Release G: Policy Clamp installation with Microk8s and Socks proxy

- Installation of MicroK8s
- Installation of ONAP with Policy Clamp in Microk8s
- Using Socks proxy to access the Microk8s cluster
    - Creating ssh tunnel for Socks proxy
    - Enabling Socks proxy with Firefox
    - Enabling socks proxy in windows environment
    - Using Socks proxy in Kubernetes config

Installation of MicroK8s

Install Microk8s with the following instructions,

https://ubuntu.com/tutorials/install-a-local-kubernetes-with-microk8s#1-overview

It is possible to enable multi node setup with Microk8s, In this exploration single node Microk8s cluster is used.

Enable host-access add-on in Microk8s to allow the Kubernetes containers to access the host.

https://microk8s.io/docs/addon-host-access

NOTE: Once the host-access enabled, The host can be accessed using IP address "**10.0.1.1**" from kubernetes pods.

Installation of ONAP with Policy Clamp in Microk8s

Policy clamp kubernetes participant used in the charts doesn't support the http based chartmuseum servers. It is supported in later version of the k8sparticipant (6.3.0).

To enable the http based chartmuseum access, k8sparticipant participant image needs to be upgraded with the configuration change as shown below.

- Update the image version to 6.3.0 in **smo-install/onap_oom/kubernetes/policy/components/policy-clamp-ac-k8s-ppnt/values.yaml**

```
###################################################################
# Application configuration defaults.
###################################################################
# application image
image: onap/policy-clamp-ac-k8s-ppnt:6.3.0
pullPolicy: Always
```

- Update the configuration(**smo-install/onap_oom/kubernetes/policy/components/policy-clamp-ac-k8s-ppnt/resources/config/KubernetesParticipantParameters.yaml**) to have the support for http with the chartmuseum configuration

```
helm:
  repos:
    -
      repoName: bitnami
      address: https://charts.bitnami.com/bitnami
    -
      repoName: chartmuseum
      address: 'http://10.0.1.1:18080'

 protocols:
    - http
    - https
```

- By default, policy clamp gui service is not enabled, Enable the policy-gui services in helm-override (**smo-install/helm-override/default/onap-override.yaml**)

```
    policy-gui:
      enabled: true
      image: onap/policy-gui:2.2.1
```

Install ONAP using it/dep repository and the instructions for the installation of ONAP is available in **smo-install/README.md**. This should create namespaces onap,nonrtric and network with all the containers running.

NOTE: Initial installation takes more time as the docker images are getting downloaded newly and the consequent installation will be faster.

Using Socks proxy to access the Microk8s cluster

Socks proxy can be used to access the containers/GUI from Microk8s cluster.

Creating ssh tunnel for Socks proxy

```
aravind.est@DESKTOP-CHC361K:~$ ssh -D9999 ubuntu@129.192.82.102
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-194-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information disabled due to load higher than 6.0

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

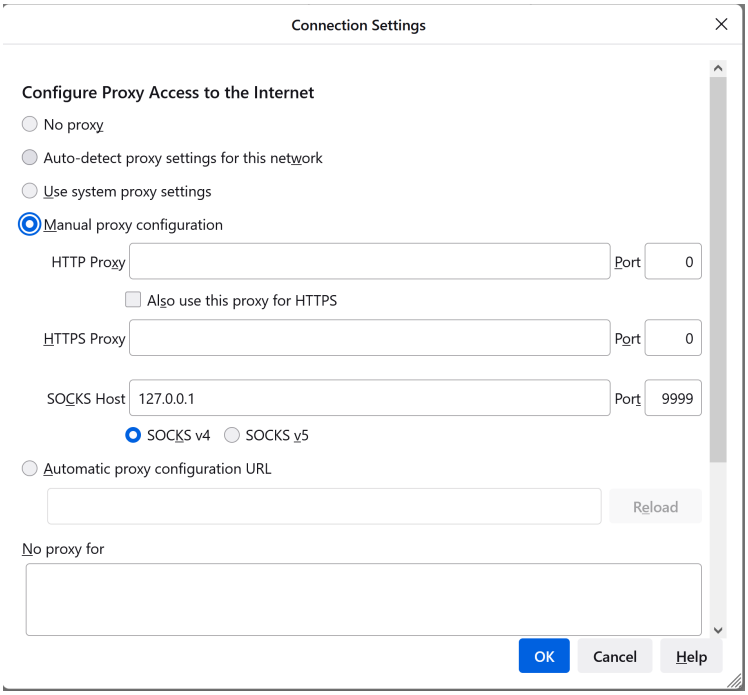
 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

138 packages can be updated.
39 updates are security updates.
```

Once this tunnel is enabled, socks proxy can be configured in windows application/environment.

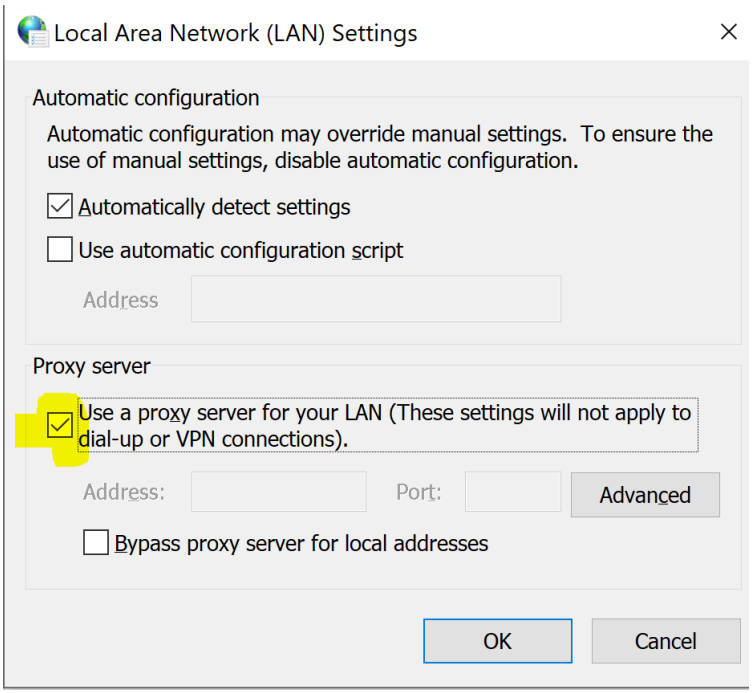
Enabling Socks proxy with Firefox

Socks proxy can be configured in Firefox as shown below,Go to, **Settings General Network Settings Manual Proxy Configuration** and configure as shown below.

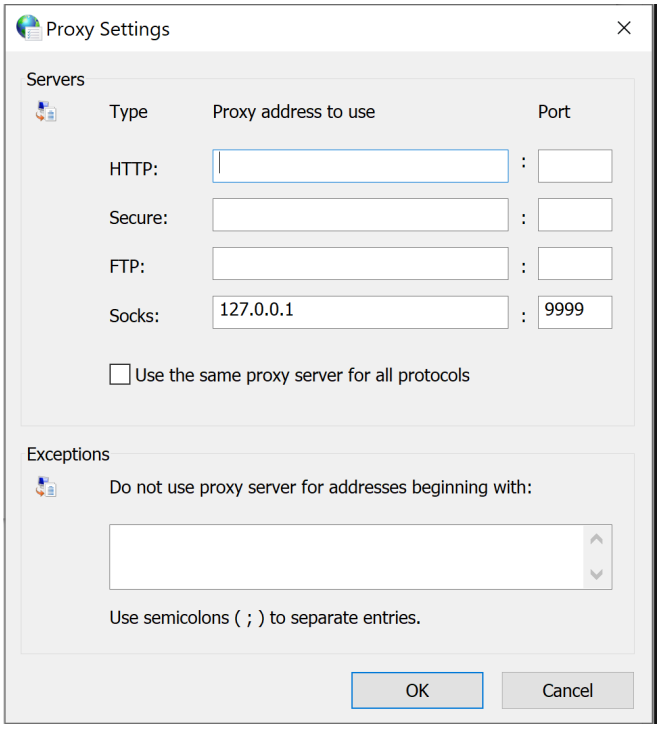


Enabling socks proxy in windows environment

Go to, **Internet Options Connections LAN Settings Proxy server**



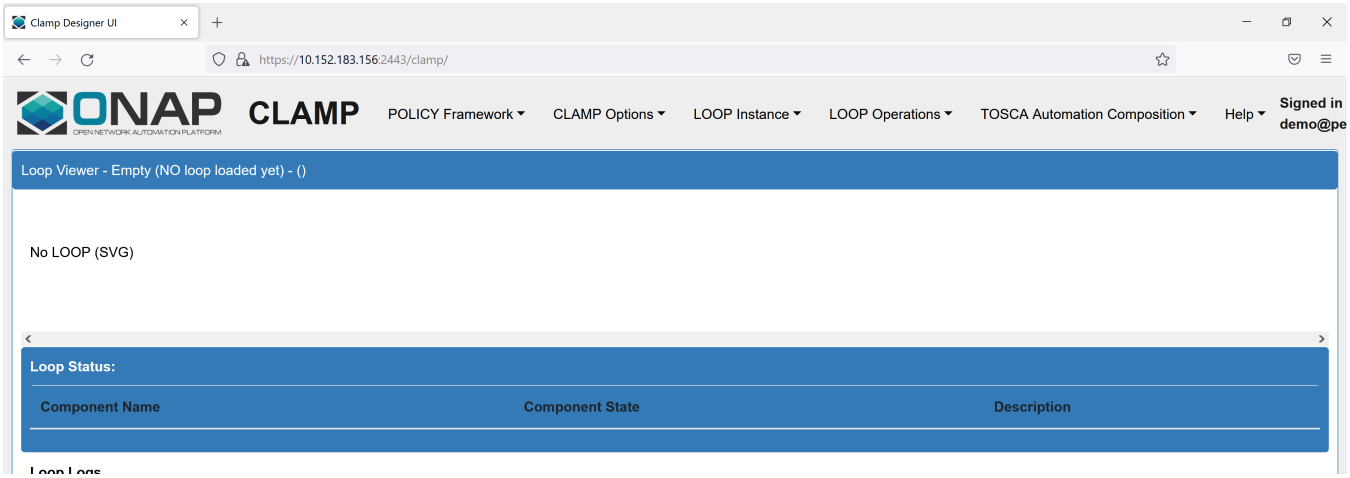
Enable the checkbox and click on **Advanced**,



Once the Socks proxy is configured, Kubernetes services in Microk8s can be accessed using Nodeport and ClusterIP.

```
sdnc-cluster                    ClusterIP   None            <none>        2550/TCP                 39h
sdnc-dmaap-listener             ClusterIP   None            <none>        <none>                   39h
sdnc-callhome                   NodePort    10.152.183.115  <none>        6666:30266/TCP           39h
sdnc-ansible-server             ClusterIP   10.152.183.148  <none>        8000/TCP                 39h
neng-serv                       ClusterIP   10.152.183.46   <none>        8080/TCP                 39h
sdnc-oam                        ClusterIP   10.152.183.93   <none>        8282/TCP,8202/TCP        39h
sdnrdb-service                  ClusterIP   10.152.183.244  <none>        9300/TCP                 39h
sdnc-web-service                NodePort    10.152.183.140  <none>        8443:30205/TCP           39h
sdnrdb                          ClusterIP   10.152.183.219  <none>        9200/TCP                 39h
sdnc                            NodePort    10.152.183.204  <none>        8443:30267/TCP           39h
sdnc-dgbuilder                  NodePort    10.152.183.109  <none>        3000:30203/TCP           39h
dcae-pmsh                       ClusterIP   10.152.183.83   <none>        8443/TCP                 20h
dcae-pmsh-postgres              ClusterIP   10.152.183.40   <none>        5432/TCP                 20h
dcae-pmsh-pg-replica            ClusterIP   10.152.183.210  <none>        5432/TCP                 20h
dcae-pmsh-pg-primary            ClusterIP   10.152.183.84   <none>        5432/TCP                 20h
```

Accessing policy clamp UI in Firefox.



Using Socks proxy in Kubernetes config

Socks proxy can be used to access the Kubernetes cluster. Get the kubeconfig file from the cluster which needs to be accessed and add the configuration as shown below,

```yaml
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: <DATA>
    server: https://10.6.0.223:16443
    proxy-url: socks5://localhost:9999
  name: microk8s-cluster
contexts:
- context:
    cluster: microk8s-cluster
    user: admin
  name: microk8s
current-context: microk8s
kind: Config
preferences: {}
users:
- name: admin
  user:
    token: <TOKEN>
```

proxy-url property should be configured based on the tunneling port.

Observations:

- ClusterIP and Nodeport are assigned per deployment basis. So there is no guarantee that the same IP address will be assigned to the same service on new deployment.
- Chrome doesn't have separate socks proxy configuration, instead it uses the windows environment proxy configuration.

- Socks proxy can be configured in intellij and eclipse IDE.
- Visual studio code doesn't have the support for socks proxy. It has the http based proxy configuration.
    - To enable the socks proxy support, pproxy(https://pypi.org/project/pproxy/) plugin can be used. This converts the socks proxy to http proxy and the http proxy can be configured in VS Code.