Specifications and Implementation of base machine setup

1. Specifications

- Automation scripts sets up Virtual Labs cluster for us with minimal steps.
- The automation scripts can be run in any machine with minimum requirements as follows.

```
   Operating System: CentOS-6.x   - Used CentOS-6.9
   RAM: 4GB RAM
   Cpus: 2
   Disk Space: 100 GB to 150 GB
```

- Scripts are tested on following systems specs

```
Processor : Intel(R) Core(TM) i7-3770 , CPU with 3.40GHz (Model : intel-db75en)
RAM : 16 GB
No.of Cores (4 CPUs): 8
Disk Space: 100GB
```

- Any linux(CentOS) user, can run the scripts to setup the cluster. **Internet is needed to setup cluster**.
- Virtual-Labs cluster will be setup with our domain name `*{{clustername}}.vlabs.ac.in`.

  Here *** means, subdomain in `vlabs.ac.in`. example `cse01.vlabs.ac.in`

2. Base Machine Setup

`basesetup` role contains following ansible playbooks. Each playbook is included in `main.yml` file. The `main.yml` is the main play book to setup base machine. The `main.yml` playbook runs other playbooks included in it in a sequential order

2.1.1. Create br0 interface

- The following playbook `br0.yml` creates `br0` interface for host machine.This Creates ifcfg-br0 file in *etc/sysconfig/network-scripts* and adds data mentioned in block field

```
---
   - copy:
       content: ""
       dest: "/etc/sysconfig/network-scripts/ifcfg-br0"


   - name: creating br0 bridge
     blockinfile:
        path: /etc/sysconfig/network-scripts/ifcfg-br0
        block: |
         DEVICE=br0
         BOOTPROTO=static
         ONBOOT=yes
         TYPE=Bridge
         NM_CONTROLLED=no
         IPADDR={{ipaddress.hostmachine}}
         NETMASK={{net_mask}}
         GATEWAY={{internet_gateway}}
         DNS1={{dnsaddress.dns1}}
         DNS2={{dnsaddress.dns2}}

   - name: Network restart
     service: name=network state=restarted

...
```

2.1.2. Create eth0 interface on host machine.

Similar to above file,this also create ifcfg-eth0 file in *etc/sysconfig/network-scripts* and adds data mentioned in block field to the file

```
---

   - copy:
      content: ""
      dest: "/etc/sysconfig/network-scripts/ifcfg-eth0"

   - name: creating eth0 bridge
     blockinfile:
        path: /etc/sysconfig/network-scripts/ifcfg-eth0
        block: |
         DEVICE=eth0
         TYPE=Ethernet
         ONBOOT=yes
         NM_CONTROLLED=no
         BOOTPROTO=none
         BRIDGE=br0

...
```

2.1.3. Create br1 interface on host machine and restart network

Copies the content in block part to the destination mentioned in `path`. Then network restart and shuts network-manager off and restarts.Now you can see the bridges on your machine using brctl show command.

```
---

  - copy:
     content: ""
     dest: "/etc/sysconfig/network-scripts/ifcfg-br1"

  - name: creating Br1 bridge
    blockinfile:
       path: /etc/sysconfig/network-scripts/ifcfg-br1
       block: |
        DEVICE=br1
        TYPE=Bridge
        ONBOOT=yes
        NM_CONTROLLED=no
        BOOTPROTO=none

  - service:
     name: network
     state: restarted
  - name: Manager off
    command: chkconfig NetworkManager off
    register: out
  - debug: var=out.stdout_lines

  - name: Manager on
    command: chkconfig network on
    register: out
  - debug: var=out.stdout_lines

  - name: Running brctl show command
    command: brctl show
    register: out
  - debug: var=out.stdout_lines
```

3. Cluster node creation

3.1. Cluster nodes/servers creation

Following script does

- Downloads required CentOS template. {{centos-template}} is defined in common-vars
- Runs shell script `cluster.sh` to create basic cluster container(without network configuration in containers).

```
cluster="{{clustername}}"
    public_bridge="br0"
    private_bridge="br1"
    for i in {1001..1010}
    do
    vzctl create $i  --ostemplate centos-6-x86_64
    if [ $i == 1001 ]
    then
    vzctl set $i --hostname router.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth0,,,,$public_bridge --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1002 ]
    then
    vzctl set $i --hostname ansible.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth0,,,,$public_bridge --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1003 ]
    then
    vzctl set $i --hostname ossec-server.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save

    elif [ $i == 1004 ]
    then
    vzctl set $i --hostname rsyslog.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1005 ]
    then
    vzctl set $i --hostname privatedns.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save

    elif [ $i == 1006 ]
    then
    vzctl set $i --hostname publicdns.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1007 ]
    then
    vzctl set $i --hostname reverseproxy.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1008 ]
    then
    vzctl set $i --hostname nagios.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1009 ]
    then
    vzctl set $i --hostname ads.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --diskspace 20G --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    elif [ $i == 1010 ]
    then
    vzctl set $i --hostname rsnapshot.$cluster.vlabs.ac.in --save
    vzctl set $i --netif_add eth1,,,,$private_bridge --save
    vzctl start $i
    vzctl set $i --onboot yes --save
    fi
    done
```