

ЗАДАНИЕ № 3

1) Создайте namespace kubedoom “`kubectl create ns kubedoom`”. Напишите deployment для запуска игры Kube DOOM. Приложение должно запускаться из образа `storaxdev/kubedoom:0.5.0`. Должен быть описан порт: 5900 TCP. Для указания протокола используется поле `protocol` в описании порта. В деплойменте должна быть одна реплика, при этом при обновлении образа не должно одновременно работать две реплики (см. `maxSurge` и `maxUnavailable` из лекции).

Добавьте в шаблон контейнера параметры:

`hostNetwork: true`

`serviceAccountName: kubedoom`

Запустите получившийся деплоймент в кластере Kubernetes в namespace `kubedoom`. Pod не должен самопроизвольно рестартовать. В случае возникновения проблем смотрите в описание Pod, ReplicaSet, Deployment. Например:

`kubectl describe pod <pod name>`

`kubectl logs pod <pod name>`

Разверните в кластере манифест:

`apiVersion: v1`

`kind: ServiceAccount`

`metadata:`

`name: kubedoom`

`namespace: kubedoom`

`apiVersion: rbac.authorization.k8s.io/v1`

`kind: ClusterRoleBinding`

`metadata:`

`name: kubedoom`

`roleRef:`

`apiGroup: rbac.authorization.k8s.io`

`kind: ClusterRole`

`name: cluster-admin`

`subjects:`

`- kind: ServiceAccount`

`name: kubedoom`

`namespace: kubedoom`

Этот манифест создаст в кластере сервисную учетную запись и даст ей права Cluster-admin

Для подключения к игре вам нужно выполнить `kubectl portforward` и используйте VNC клиент.

Подробнее о KubeDoom читайте по ссылке: <https://github.com/storax/kubedoom>

РЕШЕНИЕ:

1) Создадим namespace kubedoom: `kubectl apply -f namespace.yaml`

Создаем сервисную учетную запись: `kubectl apply -f sa.yaml`

Проверим УЗ: `kubectl --namespace kubedoom get serviceaccount`

Создаем deployment: `kubectl apply -f deployment.yaml`

Убедимся, что поды поднялись: `kubectl --namespace kubedoom get pods`

`kubectl --namespace kubedoom describe pods kubedoom-57b69fdb-c-jz9s4`

```

root@alexi-Virtual-Machine:/home/alexi# kubectl --namespace kubedoom describe pods kubedoom-57b69fdb-jz9s4
Name: kubedoom-57b69fdb-jz9s4
Namespace: kubedoom
Priority: 0
Node: kub1-default1-0/10.0.0.4
Start Time: Wed, 27 Apr 2022 23:38:33 +0300
Labels: app=kubedoom
         pod-template-hash=57b69fdb
Annotations: kubernet.es.io/limit-ranger: LimitRanger plugin set: cpu, memory request for container kubedoom; cpu
Status: Running
IP: 10.0.0.4
IPs:
  IP: 10.0.0.4
Controlled By: ReplicaSet/kubedoom-57b69fdb
Containers:
  kubedoom:
    Container ID: cri-o://5618aa7855a96d1c7ee81462572ab7dce2d86e885567446d7366b92bcb245d53
    Image: storaxdev/kubedoom:0.5.0
    Image ID: docker.io/storaxdev/kubedoom@sha256:88dfe1b5430a678a8f399593fd9b2d256c86b29518a6924965fe1f175
    Port: 5900/TCP
    Host Port: 5900/TCP
    State: Running
      Started: Wed, 27 Apr 2022 23:38:34 +0300
    Ready: True
    Restart Count: 0
    Limits:
      cpu: 500m
      memory: 512Mi
    Requests:
      cpu: 100m
      memory: 64Mi
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sv22q (ro)
Conditions:
  Type           Status
  Initialized    True
  Ready          False
  ContainersReady True
  PodScheduled   True

```

Пробросим порт для VNC: `kubectl --namespace=kubedoom port-forward kubedoom-57b69fdb-jz9s4 5901:5900`

```

root@alexi-Virtual-Machine:/home/alexi# kubectl --namespace=kubedoom port-forward kubedoom-57b69fdb-jz9s4 5901:5900
Forwarding from 127.0.0.1:5901 -> 5900
Forwarding from [::1]:5901 -> 5900

```

Пароль для подключения: idbehold

