

1.2. Настройка kubelet

Нужно поправить файлы конфигов на всех серверах кластера после присоединения к кластеру (до этого `/var/lib/kubelet/config.yaml` может и не оказаться):

1. Добавить в `/var/lib/kubelet/kubeadm-flags.env` флаги `--runtime-cgroups=/systemd/system.slice` и `--kubelet-cgroups=/systemd/system.slice`:

```
sudo sed -i 's/--network-plugin=cni/--network-plugin=cni
--runtime-cgroups=/systemd/system.slice
--kubelet-cgroups=/systemd/system.slice/ '
/var/lib/kubelet/kubeadm-flags.env
```

2. Настроить совместимые версии драйверов **kubelet** и **docker** (systemd/sysv) проверить версию драйвера cgroup-driver в docker, обычно это cgroupfs и установить такой же для **kubelet** - в файле `/var/lib/kubelet/config.yaml` в заменить значение `cgroupDriver`:

```
sudo sed -i 's/cgroupfs/systemd/' /var/lib/kubelet/config.yaml
```

3. Включить возможность управления ресурсами, в файле `/etc/systemd/system/kubelet.service.d/10-kubeadm.conf` добавить в начало:

```
[Service]
CPUAccounting=true
MemoryAccounting=true
```

```
sudo sed -i '/\[Service\]/aCPUAccounting=true\nMemoryAccounting=true'
/etc/systemd/system/kubelet.service.d/10-kubeadm.conf
```

4. Перезапустить сервис kubelet:

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart kubelet
```

Описание Kube-APIserver

Должно быть идентичным на всех control plane нодах (`/etc/kubernetes/manifests/kube-apiserver.yaml`)

Изменены флаги запуска kube-apiserver

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    scheduler.alpha.kubernetes.io/critical-pod: ""
  creationTimestamp: null
```

```

labels:
  component: kube-apiserver
  tier: control-plane
name: kube-apiserver
namespace: kube-system
spec:
  containers:
    - command:
      - kube-apiserver
      - --authorization-mode=Node,RBAC
      - --advertise-address=192.168.66.206
      - --allow-privileged=true
      - --client-ca-file=/etc/kubernetes/pki/ca.crt
      - --disable-admission-plugins=PersistentVolumeLabel
      -
      --enable-admission-plugins=NodeRestriction,NamespaceLifecycle,LimitRanger,
      ServiceAccount,DefaultStorageClass,DefaultTolerationSeconds,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,ResourceQuota
      - --enable-aggregator-routing=true
      - --enable-bootstrap-token-auth=true
      - --enable-swagger-ui=true
      - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
      - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
      - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
      - --etcd-servers=https://127.0.0.1:2379
      - --insecure-port=0
      -
      --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.
      crt
      - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
      - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
      - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
      - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
      - --requestheader-allowed-names=front-proxy-client
      - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
      - --requestheader-extra-headers-prefix=X-Remote-Extra-
      - --requestheader-group-headers=X-Remote-Group
      - --requestheader-username-headers=X-Remote-User
      - --secure-port=6443
      - --service-account-key-file=/etc/kubernetes/pki/sa.pub
      - --service-cluster-ip-range=10.96.0.0/12
      - --service-node-port-range=80-49151
      - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
      - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
    image: k8s.gcr.io/kube-apiserver-amd64:v1.11.2
    imagePullPolicy: IfNotPresent
    livenessProbe:
      failureThreshold: 8
      httpGet:
        host: 192.168.66.206
        path: /healthz
        port: 6443
        scheme: HTTPS

```

```
    initialDelaySeconds: 15
    timeoutSeconds: 15
  name: kube-apiserver
  resources:
    requests:
      cpu: 250m
  volumeMounts:
  - mountPath: /etc/kubernetes/pki
    name: k8s-certs
    readOnly: true
  - mountPath: /etc/ssl/certs
    name: ca-certs
    readOnly: true
  - mountPath: /etc/pki
    name: etc-pki
    readOnly: true
  hostNetwork: true
  priorityClassName: system-cluster-critical
  volumes:
  - hostPath:
      path: /etc/kubernetes/pki
      type: DirectoryOrCreate
    name: k8s-certs
  - hostPath:
      path: /etc/ssl/certs
      type: DirectoryOrCreate
    name: ca-certs
  - hostPath:
      path: /etc/pki
```

```
    type: DirectoryOrCreate
  name: etc-pki
status: {}
```