kubespray install

* bug issue : https://github.com/kubernetes-sigs/kubespray/issues

$ sudo apt update
$ sudo apt install git python3 python3-pip
$ git clone https://github.com/kubernetes-sigs/kubespray.git

vagrant@control:~/kubespray$ cat requirements.txt
ansible==9.13.0
# Needed for community.crypto module
cryptography==44.0.1
# Needed for jinja2 json_query templating
jmespath==1.0.1
# Needed for ansible.utils.ipaddr
netaddr==1.3.0
vagrant@control:**~/kubespray**$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting ansible==9.13.0
  Downloading ansible-9.13.0-py3-none-any.whl (51.5 MB)

51.5/51.5 MB 6.6 MB/s eta 0:00:00
Collecting cryptography==44.0.1
  Downloading cryptography-44.0.1-cp39-abi3-manylinux_2_34_x86_64.whl (4.2 MB)

4.2/4.2 MB 10.7 MB/s eta 0:00:00
Collecting jmespath==1.0.1
...

=> ansible 과 jinja2 템플릿등이 설치됩니다.

아래처럼 ansible 이 설치되어 있는 경로를 찾고 PATH 에 포함되어 있는지 확인.
$ **sudo find / -name ansible**
/home/vagrant/.local/bin/ansible -> $PATH 변수에 포함이 안되어 있는 실행파일 경로

* 명령어 앞에 프롬프트가 '$' 만 표시되어 있는 경우 현재 디렉토리 위치는 어떤디렉토리라도 상관없습니다.

$ echo $PATH ; ansible 파일의 경로가 PATH 변수에 포함되어 있지 않으면
  아래처럼 ansible 파일 경로를 PATH 환경뼌수에 추가
$ export PATH=$PATH:/home/vagrant/.local/bin (source ~/.profile 명령을 실행해도 된다)
* 새로 login 하더라도 PATH 변수를 적용하기위해서 source ~/.profile 을 실행

ansible manage node(ansible로 관리할 관리대상장비)를 ansible server가 암호를 요구하지 않는 비대화식으로 명령을 전달하기 위해서는공개키를 생성후 ansible managed-node 에 업로드 하여여 한다.

ansible control-server에 /etc/hosts 파일에 아래 내용추가(managed-node 주소를 등록)

```
vagrant@control:~/kubespray$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
# -- 이 라인 아래부터 추가한 내용
192.168.11.10   master
192.168.11.20   worker1
192.168.11.30   worker2

vagrant@control:~/kubespray$ ssh-keygen  -> 공개키/개인키 생성
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:YPvKpwyAduYWnkEIZQAPzakqKXCeVBgNrStzI/y5dOo vagrant@control
The key's randomart image is:
+---[RSA 3072]----+
|*==B             |
|.+* +            |
| o.+  o          |
|o.=  . o         |
|=*.B  . S        |
|X.@.+  .         |
|o=.Bo.  .        |
|  ooo+ ..        |
|  .E. =o         |
+----[SHA256]-----+
vagrant@control:~/kubespray$

vagrant@control:~/kubespray$ ssh-copy-id 192.168.38.10   공개키를 안전하게 전송
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/vagrant/.ssh/id_rsa.pub"
The authenticity of host '192.168.38.10 (192.168.38.10)' can't be established.
ED25519 key fingerprint is SHA256:6DM2fU3KK/NeRjq6XQsFpg7idhDUfe2h7hZyOKrc+QU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
vagrant@192.168.38.10's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '192.168.38.10'"
and check to make sure that only the key(s) you wanted were added.

vagrant@control:~/kubespray$
```

나머지 node 에 대해서도 반복한다.

vagrant@control:~/kubespray$ ssh-copy-id 192.168.38.20
vagrant@control:~/kubespray$ ssh-copy-id 192.168.38.30

* managed-node 에서 방화벽 서비스(ubuntu는 ufw)를 중단하고 해야 방화벽 서비스를 중단하는것은
(* ufw 나 firewalld 를 중단해도 iptables 는 계속 사용됩니다.)
ansible playbook 에 설정되어 있으므로 직접 처리할 필요는 없습니다.

vagrant@control:**~/kubespray**$ cat contrib/os-services/roles/prepare/tasks/main.yml ;
이파일에 방화벽서비스 해제가 설정되어 있음.

ansible-playbook 으로 managed-node 를 제어하기위해서는 inventory 파일이 필수입니다.
inventory 는 managed-node 의 주소이며(ip address 나 domain name, hostname 이 가능합니다)
아래처럼 inventory sample 파일이 있는 디렉토리를 이름을 변경하여 복사합니다.

vagrant@control:**~/kubespray**$ cp -rfp inventory/sample/ inventory/mycluster/

#hosts.yaml -- cluster 구성을 위한 inventory (*. 실습에서는 이 파일을 사용하지 않는다)

```
all:
  hosts:
    master:
      ansible_host: 192.168.11.10
      ip: 192.168.11.10
      access_ip: 192.168.11.10
    worker1:
      ansible_host: 192.168.11.20
      ip: 192.168.11.20
      access_ip: 192.168.11.20
    worker2:
      ansible_host: 192.168.11.30
      ip: 192.168.11.30
      access_ip: 192.168.11.30
  children:
    kube_control_plane:
      hosts:
        master:
    kube_node:
      hosts:
        master:
        worker1:
        worker2:
    etcd:
      hosts:
        master:
    k8s_cluster:
      children:
        kube_control_plane:
        kube_node:
    calico_rr:
      hosts: {}
```

*. children 에 설정되 있는 각 title은 한개이상의 host 가 포함되어 있는 host 그룹을 의미합니다.

위에 설정은 kube_control_plane 은 master 이며 kubernetes node 는 master, worker1, worker2 세대로
구성하며 etcd 는 master 에 설치
k8s_cluster 는kube_contrl_plane 그룹과 kube_node 그룹으로 구성

yaml 로 작성하지 않고 아래처럼 작성해도 된다.

vagrant@control:**~/kubespray**$ cat inventory/mycluster/inventory.ini

```
[kube_control_plane]
master ansible_host=192.168.38.10 ip=192.168.38.10

[etcd:children]
kube_control_plane

[kube_node]
worker1 ansible_host=192.168.38.20 ip=192.168.38.20
worker2 ansible_host=192.168.38.30 ip=192.168.38.30

[calico_rr]

[k8s_cluster:children]
kube_control_plane
kube_node
calico_rr
```

vagrant@control:**~/kubespray**$ ansible-playbook -i **inventory/mycluster/inventory.ini** --become --become-user=root cluster.yml <- yaml 형식대신에 두번째 inventory 파일을 선택함

[kubernetes/preinstall : Stop if some versions have a 'v' left at the start] ********** fatal: [worker1]: FAILED! => {"msg": "The conditional check 'not etcd_version.startswith('v')' fail ed. The error was: error while evaluating conditional (not etcd_version.startswith('v')): {{ etcd_s upported_versions[kube_major_version] }}: 'dict object' has no attribute 'v1.32'. 'dict object' has no attribute 'v1.32'. {{ etcd_supported_versions[kube_major_version] }}: 'dict object' has no attr ibute 'v1.32'. 'dict object' has no attribute 'v1.32'"}fatal: [worker2]: FAILED! => {"msg": "The conditional check 'not etcd_version.startswith('v')' fail ed. The error was: error while evaluating conditional (not etcd_version.startswith('v')): {{ etcd_s upported_versions[kube_major_version] }}: 'dict object' has no attribute 'v1.32'. 'dict object' has no attribute 'v1.32'. {{ etcd_supported_versions[kube_major_version] }}: 'dict object' has no attr ibute 'v1.32'. 'dict object' has no attribute 'v1.32'"}fatal: [master]: FAILED! => {"msg": "The conditional check 'not etcd_version.startswith('v')' faile d. The error was: error while evaluating conditional (not etcd_version.startswith('v')): {{ etcd_su pported_versions[kube_major_version] }}: 'dict object' has no attribute 'v1.32'. 'dict object' has no attribute 'v1.32'. {{ etcd_supported_versions[kube_major_version] }}: 'dict object' has no attri bute 'v1.32'. 'dict object' has no attribute 'v1.32'"}

위에서와 같은 에러가 출력이 되고 진행이 멈추면
$ grep -IR "etcd_supported_versions" kubespray/
kubespray/roles/kubespray-defaults/defaults/main/download.yml
kubespray/extra_playbooks/roles/kubespray-defaults/defaults/main/download.yml

첫번째 파일에서 v1.32 부분을 찾아서 --> 1.32 로 변경해야 한다.
kubespray/roles/kubespray-defaults/defaults/main/download.yml - 이파일에서 수정

모든 작업이 완료된후 ssh 192.168.x.x (master node 주소) 로 접속해서
아래의 명령어들로 cluster 상태 확인

```
vagrant@master:~$ sudo kubectl get nodes
NAME      STATUS   ROLES           AGE   VERSION
master    Ready    control-plane   55m   v1.32.2
worker1   Ready    <none>          52m   v1.32.2
worker2   Ready    <none>          52m   v1.32.2
vagrant@master:~$ sudo kubectl get pod -n kube-system
NAME                                       READY   STATUS    RESTARTS      AGE
calico-kube-controllers-6747688466-mjjql 1/1     Running   0             51m
calico-node-4zwmd                          1/1     Running   0             51m
calico-node-7pmst                          1/1     Running   0             51m
calico-node-q7nhn                          1/1     Running   0             51m
coredns-5c54f84c97-5ndlw                   1/1     Running   0             51m
coredns-5c54f84c97-wsq44                   1/1     Running   0             39m
dns-autoscaler-76ddddbbc-8t95n             1/1     Running   0             40m
kube-apiserver-master                      1/1     Running   1             55m
kube-controller-manager-master             1/1     Running   6 (34m ago)   55m
kube-proxy-2mh77                           1/1     Running   0             52m
kube-proxy-llkxx                           1/1     Running   0             52m
kube-proxy-qqs5j                           1/1     Running   0             52m
kube-scheduler-master                      1/1     Running   2 (40m ago)   55m
nginx-proxy-worker1                        1/1     Running   0             52m
nginx-proxy-worker2                        1/1     Running   0             52m
nodelocaldns-4wpvp                         1/1     Running   0             40m
nodelocaldns-fbctq                         1/1     Running   0             40m
nodelocaldns-pp2vk                         1/1     Running   0             40m
vagrant@master:~$
```

```
vagrant@master:~$ sudo kubectl run apache --image httpd
pod/apache created
vagrant@master:~$ sudo kubectl get pods -o wide
NAME     READY   STATUS    RESTARTS   AGE   IP             NODE      NOMINATED
NODE   READINESS GATES
apache   1/1     Running   0          9s    10.233.125.2   worker2   <none>
     <none>
vagrant@master:~$ curl 10.233.125.2
<html><body><h1>It works!</h1></body></html>
vagrant@master:~$
```