Setting up your own domain for workloads in Cloudera on cloud (Preview)

Date published: 2026-04-01
Date modified: 2026-04-01

Legal Notice

This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

Contents

Overview

Cloudera on cloud enables users to launch data workloads, from Data Lake, Cloudera Data Hubs to specialized data services like Cloudera Data Warehouse. A domain name is dynamically assigned to the launched workloads with `cloudera.site` suffix that is owned by Cloudera. Now, users can configure their workloads with a domain of their choice that they own. The following configuration guides will guide you through the steps to set up your own domain for workloads in Cloudera on cloud.

Limitations

- Setting up your own domain is currently only available as a Technical Preview. You need to have the following entitlement to use this feature:
    - IAM_WORKLOAD_SSO_FROM_CUSTOM_DOMAIN

  For more information about how to obtain the list of entitlements, contact Cloudera Customer Support.
- The feature is only supported for AWS and Azure deployments.
- TLS with user signed certificates does not work for Hue since TLS termination is being done at the Load Balancer itself.

Setting up the domain for an existing cluster

Data Lake and Cloudera Data Hub clusters

The following steps will show how to set up the domain for an existing Data Lake or Cloudera Data Hub cluster. In the example, the domain is changed from `data-lake.env.account-subdomain.cloudera.site` to `dl.cdp.acme.corp`.

Prerequisites

- Ensure that you have a Data Lake and Cloudera Data Hub cluster in healthy state.
- Ensure that you prepared your CA certificate and private key.
- Ensure that you have the SSH keys that are required to access the gateway nodes and the root credentials.
- Ensure that you have access to an authoritative DNS server for your domain.

- Ensure that `*.cdp.acme.corp` is added as a trusted SAML Assertion Consumer Service (ACS) subdomain of your service provider for the given account.

Steps

1. Navigate to your environment in Cloudera Management Console, and select the **Data Lake**.
2. On the **Data Lake** details page, select Nodes, and find the **Gateway** node.
3. Access the Gateway node as a `root` user using `ssh`.
4. Copy the certificate and private key using `scp` to the gateway nodes as `/etc/certs/custom-domain.pem` and `/etc/certs/custom-domain-key.pem`.
5. Edit the `/etc/nginx/sites-enabled/ssl-user-facing.conf` by duplicating the `server` block already present in the file.
6. Add the following line in the duplicated server block.

```
None
server_name dl.cdp.acme.corp;
```

7. Modify the certificate paths in the duplicated block to point at the new certificates as shown in the following following:

```
None
ssl_certificate      /etc/certs/custom-domain.pem;
ssl_certificate_key  /etc/certs/custom-domain-key.pem;
```

8. Restart nginx with the following command:

```
None
systemctl restart nginx
```

9. Repeat Step 2, 3, 4, and 5 for every Gateway node of the cluster.
10. Add a CNAME record in the managed DNS server for all the `<subdomain>.cdp.acme.corp` FQDNS, and point them at their corresponding `.cloudera.site` domain.
11. Make a copy of the following files:
    - `/var/lib/knox/gateway/conf/topologies/cdp-proxy.xml` as `/var/lib/knox/gateway/conf/topologies/byod-proxy.xml`
    - `/var/lib/knox/gateway/conf/topologies/knoxsso.xml` as `/var/lib/knox/gateway/conf/topologies/byodsso.xml`

12. Ensure that these files can be read by the Knox user and group.
    It is recommended that the ownership be changed using `chown` to `knox:knox`.
13. Edit `/var/lib/knox/gateway/conf/topologies/byodsso.xml` with the following steps:
    - Replace the `/tmp/sp-metadata.xml` text with `/tmp/sp-metadata-byod.xml`.
      It is an expected behavior that this file does not exist until the first user request reaches the Knox service. At this step, you are only configuring the path where a temporary working file gets written.
    - Replace all occurrences of `original.cloudera.site/cluster-name/knoxsso` with `dl.cdp.acme.corp/cluster-name/byodsso`.
      The `original.cloudera.site` is a placeholder for the original `.cloudera.site` domain, and `cluster-name` is a placeholder for the cluster's name.
    - Update `knoxsso.redirect.whitelist.regex` to set up allowing redirects to the new custom domain.
14. Edit `/var/lib/knox/gateway/conf/topologies/byod-proxy.xml` with the following steps:
    - Replace `topology.name` to `byod-sso`.
    - Change `sso.authentication.provider.url` to replace `/cluster-name/knoxsso` with `/cluster-name/byodsso`.
      The `cluster-name` is a placeholder for the cluster's name.
15. Validate the setup by accessing the workload cluster through workload service UI, such as Cloudera Manager or Hue.

Important!
Any operational change in the Data Lake or Cloudera Data Hub workload cluster, such as repair, scale, or upgrade) will revert these changes, and the steps to configure your own domain must be completed again after the operation is successful.

Cloudera Data Warehouse

The following steps will show how to set up the domain for an existing Cloudera Data Warehouse cluster. In the example, the domain is changed from `service-workspace.env.account-subdomain.cloudera.site` to `cdw.cdp.acme.corp`.

Prerequisites

- Ensure that you have a Cloudera Data Warehouse cluster in healthy state.
- Ensure that you prepared your CA certificate and private key.

- Ensure that you have the Kubeconfig and credentials required to access `kubernetes-api-server`.
- Ensure that you have access to an authoritative DNS server for your domain.
- Ensure that `*.cdp.acme.corp` is added as a trusted SAML Assertion Consumer Service (ACS) subdomain of your service provider for the given account.

Steps

1. Use the following command to find the ingress of your cluster.
   The namespace, where the ingress is deployed, might change as per launch configuration.

```
None
kubectl get ingress -A | grep -w hue-ingress
```

2. Create a TLS type secret in the ingress namespace, and add the private-key and certificate to the secret.

```
None
apiVersion: v1
kind: Secret
metadata:
 name: <ingress>-tls
 namespace: <ingress-namespace>
data:
 tls.crt: <base64 encoded cert>
 tls.key: <base64 encoded key>
type: kubernetes.io/tls
```

3. Edit the ingress definition with additional values to bind against the new `cdw.cdp.acme.corp domain` while duplicating the backend information.

```
None
kubectl -n <namespace> edit ingress/hue-ingress
```

```
None
spec:
  tls:
```

```
    - hosts:
      - *.cdw.cdp.acme.corp
      secretName: <ingress>-tls
  rules:
    - host: hue.cdw.cdp.acme.corp
      http:
        paths:
        - backend:
            service:
              name: coordinator
              port:
                number: 28000
          path: /
          pathType: ImplementationSpecific
```

4. Edit the SAML configuration in the relevant ConfigMap to redirect the browser back to the new custom domain after workload login.

```
None
kubectl -n <namespace> edit configmap/hue-saml-conf
```

```
None
apiVersion: v1
data:
 saml.ini: |
   [desktop]

redirect_whitelist="^\/.*$,^http:\/\/hue.cdw.cdp.acme.corp\/.*$"
   [[auth]]
   backend=libsaml.backend.SAML2Backend
   [libsaml]
   base_url=https://hue.cdw.cdp.acme.corp
   xmlsec_binary=/usr/bin/xmlsec1
   metadata_file=/etc/hue/conf/saml/idpmetadata.xml
```

```
   key_file=/var/run/hue/samlcert/server.key
   cert_file=/var/run/hue/samlcert/server.crt
   username_source=nameid
   entity_id="https://hue.cdw.cdp.acme.corp/saml2/acs/"
   authn_requests_signed=true
   logout_enabled=false
   attribute_map_dir=/etc/hue/conf/saml_attributes

user_attribute_mapping='{"uid":"username","first_name":"first_nam
e","last_name":"last_name"}'

name_id_format='urn:oasis:names:tc:SAML:2.0:nameid-format:persist
ent'
   want_response_signed=true
   want_assertions_signed=true
   required_groups_attribute="groups"

logout_url="https://console.dps.mow-dev.cloudera.com/consoleauth/
logout"
kind: ConfigMap
metadata:
 annotations:
   ...
 labels:
   app.kubernetes.io/managed-by: Helm
 name: hue-saml-conf
 namespace: <ingress-namespace>
```

5. Perform Step 1, 2, 3 and 4 for all services that need to be available over the custom domain.
6. Restart all relevant pods using the following command:

```
None
kubectl -n <namespace> rollout restart all
```

7. Wait for the pods to return to a healthy state.
8. Add a CNAME record in the managed DNS server for all the `<subdomain>.cdw.cdp.acme.corp` FQDNS and point them at their corresponding `.cloudera.site` domain.
9. Validate the setup by accessing the workload cluster through workload service UI, such as Cloudera Manager or Hue.

Important!
Any operational change in the Cloudera Data Warehouse cluster, such as repair, scale, or upgrade) will revert these changes, and the steps to configure your own domain must be completed again after the operation is successful.