**SDWAN-zWAN**

zWAN CPE Firmware Image Installation on different platforms



*Published by Software Engineering*

AMZETTA-CONFIDENTIAL

Copyright AmZetta Technologies

5555 Oakbrook Pkwy Suite 280, Norcross, Georgia 30093, U.S.A.

All Rights Reserved

Confidential, Unpublished Document

Property of AmZetta Technologies

Logo

Description automatically generated

* **X86 CPE HARDWARE – Firmware Image Installation to HD/SSD with VGA/HDMI Port:**
* Host system (Ubuntu) pre-req:
* apt-get install qemu-utils
* Download qcow2 img convert script ([vcpe\_qcow2\_to\_verboot\_raw\_img.sh](ftp://zWAN@ussftp.amzetta.com/sdwan-zwan-build/1.0-1.1006-QA/vcpe_qcow2_to_verboot_raw_img.sh)) and qcow2 files into local ubuntu desktop machine and execute the below command.
  + **sudo su**
  + **./vcpe\_qcow2\_to\_verboot\_raw\_img.sh  $image.qcow2.xz**

Note: It will automatically generate PRODUCT\_ID for the output image. Now x86 img file created along with a unique product id.

* Connect the x86 Device with Monitor, Keyboard and mouse.
* Make 2 Pen drive available with 16 GB and 32 GB
* Make the 16 GB Pen drive [bootable with ubuntu desktop 18 or 20 OS](https://ubuntu.com/tutorials/create-a-usb-stick-on-ubuntu#1-overview) using any bootable application like rufus tool.
* Format the other 32GB pen drive with NTFS format. Copy the generated .img file in bootable pen drive. (Approx 20GB)
* Connect the 16GB bootable pen-drive in x86 hardware, get into the BIOS screen after powering it ON -> enable UEFI mode if supported.
* If the pen drive is not listed in the boot priority selection (BOOT OPTION #1) menu, kindly enable the bootable device in the Hard Drive BBS Priorities. Once enabled, it will be available in the boot options menu. Select it as the first priority and continue.
* Boot with Ubuntu OS by choosing option “Try Ubuntu” and then login into the OS.
* Open Terminal, execute the command **fdisk -l** and note down the x86 hardware internal hard disk drive letter name (example: /dev/sdX).
* Now connect the 32 GB Pen drive with the image copied to the x86 device.
* Open Terminal, execute the command **fdisk -l** and note down the 32Gb connected Pen drive path (example: /media/ubuntu/<pendrive name>/<cpeimage.img>).
* Execute the below command to flash the cpe firmware in x86 hardware.

**sudo dd if=/media/ubuntu/pendrive name/cpe.xxxx.img of=/dev/sdX bs=4M status=progress oflag=direct conv=fsync && sync**

* + - **where if ->refers to image pendrive path and of -> refers to internal HDD path of x86 device**
* Once the above command is completed, then reboot the device. Again, get into BIOS and make sure that the first boot options UEFI OS boot HDD.
* Once the system starts with initial screen remove the pen drive. Now the system will be booting with internal hard disk with flashed firmware.

**X86 CPE HARDWARE – Firmware Image Installation to HD/SSD with Serial Console (No Display card - VGA / HDMI):**

Some of x86\_64 CPE platforms (e.g Lanner / CAF-0262) do not come with any display ports. They have only have SERIAL/USB ports.

To flash zWAN Firmware on such platforms, we need to connect USB-to-Serial cable in x86 device to take console with the help of a windows machine using putty tool.

Follow the below steps to flash firmware via text console mode,

* Boot the CPE using Live Ubuntu server Image. (Refer the above steps to make the bootable process)
* Edit the Try or Install Ubuntu server option in GRUB mode and press e and do the below changes and give ctrl+x.

***console=ttyS0 console=ttyS0,115200n8***

A screenshot of a computer

Description automatically generated

* Once it is booted into Ubuntu Live image (Can go via help menu to enter shell or ctrl+alt+F2).
* Copy the latest zWAN FW image file into another USB pen drive and connect it to the CPE
* Mount the USB pendrive into /media/usb0 using below commands.
  + - sudo su
    - mkdir -p /media/usb0
    - ***mount /dev/sd<X> , where x is the path of the USB pendive we connected.***
    - ***cd /media/usb0*** and type ls to ensure the FW image file is present.
* Type ***dd if=/media/usb0/(Image name.img) of=/dev/sda bs=4M status=progress oflag=direct conv=fsync && sync***
* Shutdown the CPE and remove all the pen drives
* Boot it thru' Hard drive image

a. UEFI Boot mode should be enabled in CPE's BIOS to boot into zWAN Firmware successfully.

* It will boot into zWAN CPE firmware.
* Onboard it to the director and test the functionality.

**X86 VCPE (Virtual CPE) : KVM Host - Firmware Image Installation :**

* Refer the below git link for more details.

<https://gitlab.amzetta.com/sdwan/vcpe-deployment>

**vCPE deployment**

**Overall Customer Environment**

 Provisioning servers expected on customer’s network.

 KVM host server with sufficient memory, cpu, and NICs

 Host server is running Ubuntu 18.04 or 20.04

 Server NICs can be shared (WAN shared between multiple CPEs as WAN, LAN shared between LAN VMs and physical systems)

**Process**

 run all the commands as root user (sudo su)

 download the vcpe deployment script file and Untar the vCPE-deployment.tar in the customer server

 tar –xvf vCPE-deployment.tar --one-top-level

 cd into the vCPE-deployment folder

dos2unix \*.sh

chmod +x \*.sh

**customer\_deploy.sh**

bash customer\_deploy.sh

 Will install the necessary packages for KVM deployment

 Check the system hardware that it supports VMs

 Will prompt the user for the FTP credentials, and location to download the cpe and lan base images

 Will import the base images as golden base vms so that vCPE or LAN vms can be cloned of.

 Please note customer\_deploy.sh is only needed to create the golden image for cpe and lan. So need to be executed only once. If you need to pull a new base image, only then execute customer\_deploy.sh over again.

 Once this is completed, then proceed to add\_cpe.sh

**add\_cpe.sh**

It can be deployed using prompted questions or a pre-populated YAML file configuration.

bash add\_cpe.sh

OR

bash add\_cpe.sh config.yaml

 Will prompt the user for multiple questions

 Asks for the LAN VM count

 WAN and physical connections are determined

 Units are cloned and started

 CPEs can be deployed from a yaml based configuration too..

 You can execute add\_cpe.sh multiple times if you need more vCPEs

**cleanup\_cpe.sh**

bash cleanup\_cpe.sh <CPENAME>

* **ARM CPE Firmware Image Installation on SDMMC and EMMC:**
* Use Ubuntu 18.04 or 20.04 to flash: apt-get –y install ifupdown unzip build-essential libelf-dev zlib1g-dev python3-distutils
* Copy flasher.sh script and image into a linux (Ubuntu based system) and flash image to sdcard. Before flashing ensure your device is unmounted.
  + Example**: bash flasher.sh /dev/sdc 0 build-bpir64-1.0-1.1006-QA10-May04-2022-124336.bin 7456 bpir64**
  + Example**: bash flasher.sh /dev/sdc 0 build-bpir64-1.0-1.1006-QA10-May04-2022-124336.bin 7456 amz7622**
  + Example**: bash flasher.sh /dev/sdc 0 build-bpir64-1.0-1.1006-QA10-May04-2022-124336.bin 15028 cwan801**
* wherearg1: sdcard device,arg2: 0-sdcard/1-emmc, arg3: image name, arg4:sdcard size, arg5: model
* Insert SDcard to CPE and boot from sdcard.
  + *For BPI*, flash sdcard to emmc using secure shell. It will set to boot from emmc.
* If you want to boot again with sdcard follow the below steps:
  + Restart system and enter to uboot console
  + Execute following commands which will set to boot from sd card.
  + run **sdnewboot**
  + Switch position: 1 - to use uboot from sdcard (0 for to use uboot from emmc)
* amz\_Secureshell options for testing
  + Option 17: is to sdcard to emmc flash *(for BPI)*
  + Option 19: is for update/recovery firmware testing operations.
* **FLASH ARM64 (BPIR64/CWAN801/AMZ7622) Firmware Image to SD CARD:**
* Insert the 16GB SD card into a card reader on any Ubuntu system.
* Copy the build image (specify the latest version) from the build system using the following path: root@system-ip:/home/release-images to a specific directory on the Ubuntu system where the SD card is already inserted.

Syntax:

**#scp flasher.sh build-xxx.bin root@system-ip:/home/**

* Once the image has been copied in the specified path run this command.

Syntax:

**#bash -x flasher.sh sd-path flag build-xxx.bin 7456 hardware-model**

* Sd-path - /dev/sdc or /dev/sdb
* Flag - 0/1 (Ex: 0(sd-card)/1(EMMC)).
* Hardware Model: bpir64/cwan801/amz7622
* The flasher.sh script will begin flashing the ARM image onto the SD card in the next 2 to 3 minutes. Once it completes, remove the SD card and insert it into any BPIR64 hardware.
* Power on the hardware, and once the boot menu appears, select the U-Boot console option and press ENTER. The U-Boot console will appear. Type the command **'run sdnewboot'** in the U-Boot console terminal and press ENTER; this will boot the image from the SD card.
* Do not forget to flash the image onto the eMMC by selecting the 17th option in the amz\_Secureshell. This will write the image onto the eMMC memory permanently.
* Reboot the system.
* Once the system starts with initial screen remove thesd-card. Now the system will be booting with internal eMMC memory with flashed firmware.
* **FLASH ARM(BPI-R3) IMAGE TO EMMC:**
* Format USB pen drive in ext4 from any Linux machine and copy the BPI\_R3.img and flash\_emmc.sh file to the USB pen drive
* Connect the Pen drive to the BPI-R3 device.
* Put/make down all 4 flipflops switch (Below Wi-Fi antenna side view). Power on the unit.
* Now by default the device will boot in NOR mode with default opwnwrt image.
* Once the device booted successfully, avigate to the usb mount location( example: cd /mnt/sda1)
* Execute the command - sh -x flash\_emmc.sh filename.img . It will start flashing to internal emmc drive,
* Once the flash get completed successfully, Power off the unit.
* Now in flipflop switch from left side - down the first and fourth and then UP second and third.
* Power on the unit, now it will boot our firmware from emmc drive.
* Now onboard the device to our director.
* **CPE FIRMWARE UPGRADE (AND SAMPLE CONFIG FILE)**:

|  |
| --- |
| Keep the patch image and manifest.json file in Director minio console |
| Sample firmware update manifest.json:  {  "model": "ZMTKTNA3QP055VL",  "version": "1.0-1.1006-QA1",  "images": [  {  "from" : "1.0-1.1005-QA3",  "to" : "1.0-1.1006-QA1",  "path": "https://zwan-fw.s3.amazonaws.com/ZMTKTNA3QP055VL/1.0-1.1006-QA1/cpe-fw-package-1.0-1.1006-QA1-from-1.0-1.1005-QA3.squash"  },  {  "from" : "\*",  "to" : "1.0-1.1006-QA1",  "path": "https://zwan-fw.s3.amazonaws.com/ZMTKTNA3QP055VL/1.0-1.1006-QA1/cpe-fw-package-1.0-1.1006-QA3-from-1.0-1.1005-QAbase.squash"  }  ]  } |
| Update the link to manifest.json into Provider MSP:  <https://PROVIDER_IP:7080/#/system-tools/firmware>  <https://PROVIDER_IP:7080/minioconsole/> |

**VCPE DEPLOYMENT:**

Refer Git [Readme](https://gitlab.amzetta.com/sdwan/vcpe-deployment/blob/development/README.md)

**DEFAULT CREDENTIALS:**

Managed service provider(MSP) to create additional tenants:

[http://providerip:7080](http://10.11.111.29:7080/#/tenant-management/tenant) ([admin@zwan-msp.com/zWAN@teamw0rk](mailto:admin@zwan-msp.com/zWAN@teamw0rk))

minio login : minioadmin / minioadmin

Once tenant is created, tenant can be access as below:

[https://providerip:8443/<tenantname](https://providerip:8443/%3ctenantname)>

Default tenant zwan-tenant created and can be accessed as below:

<https://ipaddress/zwan-tenant> ([admin@zwan-tenant.com / zWAN@teamw0rk](mailto:admin@zwan-tenant.com/zWAN@teamw0rk))

**CPE LOCAL UI:**

[http://CPE\_LANIP](https://10.11.111.29:8443/zwan-tenant) (router / zWAN@teamw0rk)

Secure shell: ssh –p 1222 router@CPE\_LANIP (router / zWAN@teamw0rk)