Table of Contents

Red Hat OpenShift Container Platform 4

Break-fix Practice 7

BREAK-FIX TRAINING
Complete this content by resolving a hands-on lab activity based on a customer issue or scenario.

Course: tech-breakfix-shift-005
Version: 1.0, June 2021

How to use this module:

- Look for gray < and > marks on either the bottom or the left and right sides of this pane, depending on the size of the window. Click those to navigate to the previous or next page, respectively.
- Jump to a specific page using the navigation links at the left.
- Play audio for a page using the player at the top of that page. Audio often provides more complete information than the text and graphics alone. A transcript may be available from a link on the same page.

In this break-fix:

Type of issue: An HTTP application deployment is failing due to SDN pod issue in a Red Hat OpenShift environment.

This training assumes that you have the following prior experience:

- You are a Red Hat Certified System Administrator (RHCSA) or have equivalent experience administrating Red Hat Enterprise Linux 7 or 8
- You are a Red Hat Certified Specialist in OpenShift Administration on version 4 or have equivalent experience

How break-fix training works:

1. Challenge yourself in the **Break-fix Activity**. It describes a problem scenario that's simulated in the lab environment where you can investigate and solve the problem.
2. Follow up with the **Guided Solution**:
    - If you couldn't solve the problem on your own, read and follow the instructions in the Guided Solution to complete this training.
    - If you were successful solving the problem, read the solution for useful tips you can apply when solving similar problems.

Lab environment

Successful completion for this training includes hands-on lab activities hosted in a cloud-based lab environment.

LAUNCH THE LAB ENVIRONMENT

The lab environment is labeled by the course code: **tech-breakfix-shift-005**

Use this code when following instructions to launch this lab environment:

- Instructions for Red Hat internal users using OpenTLC
- Instructions for Partner users using OpenTLC

SYSTEM INFO

System	IP	Credentials	Description
bastion	dynamic (192.168.)	Check the email from "Red Hat OPENTLC <noreply@opentlc.com>"	Central SSH access point, used to access the RHOCP 4 API
cluster-GUID-master-0	dynamic (10.0.0.)	CLI: none Web UI: kubeadmin, */home/cloud-user/cluster-GUID/auth/kubeadmin-password*	RHOCP 4 control plane, RHOCP 4 master nodes
cluster-GUID-master-1	dynamic (10.0.0.)	CLI: none Web UI: kubeadmin, */home/cloud-user/cluster-GUID/auth/kubeadmin-password*	RHOCP 4 control plane, RHOCP 4 master nodes
cluster-GUID-master-2	dynamic (10.0.0.)	CLI: none Web UI: kubeadmin, */home/cloud-user/cluster-GUID/auth/kubeadmin-password*	RHOCP 4 control plane, RHOCP 4 master node
cluster-GUID-worker-0	dynamic (10.0.0.)	None	Two RHOCP 4 worker nodes
cluster-GUID-worker-1	dynamic (10.0.0.)	None	Two RHOCP 4 worker nodes

SSH ACCESS

(1) Use the SSH command shown here to access your environment, modifying the command based on the information you received by email:

```
$ ssh lab-user@bastion.guid.dynamic.opentlc.com
```

NOTE: The user ID to use in this *ssh* line is based on your login credentials for OpenTLC, so modify that as appropriate for your account.

(2) When prompted, log in to your lab environment using one of these options:

- A password set by OpenTLC and provided in the information email.
- An SSH key pair configured as described here: http://www.opentlc.com/ssh.html

```
$ ssh flastname-redhat.com@bastion.guid.dynamic.opentlc.com
The authenticity of host 'bastion.guid.dynamic.opentlc.com (150.239.27.89)' can't be established.
ECDSA key fingerprint is SHA256:1tzVBGrbN3wzvOTsxxxxxxxxxXzOnIvR7xvjV3U+ffgks.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bastion.guid.dynamic.opentlc.com,150.239.27.89' (ECDSA) to the list of known hosts.
flastname-redhat.com@bastion.guid.dynamic.opentlc.com's password: <PASSWORD>
Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

[flastname-redhat.com@bastion ~]$ sudo su -
Last login: Mon Dec 28 12:58:04 EST 2020 from 165.227.175.77on pts/0

[root@bastion ~]#
```

▼ Show transcript

Successful completion for this training includes hands-on lab activities. Use the information on this page to launch you cloud-based lab environment, locate the URLs and credentials to access that environment, familiarize yourself with the network setup, and use SSH or a local web browser to access lab systems.

Break-fix activity

This section presents a scenario based on a real customer issue.

1. Read the scenario and success criteria.
2. Follow any instructions provided for setting up your lab environment.
3. Resolve the issue as simulated in your lab environment.
4. After resolving the issue, run the grading script and submit the completion code as prompted.

You must successfully complete this section to receive a passing grade for this module.

If you exhaust your experience and resources, and you're unable to solve the issue on your own, use the guided solution to step through one possible solution, with tips for addressing similar issues.

Background

SCENARIO

Your team manages a Red Hat OpenShift Container Platform 4 infrastructure for your company. A group of developers in the company requires a website for internal processes. To meet that requirement, one of your team members tried to deploy the *hello-openshift* application in the **test** project. That deployment failed.

Before deploying the application, your team member did some checks on worker nodes and completed the maintenance activity. Now you have to address this issue and successfully deploy *hello-openshift* application in **test** project.

Here are the errors your team member noticed (switch user to *cloud-user* for this activity):

```
[your-username@bastion ~]$ sudo su -
Last login: Mon Dec 28 12:58:04 EST 2020 from 165.227.175.77on pts/0

[root@bastion ~]#

[root@bastion ~]# su - cloud-user
Last login: Mon Dec 28 12:59:04 EST 2020 from 52.52.99.133 on pts/0
```

Your team member noticed that a hello-openshift pod is stuck in *ContainerCreating* status. We'll recreate the same scenario by running the *tech-breakfix-shift-005 break* command (more on that on the next page). So, you wouldn't be able to see any running pods as of now in the test namespace.

```
[cloud-user@bastion ~]$ oc get pods -n test
NAME                              READY   STATUS              RESTARTS   AGE
hello-openshift-54f8dbb595-qhs4f   0/1     ContainerCreating   0          4m38s
```

Your team member has also shared this additional information about the *hello-openshift* application:

```
Application Name: hello-openshift
Image used: openshift/hello-openshift:latest
Project: test
```

SUCCESS CRITERIA

Successful completion of this activity requires that you deploy the *hello-openshift* application in the **test** project. You must ensure that container is deployed successfully:

```
[cloud-user@bastion ~]$ oc get pods -n test
NAME                              READY   STATUS    RESTARTS   AGE
hello-openshift-b6d7b8dc8-lthd7   1/1     Running   0          36s
```

Activity

Here are some extra things you'll need to in your lab environment:

(1) Recreate this scenario by running the following on the central SSH access point for the learning environment (*bastion*) as cloud-user user:

```
[cloud-user@bastion ~]$ tech-breakfix-shift-005 break
Initiating tech-breakfix-shift-005 with option(s) break
....
Applying break.                DONE
Testing.......                SUCCESS
Completed tech-breakfix-shift-005 with option(s) break successfully
```

(2) Check the container status:

```
[cloud-user@bastion ~]$ oc get pods -n test
NAME                              READY   STATUS              RESTARTS   AGE
hello-openshift-54f8dbb595-qhs4f   0/1     ContainerCreating   0          4m38s
```

(3) As needed, refer to the application deployment configuration file for this deployment:

```
[cloud-user@bastion ~]$ ll /var/tmp/hello-openshift.yaml
-rw-rw-r--. 1 cloud-user cloud-user 367 Apr  6 11:11 /var/tmp/hello-openshift.yaml
```

Note: If you want to view the source code behind *hello-openshift*, see this GitHub link.

(4) After you resolve the issue, and deploy *hello-openshift* in the *test* project.

Check your work

As *cloud-user* user on *bastion*, run **tech-breakfix-shift-005 grade** to verify your work and receive a completion code. Then, submit that completion code as prompted below:

```
[cloud-user@bastion ~]$ tech-breakfix-shift-005 grade
Initiating /usr/local/bin/tech-breakfix-shift-005 with option(s) grade
Grading.  Please wait.
Checking pod status.            SUCCESS
Success.
COMPLETION CODE: <check your output for this value>
Completed /usr/local/bin/tech-breakfix-shift-005 with option(s) grade successfully
```

Submit that completion code here to receive a grade for this training:

> COMPLETION CODE provided by *tech-breakfix-shift-005 grade*: _____
>
> ans: OMIT

Guided solution

This section presents a guided solution to the break-fix activity.

Use this section to:

- Learn one possible path to resolving the customer issue.
- Get some tips for investigating and resolving similar issues.

If you're still stuck after exhausting your expertise and resources on the break-fix activity:

- Use this solution to complete the activity.
- Check your work when you finish to receive a grade for this activity.

Solution (1 of 2)

The steps detailed in these solution pages are optional. They provide additional guidance for those struggling with the Break-fix Activity. **We encourage you to only use these pages if you are completely stuck in solving the issue.**

(1) If you previously ran the *tech-breakfix-shift-005 break* command, run this fix command to return the environment to working order:

```
[cloud-user@bastion ~]$ tech-breakfix-shift-005 fix
Initiating /usr/local/bin/tech-breakfix-shift-005 with option(s) fix
Fixing will take more than 2 minutes
Removing break......            DONE
Applying solution......            DONE
Testing.                SUCCESS
Fixing.                    SUCCESS
Completed /usr/local/bin/tech-breakfix-shift-005 with option(s) fix successfully
```

If the fix argument doesn't restore your environment, you'll need to deploy a new lab environment from OpenTLC.

Note

After you use this option to restore the system, you'll sill have to break and fix the system to receive a completion code from the grading script. We've built that in to this guided solution.

(2) To explore this environment, export the KUBECONFIG variable using the configuration created by the installer:

```
[cloud-user@bastion ~]$ export KUBECONFIG=~/.kube/config
```

(3) Check the container status:

```
[cloud-user@bastion ~]$ oc get all -n test
NAME                                  READY   STATUS    RESTARTS   AGE
pod/hello-openshift-54f8dbb595-tkjcz   1/1     Running   0          2m27s

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-openshift   1/1     1            1           2m27s

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-openshift-54f8dbb595   1         1         1       2m27s
```

(4) Check the events of running *hello-openshift* pod and notice that it was stuck at the *ContainerCreating* state because of a failure in creating the pod network:

```
[cloud-user@bastion ~]$ oc describe pod/hello-openshift-559f97ff44-t8jk8 -n test
...
...
...
Events:
  Type    Reason          Age    From               Message
  ----    ------          ----   ----               -------
  Normal  Scheduled       2m53s  default-scheduler  Successfully assigned test/hello-openshift-54f8dbb595-tkjcz to cluster-bjh
sf-kcsx2-worker-0-gp58f
  Normal  AddedInterface  2m30s  multus             Add eth0 [10.131.0.221/23]
  Normal  Pulling         2m30s  kubelet            Pulling image "openshift/hello-openshift:latest"
  Normal  Pulled          2m29s  kubelet            Successfully pulled image "openshift/hello-openshift:latest" in 1.04634484
1s
  Normal  Created         2m29s  kubelet            Created container hello-openshift1
  Normal  Started         2m29s  kubelet            Started container hello-openshift1
```

(5) Check the SDN and OVS pods' statuses on the worker node where application is deployed:

```
[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-gp58f
ovs-w64td              1/1     Running   0          3m3s    10.0.0.91    cluster-bjhsf-kcsx2-worker-0-gp58f   <none>
 <none>
sdn-zt464              2/2     Running   0          3m29s   10.0.0.91    cluster-bjhsf-kcsx2-worker-0-gp58f   <none>
 <none>

[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-k8dmf
ovs-6vxrk              1/1     Running   0          2m29s   10.0.2.168   cluster-bjhsf-kcsx2-worker-0-k8dmf   <none>
 <none>
sdn-r82j6              2/2     Running   0          2m59s   10.0.2.168   cluster-bjhsf-kcsx2-worker-0-k8dmf   <none>
 <none>
```

These pods must have restarted during the fix, causing the old logs to be inaccessible.

Next, you'll run the break command again to find out what could had happened.

Solution (2 of 2)

Break the environment again, and step through the fix:

(6) Apply the break as at the beginning of this break-fix activity:

```
[cloud-user@bastion ~]$ tech-breakfix-shift-005 break
Initiating /usr/local/bin/tech-breakfix-shift-005 with option(s) break
....
Applying break.                              DONE
Testing (two minutes please).......    SUCCESS
Completed /usr/local/bin/tech-breakfix-shift-005 with option(s) break successfully
```

(7) Verify that you're working in the **test** project space:

```
[cloud-user@bastion ~]$ oc project test
Now using project "test" on server "https://api.cluster-96b1.dynamic.opentlc.com:6443".
```

(8) List all resource objects in the *test* namespace:

```
[cloud-user@bastion ~]$ oc get all -n test
NAME                                   READY   STATUS              RESTARTS   AGE
pod/hello-openshift-54f8dbb595-qhs4f   0/1     ContainerCreating   0          6m55s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-openshift     0/1     1            0           6m55s

NAME                                           DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-openshift-54f8dbb595     1         1         0       6m55s
```

(9) Check the pod logs, and note that the container is at *waiting to start*:

```
[cloud-user@bastion ~]$ oc logs pod/hello-openshift-54f8dbb595-qhs4f -n test
Error from server (BadRequest): container "hello-openshift1" in pod "hello-openshift-54f8dbb595-qhs4f" is waiting to start: Co
ntainerCreating
```

(10) Verify similar information when you run *oc describe pod*:

```
[cloud-user@bastion ~]$ oc describe pod/hello-openshift-54f8dbb595-qhs4f -n test
Name:           hello-openshift-54f8dbb595-qhs4f
Namespace:      test
Priority:       0
Node:           cluster-bjhsf-kcsx2-worker-0-gp58f/10.0.0.91
Start Time:     Tue, 06 Apr 2021 11:11:37 -0400
Labels:         app=hello-openshift
                pod-template-hash=54f8dbb595
Annotations:    openshift.io/scc: restricted
Status:         Pending
...
...
...
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
...
...
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
...
...
Events:
  Type     Reason                  Age        From                                                Message
  ----     ------                  ----       ----                                                -------
  ...
  ...
  ...
  Warning  FailedCreatePodSandBox  77s    kubelet            Failed to create pod sandbox: rpc error: code = Unknown desc = fa
iled to create pod network sandbox k8s_hello-openshift-54f8dbb595-qhs4f_test_d5ddc524-7ca2-416b-a597-949ad3183673_0(d991c21ac7
1c5e0390fba200de3ec5236199542071e951f62c14ba2bee88a16a): Multus: [test/hello-openshift-54f8dbb595-qhs4f]: PollImmediate error
waiting for ReadinessIndicatorFile: timed out waiting for the condition
```

Notice here that *hello-openshift* application is **Waiting**.

(11) Next, check the SDN and OVS pods' statuses on the worker node where the application is being deployed and on another worker node:

```
[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-k8dmf
ovs-bt7sx   0/1  Running   0  21h   10.0.2.168   cluster-bjhsf-kcsx2-worker-0-k8dmf   <none>     <none>
sdn-7fw7h   1/2  CrashLoopBackOff   6  21h  10.0.2.168  cluster-bjhsf-kcsx2-worker-0-k8dmf   <none>  <none>

[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-gp58f
ovs-6wpml   0/1   Running   0   21h   10.0.0.91    cluster-bjhsf-kcsx2-worker-0-gp58f   <none>   <none>
sdn-snv29   1/2  CrashLoopBackOff   6   21h  10.0.0.91   cluster-bjhsf-kcsx2-worker-0-gp58f   <none>  <none>
```

(12) Let's look at the logs for these pods to find more information:

```
[cloud-user@bastion ~]$  oc logs ovs-bt7sx -n openshift-sdn
2021-04-06T15:11:27.675Z|00001|vlog|INFO|opened log file /var/log/openvswitch/ovsdb-server.log
2021-04-06T15:11:27.676Z|00002|daemon_unix|EMER|/var/run/openvswitch/ovsdb-server.pid.tmp: create failed (Permission denied)

[cloud-user@bastion ~]$ oc logs ovs-6wpml -n openshift-sdn
2021-04-06T15:11:27.229Z|00001|vlog|INFO|opened log file /var/log/openvswitch/ovsdb-server.log
2021-04-06T15:11:27.230Z|00002|daemon_unix|EMER|/var/run/openvswitch/ovsdb-server.pid.tmp: create failed (Permission denied)

[cloud-user@bastion ~]$ oc logs sdn-snv29 -n openshift-sdn -c sdn
...
...
I0406 15:28:33.153906 2504228 healthcheck.go:42] waiting for OVS to start: dial unix /var/run/openvswitch/db.sock: connect: co
nnection refused
I0406 15:28:34.154169 2504228 healthcheck.go:42] waiting for OVS to start: dial unix /var/run/openvswitch/db.sock: connect: co
nnection refused

[cloud-user@bastion ~]$ oc logs sdn-7fw7h -n openshift-sdn -c sdn
...
...
I0406 15:28:23.013325 3846394 healthcheck.go:42] waiting for OVS to start: dial unix /var/run/openvswitch/db.sock: connect: co
nnection refused
I0406 15:28:24.013618 3846394 healthcheck.go:42] waiting for OVS to start: dial unix /var/run/openvswitch/db.sock: connect: co
nnection refused
```

It appears that the OVS and SDN pods are having some problem.

It looks like there is a permission issue in "/var/run/openvswitch/" on the worker nodes. Remember in the scenario that there was some maintenance activity done on the worker nodes. That might have resulted in this issue.

(13) Let's debug the both worker nodes:

```
[cloud-user@bastion ~]$ oc debug node/cluster-bjhsf-kcsx2-worker-0-gp58f
Creating debug namespace/openshift-debug-node-gd4k9 ...
Starting pod/cluster-bjhsf-kcsx2-worker-0-gp58f-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.0.0.91
If you don't see a command prompt, try pressing enter.
sh-4.4# chroot /host
sh-4.4#
```

Reminder: Red Hat doesn't recommend using SSH to directly access a node. Instead, use *oc debug node/* to create a debug pod for the node.

(14) Check the permissions of the "/var/run/openvswitch/" directory. You should see that it has permissions set to 0555:

```
sh-4.4# ls -al /var/run/openvswitch/
total 0
d--x--x--x.  2 openvswitch hugetlbfs    140 Apr  6 15:11 .
drwxr-xr-x. 45 root        root        1180 Apr  6 15:11 ..
srwxr-x---.  1 openvswitch hugetlbfs      0 Apr  5 17:27 br0.mgmt
...
...
...
```

(15) Fix the directory permissions to *0755*, and restart the *openvswitch* service:

```
sh-4.4# chmod 0755 /var/run/openvswitch/
sh-4.4# ls -al /var/run/openvswitch/
total 0
drwxr-xr-x.  2 openvswitch hugetlbfs    140 Apr  6 15:11 .
drwxr-xr-x. 45 root        root        1180 Apr  6 15:11 ..
srwxr-x---.  1 openvswitch hugetlbfs      0 Apr  5 17:27 br0.mgmt
...
...
sh-4.4# systemctl restart openvswitch
```

(16) Repeat steps 13-15 for another worker node.

(17) After you make these changes, both the pods should now be in running state. In case if you face any issue, restart the OVS and SDN pods on those worker nodes:

```
[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-k8dmf
ovs-bt7sx   0/1  Running   0   22h   10.0.2.168   cluster-bjhsf-kcsx2-worker-0-k8dmf   <none>    <none>
sdn-7fw7h   1/2  CrashLoopBackOff   9  22h   10.0.2.168  cluster-bjhsf-kcsx2-worker-0-k8dmf  <none>   <none>

[cloud-user@bastion ~]$ oc get pods -o wide -n openshift-sdn | grep -i cluster-bjhsf-kcsx2-worker-0-gp58f
ovs-6wpml  0/1   Running    0  22h   10.0.0.91    cluster-bjhsf-kcsx2-worker-0-gp58f   <none>   <none>
sdn-snv29  1/2   CrashLoopBackOff   8  22h   10.0.0.91   cluster-bjhsf-kcsx2-worker-0-gp58f   <none>   <none>

[cloud-user@bastion ~]$ oc delete pod ovs-bt7sx sdn-7fw7h ovs-6wpml sdn-snv29 -n openshift-sdn
pod "ovs-bt7sx" deleted
pod "sdn-7fw7h" deleted
pod "ovs-6wpml" deleted
pod "sdn-snv29" deleted
```

(18) Check the status of *hello-openshift* application and confirm that the issue is fixed:

```
[cloud-user@bastion ~]$ oc get pods -n test
NAME                              READY   STATUS    RESTARTS   AGE
hello-openshift-6c8bf774bb-rg2zk   1/1     Running   0          15m
```

(19) Return to the Check your work page and run the grading command again to verify your work and receive the COMPLETION CODE. Submit that code as prompted on that page.

Resources

This list includes content referenced during research for this training plus links you can use for ongoing reference or additional learning:

CrashLoopBackOff status for Openshift Pod (Solution)

Master Troubleshooting Article OpenShift Container Platform 4.x

Troubleshooting OpenShift Container Platform 4.x: openshift-sdn (Solution)

Thanks for your feedback!

Did you like it? Have suggestions?
Let us know so we can continue enhancing our training courses.

Provide Feedback