Lab Create

This lab is intended to get the team familar with creating a pod in the team namespace. There is no diagnosis or problem to be researched. The lab is complete once the pod is created in the team namespace.

Resources

- K8 yaml - house.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/house:latest
ports	none
Docker	CMD ["/bin/bash", "-c", "./house.sh"]

Tasks

Task description
Download the resource K8 yaml file.
Edit and save the file after replacing all references of **<team>** with your team name.
Create the K8 objects using kubectl create
Did the pod deploy successfully? If not, correct the issue and re-create the K8 objects.

Hint Create

To create the pod use the command: **kubectl create -f <file>;** (replace <file> with the name of the yaml file you have saved an editied.)

Step-by-Step Create

Diagnosis

No diagnosis is necessary for this lab. A new pod should be created after editing the yaml file and using the kubectl create command.

Problem discovered

N/A

Resolution

Edit the yaml file and modify all references of <team> to your team name.

```
Example yaml file that needs to be edited.

--- # Fast Start :: Problem Diagnosis and Troubleshooting Lab
---
apiVersion: apps/v1
```

```
kind: Deployment
metadata:
  name: <team>-house
  namespace: <team>
  labels:
    app: <team>-house
spec:
  selector:
    matchLabels:
      app: <team>-house
  replicas: 1
  template:
    metadata:
      labels:
        app: <team>-house
    spec:
      containers:
      - name: <team>-house
        image: ibmicpcoc/house:latest
        imagePullPolicy: Always
        command: ["/bin/bash", "-c", "/app/avail.sh"]
        env:
          - name: APP_NAMESPACE
            valueFrom:
              fieldRef:
                fieldPath: metadata.namespace
          - name: APP_NAME
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: COLLECTOR_CONFIG
            valueFrom:
              configMapKeyRef:
                name: <team>-collector-config
                key: COLLECTOR_CONFIG
          - name: INSTRUCTOR_CONFIG
            valueFrom:
              configMapKeyRef:
                name: <team>-collector-config
                key: INSTRUCTOR_CONFIG
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
```

Saved the modified file and create the pod "house".

```
Command to create the K8 objects:
    kubectl create -f house.yaml

Result output:
    deployment.apps/house created
```

```
----
Verify the pod deployed successfully.

Command to get pods in namespace:
    kubectl -n <team> get pods         # change <team> to your team namespace
```

Lab Syntax

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Use the debug flow to guide the steps you should attempt in diagnosis of the issue.

Resources

- K8 yaml - baker.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/baker:latest
ports	none
Docker	CMD ["/bin/bash", "-c", "./baker.sh"]

Tasks

Task description
Download the resource K8 yaml file.
Edit and save the file after replacing all references of <team> with your team name / namespace.
Research why the pod did not deploy.
Resolve the issue and create the K8 objects.
Did the pod deploy successfully? If not, correct the issue and re-create the K8 objects.

Hint Syntax

Deployment.spec.template.spec.containers expects an array of entires.

Arrays are defined with a hyphen.

Review and compare the **house.yaml** file for an example of properly defined K8 objects.

Step-by-Step Syntax

Diagnosis

When attempting to create the pod the yaml is not properly defined. This error message is being shown:

error: error validating "baker.yaml": error validating data: ValidationError(Deployment.spec.template.spec.containers): invalid type for io.k8s.api.core.v1.PodSpec.containers: got "map", expected "array"; if you choose to ignore these errors, turn validation off with --validate=false

Problem discovered

The Deployment.spec.template.spec.containers portion of the yaml file is not properly formatted. Got "map", expected "array". Container does not have an array of entires.

Resolution

Edit the yaml file and correct the definition to include a hyphen before the "name:" parameter of the containers portion.

```
Example saved file with hyphen  (portion of file shown below)

apiVersion: apps/v1
kind: Deployment
metadata:
  name: <team>-baker
  namespace: pink
  labels:
    app: <team>-baker
spec:
  selector:
    matchLabels:
      app: <team>-baker
  replicas: 1
  template:
    metadata:
      labels:
        app: <team>-baker
    spec:
      containers:
      - name: <team>-baker                <=== Add the hyphen to this line
        image: ibmicpcoc/baker:latest
        imagePullPolicy: Always
```

Saved the modified file and create the pod "baker".

```
Command to create the K8 objects:
    kubectl create -f baker.yaml

Result output:
    deployment.apps/baker created


----
Verify the pod deployed successfully.
```

```
Command to get pods in namespace:
    kubectl -n <team> get pods         # change <team> to your team namespace
```

Lab Resources

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Use the debug flow to guide the steps you should attempt in diagnosis of the issue.

Useful information

K8 yaml - carbon.yaml
Dockerfile - Dockerfile

Item	Value
spec.template.spec.containers[*].resouces.request.cpu	100m
spec.template.spec.containers[*].resouces.request.memory:	100Mi
spec.template.spec.containers[*].image:	ibmicpcoc/carbon:latest
spec.template.spec.containers[*].ports	none
Docker CMD	["/bin/bash", "-c", "./carbon.sh"]

Tasks

Task description
Within your team namespace diagnose the pod that begins with **<team> -carbon**
Use the label option -l app=<team>-carbon when getting the pod information.
Download the resource K8 yaml file.
Edit and save the file after replacing all references of <team> with your team name / namespace.
Create the K8 objects.
Did the pod deploy successfully? If not, correct the issue and re-create the K8 objects.

Hint Resources

- Describe the pod.
- Get events from the namespace, kubectl get events -n **<team>**
- A single cpu is defined with 1000m. The container cpu resources should use **1/10** of a cpu.
- Editing a running pod is another method to change the pod. Use the command KUBE_EDITOR="nano" kubectl edit deployment/**<team>-carbon** and edit the running pod. Nano is the editor defined in the command. Remove the KUBE_EDITOR parm to use the default editor on your machine.

Step-by-Step Resources

Diagnosis

When attempting to deploy the pod the yaml file is not properly defined.

Check the Pod status

```
Command:
    kubectl -n <team> get pods -l app=<team>-carbon.  # replace <team>

Example output:
    NAME                              READY     STATUS    RESTARTS   AGE
    pink-carbon-5c96bc649-tjnhb       0/1       Pending   0          2m
```

Describe the pod

```
Name:               pink-carbon-5c96bc649-tjnhb
Namespace:          pink
Priority:           0
PriorityClassName:  <none>
Node:               <none>
Labels:             app=pink-carbon
                    pod-template-hash=175267205
Annotations:        kubernetes.io/psp=ibm-privileged-psp
Status:             Pending
IP:
Controlled By:      ReplicaSet/pink-carbon-5c96bc649
Containers:
  pink-carbon:
    Image:      ibmicpcoc/carbon:latest
    Port:       <none>
    Host Port:  <none>
    Requests:
      cpu:     25
      memory:  100Mi
    Environment:
      APP_NAMESPACE:      pink (v1:metadata.namespace)
      APP_NAME:           pink-carbon-5c96bc649-tjnhb (v1:metadata.name)
      COLLECTOR_CONFIG:   <set to the key 'COLLECTOR_CONFIG' of config map 'pink-collector-config'>   Optional:
false
      INSTRUCTOR_CONFIG:  <set to the key 'INSTRUCTOR_CONFIG' of config map 'pink-collector-config'>  Optional:
false
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-mq64m (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  default-token-mq64m:
    Type:        Secret (a volume populated by a Secret)
```

```
    SecretName:  default-token-mq64m
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/memory-pressure:NoSchedule
                 node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type     Reason            Age                 From               Message
  ----     ------            ----                ----               -------
  Warning  FailedScheduling  58s (x121 over 5m)  default-scheduler  0/4 nodes are available: 4 Insufficient cpu.
$
```

In the "Events" section review the "Message" from the entry with "Type" Warning and "Reason" FailedScheduling

```
...  0/4 nodes are available: 4 Insufficient cpu.
$
```

Example of Get Events in namespace

```
Command:
    kubectl -n <team> get events

Example output:

LAST SEEN   FIRST SEEN   COUNT     NAME                                            KIND         SUBOBJECT
TYPE      REASON              SOURCE                   MESSAGE
7m          7m           1         pink-carbon.157be1efb7ad1a77                    Deployment
Normal    ScalingReplicaSet   deployment-controller     Scaled up replica set pink-carbon-5c96bc649 to 1
7m          7m           1         pink-carbon-5c96bc649.157be1efb85494ba          ReplicaSet
Normal    SuccessfulCreate    replicaset-controller     Created pod: pink-carbon-5c96bc649-tjnhb
2m          7m           121       pink-carbon-5c96bc649-tjnhb.157be1efb858b4b3    Pod
Warning   FailedScheduling    default-scheduler         0/4 nodes are available: 4 Insufficient cpu.
```

Problem discovered

Events output indicates the pod is FailedScheduling because there are not enough CPU resources available.

Resolution

At least two methods exist to correct the issue.

> *The first method is deleting the old pod, edit the yaml file, and re-create the pod.*

This approach is later referred to as: delete-create-pod

Edit the yaml file and modify *cpu* to decrease the amount of cpu to 10% of a single CPU.

Delete the running pod

```
Command to delete the existing pod:
    kubectl delete -f carbon.yaml
```

```
Result output:
    deployment.apps "carbon" deleted
```

Edit file carbon.yaml (only a portion of file shown below)

```
    spec:
      selector:
        matchLabels:
         app: <team>-carbon
     replicas: 1
     template:
       metadata:
         labels:
           app: <team>-carbon
       spec:
         containers:
         - name: <team>-carbon
           image: ibmicpcoc/carbon:latest
           resources:
             requests:
               cpu: 25000m                  <=== change value to 100m
               memory: 100Mi
```

Create the k8 deployment

```
Command:
    kubectl create -f carbon.yaml

Result output:
deployment.apps/<team>-carbon created
```

> *The second method is editing the running pod. Edit and save edit the file, and re-create the pod.*

This approach is later referred as: edit-running-pod

Edit the running pod. The kubernetes object content is available in the editor (shown below). Note the content has both the spec: and status: sections.

Locate the line cpu: "25" and change the line to cpu: 100m (without quotes)

```
Command to edit the running pod:
    KUBE_EDITOR="nano" kubectl edit deployment/<team>-carbon     # replace <team>

Content shown when editor is open.  The pink-carbon deployment is being shown:

# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: extensions/v1beta1
kind: Deployment
```

```
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: 2019-01-21T14:01:56Z
  generation: 1
  labels:
    app: pink-carbon
  name: pink-carbon
  namespace: pink
  resourceVersion: "5834141"
  selfLink: /apis/extensions/v1beta1/namespaces/pink/deployments/pink-carbon
  uid: 1d02fbe9-1d85-11e9-b012-06ed6a534df5
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: pink-carbon
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: pink-carbon
    spec:
      containers:
      - env:
        - name: APP_NAMESPACE
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.namespace
        - name: APP_NAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        - name: COLLECTOR_CONFIG
          valueFrom:
            configMapKeyRef:
              key: COLLECTOR_CONFIG
              name: pink-collector-config
        - name: INSTRUCTOR_CONFIG
          valueFrom:
            configMapKeyRef:
              key: INSTRUCTOR_CONFIG
              name: pink-collector-config
        image: ibmicpcoc/carbon:latest
        imagePullPolicy: Always
        name: pink-carbon
```

```
        resources:
          requests:
            cpu: "25"                       <=== change value to 100m without quotes
            memory: 100Mi
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
status:
  conditions:
  - lastTransitionTime: 2019-01-21T14:01:56Z
    lastUpdateTime: 2019-01-21T14:01:56Z
    message: Deployment does not have minimum availability.
    reason: MinimumReplicasUnavailable
    status: "False"
    type: Available
  - lastTransitionTime: 2019-01-21T14:11:57Z
    lastUpdateTime: 2019-01-21T14:11:57Z
    message: ReplicaSet "pink-carbon-5c96bc649" has timed out progressing.
    reason: ProgressDeadlineExceeded
    status: "False"
    type: Progressing
  observedGeneration: 1
  replicas: 1


NOTE: You must save the file for the changes to take effect.

Result output:
    deployment.extensions/pink-carbon edited
```

Did this resolve the issue?

```
Command to get pods in namespace:
    kubectl -n <namespace> get pods

Example output:
    NAME                              READY     STATUS    RESTARTS   AGE
    pink-carbon-7784b95958-pct15      1/1       Running   0          2m
```

Lab Images

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Use the debug flow to guide the steps you should attempt in diagnosis of the issue.

Resources

- K8 yaml - doors.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/doors:latest
ports	none
Docker	CMD ["node", "app.js"]

Tasks

Task description
Within your team namespace diagnose the pod that begins with <team>-doors
Use the label option -l app=<team>-doors when getting the pod status.
Download the resource K8 yaml file.
Use either of the delete-create-pod or edit-running-pod approaches to resolve the issue.
Did the pod deploy successfully? If not, correct the issue and re-create the K8 objects.

Hint Images

Check the "tag" of the image that is being pulled.

Step-by-Step Images

Diagnosis

Pod status

```
Command:
    kubectl -n <your namespace> get pods -l app=medsearch

Example output:
NAME                          READY     STATUS             RESTARTS   AGE
medsearch-78b7f6598d-p8kvf    0/1       ImagePullBackOff   0          10m
```

Describe the pod (complete output from command is shown)

```
Name:               pink-doors-778f55d487-5vvnb
Namespace:          pink
Priority:           0
PriorityClassName:  <none>
```

```
Node:               10.186.56.85/10.186.56.85
Start Time:         Mon, 21 Jan 2019 10:18:18 -0600
Labels:             app=pink-doors
                    pod-template-hash=3349118043
. . .
        portions of output removed
. . .

Events:
  Type     Reason     Age                From                    Message
  ----     ------     ----               ----                    -------
  Normal   Scheduled  46s                default-scheduler       Successfully assigned pink/pink-doors-778f55d487-
5vvnb to 10.186.56.85
  Normal   Pulling    28s (x2 over 43s)  kubelet, 10.186.56.85  pulling image "ibmicpcoc/doors:last"
  Warning  Failed     27s (x2 over 43s)  kubelet, 10.186.56.85  Failed to pull image "ibmicpcoc/doors:last": rpc
error: code = Unknown desc = Error response from daemon: manifest for ibmicpcoc/doors:last not found
  Warning  Failed     27s (x2 over 43s)  kubelet, 10.186.56.85  Error: ErrImagePull
  Normal   BackOff    12s (x3 over 42s)  kubelet, 10.186.56.85  Back-off pulling image "ibmicpcoc/doors:last"
  Warning  Failed     12s (x3 over 42s)  kubelet, 10.186.56.85  Error: ImagePullBackOff
```

Multiple Warning messages are displayed in the Event setion. Review all of the Warning messages.

In the "Events" section review the "Message" from the entry with "Type" Warning and "Reason" Failed

```
...  Failed to pull image "ibmicpcoc/doors:last": rpc error: code = Unknown desc = Error response from daemon:
manifest for ibmicpcoc/doors:last not found

(output is from the first Failed message)
```

Problem discovered

The image cannot be located as indicated by the "Failed to pull image" message. The image tag last on the container is incorrect. The image tag should be latest.

Resolution

The edit-running-pod is shown in the following example to resolve the issue:

```
Command to edit the running pod:
    KUBE_EDITOR="nano" kubectl -n <team> edit deployment/<team>-doors


Example is from the pink namespace.  Modify the tag of the image to "latest"


# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
```

```
  annotations:
    deployment.kubernetes.io/revision: "1"
  creationTimestamp: 2019-01-21T16:18:18Z
  generation: 1
  labels:
    app: pink-doors
  name: pink-doors
  namespace: pink
  resourceVersion: "5853628"
  selfLink: /apis/extensions/v1beta1/namespaces/pink/deployments/pink-doors
  uid: 29914949-1d98-11e9-b012-06ed6a534df5
spec:
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: pink-doors
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: pink-doors
    spec:
      containers:
      - env:
        - name: APP_NAMESPACE
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.namespace
        - name: APP_NAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        - name: COLLECTOR_CONFIG
          valueFrom:
            configMapKeyRef:
              key: COLLECTOR_CONFIG
              name: pink-collector-config
        - name: INSTRUCTOR_CONFIG
          valueFrom:
            configMapKeyRef:
              key: INSTRUCTOR_CONFIG
              name: pink-collector-config
        image: ibmicpcoc/doors:last            <=== change the :last to :latest
        imagePullPolicy: Always
```

```
Ensure you have saved the modified file.


Result output:
    deployment/pink-doors
```

Validate the pod status is Running.

```
Command:
    kubectl -n <team> get pods

Example output:
NAME                              READY     STATUS    RESTARTS   AGE
pink-doors-767f49c748-6gvcg       1/1       Running   0          1m
```

Lab Security

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Use the debug flow to guide the steps you should attempt in diagnosis of the issue.

Resources

- K8 yaml - avail.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/avail:latest
ports	none
Run K8 spec	command: ["/bin/bash", "-c", "/app/avail.sh"]

Tasks

Task description
Within the "avail" namespace research the pod that begins with "avail".
Why is the pod not deploying?
Review K8 definitions for controlling privleges e.g. PSP, RoleBinding, Roles etc.
Download the resource K8 yaml file.

Task description
Edit the file replacing <team> with your team name.
Create the K8 objects.

Hint Security

What rolebinding is defined for avail namespace?
What rolebinding is defined for <team> namespace?
Review the clusterroles for the cluster.
Reivew the pod security policies for the cluster.

Step-by-Step Security

Diagnosis

```
Command to check pods in namespace:

    kubectl -n avail get pods

Example output:

NAME                          READY     STATUS                       RESTARTS   AGE
avail-all-65b8448469-rqt5g   0/1       CreateContainerConfigError   0          1d

Command to describe the selected pod in the namespace:

    kubectl -n avail describe pod avail-all-65b8448469-rqt5g

Example output:

Name:               avail-all-65b8448469-rqt5g
Namespace:          avail
Priority:           0
PriorityClassName:  <none>
Node:               10.186.56.85/10.186.56.85
Start Time:         Sat, 19 Jan 2019 13:57:24 -0600
Labels:             app=avail-all
                    pod-template-hash=2164004025

    . . . < portions of the describe output not shown> . . .

Events:
      Type     Reason     Age                From                    Message
      ----     ------     ----               ----                    -------
      Normal   Scheduled  28m                default-scheduler       Successfully assigned avail/avail-698964bc87-
5k8vf to 10.186.56.85
      Normal   Pulled     26m (x8 over 28m)  kubelet, 10.186.56.85   Successfully pulled image "avail"
      Warning  Failed     26m (x8 over 28m)  kubelet, 10.186.56.85   Error: container has runAsNonRoot and image
will run as root
```

In the "Events" section review the "Message" from the entry with "Type" Warning and "Reason" Failed

```
...  Error: container has runAsNonRoot and image will run as root
```

What rolebinding are defined for the **avail** namespace?

```
Command to check rolebindings:
    kubectl get rolebinding -n avail

Example output:

No resources found.
```

Compare rolebindings for your **team** namespace.

```
Command to check rolebindings:
    kubectl get rolebinding -n <team>

Example output:

NAME                                    AGE       ROLE                                    USERS     GROUPS
SERVICEACCOUNTS
ibm-privileged-clusterrole-rolebinding   16h       ClusterRole/ibm-privileged-clusterrole
system:serviceaccounts:aqua
```

Review the clusterrole definitions for the cluster.

```
Command to view clusterrole
    k get clusterrole

Example output:

NAME                                                                   AGE
admin                                                                  17h
cluster-admin                                                          17h
edit                                                                   17h
extension                                                              17h
ibm-anyuid-clusterrole                                                 17h
ibm-anyuid-hostaccess-clusterrole                                      17h
ibm-anyuid-hostpath-clusterrole                                        17h
ibm-cert-manager-cert-manager                                          17h
ibm-privileged-clusterrole                                             17h
ibm-restricted-clusterrole                                             17h
icp-admin-aggregate                                                    17h
icp-edit-aggregate                                                     17h
icp-operate-aggregate                                                  17h
icp-view-aggregate                                                     17h

    . . . data truncated . . .
```

Describe the clusterrole for ibm-privileged-clusterrole

```
Command to describe:
    kubectl describe clusterrole ibm-privileged-clusterrole

Example output;

Name:         ibm-privileged-clusterrole
Labels:       <none>
Annotations:  kubectl.kubernetes.io/last-applied-configuration=
{"apiVersion":"rbac.authorization.k8s.io/v1","kind":"ClusterRole","metadata":{"annotations":{},"name":"ibm-
privileged-clusterrole","namespace":""},"rul...
PolicyRule:
  Resources                       Non-Resource URLs  Resource Names        Verbs
  ---------                       -----------------  --------------        -----
  podsecuritypolicies.extensions  []                 [ibm-privileged-psp]  [use]
```

Review the Pod Security Policies.

```
Command to view Pod Security Policy:

    kubectl get psp

Example output:

NAME                         PRIV      CAPS
SELINUX    RUNASUSER          FSGROUP     SUPGROUP    READONLYROOTFS   VOLUMES
ibm-anyuid-hostaccess-psp   false
SETPCAP,AUDIT_WRITE,CHOWN,NET_RAW,DAC_OVERRIDE,FOWNER,FSETID,KILL,SETUID,SETGID,NET_BIND_SERVICE,SYS_CHROOT,SETFCAP
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *
ibm-anyuid-hostpath-psp     false
SETPCAP,AUDIT_WRITE,CHOWN,NET_RAW,DAC_OVERRIDE,FOWNER,FSETID,KILL,SETUID,SETGID,NET_BIND_SERVICE,SYS_CHROOT,SETFCAP
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *
ibm-anyuid-psp              false
SETPCAP,AUDIT_WRITE,CHOWN,NET_RAW,DAC_OVERRIDE,FOWNER,FSETID,KILL,SETUID,SETGID,NET_BIND_SERVICE,SYS_CHROOT,SETFCAP
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false
configMap,emptyDir,projected,secret,downwardAPI,persistentVolumeClaim
ibm-privileged-psp          true      *
RunAsAny   RunAsAny           RunAsAny    RunAsAny    false            *
ibm-restricted-psp          false
RunAsAny   MustRunAsNonRoot   MustRunAs   MustRunAs   false
configMap,emptyDir,projected,secret,downwardAPI,persistentVolumeClaim
```

Problem discovered

The "avail" namespace does not have the proper authourity to run the "avail" pod. The avail pod must be deployed within a namespace that has the proper authority. Your team namespace has the proper authority.

Resolution

Download the K8 Yaml file from the resources section and save locally. Once saved, edit the file and change the namespace metadata parameter in the file and deploy the pod.

```
Example saved file avail.yaml (only a portion of file is shown below)

apiVersion: apps/v1
kind: Deployment
metadata:
  name: avail
  namespace: <team>       # change <team> to your namesapce and save the file

----
Command to create the new pod:
    kubectl create -f avail.yaml

Result output:
    deployment.apps/avail created


----
Verify issue is resolved. Pod status should be "Running":

Command to get pods in namespace:
    kubectl -n <team> get pods        # change <team> to your team namespace

Example output:
    avail-698964bc87-2fpw8   1/1       Running   0          1m
```

Lab Networking

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Resources

- K8 yaml - eagle.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/eagle:latest
ports	4100
Docker	CMD ["node", "server.js"]

Tasks

Task description
This lab uses the pod with a name that starts with **<team>-eagle**
The web application is not working properly. The application is has a K8 Deployment and Service defined.
Research why the web application is not working properly.
Once you have resolved the issue locate the NodePort (is a number in the 30000 range) for the service. Example: kubectl get svc -n <team> -o wide
Using the same IP that has been used to access the Collector now access the the web application using the newly located node port number. Example url to access web application: **http://xxx.xxx.xxx.xxx:NodePort**
Once the web application is successfully accessed press the button to complete the lab.

Hint Networking

- All exposed port definitions must match.
- What port should the application be available on? Refer to useful information.

Step-by-Step Networking

Diagnosis

The pod is running successfully yet describing the pod can provide information about the configured K8 objects. Describe the pod that begins with: <team>-eagle

```
Commad to get pods in namespace
    kubectl -n <team> get pods               # Replace <team> with namespace name

Command to describe the pod                  # Use the pod name from the previous output
    kubectl -n <team> describe pod <pod>     # Replace <team> with namespace name
```

Review the port definitions from the describe output

```
Show something here
```

Problem discovered

The ports do not match for the Deployment and Service definitions.

Resolution

Edit the Service definition and change the port from 4010 to 4100.

```
Add detailed steps here
```

Lab Running

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Resources

- K8 yaml - floor.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/floor:latest
ports	none
YAML	command: ["node", "app.js"]

Tasks

Task description
A container wihtin a successfully deployed pod is not working properly. Research the running container to diagnose the issue.
View the logs of the running container.
Correct the issue inside the running container.

Hint Running

- Exec into the running container
- Use touch, nano, or echo with piping to assit in resolving the issue

---

Step-by-Step Running

Diagnosis

Check the logs of the running container that begins with <team>

```
Command to get pods in namespace
    kubectl -n <team> get pods                <=== Replace <team>

Example output from "pink" namespace
    NAME                              READY     STATUS    RESTARTS   AGE
    pink-floor-6ff9f54f44-zpchp       1/1       Running   0          41s


Get the logs for the pod
    kubectl -n <team> logs -f <pod>           <=== Replace <team> and <pod>
                                              Use the pod name from the get pods result

Instructions from viewing the log

1/21/2019, 10:21:14 PM :: clnt012i - Check for file: /app/team.txt check count: 43
1/21/2019, 10:21:14 PM :: clnt013i - The file team.txt in the /app directory must exist for this lab to be
```

```
completed.
1/21/2019, 10:21:14 PM :: clnt014i - Create the file in the running container.
```

Problem discovered

The file team.txt is missing from the /app directory in the running container.

Resolution

Two methods can be used to resolve of creating the file.

> *First method is to run a "command" using the kubectl CLI from outside the container.*

```
Command to get pods in namespace
    kubectl -n <team> get pods                <=== Replace <team>

Example output from "pink" namespace
    NAME                            READY     STATUS    RESTARTS   AGE
    pink-floor-6ff9f54f44-zpchp     1/1       Running   0          41s


Add the team.txt file using the touch command from outside the container.
    kubectl exec -n pink pink-floor-6ff9f54f44-zpchp -- sh -c "touch /app/team.txt"

    The above command is using 'sh'.  The 'sh' capability must be installed in the container for this to work.


Example result output:  (wait a few seconds for the messages to show)

1/21/2019, 10:25:30 PM :: clnt014i - Create the file in the running container.
1/21/2019, 10:25:45 PM :: ------------------------------------------------------------------------------
1/21/2019, 10:25:45 PM :: clnt008i - File located.  Reporting to collector.
1/21/2019, 10:25:45 PM :: ------------------------------------------------------------------------------
1/21/2019, 10:25:45 PM :: clnt007i - Student count: 61 from /pink/pink-floor-6ff9f54f44-zpchp
1/21/2019, 10:25:45 PM :: clnt010i - Instructor count: 1 from /pink/pink-floor-6ff9f54f44-

The clnt007i and clnt010i messages are produced once the file has been loacted.
```

> *Second method is to exec into the running container and create the file from a shell prompt. This method requires 'sh' capability must be installed in the container for this to work.*

```
Command to get pods in namespace
    kubectl -n <team> get pods                <=== Replace <team>

Example output from "pink" namespace
    NAME                            READY     STATUS    RESTARTS   AGE
```

```
    pink-floor-6ff9f54f44-zpchp       1/1       Running   0          41s

Open a terminal session with the running session


Add the team.txt file using the touch command from outside the container.
    kubectl exec -it -n pink pink-floor-6ff9f54f44-zpchp -- sh

    The above command is using 'sh'.  The 'sh' capability must be installed in the container for this to work.

Example result output:
    /app #


Create the file using touch by entering the following command:
    touch team.txt

Notice the "/app" directory is not included as part of the touch command since the prompt is open to that directory.


Example result output:  (wait a few seconds for the messages to show)

1/21/2019, 10:25:30 PM :: clnt014i - Create the file in the running container.
1/21/2019, 10:25:45 PM :: ------------------------------------------------------------------------------
1/21/2019, 10:25:45 PM :: clnt008i - File located.  Reporting to collector.
1/21/2019, 10:25:45 PM :: ------------------------------------------------------------------------------
1/21/2019, 10:25:45 PM :: clnt007i - Student count: 61 from /pink/pink-floor-6ff9f54f44-zpchp
1/21/2019, 10:25:45 PM :: clnt010i - Instructor count: 1 from /pink/pink-floor-6ff9f54f44-

The clnt007i and clnt010i messages are produced once the file has been loacted.
```

Lab Starting

All references to "team" or <team> should be replaced with your team name which is the same as your namespace.

Resources

- K8 yaml - gonzo.yaml
- Dockerfile - Dockerfile

Useful information

Item	Value
cpu:	100m
memory:	100Mi
image:	ibmicpcoc/gonzo:latest
ports	none
YAML	command: ["/bin/bash", "-c", "/app/gonzo.sh"]

Tasks

Task description
A pod that begins with <team>-gonzo is failing creation.
Research the issue to determine what is causing the failure.
Edit the gonzo.yaml file to correct the issue.
Verify the deployment successfully deplloyed

Hint Starting

- What ENTRYPOINT or CMD is defined for the Docker image?
- What container "command" parameter is defined for the pod definition?
- Command: docker history ibmicpcoc/gonzo --no-trunc can also be used to check the docker image.
- The gonzo.yaml must be modified to correct the issue. You will not be allowd to rebuild or modify the Docker image.

Step-by-Step Starting

Diagnosis

```
Command to get pods in namespace
    kubectl -n <team> get pods                <=== Replace <team>

Example output from "pink" namespace
    NAME                              READY     STATUS             RESTARTS   AGE
    pink-gonzo-75d79787b7-88pnr       0/1       CrashLoopBackOff   4          2m


Command to describe pod that is failing.  Following example using above pod and pink namespace.
    kubectl describe pod pink-gonzo-75d79787b7-88pnr -n pink

Example output:

Name:               pink-gonzo-75d79787b7-88pnr
Namespace:          pink
Priority:           0
PriorityClassName:  <none>
Node:               10.186.56.85/10.186.56.85
Start Time:         Mon, 21 Jan 2019 18:13:15 -0600
Labels:             app=pink-gonzo
                    pod-template-hash=3183534363
. . .
        portions of output removed
. . .

Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
```

```
  default-token-mq64m:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-mq64m
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/memory-pressure:NoSchedule
                 node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type     Reason     Age                 From                   Message
  ----     ------     ----                ----                   -------
  Normal   Scheduled  11m                 default-scheduler      Successfully assigned pink/pink-gonzo-75d79787b7-
88pnr to 10.186.56.85
  Normal   Created    10m (x4 over 11m)   kubelet, 10.186.56.85  Created container
  Normal   Started    10m (x4 over 11m)   kubelet, 10.186.56.85  Started container
  Normal   Pulling    9m (x5 over 11m)    kubelet, 10.186.56.85  pulling image "ibmicpcoc/gonzo:latest"
  Normal   Pulled     9m (x5 over 11m)    kubelet, 10.186.56.85  Successfully pulled image "ibmicpcoc/gonzo:latest"
  Warning  BackOff    58s (x46 over 11m)  kubelet, 10.186.56.85  Back-off restarting failed container
```

In the "Events" section review the "Message" from the entry with "Type" Warning and "Reason" BackOff

```
... Back-off restarting failed container
```

Check the image for the command or entrypoint defined to execute when the container is created

```
Review the Dockerfile provided in the Resources section of this lab.

Browse the Dockerfile
    Click the Dockerfile link in resource section and review the entrypoint or command defined to start when
container is created.

(or)

Check the Docker image
    docker history ibmicpcoc/gonzo --no-trunc
```

Problem discovered

The container is ending as soon as it starts. The entrypoint or command that executes when the container starts is not defined in either the Dockerfile or gonzo.yaml file.

Resolution

Add the "command" parameter to the pod container definition using the file gonzo.yaml provided in the Resources section of this lab. The "command" parameter should start the bash script /app/gonzo.sh using /bin/bash

```
command: ["/bin/bash", "-c", "/app/gonzo.sh"]
```

```
Add the "command" parameter to the container:apiVersion: apps/v1kind: Deploymentmetadata:  name: pink-gonzo
namespace: pink  labels:    app: pink-gonzospec:  selector:    matchLabels:      app: pink-gonzo  replicas: 1
template:    metadata:      labels:        app: pink-gonzo    spec:      containers:      - name: pink-gonzo
image: ibmicpcoc/gonzo:latest        imagePullPolicy: Always        command: ["/bin/bash", "-c", "/app/gonzo.sh"]
<=== insert this line. . .  reaminder of file not shown . . .Save the modifed fileCommand to delete the current
deployed pod    kubectl -n <team> delete -f gonzo.yamlExample output:    deployment.apps/pink-gonzo deleteCommand to
deploy the updated pod    kubectl -n <team> create -f gonzo.yamlExample output:    deployment.apps/pink-gonzo
createdCommand to verify the updated pod is running    kubectl -n <team> get podsExample output:    NAME
READY     STATUS      RESTARTS   AGE    pink-gonzo-67834787b7-234xy       1/1       Running     0          2m
```