Note:
This is the latest CD version of IBM Cloud Pak for Integration. For the SC-2 (long term support) release, see What's new in Cloud Pak for Integration 16.1.0.

Troubleshooting

Last Updated: 2025-12-17

The must-gather tool gathers information about IBM Cloud Pak® for Integration and the OpenShift cluster where it is installed. The information that is collected is required by IBM Support for troubleshooting user issues.

Before you open a case with IBM Support, get the logs and other important information by using the must-gather tool.

Gathering diagnostic information

To help IBM Support troubleshoot any issues with your Cloud Pak for Integration installation, gather:

- Cloud Pak For Integration information
- IBM Cloud Pak foundational services logs only
- Red Hat OpenShift Container Platform logs
- IBM App Connect logs

To get troubleshooting information and workarounds for issues you may encounter, see Common issues and solutions.

Before you begin

Meet the following prerequisites:

- You have admin permissions, which are required to run the must-gather tool.
- The OpenShift Container Platform CLI (`oc`) is installed. For more information, see Installing the OpenShift CLI.
- You have access to the must-gather image. For more information, see Using MustGather diagnostics through the command line for cluster admin in the IBM Cloud Paks documentation.

Gathering Cloud Pak for Integration information

Running the must-gather tool as an administrator

When you run the must-gather tool with admin permissions, the tool automatically finds namespaces where instances are installed and collects all the required logs. It also collect logs from any namespace with the prefix `openshift-` in the name.

If you are installing Cloud Pak for Integration in an air-gapped or disconnected environment, skip to the "Air-gapped and disconnected environments" section.

1. Log in to the OpenShift cluster:

```
oc login
```

2. Go to the directory where you want to store the must-gather data.

3. Collect the must-gather information. Make sure you include the quotation marks in the command.

   - To collect Cloud Pak for Integration must-gather only, run:

     ```
     oc adm must-gather --image=icr.io/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-c cp4i'"
     ```

   - To collect Cloud Pak for Integration must-gather with a version table, run:

     ```
     oc adm must-gather --image=icr.io/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-c cp4i -v True'"
     ```

   - To get additional options (help), run:

     ```
     oc adm must-gather --image=icr.io/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-h'"

     options:
     -h, --help          Show this help message and exit
     -n NAMESPACE, --namespace NAMESPACE
                         Namespace you want to collect the logs for, only works with -c appconnect
     -p PODNAME, --podname PODNAME
                         Pod name you want to collect logs for, only works with -c appconnect
     -c COMPONENT, --component COMPONENT
                         Component you want to collect logs for. For ACE = appconnect, CP4I = cp4i
     -v VERSION, --version VERSION
                         To get a table of all the versions installed, set -v True
     --previous_pod_logs, PREVIOUS POD LOGS
                         To collect pod logs for terminated pods, set --previous_pod_logs True
     -ne, --namespaced_events NAMESPACED EVENTS
                         To collect events per namespace, set -ne True
     ```

Running the must-gather tool as a non-administrator user

If you do not have admin permissions, you can collect logs only from the namespaces where you have user permissions.

1. Log in to the OpenShift cluster:

   ```
   oc login
   ```

2. Run the following command to create a directory called `mustgather` (where you will store the logs that are collected) and to apply permissions so that logs can be written to the directory. This directory is mounted when you run the command for the must-gather tool in the last step.

   ```
   mkdir -p mustgather
   chmod 777 mustgather

   Change the dir to mustgather folder

   cd mustgather
   ```

3. Collect the logs. For the value of `namespaces`, enter a list of comma-separated namespaces (to which you have permissions) that you want to collect logs from.

   ```
   image_name=icr.io/cpopen/cpfs/must-gather:latest

   namespaces=<my_namespaces>
   ```

   For example:

```
namespaces=navigator-ns,ibm-common-services,apic-ns,ace-dashboard-ns,ace-integrationserver-ns
```

Now run the must-gather tool. Make sure that you include the quotation marks in the command.

```
podman run --platform linux/amd64  --pull=always -v ${KUBECONFIG:-~/.kube/config}:/kube/config -v $(pwd):/tmp/must-gather \
--env MYENV=NONE --env KUBECONFIG=/kube/config \
--env MUSTGATHERPATH=/tmp/must-gather \
${image_name} gather -m cp4i --params "-c cp4i --non_admin True -n ${namespaces}"
```

Air-gapped and disconnected environments

This section applies to installations in air-gapped and disconnected environments only.

You can use the must-gather tool in an air-gapped environment because when you mirrored the Cloud Pak foundational services images (using the procedure in Mirroring images with a bastion host or Mirroring images with a portable compute or storage device (file system)), the must-gather image was mirrored to your local registry.

1. Set the registry where your local images are mirrored:

```
export LOCAL_REGISTRY=<YOUR_LOCAL_REPOSITORY>
```

2. Collect the must-gather information. Make sure you include the quotation marks in the command.

   - To collect Cloud Pak for Integration must-gather only, run:

```
oc adm must-gather --image=${LOCAL_REGISTRY}/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-c cp4i'"
```

   - To collect Cloud Pak for Integration must-gather with a version table, run:

```
oc adm must-gather --image=${LOCAL_REGISTRY}/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-c cp4i -v True'"
```

   - To get additional options (help), run:

```
oc adm must-gather --image=${LOCAL_REGISTRY}/cpopen/cpfs/must-gather:latest -- gather -m cp4i --params "'-h'"

options:
-h, --help          Show this help message and exit
-n NAMESPACE, --namespace NAMESPACE
                    Namespace you want to collect the logs for, only works with -c appconnect
-p PODNAME, --podname PODNAME
                    Pod name you want to collect logs for, only works with -c appconnect
-c COMPONENT, --component COMPONENT
                    Component you want to collect logs for. For ACE = appconnect, CP4I = cp4i
-v VERSION, --version VERSION
                    To get a table of all the versions installed, set -v True
 --previous_pod_logs, PREVIOUS POD LOGS
                      To collect pod logs for terminated pods, set --previous_pod_logs True
 -ne, --namespaced_events NAMESPACED EVENTS
                      To collect events per namespace, set -ne True
```

   - To confirm that you have mirrored the latest available version of the must-gather tool, run:

```
skopeo list-tags docker://[LOCAL_REGISTRY:5000]/cpopen/cpfs/must-gather
{
    "Repository": "${LOCAL_REGISTRY}/cpopen/cpfs/must-gather",
    "Tags": [
        "4.5.16"
        .....
```

```
            "4.6.7",
            "4.6.8",
            "4.6.9",
            "latest"
        ]
    }
```

Information that is collected

The must-gather tool collects information about the following resources:

```
Certificate
Clients
ConfigMaps
Cp4iServiceBindings
Cloud Pak CustomResources
Deployments
InstallPlan
Jobs
OperandRequests
PersistentVolumeClaims
Pod-Description
Pod-Logs
ReplicaSets
Routes
Services
StatefulSets
Subscriptions
CatalogSources
ServiceAccounts
Events
OperandBindInfo
OperandRegistries
OperandConfigs
Issuer
```

It collects information from any namespace where operators and instances are installed, as well as the following namespaces:

```
ibm-common-services
openshift-marketplace
openshift-operator-lifecycle-manager
openshift-operators
openshift-ingress
kube-public
```

The must-gather tool collects only the name of the secrets from the listed namespaces. It doesn't collect the data in those secrets.

Running the must-gather on Azure Kubernetes Service

The script in this section sets up a kubeconfig file (which provides necessary information to the kubectl command-line tool) and runs the mustgather tool for you.

Prior to running the mustgather tool, the script handles these steps:

1. Downloads the current kubeconfig from the AKS cluster and saves it to a file named `temp-kubeconfig.yaml`.
2. Extracts these key Azure AD authentication parameters: `--client-id`, `--tenant-id`, and `--server-id`. The script uses these values with `kubelogin` to obtain a raw Azure AD access token outside of the container. This step is necessary because AKS uses Azure AD for authentication, and running `az login` inside a container isn't practical. By using the Azure AD access token outside of the container (along with the cluster's base64-encoded certificate authority data and API server endpoint), the script generates a new static kubeconfig file. This kubeconfig can be used within the mustgather container to securely authenticate with the AKS cluster, without having to rely on dynamic Azure CLI logins.

> **Tip:** The static kubeconfig is valid for 1 hour. If you are unable to collect the logs in the first hour, you must run the script again.

f

3. Log in to your Azure account: `az login`
4. Run the following commands.

```
mkdir -p mustgather
chmod 777 mustgather
cd mustgather
```

The first command creates a directory called `mustgather`, which stores the logs that are collected. The second command applies permissions so that logs can be written to the directory. The third command changes the directory to the `mustgather` folder. This directory is mounted when you run the command.

5. Copy and paste the script after updating the values as indicated.
   - RESOURCE_GROUP: Resource group for the AKS cluster
   - CLUSTER_NAME: Name of your AKS cluster
   - NAMESPACE: Default namespace for the kubeconfig file
   - NAMESPACES_FOR_MG: Comma-separated list of Kubernetes namespaces to gather logs from

```
#!/bin/bash

# CONFIGURATION
RESOURCE_GROUP="<aks_cluster_resource_group>"
CLUSTER_NAME="<aks_cluster_name>"
NAMESPACE="<kubeconfig_namespace>"
KUBECONFIG_FILE="temp-kubeconfig.yaml"
STATIC_KUBECONFIG="static-kubeconfig.yaml"
IMAGE="icr.io/cpopen/cpfs/must-gather:latest"
NAMESPACES_FOR_MG="<namespaces_for_gathering_logs>"

# Colors for output
GREEN='\033[0;32m'
NC='\033[0m' # No Color

set -e

echo -e "${GREEN}🔹 Getting AKS kubeconfig...${NC}"
az aks get-credentials \
--resource-group "$RESOURCE_GROUP" \
--name "$CLUSTER_NAME" \
--file "$KUBECONFIG_FILE" \
--overwrite-existing

echo -e "${GREEN}🔹 Extracting Azure AD parameters from kubeconfig...${NC}"
ARGS=$(kubectl config view --raw --kubeconfig="$KUBECONFIG_FILE" -o jsonpath="{.users[0].user.exec.args[*]}")

echo -e "${GREEN}🔍 Debug: ARGS = $ARGS${NC}"

SERVER_ID=$(echo "$ARGS" | sed -n 's/.*--server-id \([^ ]*\).*/\1/p')
CLIENT_ID=$(echo "$ARGS" | sed -n 's/.*--client-id \([^ ]*\).*/\1/p')
TENANT_ID=$(echo "$ARGS" | sed -n 's/.*--tenant-id \([^ ]*\).*/\1/p')

if [[ -z "$SERVER_ID" || -z "$CLIENT_ID" || -z "$TENANT_ID" ]]; then
echo "X Failed to extract Azure AD auth parameters from kubeconfig."
exit 1
fi

echo "SERVER_ID = ${SERVER_ID}"
echo "CLIENT_ID = ${CLIENT_ID}"
echo "TENANT_ID" = ${TENANT_ID}

echo -e "${GREEN}🔹 Getting Azure access token...${NC}"
ACCESS_TOKEN=$(kubelogin get-token \
--login azurecli \
--environment AzurePublicCloud \
--server-id "$SERVER_ID" \
--client-id "$CLIENT_ID" \
--tenant-id "$TENANT_ID" | jq -r .status.token)

echo "ACCESS_TOKEN: ${ACCESS_TOKEN}"

if [[ -z "$ACCESS_TOKEN" ]]; then
echo "X Failed to retrieve token from kubelogin."
exit 1
fi

echo -e "${GREEN}🔹 Extracting CA certificate from kubeconfig...${NC}"
CA_DATA=$(kubectl config view --raw --kubeconfig="$KUBECONFIG_FILE" \
-o jsonpath="{.clusters[0].cluster.certificate-authority-data}")

SERVER=$(kubectl config view --raw --kubeconfig="$KUBECONFIG_FILE" \
-o jsonpath="{.clusters[0].cluster.server}")

echo -e "${GREEN}🔹 Writing static kubeconfig to $STATIC_KUBECONFIG...${NC}"

cat <<EOF > "$STATIC_KUBECONFIG"
apiVersion: v1
kind: Config
clusters:
- cluster:
    certificate-authority-data: $CA_DATA
    server: $SERVER
name: $CLUSTER_NAME
contexts:
- context:
    cluster: $CLUSTER_NAME
    user: token-user
    namespace: $NAMESPACE
name: $CLUSTER_NAME
current-context: $CLUSTER_NAME
users:
- name: token-user
user:
    token: $ACCESS_TOKEN
EOF

echo -e "${GREEN}✅ Static kubeconfig saved to $STATIC_KUBECONFIG${NC}"
echo -e "${GREEN}ℹ️ Token is valid for approximately 1 hour.${NC}"

read -p "Press any key to run the mustgather"

podman run  \
--platform linux/amd64 \
--pull=IfNotPresent \
-v $(pwd)/${STATIC_KUBECONFIG}:/kube/config:ro \
-v "$(pwd):/tmp/must-gather" \
--env MYENV=NONE \
--env KUBECONFIG=/kube/config \
--env MUSTGATHERPATH=/tmp/must-gather \
"${IMAGE}" \
gather -m cp4i --params "-c cp4i --non_admin True -n ${NAMESPACES_FOR_MG}"
```

6. Give the permissions to the script chmod `+x integration-must-gather.sh`
7. Run the script `./integration-must-gather.sh`

Gathering Cloud Pak foundational services logs only

The script in the previous section, Gathering extended Cloud Pak For Integration information, also returns the pod logs for Cloud Pak foundational services.

However, if you need to work directly with the Cloud Pak foundational services support team, see Collecting Cloud Pak Cluster and Common services for problem determination in the Support documentation. This page describes how to collect logs that are needed by the team that investigates your issue.

Gathering OpenShift Container Platform logs

OpenShift has its own must-gather tool. To gather debugging information about your OpenShift, see Gathering data about your cluster).

Gathering App Connect logs

For information on collecting must-gather information that is specific to App Connect, see Gathering diagnostic information.

Common issues and workarounds

To get specific information on troubleshooting for limitations and common issues in Cloud Pak for Integration, see Common issues and solutions.