

Maryam Taherzadeh
Week 6
19529-Cloud Computing

Step 1: Create the Node.js app, app.js and Dockerfile

```
taherzadeh19529@cloudshell:~ (cs571-project)$ cat app.js
const http = require('http');
const os = require('os');

console.log("Kubia server starting...");

var requestCount = 0;

var handler = function(request, response) {
    console.log("Received request from "
        +request.connection.remoteAddress);
    requestCount++;
    if (requestCount > 5) {
        response.writeHead(500);
        response.end("I'm not well. Please restart me!");
        return;
    }
    response.writeHead(200);
    response.end("You've hit " + os.hostname() + "\n");
};

var www = http.createServer(handler);
www.listen(8080);

taherzadeh19529@cloudshell:~ (cs571-project)$ cat Dockerfile
FROM node:7
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]
```

2- Build kubia-unhealthy image

```
taherzadeh19529@cloudshell:~/cs571-demo-project-304923$ cd app_js_unhealthy
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-demo-project-304923)$ sudo docker build -t kubia-unhealthy .
Sending build context to Docker daemon 17.92kB
Step 1/3 : FROM node:7
--> d9aed20b68a4
Step 2/3 : ADD app.js /app.js
--> Using cache
--> 1a4a5315f550
Step 3/3 : ENTRYPOINT ["node", "app.js"]
--> Using cache
--> c2338cf2041d
Successfully built c2338cf2041d
Successfully tagged kubia-unhealthy:latest
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-demo-project-304923)$ sudo docker run --name kubia-container -p 8080:8080 -d kubia-unhealthy
```

```
see 'docker run --help'.
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-demo-project-304923)$ curl localhost:8080
You've hit da79d7acb0b6
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-demo-project-304923)$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
kubia-unhealthy    latest   c2338cf2041d  36 minutes ago  660MB
tahezadeh19529/repo_name  latest   c2338cf2041d  36 minutes ago  660MB
gcr.io/k8s-minikube/kicbase v0.0.17  a9b1f16d8ece  6 weeks ago   985MB
node               7        d9aed20b68a4  3 years ago   660MB
```

3-Run image on container and Check it is correctly running

4- Check images is in reporistory

5- Check container ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
da79d7acb0b6	kubia-unhealthy	"node app.js"	36 minutes ago	Up 36 minutes	0.0.0.0:8080->8080/tcp
84116ccca9b64	gcr.io/k8s-minikube/kicbase:v0.0.17	"/usr/local/bin/entr..."	About an hour ago	Up About an hour	127.0.0.1:49156->2376/tcp, 127.0.0.1:49157->5000/tcp, 127.0.0.1:49153->8443/tcp

6- Follow the same process and push the image to docker hub using

push the docker image to docker hub

login to your docker hub with

sudo docker login -u="username" -p="password"

push the docker image to your repository:

sudo docker push username/repo_name

```
taherzadeh19529@cloudshell:~ (cs571-project)$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED   STATUS    PORTS
NAMES
b2f947def85   b3c4765518d7   "node app.js"   3 hours ago   Up 3 hours   0.0.0.0:8080->8080/tcp
kubia-container
taherzadeh19529@cloudshell:~ (cs571-project)$ sudo docker tag kubia-unhealthy tahezadeh19529/repo
taherzadeh19529@cloudshell:~ (cs571-project)$ sudo docker tag kubia-unhealthy tahezadeh19529/repo_name
taherzadeh19529@cloudshell:~ (cs571-project)$ sudo docker push tahezadeh19529/repo_name
```

```
taherzadeh19529@cloudshell:~ (cs571-project)$ sudo docker push tahezadeh19529/repo_name
Using default tag: latest
The push refers to repository [docker.io/tahezadeh19529/repo_name]
c3a78c36ddc1: Layer already exists
ab90d83fa34a: Layer already exists
8ee318e54723: Layer already exists
e6695624484e: Layer already exists
da59b99bbd3b: Layer already exists
5616a6292c16: Layer already exists
f3ed6cb59ab0: Layer already exists
654f45ecb7e3: Layer already exists
2c40c66f7667: Layer already exists
latest: digest: sha256:ae08ac6534d858aa7eff527fec7e9c2f628dec10b1eb8be9229c84afbc72dell size: 2213
```

Step 2: Create a new pod that includes an HTTP GET liveness probe

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ cat kubia-liveness-probe.yaml
apiVersion: v1
kind: Pod
metadata:
  name: kubia-liveness-probe
spec:
  containers:
  - image: tahezadeh19529/repo_name
    name: kubia
    livenessProbe:
      httpGet:
        path: /
        port: 8080
```

- Create a pod

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl create -f kubia-liveness-probe.yaml
pod/kubia-liveness-probe created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ █
```

- Check the status of the pod

```
pod/kubia-liveness-probe created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME                      READY   STATUS    RESTARTS   AGE
batch-job-every-fifteen-minutes-1615061700-4nx8t  0/1     Completed   0          37m
batch-job-every-fifteen-minutes-1615062600-shfz5   0/1     Completed   0          22m
batch-job-every-fifteen-minutes-1615063500-kccg8   0/1     Completed   0          7m41s
batch-job-svm4p           0/1     Completed   0          91m
kubia-hmdgs               0/1     ImagePullBackOff 0          146m
kubia-hmrgp                0/1     ImagePullBackOff 0          146m
kubia-liveness             0/1     CrashLoopBackOff 170        12h
kubia-liveness-probe      0/1     CrashLoopBackOff 2          46s
kubia-w8tpj                0/1     ImagePullBackOff 0          131m
multi-completion-batch-job-6p2t7  0/1     Completed   0          76m
multi-completion-batch-job-7v9x7  0/1     Completed   0          80m
```

```
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl describe po kubia-liveness-probe
Name:           kubia-liveness-probe
Namespace:      default
Priority:      0
Node:          gke-kubia-default-pool-bb069bc2-2zp0/10.138.0.20
Start Time:    Sat, 06 Mar 2021 20:51:56 +0000
Labels:         <none>
Annotations:   <none>
Status:        Running
IP:            10.0.1.18
IPs:
  IP: 10.0.1.18
Containers:
  kubia:
    Container ID: docker://37111eddfe440ad2517957513373cb3efddcbef7b764de8ae76ba795e81afd2
    Image:          taherzadeh19529/repo_name
    Image ID:      docker-pullable://taherzadeh19529/repo_name@sha256:4e08ac6534d858aa7eff527fec7e9c2f628dec10b1eb8be9229c
    84afbc72de11
      Port:        <none>
      Host Port:   <none>
      State:       Waiting
        Reason:    CrashLoopBackOff
      Last State: Terminated
        Reason:    Error
        Exit Code:  1
      Started:    Sat, 06 Mar 2021 20:57:59 +0000
      Finished:   Sat, 06 Mar 2021 20:57:59 +0000
    Ready:        False
    Restart Count: 6
    Liveness:     http-get http://:8080/ delay=0s timeout=1s period=10s #success=1 #failure=3
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-8vckd (ro)
Conditions:
  Type      Status
  Initialized  True
  Ready      False
  ContainersReady  False
  PodScheduled  True
```

- Getting information about a Replication Controller

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get rs
NAME      DESIRED   CURRENT   READY   AGE
kubia     3          3          0       10s
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl describe rs
```

- Checking Node status

```
kubia-default-pool-dbees542c-662c.
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get node
NAME                  STATUS   ROLES   AGE   VERSION
gke-kubia-default-pool-6af3ddf8-5ftv  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-bb069bc2-2zp0  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-dbees542c-662c Ready    <none>  10h   v1.18.12-gke.1210
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get rs
```

Simulating a node failure by shutting down its network interface

```
gcloud compute ssh gke-kubia-default-pool-dbess542c-662c --zone uswest1-b
```

```

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
    inet 10.138.0.19 netmask 255.255.255.255 broadcast 0.0.0.0
        inet6 fe80::4001:aff:fe8a:13 prefixlen 64 scopeid 0x20<link>
            ether 42:01:0a:8a:00:13 txqueuelen 1000 (Ethernet)
            RX packets 1405003 bytes 967060937 (922.2 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 899056 bytes 313349519 (298.8 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 173299 bytes 89184291 (85.0 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 173299 bytes 89184291 (85.0 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
veth1b6a98bb: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
    inet6 fe80::dc00:6fff:fe3c:66be prefixlen 64 scopeid 0x20<link>
        ether de:00:6f:3c:66:be txqueuelen 0 (Ethernet)
            RX packets 1 bytes 42 (42.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 5 bytes 446 (446.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
veth807029a8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
    inet6 fe80::74f9:a8ff:fe66:64fb prefixlen 64 scopeid 0x20<link>
        ether 76:f9:a8:e6:64:fb txqueuelen 0 (Ethernet)
            RX packets 388164 bytes 134972348 (128.7 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 472886 bytes 366632993 (349.6 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
vethb7e15df4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
    inet6 fe80::c894:2eff:fe66:605c prefixlen 64 scopeid 0x20<link>
        ether ca:94:2e:66:60:5c txqueuelen 0 (Ethernet)
            RX packets 179072 bytes 24499213 (23.3 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 186959 bytes 70190920 (66.9 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
taherzadeh19529@gke-kubia-default-pool-dbee542c-662c ~ $ sudo ifconfig eth0 down

```

Then open another terminal session and check node status: which is showing NotReady

```

taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get node
NAME           STATUS   ROLES   AGE    VERSION
gke-kubia-default-pool-6af3ddf8-5ftv  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-bb069bc2-2zp0  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-dbee542c-662c  NotReady <none>  10h   v1.18.12-gke.1210
taherzadeh19529@cloudshell:~ (cs571-project)$ 

```

- Let's bring the node back and you can see the node is in ready state

```
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get node
NAME           STATUS   ROLES   AGE    VERSION
gke-kubia-default-pool-6af3ddf8-5ftv  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-bb069bc2-2zp0  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-dbee542c-662c  NotReady <none>  10h   v1.18.12-gke.1210
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
kubia-hmdgs 0/1     ImagePullBackOff  0          21m
kubia-hmrgp  0/1     ImagePullBackOff  0          21m
kubia-hpsrx  0/1     Terminating      0          21m
kubia-liveness 0/1     CrashLoopBackOff  141       10h
kubia-liveness-probe 0/1     CrashLoopBackOff  19        72m
taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute instance reset gke-kubia-default-pool-dbe
taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute instance reset gke-kubia-default-pool-dbe
e542c-662c --zone us-west1
ERROR: (gcloud.compute) Invalid choice: 'instance'.
Maybe you meant:
  gcloud compute

To search the help text of gcloud commands, run:
  gcloud help -- SEARCH TERMS
taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute instances reset gke-kubia-default-pool-db
ee542c-662c --zone us-west1
ERROR: (gcloud.compute.instances.reset) Could not fetch resource:
- Invalid value for field 'zone': 'us-west1'. Unknown zone.

taherzadeh19529@cloudshell:~ (cs571-project)$ gcloud compute instances reset gke-kubia-default-pool-db
ee542c-662c --zone us-west1-b
Updated [https://www.googleapis.com/compute/v1/projects/cs571-project/zones/us-west1-b/instances/gke-k
ubia-default-pool-dbee542c-662c].
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get node
NAME           STATUS   ROLES   AGE    VERSION
gke-kubia-default-pool-6af3ddf8-5ftv  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-bb069bc2-2zp0  Ready    <none>  10h   v1.18.12-gke.1210
gke-kubia-default-pool-dbee542c-662c  Ready    <none>  10h   v1.18.12-gke.1210
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get rc
NAME  DESIRED  CURRENT  READY   AGE
kubia  3         3        0       25m
```

Horizontally Scaling Pods

- Scaling up a ReplicationController

```
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl scale rc kubia --replicas=10
replicationcontroller/kubia scaled
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl delete rc kubia --cascade=false
```

```
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl delete rc kubia --cascade=false
warning: --cascade=false is deprecated (boolean value) and can be replaced with --cascade=orphan.
replicationcontroller "kubia" deleted
```

```
replicationcontroller/kubia deleted
taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl get rc
No resources found in default namespace.
```

Create a ReplicaSet

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ cat kubia-replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: kubia
spec:
  replicas: 3
  selector:
    matchLabels:
      app: kubia
  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
        - name: kubia
          image: tahezadeh19529/repo_name:kubia
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl create -f kubia-replicaset.yaml
replicaset.apps/kubia created
```

Check to confirm rs is created

```
replicaset.apps/kubia created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get rs
NAME      DESIRED   CURRENT   READY   AGE
kubia     3          3          0       10s
```

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl describe rs
Name:           kubia
Namespace:      default
Selector:       app=kubia
Labels:         <none>
Annotations:    <none>
Replicas:       3 current / 3 desired
Pods Status:   0 Running / 3 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=kubia
  Containers:
    kubia:
      Image:      tahezadeh19529/repo_name:kubia
      Port:       <none>
      Host Port: <none>
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
  Events:
    Type      Reason     Age   From            Message
    ----      ----     --   --              --
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-7gzbr
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-4sqdd
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-zj7fs
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-fz6dd
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-zd9dn
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-7cm8k
    Normal   SuccessfulDelete 90s   replicaset-controller  Deleted pod: kubia-dnsqr
```

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: kubia
spec:
  replicas: 3
  selector:
    matchLabels:
      app: kubia
  template:
    metadata:
      labels:
        app: kubia
    spec:
      containers:
        - name: kubia
          image: tahezadeh19529/repo_name
```

You'll create a DaemonSet that runs a mock ssd-monitor every five seconds.

- Create daemonset
- Check status of node, pod and ds
- Adding the required label to your nodes. (node name is different for every user)
- Check the status of pod you will see ssd-monitor pod running
- Running jobs pods sequentially

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ cat ssd-monitor-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: ssd-monitor
spec:
  selector:
    matchLabels:
      app: ssd-monitor
  template:
    metadata:
      labels:
        app: ssd-monitor
    spec:
      nodeSelector:
        disk: ssd
      containers:
        - name: main
          image: luksa/ssd-monitor
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl create -f ssd-monitor-daemonset.yaml
daemonset.apps/ssd-monitor created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get rs
NAME   DESIRED   CURRENT   READY   AGE
kubia   3          3          0       4m37s
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get ds
NAME   DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
ssd-monitor   0          0          0       0           0           disk=ssd   36s
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME   READY   STATUS   RESTARTS   AGE
kubia-hmdgs   0/1     ImagePullBackOff   0       35m
kubia-hmrgp   0/1     ImagePullBackOff   0       35m
kubia-liveness   1/1     Running   145      10h
kubia-liveness-probe   1/1     Running   23       86m
kubia-w87pj   0/1     ImagePullBackOff   0       20m
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get node
NAME   STATUS   ROLES   AGE   VERSION
gke-kubia-default-pool-6af3ddf8-5ftv   Ready   <none>   11h   v1.18.12-gke.1210
gke-kubia-default-pool-bb069bc2-2zp0   Ready   <none>   11h   v1.18.12-gke.1210
gke-kubia-default-pool-dbee542c-662c   Ready   <none>   11h   v1.18.12-gke.1210
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl label node gke-kubia-default-pool-dbee542c-662c disk=ssd
node/gke-kubia-default-pool-dbee542c-662c labeled
```

```
taherzadeh19529@cloudshell:~ (cs571-project)$ cat exporter.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: batch-job
spec:
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure
      containers:
        - name: main
          image: tahezadeh19529/batch-job

taherzadeh19529@cloudshell:~ (cs571-project)$ kubectl create -f exporter.yaml
job.batch/batch-job created
```

- Examine jobs pod logs
- Running multiple pod instances in a job

```
job.batch/batch-job created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get jobs
batch-job  1/1           2m2s   2m41s
apiVersion: batch/v1
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME         READY   STATUS    RESTARTS   AGE
batch-job-svm4p  0/1     Completed   0          3m12s
kubia-hmdgs   0/1     ImagePullBackOff 0          58m
kubia-hmrqp   0/1     ImagePullBackOff 0          58m
kubia-liveness 0/1     CrashLoopBackOff 149       10h
kubia-liveness-probe 1/1     Running   28          109m
kubia-w87pj    0/1     ImagePullBackOff 0          43m
ssd-monitor-7wq9t 1/1     Running   0          22m
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl logs batch-job-svm4p
Sat Mar  6 19:21:27 UTC 2021 Batch job starting
Sat Mar  6 19:23:27 UTC 2021 Finished successfully
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ exit
```

- Running multiple pod instances in a job

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl create -f multi-completion-batch-job.yaml
apiVersion: batch/v1
job.batch/multi-completion-batch-job created
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ cat multi-completion-batch-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: multi-completion-batch-job
spec:
  completions: 5
  template:
    metadata:
      labels:
        app: batch-job
    spec:
      restartPolicy: OnFailure
      containers:
      - name: main
        image: luksa/batch-job
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME          READY   STATUS    RESTARTS   AGE
batch-job-svm4p   0/1     Completed   0          8m47s
kubia-hmdgs     0/1     ImagePullBackOff   0          64m
kubia-hmrgp      0/1     ImagePullBackOff   0          64m
kubia-liveness   0/1     CrashLoopBackOff  151       10n
kubia-liveness-probe 0/1     CrashLoopBackOff  29        115m
kubia-w87pj      0/1     ImagePullBackOff   0          49m
multi-completion-batch-job-ptck6  1/1     Running    0          24s
ssd-monitor-7wq9t  1/1     Running    0          27m
```

- Running job pods in parallel

- You can see 4 jobs completed

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
batch-job-svm4p         0/1     Completed  0          14m
kubia-hmdgs             0/1     ImagePullBackOff 0          69m
kubia-hmrgp              0/1     ImagePullBackOff 0          69m
kubia-liveness           1/1     Running   153        10h
kubia-liveness-probe    1/1     Running   31         121m
kubia-w87pj              0/1     ImagePullBackOff 0          54m
multi-completion-batch-job-7v9x7 0/1     Completed  0          3m59s
multi-completion-batch-job-klm8l 1/1     Running   0          117s
multi-completion-batch-job-ptck6 0/1     Completed  0          6m1s
ssd-monitor-7wq9t        1/1     Running   0          33m
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl get po
NAME                    READY   STATUS    RESTARTS   AGE
batch-job-svm4p         0/1     Completed  0          15m
kubia-hmdgs             0/1     ImagePullBackOff 0          71m
kubia-hmrgp              0/1     ImagePullBackOff 0          71m
kubia-liveness           0/1     CrashLoopBackOff 153        10h
kubia-liveness-probe    0/1     CrashLoopBackOff 31         122m
kubia-w87pj              0/1     ImagePullBackOff 0          56m
multi-completion-batch-job-6p2t7 1/1     Running   0          85s
multi-completion-batch-job-7v9x7 0/1     Completed  0          5m28s
multi-completion-batch-job-klm8l 0/1     Completed  0          3m26s
multi-completion-batch-job-ptck6 0/1     Completed  0          7m30s
ssd-monitor-7wq9t        1/1     Running   0          34m
```

- Creating CronJob

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ cat cronjob.yaml
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: batch-job-every-fifteen-minutes
spec:
  schedule: "0,15,30,45 * * * *"
  jobTemplate:
    spec:
      template:
        metadata:
          labels:
            app: periodic-batch-job
        spec:
          restartPolicy: OnFailure
          containers:
            - name: main
              image: luksa/batch-job
```

```
taherzadeh19529@cloudshell:~/app_js_unhealthy (cs571-project)$ kubectl create -f cronjob.yaml
cronjob.batch/batch-job-every-fifteen-minutes created
```

