Jason Hodge

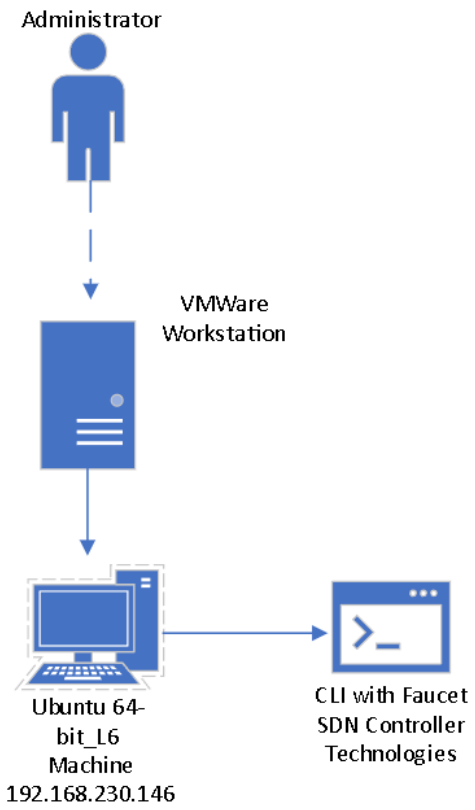
Lab 06 – Software Defined Networking with Faucet

April 26, 2024

Description:

The primary objective of this lab was to learn about and utilize Software Defined Networking (SDN) through Faucet SDN Controller and its built-in tools including Prometheus, Grafana, and Gauge. In this lab, these features were built out to demonstrate how a networking practitioner can create virtual networks with virtual hosts, virtual switches, and virtual controllers, as well as the ability to develop customized SDN applications in order to setup ACLs, VLANs, routing configurations, and conntrack for firewall implementation rules and source NAT (sNAT).

Topology:



This is an overview of the virtual machines built in this lab.

Key Syntax:

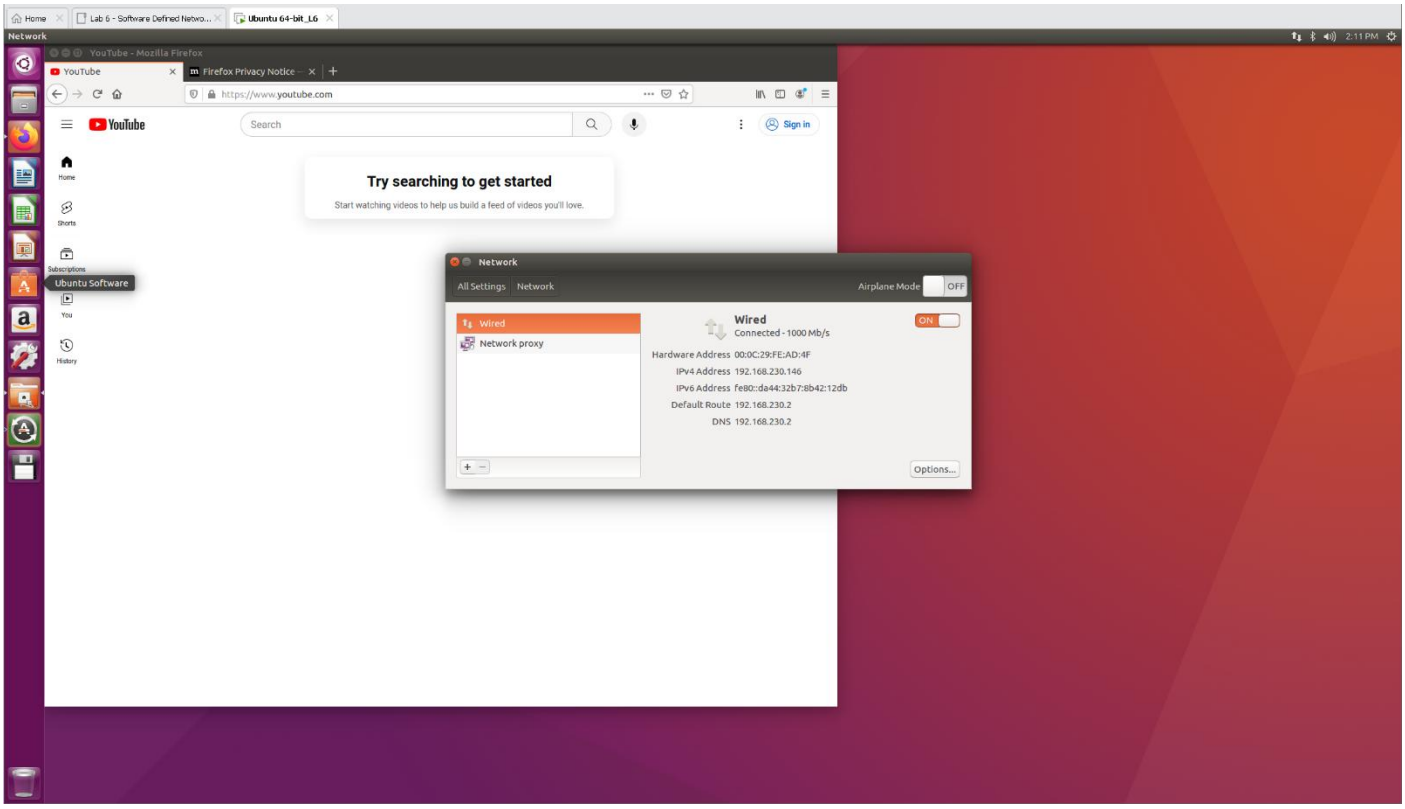
Linux Vi Editor Commands

- Press the ESC key for normal mode.
- Press i Key for insert mode.
- Press :q! keys to exit from the editor without saving a file.
- Press :wq! Keys to save the updated file and exit from the editor.
- Press :w test.txt to save the file as test.txt
- Press ESC key and ‘d’ to delete a line and ‘dd’ to delete multiple lines.

Other Commands are explained throughout as they are used.

Verification:

TASK ONE: Ubuntu 16.04 Installation



Here we can see Ubuntu 16.04.7 is up and running connected to the internet.

TASK TWO: Install the Faucet SDN Controller

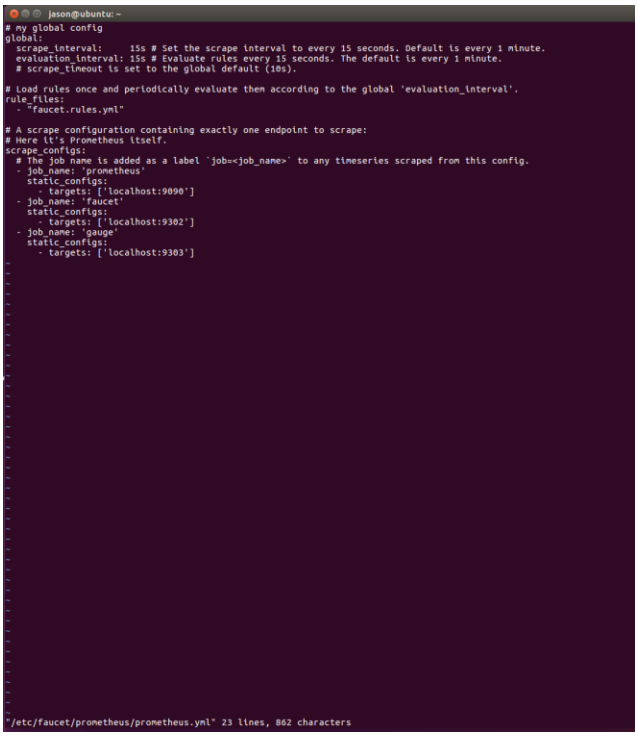
```
jason@ubuntu:~$ sudo apt-get install curl gnupg apt-transport-https lsb-release
[sudo] password for jason:
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (1.4.20-1ubuntu3.3).
lsb-release is already the newest version (9.20160110ubuntu0.2).
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  apt-transport-https
1 upgraded, 1 newly installed, 0 to remove and 131 not upgraded.
Need to get 165 kB of archives.
After this operation, 341 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apt-transport-https amd64 1.2.35 [26.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 curl amd64 7.47.0-1ubuntu2.19 [139 kB]
Fetched 165 kB in 0s (1,363 kB/s)
(Reading database ... 177268 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.2.35_amd64.deb ...
Unpacking apt-transport-https (1.2.35) over (1.2.32ubuntu0.2) ...
Selecting previously unselected package curl.
Preparing to unpack .../curl_7.47.0-1ubuntu2.19_amd64.deb ...
Unpacking curl (7.47.0-1ubuntu2.19) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up apt-transport-https (1.2.35) ...
Setting up curl (7.47.0-1ubuntu2.19) ...
jason@ubuntu:~$ sudo mkdir -p /etc/apt/keyrings/
jason@ubuntu:~$ curl -1sLf https://packagecloud.io/faucetsdn/faucet/gpgkey | sudo gpg --dearmor -o /etc/apt/keyrings/faucet.gpg
File `/etc/apt/keyrings/faucet.gpg' exists. Overwrite? (y/N) y
jason@ubuntu:~$ echo "deb [signed-by=/etc/apt/keyrings/faucet.gpg] https://packagecloud.io/faucetsdn/faucet/$(lsb_release -si | awk '{print tolower($0)}')/ $(lsb_release -sc) main" |
 sudo tee /etc/apt/sources.list.d/faucet.list
deb [signed-by=/etc/apt/keyrings/faucet.gpg] https://packagecloud.io/faucetsdn/faucet/ubuntu/ xenial main
jason@ubuntu:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Get:5 https://packagecloud.io/faucetsdn/faucet/ubuntu xenial InRelease [25.1 kB]
Fetched 25.1 kB in 0s (28.6 kB/s)
Reading package lists... Done
jason@ubuntu:~$
```

Here I installed the faucet repository and its dependencies to my machine.

```
jason@ubuntu:~$ sudo apt-get install faucet-all-in-one
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  daemon docutils-common faucet gauge grafana ieee-data javascript-common libexpat1-dev libjs-bootstrap libjs-d3 libjs-eonasdan-bootstrap-datetimepicker libjs-jquery
  libjs-jquery-hotkeys libjs-moment libjs-mustache libjs-rickshaw libpython3-dev libpython3.5 libpython3.5-dev libpython3.5-minimal libpython3.5-stdlib prometheus
  prometheus-node-exporter python-babel-localedata python-pip-whl python3-babel python3-beka python3-certifi python3-chardet python3-chewie python3-crypto python3-dateutil
  python3-debtcollector python3-decorator python3-dev python3-dnspython python3-docutils python3-ecdsa python3-eventlet python3-faucet python3-greenlet python3-influxdb
  python3-monotonic python3-msgpack python3-netaddr python3-networkx python3-openvswitch python3-oslo.config python3-paramiko python3-pbr python3-pip python3-prometheus-client
  python3-pygments python3-pytricia python3-repoze.lru python3-requests python3-roman python3-routes python3-ryu python3-setuptools python3-sortedcontainers python3-stevedore
  python3-tinyrpc python3-transitions python3-tz python3-urllib3 python3-webob python3-wheel python3-wrapt python3-yaml python3.5 python3.5-dev python3.5-minimal
Suggested packages:
  python-faucet-doc apache2 | lighttpd | httpd python3-crypto-dbg python-crypto-doc python-debtcollector-doc texlive-latex-recommended texlive-latex-base texlive-lang-french
  fonts-linuxlibertine | ttf-linux-libertine docutils-doc python-eventlet-doc python-greenlet-doc python-greenlet-dev python3-greenlet-dbg ipython3 python-netaddr-docs
  python-networkx-doc ttf-bitstream-vera python3-idna python3-openssl python3-socks python3-paste python-ryu-doc python-setuptools-doc python-sortedcontainers-doc
  python-tinyrpc-doc python-webob-doc python3.5-venv python3.5-doc binfmt-support
The following NEW packages will be installed:
  daemon docutils-common faucet faucet-all-in-one gauge grafana ieee-data javascript-common libexpat1-dev libjs-bootstrap libjs-d3 libjs-eonasdan-bootstrap-datetimepicker
  libjs-jquery libjs-jquery-hotkeys libjs-moment libjs-mustache libjs-rickshaw libpython3-dev libpython3.5-dev prometheus prometheus-node-exporter python-babel-localedata
  python-pip-whl python3-babel python3-beka python3-certifi python3-chewie python3-crypto python3-dateutil python3-debtcollector python3-decorator python3-dev python3-dnspython
  python3-docutils python3-ecdsa python3-eventlet python3-faucet python3-greenlet python3-influxdb python3-monotonic python3-msgpack python3-netaddr python3-networkx
  python3-openvswitch python3-oslo.config python3-paramiko python3-pbr python3-pip python3-prometheus-client python3-pygments python3-pytricia python3-repoze.lru python3-roman
  python3-routes python3-ryu python3-setuptools python3-sortedcontainers python3-stevedore python3-tinyrpc python3-transitions python3-tz python3-webob python3-wheel python3-wrapt
  python3-yaml python3.5-dev
The following packages will be upgraded:
  libpython3.5 libpython3.5-minimal libpython3.5-stdlib python3-chardet python3-requests python3-urllib3 python3.5 python3.5-minimal
8 upgraded, 66 newly installed, 0 to remove and 123 not upgraded.
Need to get 112 MB/117 MB of archives.
After this operation, 333 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-wrapt amd64 1.8.0-5build2 [25.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 libjs-jquery-hotkeys all 0~20130707+git2d51e3a9+dfsg-2ubuntu1 [11.4 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 daemon amd64 0.6.4-1 [98.2 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 docutils-common all 0.12+dfsg-1 [141 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-dateutil all 2.4.2-1 [39.1 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-certifi all 2015.11.20.1-2 [173 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-tz all 2014.10~dfsg1-0ubuntu2 [24.6 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-msgpack amd64 0.4.6-1build1 [57.3 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-greenlet amd64 0.4.9-2fakesync1 [15.8 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 ieee-data all 20150531.1 [830 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-netaddr all 0.7.18-1 [173 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-sortedcontainers all 1.4.4-1 [25.2 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python-babel-localedata all 1.3+dfsg.1-6 [1,931 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-babel all 1.3+dfsg.1-6 [67.5 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-debtcollector all 1.3.0-2 [12.9 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-roman all 2.0.0-2 [8,140 B]
Get:17 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-docutils all 0.12+dfsg-1 [346 kB]
Get:18 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-stevedore all 1.12.0-1 [17.8 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-oslo.config all 1:3.9.0-3 [64.7 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 python3-crypto amd64 2.6.1-6ubuntu0.16.04.3 [246 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 python3-ecdsa all 0.13-2ubuntu0.16.04.1 [36.3 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 python3-paramiko all 1.16.0-1ubuntu0.2 [110 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 python-pip-whl all 8.1.1-2ubuntu0.6 [1,112 kB]
Get:24 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 python3-pip all 8.1.1-2ubuntu0.6 [109 kB]
Get:25 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-repoze.lru all 0.6-6 [11.2 kB]
Get:26 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 python3-routes all 2.2-1ubuntu2 [28.0 kB]
Get:27 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python3-webob all 1.5.1-1 [61.1 kB]
```

```
Setting up prometheus (2.2.1+ds-2) ...
Setting up grafana (7.3.4) ...
Adding system user `grafana' (UID 122) ...
Adding new user `grafana' (UID 122) with group `grafana' ...
Not creating home directory `/usr/share/grafana'.
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
 sudo /bin/systemctl daemon-reload
 sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
 sudo /bin/systemctl start grafana-server
Setting up javascript-common (11) ...
Setting up libexpat1-dev:amd64 (2.1.0-7ubuntu0.16.04.5) ...
Setting up libpython3.5-dev:amd64 (3.5.2-2ubuntu0~16.04.13) ...
Setting up libpython3-dev:amd64 (3.5.1-3) ...
Setting up python3.5-dev (3.5.2-2ubuntu0~16.04.13) ...
Setting up python3-dev (3.5.1-3) ...
Setting up python3-pygments (2.1+dfsg-1ubuntu0.2) ...
Setting up python3-setuptools (20.7.0-1) ...
Setting up python3-wheel (0.29.0-1) ...
Setting up prometheus-node-exporter (0.11.0+ds-1) ...
Processing triggers for sgml-base (1.26+nmu4ubuntu1) ...
Setting up python3-docutils (0.12+dfsg-1) ...
update-alternatives: using /usr/share/docutils/scripts/python3/rst-buildhtml to provide /usr/bin/rst-buildhtml (rst-buildhtml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2html to provide /usr/bin/rst2html (rst2html) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2latex to provide /usr/bin/rst2latex (rst2latex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2man to provide /usr/bin/rst2man (rst2man) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt to provide /usr/bin/rst2odt (rst2odt) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2odt_prepstyles to provide /usr/bin/rst2odt_prepstyles (rst2odt_prepstyles) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2pseudoxml to provide /usr/bin/rst2pseudoxml (rst2pseudoxml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2s5 to provide /usr/bin/rst2s5 (rst2s5) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xetex to provide /usr/bin/rst2xetex (rst2xetex) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rst2xml to provide /usr/bin/rst2xml (rst2xml) in auto mode
update-alternatives: using /usr/share/docutils/scripts/python3/rstpep2html to provide /usr/bin/rstpep2html (rstpep2html) in auto mode
Setting up python3-oslo.config (1:3.9.0-3) ...
update-alternatives: using /usr/bin/python3-oslo-config-generator to provide /usr/bin/oslo-config-generator (oslo-config-generator) in auto mode
Setting up python3-ryu (4.32-1) ...
Setting up python3-faucet (1.9.53) ...
Setting up faucet (1.9.53) ...
Adding group `faucet' (GID 131) ...
Done.
Adding system user `faucet' (UID 123) ...
Adding new user `faucet' (UID 123) with group `faucet' ...
Not creating home directory `/etc/faucet'.
Setting up gauge (1.9.53) ...
Setting up faucet-all-in-one (1.9.53) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
jason@ubuntu:~$
```

Here I installed the required packages using faucet-all-in-one with all the proper dependencies.



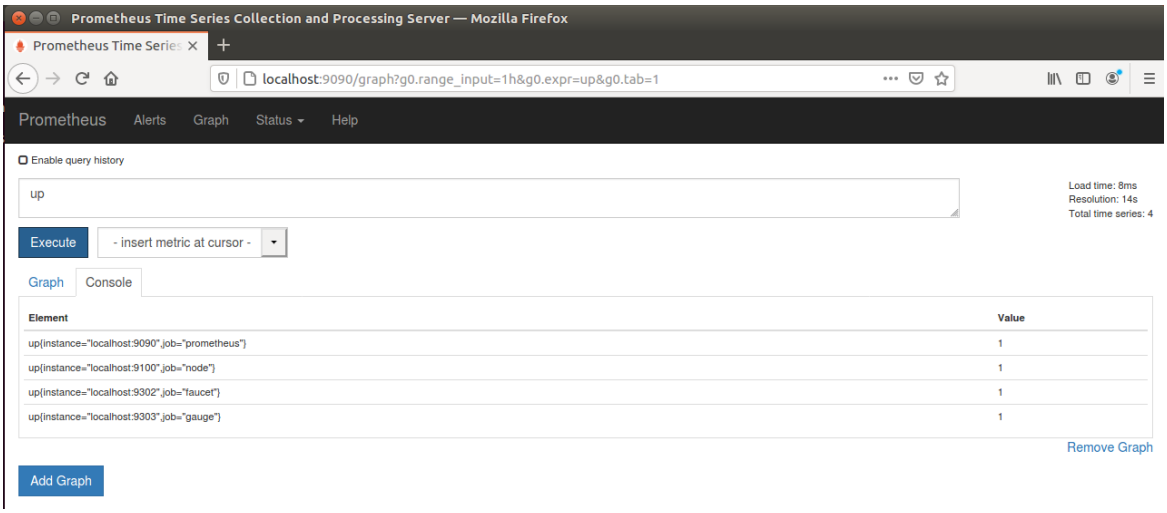
Here we see the Prometheus.yml file contents.

```
jason@ubuntu:~$ sudo vi /etc/faucet/prometheus/prometheus.yml
[sudo] password for jason:
jason@ubuntu:~$
```

```
jason@ubuntu: ~
# Set the command-line arguments to pass to the server.
ARGS="--config.file=/etc/faucet/prometheus/prometheus.yml"

# Prometheus supports the following options:
#  --config.file="/etc/prometheus/prometheus.yml"
#                            Prometheus configuration file path.
#  --web.listen-address="0.0.0.0:9090"
#                            Address to listen on for UI, API, and telemetry.
#  --web.read-timeout=5m     Maximum duration before timing out read of the
#                            request, and closing idle connections.
#  --web.max-connections=512  Maximum number of simultaneous connections.
#  --web.external-url=<URL>   The URL under which Prometheus is externally
#                            reachable (for example, if Prometheus is served
#                            via a reverse proxy). Used for generating
#                            relative and absolute links back to Prometheus
#                            itself. If the URL has a path portion, it will
#                            be used to prefix all HTTP endpoints served by
#                            Prometheus. If omitted, relevant URL components
#                            will be derived automatically.
#  --web.route-prefix=<path>  Prefix for the internal routes of web endpoints.
#                            Defaults to path of --web.external-url.
#  --web.local-assets="/usr/share/prometheus/web/"
#                            Path to static asset/templates directory.
#  --web.user-assets=<path>   Path to static asset directory, available at
#                            /user.
#  --web.enable-lifecycle     Enable shutdown and reload via HTTP request.
#  --web.enable-admin-api     Enables API endpoints for admin control actions.
#  --web.console.templates="/etc/prometheus/consoles"
#                            Path to the console template directory,
#                            available at /consoles.
#  --web.console.libraries="/etc/prometheus/console_libraries"
#                            Path to the console library directory.
#  --storage.tsdb.path="/var/lib/prometheus/metrics2/"
#                            Base path for metrics storage.
#  --storage.tsdb.min-block-duration=2h
#                            Minimum duration of a data block before being
#                            persisted.
#  --storage.tsdb.max-block-duration=<duration>
#                            Maximum duration compacted blocks may span.
#                            (Defaults to 10% of the retention period)
#  --storage.tsdb.retention=15d
#                            How long to retain samples in the storage.
#  --storage.tsdb.use-lockfile
#                            Create a lockfile in data directory.
#  --alertmanager.notification-queue-capacity=10000
#                            The capacity of the queue for pending alert
#                            manager notifications.
#  --alertmanager.timeout=10s
#                            Timeout for sending alerts to Alertmanager.
#  --query.lookback-delta=5m  The delta difference allowed for retrieving
#                            metrics during expression evaluations.
#  --query.timeout=2m         Maximum time a query may take before being
#                            aborted.
#  --query.max-concurrency=20
#                            Maximum number of queries executed concurrently.
#  --log.level=info           Only log messages with the given severity or
#                            above. One of: [debug, info, warn, error]
~
~
~
~
~
~
~
~
~
~
~
~
```

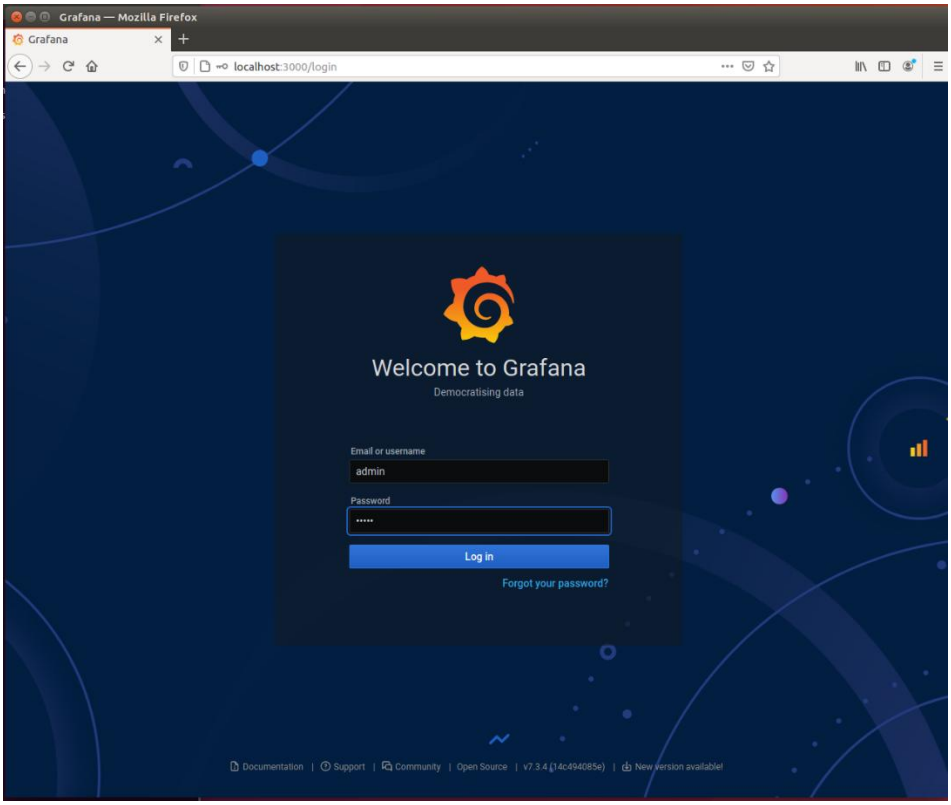
Here I updated the ‘ARGS’ field contents with the proper configuration path.



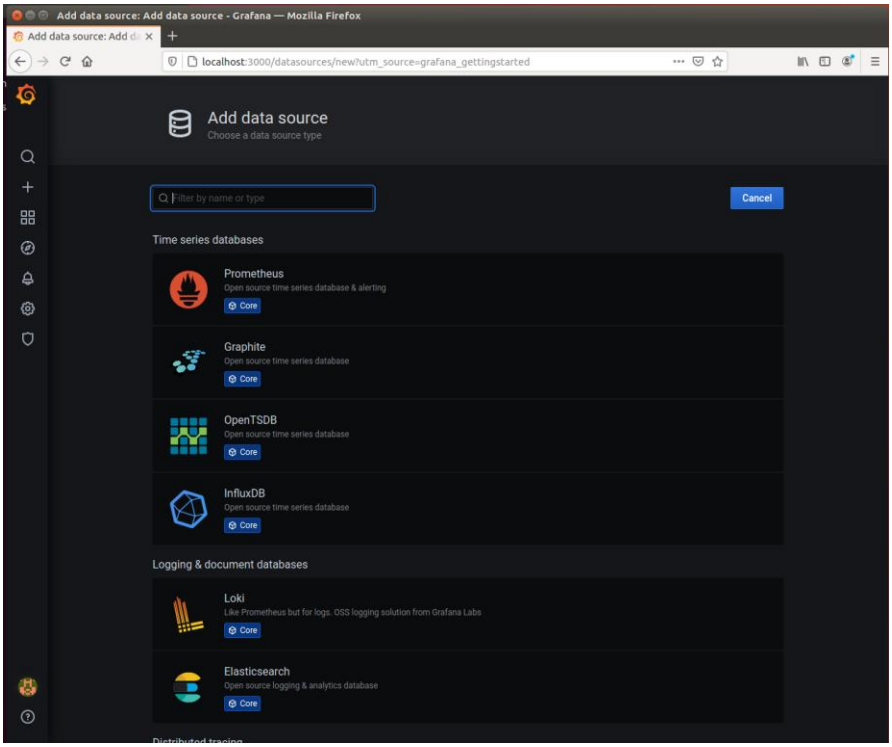
Here we can see Prometheus is up and running at localhost:9090

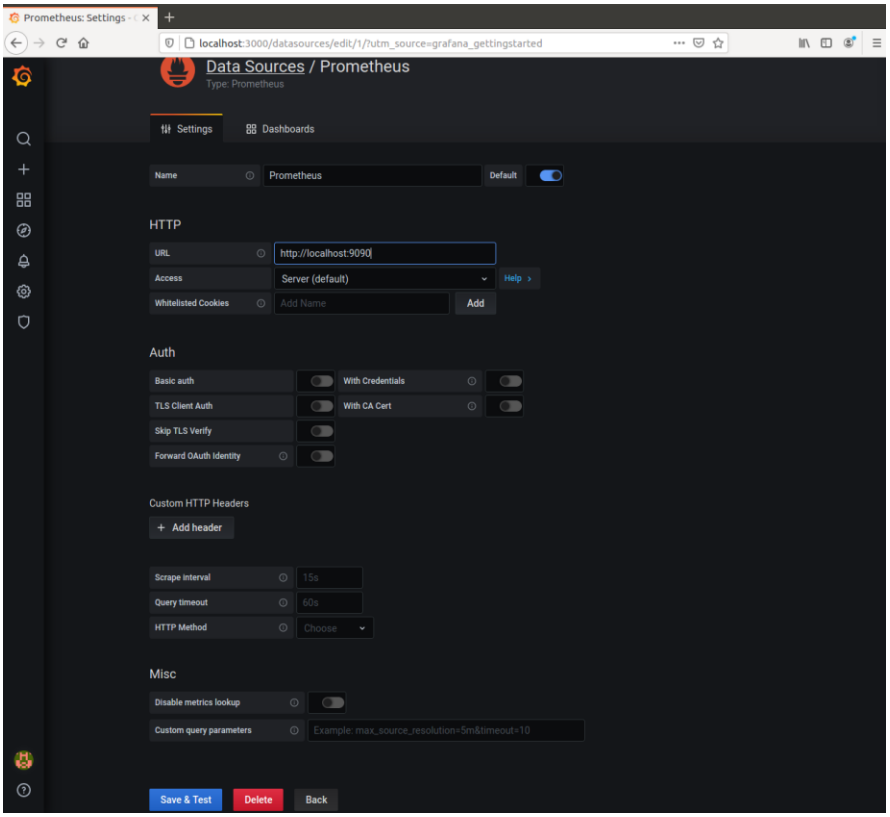
```
jason@ubuntu:~$ sudo systemctl daemon-reload
jason@ubuntu:~$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service to /usr/lib/systemd/system/grafana
-server.service.
jason@ubuntu:~$ sudo systemctl start grafana-server
```

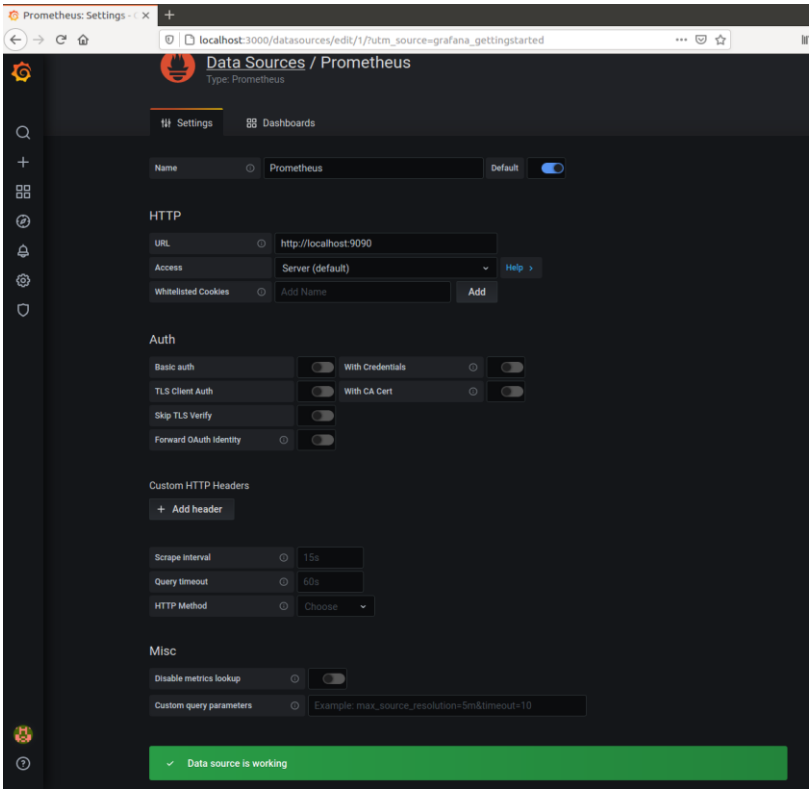
Firefox Web Browser



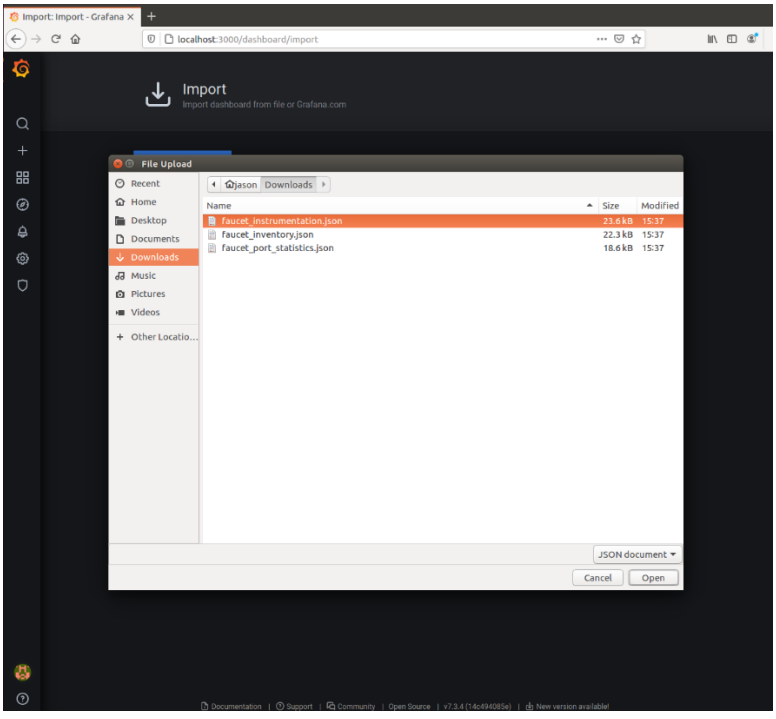
Here I configured Grafana and we can see it is up and running at localhost:3000.


Add data source: Add data source - Grafana — Mozilla Firefox
localhost:3000/datasources/new?utm_source=grafana_gettingstarted
Add data source
Choose a data source type
Filter by name or type
Cancel
Time series databases
Prometheus
Open source time series database & alerting
Core
Graphite
Open source time series database
Core
OpenTSDB
Open source time series database
Core
InfluxDB
Open source time series database
Core
Logging & document databases
Loki
Like Prometheus but for logs. OSS logging solution from Grafana Labs
Core
Elasticsearch
Open source logging & analytics database
Core

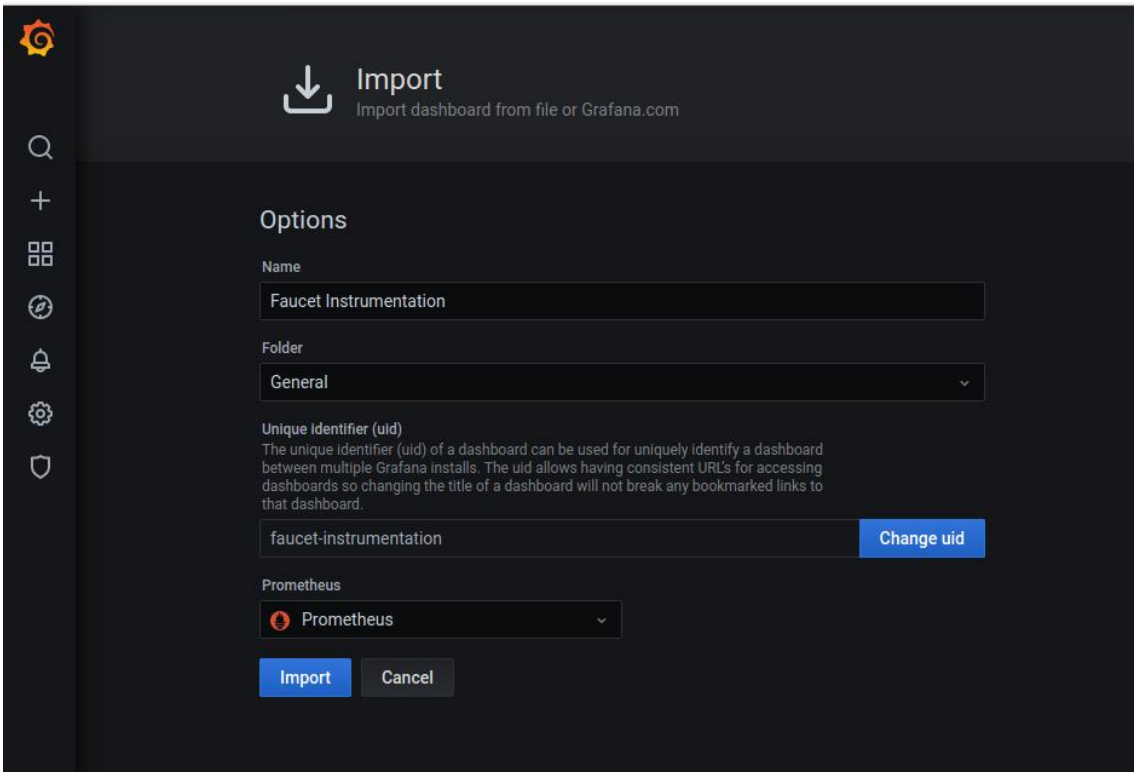

Prometheus: Settings
localhost:3000/datasources/edit/1/?utm_source=grafana_gettingstarted
Data Sources / Prometheus
Type: Prometheus
Settings
Dashboards
Name
Prometheus
Default
HTTP
URL
http://localhost:9090
Access
Server (default)
Help
Whitelisted Cookies
Add Name
Add
Auth
Basic auth
With Credentials
TLS Client Auth
With CA Cert
Skip TLS Verify
Forward OAuth Identity
Custom HTTP Headers
Add header
Scrape interval
15s
Query timeout
60s
HTTP Method
Choose
Misc
Disable metrics lookup
Custom query parameters
Example: max_source_resolution=5m&timeout=10
Save & Test
Delete
Back

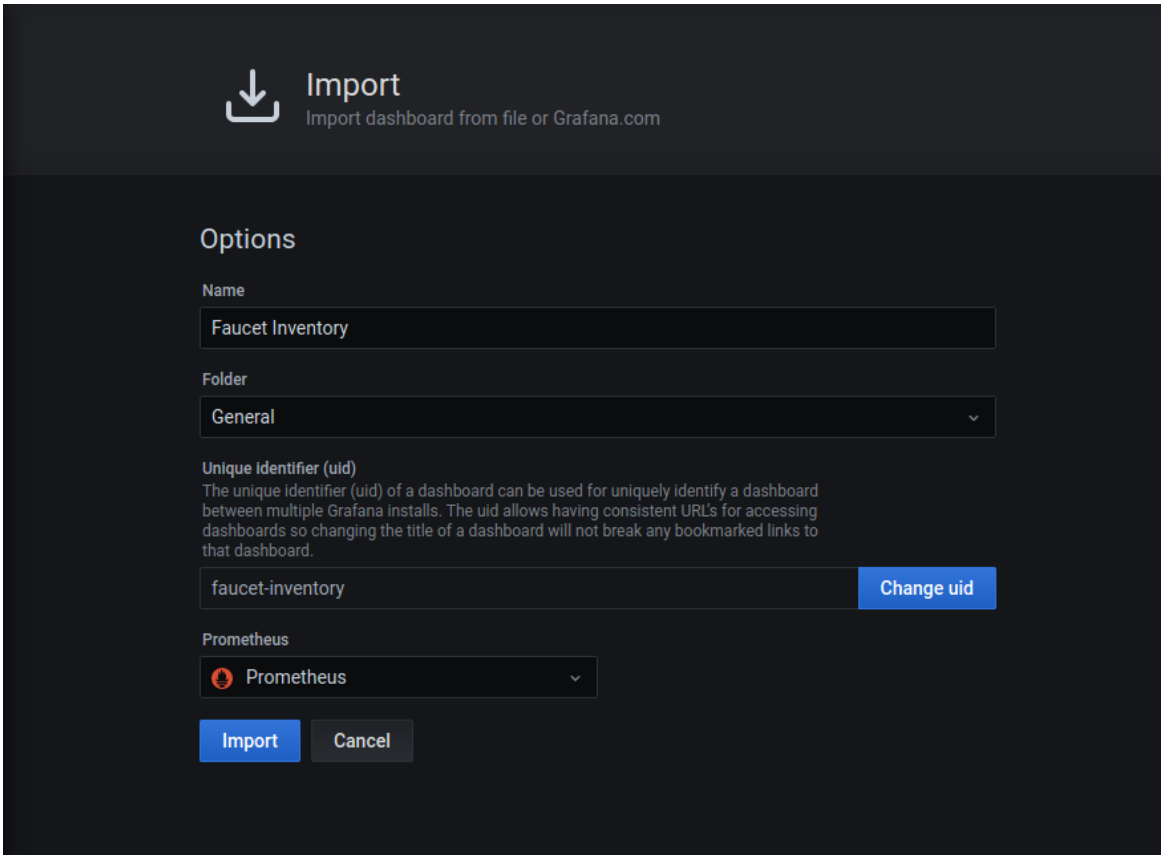


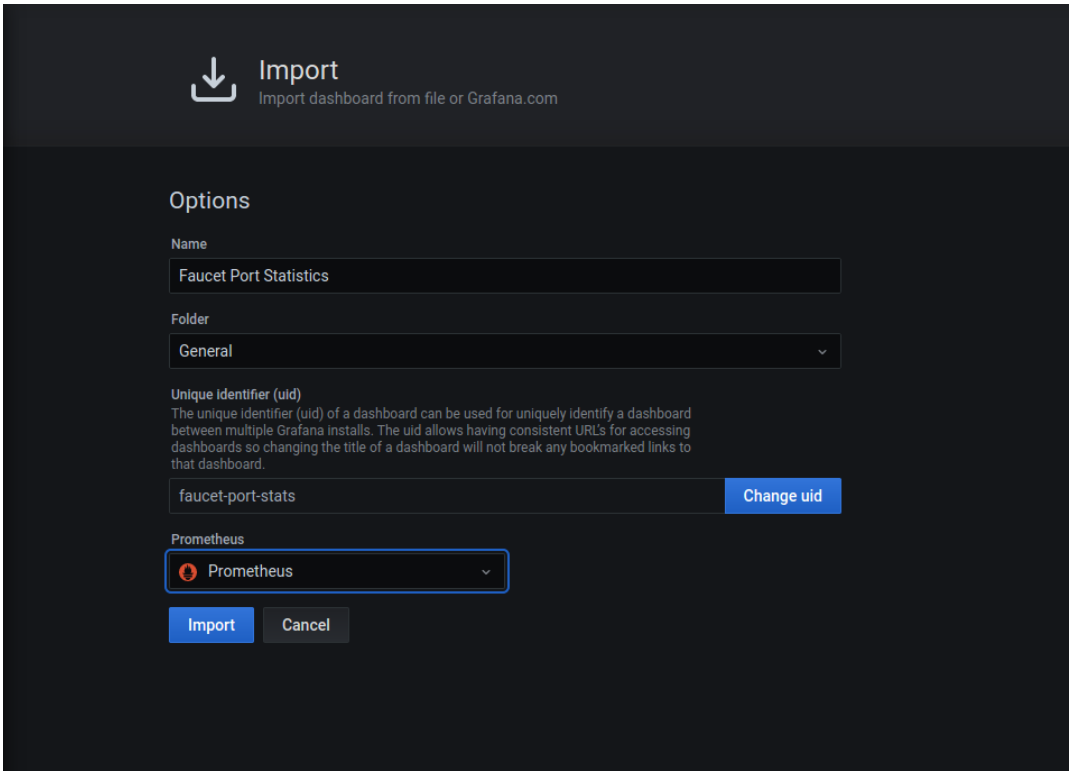
Here I selected the data source Prometheus and walked through the steps following by setting the URL path to Localhost:9090 and saving and testing the update.

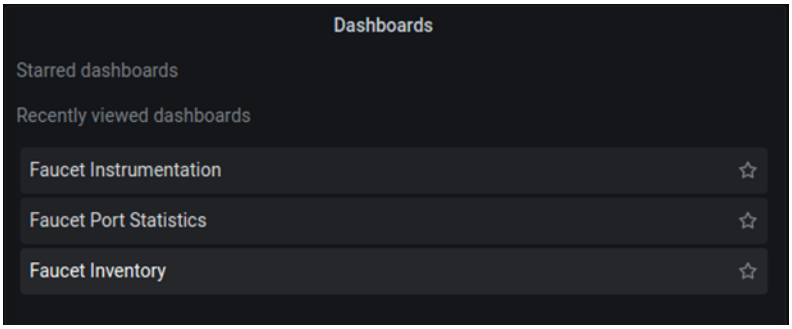


Here we can see the premade Grafana dashboards have been downloaded to the VM.


Import
Import dashboard from file or Grafana.com
Options
Name
Faucet Instrumentation
Folder
General
Unique identifier (uid)
The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
faucet-instrumentation
Change uid
Prometheus
Prometheus
Import
Cancel

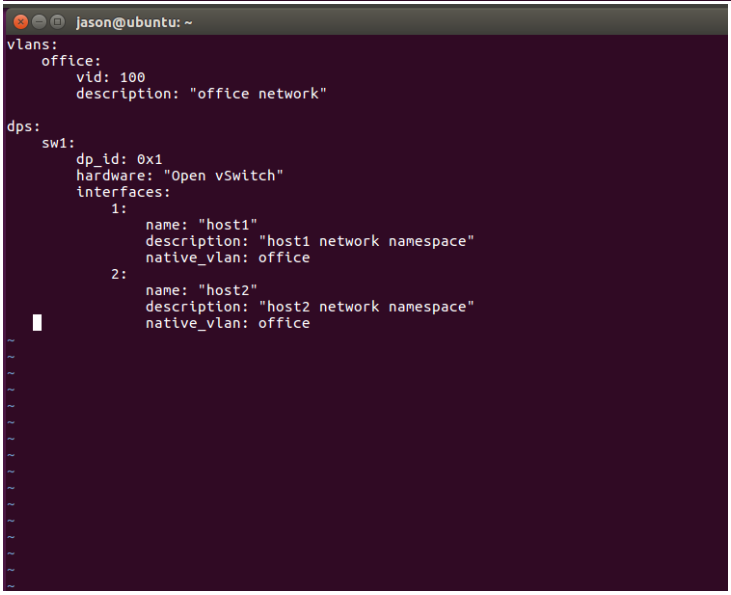

Import
Import dashboard from file or Grafana.com
Options
Name
Faucet Inventory
Folder
General
Unique identifier (uid)
The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
faucet-inventory
Change uid
Prometheus
Prometheus
Import
Cancel





These last snapshots are importing the dashboard JSON files to Grafana.

```
jason@ubuntu:~$ sudo vi /etc/faucet/faucet.yaml
[sudo] password for jason:
```



Here we can see the current contents of the faucet.yaml file.

```
jason@ubuntu:~$ check_faucet_config /etc/faucet/faucet.yaml
['{\n'
 '    "advertise_interval": 30,\n'
 '    "arp_neighbor_timeout": 30,\n'
 '    "cache_update_guard_time": 150,\n'
 '    "combinatorial_port_flood": false,\n'
 '    "cookie": 1524372928,\n'
 '    "description": "sw1",\n'
 '    "dot1x": {},\n'
 '    "dp_id": 1,\n'
 '    "drop_broadcast_source_address": true,\n'
 '    "drop_spoofed_faucet_mac": true,\n'
 '    "egress_pipeline": false,\n'
 '    "fast_advertise_interval": 5,\n'
 '    "faucet_dp_mac": "0e:00:00:00:00:01",\n'
 '    "global_vlan": 0,\n'
 '    "group_table": false,\n'
 '    "hardware": "Open vSwitch",\n'
 '    "high_priority": 9001,\n'
 '    "highest_priority": 9099,\n'
 '    "idle_dst": true,\n'
 '    "ignore_learn_ins": 10,\n'
 '    "interface_ranges": {},\n'
 '    "interfaces": {\n'
 '        "1": {\n'
 '            "description": "host1 network namespace",\n'
 '            "name": "host1",\n'
 '            "native_vlan": "office"\n'
 '        },\n'
 '        "2": {\n'
 '            "description": "host2 network namespace",\n'
 '            "name": "host2",\n'
 '            "native_vlan": "office"\n'
 '        }\n'
 '    },\n'
 '    "lacp_timeout": 30,\n'
 '    "learn_ban_timeout": 51,\n'
 '    "learn_jitter": 51,\n'
 '    "lldp_beacon": {},\n'
 '    "low_priority": 9000,\n'
 '    "lowest_priority": 0,\n'
 '    "max_host_fib_retry_count": 10,\n'
 '    "max_hosts_per_resolve_cycle": 5,\n'
 '    "max_resolve_backoff_time": 64,\n'
 '    "max_wildcard_table_size": 1280,\n'
 '    "metrics_rate_limit_sec": 0,\n'
 '    "min_wildcard_table_size": 32,\n'
 '    "multi_out": true,\n'
 '    "nd_neighbor_timeout": 30,\n'
 '    "port_table_scale_factor": null,\n'
 '    "priority_offset": 0,\n'
 '    "proactive_learn_v4": true,\n'
 '    "proactive_learn_v6": true,\n'
 '    "stack": {\n'
 '        "down_time_multiple": 3,\n'
 '        "min_lacp_health": null,\n'
 '        "min_stack_health": null,\n'
 '        "route_learning": false\n'
 '    },\n'
 '    "strict_packet_in_cookie": true,\n'
 '    "table_sizes": {},\n'
 '    "timeout": 300,\n'
 '    "use_classification": false,\n'
 '    "use_idle_timeout": false\n'
 '}']
jason@ubuntu:~$
```

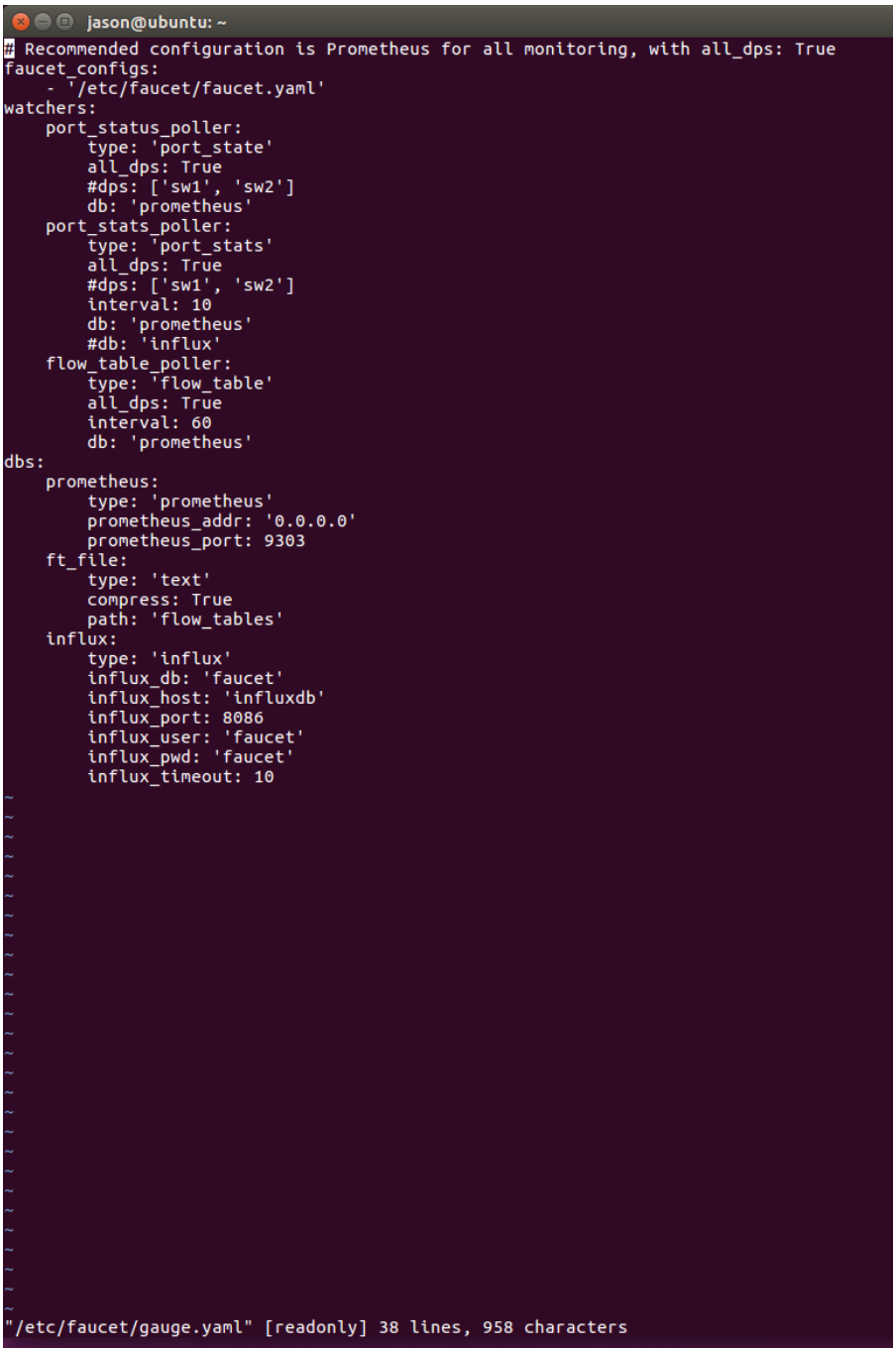
System Settings

```
jason@ubuntu:~$ sudo systemctl reload faucet
[sudo] password for jason:
jason@ubuntu:~$ less /var/log/faucet/faucet.log
```

jason@ubuntu: ~

```
Apr 22 09:12:34 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 22 09:12:34 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 22 09:12:34 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 22 09:12:34 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 22 09:12:34 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 22 09:12:34 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 14:26:17 faucet INFO     version 1.9.53
Apr 25 14:26:17 faucet INFO     Reloading configuration
Apr 25 14:26:17 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 14:26:17 faucet.config INFO     including file: /etc/faucet/acls.yaml
Apr 25 14:26:17 faucet INFO     Add new datapath DPID 1 (0x1)
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['10.0.200.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['10.0.100.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 14:26:17 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     version 1.9.53
Apr 25 20:48:28 faucet INFO     Reloading configuration
Apr 25 20:48:28 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 20:48:28 faucet.config INFO     including file: /etc/faucet/acls.yaml
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 1 (0x1)
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 21:20:02 faucet INFO     Reloading configuration
Apr 25 21:20:02 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 21:20:02 faucet INFO     Reconfiguring existing datapath DPID 1 (0x1)
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 DP routers config changed - requires cold start
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 table IDs changed, old {0, 1, 2, 3, 4, 5, 6, 7, 8} new {0, 1, 2
, 3}
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 cold starting
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 forcing DP reconnection to ensure ports are synchronized
Apr 25 21:20:02 faucet.valve ERROR    DPID 1 (0x1) sw1 send_flow_msgs: DP not up
Apr 25 21:20:02 faucet INFO     Deleting de-configured DPID 2 (0x2)
(END)
```

As we can see here, this command was used to verify faucet has correctly interpreted the configuration. Also, after every change made to the faucet.yaml file we need to reload faucet for the changes to take place. We are also taking a look at the logs to see what is going on behind the scenes.




```
jason@ubuntu:~$ vi /etc/faucet/gauge.yaml
jason@ubuntu:~$ sudo systemctl restart gauge
```

Here we can see the gauge.yaml file, which we did not need to change anything with.

```
jason@ubuntu:~$ sudo apt-get install openvswitch-switch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openvswitch-common python-six
The following NEW packages will be installed:
  openvswitch-common openvswitch-switch python-six
0 upgraded, 3 newly installed, 0 to remove and 40 not upgraded.
Need to get 1,827 kB of archives.
After this operation, 8,100 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 python-six all 1.10.0-3 [10.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openvswitch-common amd64 2.5.9-0ubuntu0.16.04.3 [6
57 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 openvswitch-switch amd64 2.5.9-0ubuntu0.16.04.3 [1
,158 kB]
Fetched 1,827 kB in 0s (5,881 kB/s)
Selecting previously unselected package python-six.
(Reading database ... 221300 files and directories currently installed.)
Preparing to unpack .../python-six_1.10.0-3_all.deb ...
Unpacking python-six (1.10.0-3) ...
Selecting previously unselected package openvswitch-common.
Preparing to unpack .../openvswitch-common_2.5.9-0ubuntu0.16.04.3_amd64.deb ...
Unpacking openvswitch-common (2.5.9-0ubuntu0.16.04.3) ...
Selecting previously unselected package openvswitch-switch.
Preparing to unpack .../openvswitch-switch_2.5.9-0ubuntu0.16.04.3_amd64.deb ...
Unpacking openvswitch-switch (2.5.9-0ubuntu0.16.04.3) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
Setting up python-six (1.10.0-3) ...
Setting up openvswitch-common (2.5.9-0ubuntu0.16.04.3) ...
Setting up openvswitch-switch (2.5.9-0ubuntu0.16.04.3) ...
update-alternatives: using /usr/lib/openvswitch-switch/ovs-vswitchd to provide /usr/sbin/ovs-vswitchd (ovs-vswitchd) i
n auto mode
openvswitch-nonetwork.service is a disabled or a static unit, not starting it.
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
```

Here I installed Open vSwitch for OpenFlow support in connecting a datapath.

```
jason@ubuntu:~$ # Run command inside network namespace
jason@ubuntu:~$ as_ns () {
>       NAME=$1
>       NETNS=faucet-${NAME}
>       shift
>       sudo ip netns exec ${NETNS} $@
> }
jason@ubuntu:~$ # Create network namespace
jason@ubuntu:~$ create_ns () {
>       NAME=$1
>       IP=$2
>       NETNS=faucet-${NAME}
>       sudo ip netns add ${NETNS}
>       sudo ip link add dev veth-${NAME} type veth peer name veth0 netns ${NETNS}
>       sudo ip link set dev veth-${NAME} up
>       as_ns ${NAME} ip link set dev lo up
>       [ -n "${IP}" ] && as_ns ${NAME} ip addr add dev veth0 ${IP}
>       as_ns ${NAME} ip link set dev veth0 up
> }
jason@ubuntu:~$ create_ns host1 192.168.0.1/24
jason@ubuntu:~$ ip netns list
faucet-host1 (id: 0)
jason@ubuntu:~$ create_ns host2 192.168.0.2/24
jason@ubuntu:~$
```

Here I defined some key functions necessary for commands to be run. I also created 2 virtual hosts.

```
jason@ubuntu:~$ sudo ovs-vsctl add-br br0 \
> -- set bridge br0 other-config:datapath-id=0000000000000001 \
> -- set bridge br0 other-config:disable-in-band=true \
> -- set bridge br0 fail_mode=secure \
> -- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \
> -- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \
> -- set-controller br0 tcp:127.0.0.1:6653 tcp:127.0.0.1:6654
jason@ubuntu:~$ sudo ovs-vsctl show
6196035a-18e3-4e12-b5e1-cb4f55dcc698
    Bridge "br0"
        Controller "tcp:127.0.0.1:6653"
            is_connected: true
        Controller "tcp:127.0.0.1:6654"
            is_connected: true
        fail_mode: secure
        Port "veth-host1"
            Interface "veth-host1"
        Port "veth-host2"
            Interface "veth-host2"
        Port "br0"
            Interface "br0"
                type: internal
    ovs_version: "2.5.9"
jason@ubuntu:~$
```

Here I configured a bridge and added two ports to it.

```
jason@ubuntu:~$ less /var/log/faucet/faucet.log
```

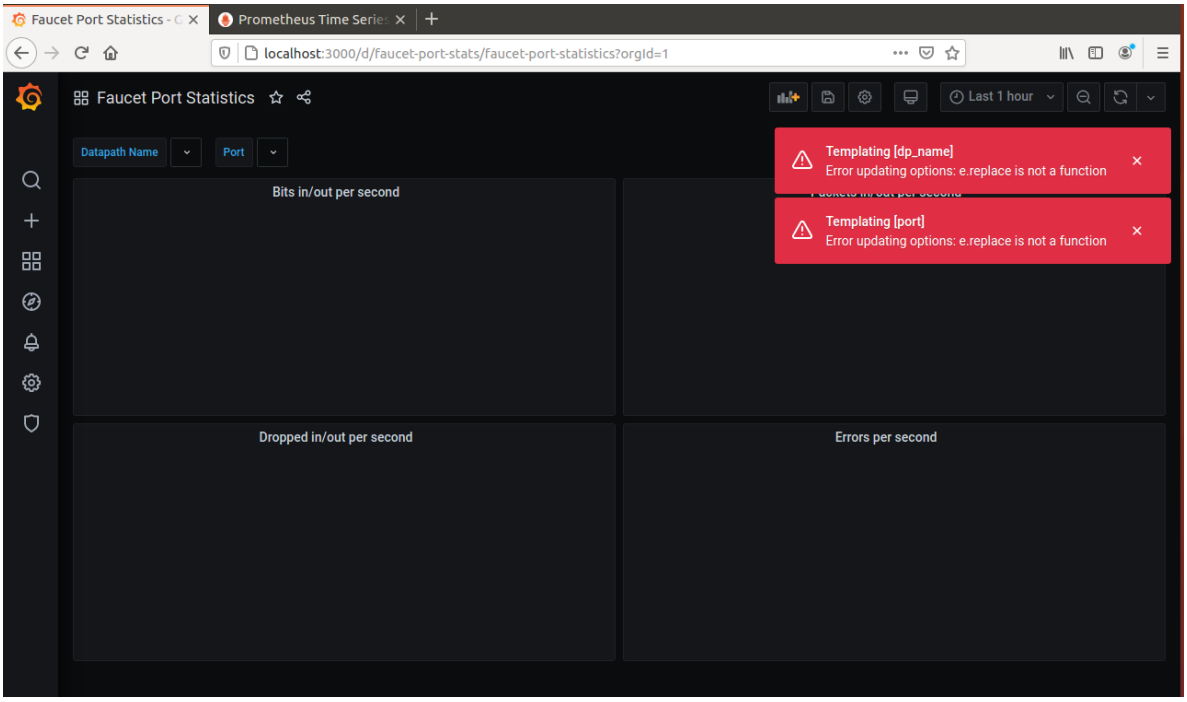
```
jason@ubuntu: ~
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['10.0.100.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 14:26:17 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     version 1.9.53
Apr 25 20:48:28 faucet INFO     Reloading configuration
Apr 25 20:48:28 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 20:48:28 faucet.config INFO     including file: /etc/faucet/acls.yaml
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 1 (0x1)
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 21:20:02 faucet INFO     Reloading configuration
Apr 25 21:20:02 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 21:20:02 faucet INFO     Reconfiguring existing datapath DPID 1 (0x1)
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 DP routers config changed - requires cold start
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 table IDs changed, old {0, 1, 2, 3, 4, 5, 6, 7, 8} new {0, 1, 2
, 3}
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 cold starting
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 forcing DP reconnection to ensure ports are synchronized
Apr 25 21:20:02 faucet.valve ERROR    DPID 1 (0x1) sw1 send_flow_msgs: DP not up
Apr 25 21:20:02 faucet INFO     Deleting de-configured DPID 2 (0x2)
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Cold start configuring DP
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Port 1 (host1 network namespace) configured
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Port 2 (host2 network namespace) configured
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Configuring VLAN office vid:100 untagged: Port 1,Port 2
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 table ID 0 table config match_types: (('eth_dst', True), ('eth_
type', False), ('in_port', False), ('vlan_vid', False)) name: vlan next_tables: ['eth_src'] output: True set_fields: (
'vlan_vid',) size: 32 vlan_port_scale: 1.5
table ID 1 table config match_types: (('eth_dst', True), ('eth_src', False), ('eth_type', False), ('in_port', False),
('vlan_vid', False)) miss_goto: eth_dst name: eth_src next_tables: ['eth_dst', 'flood'] output: True set_fields: ('vla
n_vid', 'eth_dst') size: 32 table_id: 1 vlan_port_scale: 4.1
table ID 2 table config exact_match: True match_types: (('eth_dst', False), ('vlan_vid', False)) miss_goto: flood name
: eth_dst output: True size: 32 table_id: 2 vlan_port_scale: 4.1
table ID 3 table config match_types: (('eth_dst', True), ('in_port', False), ('vlan_vid', False)) name: flood output:
True size: 32 table_id: 3 vlan_port_scale: 7.0
Apr 25 21:37:16 faucet.valve INFO     DPID 1 (0x1) sw1 L2 learned on Port 2 a2:89:fd:c9:01:3f (L2 type 0x86dd, L2 dst
33:33:00:00:00:02, L3 src fe80::a089:fdff:fec9:13f, L3 dst ff02::2) Port 2 VLAN 100 (1 hosts total)
Apr 25 21:39:06 faucet.valve INFO     DPID 1 (0x1) sw1 L2 learned on Port 1 d6:16:20:a1:6c:c4 (L2 type 0x86dd, L2 dst
33:33:00:00:00:02, L3 src fe80::d416:20ff:fea1:6cc4, L3 dst ff02::2) Port 1 VLAN 100 (2 hosts total)
(END)
```

Here I checked the logs to see the hosts being discovered.

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.316 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.031 ms
^C
--- 192.168.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.031/0.108/0.316/0.120 ms
jason@ubuntu:~$ less /var/log/faucet/faucet.log
```

```
jason@ubuntu: ~
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 14:26:17 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 14:26:17 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     version 1.9.53
Apr 25 20:48:28 faucet INFO     Reloading configuration
Apr 25 20:48:28 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 20:48:28 faucet.config INFO     including file: /etc/faucet/acls.yaml
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 2 (0x2)
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv4 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN guest vid:200 tagged: Port 24 un
tagged: Port 2,Port 4 with VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 2 (0x2) sw2 IPv6 routing is active on VLAN office vid:100 tagged: Port 24 u
ntagged: Port 1 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 20:48:28 faucet INFO     Add new datapath DPID 1 (0x1)
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['10.0.200.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv4 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['10.0.100.254/24']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN guest vid:200 untagged: Port 3 w
ith VIPs ['2001:200::1/64', 'fe80::c00:ff:fe00:2001/64']
Apr 25 20:48:28 faucet.valve INFO     DPID 1 (0x1) sw1 IPv6 routing is active on VLAN office vid:100 tagged: Port 5 un
tagged: Port 1,Port 2,Port 4 with VIPs ['2001:100::1/64', 'fe80::c00:ff:fe00:1001/64']
Apr 25 21:20:02 faucet INFO     Reloading configuration
Apr 25 21:20:02 faucet INFO     configuration /etc/faucet/faucet.yaml changed, analyzing differences
Apr 25 21:20:02 faucet INFO     Reconfiguring existing datapath DPID 1 (0x1)
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 DP routers config changed - requires cold start
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 table IDs changed, old {0, 1, 2, 3, 4, 5, 6, 7, 8} new {0, 1, 2
, 3}
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 cold starting
Apr 25 21:20:02 faucet.valve INFO     DPID 1 (0x1) sw1 forcing DP reconnection to ensure ports are synchronized
Apr 25 21:20:02 faucet.valve ERROR    DPID 1 (0x1) sw1 send_flow_msgs: DP not up
Apr 25 21:20:02 faucet INFO     Deleting de-configured DPID 2 (0x2)
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Cold start configuring DP
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Port 1 (host1 network namespace) configured
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Port 2 (host2 network namespace) configured
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 Configuring VLAN office vid:100 untagged: Port 1,Port 2
Apr 25 21:36:52 faucet.valve INFO     DPID 1 (0x1) sw1 table ID 0 table config match_types: (('eth_dst', True), ('eth_
type', False), ('in_port', False), ('vlan_vid', False)) name: vlan next_tables: ['eth_src'] output: True set_fields: (
'vlan_vid',) size: 32 vlan_port_scale: 1.5
table ID 1 table config match_types: (('eth_dst', True), ('eth_src', False), ('eth_type', False), ('in_port', False),
('vlan_vid', False)) miss_goto: eth_dst name: eth_src next_tables: ['eth_dst', 'flood'] output: True set_fields: ('vla
n_vid', 'eth_dst') size: 32 table_id: 1 vlan_port_scale: 4.1
table ID 2 table config exact_match: True match_types: (('eth_dst', False), ('vlan_vid', False)) miss_goto: flood name
: eth_dst output: True size: 32 table_id: 2 vlan_port_scale: 4.1
table ID 3 table config match_types: (('eth_dst', True), ('in_port', False), ('vlan_vid', False)) name: flood output:
True size: 32 table_id: 3 vlan_port_scale: 7.0
Apr 25 21:37:16 faucet.valve INFO     DPID 1 (0x1) sw1 L2 learned on Port 2 a2:89:fd:c9:01:3f (L2 type 0x86dd, L2 dst
33:33:00:00:00:02, L3 src fe80::a089:fdff:fec9:13f, L3 dst ff02::2) Port 2 VLAN 100 (1 hosts total)
Apr 25 21:39:06 faucet.valve INFO     DPID 1 (0x1) sw1 L2 learned on Port 1 d6:16:20:a1:6c:c4 (L2 type 0x86dd, L2 dst
33:33:00:00:00:02, L3 src fe80::d416:20ff:fea1:6cc4, L3 dst ff02::2) Port 1 VLAN 100 (2 hosts total)
Apr 25 21:41:43 faucet.valve INFO     DPID 1 (0x1) sw1 L2 learned on Port 2 a2:89:fd:c9:01:3f (L2 type 0x0806, L2 dst
d6:16:20:a1:6c:c4, L3 src 192.168.0.2, L3 dst 192.168.0.1) Port 2 VLAN 100 (2 hosts total)
(END)
```

Here we can see the ping is functional and the newly added host have been recognized.



This shows a weird bug issue between Grafana, and the version of Ubuntu being used 16.04.7-desktop.

```
jason@ubuntu:~$ sudo apt-get install iperf3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libiperf0
The following NEW packages will be installed:
  iperf3 libiperf0
0 upgraded, 2 newly installed, 0 to remove and 40 not upgraded.
Need to get 58.5 kB of archives.
After this operation, 238 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 libiperf0 amd64 3.0.11-1 [50.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 iperf3 amd64 3.0.11-1 [8,090 B]
Fetched 58.5 kB in 0s (531 kB/s)
Selecting previously unselected package libiperf0:amd64.
(Reading database ... 221405 files and directories currently installed.)
Preparing to unpack .../libiperf0_3.0.11-1_amd64.deb ...
Unpacking libiperf0:amd64 (3.0.11-1) ...
Selecting previously unselected package iperf3.
Preparing to unpack .../iperf3_3.0.11-1_amd64.deb ...
Unpacking iperf3 (3.0.11-1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libiperf0:amd64 (3.0.11-1) ...
Setting up iperf3 (3.0.11-1) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
```

Here I installed iperf3 to generate a large amount of traffic for testing.

```
jason@ubuntu:~$ as_ns host1 iperf3 --server --pidfile /run/iperf3-host1.pid --daemon
iperf3: unrecognized option '--pidfile'
Usage: iperf [-s|-c host] [options]
       iperf [-h|--help] [-v|--version]

Server or Client:
  -p, --port      #         server port to listen on/connect to
  -f, --format    [kmgKMG]  format to report: Kbits, Mbits, KBytes, MBytes
  -i, --interval  #         seconds between periodic bandwidth reports
  -F, --file name           xmit/recv the specified file
  -A, --affinity n/n,m      set CPU affinity
  -B, --bind      <host>    bind to a specific interface
  -V, --verbose             more detailed output
  -J, --json                output in JSON format
  -d, --debug               emit debugging output
  -v, --version             show version information and quit
  -h, --help                show this message and quit
Server specific:
  -s, --server              run in server mode
  -D, --daemon              run the server as a daemon
  -1, --one-off             handle one client connection then exit
Client specific:
  -c, --client    <host>    run in client mode, connecting to <host>
  -u, --udp                 use UDP rather than TCP
  -b, --bandwidth #[KMG][/#] target bandwidth in bits/sec (0 for unlimited)
                            (default 1 Mbit/sec for UDP, unlimited for TCP)
                            (optional slash and packet count for burst mode)
  -t, --time      #         time in seconds to transmit for (default 10 secs)
  -n, --bytes     #[KMG]    number of bytes to transmit (instead of -t)
  -k, --blockcount #[KMG]   number of blocks (packets) to transmit (instead of -t or -n)
  -l, --len       #[KMG]    length of buffer to read or write
                            (default 128 KB for TCP, 8 KB for UDP)
  -P, --parallel  #         number of parallel client streams to run
  -R, --reverse             run in reverse mode (server sends, client receives)
  -w, --window    #[KMG]    set window size / socket buffer size
  -C, --linux-congestion <algo>  set TCP congestion control algorithm (Linux only)
  -M, --set-mss   #         set TCP maximum segment size (MTU - 40 bytes)
  -N, --nodelay             set TCP no delay, disabling Nagle's Algorithm
  -4, --version4            only use IPv4
  -6, --version6            only use IPv6
  -S, --tos N               set the IP 'type of service'
  -L, --flowlabel N         set the IPv6 flow label (only supported on Linux)
  -Z, --zerocopy            use a 'zero copy' method of sending data
  -O, --omit N              omit the first n seconds
  -T, --title str           prefix every output line with this string
  --get-server-output       get results from server

[KMG] indicates options that support a K/M/G suffix for kilo-, mega-, or giga-

iperf3 homepage at: http://software.es.net/iperf/
Report bugs to:     https://github.com/esnet/iperf
jason@ubuntu:~$ as_ns host2 iperf3 --client 192.168.0.1
iperf3: error - unable to connect to server: Connection refused
jason@ubuntu:~$
```

Here we can see iperf was successfully installed, but the connection was refused due to an unknown reason, likely software issues.

TASK THREE: ACL Tutorial

faucet.yaml

etc > faucet > ! faucet.yaml

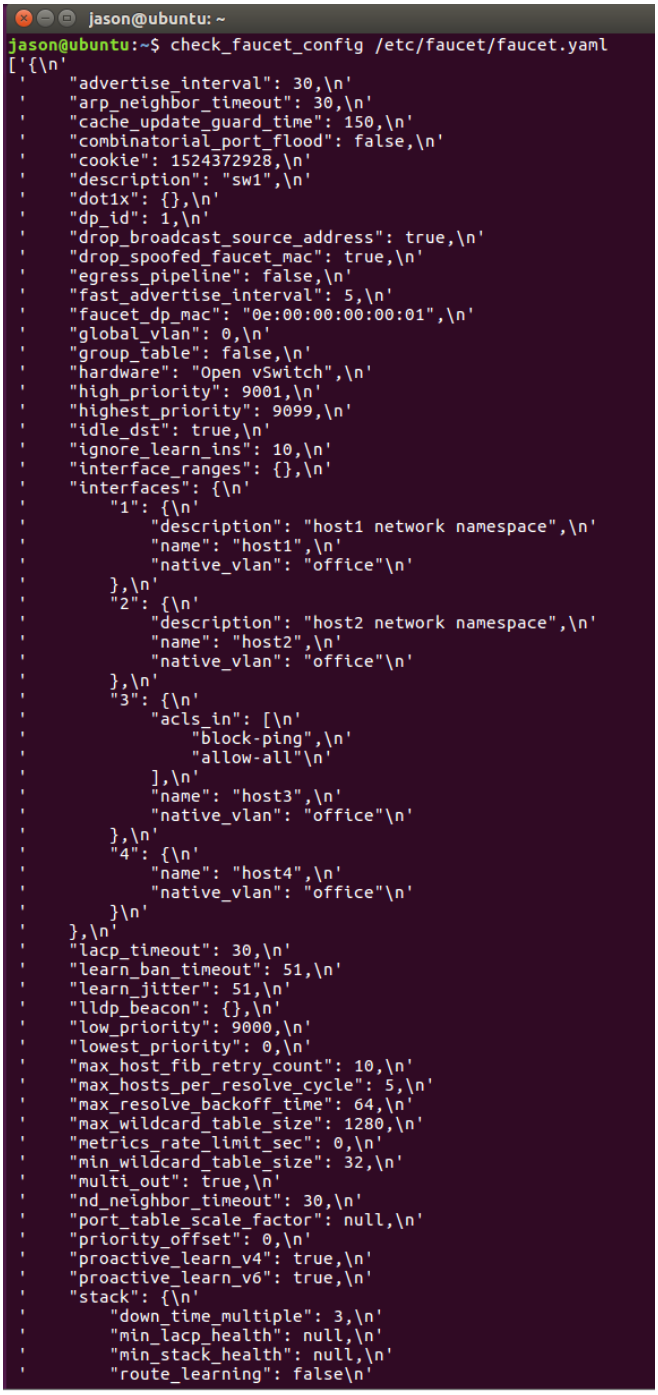
```yaml
vlans:
    office:
        vid: 100
        description: "office network"

dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: office
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: office
            3:
                name: "host3"
                native_vlan: office
                acls_in: [block-ping, allow-all]
            4:
                name: "host4"
                native_vlan: office
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
    allow-all:
        - rule:
            actions:
                allow: True
```

Here is the updated yaml file with acls added.

```
jason@ubuntu:~$ create_ns host3 192.168.0.3/24
jason@ubuntu:~$ create_ns host4 192.168.0.4/24
jason@ubuntu:~$ sudo ovs-vsctl add-port br0 veth-host3 -- set interface veth-host3 ofport_request=3 \
>               -- add-port br0 veth-host4 -- set interface veth-host4 ofport_request=4
jason@ubuntu:~$
```

Here I created 2 more hosts and connected them to the bridge br0.



Here I verified faucet has correctly interpreted the configuration.

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
25 packets transmitted, 0 received, 100% packet loss, time 24578ms

jason@ubuntu:~$ as_ns host1 ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.393 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.035 ms
^C
--- 192.168.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.030/0.104/0.393/0.144 ms
jason@ubuntu:~$ as_ns host1 ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4) 56(84) bytes of data.
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=0.260 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=0.025 ms
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=0.026 ms
64 bytes from 192.168.0.4: icmp_seq=4 ttl=64 time=0.040 ms
64 bytes from 192.168.0.4: icmp_seq=5 ttl=64 time=0.028 ms
^C
--- 192.168.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4091ms
rtt min/avg/max/mdev = 0.025/0.075/0.260/0.092 ms
jason@ubuntu:~$
```

Here we can see the ping to host3 fails and the pings to hosts 2 and 4 are good.

```
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
                mirror: 4
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
                mirror: 4
    allow-all:
        - rule:
            actions:
                allow: True
```

Here I added the mirror action to block-ping ACL.

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host4 tcpdump -l -e -n -i veth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth0, link-type EN10MB (Ethernet), capture size 262144 bytes



08:29:33.835216 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192.168.0.3: ICMP echo request, id 3222, seq 1, length 64
08:29:33.835286 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 1, length 64
08:29:34.846180 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192.168.0.3: ICMP echo request, id 3222, seq 2, length 64
08:29:34.846194 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 2, length 64
08:29:35.870133 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192.168.0.3: ICMP echo request, id 3222, seq 3, length 64
08:29:35.870148 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 3, length 64
08:29:36.894309 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192.168.0.3: ICMP echo request, id 3222, seq 4, length 64
08:29:36.894322 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 4, length 64
08:29:37.917935 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192.168.0.3: ICMP echo request, id 3222, seq 5, length 64
08:29:37.917947 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 5, length 64
08:29:38.878208 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype ARP (0x0806), length 42: Request who-has 192.168.0.3 tell 192.168.0.1, length 28
08:29:38.878315 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype ARP (0x0806), length 42: Reply 192.168.0.1 is-at c6:8d:a7:43:00:74, length 28
08:29:38.942427 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 6, length 64
08:29:39.965972 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 7, length 64
08:29:40.990158 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 8, length 64
08:29:42.014481 0e:b6:bf:da:9f:e3 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 3222, seq 9, length 64

08:30:47.998511 6a:7b:b7:f7:93:a0 > 33:33:00:00:00:02, ethertype IPv6 (0x86dd), length 70: fe80::687b:b7ff:fef7:93a0 > ff02::2: ICMP6, router solicitation, length 16
^C
17 packets captured
17 packets received by filter
0 packets dropped by kernel
jason@ubuntu:~$
```

Here I reloaded faucet to update the changes made and performed a tcpdump on host 4.

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
[sudo] password for jason:
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8179ms
```

Here I checked the ping to host 3.

```
native_vlan: office
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
                output:
                    - port: 4
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
                output:
                    - port: 4
    allow-all:
        - rule:
            actions:
                allow: True
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
15 packets transmitted, 0 received, 100% packet loss, time 14339ms

jason@ubuntu:~$
```

Here I added the output action, which is similar to the mirroring action and reloaded and validated the ping to host 3.

```
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
                output:
                    - set_fields:
                        - eth_src: "00:00:00:00:00:02"
                    - port: 4
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
                output:
                    - set_fields:
                        - eth_src: "00:00:00:00:00:02"
                    - port: 4
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5099ms

jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7167ms
```

Here I added ACLs to host 2 to change the MAC source address. I also reloaded and confirmed the pings do not work.

```
jason@ubuntu:~$ as_ns host2 ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.226 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.042 ms
^C
--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.042/0.090/0.226/0.078 ms
jason@ubuntu:~$
```

Here we can see the ping from host 2 to host 1 is functional.

```
jason@ubuntu:~$ as_ns host4 tcpdump -l -e -n -i veth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth0, link-type EN10MB (Ethernet), capture size 262144 bytes
08:41:38.623136 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 1, length 64
08:41:39.645959 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 2, length 64
08:41:40.670594 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 3, length 64
08:41:41.694165 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 4, length 64
08:41:42.718150 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 5, length 64
08:41:43.742431 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4201, seq 6, length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
jason@ubuntu:~$
```

```
jason@ubuntu:~$ as_ns host1 tcpdump -l -e -n -i veth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth0, link-type EN10MB (Ethernet), capture size 262144 bytes
08:43:43.171456 56:3b:a9:16:24:76 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.2 > 192
.168.0.1: ICMP echo request, id 4209, seq 1, length 64
08:43:43.171479 c6:8d:a7:43:00:74 > 56:3b:a9:16:24:76, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.2: ICMP echo reply, id 4209, seq 1, length 64
08:43:44.173451 56:3b:a9:16:24:76 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.2 > 192
.168.0.1: ICMP echo request, id 4209, seq 2, length 64
08:43:44.173475 c6:8d:a7:43:00:74 > 56:3b:a9:16:24:76, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.2: ICMP echo reply, id 4209, seq 2, length 64
08:43:45.182468 56:3b:a9:16:24:76 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.2 > 192
.168.0.1: ICMP echo request, id 4209, seq 3, length 64
08:43:45.182481 c6:8d:a7:43:00:74 > 56:3b:a9:16:24:76, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.2: ICMP echo reply, id 4209, seq 3, length 64
08:43:46.206035 56:3b:a9:16:24:76 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.2 > 192
.168.0.1: ICMP echo request, id 4209, seq 4, length 64
08:43:46.206057 c6:8d:a7:43:00:74 > 56:3b:a9:16:24:76, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.2: ICMP echo reply, id 4209, seq 4, length 64
08:43:48.286202 c6:8d:a7:43:00:74 > 56:3b:a9:16:24:76, ethertype ARP (0x0806), length 42: Request who-has 19
2.168.0.2 tell 192.168.0.1, length 28
08:43:48.286368 56:3b:a9:16:24:76 > c6:8d:a7:43:00:74, ethertype ARP (0x0806), length 42: Reply 192.168.0.2
is-at 56:3b:a9:16:24:76, length 28
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
jason@ubuntu:~$
```

Here we can see I started a tcpdump on both hosts 1 and 4 to check allowed connections.

```
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
                output:
                    - vlan_vid: 3
                    - port: 4
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
                output:
                    - vlan_vid: 3
                    - port: 4
```

```
jason@ubuntu:~$ as_ns host4 tcpdump -l -e -n -i veth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth0, link-type EN10MB (Ethernet), capture size 262144 bytes
08:49:06.507197 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.3: ICMP echo request, id 4274, seq 1, length 64
08:49:06.507253 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4274, seq 1, length 64
08:49:07.518314 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.3: ICMP echo request, id 4274, seq 2, length 64
08:49:07.518328 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4274, seq 2, length 64
08:49:08.542100 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.3: ICMP echo request, id 4274, seq 3, length 64
08:49:08.542113 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4274, seq 3, length 64
08:49:09.566299 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype IPv4 (0x0800), length 98: 192.168.0.1 > 192
.168.0.3: ICMP echo request, id 4274, seq 4, length 64
08:49:09.566315 00:00:00:00:00:02 > c6:8d:a7:43:00:74, ethertype IPv4 (0x0800), length 98: 192.168.0.3 > 192
.168.0.1: ICMP echo reply, id 4274, seq 4, length 64
08:49:11.614124 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype ARP (0x0806), length 42: Request who-has 19
2.168.0.3 tell 192.168.0.1, length 28
08:49:11.614160 c6:8d:a7:43:00:74 > 0e:b6:bf:da:9f:e3, ethertype ARP (0x0806), length 42: Reply 192.168.0.1
is-at c6:8d:a7:43:00:74, length 28
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
jason@ubuntu:~$
```

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
^C
--- 192.168.0.3 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3059ms
```

Here I added the ‘vlan_vid’ and ‘port’ field to the yaml file to identify what port the traffic originated on on the switch. I also verified the ping from host 1 to host 3 is not functional as this is disabled on vlan3.

TASK FOUR: VLAN Tutorial

```
jason@ubuntu:~$ # Run command inside network namespace
jason@ubuntu:~$ as_ns () {
>     NAME=$1
>     NETNS=faucet-${NAME}
>     shift
>     sudo ip netns exec ${NETNS} $@
> }
jason@ubuntu:~$ # Create network namespace
jason@ubuntu:~$ create_ns () {
>     NAME=$1
>     IP=$2
>     NETNS=faucet-${NAME}
>     sudo ip netns add ${NETNS}
>     sudo ip link add dev veth-${NAME} type veth peer name veth0 netns ${NETNS}
>     sudo ip link set dev veth-${NAME} up
>     as_ns ${NAME} ip link set dev lo up
>     [ -n "${IP}" ] && as_ns ${NAME} ip addr add dev veth0 ${IP}
>     as_ns ${NAME} ip link set dev veth0 up
> }
jason@ubuntu:~$ # Clean up namespaces, bridges and processes created during faucet tutorial
jason@ubuntu:~$ cleanup () {
>     for NETNS in $(sudo ip netns list | grep "faucet-" | awk '{print $1}'); do
>         [ -n "${NETNS}" ] || continue
>         NAME=${NETNS#faucet-}
>         if [ -f "/run/dhclient-${NAME}.pid" ]; then
>             # Stop dhclient
>             sudo pkill -F "/run/dhclient-${NAME}.pid"
>         fi
>         if [ -f "/run/iperf3-${NAME}.pid" ]; then
>             # Stop iperf3
>             sudo pkill -F "/run/iperf3-${NAME}.pid"
>         fi
>         if [ -f "/run/bird-${NAME}.pid" ]; then
>             # Stop bird
>             sudo pkill -F "/run/bird-${NAME}.pid"
>         fi
>         # Remove netns and veth pair
>         sudo ip link delete veth-${NAME}
>         sudo ip netns delete ${NETNS}
>     done
>     for isl in $(ip -o link show | awk -F': ' '{print $2}' | grep -oE "^l-br[0-9](_[0-9]*)?-br[0-9](_[0-9]*)?"); do
>         # Delete inter-switch links
>         sudo ip link delete dev $isl 2>/dev/null || true
>     done
>     for DNSMASQ in /run/dnsmasq-vlan*.pid; do
>         [ -e "${DNSMASQ}" ] || continue
>         # Stop dnsmasq
>         sudo pkill -F "${DNSMASQ}"
>     done
>     # Remove faucet dataplane connection
>     sudo ip link delete veth-faucet 2>/dev/null || true
>     # Remove openvswitch bridges
>     sudo ovs-vsctl --if-exists del-br br0
>     sudo ovs-vsctl --if-exists del-br br1
>     sudo ovs-vsctl --if-exists del-br br2
>     sudo ovs-vsctl --if-exists del-br br3
> }
jason@ubuntu:~$ # Add tagged VLAN interface to network namespace
jason@ubuntu:~$ add_tagged_interface () {
>      NAME=$1
>      VLAN=$2
>      IP=$3
>      NETNS=faucet-${NAME}
>      as_ns ${NAME} ip link add link veth0 name veth0.${VLAN} type vlan id ${VLAN}
>      [ -n "${IP}" ] && as_ns ${NAME} ip addr add dev veth0.${VLAN} ${IP}
>      as_ns ${NAME} ip link set dev veth0.${VLAN} up
>      as_ns ${NAME} ip addr flush dev veth0
> }
jason@ubuntu:~$
```

Here I added the proper dependencies functions.

```
jason@ubuntu:~$ cleanup
jason@ubuntu:~$ create_ns host1 192.168.0.1/24
jason@ubuntu:~$ create_ns host2 192.168.0.2/24
jason@ubuntu:~$ sudo ovs-vsctl add-br br0 \
> -- set bridge br0 other-config:datapath-id=0000000000000001 \
> -- set bridge br0 other-config:disable-in-band=true \
> -- set bridge br0 fail_mode=secure \
> -- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \
> -- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \
> -- set-controller br0 tcp:127.0.0.1:6653 tcp:127.0.0.1:6654
jason@ubuntu:~$ create_ns host3 0.0.0.0
jason@ubuntu:~$ create_ns host4 0.0.0.0
jason@ubuntu:~$ add_tagged_interface host3 100 192.168.0.3/24
jason@ubuntu:~$ add_tagged_interface host4 100 192.168.0.4/24
jason@ubuntu:~$
jason@ubuntu:~$ create_ns host5 192.168.2.5/24
jason@ubuntu:~$ create_ns host6 192.168.2.6/24
jason@ubuntu:~$
jason@ubuntu:~$
```

Here I ran a cleanup to get rid of the previous configurations. I also created 6 hosts with 2 bridged, 2 tagged with interfaces, and 2 with a VLAN range.

```
jason@ubuntu:~$ create_ns host7 0.0.0.0
jason@ubuntu:~$ create_ns host8 0.0.0.0
jason@ubuntu:~$ add_tagged_interface host7 300 192.168.3.7/24
jason@ubuntu:~$ add_tagged_interface host8 300 192.168.3.8/24
jason@ubuntu:~$
jason@ubuntu:~$ create_ns host9 0.0.0.0
jason@ubuntu:~$ add_tagged_interface host9 100 192.168.0.9/24
jason@ubuntu:~$ add_tagged_interface host9 200 192.168.2.9/24
jason@ubuntu:~$ add_tagged_interface host9 300 192.168.3.9/24
jason@ubuntu:~$
```

Here I added 3 more hosts and tagged some interfaces on them.

```
jason@ubuntu:~$ sudo ovs-vsctl add-port br0 veth-host3 -- set interface veth-host3 ofport_request=3 \
> -- add-port br0 veth-host4 -- set interface veth-host4 ofport_request=4 \
> -- add-port br0 veth-host5 -- set interface veth-host5 ofport_request=5 \
> -- add-port br0 veth-host6 -- set interface veth-host6 ofport_request=6 \
> -- add-port br0 veth-host7 -- set interface veth-host7 ofport_request=7 \
> -- add-port br0 veth-host8 -- set interface veth-host8 ofport_request=8 \
> -- add-port br0 veth-host9 -- set interface veth-host9 ofport_request=9
```

Here I added the hosts to the bridged interface br0.

faucet.yaml

etc > faucet > faucet.yaml

```yaml
vlans:
    vlan100:
        vid: 100
    vlan200:
        vid: 200
    vlan300:
        vid: 300
dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host2 network namespace"
                native_vlan: vlan100
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: vlan100
            3:
                name: "host3"
                tagged_vlans: [vlan100]
            4:
                name: "host4"
                tagged_vlans: [vlan100]
            5:
                name: "host5"
                native_vlan: vlan200
            6:
                name: "host6"
                native_vlan: vlan200
            7:
                name: "host7"
                tagged_vlans: [vlan300]
            8:
                name: "host8"
                tagged_vlans: [vlan300]
            9:
                name: "host9"
                tagged_vlans: [vlan100,vlan200,vlan300]
```

```
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 port 5 added
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 port 6 added
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 port 7 added
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 port 8 added
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 port 9 added
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 ports added: {8, 9, 5, 6, 7}
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 ports changed: {1, 2, 3, 4}
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 table IDs changed, old {0, 1, 2, 3, 4} new {0, 1, 2, 3}
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 cold starting
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 forcing DP reconnection to ensure ports are synchronized
Apr 26 09:24:09 faucet.valve WARNING  DPID 1 (0x1) sw1 datapath down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 1 (host2 network namespace) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 2 (host2 network namespace) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 3 (host3) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 4 (host4) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 5 (host5) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 6 (host6) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 7 (host7) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 8 (host8) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 9 (host9) down
Apr 26 09:24:09 faucet.valve WARNING  DPID 1 (0x1) sw1 datapath down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 1 (host2 network namespace) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 2 (host2 network namespace) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 3 (host3) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 4 (host4) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 5 (host5) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 6 (host6) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 7 (host7) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 8 (host8) down
Apr 26 09:24:09 faucet.valve INFO     DPID 1 (0x1) sw1 Port 9 (host9) down
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Cold start configuring DP
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Port 1 (host2 network namespace) configured
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Port 2 (host2 network namespace) configured
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Port 3 (host3) configured
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Port 4 (host4) configured
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Configuring VLAN vlan200 vid:200 tagged: Port 9 untagged: Port 5,Port 6
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Configuring VLAN vlan300 vid:300 tagged: Port 7,Port 8,Port 9
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 Configuring VLAN vlan100 vid:100 tagged: Port 3,Port 4,Port 9 untagged: Port 1,Port 2
Apr 26 09:24:10 faucet.valve INFO     DPID 1 (0x1) sw1 table ID 0 table config match_types: (('eth_dst', True), ('eth_type', False), ('in_port', False), ('vlan_vid', False)) name: vlan next_tables: ['eth_src']
 output: True set_fields: ('vlan_vid',) size: 64 vlan_port_scale: 1.5
table ID 1 table config match_types: (('eth_dst', True), ('eth_src', False), ('eth_type', False), ('in_port', False), ('vlan_vid', False)) miss_goto: eth_dst name: eth_src next_tables: ['eth_dst', 'flood'] out
put: True set_fields: ('vlan_vid', 'eth_dst') size: 128 table_id: 1 vlan_port_scale: 4.1
table ID 2 table config exact_match: True match_types: (('eth_dst', False), ('vlan_vid', False)) miss_goto: flood name: eth_dst output: True size: 128 table_id: 2 vlan_port_scale: 4.1
table ID 3 table_config match_types: (('eth_dst', True), ('in_port', False), ('vlan_vid', False)) name: flood output: True size: 192 table_id: 3 vlan_port_scale: 7.0
jason@ubuntu:~$
```

```
jason@ubuntu:~$ sudo vi /etc/faucet/faucet.yaml
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.199 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.028 ms
^C
--- 192.168.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2051ms
rtt min/avg/max/mdev = 0.028/0.086/0.199/0.080 ms
jason@ubuntu:~$ as_ns host3 ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4) 56(84) bytes of data.
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=0.206 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=0.034 ms
^C
--- 192.168.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.027/0.089/0.206/0.082 ms
```

Here I changed the yaml file to reflect basic VLAN settings as well as checked the hosts log, reloaded, and checked some routes.

```
jason@ubuntu:~$ as_ns host5 ping 192.168.2.6
PING 192.168.2.6 (192.168.2.6) 56(84) bytes of data.
64 bytes from 192.168.2.6: icmp_seq=1 ttl=64 time=0.266 ms
64 bytes from 192.168.2.6: icmp_seq=2 ttl=64 time=0.028 ms
64 bytes from 192.168.2.6: icmp_seq=3 ttl=64 time=0.030 ms
^C
--- 192.168.2.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.028/0.108/0.266/0.111 ms
jason@ubuntu:~$ as_ns host7 ping 192.168.3.8
PING 192.168.3.8 (192.168.3.8) 56(84) bytes of data.
From 192.168.3.7 icmp_seq=1 Destination Host Unreachable
From 192.168.3.7 icmp_seq=2 Destination Host Unreachable
From 192.168.3.7 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.3.8 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3051ms
pipe 4
```

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.3
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=0.214 ms
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=0.035 ms
^C
--- 192.168.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.035/0.094/0.214/0.085 ms
jason@ubuntu:~$
```

Here we can see hosts are able to ping each other.

```
jason@ubuntu:~$ as_ns host5 ip address add 192.168.0.5 dev veth0
jason@ubuntu:~$ as_ns host1 ping 192.168.0.5
PING 192.168.0.5 (192.168.0.5) 56(84) bytes of data.
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable
From 192.168.0.1 icmp_seq=2 Destination Host Unreachable
From 192.168.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.0.5 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3067ms
pipe 4
jason@ubuntu:~$
```

Here we can see a failed ping between hosts in different VLANs.

```
jason@ubuntu:~$ as_ns host1 ping 192.168.0.9
PING 192.168.0.9 (192.168.0.9) 56(84) bytes of data.
64 bytes from 192.168.0.9: icmp_seq=1 ttl=64 time=0.218 ms
64 bytes from 192.168.0.9: icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from 192.168.0.9: icmp_seq=3 ttl=64 time=0.027 ms
^C
--- 192.168.0.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.027/0.093/0.218/0.088 ms
jason@ubuntu:~$ as_ns host3 ping 192.168.0.9
PING 192.168.0.9 (192.168.0.9) 56(84) bytes of data.
64 bytes from 192.168.0.9: icmp_seq=1 ttl=64 time=0.271 ms
64 bytes from 192.168.0.9: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 192.168.0.9: icmp_seq=3 ttl=64 time=0.035 ms
^C
--- 192.168.0.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.030/0.112/0.271/0.112 ms
jason@ubuntu:~$ as_ns host5 ping 192.168.2.9
PING 192.168.2.9 (192.168.2.9) 56(84) bytes of data.
64 bytes from 192.168.2.9: icmp_seq=1 ttl=64 time=0.308 ms
64 bytes from 192.168.2.9: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 192.168.2.9: icmp_seq=3 ttl=64 time=0.032 ms
^C
--- 192.168.2.9 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.032/0.125/0.308/0.129 ms
jason@ubuntu:~$ as_ns host7 ping 192.168.3.9
PING 192.168.3.9 (192.168.3.9) 56(84) bytes of data.
From 192.168.3.7 icmp_seq=1 Destination Host Unreachable
From 192.168.3.7 icmp_seq=2 Destination Host Unreachable
From 192.168.3.7 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.3.9 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3079ms
pipe 4
jason@ubuntu:~$
```

Here we can see all the pings are valid except the random outlier from host 7 to 9, which I found odd as everything was configured properly.

```
acls:
    block-ping:
        - rule:
            dl_type: 0x800      # IPv4
            ip_proto: 1         # ICMP
            actions:
                allow: False
        - rule:
            dl_type: 0x86dd     # IPv6
            ip_proto: 58        # ICMPv6
            actions:
                allow: False
vlans:
    vlan100:
        vid: 100
    vlan200:
        vid: 200
    vlan300:
        vid: 300
        acls_in: [block-ping] # Apply ACL only on vlan300
```

```
jason@ubuntu:~$ as_ns host7 ping 192.168.3.8
PING 192.168.3.8 (192.168.3.8) 56(84) bytes of data.
64 bytes from 192.168.3.8: icmp_seq=1 ttl=64 time=0.594 ms
64 bytes from 192.168.3.8: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 192.168.3.8: icmp_seq=3 ttl=64 time=0.030 ms
^C
--- 192.168.3.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.029/0.217/0.594/0.266 ms
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host7 ping 192.168.3.8
PING 192.168.3.8 (192.168.3.8) 56(84) bytes of data.
From 192.168.3.7 icmp_seq=9 Destination Host Unreachable
From 192.168.3.7 icmp_seq=10 Destination Host Unreachable
From 192.168.3.7 icmp_seq=11 Destination Host Unreachable
^C
--- 192.168.3.8 ping statistics ---
12 packets transmitted, 0 received, +3 errors, 100% packet loss, time 11255ms
pipe 4
```

Here I edited the yaml file to reflect blocking any ICMP packets on VLAN 300.

TASK FIVE: Routing Tutorial

```
jason@ubuntu:~$ cleanup
jason@ubuntu:~$ # Run command inside network namespace
jason@ubuntu:~$ as_ns () {
>     NAME=$1
>     NETNS=faucet-${NAME}
>     shift
>     sudo ip netns exec ${NETNS} $@
> }
jason@ubuntu:~$ # Create network namespace
jason@ubuntu:~$ create_ns () {
>     NAME=$1
>     IP=$2
>     NETNS=faucet-${NAME}
>     sudo ip netns add ${NETNS}
>     sudo ip link add dev veth-${NAME} type veth peer name veth0 netns ${NETNS}
>     sudo ip link set dev veth-${NAME} up
>     as_ns ${NAME} ip link set dev lo up
>     [ -n "${IP}" ] && as_ns ${NAME} ip addr add dev veth0 ${IP}
>     as_ns ${NAME} ip link set dev veth0 up
> }
jason@ubuntu:~$ # Clean up namespaces, bridges and processes created during faucet tutorial
jason@ubuntu:~$ cleanup () {
>     for NETNS in $(sudo ip netns list | grep "faucet-" | awk '{print $1}'); do
>         [ -n "${NETNS}" ] || continue
>         NAME=${NETNS#faucet-}
>         if [ -f "/run/dhclient-${NAME}.pid" ]; then
>             # Stop dhclient
>             sudo pkill -F "/run/dhclient-${NAME}.pid"
>         fi
>         if [ -f "/run/iperf3-${NAME}.pid" ]; then
>             # Stop iperf3
>             sudo pkill -F "/run/iperf3-${NAME}.pid"
>         fi
>         if [ -f "/run/bird-${NAME}.pid" ]; then
>             # Stop bird
>             sudo pkill -F "/run/bird-${NAME}.pid"
>         fi
>         # Remove netns and veth pair
>         sudo ip link delete veth-${NAME}
>         sudo ip netns delete ${NETNS}
>     done
>     for isl in $(ip -o link show | awk -F': ' '{print $2}' | grep -oE "^l-br[0-9](_[0-9]*)?-br[0-9](_[0-9]*)?"); do
>         # Delete inter-switch links
>         sudo ip link delete dev $isl 2>/dev/null || true
>     done
>     for DNSMASQ in /run/dnsmasq-vlan*.pid; do
>         [ -e "${DNSMASQ}" ] || continue
>         # Stop dnsmasq
>         sudo pkill -F "${DNSMASQ}"
>     done
>     # Remove faucet dataplane connection
>     sudo ip link delete veth-faucet 2>/dev/null || true
>     # Remove openvswitch bridges
>     sudo ovs-vsctl --if-exists del-br br0
>     sudo ovs-vsctl --if-exists del-br br1
>     sudo ovs-vsctl --if-exists del-br br2
>     sudo ovs-vsctl --if-exists del-br br3
> }
jason@ubuntu:~$
```

Here I added the proper dependencies functions.

```
jason@ubuntu:~$ cleanup
jason@ubuntu:~$ create_ns host1 10.0.0.1/24
jason@ubuntu:~$ create_ns host2 10.0.1.2/24
jason@ubuntu:~$ sudo ovs-vsctl add-br br0 \
> -- set bridge br0 other-config:datapath-id=0000000000000001 \
> -- set bridge br0 other-config:disable-in-band=true \
> -- set bridge br0 fail_mode=secure \
> -- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \
> -- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \
> -- set-controller br0 tcp:127.0.0.1:6653 tcp:127.0.0.1:6654
jason@ubuntu:~$
```

I then ran a cleanup and created 2 hosts and added them to the bridge br0.

faucet.yaml

etc > faucet > faucet.yaml

```yaml
vlans:
    vlan100:
        vid: 100
        faucet_vips: ["10.0.0.254/24"]  # Faucet's virtual IP address for vlan100
        faucet_mac: "00:00:00:00:00:11"
    vlan200:
        vid: 200
        faucet_vips: ["10.0.1.254/24"]  # Faucet's virtual IP address for vlan200
        faucet_mac: "00:00:00:00:00:22"
dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: vlan100
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: vlan200
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ip route add default via 10.0.0.254 dev veth0
jason@ubuntu:~$ as_ns host2 ip route add default via 10.0.1.254 dev veth0
jason@ubuntu:~$
jason@ubuntu:~$ as_ns host1 ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
^C
--- 10.0.1.2 ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13299ms
```

Here I updated the yaml file to reflect a gateway between the two hosts and the routing between them. I also reloaded, added default routes, and verified the dropped ping due to being on different VLANs with different subnets.

faucet > ! faucet.yaml

```
vlans:
    vlan100:
        vid: 100
        faucet_vips: ["10.0.0.254/24"]  # Faucet's virtual IP address for vlan100
        faucet_mac: "00:00:00:00:00:11"
    vlan200:
        vid: 200
        faucet_vips: ["10.0.1.254/24"]  # Faucet's virtual IP address for vlan200
        faucet_mac: "00:00:00:00:00:22"

routers:
    router-1:                           # Router name
        vlans: [vlan100, vlan200]       # Names of vlans to allow routing between

dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: vlan100
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: vlan200
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=2 ttl=63 time=0.191 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=63 time=0.034 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=63 time=0.028 ms
^C
--- 10.0.1.2 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3075ms
rtt min/avg/max/mdev = 0.028/0.084/0.191/0.075 ms
jason@ubuntu:~$
```

Next I did the same thing, but I added the routers section, which allowed routing between them as seen with the ping from host 1 to 2.

```
jason@ubuntu:~$ cleanup
jason@ubuntu:~$ create_ns host1 10.0.0.1/24
jason@ubuntu:~$ create_ns host2 10.0.0.2/24
jason@ubuntu:~$ create_ns server 10.0.1.1/24
jason@ubuntu:~$
jason@ubuntu:~$ as_ns host1 ip route add default via 10.0.0.254
jason@ubuntu:~$ as_ns host2 ip route add default via 10.0.0.254
jason@ubuntu:~$ as_ns server ip route add default via 10.0.1.254
jason@ubuntu:~$
jason@ubuntu:~$ sudo ovs-vsctl add-br br0 \
> -- set bridge br0 other-config:datapath-id=0000000000000001 \
> -- set bridge br0 other-config:disable-in-band=true \
> -- set bridge br0 fail_mode=secure \
> -- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \
> -- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \
> -- add-port br0 veth-server -- set interface veth-server ofport_request=3 \
> -- set-controller br0 tcp:127.0.0.1:6653 tcp:127.0.0.1:6654
jason@ubuntu:~$
```

Here I ran a cleanup, and created 2 hosts, a server, and assigned default routes between them. I also added them to the bridge br0. This was done to set up static routing.

faucet.yaml

etc > faucet > faucet.yaml

```yaml
vlans:
    hosts:
        vid: 100
        description: "vlan for clients"
        faucet_mac: "00:00:00:00:00:11"
        faucet_vips: ["10.0.0.254/24"]

    servers:
        vid: 200
        description: "vlan for servers"
        faucet_mac: "00:00:00:00:00:22"
        faucet_vips: ["10.0.1.254/24"]
        routes:
            - route:
                ip_dst: "192.0.2.0/24"
                ip_gw: '10.0.1.1'
routers:
    router-hosts-servers:
        vlans: [hosts, servers]
dps:
    br0:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: hosts
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: hosts
            3:
                name: "server"
                description: "server network namespace"
                native_vlan: servers
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=2 ttl=63 time=0.241 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=63 time=0.031 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=63 time=0.030 ms
^C
--- 10.0.1.1 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3059ms
rtt min/avg/max/mdev = 0.030/0.100/0.241/0.099 ms
jason@ubuntu:~$
jason@ubuntu:~$ as_ns server ip address add 192.0.2.1/24 dev veth0
jason@ubuntu:~$ as_ns host1 ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
64 bytes from 192.0.2.1: icmp_seq=1 ttl=63 time=0.224 ms
64 bytes from 192.0.2.1: icmp_seq=2 ttl=63 time=0.036 ms
64 bytes from 192.0.2.1: icmp_seq=3 ttl=63 time=0.036 ms
^C
--- 192.0.2.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2031ms
rtt min/avg/max/mdev = 0.036/0.098/0.224/0.089 ms
jason@ubuntu:~$
```

Here I updated the yaml file and pinged the static routes set up.

faucet > ! faucet.yaml

```yaml
vlans:
    hosts:
        vid: 100
        description: "vlan for clients"
        faucet_mac: "00:00:00:00:00:11"
        faucet_vips: ["10.0.0.254/24"]

    servers:
        vid: 200
        description: "vlan for servers"
        faucet_mac: "00:00:00:00:00:22"
        faucet_vips: ["10.0.1.254/24"]

routers:
    router-hosts-servers:
        vlans: [hosts, servers]
dps:
    br0:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: hosts
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: hosts
            3:
                name: "server"
                description: "server network namespace"
                native_vlan: servers
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
^C
--- 192.0.2.1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9195ms
```

Here I added the routes section which removes the static route for BGP. We can see the ping fails here.

```
jason@ubuntu:~$ create_ns bgp 10.0.1.2/24
jason@ubuntu:~$ sudo ovs-vsctl add-port br0 veth-bgp -- set interface veth-bgp ofport_request=4
jason@ubuntu:~$
jason@ubuntu:~$ sudo ip link add veth-faucet type veth peer name veth-faucet-ovs
jason@ubuntu:~$ sudo ovs-vsctl add-port br0 veth-faucet-ovs -- set interface veth-faucet-ovs ofport_request=5
jason@ubuntu:~$ sudo ip addr add 10.0.1.3/24 dev veth-faucet
jason@ubuntu:~$ sudo ip link set veth-faucet up
jason@ubuntu:~$ sudo ip link set veth-faucet-ovs up
jason@ubuntu:~$
```

Here I setup the BGP links and enabled functionality.

```
jason@ubuntu:~$ sudo apt-get install bird
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  bird-doc
The following NEW packages will be installed:
  bird
0 upgraded, 1 newly installed, 0 to remove and 40 not upgraded.
Need to get 316 kB of archives.
After this operation, 1,091 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 bird amd64 1.5.0-4build1 [316 kB]
Fetched 316 kB in 0s (1,148 kB/s)
Selecting previously unselected package bird.
(Reading database ... 221420 files and directories currently installed.)
Preparing to unpack .../bird_1.5.0-4build1_amd64.deb ...
Unpacking bird (1.5.0-4build1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
Setting up bird (1.5.0-4build1) ...

Creating config file /etc/bird/bird.conf with new version

Creating config file /etc/bird/bird6.conf with new version
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
jason@ubuntu:~$ sudo systemctl stop bird
jason@ubuntu:~$ sudo systemctl stop bird6
jason@ubuntu:~$
```

Here I installed bird and stopped the running process that happened automatically upon install.

usr > share > bird > bird.conf

```
1   protocol kernel {
2       scan time 60;
3       import none;
4   }
5
6   protocol device {
7       scan time 60;
8   }
9
10  # Generate static route inside bird
11  protocol static {
12      route 192.0.2.0/24 via 10.0.1.1;
13  }
14
15  # BGP peer with faucet
16  # Import all routes and export our static route
17  protocol bgp faucet {
18      local as 65001;
19      neighbor 10.0.1.3 port 9179 as 65000;
20      export all;
21      import all;
22  }
```

Here I updated the BIRD configuration file.

etc > faucet > ! faucet.yaml

```yaml
vlans:
    hosts:
        vid: 100
        description: "vlan for clients"
        faucet_mac: "00:00:00:00:00:11"
        faucet_vips: ["10.0.0.254/24"]

    servers:
        vid: 200
        description: "vlan for servers"
        faucet_mac: "00:00:00:00:00:22"
        faucet_vips: ["10.0.1.254/24"]

routers:
    router-hosts-servers:
        vlans: [hosts, servers]
    bird:
        bgp:
            vlan: servers                       # The VLAN faucet use for BGP
            as: 65000                           # Faucet's AS number
            port: 9179                          # BGP port for Faucet to listen on.
            routerid: '10.0.1.3'                # Faucet's Unique ID.
            server_addresses: ['10.0.1.3']      # Faucet's listen IP for BGP
            neighbor_addresses: ['10.0.1.2']    # Neighbouring IP addresses (IPv4/IPv6)
            neighbor_as: 65001                  # Neighbour's AS number
dps:
    br0:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: hosts
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: hosts
            3:
                name: "server"
                description: "server network namespace"
                native_vlan: servers
            4:
                name: "bgp"
                description: "BIRD BGP router"
                native_vlan: servers
            5:
                name: "faucet"
                description: "faucet dataplane connection"
                native_vlan: servers
```

```
jason@ubuntu:~$ sudo vi /etc/bird/bird.conf
jason@ubuntu:~$
jason@ubuntu:~$ as_ns bgp bird -P /run/bird-bgp.pid
bird: Cannot create control socket /run/bird/bird.ctl: No such file or directory
jason@ubuntu:~$
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$
jason@ubuntu:~$ as_ns bgp birdc show protocols all faucet
Unable to connect to server control socket (/run/bird/bird.ctl): No such file or directory
jason@ubuntu:~$ as_ns bgp birdc show route export faucet
Unable to connect to server control socket (/run/bird/bird.ctl): No such file or directory
jason@ubuntu:~$ as_ns bgp birdc show route protocol faucet
Unable to connect to server control socket (/run/bird/bird.ctl): No such file or directory
jason@ubuntu:~$ as_ns host1 ping 192.0.2.1
PING 192.0.2.1 (192.0.2.1) 56(84) bytes of data.
^C
--- 192.0.2.1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2038ms

jason@ubuntu:~$
```

Here I updated the faucet.yaml file to talk to BIRD for BGP configuration. BIRD did not work as it seems there could be a software issue with versions.

TASK SIX: Connection tracking Tutorial

```
jason@ubuntu:~$ cleanup
jason@ubuntu:~$ sudo apt-get install conntrack
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  conntrack
0 upgraded, 1 newly installed, 0 to remove and 40 not upgraded.
Need to get 27.3 kB of archives.
After this operation, 91.1 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 conntrack amd64 1:1.4.3-3 [27.3 kB]
Fetched 27.3 kB in 0s (197 kB/s)
Selecting previously unselected package conntrack.
(Reading database ... 221455 files and directories currently installed.)
Preparing to unpack .../conntrack_1%3a1.4.3-3_amd64.deb ...
Unpacking conntrack (1:1.4.3-3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up conntrack (1:1.4.3-3) ...
jason@ubuntu:~$ # Run command inside network namespace
jason@ubuntu:~$ as_ns () {
>     NAME=$1
>     NETNS=faucet-${NAME}
>     shift
>     sudo ip netns exec ${NETNS} $@
> }
jason@ubuntu:~$ # Create network namespace
jason@ubuntu:~$ create_ns () {
>     NAME=$1
>     IP=$2
>     NETNS=faucet-${NAME}
>     sudo ip netns add ${NETNS}
>     sudo ip link add dev veth-${NAME} type veth peer name veth0 netns ${NETNS}
>     sudo ip link set dev veth-${NAME} up
>     as_ns ${NAME} ip link set dev lo up
>     [ -n "${IP}" ] && as_ns ${NAME} ip addr add dev veth0 ${IP}
>     as_ns ${NAME} ip link set dev veth0 up
> }
```

Here I ran cleanup and installed conntrack ACLs for stateful firewall rules.

```
jason@ubuntu:~$ # Clean up namespaces, bridges and processes created during faucet tutorial
jason@ubuntu:~$ cleanup () {
>     for NETNS in $(sudo ip netns list | grep "faucet-" | awk '{print $1}'); do
>         [ -n "${NETNS}" ] || continue
>         NAME=${NETNS#faucet-}
>         if [ -f "/run/dhclient-${NAME}.pid" ]; then
>             # Stop dhclient
>             sudo pkill -F "/run/dhclient-${NAME}.pid"
>         fi
>         if [ -f "/run/iperf3-${NAME}.pid" ]; then
>             # Stop iperf3
>             sudo pkill -F "/run/iperf3-${NAME}.pid"
>         fi
>         if [ -f "/run/bird-${NAME}.pid" ]; then
>             # Stop bird
>             sudo pkill -F "/run/bird-${NAME}.pid"
>         fi
>         # Remove netns and veth pair
>         sudo ip link delete veth-${NAME}
>         sudo ip netns delete ${NETNS}
>     done
>     for isl in $(ip -o link show | awk -F': ' '{print $2}' | grep -oE "^l-br[0-9](_[0-9]*)?-br[0-9](_[0-9]*)?"); do
>         # Delete inter-switch links
>         sudo ip link delete dev $isl 2>/dev/null || true
>     done
>     for DNSMASQ in /run/dnsmasq-vlan*.pid; do
>         [ -e "${DNSMASQ}" ] || continue
>         # Stop dnsmasq
>         sudo pkill -F "${DNSMASQ}"
>     done
>     # Remove faucet dataplane connection
>     sudo ip link delete veth-faucet 2>/dev/null || true
>     # Remove openvswitch bridges
>     sudo ovs-vsctl --if-exists del-br br0
>     sudo ovs-vsctl --if-exists del-br br1
>     sudo ovs-vsctl --if-exists del-br br2
>     sudo ovs-vsctl --if-exists del-br br3
> }
jason@ubuntu:~$ cleanup
jason@ubuntu:~$
```

Here I added the proper dependencies functions.

faucet > ! faucet.yaml

```
vlans:
    vlan100:
        vid: 100
        faucet_vips: ["10.0.0.254/24"]  # Faucet's virtual IP address for vlan100
        faucet_mac: "00:00:00:00:00:11"
    vlan200:
        vid: 200
        faucet_vips: ["10.0.1.254/24"]  # Faucet's virtual IP address for vlan200
        faucet_mac: "00:00:00:00:00:22"
dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: vlan100
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: vlan200
routers:
    router-1:                           # Router name
        vlans: [vlan100, vlan200]       # Names of vlans to allow routing between
```

```
jason@ubuntu:~$ create_ns host1 10.0.0.1/24
jason@ubuntu:~$ create_ns host2 10.0.1.2/24
jason@ubuntu:~$ sudo ovs-vsctl add-br br0 \
> -- set bridge br0 other-config:datapath-id=0000000000000001 \
> -- set bridge br0 other-config:disable-in-band=true \
> -- set bridge br0 fail_mode=secure \
> -- add-port br0 veth-host1 -- set interface veth-host1 ofport_request=1 \
> -- add-port br0 veth-host2 -- set interface veth-host2 ofport_request=2 \
> -- set-controller br0 tcp:127.0.0.1:6653
jason@ubuntu:~$
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ as_ns host1 ip route add default via 10.0.0.254 dev veth0
jason@ubuntu:~$ as_ns host2 ip route add default via 10.0.1.254 dev veth0
```

Here I updated the yaml file with basic firewall rule sets and created hosts, set the to the bridge br0m and added default routes.

```
jason@ubuntu:~$ as_ns host1 ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=2 ttl=63 time=0.618 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=63 time=0.026 ms
64 bytes from 10.0.1.2: icmp_seq=4 ttl=63 time=0.036 ms
^C
--- 10.0.1.2 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3055ms
rtt min/avg/max/mdev = 0.026/0.226/0.618/0.277 ms
jason@ubuntu:~$ as_ns host2 ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=63 time=0.328 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=63 time=0.035 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=63 time=0.027 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2041ms
rtt min/avg/max/mdev = 0.027/0.130/0.328/0.140 ms
jason@ubuntu:~$
```

Here I verified the 2 hosts can ping each other.

faucet > ! faucet.yaml

```yaml
acls:
    conntrack_fw:
        # Permit all ARP traffic such that hosts can resolve one another's MACs
        - rule:
            eth_type: 0x0806 # arp
            actions:
                allow: True
        # Begin tracking ALL untracked IPv4 connections
        - rule:
            eth_type: 0x0800 # ipv4
            ct_state: 0/0x20  # match -trk (untracked)
            actions:
                # Re-inject the tracked packet into the OpenFlow pipeline, containing
                # additional connection metadata, to default table 0. The tracked packet
                # is again evaluated by Faucet ACLs in table 0. The original, untracked
                # packet is effectively dropped.
                ct:
                    zone: 10 # arbitrary conntrack zone ID to match against later
                    table: 0
        # Commit NEW IPv4 connections from host1 to host2
        - rule:
            eth_type: 0x0800 # ipv4
            ipv4_src: 10.0.0.1
            ipv4_dst: 10.0.1.2
            ct_state: 0x21/0x21 # match +new - packets to establish a new connection
            actions:
                # Commit the connection to the connection tracking module which will be
                # stored beyond the lifetime of packet in the pipeline.
                ct:
                    zone: 10  # the same conntrack zone ID as above
                    flags: 1  # "commit" the new connection
                    table: 1  # implicit "allow" new connection packet(s) via faucet table 1
        # Allow packets in either direction from existing connections initiated by
        # host1 only
        - rule:
            eth_type: 0x0800 # ipv4
            ct_zone: 10 # match packets associated with our conntrack zone ID
            ct_state: 0x22/0x22 # match +est - packets in an established connection
            actions:
                allow: True
        # Block all unwanted packets and new connections from host2 to host1
        - rule:
            eth_type: 0x0800 # ipv4
            ipv4_src: 10.0.1.2
            ipv4_dst: 10.0.0.1
            actions:
                allow: False
vlans:
    vlan100:
        vid: 100
        faucet_vips: ["10.0.0.254/24"]  # Faucet's virtual IP address for vlan100
        faucet_mac: "00:00:00:00:00:11"
```

```yaml
dps:
    sw1:
        dp_id: 0x1
        hardware: "Open vSwitch"
        interfaces:
            1:
                name: "host1"
                description: "host1 network namespace"
                native_vlan: vlan100
                acls_in:
                    - conntrack_fw
            2:
                name: "host2"
                description: "host2 network namespace"
                native_vlan: vlan200
                acls_in:
                    - conntrack_fw
routers:
    router-1:                           # Router name
        vlans: [vlan100, vlan200]       # Names of vlans to allow routing between
```

Here I added connection tracking to the yaml file.

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ ovs-ofctl dump-flows br0 -O OpenFlow13 | grep =ct
ovs-ofctl: /var/run/openvswitch/br0.mgmt: failed to open socket (Permission denied)
jason@ubuntu:~$ ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.0.1,nw_dst=10.0.1.2
2024-04-26T18:05:28Z|00001|daemon_unix|WARN|/var/run/openvswitch/ovs-vswitchd.pid: open: Permission denied
ovs-appctl: cannot read pidfile "/var/run/openvswitch/ovs-vswitchd.pid" (Permission denied)
jason@ubuntu:~$ as_ns host1 ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=63 time=0.164 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=63 time=0.062 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=63 time=0.031 ms
^C
--- 10.0.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2035ms
rtt min/avg/max/mdev = 0.031/0.085/0.164/0.057 ms
jason@ubuntu:~$ sudo conntrack -L | grep 10.0.1.2
conntrack v1.4.3 (conntrack-tools): 0 flow entries have been shown.
jason@ubuntu:~$ ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.1.2,nw_dst=10.0.0.1
2024-04-26T18:06:31Z|00001|daemon_unix|WARN|/var/run/openvswitch/ovs-vswitchd.pid: open: Permission denied
ovs-appctl: cannot read pidfile "/var/run/openvswitch/ovs-vswitchd.pid" (Permission denied)
jason@ubuntu:~$ as_ns host2 ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=63 time=0.169 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=63 time=0.032 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=63 time=0.027 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.027/0.076/0.169/0.065 ms
jason@ubuntu:~$ sudo -i
root@ubuntu:~# ovs-ofctl dump-flows br0 -O OpenFlow13 | grep =ct
root@ubuntu:~# ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.0.1,nw_dst=10.0.1.2
Bridge: br0
Flow: tcp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.0.1,nw_dst=10.0.1.2,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0

Rule: table=0 cookie=0x5adc15c0 priority=4096,in_port=1,vlan_tci=0x0000/0x1fff
OpenFlow actions=push_vlan:0x8100,set_field:4196->vlan_vid,goto_table:1

        Resubmitted flow: tcp,in_port=1,dl_vlan=100,dl_vlan_pcp=0,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.0.1,nw_dst=10.0.1.2,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0
        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
        Resubmitted  odp: drop
        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
        Rule: table=1 cookie=0x5adc15c0 priority=4096,dl_vlan=100
        OpenFlow actions=CONTROLLER:96,goto_table:4

                Resubmitted flow: unchanged
                Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                Resubmitted  odp: push_vlan(vid=100,pcp=0)
                Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                Rule: table=4 cookie=0x5adc15c0 priority=0
                OpenFlow actions=goto_table:5

                        Resubmitted flow: unchanged
                        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                        Resubmitted  odp: push_vlan(vid=100,pcp=0)
                        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                        Rule: table=5 cookie=0x5adc15c0 priority=8192,dl_vlan=100
                        OpenFlow actions=pop_vlan,output:1
                        skipping output to input port

Final flow: unchanged
Megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
Datapath actions: push_vlan(vid=100,pcp=0)
This flow is handled by the userspace slow path because it:
        - Sends "packet-in" messages to the OpenFlow controller.
```

```
root@ubuntu:~# sudo conntrack -L | grep 10.0.1.2
conntrack v1.4.3 (conntrack-tools): 0 flow entries have been shown.
root@ubuntu:~# ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.1.2,nw_dst=10.0.0.1
Bridge: br0
Flow: tcp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.1.2,nw_dst=10.0.0.1,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0

Rule: table=0 cookie=0x5adc15c0 priority=4096,in_port=1,vlan_tci=0x0000/0x1fff
OpenFlow actions=push_vlan:0x8100,set_field:4196->vlan_vid,goto_table:1

        Resubmitted flow: tcp,in_port=1,dl_vlan=100,dl_vlan_pcp=0,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.1.2,nw_dst=10.0.0.1,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0
        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
        Resubmitted  odp: drop
        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
        Rule: table=1 cookie=0x5adc15c0 priority=4096,dl_vlan=100
        OpenFlow actions=CONTROLLER:96,goto_table:4

                Resubmitted flow: unchanged
                Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                Resubmitted  odp: push_vlan(vid=100,pcp=0)
                Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                Rule: table=4 cookie=0x5adc15c0 priority=0
                OpenFlow actions=goto_table:5

                        Resubmitted flow: unchanged
                        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                        Resubmitted  odp: push_vlan(vid=100,pcp=0)
                        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                        Rule: table=5 cookie=0x5adc15c0 priority=8192,dl_vlan=100
                        OpenFlow actions=pop_vlan,output:1
                        skipping output to input port

Final flow: unchanged
Megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
Datapath actions: push_vlan(vid=100,pcp=0)
This flow is handled by the userspace slow path because it:
        - Sends "packet-in" messages to the OpenFlow controller.
```

Here I checked that the ACLs were added and saw how the OVS interfaces with conntrack deal with tracked packets.

faucet > ! faucet.yaml

```yaml
acls:
    conntrack_fw:
        - rule:
            eth_type: 0x0806 # arp
            actions:
                allow: True
        - rule:
            eth_type: 0x0800 # ipv4
            ct_state: 0/0x20  # match -trk (untracked)
            actions:
                ct:
                    zone: 10
                    table: 0
        - rule:
            eth_type: 0x0800 # ipv4
            ipv4_src: 10.0.0.1
            ipv4_dst: 10.0.1.2
            ct_state: 0x21/0x21 # match +new - packets to establish a new connection
            actions:
                ct:
                    zone: 10
                    flags: 1  # "commit" the new connection
                    table: 1
                    # sNAT the connection to the faucet VIP
                    nat:
                        flags: 1
                        range_ipv4_min: 10.0.0.254
                        range_ipv4_max: 10.0.0.254
        - rule:
            eth_type: 0x0800 # ipv4
            ct_zone: 10
            ct_state: 0x22/0x22 # match +est - packets in an established connection
            actions:
                ct:
                    zone: 10
                    flags: 1  # NAT must include "commit" - this is a NO-OP for existing connection
                    table: 1
                    # sNAT the packets in an existing connection appropriately according to their d
                    nat:
                        flags: 1
        - rule:
            eth_type: 0x0800 # ipv4
            ipv4_src: 10.0.1.2
            ipv4_dst: 10.0.0.1
            actions:
                allow: False
```

```
jason@ubuntu:~$ sudo systemctl reload faucet
jason@ubuntu:~$ ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.0.1,nw_dst=10.0.1.2
2024-04-26T18:11:24Z|00001|daemon_unix|WARN|/var/run/openvswitch/ovs-vswitchd.pid: open: Permission denied
ovs-appctl: cannot read pidfile "/var/run/openvswitch/ovs-vswitchd.pid" (Permission denied)
jason@ubuntu:~$ sudo -i
root@ubuntu:~# ovs-appctl ofproto/trace br0 in_port=1,tcp,nw_src=10.0.0.1,nw_dst=10.0.1.2
Bridge: br0
Flow: tcp,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.0.1,nw_dst=10.0.1.2,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0

Rule: table=0 cookie=0x5adc15c0 priority=4096,in_port=1,vlan_tci=0x0000/0x1fff
OpenFlow actions=push_vlan:0x8100,set_field:4196->vlan_vid,goto_table:1

        Resubmitted flow: tcp,in_port=1,dl_vlan=100,dl_vlan_pcp=0,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_src=10.0.0.1,nw_dst=10.0.1.2,nw_tos=0,nw_ecn=0,nw_ttl=0,tp_src=0,tp_dst=0,tcp_flags=0
        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
        Resubmitted  odp: drop
        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
        Rule: table=1 cookie=0x5adc15c0 priority=4096,dl_vlan=100
        OpenFlow actions=CONTROLLER:96,goto_table:4

                Resubmitted flow: unchanged
                Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                Resubmitted  odp: push_vlan(vid=100,pcp=0)
                Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                Rule: table=4 cookie=0x5adc15c0 priority=0
                OpenFlow actions=goto_table:5

                        Resubmitted flow: unchanged
                        Resubmitted regs: reg0=0x0 reg1=0x0 reg2=0x0 reg3=0x0 reg4=0x0 reg5=0x0 reg6=0x0 reg7=0x0
                        Resubmitted  odp: push_vlan(vid=100,pcp=0)
                        Resubmitted megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
                        Rule: table=5 cookie=0x5adc15c0 priority=8192,dl_vlan=100
                        OpenFlow actions=pop_vlan,output:1
                        skipping output to input port

Final flow: unchanged
Megaflow: recirc_id=0,ip,in_port=1,vlan_tci=0x0000,dl_src=00:00:00:00:00:00,dl_dst=00:00:00:00:00:00,nw_frag=no
Datapath actions: push_vlan(vid=100,pcp=0)
This flow is handled by the userspace slow path because it:
        - Sends "packet-in" messages to the OpenFlow controller.
root@ubuntu:~#
```

```
root@ubuntu:~# # Run command inside network namespace
root@ubuntu:~# as_ns () {
>     NAME=$1
>     NETNS=faucet-${NAME}
>     shift
>     sudo ip netns exec ${NETNS} $@
> }
root@ubuntu:~# as_ns host1 ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=63 time=0.217 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=63 time=0.032 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=63 time=0.033 ms
^C
--- 10.0.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.032/0.094/0.217/0.086 ms
root@ubuntu:~# tcpdump -n -e -ttt -i veth-host2 host 10.0.0.254
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on veth-host2, link-type EN10MB (Ethernet), capture size 262144 bytes

sudo conntrack -L | grep 10.0.0.254
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@ubuntu:~# sudo conntrack -L | grep 10.0.0.254
conntrack v1.4.3 (conntrack-tools): 0 flow entries have been shown.
root@ubuntu:~#
```

Here I added NAT functionality to the yaml file and saw how OVS and conntrack use NAT on the packets.

Conclusion:

This lab was straight forward for many of the sections as the knowledge to complete the desired tasks are easily accessible by following the tutorials. For the most part everything went smoothly in this lab except a few sections where things occurred due to software compatibility issues between Ubuntu and changes made to the latest versions of tools used. This was a great experience and I now have some working knowledge with Faucet SDN Controller and some tools and features.

References:

https://docs.faucet.nz/en/latest/tutorials/first_time.html

https://docs.faucet.nz/en/latest/tutorials/acls.html

https://docs.faucet.nz/en/latest/tutorials/vlans.html

https://docs.faucet.nz/en/latest/tutorials/routing.html

https://docs.faucet.nz/en/latest/tutorials/conntrack.html

https://www.youtube.com/watch?v=Sb4ZWpTQtOg&t=2645s

https://docs.faucet.nz/en/latest/configuration.html

https://www.javatpoint.com/linux-edit-file

https://www.baeldung.com/linux/vi-delete-selected-text#:~:text=Using%20the%20Shift%20%2B%20V%20Keys,%E2%80%93%20Line%204.