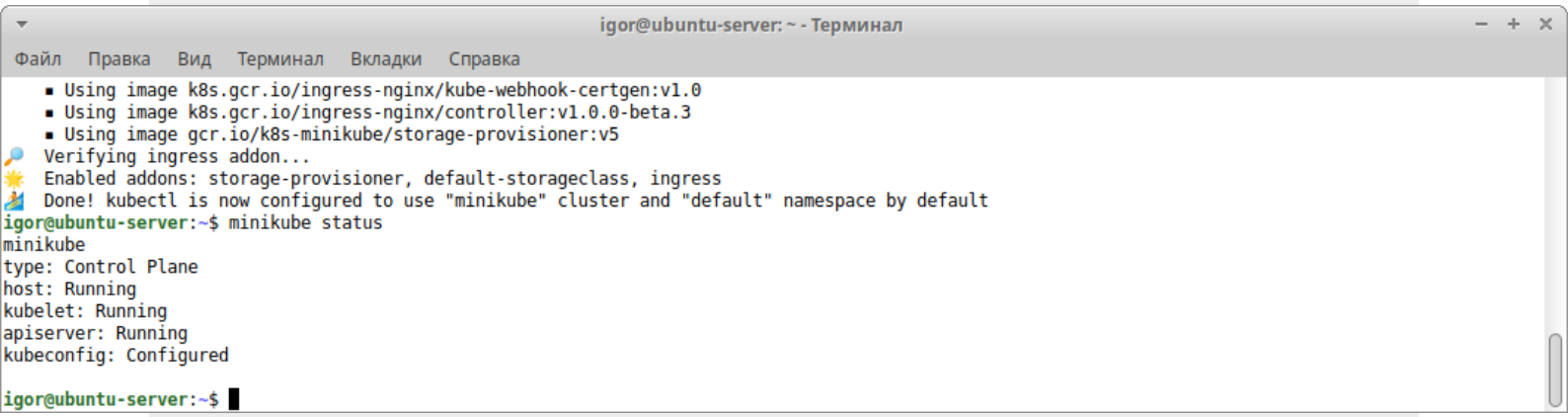
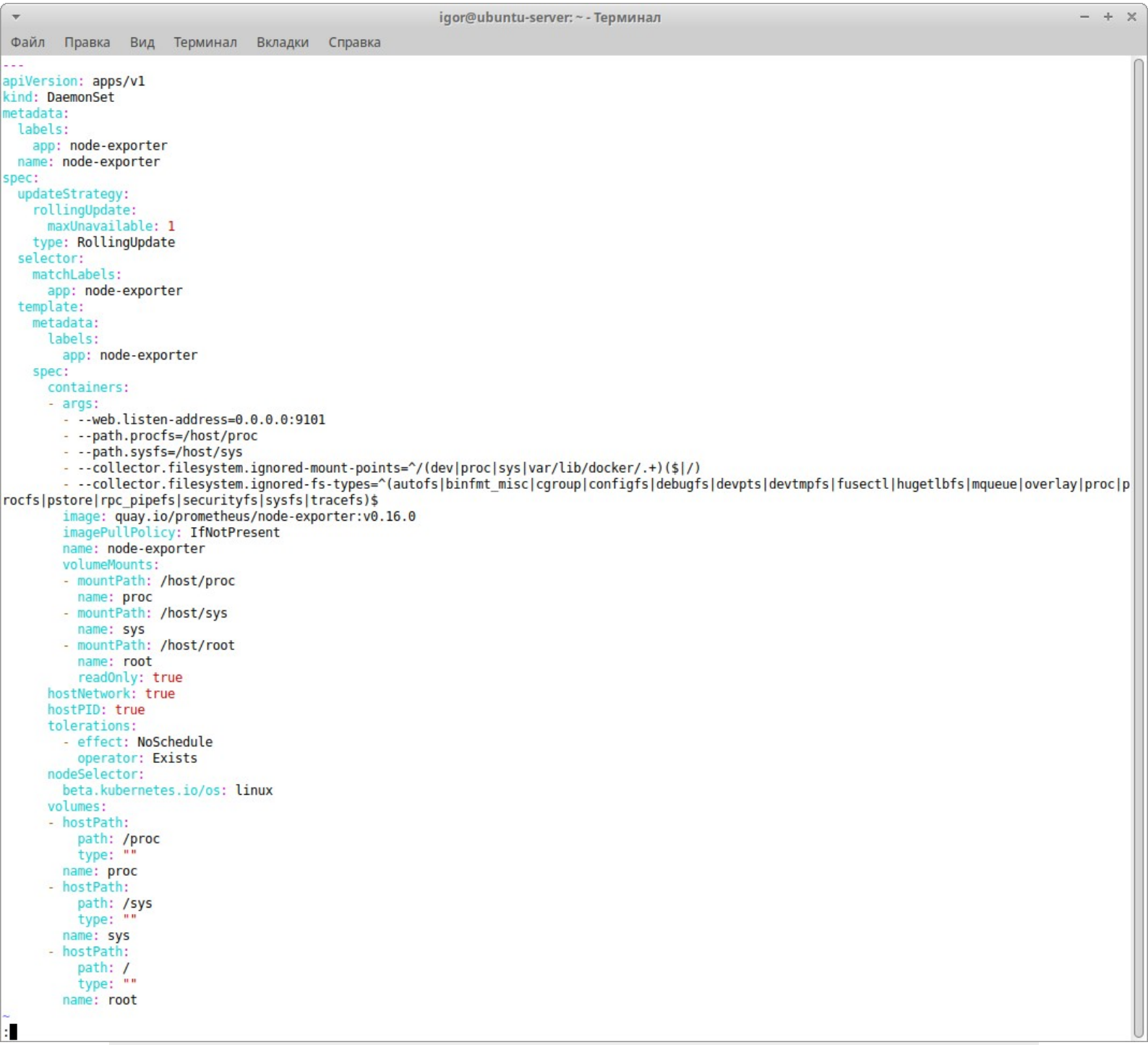
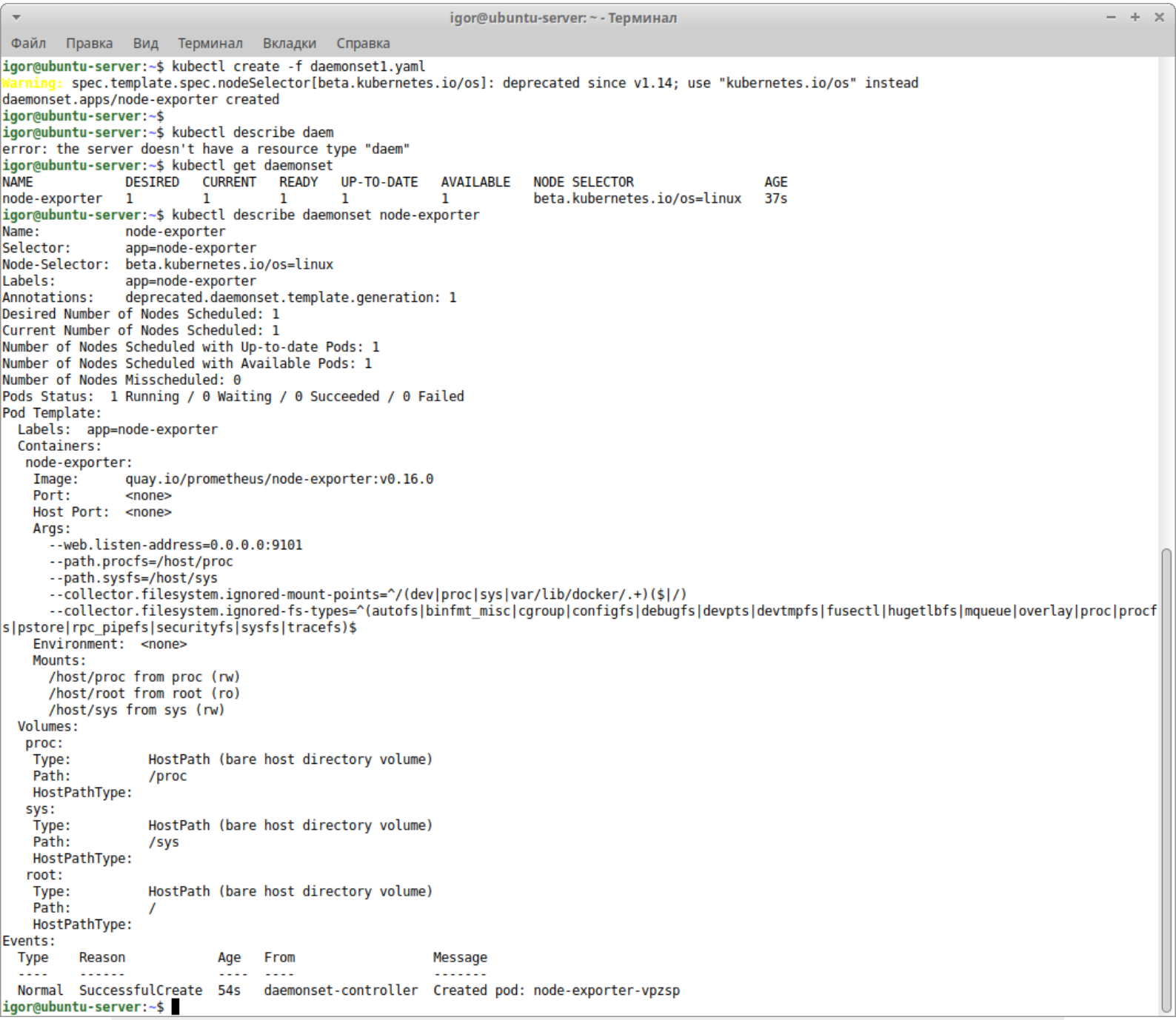
```
minikube start

minikube status
```




```
vi daemonset1.yaml
```



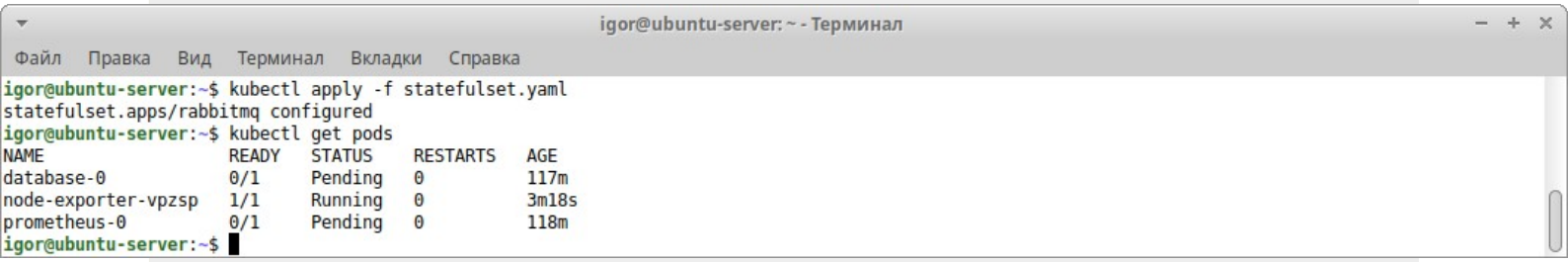

```
kubectl create -f daemonset1.yaml

Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated
since v1.14; use "kubernetes.io/os" instead

Error from server (AlreadyExists): error when creating "daemonset1.yaml":
daemonsets.apps "node-exporter" already exists

kubectl get daemonset

kubectl describe daemonset node-exporter
```



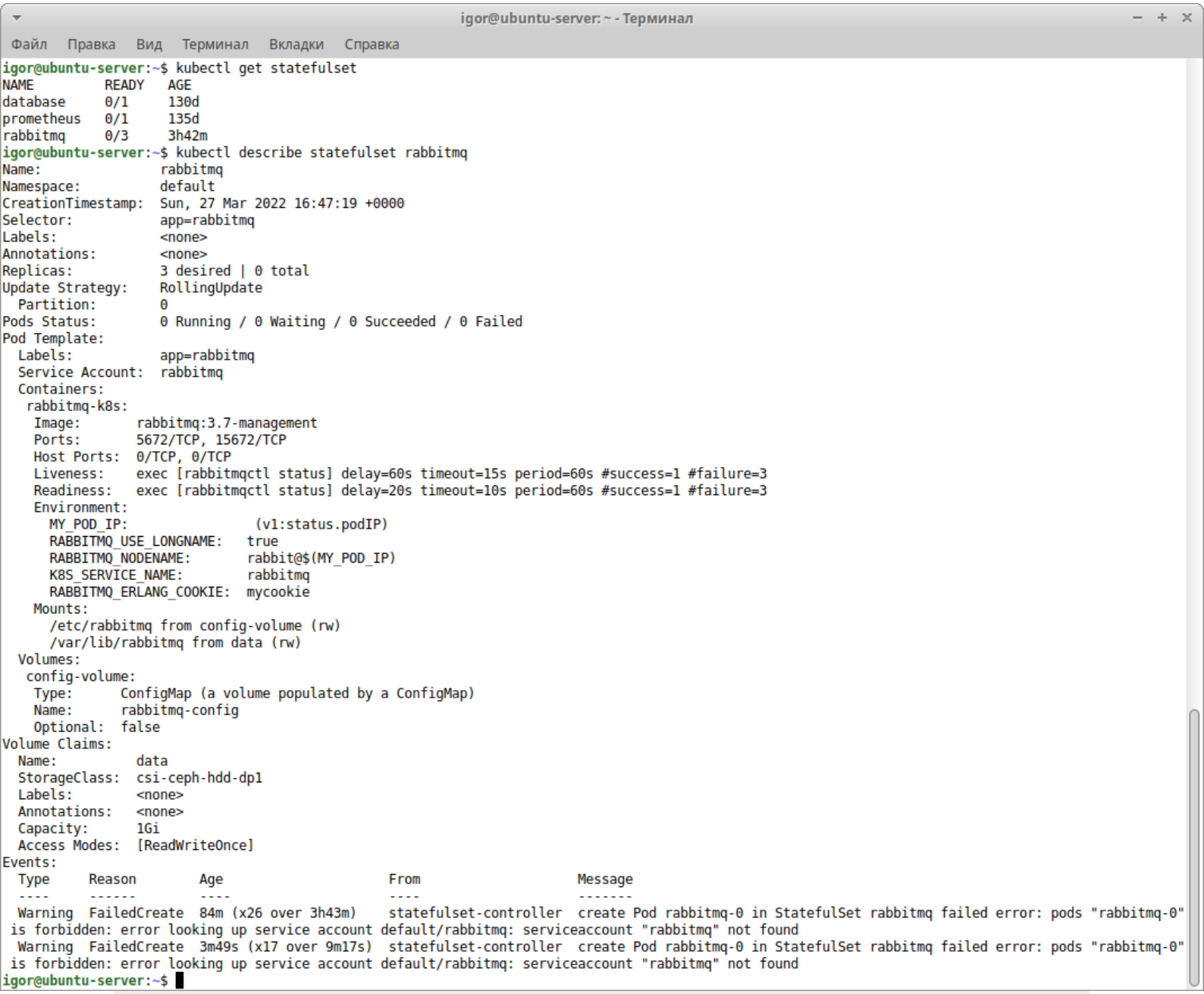

```
vi statefulset.yaml
```

```
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: rabbitmq
spec:
  serviceName: rabbitmq
  replicas: 3
  selector:
    matchLabels:
      app: rabbitmq
  template:
    metadata:
      labels:
        app: rabbitmq
    spec:
      serviceAccount: rabbitmq
      terminationGracePeriodSeconds: 10
      containers:
        - name: rabbitmq-k8s
          image: rabbitmq:3.7-management
          env:
            - name: MY_POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
            - name: RABBITMQ_USE_LONGNAME
              value: "true"
            - name: RABBITMQ_NODENAME
              value: "rabbit@$(MY_POD_IP)"
            - name: K8S_SERVICE_NAME
              value: "rabbitmq"
            - name: RABBITMQ_ERLANG_COOKIE
              value: "mycookie"
          ports:
            - name: amqp
              protocol: TCP
              containerPort: 5672
            - name: admin
              protocol: TCP
              containerPort: 15672
          livenessProbe:
            exec:
              command: ["rabbitmqctl", "status"]
            initialDelaySeconds: 60
            periodSeconds: 60
            timeoutSeconds: 15
          readinessProbe:
            exec:
              command: ["rabbitmqctl", "status"]
            initialDelaySeconds: 20
            periodSeconds: 60
            timeoutSeconds: 10
          imagePullPolicy: Always
          volumeMounts:
            - name: config-volume
              mountPath: /etc/rabbitmq
            - name: data
              mountPath: /var/lib/rabbitmq
      volumes:
        - name: config-volume
          configMap:
            name: rabbitmq-config
            items:
              - key: rabbitmq.conf
                path: rabbitmq.conf
              - key: enabled_plugins
                path: enabled_plugins
      affinity:
        podAntiAffinity:
          preferredDuringSchedulingIgnoredDuringExecution:
            - weight: 100
              podAffinityTerm:
                labelSelector:
                  matchExpressions:
                    - key: app
                      operator: In
                      values:
                        - rabbitmq
                topologyKey: kubernetes.io/hostname
  volumeClaimTemplates:
    - metadata:
        name: data
      spec:
        accessModes: ["ReadWriteOnce"]
        resources:
          requests:
            storage: 1Gi
        storageClassName: csi-ceph-hdd-dp1
~
:
```

```
kubectl apply -f statefulset.yaml

kubectl get pods
```



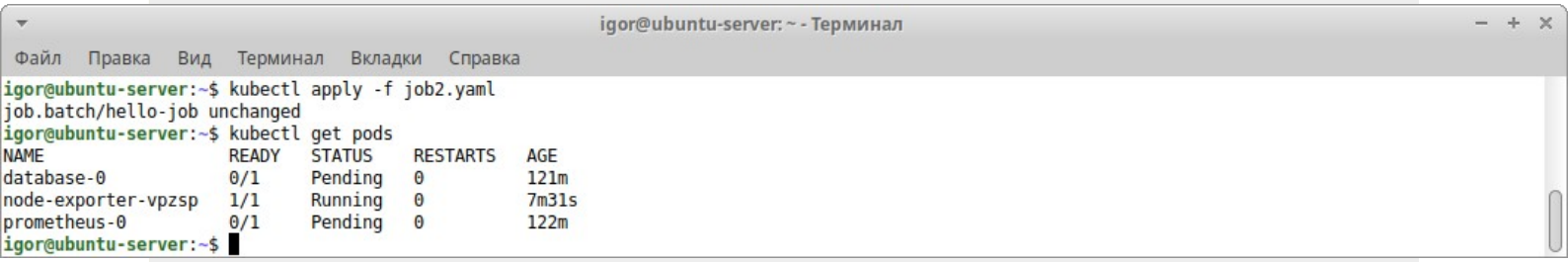

```
kubectl get statefulset

kubectl describe statefulset rabbitmq
```


igor@ubuntu-server: ~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
igor@ubuntu-server:~$ kubectl get statefulset
NAME READY AGE
database 0/1 130d
prometheus 0/1 135d
rabbitmq 0/3 3h42m
igor@ubuntu-server:~$ kubectl describe statefulset rabbitmq
Name: rabbitmq
Namespace: default
CreationTimestamp: Sun, 27 Mar 2022 16:47:19 +0000
Selector: app=rabbitmq
Labels: <none>
Annotations: <none>
Replicas: 3 desired | 0 total
Update Strategy: RollingUpdate
Partition: 0
Pods Status: 0 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
Labels: app=rabbitmq
Service Account: rabbitmq
Containers:
rabbitmq-k8s:
Image: rabbitmq:3.7-management
Ports: 5672/TCP, 15672/TCP
Host Ports: 0/TCP, 0/TCP
Liveness: exec [rabbitmqctl status] delay=60s timeout=15s period=60s #success=1 #failure=3
Readiness: exec [rabbitmqctl status] delay=20s timeout=10s period=60s #success=1 #failure=3
Environment:
MY_POD_IP: (v1:status.podIP)
RABBITMQ_USE_LONGNAME: true
RABBITMQ_NODENAME: rabbit@$(MY_POD_IP)
K8S_SERVICE_NAME: rabbitmq
RABBITMQ_ERLANG_COOKIE: mycookie
Mounts:
/etc/rabbitmq from config-volume (rw)
/var/lib/rabbitmq from data (rw)
Volumes:
config-volume:
Type: ConfigMap (a volume populated by a ConfigMap)
Name: rabbitmq-config
Optional: false
Volume Claims:
Name: data
StorageClass: csi-ceph-hdd-dp1
Labels: <none>
Annotations: <none>
Capacity: 1Gi
Access Modes: [ReadWriteOnce]
Events:
Type Reason Age From Message
Warning FailedCreate 84m (x26 over 3h43m) statefulset-controller create Pod rabbitmq-0 in StatefulSet rabbitmq failed error: pods "rabbitmq-0" is forbidden: error looking up service account default/rabbitmq: serviceaccount "rabbitmq" not found
Warning FailedCreate 3m49s (x17 over 9m17s) statefulset-controller create Pod rabbitmq-0 in StatefulSet rabbitmq failed error: pods "rabbitmq-0" is forbidden: error looking up service account default/rabbitmq: serviceaccount "rabbitmq" not found
igor@ubuntu-server:~$

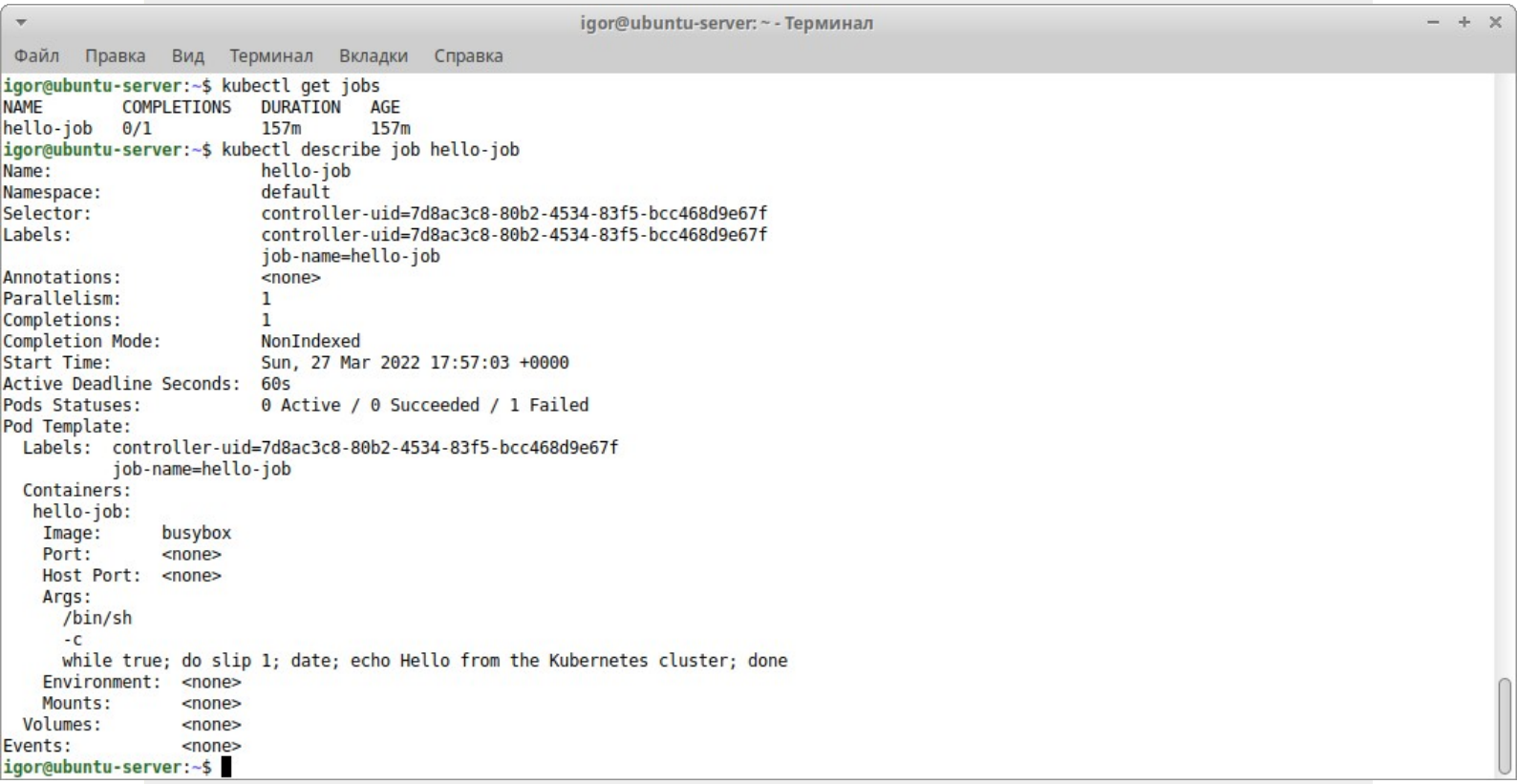
```
vi job2.yaml
```

igor@ubuntu-server: ~ - Терминал
Файл Правка Вид Терминал Вкладки Справка
---
apiVersion: batch/v1
kind: Job
metadata:
name: hello-job
spec:
backoffLimit: 2
activeDeadlineSeconds: 60
template:
spec:
containers:
- name: hello-job
image: busybox
args:
- /bin/sh
- -c
- while true; do slip 1; date; echo Hello from the Kubernetes cluster; done
restartPolicy: Never
:

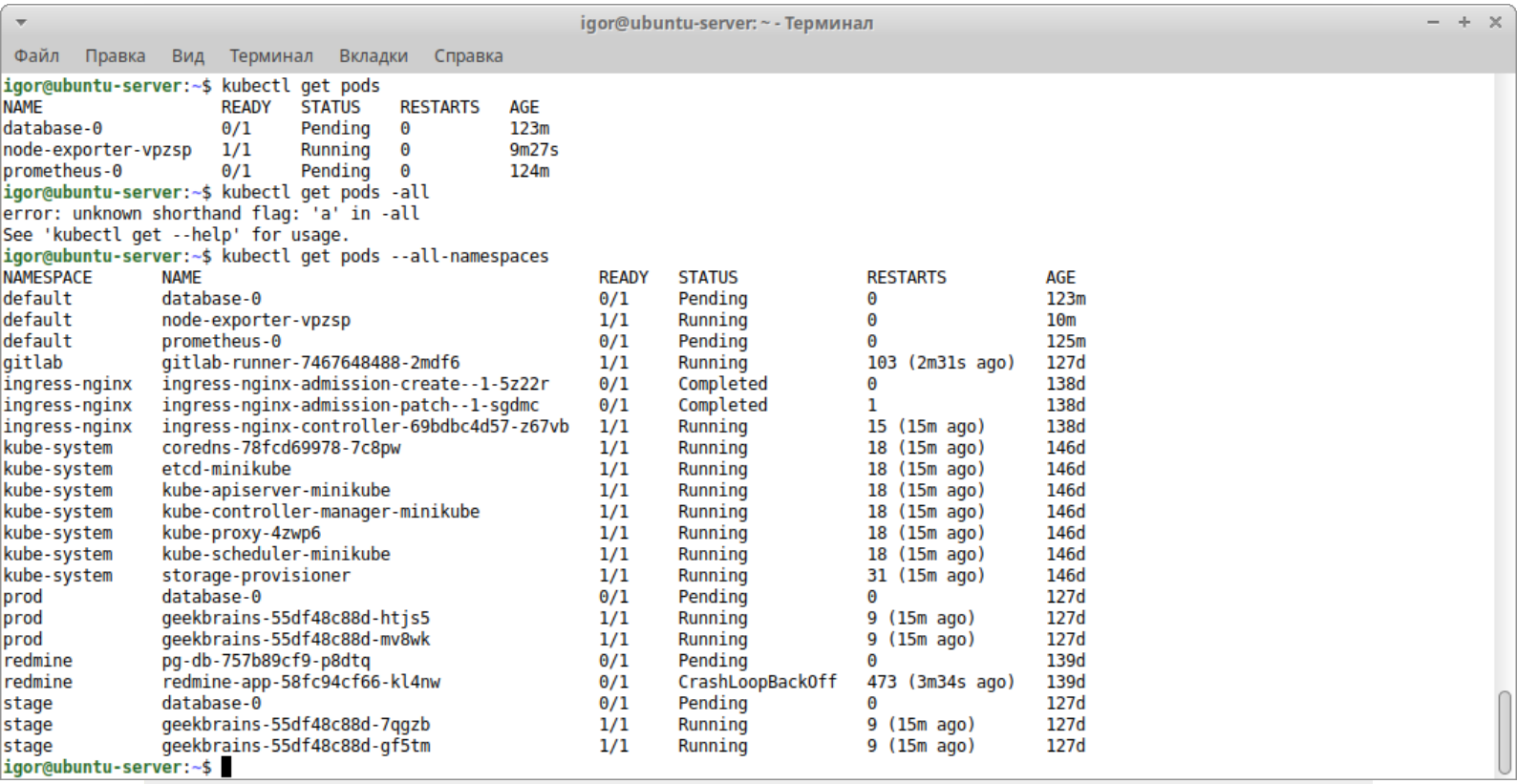
```
kubectl apply -f job2.yaml

kubectl get pods
```



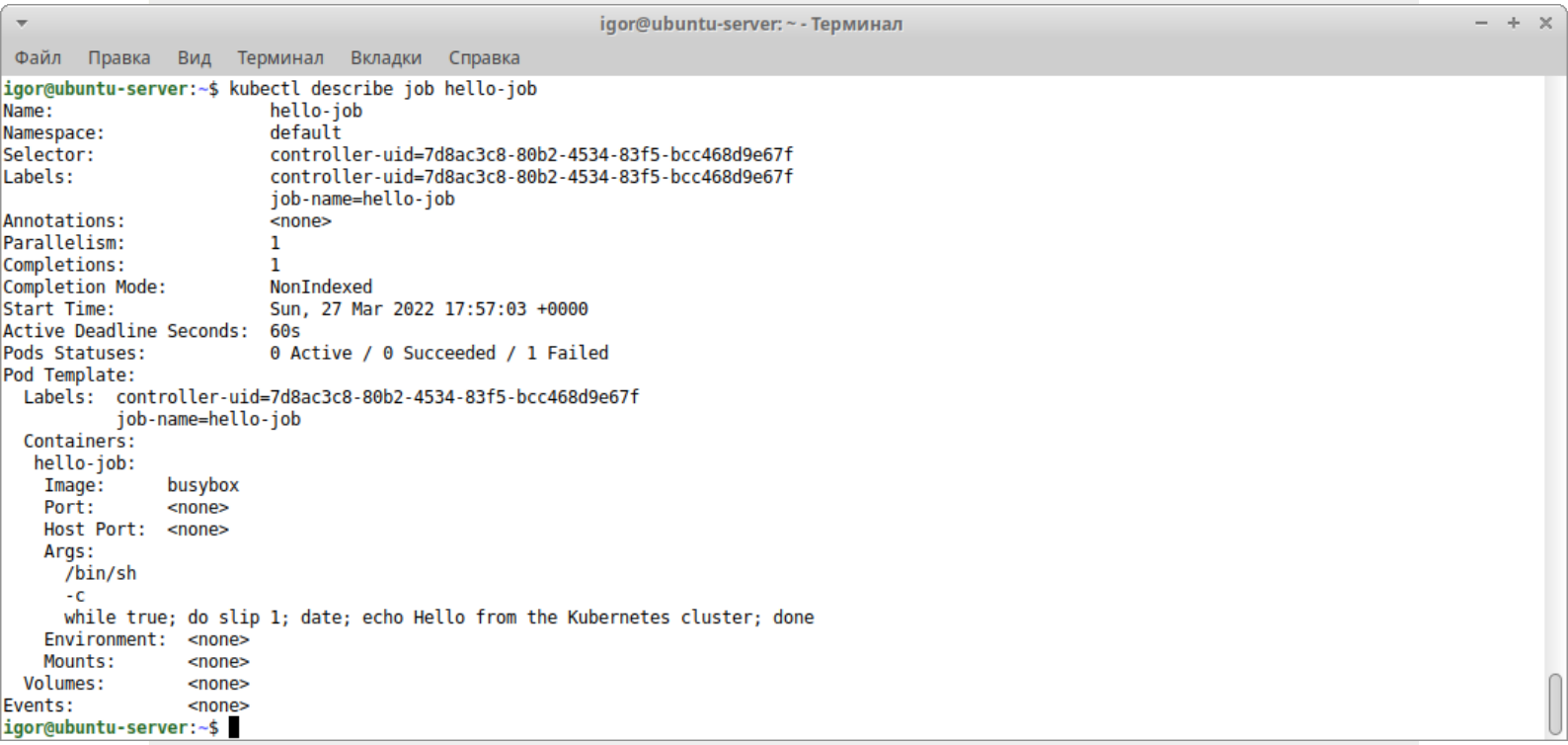

```
kubectl get jobs

kubectl describe job hello-job
```



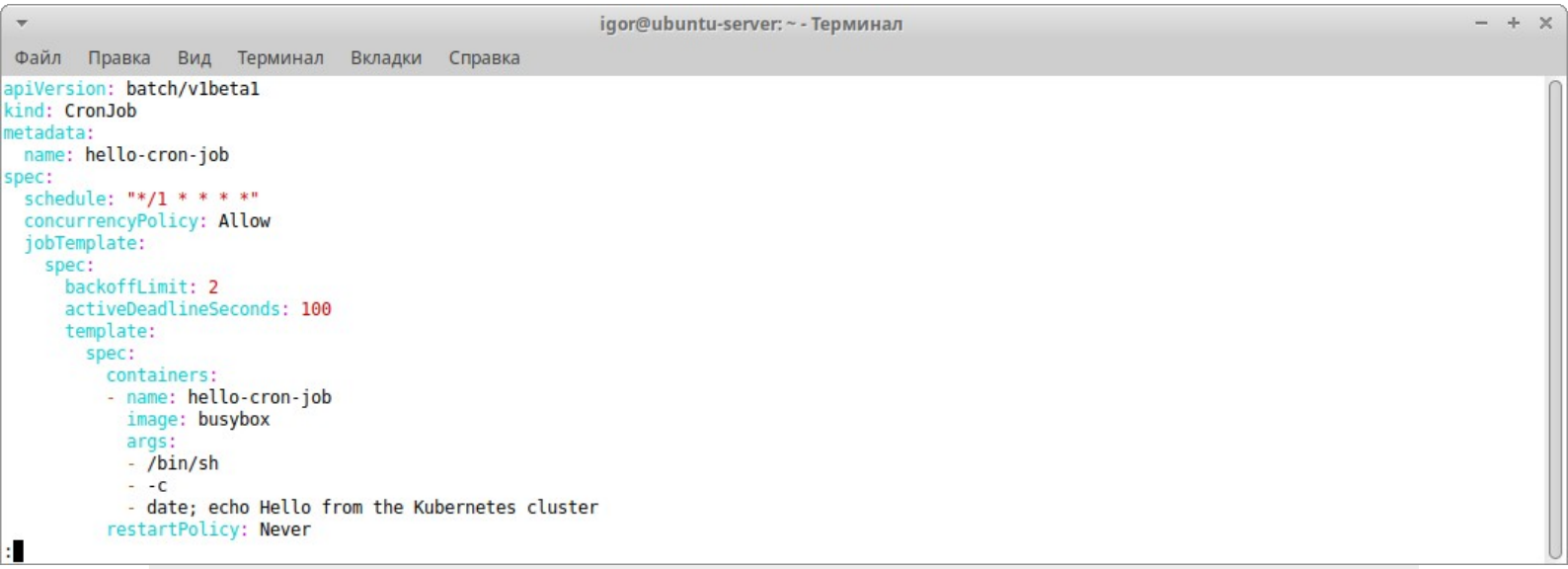

```
kubectl get pods

kubectl get pods -all

kubectl get pods --all-namespaces
```



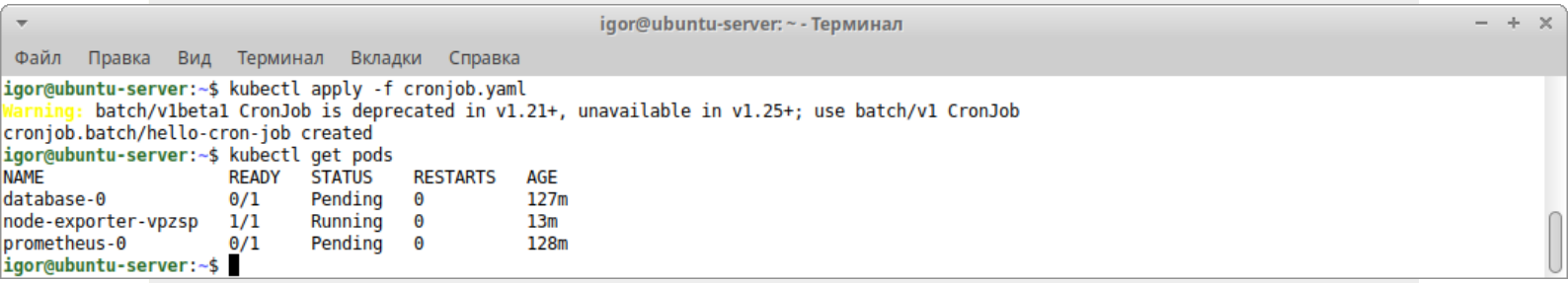

```
kubectl describe job hello-job
```



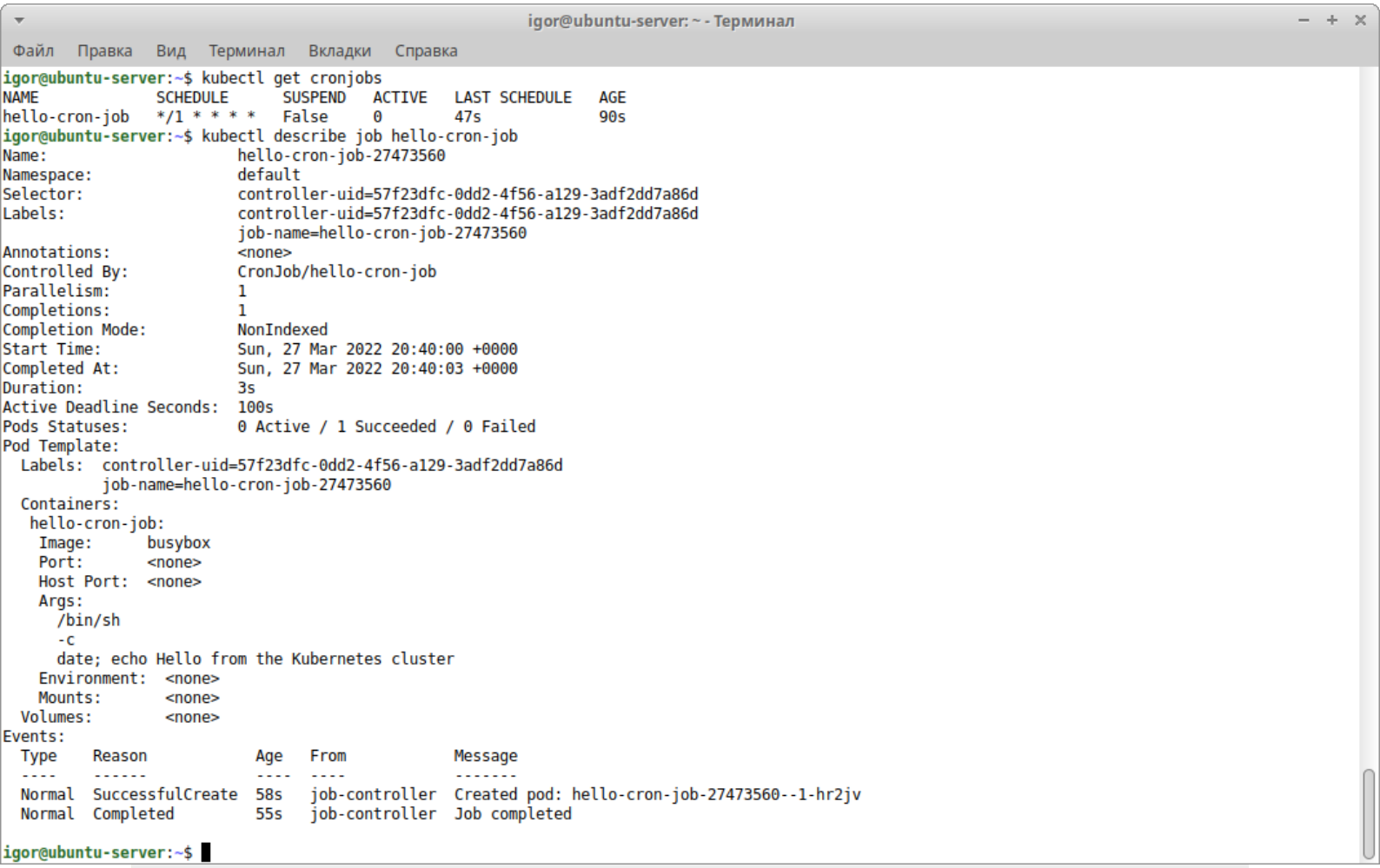

```
vi cronjob.yaml
```



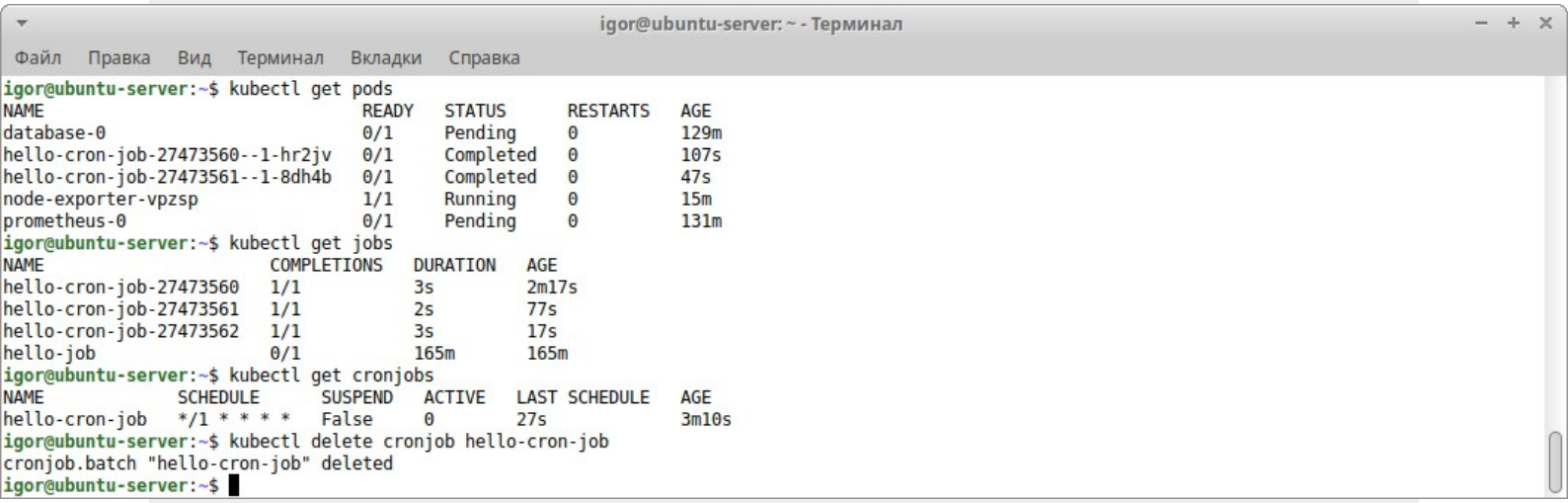

```
kubectl apply -f cronjob.yaml

Warning: batch/v1beta1 CronJob is deprecated in v1.21+, unavailable in v1.25+;
use batch/v1 CronJob

cronjob.batch/hello-cron-job created

kubectl get pods
```



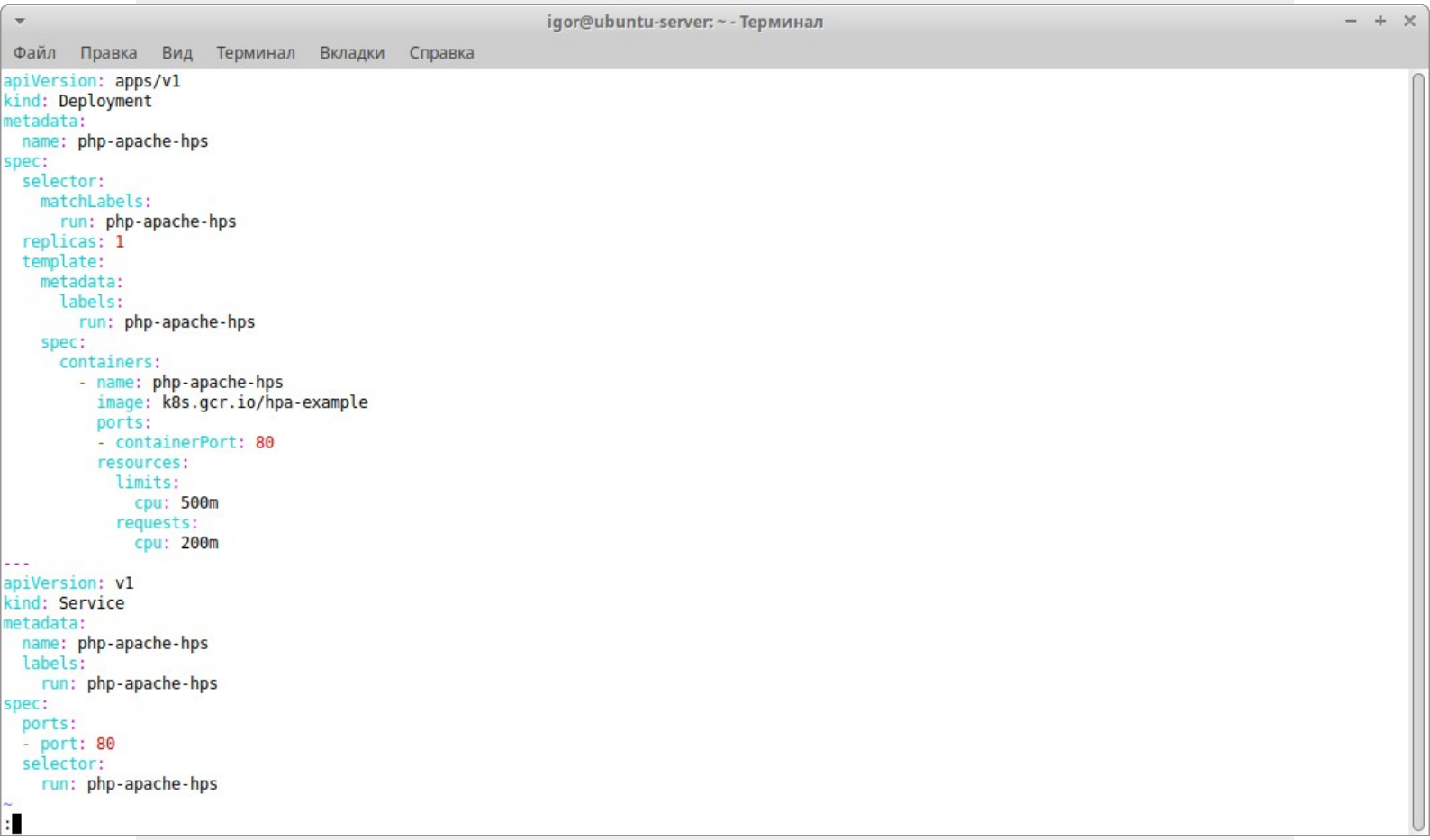

```
kubectl get cronjobs

kubectl describe job hello-cron-job
```



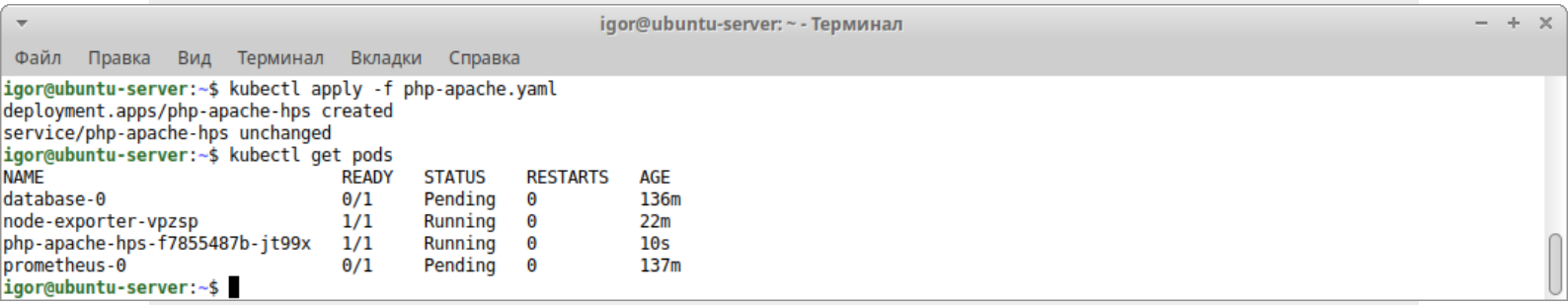

```
kubectl get pods

kubectl get jobs

kubectl get cronjobs

kubectl delete cronjob hello-cron-job
```



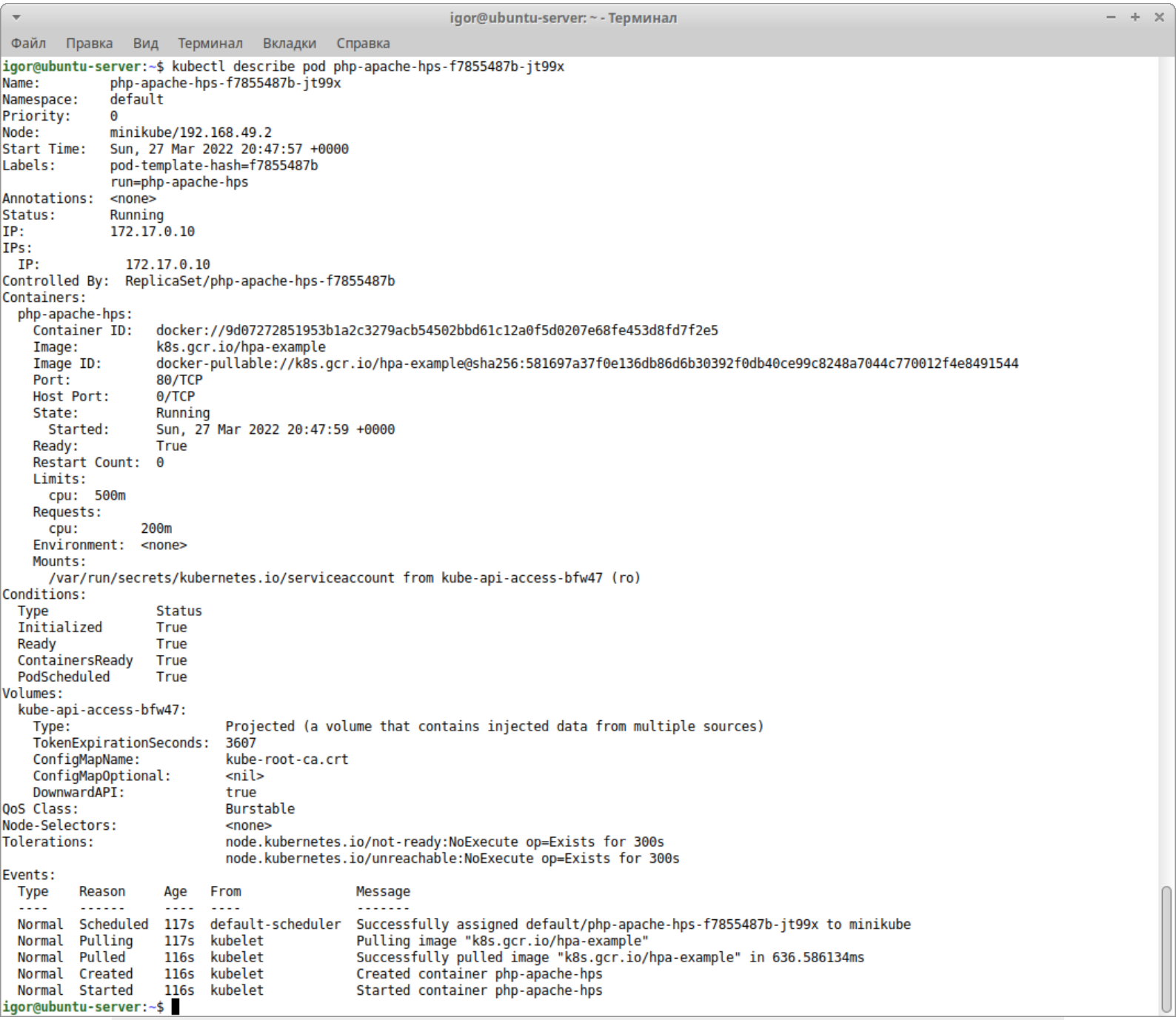
```
vi php-apache.yaml
```



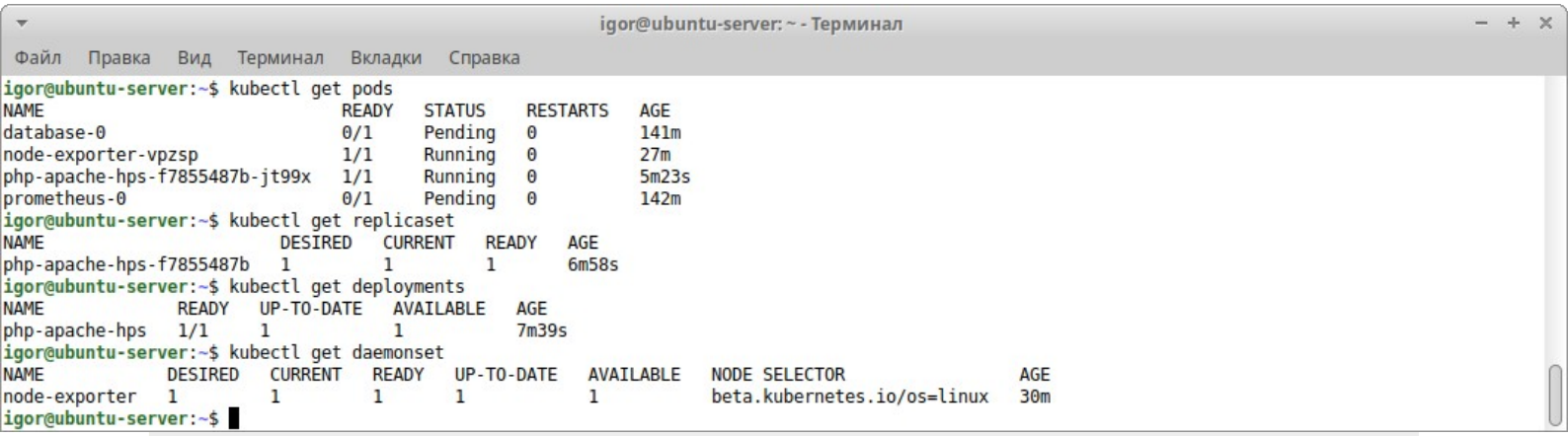

```
kubectl apply -f php-apache.yaml

kubectl get pods
```



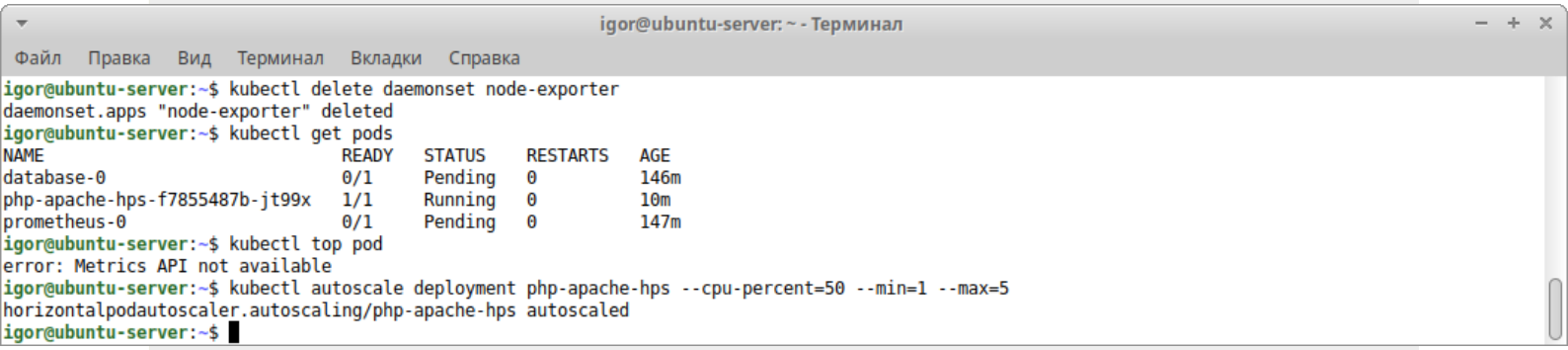

```
kubectl describe pod php-apache-hps-f7855487b-jt99x
```



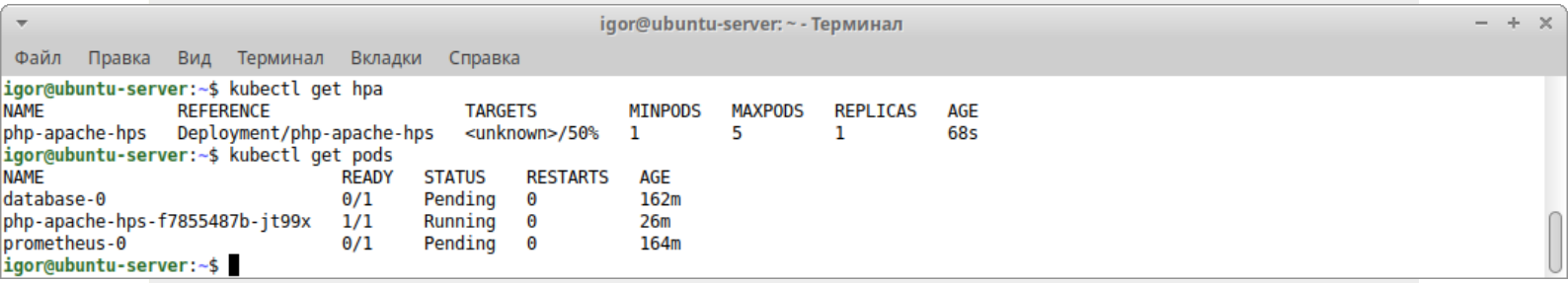

```
kubectl get pods

kubectl get replicaset

kubectl get deployments

kubectl get daemonset
```




```
kubectl delete daemonset node-exporter

kubectl get pods

kubectl top pod

kubectl autoscale deployment php-apache-hps --cpu-percent=50 --
min=1 --max=5
```

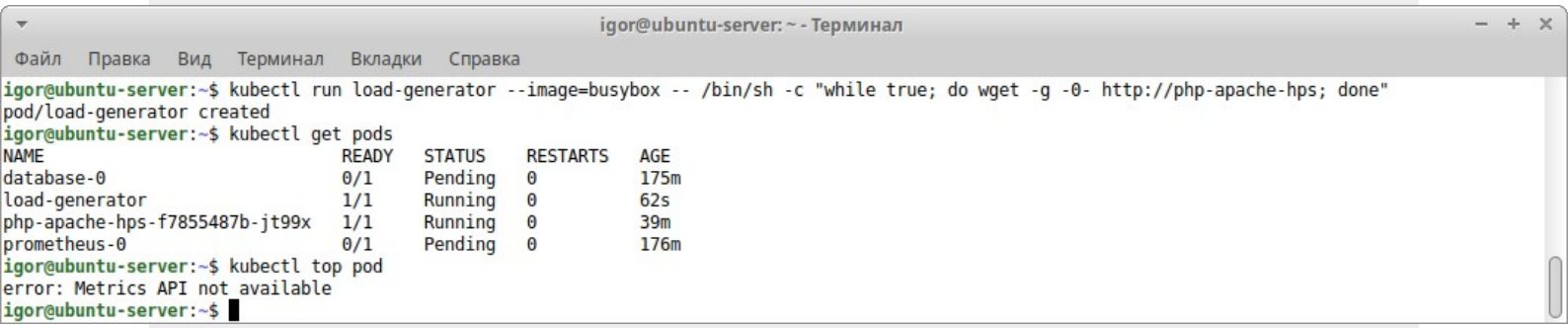



```
kubectl get hpa

kubectl get pods
```

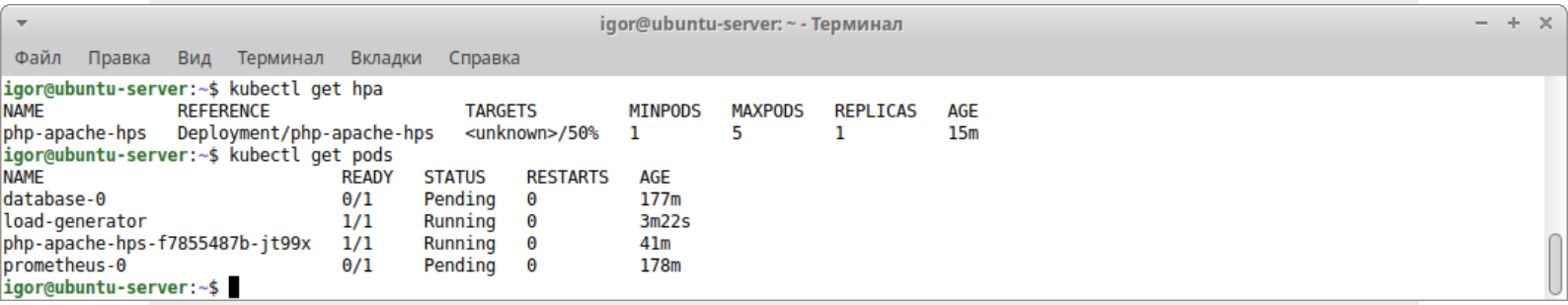



```
kubectl run load-generator --image=busybox -- /bin/sh -c "while true; do wget -
g -0- http://php-apache-hps; done"

kubectl get pods
```

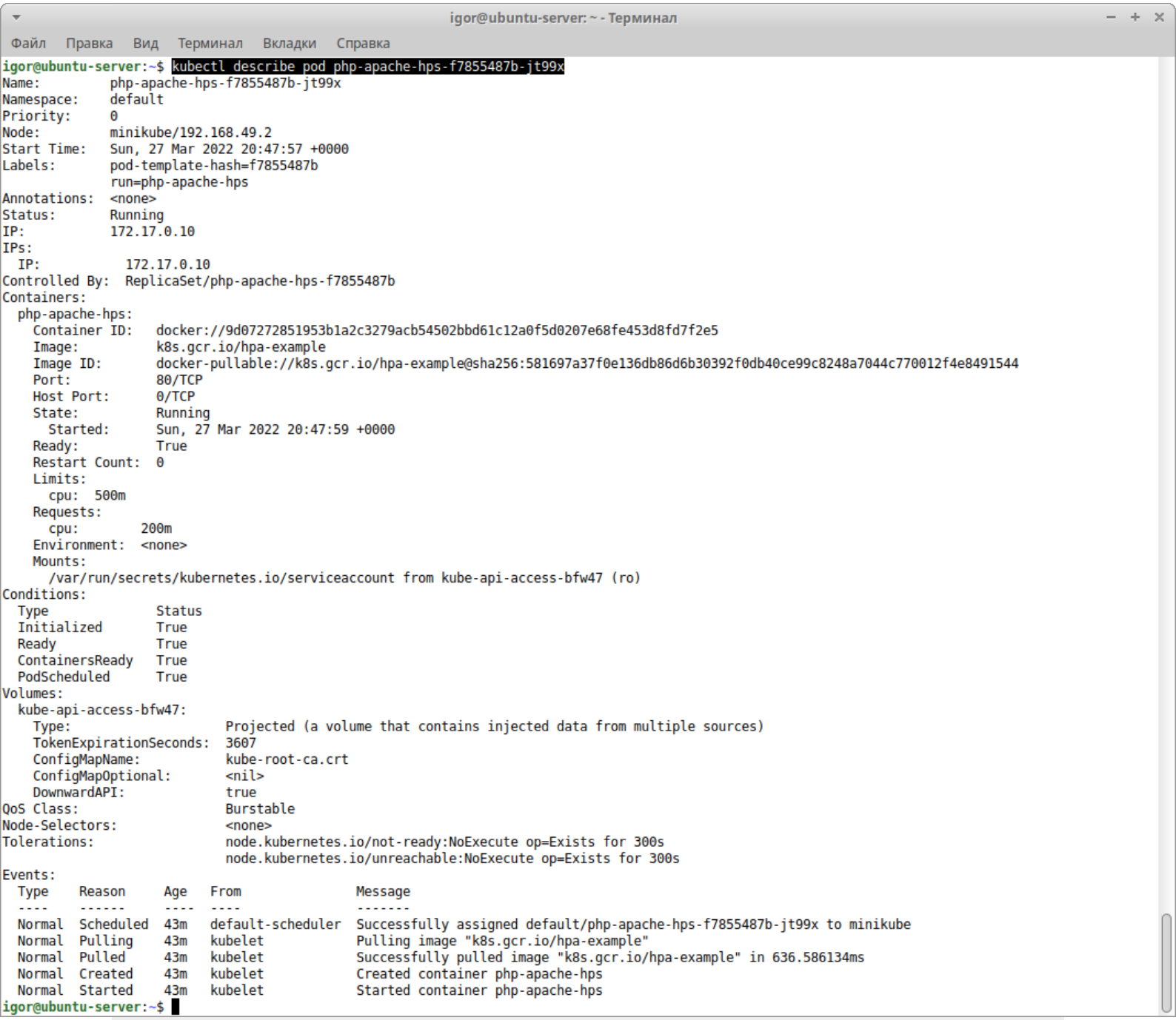
```
kubectl top pod
```



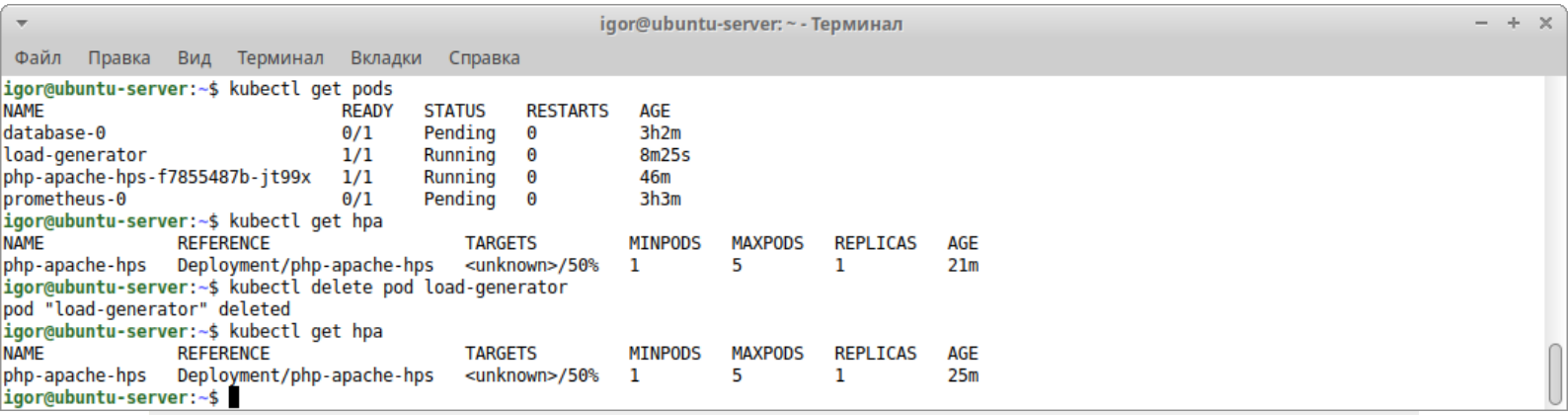

```
kubectl get hpa

kubectl get pods
```




```
kubectl describe pod php-apache-hps-f7855487b-jt99x
```




```
kubectl get pods

kubectl get hpa

kubectl delete pod load-generator

kubectl get hpa
```

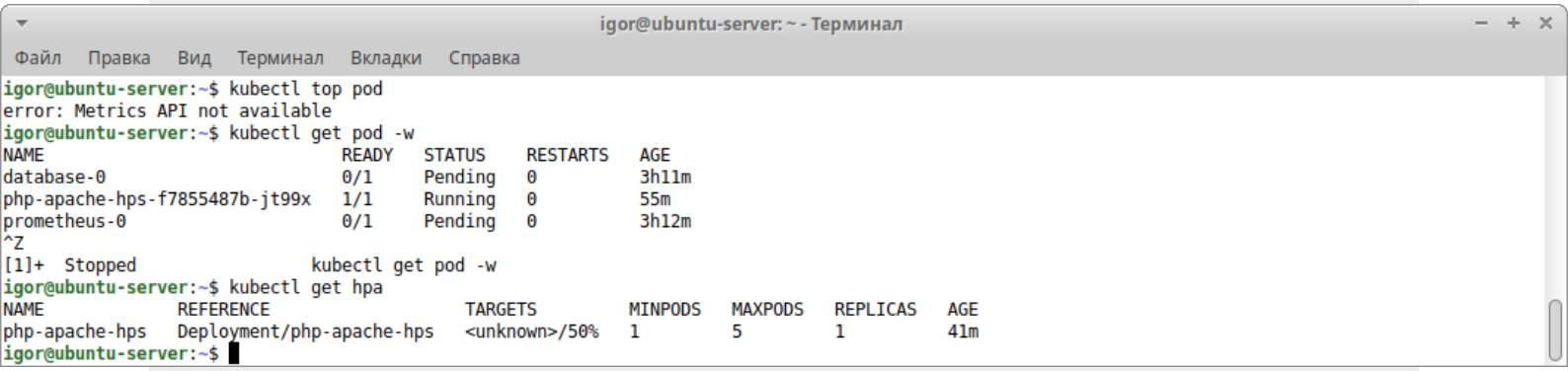



```
kubectl top pod
```

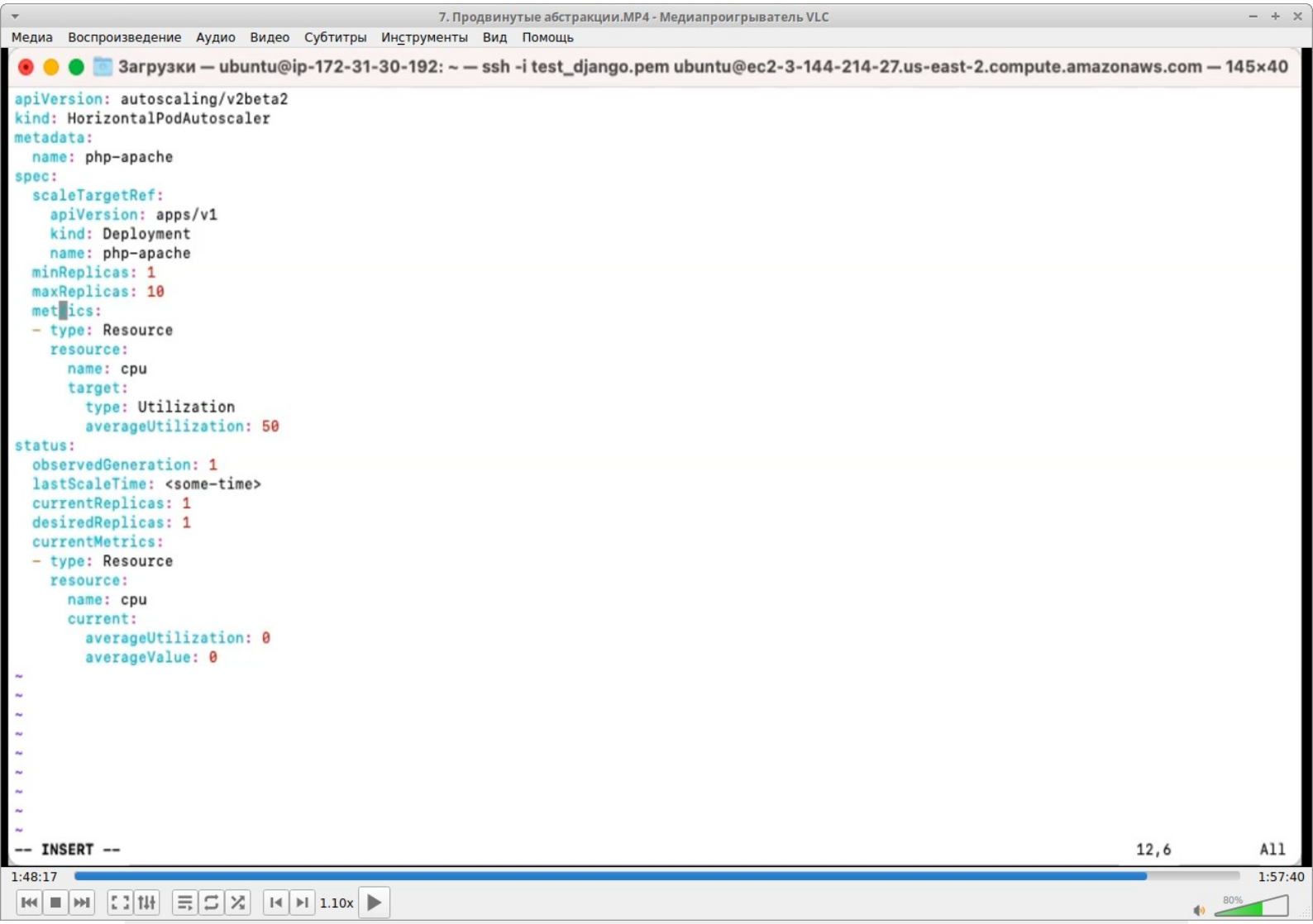
```
error: Metrics API not available
```

```
kubectl get pod -w
```

```
kubectl get hpa
```




```
vi hpa.yaml
```


7. Продвинутые абстракции.MP4 - Медиапроигрыватель VLC
Медиа Воспроизведение Аудио Видео Субтитры Инструменты Вид Помощь
Загрузки — ubuntu@ip-172-31-30-192: ~ — ssh -i test_django.pem ubuntu@ec2-3-144-214-27.us-east-2.compute.amazonaws.com — 145×40
apiVersion: autoscaling/v2beta2
kind: HorizontalPodAutoscaler
metadata:
name: php-apache
spec:
scaleTargetRef:
apiVersion: apps/v1
kind: Deployment
name: php-apache
minReplicas: 1
maxReplicas: 10
metrics:
- type: Resource
resource:
name: cpu
target:
type: Utilization
averageUtilization: 50
status:
observedGeneration: 1
lastScaleTime: <some-time>
currentReplicas: 1
desiredReplicas: 1
currentMetrics:
- type: Resource
resource:
name: cpu
current:
averageUtilization: 0
averageValue: 0
-- INSERT --
12,6
All
1:48:17
1:57:40
1.10x
80%