kubernetes配置私有仓库（仓库公开）

前提:

已经将k8s的镜像上传到仓库

A:Harbor私有仓库

B:k8s服务（docker客户端）

1、设置Harbor访问级别

在Harbor将仓库的访问级别调整为公开，允许任何人拉去镜像

2、证书配置

在B中配置Hador的CA证书，否则在走https的时候会用出现证书认证报错的问题

3、修改安装配置

vim config.yaml

点击此处展开...

```
apiVersion: kubeadm.k8s.io/v1alpha1
kind: MasterConfiguration
api:
 advertiseAddress: 192.168.253.143  ## 本机IP地址
networking:
 podSubnet: 10.244.0.0/16
apiServerCertSANs:
- 192.168.253.143  ## 本机IP地址
imageRepository: 10.22.60.169/sunline_k8s ## Harbor镜像仓库的地址
kubernetesVersion: v1.10.0
```

4、修改网络插件flannel.yaml配置

vim flannel.yaml

点击此处展开...

```
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: flannel
rules:
 - apiGroups:
 - ""
 resources:
 - pods
 verbs:
 - get
 - apiGroups:
 - ""
 resources:
 - nodes
 verbs:
 - list
```

```
 - watch
 - apiGroups:
 - ""
 resources:
 - nodes/status
 verbs:
 - patch
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: flannel
roleRef:
 apiGroup: rbac.authorization.k8s.io
 kind: ClusterRole
 name: flannel
subjects:
- kind: ServiceAccount
 name: flannel
 namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
 name: flannel
 namespace: kube-system
---
kind: ConfigMap
apiVersion: v1
metadata:
 name: kube-flannel-cfg
 namespace: kube-system
 labels:
 tier: node
 app: flannel
data:
 cni-conf.json: |
 {
 "name": "cbr0",
 "type": "flannel",
 "delegate": {
 "isDefaultGateway": true
 }
 }
 net-conf.json: |
 {
 "Network": "10.244.0.0/16",
 "Backend": {
 "Type": "vxlan"
 }
 }
---
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
 name: kube-flannel-ds
 namespace: kube-system
 labels:
 tier: node
 app: flannel
spec:
 template:
 metadata:
 labels:
 tier: node
 app: flannel
 spec:
 hostNetwork: true
 nodeSelector:
 beta.kubernetes.io/arch: amd64
 tolerations:
 - key: node-role.kubernetes.io/master
```

```
      operator: Exists
      effect: NoSchedule
      serviceAccountName: flannel
      initContainers:
      - name: install-cni
      image: 10.22.60.169/sunline_k8s/flannel:v0.9.1-amd64 ## 修改为Harbor的私有仓库地址
      command:
      - cp
      args:
      - -f
      - /etc/kube-flannel/cni-conf.json
      - /etc/cni/net.d/10-flannel.conf
      volumeMounts:
      - name: cni
      mountPath: /etc/cni/net.d
      - name: flannel-cfg
      mountPath: /etc/kube-flannel/
      containers:
      - name: kube-flannel
      image: 10.22.60.169/sunline_k8s/flannel:v0.9.1-amd64  ## 修改为Harbor的私有仓库地址
      command: [ "/opt/bin/flanneld", "--ip-masq", "--kube-subnet-mgr" ]
      securityContext:
      privileged: true
      env:
      - name: POD_NAME
      valueFrom:
      fieldRef:
      fieldPath: metadata.name
      - name: POD_NAMESPACE
      valueFrom:
      fieldRef:
      fieldPath: metadata.namespace
      volumeMounts:
      - name: run
      mountPath: /run
      - name: flannel-cfg
      mountPath: /etc/kube-flannel/
      volumes:
      - name: run
      hostPath:
      path: /run
      - name: cni
      hostPath:
      path: /etc/cni/net.d
      - name: flannel-cfg
      configMap:
      name: kube-flannel-cfg
```

5、修改dashboard.yaml配置

vim dashboard.yaml

点击此处展开...

```
# Copyright 2017 The Kubernetes Authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
```

```
# See the License for the specific language governing permissions and
# limitations under the License.

# Configuration to deploy release version of the Dashboard UI compatible with
# Kubernetes 1.8.
#
# Example usage: kubectl create -f <this_file>

# ------------------- Dashboard Secret ------------------- #

apiVersion: v1
kind: Secret
metadata:
labels:
k8s-app: kubernetes-dashboard
name: kubernetes-dashboard-certs
namespace: kube-system
type: Opaque

---
# ------------------- Dashboard Service Account ------------------- #

apiVersion: v1
kind: ServiceAccount
metadata:
labels:
k8s-app: kubernetes-dashboard
name: kubernetes-dashboard
namespace: kube-system

---
# ------------------- Dashboard Role & Role Binding ------------------- #

kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
name: kubernetes-dashboard-minimal
namespace: kube-system
rules:
# Allow Dashboard to create 'kubernetes-dashboard-key-holder' secret.
- apiGroups: [""]
resources: ["secrets"]
verbs: ["create"]
# Allow Dashboard to create 'kubernetes-dashboard-settings' config map.
- apiGroups: [""]
resources: ["configmaps"]
verbs: ["create"]
# Allow Dashboard to get, update and delete Dashboard exclusive secrets.
- apiGroups: [""]
resources: ["secrets"]
resourceNames: ["kubernetes-dashboard-key-holder", "kubernetes-dashboard-certs"]
verbs: ["get", "update", "delete"]
# Allow Dashboard to get and update 'kubernetes-dashboard-settings' config map.
- apiGroups: [""]
resources: ["configmaps"]
```

```
resourceNames: ["kubernetes-dashboard-settings"]
verbs: ["get", "update"]
# Allow Dashboard to get metrics from heapster.
- apiGroups: [""]
resources: ["services"]
resourceNames: ["heapster"]
verbs: ["proxy"]
- apiGroups: [""]
resources: ["services/proxy"]
resourceNames: ["heapster", "http:heapster:", "https:heapster:"]
verbs: ["get"]

---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
name: kubernetes-dashboard-minimal
namespace: kube-system
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: Role
name: kubernetes-dashboard-minimal
subjects:
- kind: ServiceAccount
name: kubernetes-dashboard
namespace: kube-system

---
# ------------------- Dashboard Deployment ------------------- #

kind: Deployment
apiVersion: apps/v1beta2
metadata:
labels:
k8s-app: kubernetes-dashboard
name: kubernetes-dashboard
namespace: kube-system
spec:
replicas: 1
revisionHistoryLimit: 10
selector:
matchLabels:
k8s-app: kubernetes-dashboard
template:
metadata:
labels:
k8s-app: kubernetes-dashboard
spec:
containers:
- name: kubernetes-dashboard
image: 10.22.60.169/sunline_k8s/kubernetes-dashboard-amd64:v1.8.3 ## 修改仓库地址为Harbor地址
ports:
- containerPort: 8443
protocol: TCP
args:
```

```
- --auto-generate-certificates
# Uncomment the following line to manually specify Kubernetes API server Host
# If not specified, Dashboard will attempt to auto discover the API server and connect
# to it. Uncomment only if the default does not work.
# - --apiserver-host=http://my-address:port
volumeMounts:
- name: kubernetes-dashboard-certs
mountPath: /certs
# Create on-disk volume to store exec logs
- mountPath: /tmp
name: tmp-volume
livenessProbe:
httpGet:
scheme: HTTPS
path: /
port: 8443
initialDelaySeconds: 30
timeoutSeconds: 30
volumes:
- name: kubernetes-dashboard-certs
secret:
secretName: kubernetes-dashboard-certs
- name: tmp-volume
emptyDir: {}
serviceAccountName: kubernetes-dashboard
# Comment the following tolerations if Dashboard must not be deployed on master
tolerations:
- key: node-role.kubernetes.io/master
effect: NoSchedule

---
# ------------------- Dashboard Service ------------------- #

kind: Service
apiVersion: v1
metadata:
labels:
k8s-app: kubernetes-dashboard
name: kubernetes-dashboard
namespace: kube-system
spec:
ports:
- port: 443
targetPort: 8443
nodePort: 30080 # 外放访问端口
type: NodePort
selector:
k8s-app: kubernetes-dashboard
```

6、安装登陆配置admin.yaml(无需修改)

vim admin-user.yaml

点击此处展开...

```
apiVersion: v1

kind: ServiceAccount
metadata:
name: admin-user
namespace: kube-system

---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
name: admin-user
roleRef:
apiGroup: rbac.authorization.k8s.io
kind: ClusterRole
name: cluster-admin
subjects:
- kind: ServiceAccount
name: admin-user
namespace: kube-system
```

修改好配置后：

1、查看创建集群的时候，查看仓库的网卡出口流量是否增加

2、查看docker日志是否正常

```
# journalctl -u kubelet -f
```