МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №7
«Знакомство с Kubernetes»

Практическая работа
по дисциплине «Современные технологии программирования»
студента 1 курса группы ПИ-б-о-232
Халилов Асан Русланович

направления подготовки 09.03.04 «Программная инженерия»

Симферополь, 2024

Цель: Ознакомиться на практике с инструментом оркестрации контейнеризированных приложений Kubernetes.

Ход выполнения задания.

1,2,3,4,5

```
[node1 ~]$ kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16
Initializing machine ID from random generator.
W0525 09:02:59.446824     520 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet e
rrors in the future. Automatically prepending scheme "unix" to the "criSocket" with value "/run/docker/containerd/containerd.sock". Ple
ase update your configuration!
I0525 09:02:59.725916     520 version.go:256] remote version is much newer: v1.30.1; falling back to: stable-1.27
[init] Using Kubernetes version: v1.27.14
[preflight] Running pre-flight checks
        [WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kub
elet
[preflight] The system verification failed. Printing the output from the verification:
KERNEL_VERSION: 4.4.0-210-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUACCT: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
        [WARNING SystemVerification]: failed to parse kernel config: unable to load kernel module: "configs", output: "", err: exit sta
tus 1
        [WARNING FileContent--proc-sys-net-bridge-bridge-nf-call-iptables]: /proc/sys/net/bridge/bridge-nf-call-iptables does not exist
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
W0525 09:03:00.166577     520 images.go:80] could not find officially supported version of etcd for Kubernetes v1.27.14, falling back t
o the nearest etcd version (3.5.7-0)
W0525 09:03:14.634321     520 checks.go:835] detected that the sandbox image "registry.k8s.io/pause:3.6" of the container runtime is in
consistent with that used by kubeadm. It is recommended that using "registry.k8s.io/pause:3.9" as the CRI sandbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.clu
ster.local node1] and IPs [10.96.0.1 192.168.0.18]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [localhost node1] and IPs [192.168.0.18 127.0.0.1 ::1]
[certs] Generating "etcd/peer" certificate and key
[certs] Generating "etcd/healthcheck-client" certificate and key
[certs] Generating "apiserver-etcd-client" certificate and key
[certs] Generating "sa" key and public key
[kubeconfig] Using kubeconfig folder "/etc/kubernetes"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "kubelet.conf" kubeconfig file
[kubeconfig] Writing "controller-manager.conf" kubeconfig file
[kubeconfig] Writing "scheduler.conf" kubeconfig file
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Starting the kubelet
[control-plane] Using manifest folder "/etc/kubernetes/manifests"
[control-plane] Creating static Pod manifest for "kube-apiserver"
[control-plane] Creating static Pod manifest for "kube-controller-manager"
[control-plane] Creating static Pod manifest for "kube-scheduler"
[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
W0525 09:03:31.168847     520 images.go:80] could not find officially supported version of etcd for Kubernetes v1.27.14, falling back t
o the nearest etcd version (3.5.7-0)
[wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". Th
is can take up to 4m0s
[apiclient] All control plane components are healthy after 7.502469 seconds
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
```

```
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upload-certs] Skipping phase. Please see --upload-certs
[mark-control-plane] Marking the node node1 as control-plane by adding the labels: [node-role.kubernetes.io/control-plane node.kubernet
es.io/exclude-from-external-load-balancers]
[mark-control-plane] Marking the node node1 as control-plane by adding the taints [node-role.kubernetes.io/control-plane:NoSchedule]
[bootstrap-token] Using token: 0qf246.7qk5vpgzyjt4ga4l
[bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate cre
dentials
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 192.168.0.18:6443 --token 0qf246.7qk5vpgzyjt4ga4l \
        --discovery-token-ca-cert-hash sha256:13814c75f4af5addd9368d580dc7b8766703333c65f516c9e6f9a462e7a8da44
Waiting for api server to startup
Warning: resource daemonsets/kube-proxy is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by
 kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl
apply. The missing annotation will be patched automatically.
daemonset.apps/kube-proxy configured
No resources found
```

```
[node1 ~]$ kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml
configmap/kube-router-cfg created
daemonset.apps/kube-router created
serviceaccount/kube-router created
clusterrole.rbac.authorization.k8s.io/kube-router created
clusterrolebinding.rbac.authorization.k8s.io/kube-router created
```

6

```
[node1 ~]$ kubectl get nodes
NAME    STATUS   ROLES           AGE    VERSION
node1   Ready    control-plane   2m4s   v1.27.2
```

7

```
[node1 ~]$ kubectl get pods -A
NAMESPACE     NAME                            READY   STATUS    RESTARTS   AGE
kube-system   coredns-5d78c9869d-jv5hx        1/1     Running   0          2m4s
kube-system   coredns-5d78c9869d-vzq5m        1/1     Running   0          2m4s
kube-system   etcd-node1                      1/1     Running   0          2m18s
kube-system   kube-apiserver-node1            1/1     Running   0          2m10s
kube-system   kube-controller-manager-node1   1/1     Running   0          2m13s
kube-system   kube-proxy-pxcmg                1/1     Running   0          2m4s
kube-system   kube-router-czxvs               1/1     Running   0          2m3s
kube-system   kube-scheduler-node1            1/1     Running   0          2m16s
```

8

```
[node2 ~]$ kubeadm join 192.168.0.18:6443 --token 0qf246.7qk5vpgzyjt4ga4l \
>         --discovery-token-ca-cert-hash sha256:13814c75f4af5addd9368d580dc7b8766703333c65f516c9e6f9a462e7a8da44
Initializing machine ID from random generator.
W0525 09:06:38.813303    1254 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet e
rrors in the future. Automatically prepending scheme "unix" to the "criSocket" with value "/run/docker/containerd/containerd.sock". Ple
ase update your configuration!
[preflight] Running pre-flight checks
        [WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kub
elet
[preflight] The system verification failed. Printing the output from the verification:
KERNEL_VERSION: 4.4.0-210-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUACCT: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
```

```
[node3 ~]$ kubeadm join 192.168.0.18:6443 --token 0qf246.7qk5vpgzyjt4ga4l \
>         --discovery-token-ca-cert-hash sha256:13814c75f4af5addd9368d580dc7b8766703333c65f516c9e6f9a462e7a8da44
Initializing machine ID from random generator.
W0525 09:06:47.548387    1280 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet e
rrors in the future. Automatically prepending scheme "unix" to the "criSocket" with value "/run/docker/containerd/containerd.sock". Ple
ase update your configuration!
[preflight] Running pre-flight checks
        [WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kub
elet
[preflight] The system verification failed. Printing the output from the verification:
KERNEL_VERSION: 4.4.0-210-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUACCT: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
```

```
[node4 ~]$ kubeadm join 192.168.0.18:6443 --token 0qf246.7qk5vpgzyjt4ga4l \
>         --discovery-token-ca-cert-hash sha256:13814c75f4af5addd9368d580dc7b8766703333c65f516c9e6f9a462e7a8da44
Initializing machine ID from random generator.
W0525 09:06:55.606926    1299 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet e
rrors in the future. Automatically prepending scheme "unix" to the "criSocket" with value "/run/docker/containerd/containerd.sock". Ple
ase update your configuration!
[preflight] Running pre-flight checks
        [WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kub
elet
[preflight] The system verification failed. Printing the output from the verification:
KERNEL_VERSION: 4.4.0-210-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUACCT: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
```

```
[node5 ~]$ kubeadm join 192.168.0.18:6443 --token 0qf246.7qk5vpgzyjt4ga4l \
>         --discovery-token-ca-cert-hash sha256:13814c75f4af5addd9368d580dc7b8766703333c65f516c9e6f9a462e7a8da44
Initializing machine ID from random generator.
W0525 09:07:01.797187    1311 initconfiguration.go:120] Usage of CRI endpoints without URL scheme is deprecated and can cause kubelet e
rrors in the future. Automatically prepending scheme "unix" to the "criSocket" with value "/run/docker/containerd/containerd.sock". Ple
ase update your configuration!
[preflight] Running pre-flight checks
        [WARNING Swap]: swap is enabled; production deployments should disable swap unless testing the NodeSwap feature gate of the kub
elet
[preflight] The system verification failed. Printing the output from the verification:
KERNEL_VERSION: 4.4.0-210-generic
OS: Linux
CGROUPS_CPU: enabled
CGROUPS_CPUACCT: enabled
CGROUPS_CPUSET: enabled
CGROUPS_DEVICES: enabled
CGROUPS_FREEZER: enabled
CGROUPS_MEMORY: enabled
CGROUPS_PIDS: enabled
CGROUPS_HUGETLB: enabled
CGROUPS_BLKIO: enabled
```

9

```
[node1 ~]$ kubectl get nodes
NAME    STATUS   ROLES           AGE     VERSION
node1   Ready    control-plane   3m41s   v1.27.2
node2   Ready    <none>          38s     v1.27.2
node3   Ready    <none>          29s     v1.27.2
node4   Ready    <none>          21s     v1.27.2
node5   Ready    <none>          15s     v1.27.2
```

10

```
[node1 ~]$ kubectl describe nodes node1
Name:               node1
Roles:              control-plane
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=node1
                    kubernetes.io/os=linux
                    node-role.kubernetes.io/control-plane=
                    node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/docker/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Sat, 25 May 2024 09:03:36 +0000
Taints:             node-role.kubernetes.io/control-plane:NoSchedule
Unschedulable:      false
Lease:
  HolderIdentity:  node1
  AcquireTime:     <unset>
  RenewTime:       Sat, 25 May 2024 09:07:31 +0000
Conditions:
  Type             Status  LastHeartbeatTime                 LastTransitionTime                Reason                       Message
  ----             ------  -----------------                 ------------------                ------                       -------
  MemoryPressure   False   Sat, 25 May 2024 09:04:07 +0000   Sat, 25 May 2024 09:03:34 +0000   KubeletHasSufficientMemory   kubelet has
 sufficient memory available
  DiskPressure     False   Sat, 25 May 2024 09:04:07 +0000   Sat, 25 May 2024 09:03:34 +0000   KubeletHasNoDiskPressure     kubelet has
 no disk pressure
  PIDPressure      False   Sat, 25 May 2024 09:04:07 +0000   Sat, 25 May 2024 09:03:34 +0000   KubeletHasSufficientPID      kubelet has
 sufficient PID available
  Ready            True    Sat, 25 May 2024 09:04:07 +0000   Sat, 25 May 2024 09:04:00 +0000   KubeletReady                 kubelet is
posting ready status
Addresses:
  InternalIP:  192.168.0.18
  Hostname:    node1
Capacity:
  cpu:                8
  ephemeral-storage:  65504Mi
  hugepages-1Gi:      0
  hugepages-2Mi:      1364Mi
  memory:             32946972Ki
  pods:               110
Allocatable:
  cpu:                8
  ephemeral-storage:  65504Mi
  hugepages-1Gi:      0
  hugepages-2Mi:      1364Mi
  memory:             31550236Ki
  pods:               110
System Info:
  Machine ID:                 f22d41f7139549838106caa79eef572e
  System UUID:                0BE8ED01-977E-1847-BF47-D180FF4EEC73
  Boot ID:                    7e47858a-6b84-421f-aa98-9609f7d294a0
  Kernel Version:             4.4.0-210-generic
  OS Image:                   CentOS Linux 7 (Core)
  Operating System:           linux
  Architecture:               amd64
  Container Runtime Version:  containerd://1.6.21
  Kubelet Version:            v1.27.2
  Kube-Proxy Version:         v1.27.2
PodCIDR:                      10.5.0.0/24
PodCIDRs:                     10.5.0.0/24
Non-terminated Pods:          (8 in total)
  Namespace                   Name                                CPU Requests  CPU Limits  Memory Requests  Memory Limits  Age
  ---------                   ----                                ------------  ----------  ---------------  -------------  ---
  kube-system                 coredns-5d78c9869d-jv5hx            100m (1%)     0 (0%)      70Mi (0%)        170Mi (0%)     3m45s
  kube-system                 coredns-5d78c9869d-vzq5m            100m (1%)     0 (0%)      70Mi (0%)        170Mi (0%)     3m45s
  kube-system                 etcd-node1                          100m (1%)     0 (0%)      100Mi (0%)       0 (0%)         3m59s
  kube-system                 kube-apiserver-node1                250m (3%)     0 (0%)      0 (0%)           0 (0%)         3m51s
  kube-system                 kube-controller-manager-node1       200m (2%)     0 (0%)      0 (0%)           0 (0%)         3m54s
  kube-system                 kube-proxy-pxcmg                    0 (0%)        0 (0%)      0 (0%)           0 (0%)         3m45s
  kube-system                 kube-router-czxvs                   250m (3%)     0 (0%)      250Mi (0%)       0 (0%)         3m44s
  kube-system                 kube-scheduler-node1                100m (1%)     0 (0%)      0 (0%)           0 (0%)         3m57s
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests     Limits
  --------           --------     ------
  cpu                1100m (13%)  0 (0%)
  memory             490Mi (1%)   340Mi (1%)
  ephemeral-storage  0 (0%)       0 (0%)
  hugepages-1Gi      0 (0%)       0 (0%)
  hugepages-2Mi      0 (0%)       0 (0%)
Events:
  Type     Reason                   Age                 From             Message
  ----     ------                   ----                ----             -------
  Normal   Starting                 3m43s               kube-proxy
  Normal   Starting                 4m6s                kubelet          Starting kubelet.
  Warning  InvalidDiskCapacity      4m6s                kubelet          invalid capacity 0 on image filesystem
  Normal   NodeHasSufficientMemory  4m6s (x8 over 4m6s)  kubelet          Node node1 status is now: NodeHasSufficientMemory
  Normal   NodeHasNoDiskPressure    4m6s (x8 over 4m6s)  kubelet          Node node1 status is now: NodeHasNoDiskPressure
  Normal   NodeHasSufficientPID     4m6s (x6 over 4m6s)  kubelet          Node node1 status is now: NodeHasSufficientPID
  Normal   NodeAllocatableEnforced  4m6s                kubelet          Updated Node Allocatable limit across pods
  Normal   RegisteredNode           3m46s               node-controller  Node node1 event: Registered Node node1 in Controller
```

11

```
[node1 ~]$ kubectl get pods -o wide -A
NAMESPACE     NAME                            READY   STATUS    RESTARTS   AGE     IP             NODE    NOMINATED NODE   READINESS GA
TES
kube-system   coredns-5d78c9869d-jv5hx        1/1     Running   0          3m57s   10.5.0.4       node1   <none>           <none>
kube-system   coredns-5d78c9869d-vzq5m        1/1     Running   0          3m57s   10.5.0.3       node1   <none>           <none>
kube-system   etcd-node1                      1/1     Running   0          4m11s   192.168.0.18   node1   <none>           <none>
kube-system   kube-apiserver-node1            1/1     Running   0          4m3s    192.168.0.18   node1   <none>           <none>
kube-system   kube-controller-manager-node1   1/1     Running   0          4m6s    192.168.0.18   node1   <none>           <none>
kube-system   kube-proxy-66zqx                1/1     Running   0          47s     192.168.0.14   node5   <none>           <none>
kube-system   kube-proxy-8jl7f                1/1     Running   0          61s     192.168.0.16   node3   <none>           <none>
kube-system   kube-proxy-gqsvv                1/1     Running   0          53s     192.168.0.15   node4   <none>           <none>
kube-system   kube-proxy-p74cl                1/1     Running   0          70s     192.168.0.17   node2   <none>           <none>
kube-system   kube-proxy-pxcmg                1/1     Running   0          3m57s   192.168.0.18   node1   <none>           <none>
kube-system   kube-router-8vb8k               1/1     Running   0          53s     192.168.0.15   node4   <none>           <none>
kube-system   kube-router-czxvs               1/1     Running   0          3m56s   192.168.0.18   node1   <none>           <none>
kube-system   kube-router-mtsl4               1/1     Running   0          61s     192.168.0.16   node3   <none>           <none>
kube-system   kube-router-q868r               1/1     Running   0          47s     192.168.0.14   node5   <none>           <none>
kube-system   kube-router-w7chf               1/1     Running   0          70s     192.168.0.17   node2   <none>           <none>
kube-system   kube-scheduler-node1            1/1     Running   0          4m9s    192.168.0.18   node1   <none>           <none>
```

12

```
[node1 ~]$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/nginx-app.yaml
service/my-nginx-svc created
deployment.apps/my-nginx created
[node1 ~]$ kubectl get pods -o wide
NAME                        READY   STATUS    RESTARTS   AGE   IP          NODE    NOMINATED NODE   READINESS GATES
my-nginx-cbdccf466-2frn4    1/1     Running   0          27s   10.5.2.2    node3   <none>           <none>
my-nginx-cbdccf466-jblcn    1/1     Running   0          27s   10.5.1.2    node2   <none>           <none>
my-nginx-cbdccf466-pzqcm    1/1     Running   0          27s   10.5.4.2    node5   <none>           <none>
[node1 ~]$
```

13

```
[node1 ~]$ kubectl get svc
NAME           TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes     ClusterIP      10.96.0.1      <none>        443/TCP        18m
my-nginx-svc   LoadBalancer   10.104.76.49   <pending>     80:31767/TCP   13m
[node1 ~]$
```

14

```
[node1 ~]$ curl 192.168.0.18:31767
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
```

15

```
[node1 ~]$ kubectl drain node2 --ignore-daemonsets --delete-local-data
Flag --delete-local-data has been deprecated, This option is deprecated and will be deleted. Use --delete-emptydir-data.
node/node2 cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/kube-proxy-p74cl, kube-system/kube-router-w7chf
evicting pod default/my-nginx-cbdccf466-jblcn
pod/my-nginx-cbdccf466-jblcn evicted
node/node2 drained
[node1 ~]$ kubectl drain node3 --ignore-daemonsets --delete-local-data
Flag --delete-local-data has been deprecated, This option is deprecated and will be deleted. Use --delete-emptydir-data.
node/node3 cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/kube-proxy-8jl7f, kube-system/kube-router-mtsl4
evicting pod default/my-nginx-cbdccf466-2frn4
pod/my-nginx-cbdccf466-2frn4 evicted
node/node3 drained
[node1 ~]$ kubectl drain node5 --ignore-daemonsets --delete-local-data
Flag --delete-local-data has been deprecated, This option is deprecated and will be deleted. Use --delete-emptydir-data.
node/node5 cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/kube-proxy-66zqx, kube-system/kube-router-q868r
evicting pod default/my-nginx-cbdccf466-pzqcm
pod/my-nginx-cbdccf466-pzqcm evicted
node/node5 drained
[node1 ~]$
[node1 ~]$ kubectl get pods -o wide
NAME                        READY   STATUS    RESTARTS   AGE   IP          NODE    NOMINATED NODE   READINESS GATES
my-nginx-cbdccf466-4qks5    1/1     Running   0          72s   10.5.3.3    node4   <none>           <none>
my-nginx-cbdccf466-ftq6w    1/1     Running   0          62s   10.5.3.4    node4   <none>           <none>
my-nginx-cbdccf466-shld2    1/1     Running   0          81s   10.5.3.2    node4   <none>           <none>
[node1 ~]$
```

16,17

```
[node1 ~]$ kubectl delete -f https://raw.githubusercontent.com/kubernetes/website/master/content/en/examples/application/nginx-app.yaml
service "my-nginx-svc" deleted
deployment.apps "my-nginx" deleted
[node1 ~]$ kubectl get pods -o wide
No resources found in default namespace.
[node1 ~]$
```

Ответы на вопросы.

Как узнать ip-адрес ноды подключённой к кластеру по её имени? Вы можете использовать команду kubectl get nodes -o wide для получения информации о нодах, включая их IP-адреса.

Какой предварительный этап нужно выполнить, чтобы иметь возможность запустить приложение в кластере из исходников? Перед запуском приложения в кластере Kubernetes из исходного кода, вам нужно выполнить следующие шаги: Упаковать ваше приложение в контейнеры. Определить желаемое состояние вашего приложения с использованием манифестов. Загрузить ваш код приложения на платформу SCM, такую как GitHub. Использовать инструмент CI/CD для автоматизации развертывания вашего приложения.

Что нужно знать, чтобы подключить рабочую ноду к кластеру? Для подключения рабочей ноды к кластеру Kubernetes, вам нужно выполнить следующие шаги: Сгенерировать новый токен присоединения на мастер-ноде с помощью команды kubeadm token create. Присоединить новую рабочую ноду к кластеру с использованием токена присоединения и IP-адреса и порта мастер-ноды.

Может ли быть в одном кластере несколько мастер-нод? Да, в кластере Kubernetes может быть несколько мастер-нод. Это обеспечивает высокую доступность и защиту от отказов.

Может ли существовать кластер состоящий только из мастер ноды, которая одновременно является и рабочей нодой? Да, в теории, мастер-нода может также выполнять функции рабочей ноды и запускать рабочие нагрузки. Однако это не рекомендуется для производственных сред, главным образом из-за проблем производительности. В обучающих или ресурсно-ограниченных средах, вы можете иметь только одну ноду, которая выполняет функции и мастер-ноды, и рабочей ноды.

Вывод: Я ознакомился на практике с инструментом оркестрации контейнерирированных приложений Kubernetes.