**Prime-Infrastructure-configuration-Plug-and-play – Understanding Document**

All installation is done at the clients end

Server:

[https://10.104.118.194](https://10.104.118.194/) [root/Public123]

List of page URL’s.

* Configuration / Plug and Play / Bootstrap
* Configuration / Plug and Play / Profiles
* Configuration / Plug and Play / Status
* Administration / Servers / APIC-EM Controller/  APIC-EM Controller
* Administration / Servers / APIC-EM Controller/ Global PnP/ZTD Settings

Once plug and play is done, device is added in inventory of PI.

* Configuration / Network / Network Devices

Plug and play is cloud based server where various devices can register themselves and then they can report to Prime Infrastructure

It auto discover all the devices which get connected to the enterprise. An enterprise can be a global enterprise like CISCO.

Design has to be self-intuitive for an operator and simple enough to set up the PnP server and configuration. Devices can be called on and bootstrapped in the system.

The idea behind the Plug and Play is that the end user should have a clean box and should be able to provision some details like image, configuration and should be able to manage the device using Prime Infrastructure. When the user purchase a new device configuration can be managed using 2 options

* Using CNS gateway mode
* APIC- EM Model

For both of these there is Plug and Play profile, where the user can create a high level profile which contains details of what exactly needs to be pushed. It contains only skeleton/little information on what needs to push in the device.

Configuration / Plug and Play / Profiles

When the user clicks on “Add” button, it creates a template with what are the details we are going to push & purpose.

Name and description displays what is the lump sum description about it.

Validation Criteria

It is an optional field. It is the selection of specific device type like router, switch or any specific entries.

Profile Details

**Bootstrap Template**

There are specific options like **bootstrap templates** that have the minimum set of configuration that is required for device to know that the device has come up or when the device boots up clean whom it should contact, what is the communication protocol, what is the IP address of the server. This kind of basic information is available in Bootstrap template.

*“What is Template?”*

In Prime Infrastructure there is a configuration template. It is a skeleton where what is the command and values for each of the values can be specified. Ideally there are set of predefined templates where you can select a template and provide values to it, it will automatically generate the configuration commands for the router or switch. User provides the input values and for each of the field and it generates a configuration commands to provision on the device. If talk about the Bootstrap template, it is going to generate a CLI command but specific to the minimal set of commands said earlier.

**Software Image**

End user wanted the provision of adding image on the device. For e.g. a new device is coming up, user wants a specific version of image on the device when it first comes or before any configuration on the device or before user do management part of it, user wants to install a specific image on the particular device. There is a different module called swim which is like a software container where user can upload an image. It is an optional field. If user don’t know the software provisioning he can ignore this field.

**Image location**

It is specific to the software image in the router or switch where the user wants to copy the image e.g. flash.

**Configuration Template**

Final thing is the configuration template. As said, the template is going generate the CLI commands based on what user want to provision. There are 2 things

* Predefined templates that can be used. It will automatically generates the CLI commands which are pushed to the device
* Alternatively user can also define his own templates: User defined templates

All these don’t have specific values associated to it but are skeletons. After giving the required values the user can save the profile or skeleton. This profile or skeleton will have the device specific details later.

The second part is to deploy the created profiles. That means user is going to provide the values for the template/profiles created. User clicks on “Deploy” after saving the template. It gives a popup *“Device provisioning profile”*, there are few options in this to provide details or specific details or user can import a .CSV file in this. On click of Add button user will see whatever he has selected for that template in a popup “Add devices for pre-provisioning”. E.g. Bootstrap template, configuration template. Once the user add a specific serial number in ‘Device Plug & Play ID’ information updated in this section will act as guide for provisioning that device. Actual values are provided in this section. Default values get auto-populated and if user wants to change these he can.

Configuration section gets populated dynamically with the fields based on the template/skeleton selected.

The real problem to be solved is with the big customers as they have 10000 or more devices which are geographically distributed and have different wings like retail, distributors, corporate, warehouse.

Device Management Parameters

The ultimate goal is to push the image or configuration when the device comes up clean. Once all the provision is done and a clean device comes up, it talks to the controller, it gives the configuration and image provisions. Now the device should be manageable from prime infrastructure. For that you need to provide some management credentials in Device Management Parameters. PI can then manage the device with same credentials. The final step is to add the device to Prime Inventory which means the device is manageable. After the device is added in Device Management Parameters/Inventory PnP work flow is done.

Inbetween there is a status page where user can view the in progress or configuration in progress stuff.