INTEGRATION

PXEN310-0000_TN_PXPORTAL_INTEGRATION

Summary

The document describes how to deploy the PXCom captive portal for embbeded network.

PXEN310-0000_TN_PXPORTAL_INTEGRATION
June 11, 2019
v 1.0.0

P.BARRY
pacome.barry@pxcom.aero

Versions History

Version	Date	Author	Description
1.0.0	11/06/19	P.BARRY	First release

Validations

Actor	Name	Date	Visa
Author	P.BARRY	11/06/19	
Control	[ControlName]	[Date]	
Approval	[ApprovalName]	[Date]	

Contents

1 Introduction **3**

2 PXPortal **4**

2.1 PXPortal - Dnsmasq 5

2.2 PXPortal - Nginx 5

2.3 PXPortal - Portal_Service 6

3 Deployment **7**

3.1 First Run 7

3.1.1 1. Clone ECA-captive repository 7

3.1.2 2. Run PXPortal 7

3.1.3 3. Update PXPortal 7

3.2 SSH Config 8

3.2.1 1. Generate a docker SSH Public key 8

3.2.2 2. Update HOST IP address from portal_service 8

3.2.3 3. Update docker-compose.yml ARP_CMD 8

3.3 Primary Nginx Config 9

3.4 docker-compose.yml 10

3.5 Cisco configuration 11

3.5.1 Wlan 11

3.5.2 DHCP 14

3.5.3 General 14

Introduction

A captive portal is a web page accessed with a web browser that is displayed to newly connected users of a Wi-Fi network before they are granted broader access to network resources.

To determinate if a captive portal has to be display and ask to passenger an action, a phone (Android or iOS) calls some APIs. When this API sends:

- 204 HTTP Code: device has connectivity nothing to do.
- 302 HTTP Code: device maybe have connectivity but passenger has to signin.
- *No answer*: device don't have connectivity but device doesn't known what to do

The goal of PXPortal is to simulate captive portal APIs from different constructors by answering required HTTP codes.

WARNING

The behavior describes here is experimental. Such as every phone constructor implements its own API or internal mechanism, this solution cannot work on all phones (for example some Samsung, ...). The list of captive API will be updated over time.

PXPortal

PXPortal is composed of three services:

- nginx: a web server which can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache
- dnsmasq: provides Domain Name System (DNS) forwarder, Dynamic Host Configuration Protocol (DHCP) server, router advertisement and network boot features for small computer networks, created as free software
- portal_service: internal PXCom service to simuate captive portal API

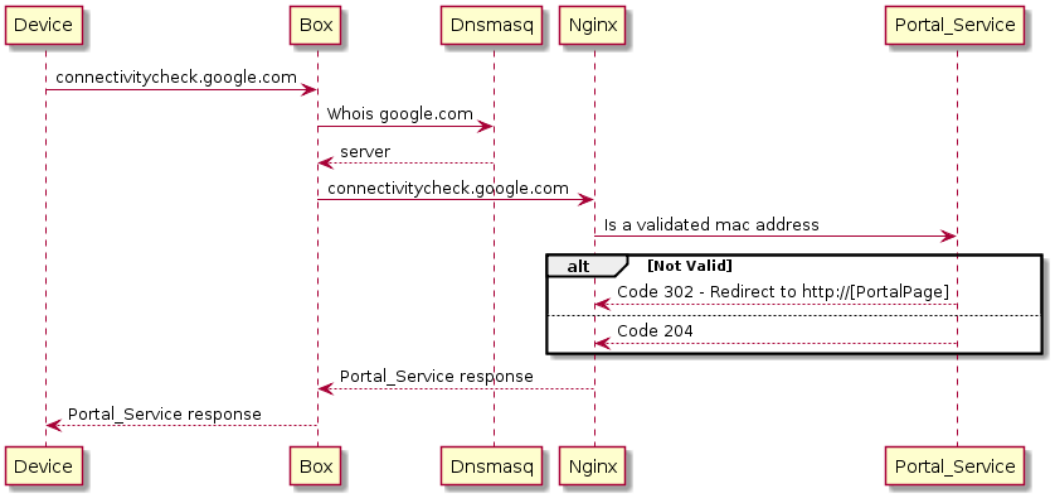


Figure 1: Device without internet but pxportal enabled

To have a validated mac address, a passenger has to accept terms of use on portal webpage. On this action, a request is sent to portal_service which will save in memory the mac address as a valid one. Next time, passenger will be considered as a valid user and no new signin notification will be shown.

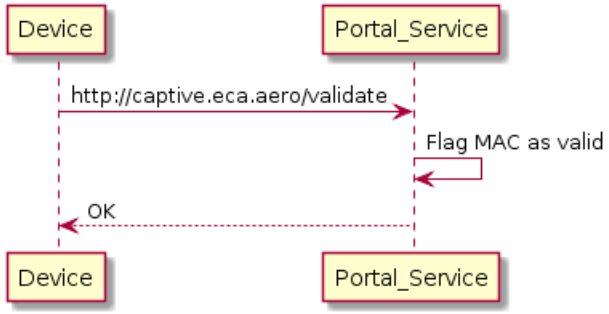


Figure 2: Device without internet but pxportal enabled

PXPortal - Dnsmasq

- redirects all captive portal API to the box. With the following configuration, PXPortal can handles Android and iOS phones and some linux.
- *(optional)* can provide a DHCP server (not used here)

```
# ANDROID
address=/google.com/10.0.0.254
address=/.google.com/10.0.0.254
address=/gstatic.com/10.0.0.254
address=/.gstatic.com/10.0.0.254
address=/android.com/10.0.0.254
address=/.android.com/10.0.0.254
# IOS
address=/apple.com/10.0.0.254
address=/.apple.com/10.0.0.254
# BROWSER
address=/firefox.com/10.0.0.254
address=/.firefox.com/10.0.0.254
# LINUX
address=/ubuntu.com/10.0.0.254
address=/.ubuntu.com/10.0.0.254
address=/gnome.org/10.0.0.254
address=/.gnome.org/10.0.0.254
```

PXPortal - Nginx

- handles all requests sent to captive portal API and forwards them to our pxportal service,
- and serves portal webpages on *http://portal.eca.aero/index.html* and *http://portal.eca.aero/ready.html*

```
server {
  server_name captive.eca.aero
  *.apple.com
  *.gstatic.com
  *.firefox.com
  *.ubuntu.com
  *.google.com
  *.android.com
  *.gnome.org;

  location / {
    proxy_pass http://pxportal_service:8889/;
    proxy_set_header Host $host;
    proxy_set_header X-Real-Ip $remote_addr;
    proxy_buffering off;
  }
}

server {
    server_name portal.eca.aero;
    root /data/webapp/portal;
}
```

PXPortal - Portal_Service

- is a server HTTP based on NodeJs and ExpressJs
- simulates captive portal APIs
- saves validated mac addresses on */validate* call. It's saved in memory that's mean, it's cleared on one service restart
- performs an ARP command to find mac address from IP client

Deployment

PXPortal can be deployed with a docker-compose.

Before continuing, be sure docker and docker-compose are installed on your machine.

First Run

1. Clone ECA-captive repository

```
git clone ssh://gitolite@git.pxcom.aero:2221/ife/ECA-captive.git
```

This repository contains:

- **docker-compose.yml**: docker config of three PXPortal services
- **conf**: dnsmasq and Nginx configuration
- **webapp**: sources of portal webpage

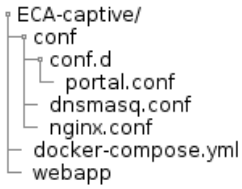


Figure 3: ECA-captive tree

2. Run PXPortal

```
# Go into ECA-captive folder
cd ECA-captive
# Start all services in daemon mode
docker-compose up -d
```

3. Update PXPortal

(if updates are available)

```
# Go into ECA-captive folder
cd ECA-captive
# Pull latest updated dockers
docker-compose pull
# Restart dockers
docker-compose up -d
```

SSH Config

Such as portal_service requires MAC addresses thanks to ARP command. But from one docker context, the command has to execute from the host in order to get MAC addresses from IP clients. This is *how to do this* in few steps

1. Generate a docker SSH Public key

```
### On your HOST
# Launch portal_service shell
docker exec -it pxportal_service sh

### On portal_service shell
# Generate a SSH key to be able to send ARP command from docker to host
ssh-keygen
# Display and copy ssh public key
cat /root/.ssh/id_rsa.pub

### Exit portal_shell
# Edit HOST ssh authorized_keys and paste previous ssh public key of portal_service
vi ~/.ssh/authorized_keys
```

2. Update HOST IP address from portal_service

```
### On your HOST
# Launch portal_service shell
docker exec -it pxportal_service sh

### On portal_service shell
# Show all network interfaces available
ifconfig
# Keep in memory the inet addr of eth0
# HOST IP address from docker should be this IP address by remplacing last number by 1
# EX: 172.20.0.2 => 172.20.0.1 = HOST_IP
# Check your ssh public key is set correctly and update known_hosts on first SSH connection
ssh [HOST_NAME]@[HOST_IP]
# EX ssh elta@172.20.0.1
# Check ARP command can be run
ssh [HOST_NAME]@[HOST_IP] arp -n
# A list of IP with matching MAC will be appeared
# If not, try again from the beginning
```

3. Update docker-compose.yml ARP_CMD

Update the environment variable **ARP_CMD** in *your docker-compose.yml* with the right HOST_NAME and HOST_IP found in step 2.

```
# Go into ECA-captive folder
cd ECA-captive
# Start all services in daemon mode
docker-compose up -d
```

Primary Nginx Config

The ECA-portal Nginx is configured as a slave reverse proxy. That's why it is not used the 80 by default.

But to be able to handle captive portal API, the primary nginx running on port 80 should redirect all unknown URLS to portal nginx. This is an example of configuration to add into a primary nginx configuration:

```
....
server {
    listen 80 default_server;
    server_name _;

    location / {
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header X-Forwarded-For $remote_addr;
      proxy_set_header X-Forwarded-Host $remote_addr;
      proxy_pass http://172.19.0.1:8031;
  }

  access_log      captive-pxportal-access.log;
  error_log       captive-pxportal-error.log;
}
....
```

docker-compose.yml

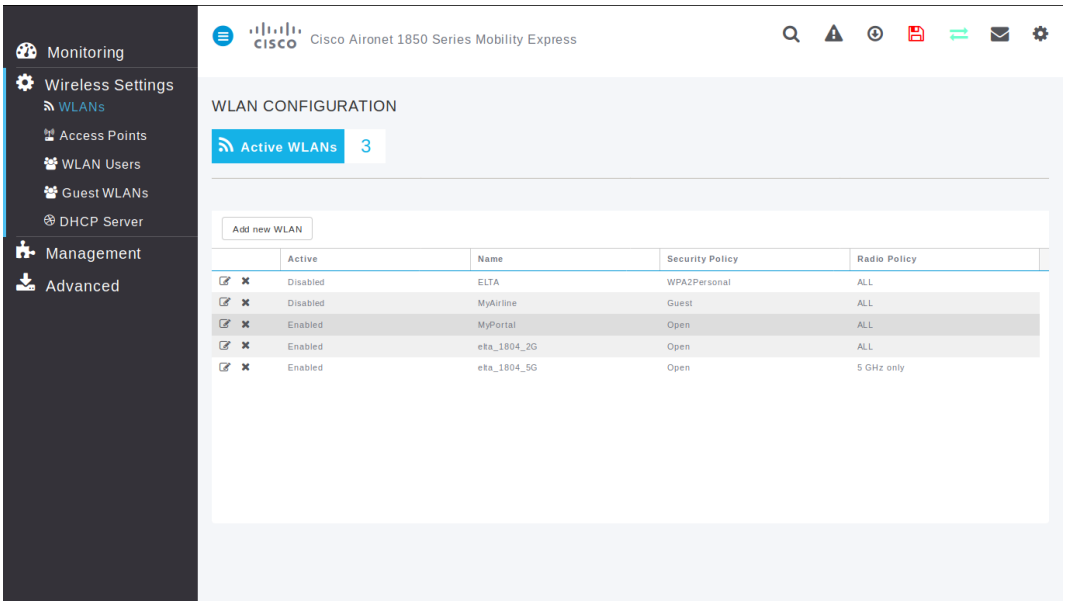
```
version: '2'

services:
  nginx:
    image: nginx:alpine
    container_name: pxportal_nginx
    networks:
      - pxportal
    ports:
      - "8031:80"
    volumes:
      - ./webapp:/data/webapp
      - ./conf/nginx.conf:/etc/nginx/nginx.conf
      - ./conf/conf.d:/etc/nginx/conf.d/

  dnsmasq:
    image: andyshinn/dnsmasq
    container_name: pxportal_dnsmasq
    ports:
      - "53:53/tcp"
      - "53:53/udp"
    cap_add:
      - NET_ADMIN
    volumes:
      - ./conf/dnsmasq.conf:/etc/dnsmasq.conf
    depends_on:
      - pxportal

  pxportal:
    container_name: pxportal_service
    image: registry.preprod.pxcom.aero/pxcom-servers/pxportal-srv:latest
    volumes:
      - ./webapp:/usr/app/webapp
      - ssh:/root/.ssh
    networks:
      - pxportal
    environment:
      - REDIRECT_TO=http://portal.eca.aero/index.html
# TO UPDATE FROM SSH Config part
      - ARP_CMD=ssh elta@172.21.0.1 arp -n
    ports:
      - "8889:8889"

volumes:
  ssh:

networks:
  pxportal:
    driver: bridge
```

Cisco configuration

Wlan

Wlan - Home

Go to **Wireless Settings / WLANs** page to create or configure a WLAN for passengers.



Wlan - General

- **Profile Name**: [Name of passengers WIFI]
- **SSID**: [Name of passengers WIFI]

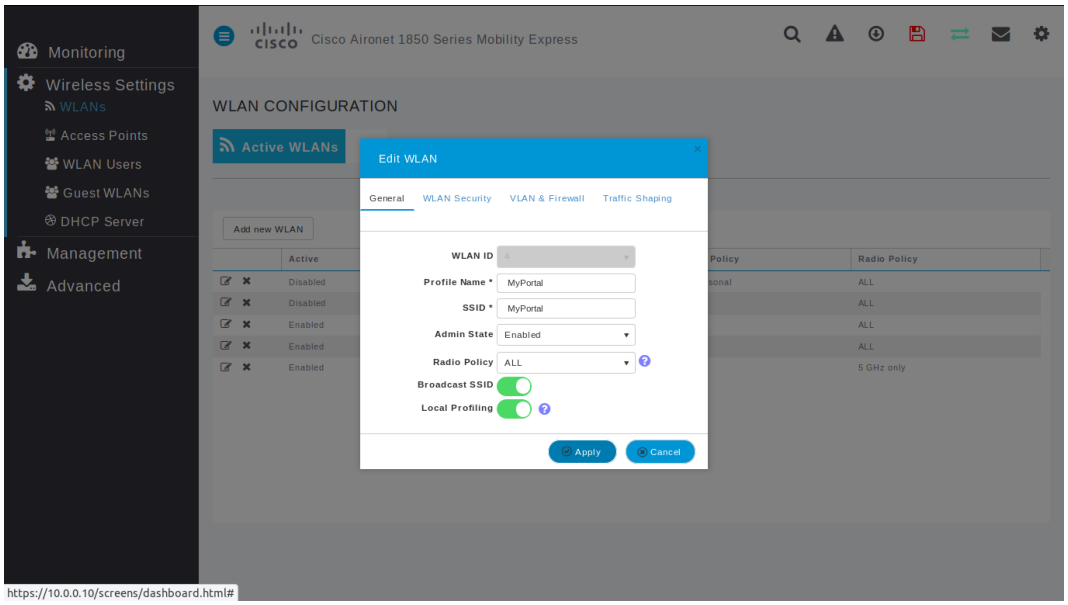
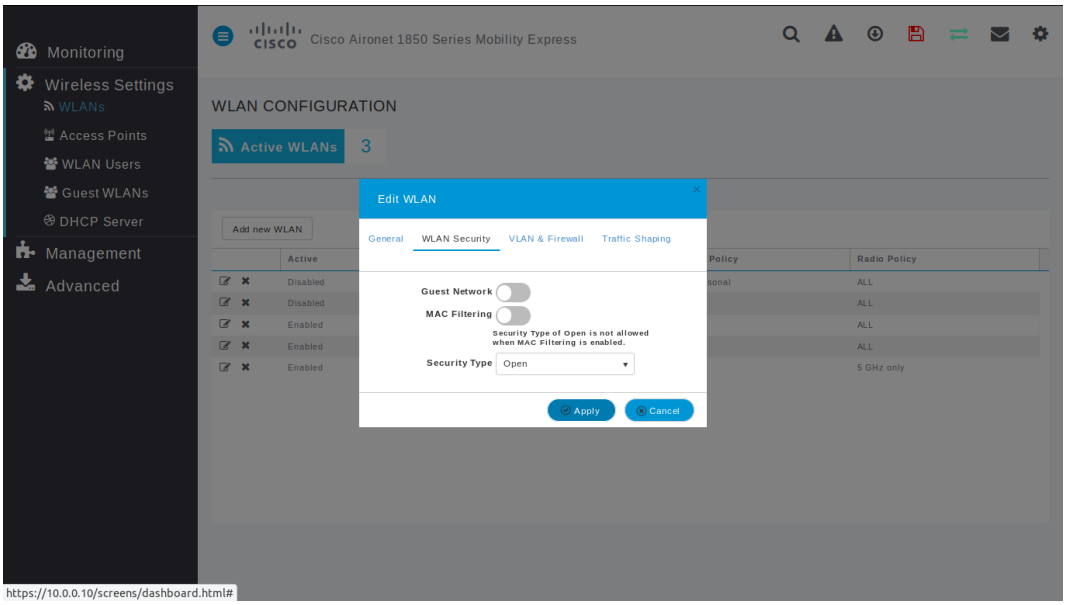


Figure 4: Wlan Edition - General

Wlan - WLAN Security

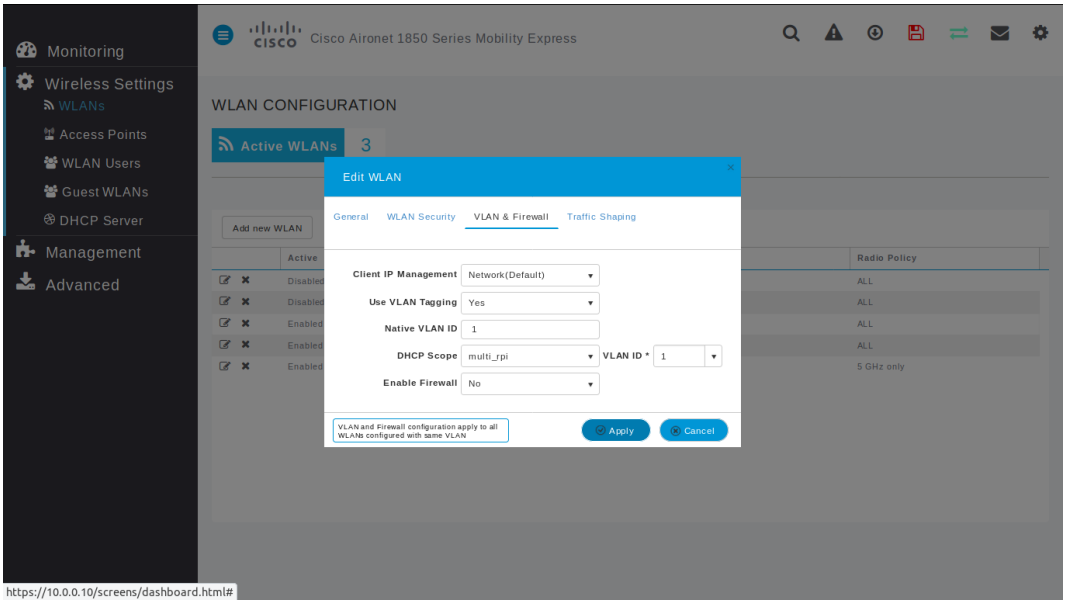
Such as CISCO captive portal forces a HTTPS page and requires to check SSL certificates from external network, PXPortal doesn't use it and provides its own captive portal. That's why **Guest Network** is disabled and the WIFI network is opened.

- **Guest Network**: *disabled*
- **Security type**: *Open*



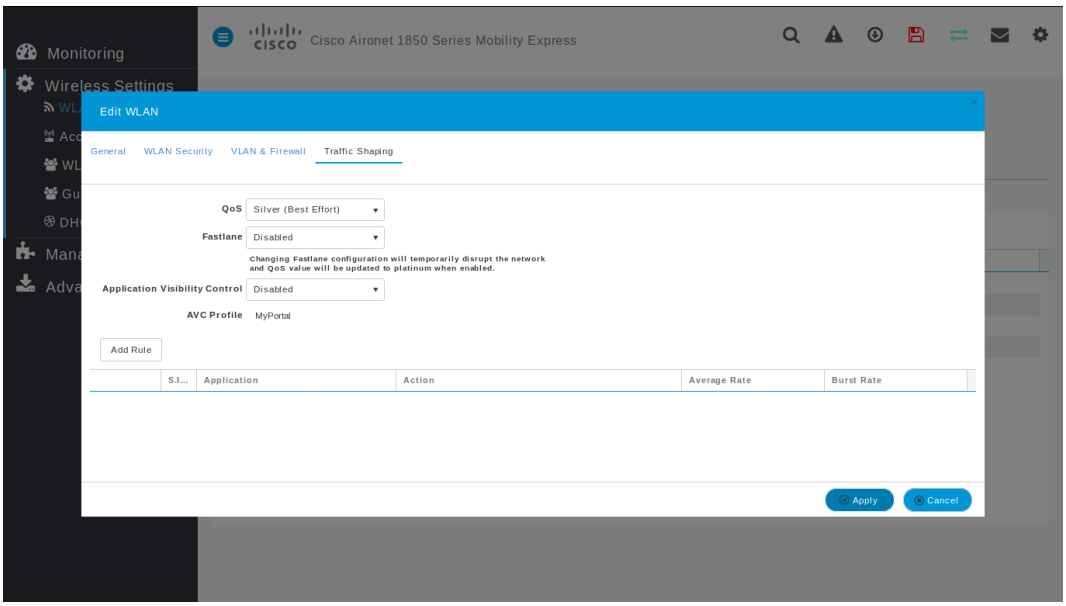
Wlan - VLAN & Firwall

Default configuration



Wlan - Traffic Shaping

Default configuration

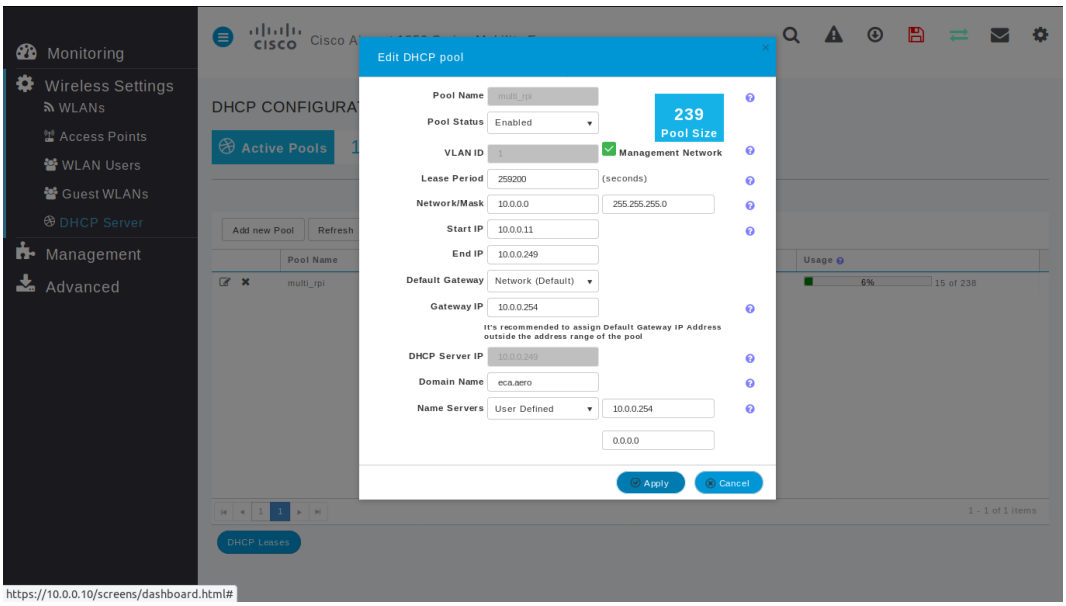


DHCP

By default, PXPortal uses Cisco DHCP.

General

- **Pool Status**: Enabled
- **Network / Mask**: 10.0.0.0 / 255.255.255.0
- **Start IP**: 10.0.0.11
- **End IP**: 10.0.0.249
- **Gateway IP**: 10.0.0.254
- **Domain Name**: eca.aero
- **Name servers**: User Defined / 10.0.0.254



List of Figures

1 Device without internet but pxportal enabled 4

2 Device without internet but pxportal enabled 4

3 ECA-captive tree 7

4 Wlan Edition - General 11