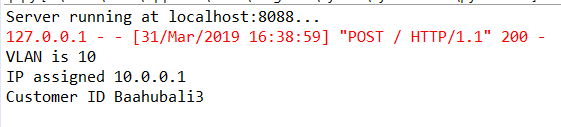
**WEEK 1: ISP FrontEnd 🡺API call**

* **The code allows Content provider to send customer information to the ISP to create respective parameters in the network.**
* **When the customer clicks on the Content Provider’s URL to create its own website. Content provider will push 3 parameters for the customer on the ISP side:**
  1. **Customer VLAN**
  2. **Customer ID**
  3. **IP address assigned for the customer URL**
* **This is the code for API call where Content provider can do “curl” request to the ISP with all the parameters that it wants to push on the ISP side**
* **CURL request from the terminal to POST:**



* **Output of the code:**



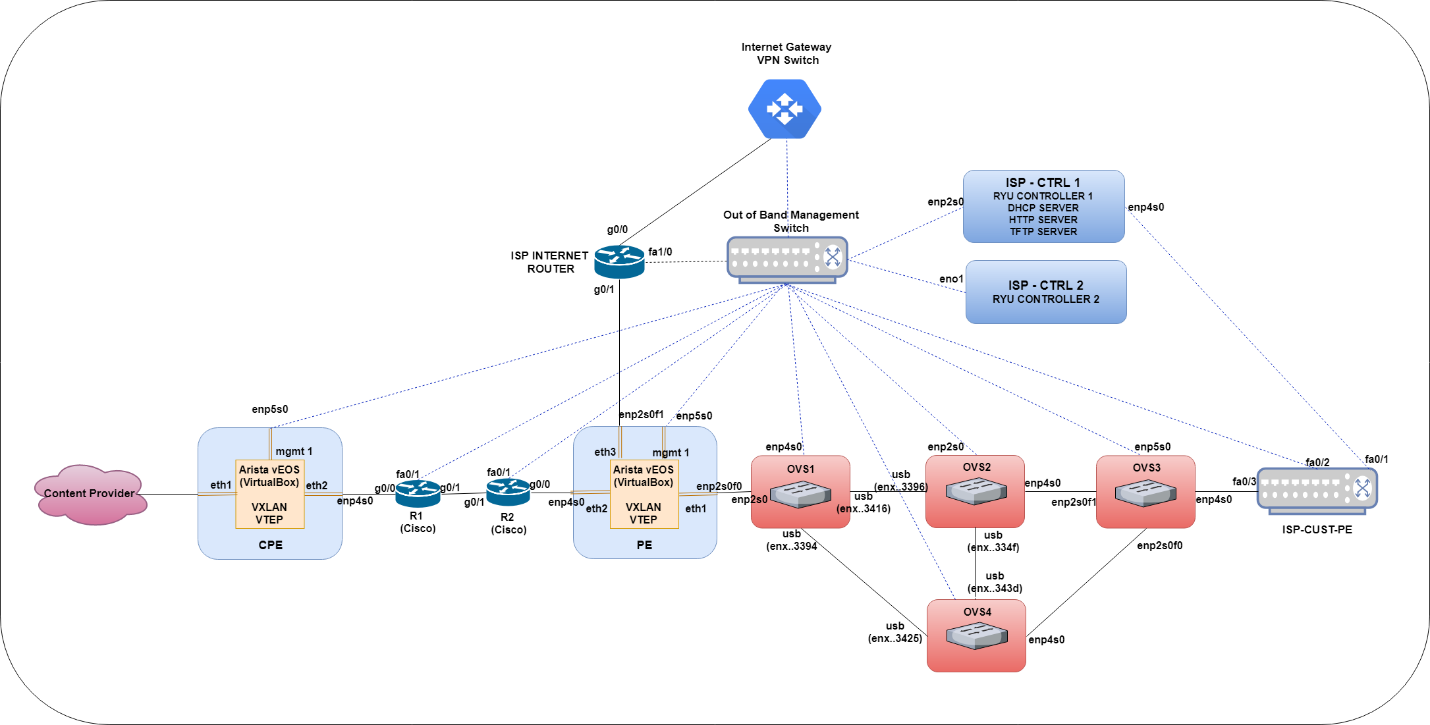
* **Respectively for every user, VLAN, IP and Customer ID can be created using a curl request from the Content Provider end.**

**APPENDIX**

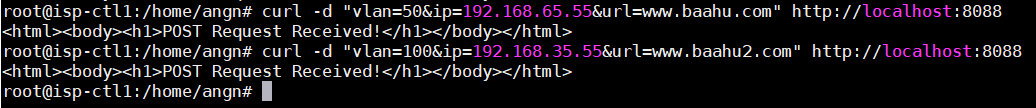
* **Sample curl commands to GET, HEAD and POST**

1. **POST ==> curl -d "foo=bar&bin=go" http://localhost:8088**
2. **GET ==> curl "http://localhost:8088?foo=bar&bin=go"**
3. **HEAD ==> curl -I http://localhost:8088**

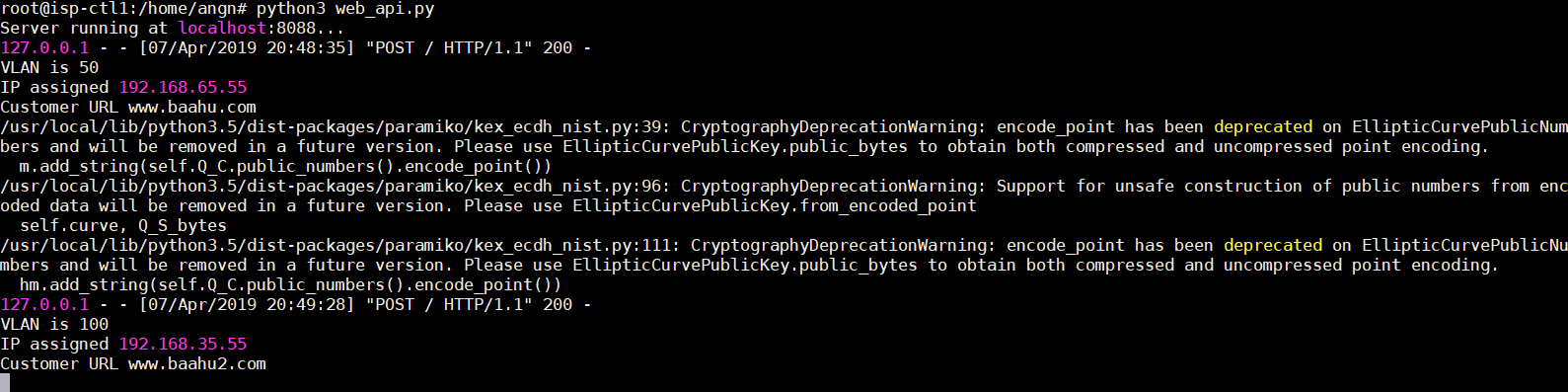
**WEEK 2 – ISP BACKEND 🡺 API working**



* **A python code “web\_api.py” is running in controller 1.**
* **It is listening for POST request from the Content Provider and configures customer specific information**
* **An example of the POST request from controller 1 terminal. However, when sending a POST request from Content Provider, ensure the POST request is addressed to controller 1 IP address (172.16.16.1)**

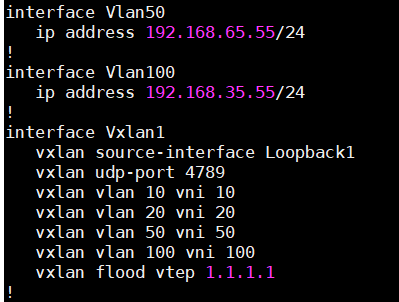


* **After receiving the POST request, controller configures the respective VLAN and IP address for the customer**
* **Controller receiving POST request**



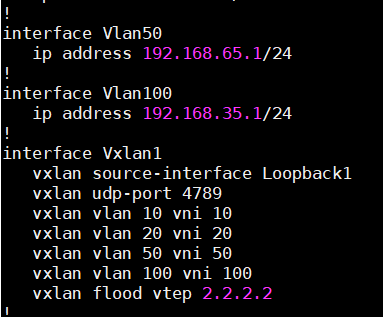
* **Configuration done on the CPE side based on the POST request**





* **Configuration done on the PE side based on the POST request**





* **Installed DNS server on controller and created zone files.**