https://kubernetes.io/ru/docs/reference/kubectl/cheatsheet/

Kubectl run <podname> --image=nginx –port=80

Kubectl port-forward <pod\_name> --address 0.0.0.0 8080:80

Kubectl get nodes

Kubectl get pods

Kubectl description pod <pod\_name>

Kubectl delete pod <pod\_name>

**Minimal configuration:**

apiVersion: v1

kind: Pod

metadata:

name: <pod\_name>

spec:

containers:

- name : <container-name>

image: <image\_name>

kubectl apply -f <manifest\_file.yml>

kubectl delete -f <manifest\_file.yml>

**Example:**

apiVersion: v1

kind: Pod

metadata:

name: myweb

spec:

containers:

- name : apache\_container

image: httpd:latest

ports:

- containerPort: 80

- containerPort: 8080

apiVersion: v1

kind: Pod

metadata:

name: myweb

labels:

env : prod

app : main

tier : frontend

f : fds

owner: TimurAtabaev

spec:

containers:

- name : nginx-server

image: nginx:latest

ports:

- containerPort: 80

- name : tomcat-server

image: tomcat:latest

ports:

- containerPort: 8080

Kubectl port-forward myweb --address 0.0.0.0 7788:80

Kubectl port-forward myweb --address 0.0.0.0 7889:8080

kubectl exec -i -t <pod\_name> --container <container\_name> -- /bin/bash

**DEPLOYMENT**

Kubectl create deployment <deployment\_name> --image nginx

Kubectl get deploy

kubectl describe deployment <deployment\_name>

kubectl describe pod <deployment\_name>

kubectl scale deployment <deployment\_name> --replicas 4

kubectl get rs #get replica-set

kubectl autoscale deployment <deployment\_name> --min=4 –-max=6 --cpu-percent=80 # min4 and max6 replicas with avarage 80% CPU usage

kubect get hpa #get horizontal pod autoscaling

kubectl rollout status deployment/first-deployment

kubectl rollout history deployment/<deployment\_name> #history of deployment

**Deployment manifest example:**

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-web-deployment-with-replicas

labels:

app: my-k8s-application

spec:

replicas: 3

selector:

matchLabels:

project: kgb

template:

metadata:

labels:

project: kgb

spec:

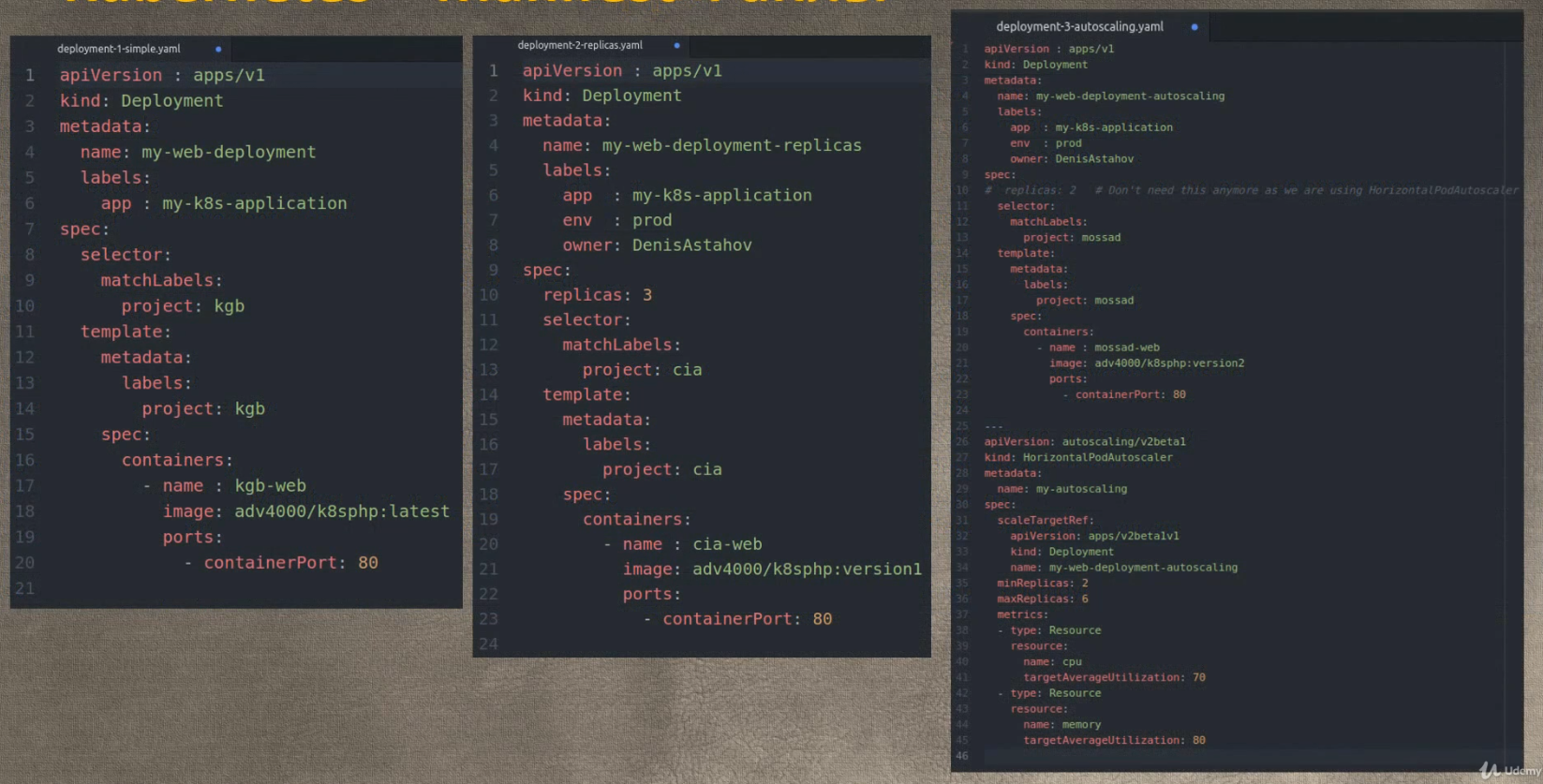
containers:

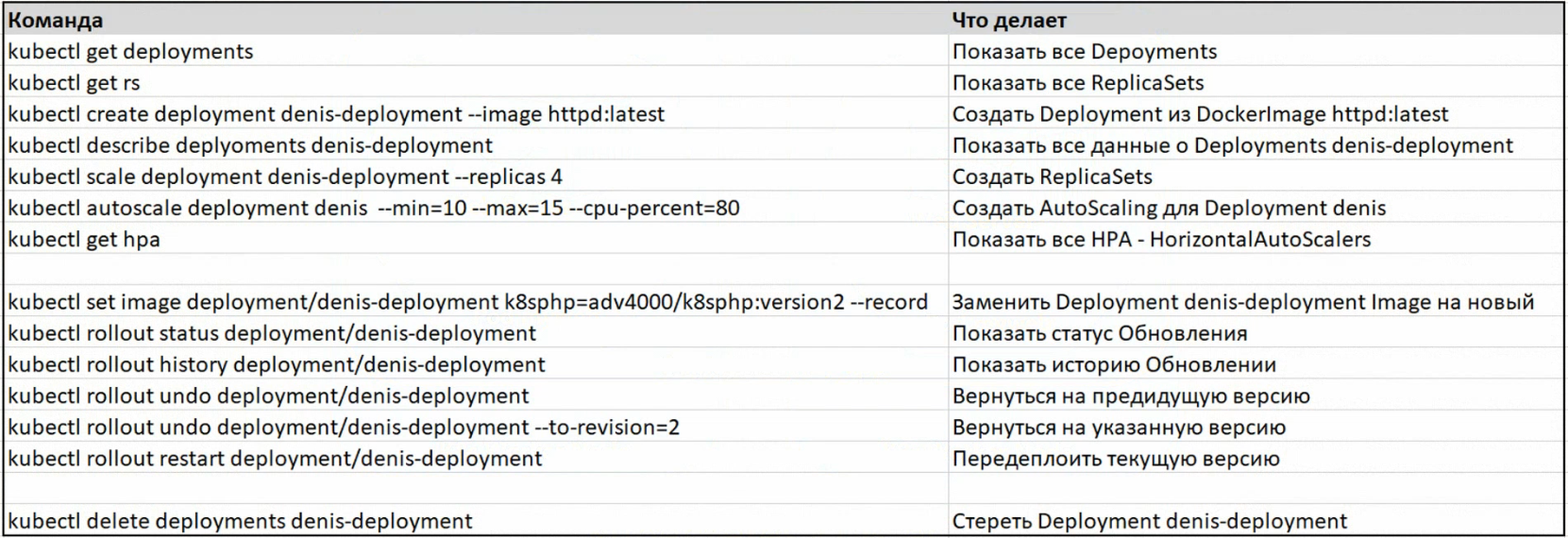
- name : apache-server

image: httpd

ports:

- containerPort: 80





SERVICES

kubectl create deploy t-deploy --image nginx #create example deploy

kubectl expose deployment t-deploy --type=ClusterIP --port 80 #create svc

kubectl expose deployment t-deploy --type=NodePort --port 80

kubectl get services

**Manifest with service:**

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-web-deploy

labels:

app: my-k8s-deploy

spec:

replicas: 3

selector:

matchLabels:

project: my-project

template: #pods

metadata:

labels:

project: my-project #Service will look for those PODS labels!

spec:

containers:

- name: web-server-container

image: nginx

ports:

- containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: my-single-pod-service

labels:

env: prod

owner: TimurAtabaev

spec:

selector:

project: my-project #selecting pods depends on this labels

ports:

- name : app-listener

protocol: TCP

port : 80 #port on LoadBalancer

targetPort: 80 #port on POD

kubectl apply -f service-manifest.yml