypto pki certificate chain TP-self-signed-1150468717
 certificate self-signed 01
crypto pki certificate chain SLA-TrustPoint
 certificate ca 01
crypto pki certificate chain ActilityTP-slrc
 certificate ca 61A845069BBFF60B
crypto pki certificate chain ActilityTP
 certificate 06BF5FDCF5EBD17C
 certificate ca 3A96CABF858AAD9A
crypto pki certificate chain ActilityTP-rrr1
 certificate ca 00F35AC229699BABA8
!
!
!
!
!
!
!
!
!
no license feature hseck9
license udi pid IR1101-K9 sn FCW24160HQ7
license boot level network-advantage
memory free low-watermark processor 45069
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
!
!
username admin privilege 15 password 0 cisco
username iox privilege 15 password 0 iox
username dockeruser
```

```
username actility privilege 15
!
redundancy
!
crypto ikev2 authorization policy FlexVPN_Author_Policy
!
!
!
!
crypto ikev2 profile FlexVPN_IKEv2_Profile
 match certificate FlexVPN_Cert_Map
 identity local dn
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint ActilityTP sign
 pki trustpoint ActilityTP-rrr1 verify
 pki trustpoint ActilityTP-slrc verify
 dpd 30 3 periodic
 aaa authorization group cert list FlexVPN_Author FlexVPN_Author_Policy
!
crypto ikev2 dpd 30 3 periodic
crypto ikev2 fragmentation mtu 1260
!
controller Cellular 0/3/0
!
!
vlan internal allocation policy ascending
!
!
!
!
!
!
!
!
!
!
crypto ipsec transform-set FlexVPN_IPsec_Transform_Set esp-aes 256 esp-sha256-hmac
 mode tunnel
!
crypto ipsec profile FlexVPN_IPsec_Profile
 set transform-set FlexVPN_IPsec_Transform_Set
 set ikev2-profile FlexVPN_IKEv2_Profile
!
!
!
!
!
!
!
!
!
!
interface Tunnel201
 ip address negotiated
 ip nat outside
 ipv6 enable
 tunnel source GigabitEthernet0/0/0
 tunnel mode ipsec dual-overlay
 tunnel destination 52.200.161.236
 tunnel path-mtu-discovery
 tunnel protection ipsec profile FlexVPN_IPsec_Profile
!
interface Tunnel202
 ip address negotiated
 ip nat outside
 ipv6 enable
 tunnel source GigabitEthernet0/0/0
```

```
 tunnel mode ipsec dual-overlay
 tunnel destination 54.226.90.83
 tunnel path-mtu-discovery
 tunnel protection ipsec profile FlexVPN_IPsec_Profile
!
interface VirtualPortGroup0
 ip address 192.168.2.1 255.255.255.0
 ip nat inside
 no mop enabled
 no mop sysid
!
interface GigabitEthernet0/0/0
 ip dhcp client client-id ascii cisco-ac4a.67f9.ae00-Gi0/0/0
 ip address dhcp
 ip nat outside
 ipv6 dhcp client request vendor
 ipv6 address dhcp
 ipv6 address autoconfig
 ipv6 enable
!
interface FastEthernet0/0/1
!
interface FastEthernet0/0/2
!
interface FastEthernet0/0/3
!
interface FastEthernet0/0/4
!
interface GigabitEthernet0/0/5
!
interface Cellular0/3/0
 description backup_WAN
 ip address negotiated
 ip nat outside
 ip tcp adjust-mss 1460
 load-interval 30
 shutdown
 dialer in-band
 dialer idle-timeout 0
 dialer-group 1
 ipv6 enable
 pulse-time 1
!
interface Cellular0/3/1
 no ip address
!
interface Vlan1
 no ip address
!
interface Async0/2/0
 no ip address
 encapsulation scada
!
interface LORAWAN0/1/0
 no ip address
 shutdown
 arp timeout 0
 no mop enabled
 no mop sysid
!
iox
ip forward-protocol nd
ip tcp selective-ack
ip tcp mss 1460
ip tcp window-size 131072
ip http server
ip http auth-retry 3 time-window 1
ip http authentication local
ip http secure-server
```

```
ip http client source-interface GigabitEthernet0/0/0
ip tftp source-interface GigabitEthernet0/0/0
ip nat inside source list Tunnel201 interface Tunnel201 overload
ip nat inside source list Tunnel202 interface Tunnel202 overload
ip nat inside source list internetacces_Fromdocker interface GigabitEthernet0/0/0 overload
ip nat inside source list internetacces_Fromdocker_cell interface Cellular0/3/0 overload
ip route 10.102.12.0 255.255.255.0 Tunnel201
ip route 10.102.22.0 255.255.255.0 Tunnel202
ip ssh bulk-mode 131072
ip ssh version 2
ip ssh pubkey-chain
  username actility
   key-hash ecdsa-sha2-nistp256 FA249B09C77A121A9759A0FC724F58A8 root@a89e080e0c1e
ip ssh server algorithm publickey ecdsa-sha2-nistp256
ip scp server enable
!
!
ip access-list extended Tunnel201
 10 permit ip host 192.168.2.9 host 10.102.12.10
ip access-list extended Tunnel202
 10 permit ip host 192.168.2.9 host 10.102.22.10
ip access-list extended internetacces_Fromdocker
 10 permit ip 192.168.2.0 0.0.0.255 host 8.8.8.8
 11 permit ip 192.168.2.0 0.0.0.255 host 52.200.161.236
ip access-list extended internetacces_Fromdocker_cell
 10 permit ip host 192.168.2.9 host 8.8.8.8
!
ip sla 1
 icmp-echo 8.8.8.8 source-interface GigabitEthernet0/0/0
ip sla schedule 1 life forever start-time now
ip sla 2
 icmp-echo 8.8.8.8 source-interface Cellular0/3/0
ip sla schedule 2 life forever start-time now
ip access-list standard 1
 11 permit any
dialer-list 1 protocol ip permit
!
!
!
!
!
!
control-plane
!
!
!
!
!
!
!
!
line con 0
 stopbits 1
line 0/0/0
line 0/2/0
line vty 0 4
 transport input ssh
line vty 5 14
 transport input ssh
!
call-home
 ! If contact email address in call-home is configured as sch-smart-licensing@cisco.com
 ! the email address configured in Cisco Smart License Portal will be used as contact email address to
send SCH notifications.
 contact-email-addr sch-smart-licensing@cisco.com
 profile "CiscoTAC-1"
  active
  destination transport-method http
ntp server 0.pool.ntp.org
```

```
ntp server 1.pool.ntp.org
ntp server 2.pool.ntp.org
!
!
!
!
!
!
!
!
!
!
!
!
event manager applet restart_actility_lrr
 event none sync yes maxrun 60
 action 1 cli command "enable"
 action 2 cli command "app-hosting stop appid APFC1"
 action 3 wait 5
 action 4 cli command "app-hosting start appid APFC1"
event manager applet Cellular_Activate
 event track 1 state down
 action 1 cli command "enable"
 action 2 cli command "configure terminal"
 action 3 cli command "interface Cellular 0/3/0"
 action 4 cli command "no shut"
 action 5 cli command "end"
event manager applet Cellular_Deactivate
 event track 1 state up
 action 1 cli command "enable"
 action 2 cli command "config terminal"
 action 3 cli command "interface Cellular 0/3/0"
 action 4 cli command "shutdown"
 action 5 cli command "end"
!
end

Router#$
```

Actility -Thing Park Network Server Options

a. About Actility Thingpark -Enterprise On Premise Platform / TPE – OCP

This deployment is an enterprise based on prem deployment and it considered as most secured and the **Thingpark server is deployed in the local site enterprise network** and is reachable via IR1101 or IR1800 GE 0/0/0 port.

Once the app is installed the LRR packet forwarder will contact the TPE-OCP via WAN port and forwards the sensor traffic to TPE dashboard.

b. About Actility Thingpark Community / TPE – Community

A self-service Device and Application makers portal to build their solutions with Thingpark.
Please refer to following URL https://community.thingpark.org/iot-solutions-catalog/ for more details and login , the navigation should be same as that of TPE – OCP.

c. About Actility Thingpark SaaS / TPE – SaaS

The Cloud deployment of Thingpark dashboard for enterprises as SaaS. Based on the location of the customer and the geography we use various cloud url for connecting to TPE-SaaS. Please make a note that this deployment is not secure, and you need tunnels / certs / and have a make a choice of WAN if your device is remotely deployed, unlike TPE-OCP (on prem).

- https://thingparkenterprise.us.actility.com/tpe/#/login
- https://thingparkenterprise.eu.actility.com/tpe/#/login
- https://thingparkenterprise.au.actility.com/tpe/#/login

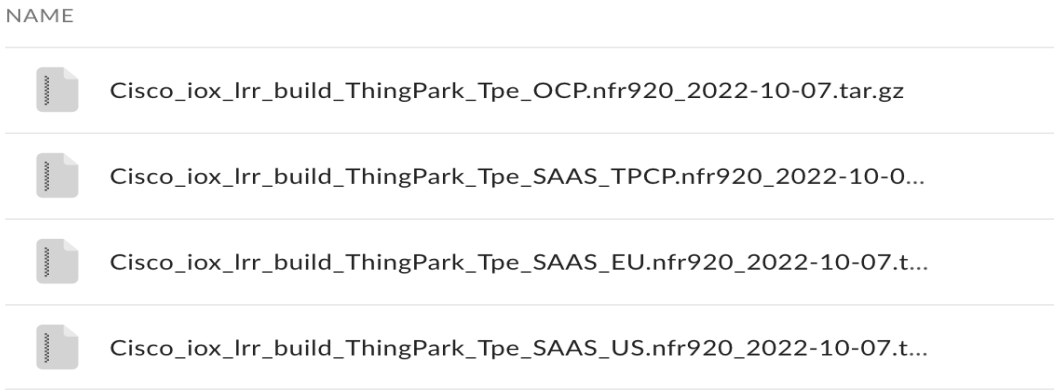
Actility – Packet forwarder as Docker App and naming conventions

Actility packet forwarders are docker apps that are sharing with various labels. We use the following label conventions to describe the app type accordingly.

a. Label **TPE_OCP** – *designated for enterprise reach.*
b. Label **TPCP** – *designated community portal reach.*
c. Label **TPE_SaaS EU** -*designated for Europe server reach*
d. Label **TPE_SaaS US** -*designated for US server reach*

Note : Please refer to the sample screen shot on how the packages are designated with label. You must make a right choice of selection as per your choice if Thingpark deployment.

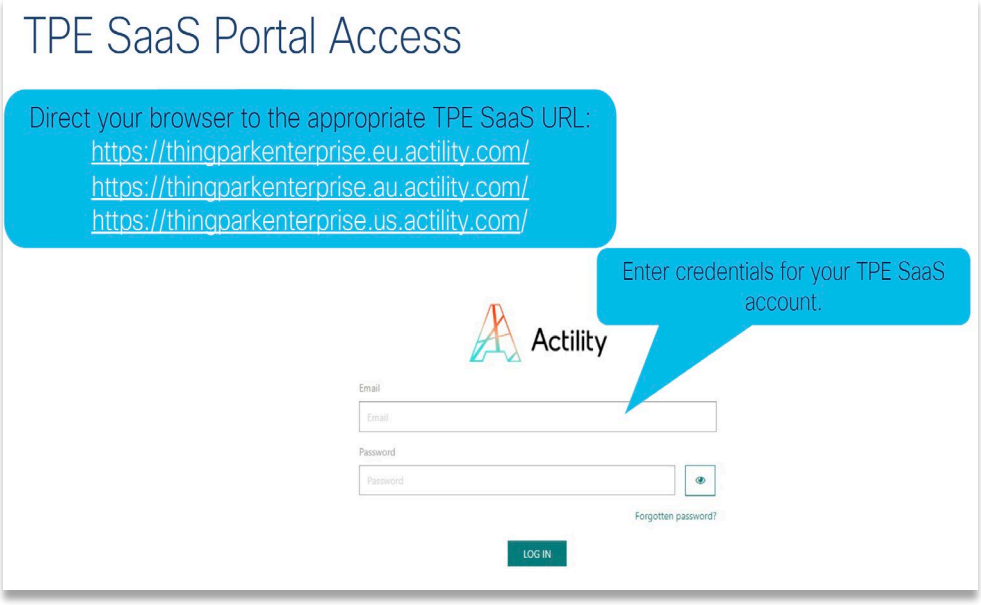
Sample Screen shot:



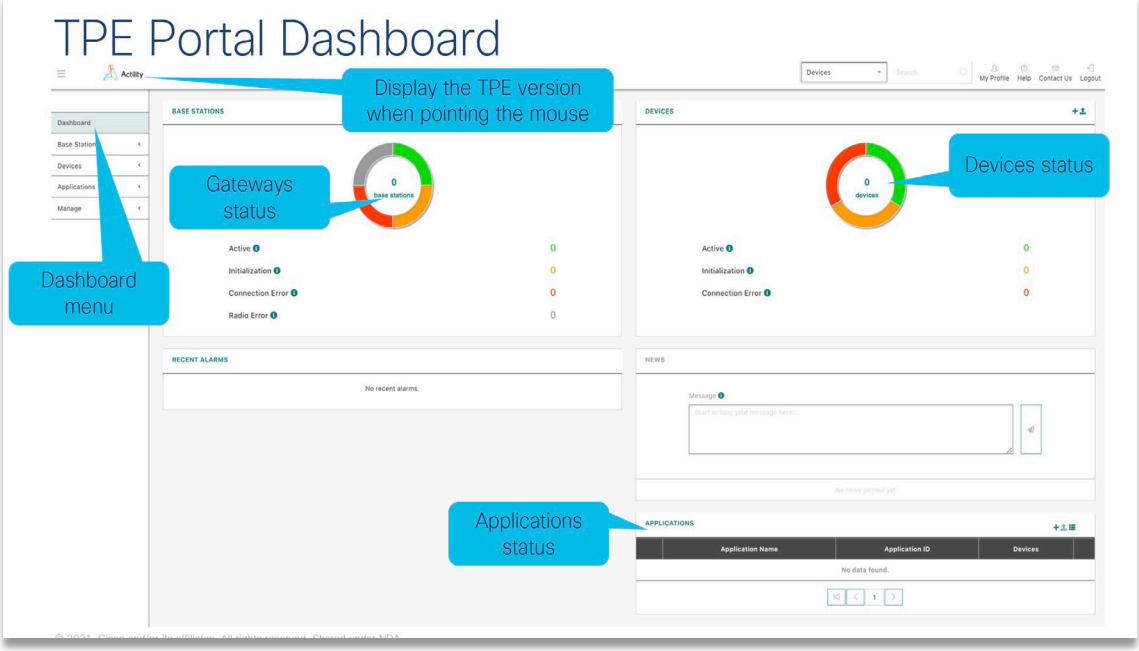
Actility -TPE – OCP step by steps

a. TPE OCP/CP/SaaS login

Please use the following url for TPE SaaS login, and make sure either you have required credentials to login or register if you need one. Your IT admin should be a provide you required credentials for you to login.



b. Understanding Dashboard Page



c. Adding a Base station

Get the LRR UID from Actility support tool

Once the app is installed and running fine, we have a way to console the app session using app-hosting connect app-id APFx session and use su support account to get to the actility support tool.

```
Router#app-hosting connect appid APFC1 session


/ #
/ # su support
```

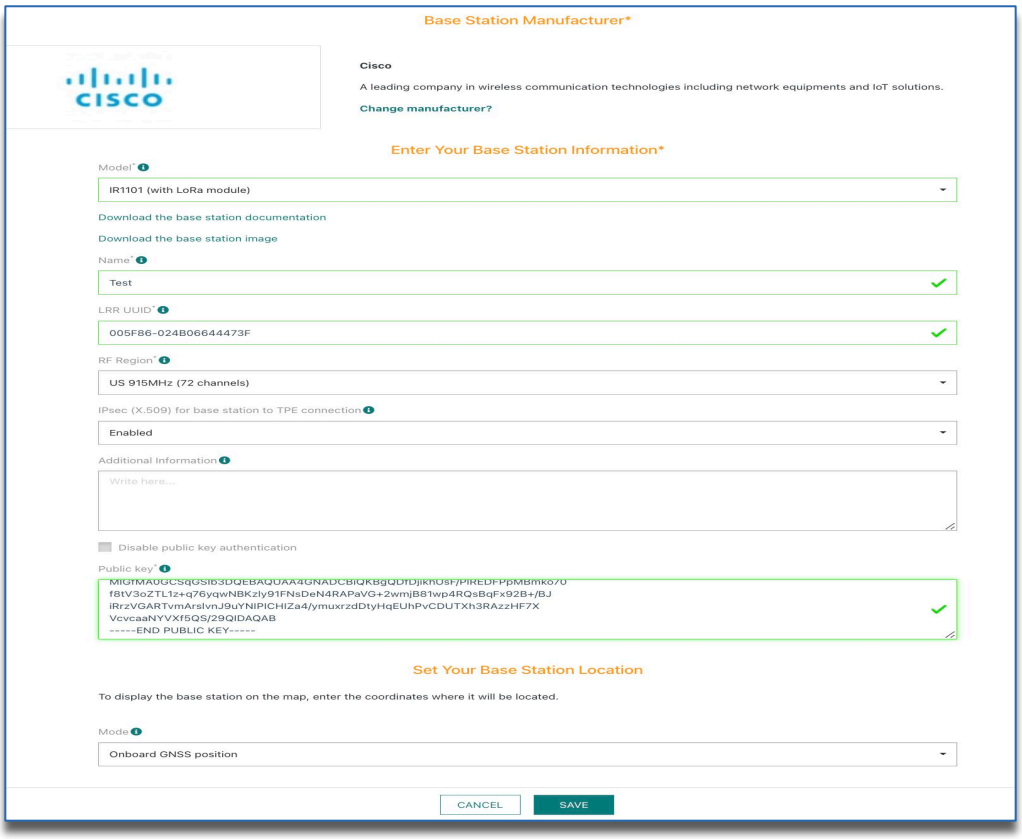
Get the LRR ID and public keys required.

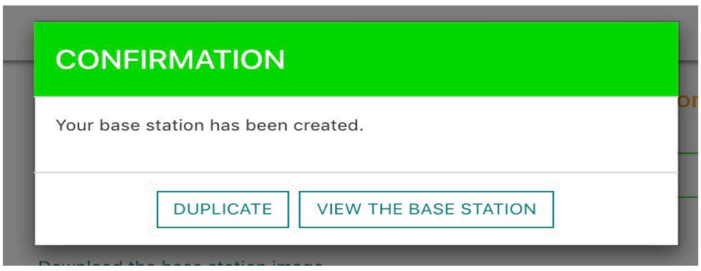
The support tool is a command utility tool. Please use tab and escape up down arrows on the keyboard to control and highlight the button and press enter to get the next prompt.

Add GW to the TPE: We return to the TPE screen that we left a few steps ago.

- Complete the fields appropriately
- Use the LRRUUID that you captured in a previous step
- Make sure" IPsec for base station to TPE connection" is enabled

- Make sure “Disable public key authentication” is unchecked, and paste public key gathered in previous step
- Press “Save” and check for confirmation

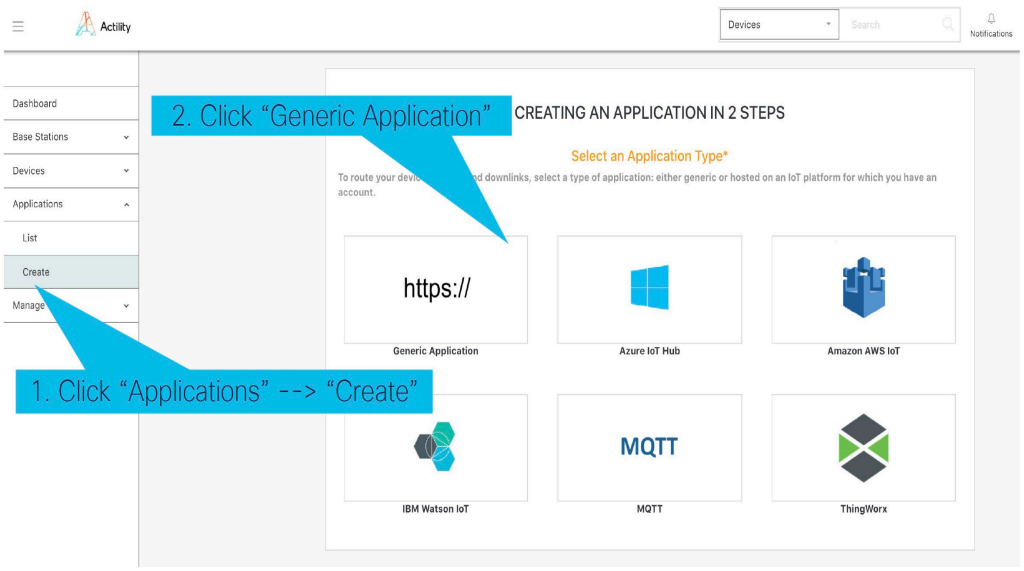




d. Add TPE application

- Before adding a sensor device, we must first define application(s) to which the sensor data will flow.
- POSThere.io is a useful tool for viewing and debugging HTTP POSTs.
- As an example, we will set-up TPE to send sensor data to POSThere.io.
- Please navigate to https://posthere.io

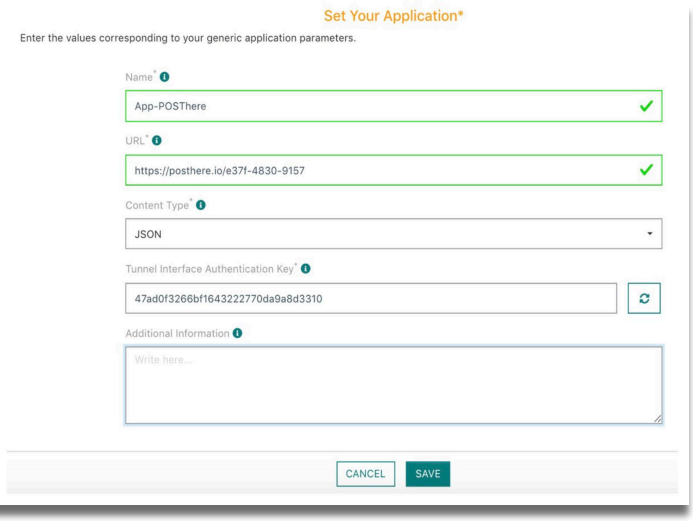
1. Copy this link into your clipboard, then click the link

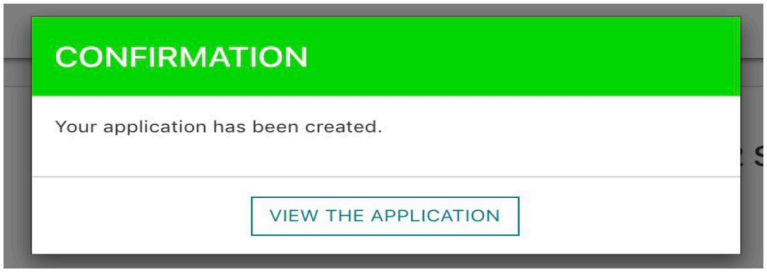


2. Click "Generic Application"

1. Click "Applications" --> "Create"

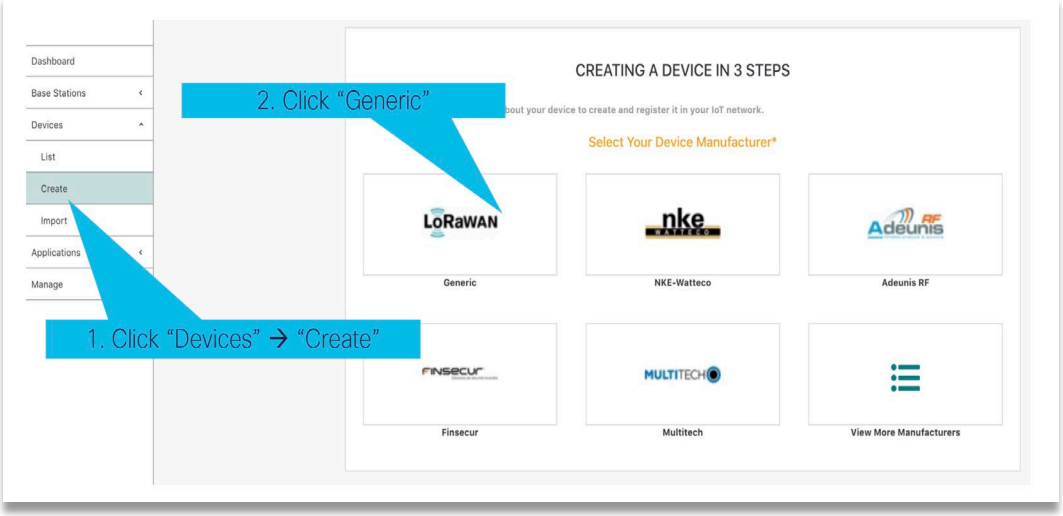
- Populate the fields as shown to the right. Note, your POSThere URL will be unique (from the previous step).
- Please leave the tunnel interface key, as is; it isn't used when interfacing to POSThere.
- Click "save"

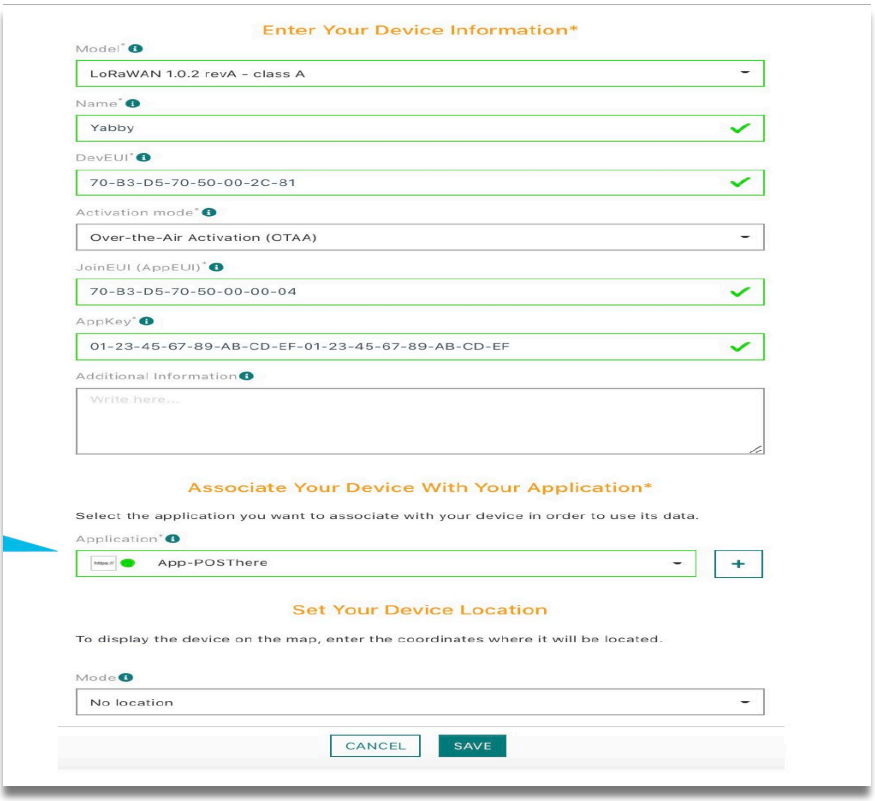


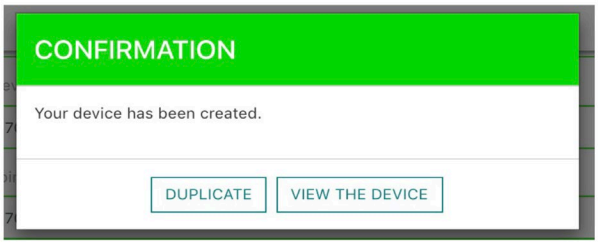


e. Adding a sensor

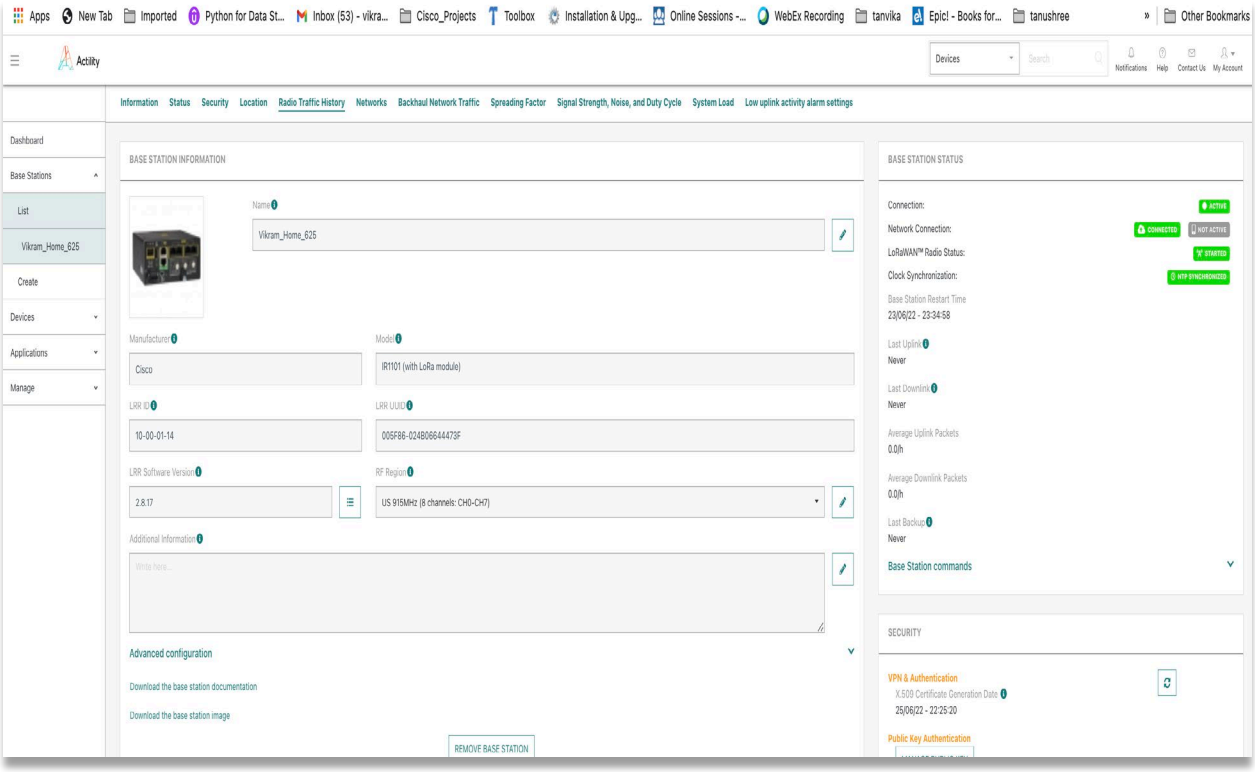
Please make sure you have the Sensor related info as DEVEUI, APPKEY, JoinEUI etc.


Dashboard
Base Stations
Devices
List
Create
Import
Applications
Manage
2. Click “Generic”
1. Click “Devices” → “Create”
CREATING A DEVICE IN 3 STEPS
Select Your Device Manufacturer*
Generic
NKE-Watteco
Adeunis RF
Finsecur
Multitech
View More Manufacturers


Enter Your Device Information*
Model*
LoRaWAN 1.0.2 revA - class A
Name*
Yabby
DevEUI*
70-B3-D5-70-50-00-2C-81
Activation mode*
Over-the-Air Activation (OTAA)
JoinEUI (AppEUI)*
70-B3-D5-70-50-00-00-04
AppKey*
01-23-45-67-89-AB-CD-EF-01-23-45-67-89-AB-CD-EF
Additional Information
Write here...
Associate Your Device With Your Application*
Select the application you want to associate with your device in order to use its data.
Application*
App-POSThere
Set Your Device Location
To display the device on the map, enter the coordinates where it will be located.
Mode
No location
CANCEL
SAVE

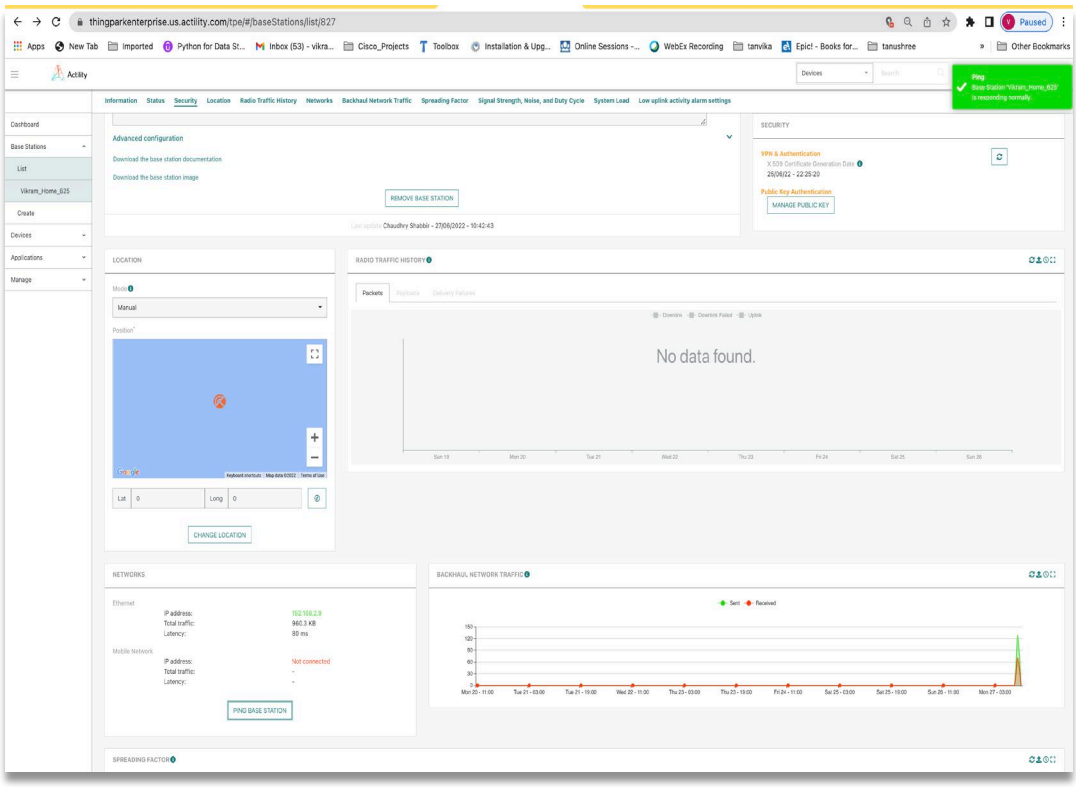


f. Active status



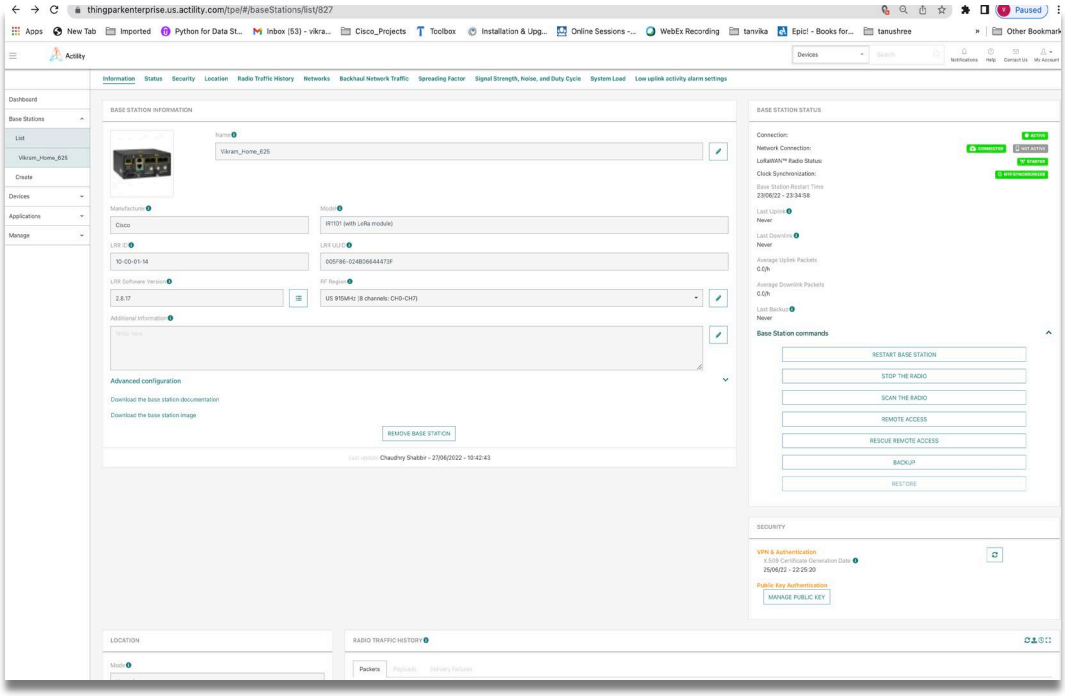
a. Ping check

Performing a ping check will show that the ping is successful to the GW.



i. Base Station command view

Once the device is up, you should be able to see the device connection status showing as green on the right side.



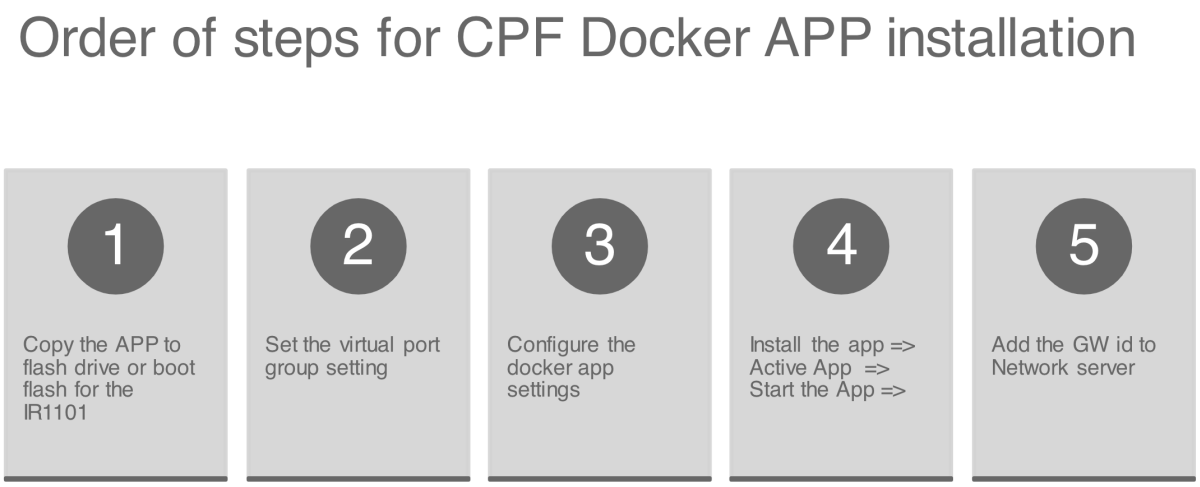
Bringing up Common packet forwarder on Routers:

Perquisites:

- You have a LoRa Network installed – open-source network server
- You have the signed CPF app downloaded from cisco site
- You have the required IR1101 or IR1800 and PIM-LPWA-800|900 module
- Validate license – network advantage license is needed
- GPS connectivity is mandatory for this deployment

Steps to bring up CPF app on IR1101

Before you begin with Common Packet Forwarder:



To configure application hosting, enable IOx and configure a VirtualPortGroup to a Layer 3 data port as described in the following sections.

Step 1: Enable IOx

Perform the following steps to enable access to Cisco IOx Local Manager. IOx Local Manager provides a web-based user interface that you can use to manage, administer, monitor, and troubleshoot apps on the host system, and to perform a variety of related activities

```
8.  Enter the following command to enable privileged EXEC mode:
    Device> enable
9.  Enter this command to enter global configuration mode:
    Device# configure terminal
10. Enter this command to enable Cisco IOx:
    Device(config)# iox
11. Enter this command to enable the HTTP server on your IPv4 or IPv6
    system:
    Device(config)# ip http server
12. Enter this command to enable a secure HTTP (HTTPS) server:
    Device(config)# ip http secure-server
13. Use the following command to establish a username-based authentication
    system and privilege level. The username privilege level must be
    configured as 15.
    Command format:
    username name privilege level password {0 | 7 | user-
    password } encrypted-password
    Command example:
    Device(config)# username cisco privilege 15 password 0 cisco
14. Enter this command to exit the interface configuration mode and return
    to the privileged EXEC mode:
    Device(config-if)# end
```

Step 2 : Configure a VirtualPortGroup to a Layer 3 Data Port

Multiple Layer 3 data ports can be routed to one or more VirtualPortGroups or containers. A VirtualPortGroup interface is a virtual interface that connects the application hosting network to the IOS routing domain.VirtualPortGroups and Layer 3 data ports must be on different subnets.

To configure a VirtualPortGroup to a Layer 3 data port, follow these steps:

11. Enter the following command to enable privileged EXEC mode. Enter your password if prompted.
    Device> **enable**
12. Enter the following command to enter global configuration mode:
    Device# **configure terminal**
13. Enter the following command to enable IP routing. The ip routing command must be enabled to allow external routing on Layer 3 data ports.
    Device(config)# **ip routing**
14. Use the following command to configure an interface and enter interface configuration mode.
    Command format: interface *type number*
    Command example: Device(config)# **interface gigabitethernet 0/0/0**
15. Enter the following command to place the interface in Layer 3 mode and make it operate more like a router interface than a switch port:
    Device(config-if)# **no switchport**
16. Use the following command to configure an IP address for the interface.
    Command format:ip address *ip-address mask*
    Command example: Device(config)# **ip address dhcp**
17. Enter the following command to exit interface configuration mode and return to global configuration mode:
    Device(config-if)# **exit**
18. Use the following command to configure an interface and enter interface configuration mode.
    Command format:interface *type number*
    Command example: Device(config)# **interface virtualportgroup 0**
19. Use the following command to configure an IP address for the interface.
    Command format:ip address *ip-address mask*
    Command example:Device(config-if)# **ip address 192.168.2.1 255.255.255.0**
20. Enter the following command to exit interface configuration mode and return to privileged EXEC mode:
    Device(config-if)# **end**

Step 3 : Configure Application Networking

Application vNIC interface is the standard Ethernet interface inside the container that connects to the platform data plane for application to send and receive packets.

1. Use the following command to enter global configuration mode, and then enter configuration commands, one per line. Press CTRL-Z when you are finished entering configuration commands.
   Device# configure terminal

2. Use the following command to configure the application and enter the application configuration mode.
   Command format:

Step 4:

```
app-hosting appid app1
Command example:
Device(config)# app-hosting appid app1
        5. Use the app-vnic command to configure the
           application interface and the gateway of the
           application. For example:
           Device(config-app-hosting)# app-vnic gateway0
           virtualportgroup 0 guest-interface 0
        6. Use the guest-ipaddress command to configure the
           application Ethernet interface IP address. For
           example:
           Device(config-app-hosting-gateway0)# guest-ipaddress
           192.168.2.9 netmask 255.255.255.0
        7. Use the app-default-gateway command to configure the
           default gateway for the application. For example:
           Device(config-app-hosting-gateway0)# app-default-
           gateway 192.168.2.1 guest-interface 0
        8. Enter the following command to exit global
           configuration mode and return to privileged EXEC
           configuration mode:
           Device# end
```

Application Lifecycle Management

This section describes how to install and uninstall apps.

Use the following command to enable privileged EXEC mode:
Device> **enable**

Use the following command to install an app from the specified location. The app can be installed from any local storage location such as, flash, bootflash, and usbflasho.

```
Command format:app-hosting install appid application-
name package package-path
Command example:
Device(config)# app-hosting install appid cp package
flash:cpf.tar
```

Use the following command to activate the application. This command validates all application resource requests, and if all resources are available, activates the application. If all resources are not available, the activation fails.

Command format:app-hosting activate appid *application-name*
Command example:Device# **app-hosting activate appid cp**

Use the following command to start the application. This command activates the application start-up scripts.

Command format:app-hosting start appid *application-name*
Command example: Device# **app-hosting start appid cp**

Use the following command to stop the application.

Command format:app-hosting stop appid *application-name*
Command example:Device# **app-hosting stop appid cp**

Use the following command to deactivate all resources that are allocated for the application.

Command format:app-hosting deactivate appid *application-name*
Command example: Device# **app-hosting deactivate appid cp**

Use the following command to uninstall the application. This command uninstalls all packaging and images that are stored and removes all changes and updates to the application.

Command format: app-hosting uninstall appid *application-name*
Command example: Device# **app-hosting uninstall appid cp**

Step 5: Verifying the Application Hosting Configuration

This section describes how to verify the application hosting configuration.

1. Use the following command to display detailed information about the application:

```
Router#show app-hosting detail
App id                  : cp
Owner                   : iox
State                   : RUNNING
Application
  Type                  : docker
  Name                  : cpf
  Version               : v1
  Description           : buildkit.dockerfile.v0
  Author                :
  Path                  : bootflash:cpfv5.tar
  URL Path              :
  Multicast             : yes
Activated profile name : custom

Resource reservation
  Memory                : 128 MB
  Disk                  : 10 MB
  CPU                   : 400 units
  CPU-percent           : 35 %
  VCPU                  : 1

Platform resource profiles
  Profile Name                  CPU(unit)  Memory(MB)  Disk(MB)
  ---------------------------------------------------------------

Attached devices
  Type              Name               Alias
  ---------------------------------------------
  serial/shell     iox_console_shell   serial0
  serial/aux       iox_console_aux     serial1
  serial/syslog    iox_syslog          serial2
  serial/trace     iox_trace           serial3

Network interfaces
   ---------------------------------------
eth0:
   MAC address          : 52:54:dd:f2:f4:87
   IPv4 address         : 192.168.0.9
   IPv6 address         : ::
   Network name         : VPG0


Docker
------
Run-time information
  Command               :
  Entry-point           : /station/cpf
  Run options in use    : --device /dev/lorawan_tty1:/dev/ttyACM0 -v /bootflash/lorawan_0:/cpf/
  Package run options   :
Application health information
  Status                : 0
  Last probe error      :
  Last probe output     :
```

Use the following command to display the list of applications and their statuses:

Device# show app-hosting list
App id State

CP RUNNING

2. Sample app settings:

```
!
interface VirtualPortGroup0
 ip address 192.168.0.1 255.255.255.0
 ip nat inside
 no mop enabled
 no mop sysid
```

lorawan_tty1 as its plugged into slot 1

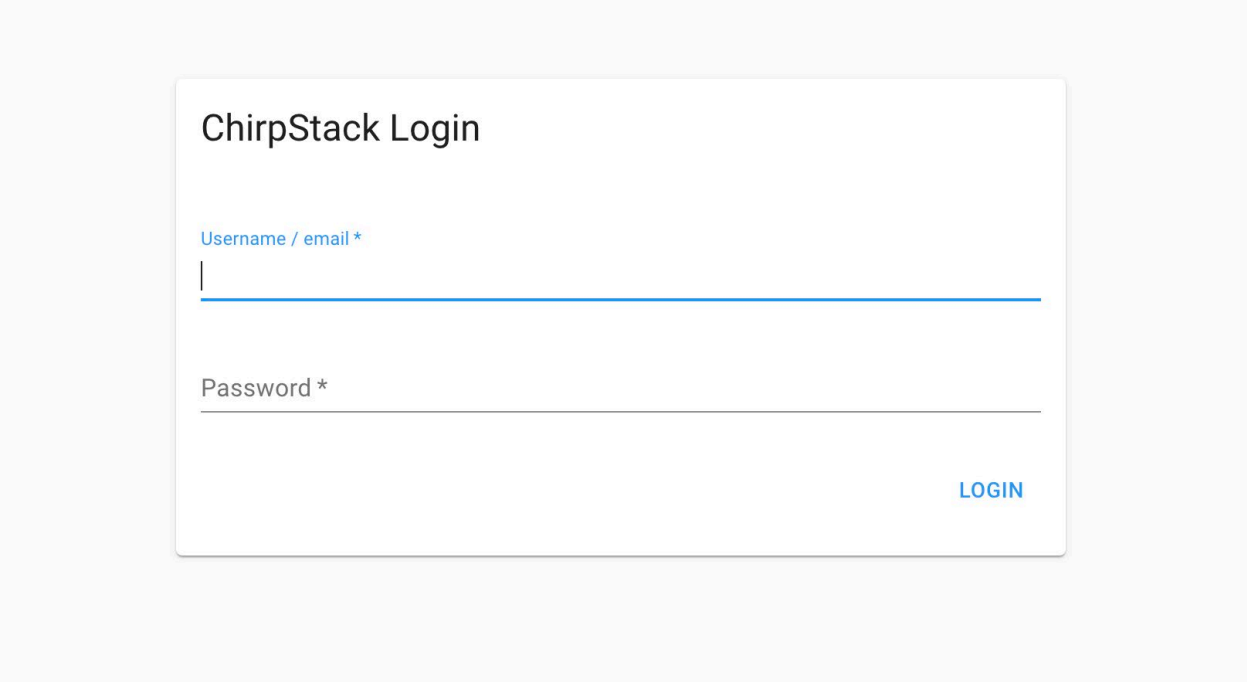
```
app-hosting appid cp
 app-vnic gateway0 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.0.9 netmask 255.255.255.0
 app-default-gateway 192.168.0.1 guest-interface 0
 app-resource docker
  run-opts 1 "--device /dev/lorawan_tty1:/dev/ttyACM0"
  run-opts 3 "-v /bootflash/lorawan_0:/cpf/"
Router#
```

```
Router#sh iox-service

IOx Infrastructure Summary:
---------------------------
IOx service (CAF)              : Running
IOx service (HA)               : Not Supported
IOx service (IOxman)           : Running
IOx service (Sec storage)      : Running
Libvirtd 5.5.0                 : Running
Dockerd v19.03.13-ce           : Running

Router#
Router#
Router#app-hosting install appid cp package bootflash:cpfv5.tar
Installing package 'bootflash:cpfv5.tar' for 'cp'. Use 'show app-hosting list' for progress.
```

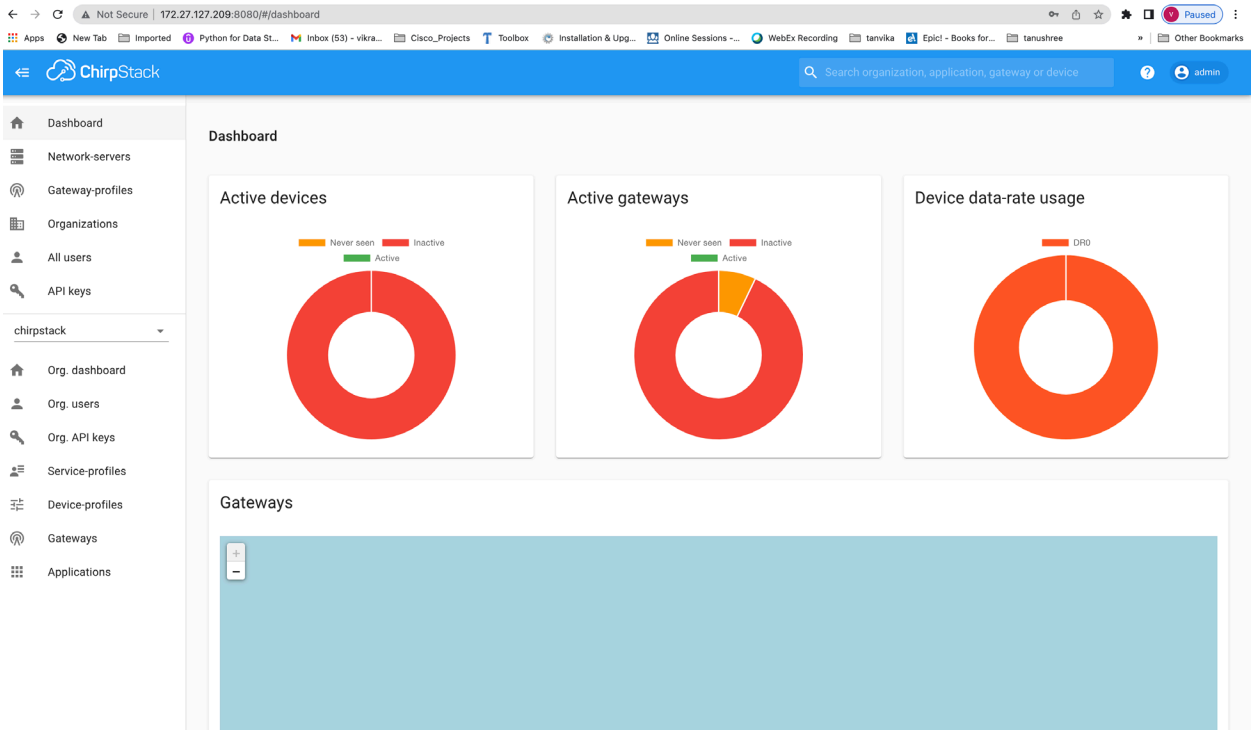
ChirpStack – Step by Steps:

Please refer the appendix for the ChirpStack installation:

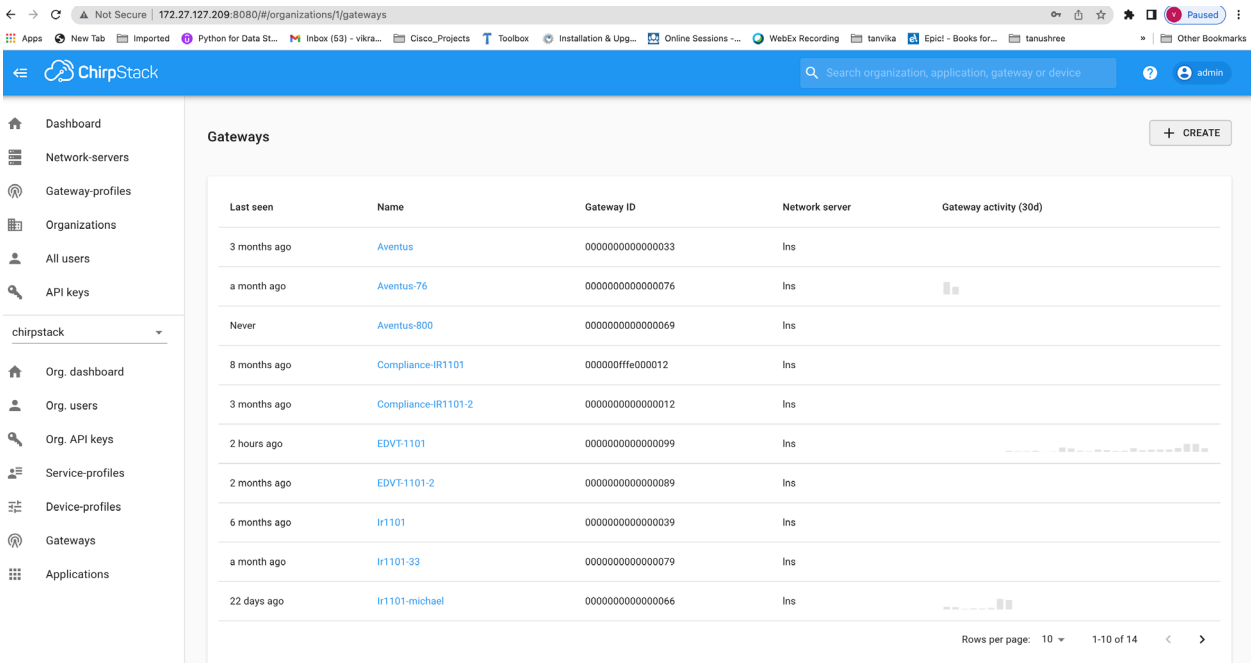
1.ChirpStack server login



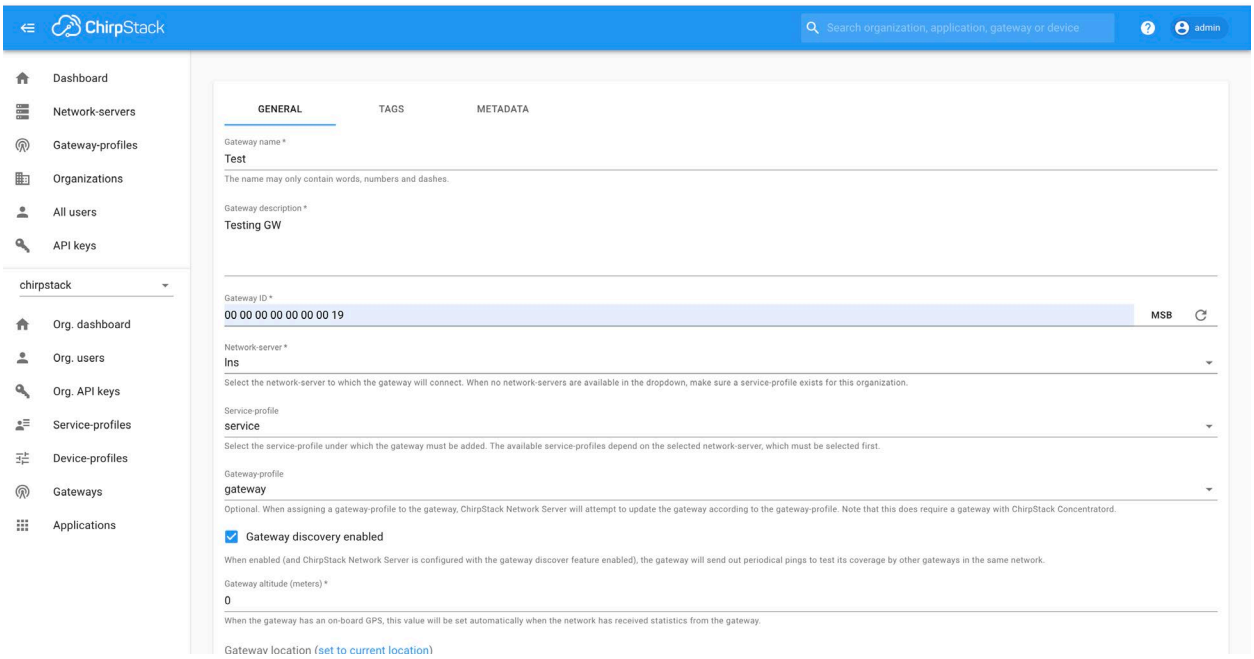
2.ChirpStack dashboard overview screen:



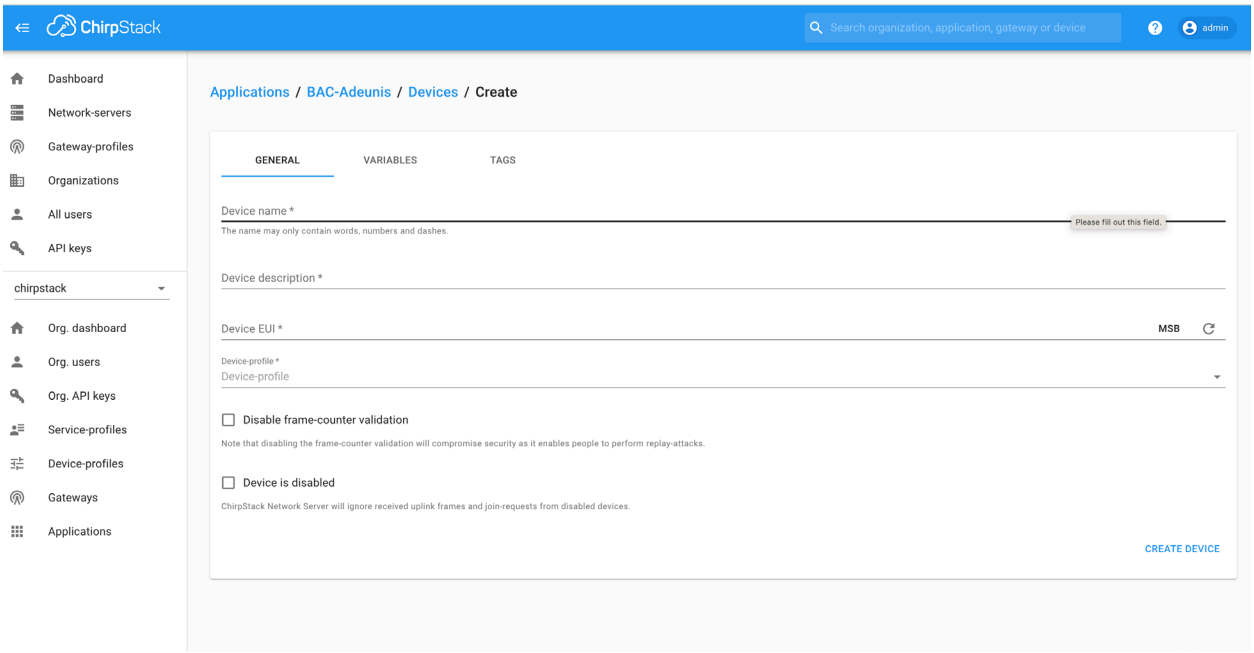
3. Create Gateway Step:



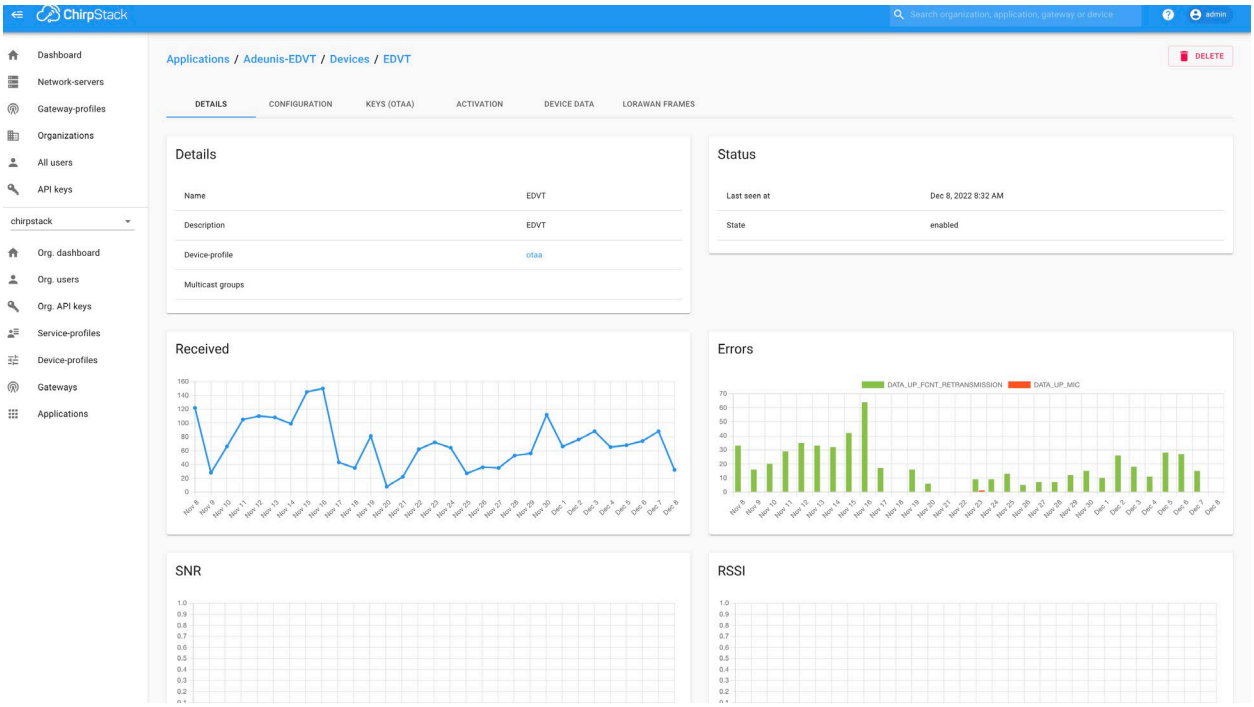
4. Adding a GW Step:



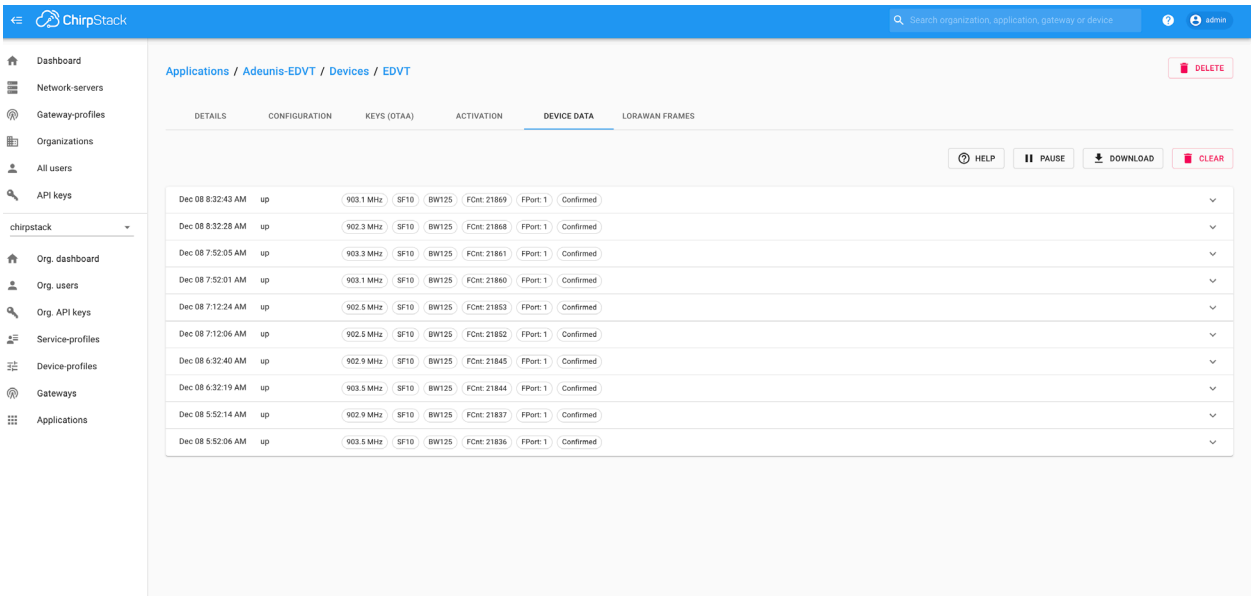
5. Adding Sensor steps:



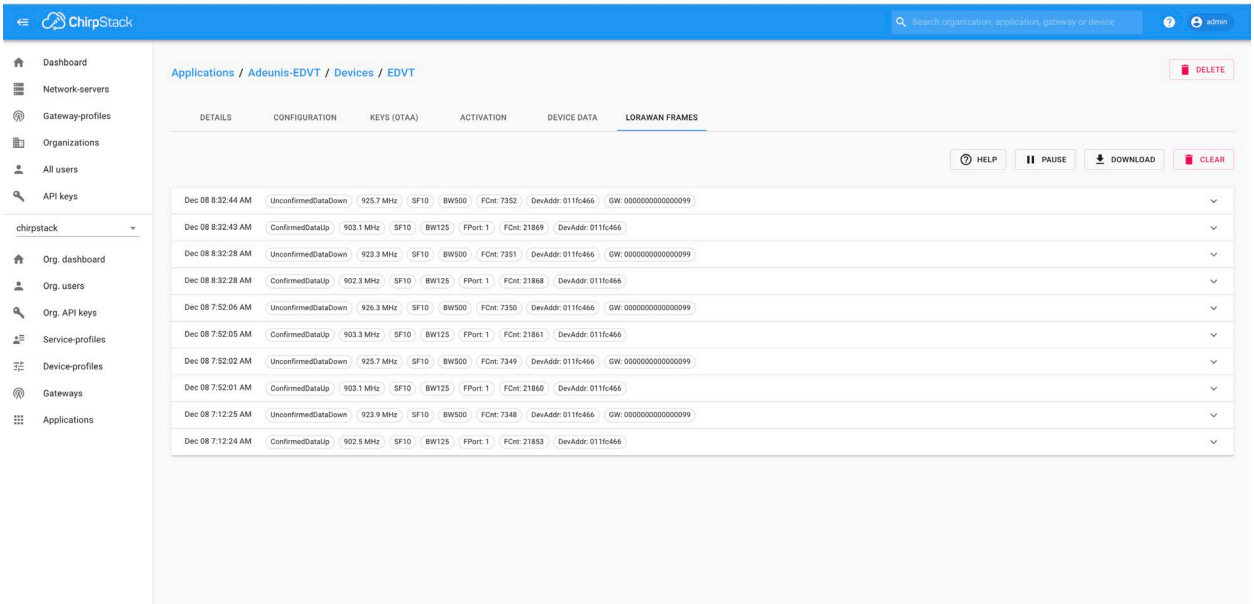
6. Monitoring the Sensor details :



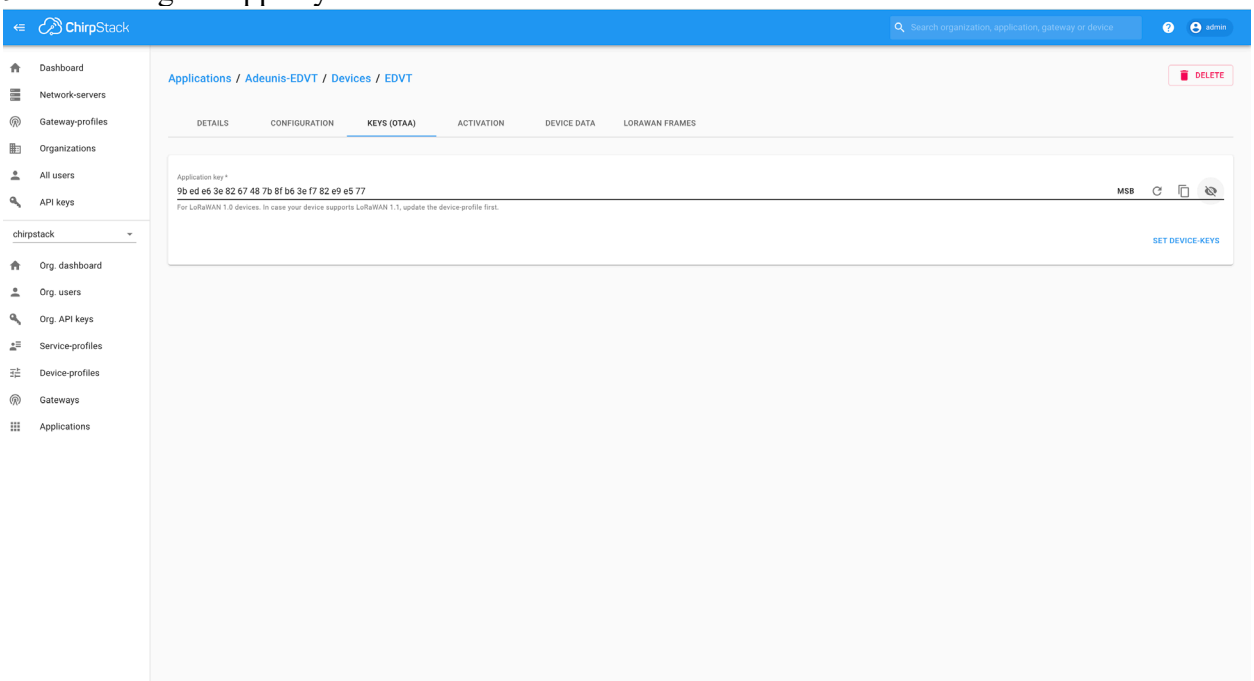
7.Check the Device DATA frames :



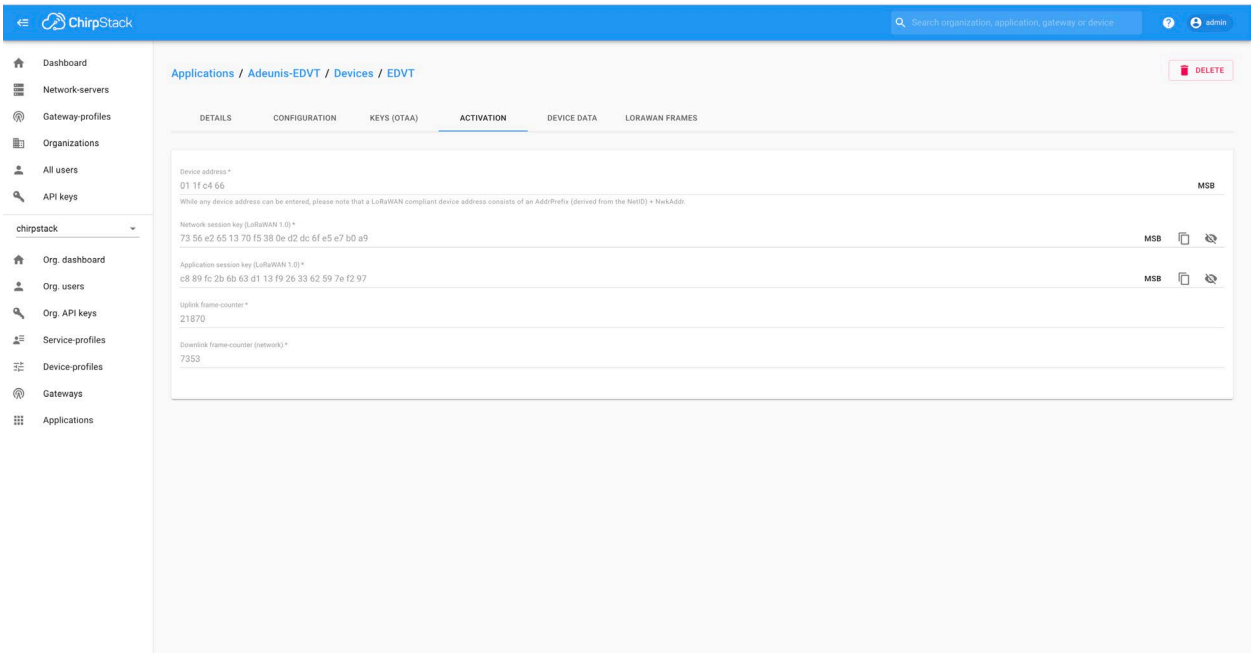
8. Checking Live LoRaWAN Frames:



9. Checking the app keys



10. Check the app session keys and network session keys



Sample running configuration with Common packet forwarder:

```
Building configuration...

Current configuration : 7256 bytes
!
! Last configuration change at 20:05:01 UTC Tue Jun 28 2022
!
version 17.9
service timestamps debug datetime msec
service timestamps log datetime msec
service internal
service call-home
platform qfp utilization monitor load 80
platform punt-keepalive disable-kernel-core
!
hostname Sparrow_Artic_900
!
boot-start-marker
boot system bootflash:ir1101-
universalk9.BLD_V179_THROTTLE_LATEST_20220616_072420.SSA.bin
boot-end-marker
!
!
no aaa new-model
!
!
login block-for 60 attempts 3 within 30
login delay 3
login on-success log
ipv6 unicast-routing
!
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-1417813608
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-1417813608
revocation-check none
```

```
rsakeypair TP-self-signed-1417813608
!
crypto pki trustpoint SLA-TrustPoint
enrollment pkcs12
revocation-check crl
!
no license feature hseck9
license udi pid IR1101-K9 sn FCW2510PMVK
license boot level network-advantage
memory free low-watermark processor 45131
!
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username iox privilege 15 password 0 iox
!
redundancy
!
vlan internal allocation policy ascending
!
interface VirtualPortGroup0
ip address 192.168.0.1 255.255.255.0
ip nat inside
no mop enabled
no mop sysid
!
interface GigabitEthernet0/0/0
ip address 172.27.127.134 255.255.255.0
ip nat outside
!
interface FastEthernet0/0/1
!
interface FastEthernet0/0/2
!
interface FastEthernet0/0/3
!
interface FastEthernet0/0/4
!
interface Vlan1
no ip address
!
```

```
interface Async0/2/0
no ip address
encapsulation scada
!
interface LORAWAN0/1/0
no ip address
common-packet-forwarder profile
country UNITEDSTATES
region-channel-plan US915
gateway-id 54
lns-ip 172.27.127.209
lns-port 6080
log-level debug lines 255
cpf enable
arp timeout 0
no mop enabled
no mop sysid
!
iox
ip forward-protocol nd
ip http server
ip http auth-retry 3 time-window 1
ip http authentication local
ip http secure-server
ip http client source-interface GigabitEthernet0/0/0
ip tftp source-interface GigabitEthernet0/0/0
ip nat inside source list NAT_ACL interface GigabitEthernet0/0/0
overload
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0/0 172.27.127.1
!
!
ip access-list standard NAT_ACL
10 permit any
!
!
control-plane
!
line con 0
stopbits 1
speed 115200
line 0/0/0
line 0/2/0
```

```
line vty 0 4
login
transport input ssh
line vty 5 14
login
transport input ssh
!
call-home
! If contact email address in call-home is configured as sch-smart-
licensing@cisco.com
! the email address configured in Cisco Smart License Portal will be
used as contact email address to send SCH notifications.
contact-email-addr sch-smart-licensing@cisco.com
profile "CiscoTAC-1"
active
destination transport-method http
!
app-hosting appid cp
app-vnic gateway0 virtualportgroup 0 guest-interface 0
guest-ipaddress 192.168.0.9 netmask 255.255.255.0
app-default-gateway 192.168.0.1 guest-interface 0
app-resource docker
run-opts 1 "--device /dev/lorawan_tty1:/dev/ttyACM0"
run-opts 3 "-v /bootflash/lorawan_0:/cpf/"
end
```

Appendix:

- About LoRaWAN: https://www.youtube.com/watch?v=Qd7kMGaQ5vI
- ChirpStack installation: https://www.youtube.com/watch?v=5CCrpqPZBwY
- To know more about Actility : https://www.actility.com/
- About cisco LPWA PIM module : cisco LoRa PIM module