

Platform health and logs

This section contains the following topics:

• Platform health and logs, on page 1

Platform health and logs

CWM is a microservice-based application that leverages Kubernetes cluster architecture as its runtime environment. The health of the CWM application can thus be checked using Kubernetes commands.



Note To see all the supported kubect1 commands, log in to the OS on your VM and use kubect1 --help.

Check pod status

- Step 1 Using a command-line terminal, log in to the OS on your virtual machine with SSH: ssh -o UserKnownHostsFile=/dev/null -p 22 nxf@<your_resource_pool_address>
 Step 2 To check status of pods for namespace zone-a (this is the default namespace for pods contaning CWM microservices), run the following command: kubectl get pods -n zone-a
- **Step 3** A list of pods will appear:

Figure 1: Get k8s pods

💿 🔘 💿 🔤 🗕 nxf@wf-nat	33:~ — s	sh -o UserK	nownHostsFile	=/dev/null
~ % ssh -	o UserKno	ownHostsFi]	Le=/dev/null	-p 8332 nxf
wf-nat.lab.tail-f.com				•
The authenticity of host '[wf-nat.l	ab.tail-	f.com]:8332	2 ([10.147.44.	16]:8332)' c
n't be established.				
ED25519 key fingerprint is				
This key is not known by any other	names			
Are you sure you want to continue c	onnectin	g (yes/no/[fingerprint])	? yes
Warning: Permanently added '[wf-nat	.lab.tai	1-f.com]:83	332' (ED25519)	to the list
of known hosts.			_	
Last login: Tue May 23 13:45:51 202	3 from 10	0.61.193.45)	
[[nxt@wt-nat33 ~]\$ kubect1 get pods	-n zone-a		DECTADIO	405
NAME	READY	STATUS	RESTARTS	AGE
api-service-c/8DC8TC8-KD88T	2/2	Running	3 (10d ago)	100
ası-service-//48a8d4b-mbnqx	2/2	Running	4 (10d ago)	100
logcl1-b4494db6-zdv6j	2/2	Running	0	10d
piugin-manager-6655c99df9-vn6jw	2/2	Running	1 (10d ago)	10d
ui-service-/cdb49/b/c-sf6/8	2/2	Running	0	10d
worker-manager-68c9/9t99/-64n4q	2/2	Running	2 (10d ago)	10d
workflow-frontend-bd9c4c554-xdsrd	2/2	Running	2 (10d ago)	10d
worktlow-history-8589b95t9f-kcgws	2/2	Running	2 (10d ago)	10d
workflow-matching-644498b786-zwqfr	2/2	Running	2 (10d ago)	10d
workflow-u1-78d5f9df58-b249v	2/2	Running	0	10d
workflow-worker-977fc69dc-6rx9b	2/2	Running	2 (10d ago)	10d
Lnxf@wf−nat33 ~]\$				

Step 4 If a pod has a status different from Running, you can 'restart' it using the following command:

kubectl delete pod <pod name> -n zone-a

The pod will be deleted, but as Kubernetes configuration is declarative, it will effectively recreate the deleted pod and rerun it.

Check and collect logs

Application logs can be checked with **Loki logCLI** command-line interface. To gather logs from the CWM platform, follow these steps:

 Step 1
 Using a command-line terminal, connect to the system using SSH client:

 ssh
 -pSSH_PORT nxf@ip_address_of_deployment

 Note
 Adjust SSH_PORT and ip_address_of_deployment accordingly.

 Step 2
 After successful login, use the command below to list all running pods:

 kubectl get pods
 -A

Example result:

[nxf@wf-nat-08 ~]\$ kubectl get pods -A NAMESPACE NAME AGE

READY STATUS RESTARTS

kube-flannel	kube-flannel-ds-trr95	1/1	Running	0
103m		1 / 1	Dunning	0
103m	coreans-nig9j	1/1	Running	0
kube-system	etcd-wf-nat-08	1/1	Running	0
103m kube-system	kube-apiserver-wf-nat-08	1/1	Running	0
103m	who controllor monogor of not 00	1 / 1	Dupping	0
103m	Kube-concrorrer-manager-wi-mac-00	1/1	Kullillig	0
kube-system 103m	kube-proxy-c25f5	1/1	Running	0
kube-system 103m	kube-scheduler-wf-nat-08	1/1	Running	0
local-path-storage 103m	local-path-provisioner-6fb6f599c7-ckcjc	1/1	Running	0
nxf-system 102m	authenticator-5db8885675-qlrmg	2/2	Running	0
nxf-system 102m	controller-cbd87f8c5-6tg6f	2/2	Running	1 (102m ago)
nxf-system	ingress-proxy-56f7c9899d-6st6j	1/1	Running	0
nxf-system	kafka-0	1/1	Running	0
nxf-system	loki-7c994678f8-fnrs9	3/3	Running	0
nxf-system	minio-0	2/2	Running	0
nxf-system	postgres-0	2/2	Running	0
102m nxf-system	promtail-v6tb4	1/1	Running	0
102m nxf-system	registry-7dd84db44f-n5q7h	2/2	Running	0
102m nxf-system	vip-wf-nat-08-28131000-772k5	0/1	Completed	0
3m42s zone-a	api-service-745759bffc-v6r25	2/2	Running	2 (100m ago)
100m zone-a	dsl-service-77d5fc96cc-5nv42	2/2	Running	3 (100m ago)
100m			5	, <u> </u>
zone-a 100m	logcli-5c7ddbc95d-mkpcc	2/2	Running	0
zone-a 100m	plugin-manager-665b7bbd4d-jvqdk	2/2	Running	1 (100m ago)
zone-a 100m	ui-service-57cf6d6bcc-smmvt	2/2	Running	0
zone-a 100m	worker-manager-6d6b445d46-r6nzk	2/2	Running	1 (99m ago)
zone-a	workflow-frontend-77bc897549-kcz5k	2/2	Running	1 (99m ago)
zone-a	workflow-history-58bdb85b8d-88t25	2/2	Running	1 (99m ago)
zone-a	workflow-history-58bdb85b8d-h22bd	2/2	Running	1 (99m ago)
zone-a	workflow-history-58bdb85b8d-ph5fh	2/2	Running	1 (99m ago)
zone-a	workflow-matching-86cfc5577c-4mxhb	2/2	Running	1 (99m ago)
100m				
zone-a 100m	workflow-ui-68f857645-9mq9v	2/2	Running	0
zone-a 100m	workflow-worker-8496898f7b-wcrqs	2/2	Running	1 (99m ago)

Step 3 Identify the logcli tool available in the zone-a namespace. In this example, it is the pod named logcli-5c7ddbc95d-mkpcc.

Step 5

Step 6

Step 4 Connect to the correct pod and list the available log labels for filtering:

```
kubectl exec --namespace=zone-a -ti logcli-5c7ddbc95d-mkpcc -- logcli labels
app
container
filename
level
namespace
node_name
pod
stream
Gather logs from all applications running in the zone-a namespace and save them to a single file. Make sure to adjust
the --since option to collect logs from the relevant time period when the troubleshooting event occurred:
kubectl exec --namespace=zone-a -ti logcli-5c7ddbc95d-mkpcc -- logcli query '{namespace="zone-a"}'
--since 60m > zone-a.log
Similarly, collect logs from other namespaces, using different files for convenience:
kubectl exec --namespace=zone-a -ti logcli-5c7ddbc95d-mkpcc -- logcli query '{namespace="nxf-system"}'
 --since 60m > nxf-system.log
kubectl exec --namespace=zone-a -ti logcli-5c7ddbc95d-mkpcc -- logcli query '{namespace="kube-system"}'
 --since 60m > kube-system.log
```

Step 7 Use the SCP tool to copy the log files from the system to your desktop:

scp -P SSH PORT nxf@ip address of deployment:"*.log".

Step 8 Finally, you can send the logs to support and provide a detailed description of the issue you are experiencing.

Note For more details on the logCLI commands and usage, refer to logCLI Grafana documentation.