WKC 4.6.5 Hotfix 2 April 2024 for Verizon Installation Instructions

Follow **Step 1** and the sub-steps to download and copy the images to a local private registry for an air-gapped environment.

Follow **Step 2** and the sub-steps to go through applying the patch using the online IBM entitled registry, or to apply the hotfix using the images downloaded to the local private registry from Step 1.

Procedure

1) To apply the patch in an air-gapped environment, proceed with the following steps.
   a) Log in to the OpenShift console as the cluster admin.
   b) Prepare the authentication credentials to access the IBM production repository. Use the same auth.json file used for CASE download and image mirroring. An example directory path:

      ```
      ${HOME}/.airgap/auth.json
      ```

      Or create an auth.json file that contains credentials to access icr.io and your local private registry. For example:

      ```
      {
      "auths": {
            "cp.icr.io":{"email":"unused","auth":"<base64 encoded id:apikey>"},
            "<private registry hostname>":{"email":"unused","auth":"<base64 encoded
            id:password>"}
        }
      }
      ```

      For more information about the auth.json file, see containers-auth.json - syntax for the registry authentication file.

   c) Install skopeo by running:

      ```
      yum install skopeo
      ```

   d) To confirm the path for the local private registry to copy the hotfix images to, run the following command:

      ```
      oc describe pod <hotfix image pod> | grep -i "image:"
      ```

      <hotfix image pod> can be the pod name for any of the images which will be patched with this hotfix.

For example:

```
oc describe pod wdp-profiling-7855f7fd8f-lsvsj | grep Image:
…
Image:  cp.icr.io/cp/cpd/wdp-
profiling@sha256:03c88c69b986f24d39e4556731c0d171169d2bd91b0fb22f6367fd51c9020e64
```

e) To get the local private registry source details, run the following commands:

```
oc get imageContentSourcePolicy
oc describe imageContentSourcePolicy [cloud-pak-for-data-mirror]
```

The local private registry mirror repository and path details should be in the output of the describe command:

```
- mirrors:
  - ${PRIVATE_REGISTRY_LOCATION}/cp/
  source: cp.icr.io/cp/cpd
```

For more information about mirroring of images, see Configuring your cluster to pull Cloud Pak for - Data images.

f) Use the *skopeo* command to copy the patch images from the IBM production registry to the local private registry. Using the appropriate auth.json file, copy the patch images from the IBM production registry to the Openshift cluster registry:

NOTE: When copy/pasting each "skopeo" command below, it is recommended to copy the command into a text editor to ensure there are no additional newline characters after the '\'. Remove any additional newline characters. Then copy/paste the command from the text editor to the command line. If these steps are not done, the command may fail.

Analytics Engine

```
skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/spark-hb-control-
plane@sha256:ef46de7224c6c37b2eadf2bfbbbaeef5be7b2e7e7c05d55c4f8b0eba1fb4e9e4 \
    <local private registry>/cp/cpd/spark-hb-control-
plane@sha256:ef46de7224c6c37b2eadf2bfbbbaeef5be7b2e7e7c05d55c4f8b0eba1fb4e9e4

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/spark-hb-
jkg@sha256:4b4eefb10d2a45ed1acab708a28f2c9d3619432f4417cfbfdc056f2ca3c085f7 \
    <local private registry>/cp/cpd/spark-hb-
jkg@sha256:4b4eefb10d2a45ed1acab708a28f2c9d3619432f4417cfbfdc056f2ca3c085f7
```

CCS

```
skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
```

```
    docker://cp.icr.io/cp/cpd/asset-files-
api@sha256:a1525c29bebed6e9a982f3a06b3190654df7cf6028438f58c96d0c8f69e674c1 \
    <local private registry>/cp/cpd/asset-files-
api@sha256:a1525c29bebed6e9a982f3a06b3190654df7cf6028438f58c96d0c8f69e674c1

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/portal-
catalog@sha256:4646053d470dbb7edc90069f1d7e0b1d26da76edd7325d22af50535a61e42fed \
    <local private registry>/cp/cpd/portal-
catalog@sha256:4646053d470dbb7edc90069f1d7e0b1d26da76edd7325d22af50535a61e42fed

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/portal-
projects@sha256:d3722fb9a7e4a97f6f6de7d2b92837475e62cd064aa6d7590342e05620b16a6a \
    <local private registry>/cp/cpd/portal-
projects@sha256:d3722fb9a7e4a97f6f6de7d2b92837475e62cd064aa6d7590342e05620b16a6a

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wdp-connect-
connection@sha256:3d5fadf3ec1645dae10136226d37542a9d087782663344a1f78e0ee3af7b5aa6
\
    <local private registry>/cp/cpd/wdp-connect-
connection@sha256:3d5fadf3ec1645dae10136226d37542a9d087782663344a1f78e0ee3af7b5aa6

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wdp-connect-
connector@sha256:1b7ecb102c8461b1b9b0df9a377695b71164b00ab72391ddf4b063bd45da670c \
    <local private registry>/cp/cpd/wdp-connect-
connector@sha256:1b7ecb102c8461b1b9b0df9a377695b71164b00ab72391ddf4b063bd45da670c
```

WKC

```
skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/ibm-cpd-wkc-
operator@sha256:9bb509867f1c5a9948796ea50b7087d23aa733eb5e88ed93a15425d98221c5d0 \
    <local private registry>/cp/cpd/ibm-cpd-wkc-
operator@sha256:9bb509867f1c5a9948796ea50b7087d23aa733eb5e88ed93a15425d98221c5d0

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wkc-metadata-imports-
ui@sha256:53c8e2a0def2aa48c11bc702fc1ddd0dda089585f65597d0e64ec6cfba3a103e \
    <local private registry>/cp/cpd/wkc-metadata-imports-
ui@sha256:53c8e2a0def2aa48c11bc702fc1ddd0dda089585f65597d0e64ec6cfba3a103e

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wdp-profiling-
ui@sha256:85e36bf943bc4ccd7cb2af0c524d5430ceabc90f2d5a5fb7e1696dbc251e5cc0 \
    <local private registry>/cp/cpd/wdp-profiling-
ui@sha256:85e36bf943bc4ccd7cb2af0c524d5430ceabc90f2d5a5fb7e1696dbc251e5cc0
```

```
skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wkc-mde-service-
manager@sha256:713684c36db568e0c9d5a3be40010b0f732fa73ede7177d9613bc040c53d6ab9 \
    <local private registry>/cp/cpd/wkc-mde-service-
manager@sha256:713684c36db568e0c9d5a3be40010b0f732fa73ede7177d9613bc040c53d6ab9

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wdp-
profiling@sha256:1cedac257eb092c1a906046bd4ff8939856c1ef5fd624d1772469f795ee7236f \
    <local private registry>/cp/cpd/wdp-
profiling@sha256:1cedac257eb092c1a906046bd4ff8939856c1ef5fd624d1772469f795ee7236f

skopeo copy --all --authfile "<folder path>/auth.json" \
    --dest-tls-verify=false --src-tls-verify=false \
    docker://cp.icr.io/cp/cpd/wdp-
profiling@sha256:ecc845503e45b4f8a0c83dce077d41c9a816cb9116d3aa411b000ec0eb916620 \
    <local private registry>/cp/cpd/wdp-
profiling@sha256:ecc845503e45b4f8a0c83dce077d41c9a816cb9116d3aa411b000ec0eb916620
```

2) To install the patch using the online IBM entitled registry, or to apply the hotfix using the images downloaded to the local private registry from Step 1, proceed with the following commands. Note that `${PROJECT_CPD_INSTANCE}` refers to the project name where WKC is installed.

   NOTE: When copy/pasting each "oc patch" command below, please ensure that it is contained on a single line without line breaks. Also, if any spaces are introduced by the copy/paste, they must be removed. If these steps are not done, the command may fail.

   a) Save a copy of the CRs currently deployed on the cluster

   ```
   oc get wkc wkc-cr -o yaml -n ${PROJECT_CPD_INSTANCE} > wkc_cr_backup_<date>.yaml

   oc get ccs ccs-cr -o yaml -n ${PROJECT_CPD_INSTANCE} > ccs_cr_backup_<date>.yaml

   oc get AnalyticsEngine analyticsengine-sample -o yaml -n ${PROJECT_CPD_INSTANCE} >
   iae_cr_backup_<date>.yaml
   ```

   where `<date>` is the date you are applying the patch.

   b) Run the following command to patch the WKC operator csv:

   ```
   oc patch csv -n ${OPERATOR_NAMESPACE} ibm-cpd-wkc.v1.6.5 --type='json' -p='[{"op":
   "replace", "path":
   "/spec/install/spec/deployments/0/spec/template/spec/containers/0/image",
   "value":"icr.io/cpopen/ibm-cpd-wkc-
   operator@sha256:9bb509867f1c5a9948796ea50b7087d23aa733eb5e88ed93a15425d98221c5d0"}]
   '
   ```

   In the above command, replace `${OPERATOR_NAMESPACE}` with the project namespace where the WKC operator has been installed. (*ibm-common-services* or *cpd-operators* for example).

Before applying the patch command, note the existing sha256 digest value for the `ibm-cpd-wkc-operator` image, this will be useful in case you would need to revert the patch.

c) Run the following command to apply the patch to the Analytics Engine:

```
oc patch AnalyticsEngine analyticsengine-sample -n ${PROJECT_CPD_INSTANCE} --
type=merge -p '{"spec":{"image_digests":{"spark-hb-control-
plane":"sha256:ef46de7224c6c37b2eadf2bfbbbaeef5be7b2e7e7c05d55c4f8b0eba1fb4e9e4","s
park-hb-jkg-
v33":"sha256:4b4eefb10d2a45ed1acab708a28f2c9d3619432f4417cfbfdc056f2ca3c085f7"}}}'
```

d) Run the following command to apply the patch to the Common Core Services custom resource (ccs-cr):

```
oc patch ccs ccs-cr -n ${PROJECT_CPD_INSTANCE} --type=merge -p
'{"spec":{"wdp_connect_connection_image":{"name":"wdp-connect-
connection@sha256","tag":"3d5fadf3ec1645dae10136226d37542a9d087782663344a1f78e0ee3a
f7b5aa6","tag_metadata":"6.3.325"},"wdp_connect_connector_image":{"name":"wdp-
connect-
connector@sha256","tag":"1b7ecb102c8461b1b9b0df9a377695b71164b00ab72391ddf4b063bd45
da670c","tag_metadata":"6.3.325"},"asset_files_api_image":{"name":"asset-files-
api@sha256","tag":"a1525c29bebed6e9a982f3a06b3190654df7cf6028438f58c96d0c8f69e674c1
","tag_metadata":"4.6.5.4.155-amd64"},"portal_projects_image":{"name":"portal-
projects@sha256","tag":"d3722fb9a7e4a97f6f6de7d2b92837475e62cd064aa6d7590342e05620b
16a6a","tag_metadata":"4.6.5.4.2504-amd64"},"portal_catalog_image":{"name":"portal-
catalog@sha256","tag":"4646053d470dbb7edc90069f1d7e0b1d26da76edd7325d22af50535a61e4
2fed","tag_metadata":"0.4.2817-amd64"}}}'
```

e) Run the following command to apply the patch to the WKC custom resource (wkc-cr):

```
oc patch wkc wkc-cr -n ${PROJECT_CPD_INSTANCE} --type=merge -p
'{"spec":{"wkc_metadata_imports_ui_image":{"name":"wkc-metadata-imports-
ui@sha256","tag":"53c8e2a0def2aa48c11bc702fc1ddd0dda089585f65597d0e64ec6cfba3a103e"
,"tag_metadata":"4.6.5511"},"wdp_profiling_ui_image":{"name":"wdp-profiling-
ui@sha256","tag":"85e36bf943bc4ccd7cb2af0c524d5430ceabc90f2d5a5fb7e1696dbc251e5cc0"
,"tag_metadata":"4.6.1203-amd64"},"wkc_mde_service_manager_image":{"name":"wkc-mde-
service-
manager@sha256","tag":"713684c36db568e0c9d5a3be40010b0f732fa73ede7177d9613bc040c53d
6ab9","tag_metadata":"1.2.55"},"wdp_profiling_image":{"name":"wdp-
profiling@sha256","tag":"ecc845503e45b4f8a0c83dce077d41c9a816cb9116d3aa411b000ec0eb
916620","tag_metadata":"4.6.5031-amd64"}}}'
```

f) Wait for the AnalyticsEngine operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get AnalyticsEngine analyticsengine-sample -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `spark-hb-jkg` and `spark-hb-control-plane` pods in `${PROJECT_CPD_INSTANCE}` should be up and running with the updated images.

g) Wait for the Common Core Services operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get ccs ccs-cr -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `wdp-connect-*`, `asset-files-api`, `portal-projects` and `portal-catalog` pods in `${PROJECT_CPD_INSTANCE}` should be up and running with the updated images, and the CCS custom resource should show a status of "Completed".

h) Wait for the WKC operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get wkc wkc-cr -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `wkc-metadata-imports-ui`, `wdp-profiling`, `wkc-mde-service-manager` and `wdp-profiling-ui` pods should be up and running with the updated images, and the WKC custom resource should show a status of "Completed".

3) Unzip the contents of the `MDI_Job_Reconciliation_Script.zip` file into a folder. This may be needed in the event that the state of an MDI job does not match the state in the MDI job log. See the included `readme.txt` file for details on how to run the script.

To revert the hotfix changes

Make sure to revert the image overrides before you install or upgrade to a newer refresh or a major release of IBM® Cloud Pak for Data.

To revert the image overrides, proceed with the following steps. Note that `${PROJECT_CPD_INSTANCE}` refers to the project name where WKC is installed.

NOTE: When copy/pasting each "oc patch" command below, please ensure that it is contained on a single line without line breaks. Also, if any spaces are introduced by the copy/paste, they must be removed. If these steps are not done, the command may fail.

a) Using the sha256 digest value for the `ibm-cpd-wkc-operator` image saved prior to applying this hotfix, run the following patch command to revert the WKC operator image.

```
oc patch csv -n ${OPERATOR_NAMESPACE} ibm-cpd-wkc.v1.6.5 --type='json' -p='[{"op":
"replace", "path":
"/spec/install/spec/deployments/0/spec/template/spec/containers/0/image",
"value":"icr.io/cpopen/ibm-cpd-wkc-operator@sha256:<original sha256 value>"}]'
```

b) Run the following command to remove image digest updates from the Analytics Engine custom resource.

```
oc patch AnalyticsEngine analyticsengine-sample --namespace ${PROJECT_CPD_INSTANCE} --
type=json --patch '[{"op":"remove","path":"/spec/image_digests"}]'
```

c) Run the following command to edit the Common Core Services custom resource:

```
oc edit ccs ccs-cr -n ${PROJECT_CPD_INSTANCE}
```

d) Remove the following lines within the CCS custom resource and replace with the copy that you saved during the installation. Save the change.

```
asset_files_api_image:
  name: asset-files-api@sha256
  tag: a1525c29bebed6e9a982f3a06b3190654df7cf6028438f58c96d0c8f69e674c1
  tag_metadata: 4.6.5.4.155-amd64
portal_catalog_image:
  name: portal-catalog@sha256
  tag: 4646053d470dbb7edc90069f1d7e0b1d26da76edd7325d22af50535a61e42fed
  tag_metadata: 0.4.2817-amd64
portal_projects_image:
  name: portal-projects@sha256
  tag: d3722fb9a7e4a97f6f6de7d2b92837475e62cd064aa6d7590342e05620b16a6a
  tag_metadata: 4.6.5.4.2504-amd64
wdp_connect_connection_image:
  name: wdp-connect-connection@sha256
  tag: 3d5fadf3ec1645dae10136226d37542a9d087782663344a1f78e0ee3af7b5aa6
  tag_metadata: 6.3.325
wdp_connect_connector_image:
  name: wdp-connect-connector@sha256
  tag: 1b7ecb102c8461b1b9b0df9a377695b71164b00ab72391ddf4b063bd45da670c
  tag_metadata: 6.3.325
```

e) Run the following command to edit the WKC custom resource:

```
oc edit wkc wkc-cr -n ${PROJECT_CPD_INSTANCE}
```

f) Remove the following lines for the patched images within the WKC custom resource and replace with the copy that you saved during the installation. Save the change.

```
wkc_metadata_imports_ui_image:
  name: wkc-metadata-imports-ui@sha256
  tag: 53c8e2a0def2aa48c11bc702fc1ddd0dda089585f65597d0e64ec6cfba3a103e
  tag_metadata: 4.6.5511
wdp_profiling_ui_image:
  name: wdp-profiling-ui@sha256
  tag: 85e36bf943bc4ccd7cb2af0c524d5430ceabc90f2d5a5fb7e1696dbc251e5cc0
  tag_metadata: 4.6.1203-amd64
wkc_mde_service_manager_image:
  name: wkc-mde-service-manager@sha256
  tag: 713684c36db568e0c9d5a3be40010b0f732fa73ede7177d9613bc040c53d6ab9
  tag_metadata: 1.2.55
wdp_profiling_image:
  name: wdp-profiling@sha256
  tag: ecc845503e45b4f8a0c83dce077d41c9a816cb9116d3aa411b000ec0eb916620
  tag_metadata: 4.6.5031-amd64
```

g) Wait for the AnalyticsEngine operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get AnalyticsEngine analyticsengine-sample -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `spark-hb-jkg` and `spark-hb-control-plane` pods in `${PROJECT_CPD_INSTANCE}` should be up and running with the original images.

h) Wait for the Common Core Services operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get ccs ccs-cr -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `wdp-connect-*`, `asset-files-api`, `portal-projects` and `portal-catalog` pods in `${PROJECT_CPD_INSTANCE}` should be up and running with the original images, and the CCS custom resource should show a status of "Completed".

i) Wait for the WKC operator reconciliation to complete. Run the following command to monitor the reconciliation status:

```
oc get wkc wkc-cr -n ${PROJECT_CPD_INSTANCE}
```

After a period of time, the `wkc-metadata-imports-ui`, `wdp-profiling`, `wkc-mde-service-manager` and `wdp-profiling-ui` pods in `${PROJECT_CPD_INSTANCE}` should be up and running with the original images, and the WKC custom resource should show a status of "Completed".