NOTE: ALL QUESTIONS HAD DEPLOYMENT CONFIG (dc)
Check for app access using "curl -s http://link.example.com" on every question

ssh controlplane
oc whoami
 - Password for kubeadmin is given in exam
oc login -u kubeadmin -p passwd https://api.ocp4.example.com:6443
oc whoami ; oc get clusterversion
sudo yum install -y httpd-tools openssl

1.Create users armstrong, jobs, wozniak, jack & eric with the password password1, password2, password3, password4 and password5 (password of each user might be different) respectively via htpasswd identity provider. Name of the secret to store these values should be ex280-htpasswd (secret name) identity, provider name should be ex280-cluster.

htpasswd -b -B -c ex280-htpasswd armstrong password
htpasswd -b -B ex280-htpasswd jobs password

cat ex280-htpasswd

oc create secret generic ex280-secret --from-file=htpasswd=ex280-htpasswd -n openshift-config
oc get secret -n openshift-config

oc get oauth -n openshift-config
oc get oauth cluster -o yaml > oauth-backup.yaml **[spec will be empty]**
oc edit oauth cluster
…
spec:
  identityProviders:
  - **h**tpasswd:
      fileData:
        name: ex280-secret ← Secret Name
    mappingMethod: claim
    name: ex280-htpasswd ← Identity Provider Name
    type: HTPasswd

oc get oauth; watch oc get pods -n openshift-authentication ← test login after new pods are created
oc login -u armstrong -p password
oc login -u kubeadmin -p password
oc get users
oc get identity ← NOTE: Identity name in the output should match the Identity name in oauth

If identity name does not matches, delete the identity first and then user:
oc delete identity "myusers:armstrong"
oc delete user armstrong

Again login to the user & confirm the identity:
oc login -u armstrong -p password
oc get users ; oc get identity

2) Create 5 new-projects with names like (example: project1, project2, project3, project4, project5) and assign the roles to the respective user create as: armstrong is admin of project1, jack & eric has view permission on project2, wozniak has edit power on project1

oc new-project project1
…
oc new-project project5

oc policy add-role-to-user admin armstrong -n project1
oc policy add-role-to-user view jack -n project2
oc policy add-role-to-user view eric -n project2
oc policy add-role-to-user edit woz -n project1 ; # oc describe rolebinding -n project1

3) Assign cluster permission to the users as: Jobs (cluster admin user) would be the cluster admin, wozniak would not be allowed cluster admin powers, No user would be able to provision projects, Jack (provisioner user) would be allowed to provision projects. kubeadmin user should be deleted.

oc adm policy add-cluster-role-to-user cluster-admin jobs
oc get clusterrolebinding -o wide | grep jobs
oc login -u jobs -p pass ← on successful login it should list many projects. It shows access to all
oc whoami; oc get nodes ← This should list the nodes if jobs is now the cluster-admin

oc adm policy remove-cluster-role-from-user cluster-admin woz ← This may fail if the role is absent

oc get clusterrolebinding -o wide | grep self-provisioner
oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth ← Ignore err

oc adm policy add-cluster-role-to-user self-provisioner jack
oc describe clussterrolebinding | less

oc get secret -A | grep kubeadmin
oc get secret kubeadmin -n kube-system -o yaml > kubeadmin-backup.yaml (This is not of much use)

oc delete secret kubeadmin -n kube-system

4)Create two groups commander and pilot. Add user armstrong as part of commander group and jack & Eric to be part of pilot group. Commander group should be allowed to edit project3 and pilot group should be allowed view project1

oc adm groups new commander armstrong
oc adm groups new pilot jack eric
oc get groups
oc policy add-role-to-group edit commander -n project3
oc policy add-role-to-group view pilot -n project1
oc describe rolebinding -n project1 ; oc describe rolebinding -n project3

5) Create a resource quota of name ex280-quota (quota name) in the project1 (project name) with resource consumption to be as: Pods=3, service=6, Memory=1Gi, CPU=2 Core,replication controllers=3

oc project project1 ← to change to project1
oc create quota ex280-quota --hard=pods=2,service=6,memory=1G,cpu=2,replicationcontrollers=3
oc get quota ; oc describe quota ex280-quota

6) Create a limit Range on project4 with limits set to be as: For Pod & Container CPU 10m to 100m and 5Mi to 500Mi, default Request: 50m for container

```
# vi limits
apiVersion: v1
kind: LimitRange
metadata:
  name: ex280-limits
  namespace: project4
spec:
  limits:
    - type: Pod
      min:
         cpu: 10m
         memory: 5Mi
      max:
         cpu: 100m
         memory: 500Mi
    - type: Container
      min:
         cpu: 10m
         memory: 5Mi
      max:
         cpu: 100m
         memory: 500Mi
      defaultRequest:
         cpu: 50m
         memory: 100Mi
```

oc create -f limits.yaml -n project4
oc get limits -n project4
oc describe limits ex280-limits -n project4 ← "ex280-limits" here is the limit name used in yaml file

---

oc logs my-pod-name -c my-container-name ← To check the logs of a specific container in a pod
oc logs my-pod-name ← To check the logs related to the pod

---

7)Deploy the application and make sure it is available in running state and must show some output on the given url.(taint/toleration)
➔ Delete route and create route

oc new-project rocky
oc new-app --name httpd httpd:2.4
oc get deploy ; oc get pods ; oc describe pod pod-name ← Look for the last logs which will give idea

oc describe nodes | grep -i taint
oc get nodes master01 -o yaml | grep -i taint -A5 -B2 ← Copy the taint section

oc edit deploy httpd ← Search for “dnsPolicy” under containers section
…
tolerations:
- effect: NoSchedule
  key: Node
  value: Worker
  operator: Equal

oc get deploy ; oc get pods
oc status
oc get route ← Delete the existing route as it will be inaccessible. Try to browse in firefox
oc delete route httpd
oc expose svc httpd --hostname httpd-rockzzz.apps.ocp4.example.com
oc expose dc httpd --port=8080 ← This will create the service
oc expose svc httpd --hostname httpd-rockzzz.apps.ocp4.example.com ← This will create the route
oc get route ← Browse the new URL in firefox (you may need to open the console)

oc get svc
oc describe svc httpd
…
Selector: name=api ← [1]
Endpoints: <none> ← Always ensure that “Endpoints” have at least one Pod internal IP

oc describe dc my-httpd
…
Pod Template:
  Labels: name=frontend ← [2]

[1] and [2] does not match. Update the “svc” to match the “dc”

oc edit svc httpd
…
spec:
  selector:
    name=frontend

oc describe svc httpd
…
Selector: name=frontend
Endpoints: <IP of the POD>

To get the IP of the pod:
oc get pods -o wide ← Check the “IP” column and refer to IP of “Completed” status

oc get networkpolicies
oc describe networkpolicy new-policy ← Check for PodSelector

8. Manually Scale up the application running on project5 upto 5 replicas(taint/toleration)

oc project project5
oc get deploy ; oc get pods ← the pods will be in “Pending” state so we need to **add toleration**

oc edit deploy website ← Search for “dnsPolicy” under containers section
…
```
      tolerations:
      - effect: NoSchedule
        key: Node
        value: Worker
        operator: Equal
```

oc get pods ; oc describe deployment website ← Check for current replicas
oc scale deploy website --replicas=5
oc scale dc website --replicas=5
oc get pods -w ; oc describe deploy website | grep -i replica
oc get dc; oc get rc; oc get all

Under “oc get pods”, you should see one pod with name ending with “deploy” & status should be “Completed” while other pods should be in “running” status

9) Apply autoscale on the application running in project7 with min pod 6, Max 9 and cpu-percent=50 and with a resource request for cpu of 10m and limit for cpu is 100m
or
Auto scale of deployment and they didn’t ask for quota they just ask for the limit like
Set cpu resource request: 10
Set cpu resource limit: 100

oc project project7
oc get deployment
oc edit deploy website ← Search for “dnsPolicy” under containers section & add “tolerations”
oc autoscale deploy app --min=6 --max=9 --cpu-percent=50 ← Check “oc autoscale deploy -h”
oc autoscale dc/app --min=6 --max=9 --cpu-percent=50
oc get hpa ; oc get pods

oc set resources deploy app --limits=cpu=100m --requests=cpu=10m ← Check “oc set resources -h”
oc set resources dc/app --limits=cpu=100m --requests=cpu=10m ; oc get all ← New rc will created

10) Create a secret in the project p7 with name magic and key-values of linux-world=Xshelf
or
In this question they ask to create secret and they give two things like:
Key name: linux-world
Key value: Xshelf

oc new-project p7
oc project ; oc get secret -n p7
oc create secret generic magic --from-literal=linux-world=Xshelf
oc get secret ; oc describe secret magic

$ oc extract secret/htpasswd-secret -n openshift-config --to /tmp/ --confirm ← To extract a secret
Now update the file in /tmp
$ oc set data secret/htpasswd-secret -n openshift-config --from-file /tmp/htpasswd ← Set the new value

11) Create a service account with name ex280-sa so that application runs with any user id
or
Create service account in the project p7

oc project p7
oc create sa ex280-sa
oc get sa

oc adm policy add-scc-to-user anyuid -z ex280-sa
oc describe clusterrolebinding | less ← search for “ex280-sa”

12.)Create a secure route for the application running in project area51 with name app-scale. The applications should be available on the given url and application should use self signed certificate with the given information as:
“/C=SI/ST=Ljubljana/L=Ljubljana/O=Security/OU=IT Department/CN=scale-area51.apps.ocp4.example.com”.
The application must produces the output

yum install -y openssl
man req ← copy & execute the two commands from example section as is to generate key & CSR
~~~
# openssl genrsa -out key.pem 2048
# openssl req -new -key key.pem -out req.pem ← This command will need input. Values in Question
~~~

man x509 ← use a command from example section which has “-x509toreq” & update the csr name with “-days”
~~~
# openssl x509 -req -in req.pem -out req.crt -signkey key.pem -days 365
~~~

oc project area51

oc get deploy ; oc get pods
oc edit deploy app-scale ← Search for “dnsPolicy” under containers section & add “tolerations”

oc create secret tls tls-cert --key key.pem --cert req.crt
oc get route ← If there is an existing route then delete it after copying the hostname
oc create route edge scale --service app-scale --hostname scale-area51.apps.ocp4.example.com --key key.pem --cert req.crt
oc get route
curl -I -v https://scale-area51.apps.ocp4.example.com
oc describe pod pod-name | grep Mounts -A4
oc get networkpolicies
oc describe networkpolicy new-policy ← Check for PodSelector

13) Deploy the application oranges with the secret magic created in question 10. or Deploy the application
→ Ans – assign secret value
Deploy the application with the service account ex280sa created in question 11.
→ Ans – assign service account

oc project p7
oc get deploy ; oc get pods
oc edit deploy oranges → Search for “dnsPolicy” under containers section & add “tolerations”
[NOTE: ENV VARIABLE TO BE ASSIGNED USING THIS SECRET]
oc set env deploy oranges --from=secret/magic --prefix MYSQL_
oc set env dc/oranges --from=secret/magic --prefix MYSQL_

oc set sa deploy oranges ex280-sa
oc set sa dc/oranges ex280-sa

oc edit deploy oranges → Check the value of “selector” and note under “matchLabel”
oc edit dc/oranges
oc get svc
oc edit svc oranges → Check the value of “selector”.

The value of “selector” in svc should match “deploy”

oc get networkpolicies
oc describe networkpolicy new-policy ← Check for PodSelector
curl -s http://quotes.apps.ocp4.example.com/ ← Link can get from “oc get route”

14) Deploy the application abc in project8 and make sure the application is up and running and available on the given url.
or
Deploy the application → Configure node label

oc project project8
oc get deploy; oc get pods
oc edit deploy abc → Search for “dnsPolicy” under containers section & add “tolerations”

oc get pods
oc describe pod pod-name → In logs it will say none node match node affinity/selector

oc describe node master01 | less → Under “Labels” section look for the label
oc edit depoy abc → Search for “nodeSelector” and correct as per the node label
oc edit dc/abc
Search for “dnsPolicy” where we add the “tolerations”. Below to it add:
nodeSelector:
    star: Trek
oc get networkpolicies
oc describe networkpolicy new-policy ← Check for PodSelector

15) Memory Insufficient error in project test for application httpd

oc project test
oc get deploy httpd ; oc get pods
oc describe pod pod-name → This will say insufficient memory
oc describe node master01

oc edit deploy httpd → Go to “containers” sections where you will find “limits/requests” field. Under “requests” field, decrease the memory value. Also you will need to add “tolerations”
oc edit dc/httpd
oc adm top nodes

oc get networkpolicies or oc get networkpolicy
oc describe networkpolicy new-policy ← Check for PodSelector & NamespaceSelector
oc edit networkpolicy new-policy
```
spec:
   podSelector:
      matchLabels:
         deployment: httpd
   ingress:
     - from:
       - namespaceSelector:
            matchLabels:
               name: network-test
         podSelector:
            matchLabels:
               deployment: sample-app
       ports:
       - port: 8080
         protocol: TCP
```

To label a namespace:
oc label namespace test-project name=test-pro
oc describe namespace <namespace name>
example: oc describe namespace test-project

Refer page# 158