CUPS Print Server Configuration Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 40 Agenda:
1. CUPS Print Server Configuration Using Ansible

CUPS (formerly an acronym for Common UNIX Printing System) is a modular printing system for Unix-like computer operating systems which allows a computer to act as a print server. A computer running CUPS is a host that can accept print jobs from client computers, process them, and send them to the appropriate printer.

CUPS consists of a print spooler and scheduler, a filter system that converts the print data to a format that the printer will understand, and a backend system that sends this data to the print device. CUPS uses the Internet Printing Protocol (IPP) as the basis for managing print jobs and queues. It also provides the traditional command line interfaces for the System V and Berkeley print systems, and provides support for the Berkeley print system's Line Printer Daemon protocol and limited support for the Server Message Block (SMB) protocol.

Let’s create an ansible playbook to configure the CUPS print server at the managed node(s).

```
$ vim print-server.yml
---
- name: CUPS Print Server Configuration Playbook
  hosts: node1
  become: true
  tasks:
   - name: Setting up the static hostname in the machine.
     hostname:
       name: print-server.nehraclasses.local
       use: systemd

   - name: Installing CUPS packages in the machine.
     dnf:
       name: cups
       state: latest

   - name: Making changes in the configuration file.
     lineinfile:
       path: /etc/cups/cupsd.conf
       insertafter: "Order allow,deny"
       line: "Allow @LOCAL"

   - name: Allowing TCP port 631 traffic in the firewall.
     firewalld:
       port: 631/tcp
       zone: public
       permanent: true
       immediate: true
       state: enabled

   - name: Starting & enabling the Cups service.
     service:
       name: cups
```

```
      state: started
      enabled: yes

  - name: Starting & enabling the Cups-browsed service.
    service:
      name: cups-browsed
      state: started
      enabled: yes
...
```

Install the ansible posix collection from the ansible galaxy so that we can use firewalld module.

```
$ ansible-galaxy collection install ansible.posix
```

Run the playbook to setup the print server at the managed node(s).

```
$ ansible-playbook print-server.yml
```

Verify the status of the cups service at the managed node(s).

```
$ ansible node1 -m command -a 'systemctl status cups'
```

Install the gutenprint-cups package, which provides CUPS drivers for Canon, Epson, HP and compatible printers. If you have an HP printer, you can easily include the hplip package as well.

```
$ ansible node1 -m command -a 'sudo dnf install gutenprint-cups hplip -y'
```

Check the connectivity with your printer using ping command.

```
$ ping 192.168.1.5
```

Check the printer details on the managed node(s) by running lpstat command as ansible ad-hoc command on the managed node(s).

```
$ ansible node1 -m command -a 'sudo lpstat -t'
```

You can attach the printer with your print server by running the following ansible ad-hoc command. (Mention the IP address and model number of the printer with port 9100)

```
$ ansible node1 -m command -a ‘sudo lpadmin -p HP-Deskjet-3540 -E -v socket://192.168.1.5:9100’
```

Verify the printer details again.

```
$ ansible node1 -m command -a 'sudo lpstat -t'
```

Set this printer as the default printer by running the below ansible ad-hoc command.

```
$ ansible node1 -m command -a 'sudo lpadmin -d HP-Deskjet-3540 -E -v socket://192.168.1.5:9100'
```

Verify the printer details again to see the default printer.

```
$ ansible node1 -m command -a 'sudo lpstat -t'
```

You can do this by using lpstat -v command as well.

```
$ ansible node1 -m command -a 'sudo lpstat -v'
```

Let’s create a file on the print server which we will print using the default printer added earlier.

```
$ ansible node1 -m shell -a 'echo “Nehra Classes Are Awesome” > /tmp/nehraclasses.txt'
```

Verify the file contents.

```
$ ansible node1 -m command -a 'cat /tmp/nehraclasses.txt'
```

Print the file by running the below ansible ad-hoc command.

```
$ ansible node1 -m command -a 'lp /tmp/nehraclasses.txt'
```

You can also use GUI for the above jobs if you want. Go to GUI, click on activities, then click on search box and write print there. After that you can clock on the printers' option to view/manage the printers. Remove the existing printer and attach your new printer in the machine. Set this as a default printer & check the status.

Now go to the /tmp directory and print the file created earlier using ansible ad-hoc commands.

```
$ cd /tmp
$ lp nehraclasses.txt
```

Configuring Printer Server Clients:

Go to the machine which you want to use as print server client or from where you want to print any file using your print server configured earlier and install the cups-ipptool packages.

```
$ sudo yum install cups-ipptool -y
```

Now restart the cups service.

```
$ sudo systemctl restart cups
```

Check the attached printer details.

```
$ lpstat -t
```

Add the printer using the IP address of your print server (not printer's) and use port 631 there.

```
$ lpadmin -p HP-Deskjet-3540 -E -v socket://192.168.229.129:631
```

Verify the attached printer details.

```
$ lpstat -t
```

Set the attached printer as default printer.

```
$ lpadmin -d HP-Deskjet-3540 -E -v socket://192.168.229.129:631
```

Verify the same using lpstat command.

```
$ lpstat -t
$ lpstat -v
```

Now, create a file which you want to print.

```
$ vim nehraclasses.txt
NEHRA CLASSES ARE AWESOME.
```

Print the file using lp command.

```
$ lp nehraclasses.txt
```

The CUPS print server is working perfectly.

Thank You