Lab 0: Environment Setup

Introduction:

This Lab exercise is to perform some of the environment specific configuration settings.

Objectives:

1. Update the Hosts file
2. Disabling SELinux
3. Disabling Firewalld Service
4. Installing and Configuring Chrony (NTP) Service
5. Enable VxLAN traffic for communication between Kubernetes pods across the cluster.
6. Install Docker

Environment Details:

Below table contains details of the servers that we will be using in this lab setup.

Host Name	IP Address	Role	OS	RAM
Xmaster	10.0.3x.1	Controller Node	CentOS-8	8GB
Xnode1	10.0.3x.2	Managed Node	CentOS-8	4GB
Xnode2	10.0.3x.3	Managed Node	CentOS-8	4GB

–

Login Details:

Login to the **Xmaster** server as **root** user with the password **redhat.**

*************In all systems xmaster/node1/node2 ********

1. Hostname Resolution

1.1 Add an entry to **/etc/hosts** file for **Local Name Resolution.**

```
# cat > /etc/hosts <<EOF
10.0.3x.1 Xmaster
10.0.3x.2 Xnode1
10.0.3x.3 Xnode2
127.0.0.1 localhost
EOF
```

1.2 Verify the **/etc/hosts** file updated successfully, by executing the below command.

```
# cat /etc/hosts
```

1.3 Test network connectivity between servers to ensure name resolution is working.

```
# ping -c 5 Xnode1
```

2. Disabling SELinux

2.1 Make sure **SELinux** is **disabled**

Security-Enhanced Linux (SELinux) is a **mandatory access control** (MAC) security mechanism implemented in the kernel.

SELinux has **three basic** modes of operation, of which Enforcing is set as the installation default mode.

· **Enforcing:** The default mode which will enable and enforce the SELinux security policy on the system, denying access and logging actions

· **Permissive:** In Permissive mode, SELinux is enabled but will not enforce the security policy, only warn and log actions. Permissive mode is useful for troubleshooting SELinux issues.

· **Disabled:** SELinux is turned off

```
cat /etc/sysconfig/selinux | grep SELINUX=
sed -i --follow-symlinks
's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
cat /etc/sysconfig/selinux | grep SELINUX=
setenforce 0
```

3. Disabling Firewalld Service

3.1 Let us disable **firewalld** Service.

· A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.

· A firewall typically establishes a barrier between a trusted network and an untrusted network, such as the Internet.

```
# systemctl disable --now firewalld
# systemctl status firewalld --no-pager
```

To latest update with port no details refer https://kubernetes.io/docs/reference/networking/ports-and-protocols/

4. Enable transparent masquerading and facilitate Virtual Extensible LAN (VxLAN) traffic for communication between Kubernetes pods across the cluster.

```
modprobe br_netfilter
lsmod | grep br_netfilter
modprobe overlay #####(not for Master nodes)#####
echo '1' > /proc/sys/net/bridge/bridge-nf-call-iptables
sysctl -a | grep net.bridge.bridge-nf-call-iptables
```

5. Install Docker for k8s

```
### Docker installation steps #######
      yum -y remove docker docker-client docker-client-latest docker-
      common docker-latest docker-latest-logrotate docker-logrotate
      docker-engine containers-common
      yum install -y yum-utils
      yum-config-manager --add-repo
      https://download.docker.com/linux/centos/docker-ce.repo
      yum install -y docker-ce docker-ce-cli containerd.io docker-
      compose-plugin
```

Both the container runtime and the kubelet have a property called "cgroup driver", which is important for the management of cgroups on Linux machines.

```
      containerd config default | sudo tee
      /etc/containerd/config.toml
      sed -i 's/SystemdCgroup = false/SystemdCgroup = true/g'
      /etc/containerd/config.toml

      systemctl start docker
      systemctl enable docker
      docker run hello-world
      ######################
```

Disable all memory swaps to increase performance

```
      swapoff -a
      sed -e '/swap/ s/^#*/#/' -i /etc/fstab
```

Note: Some of the below highlighted commands we will be executing on the **Xmaster** server later one of the lab exercises.

```
useradd -m -G wheel admin
echo "linux" | passwd --stdin admin
sed -e '/%wheel/ s/^#*/#/' -i /etc/sudoers
cat >> /etc/sudoers <<EOF
%wheel ALL=(ALL) NOPASSWD: ALL
EOF
```