Dataiku DSS Setup and Troubleshooting

Ahmed Khan

Table of Contents

Task 1: Install a DSS instance (Design Node) - please install with port 11200

Followed the steps outlined in this link: [Installing a new DSS instance — Dataiku DSS 13 documentation]
Azure environment information was already provided.

1. Verified CentOS version
   `cat /etc/centos-release`

   ```
   [assessment_admin@candidate-ahmed-khan-assessment-vm ~]$ cat /etc/centos-release
   CentOS Linux release 7.9.2009 (Core)
   ```

   `hostnamectl`

   ```
   [assessment_admin@candidate-ahmed-khan-assessment-vm ~]$ hostnamectl
      Static hostname: candidate-ahmed-khan-assessment-vm
            Icon name: computer-vm
              Chassis: vm
           Machine ID: 97da09219a2d42489c8b8f748e6d2fb7
              Boot ID: 5496856030c7407095911f0c552c2ad5
       Virtualization: microsoft
     Operating System: CentOS Linux 7 (Core)
          CPE OS Name: cpe:/o:centos:centos:7
               Kernel: Linux 3.10.0-862.11.6.el7.x86_64
         Architecture: x86-64
   [assessment_admin@candidate-ahmed-khan-assessment-vm ~]$
   ```

2. Created a non-privileged service account 'sadataiku'
   ```
   sudo useradd -r -m -s /sbin/nologin sadataiku
   sudo usermod -s /bin/bash sadataiku
   ```

3. Allow user 'sadataiku' to sudo
   ```
   sudo usermod -aG wheel sadataiku
   ```

4. Switched user to sadataiku
5. Created a directory called 'softwares' in sadataiku's home
6. Downloaded the installation file
   `wget https://downloads.dataiku.com/public/studio/12.2.3/dataiku-dss-12.2.3.tar.gz`

7. Uncompressed the file
   ```
   tar xzf dataiku-dss-12.2.3.tar.gz
   ```

8. Created a directory called 'data in sadataiku's home

9. Ran the installer
   ```
   dataiku-dss-12.2.3/installer.sh -d /home/sadataiku/data -p 11200 -l
   dss_license.json
   ```

Issue no. 1: Installation failed due to SELinux being enforced

Following error happened due to SELinux being enforced:

```
DSS cannot run unless you edit the policies to allow nginx to
serve its files.
[!] ********************************************************
[!] Warning: you have SELinux installed and enforcing.
[!] DSS cannot run unless you edit the policies to allow nginx to
serve its files.
[!] Press Enter to continue, Ctrl+C to abort
```

Resolution:
Set SELinux to permissive
sudo setenforce 0

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ getenforce
Permissive
```

10. Ran the installer again.

Issue no. 2: Package dependencies not found

```
Following error happened due to packages not found:
[*] Could not find suitable version of Java
[+] Checking required dependencies
+ Detected OS distribution : centos 7
+ Checking required packages...
*** Error: package git not found
*** Error: package nginx not found
*** Error: package java-1.8.0-openjdk not found
*** Error: package python3 not found
*** Error: package libgfortran not found
```

Resolution:
Ran the script for installing the dependencies:
sudo -i "/home/sadataiku/softwares/dataiku-dss-12.2.3/scripts/install/install-deps.sh"

11. Ran the installer again and it finished successfully

```
* Installation complete (DSS node type: design)
* Next, start DSS using:
*         '/home/sadataiku/data/bin/dss start'
* Dataiku DSS will be accessible on http://<SERVER ADDRESS>:11200
```

12. Started DSS
/home/sadataiku/data/bin/dss start

13. Verified the access to DSS
http://52.170.96.132:11200

Public IP: 52.170.96.132

14. Ran the following script to configure DSS to start automatically at server boot with

```
sudo -i "/home/sadataiku/softwares/dataiku-dss-12.2.3/scripts/install/install-boot.sh" "/home/sadataiku/data" sadataiku
```

15. Enforce SELinux

```
sudo setenforce 1
```

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ getenforce
Enforcing
```

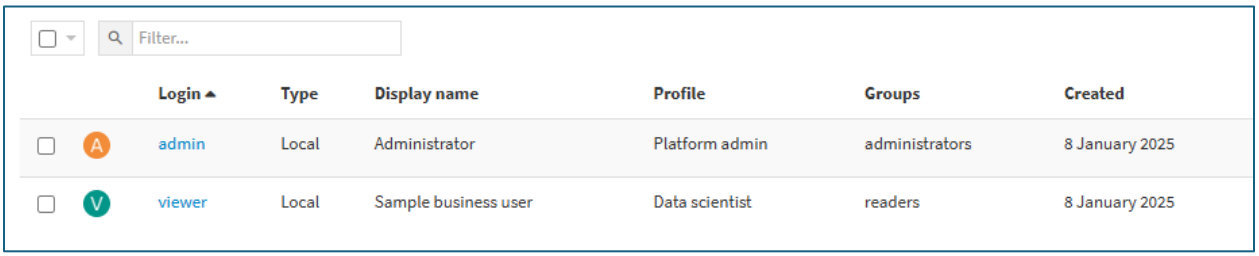
Check point: Create a sample project (Dataiku TShirts) and check that you can build the flow

1. Created a sample project TShirts
2. Downloaded 'web_new_customers' dataset to upload it again.

Issue no. 3: Insufficient permissions in the DSS profile

When tried to upload a dataset file, received the following error:
Your user profile (DATA_SCIENTIST) does not allow you to write project content.

Resolution:
On DSS, Administration --> Security --> admin --> changed profile to 'Platform admin'.

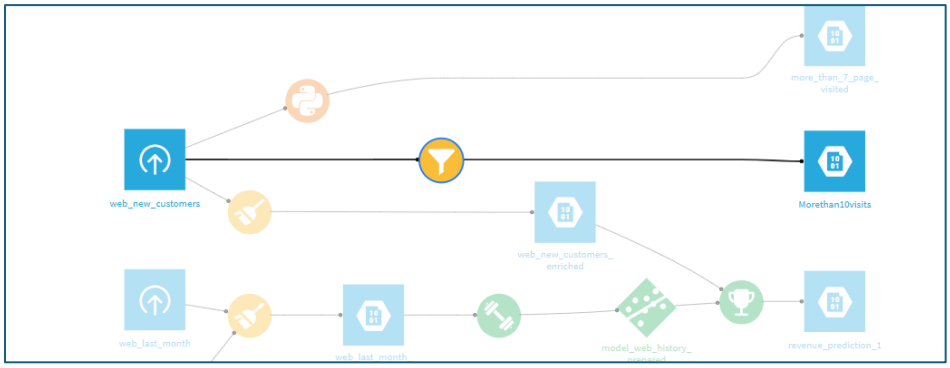


Filter...

Login	Type	Display name	Profile	Groups	Created
admin	Local	Administrator	Platform admin	administrators	8 January 2025
viewer	Local	Sample business user	Data scientist	readers	8 January 2025

Afterwards, successfully uploaded the file as 'web_new_customers_1'.

Created a new flow with output dataset called 'Morethan10visits.



Successfully ran the flow and populated the dataset.

Morethan10visits

Whole data 341 rows

customer_id	ip	pages_visited	campain
string	string	double	string
Text	IP address	Decimal	Boolean
73d1e77401	78.75.149.92	12.0	False
53a89cabb9	83.214.63.83	12.0	True
1075a35cab	117.194.98.86	10.0	False
5f3d9fe92b	189.154.99.46	10.0	False
dd8ca0910b	50.3.243.195	12.0	True
ed9af89fe6	139.81.26.4	12.0	False
b3c96496fe	141.237.85.3	12.0	False
9a45413304	107.237.130.184	12.0	True
b969f272d9	177.69.20.81	14.0	False
5bccb8264b	130.100.15.47	11.0	False
3e3b4522d3	77.191.76.151	10.0	True
1a5c37d045	115.81.190.210	10.0	True
f2a47d7edb	60.207.161.78	14.0	False

Check point: Install R and R-integration.

Followed the steps described in [R integration — Dataiku DSS 13 documentation]

1. Changed directory to data in sadataiku's home
2. Stopped DSS
```
./bin/dss stop
```
3. Ran the installation script
```
./bin/dssadmin install-R-integration
```

Issue no. 4: Installation stopped due to missing packages

```
[+] Saving installation log to /home/sadataiku/data/run/install.log
[+] Checking dependencies
+ Detected OS distribution : centos 7
+ Checking required packages...
*** Error: package R-core-devel not found
*** Error: package libicu-devel not found
*** Error: package libcurl-devel not found
*** Error: package openssl-devel not found
*** Error: package libxml2-devel not found
```

Resolution:

Ran the script for installing the dependencies:

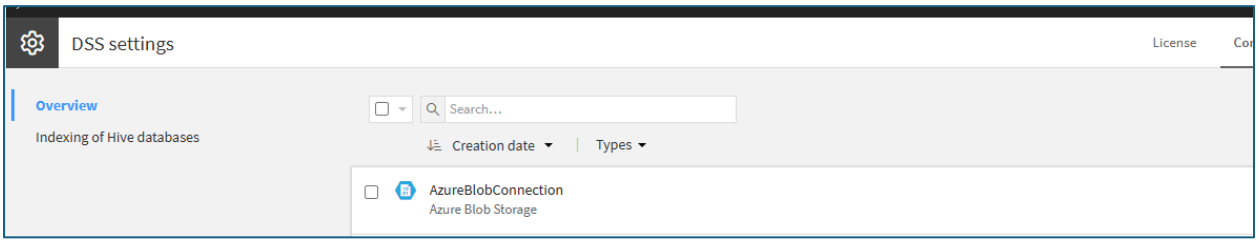
```
sudo -i "/home/sadataiku/softwares/dataiku-dss-12.2.3/scripts/install/install-deps.sh" -without-java -without-python -with-r
```

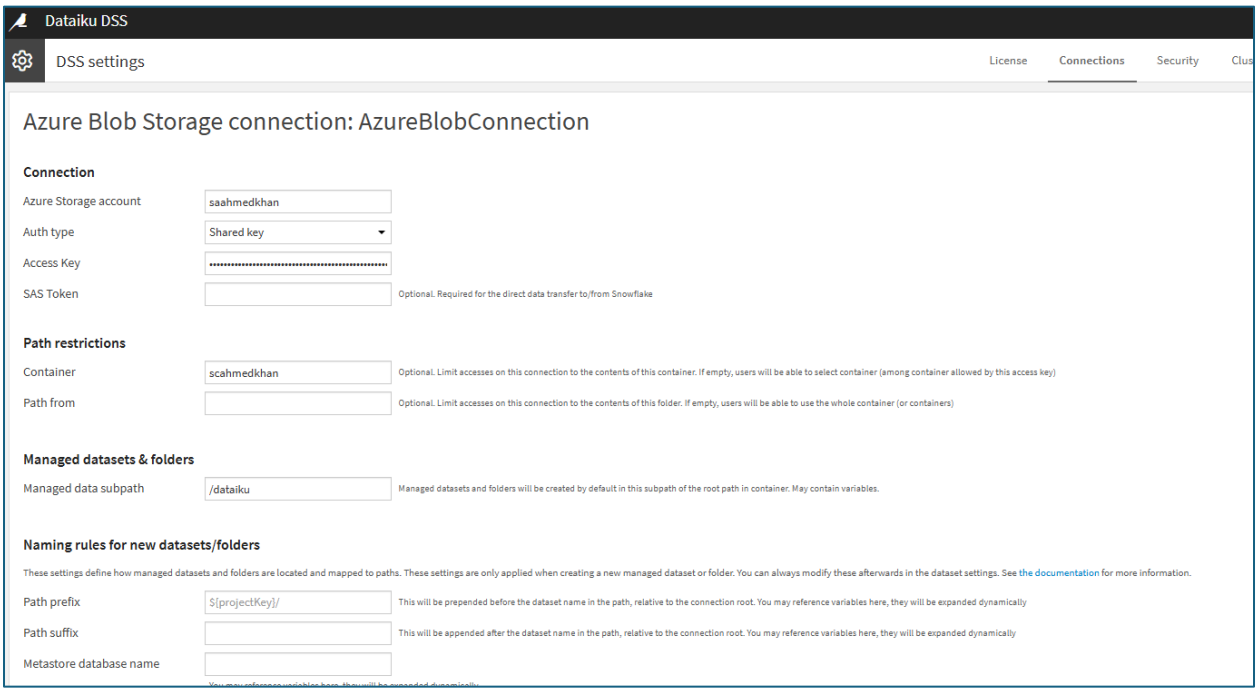
Ran the installation script again and it completed successfully.

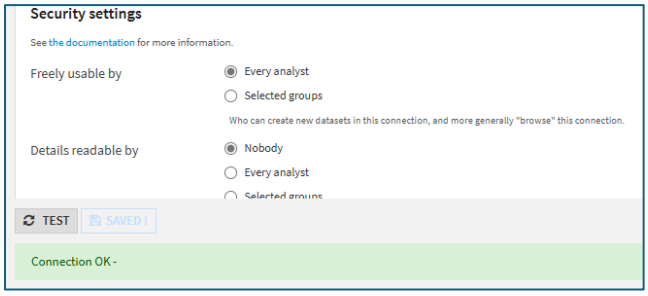
4. Started DSS
```
./bin/dss start
```

Task 2: Define a DSS connection to Azure Blob Storage Connection

1. Using the environment information document, set up the Azure Blob Storage Connection called 'AzureBlobConnection' and tested it successfully.

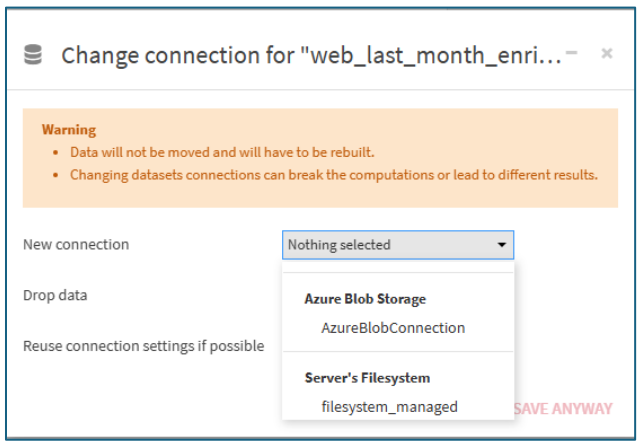


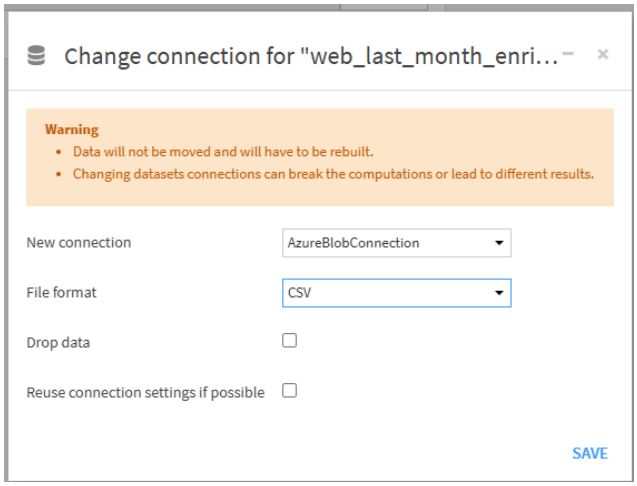




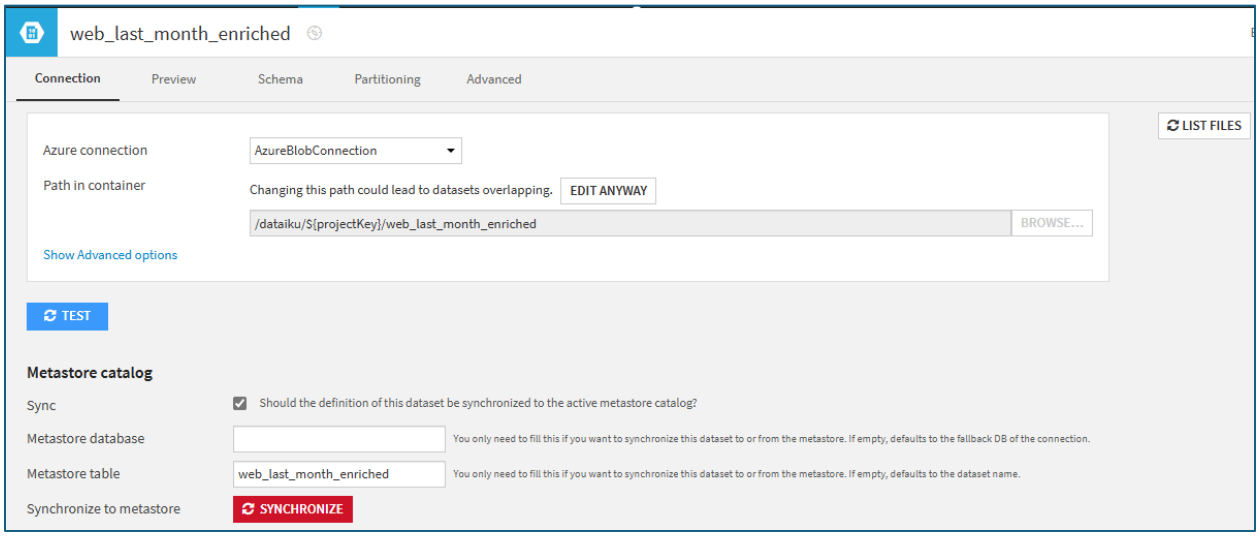
Check point: In the TShirts project, change the connection of the managed datasets to the new Azure Blob Storage connection

1. Changed the connections of all the managed datasets to Azure Blob Storage Connection.





Below is the example of ‘web_last_month_enriched’ dataset’s connection:



Task 3: Connect this DSS instance to an AKS cluster.

It is recommended to use the AKS plugin to allow DSS to create/attach an AKS cluster (Managed AKS clusters)

Followed the instructions described in this link [Using managed AKS clusters — Dataiku DSS 13 documentation]

1. Downloaded and installed Docker
   a. Install necessary prerequisites for Docker:
      ```
      sudo yum install -y yum-utils device-mapper-persistent-data lvm2
      ```
   b. Add the official Docker repository to your system:
      ```
      sudo yum-config-manager --add-repo
      https://download.docker.com/linux/centos/docker-ce.repo
      ```
   c. Update the package list:
      ```
      sudo yum update -y
      ```
   d. Install Docker CE (Community Edition):
      ```
      sudo yum install -y docker-ce docker-ce-cli containerd.io
      ```
   e. Verify the installation:
      ```
      docker --version
      ```
   f. Started Docker
      ```
      sudo systemctl start docker
      ```
   g. Added sadataiku user to docker group:
      ```
      sudo usermod -aG docker $USER
      ```
2. Downloaded and installed Azure CLI
   ```
   sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
   sudo sh -c 'echo -e "[azure-cli]\nname=Azure
   CLI\nbaseurl=https://packages.microsoft.com/yumrepos/azure-
   cli\nenabled=1\ngpgcheck=1\ngpgkey=https://packages.microsoft.com/keys/microsof
   t.asc" > /etc/yum.repos.d/azure-cli.repo'
   sudo yum install -y azure-cli
   ```
3. Downloaded and set up kubectl
   ```
   curl -LO "https://dl.k8s.io/release/$(curl -L -s
   https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
   chmod +x kubectl
   sudo mv kubectl /usr/local/bin/
   ```
4. Logged into AZ using the service principal information
   ```
   az login --service-principal --username <username> --password password --tenant
   tenant-id
   ```

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ az login --service-principal --username [redacted] --pas
sword [redacted] --tenant [redacted]
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "[redacted]",
    "id": "[redacted]",
    "isDefault": true,
    "managedByTenants": [
      {
        "tenantId": "[redacted]
      }
    ],
    "name": "Dataiku FE",
    "state": "Enabled",
    "tenantId": [redacted],
    "user": {
      "name": "[redacted]",
      "type": "servicePrincipal"
    }
  }
]
```

5. Logged into ACR container registry

   `az acr login --name ahmedkhanacr.azurecr.io`

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ az acr login --name ahmedkhanacr.azurecr.io
The login server endpoint suffix '.azurecr.io' is automatically omitted.
Login Succeeded
```

6. Installed AKS plug-in

Store | Installed

Name	By	Origin	Version	Description
AKS clusters	Dataiku	Store	3.0.1	Interact with or create Microsoft Azure Kubernetes Service clusters

7. Get the resource group name using az role command

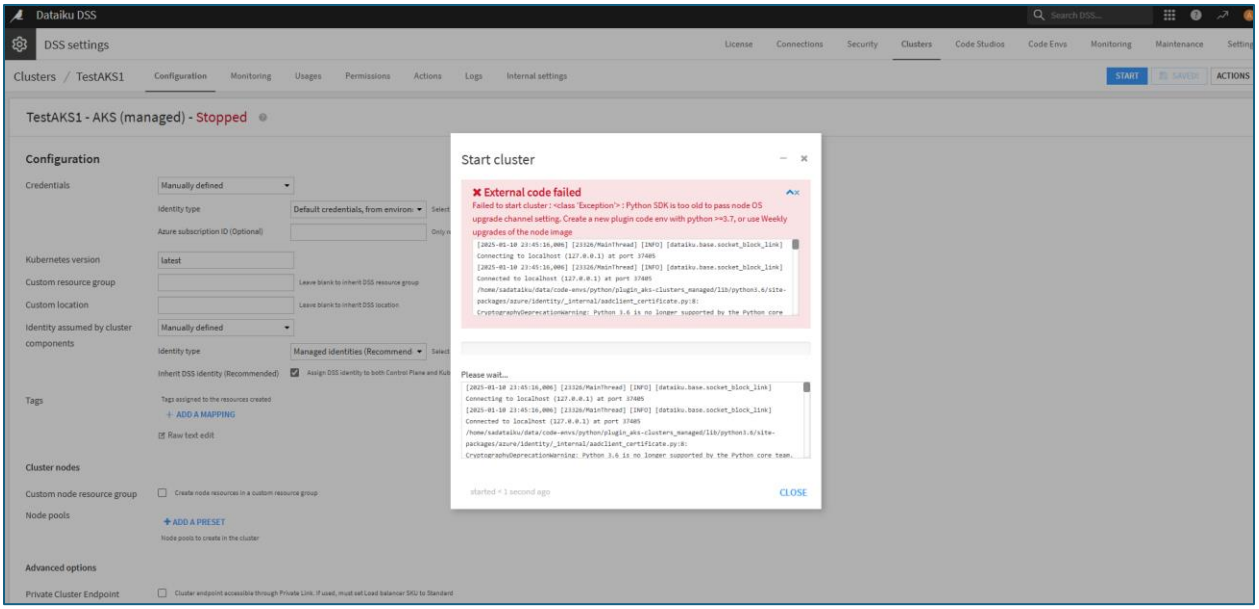
   `az role assignment list --assignee <assignee> -all`

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ az role assignment list --assignee [redacted] --all
[
  {
    "canDelegate": null,
    "condition": null,
    "conditionVersion": null,
    "description": "",
    "id": "/subscriptions/[redacted]/resourceGroups/[redacted]providers/Microsoft.Au
thorization/roleAssignments/[redacted],
    "name": "[redacted]
    "principalId": "[redacted]
    "principalName": [redacted],
    "principalType": "ServicePrincipal",
    "resourceGroup": "[redacted]
```

8. Created an AKS cluster

Issue no. 5: AKS only works with a plug-in code environment of Python 3.7 or greater.

When tried to create the AKS cluster, it failed due to an older python version (3.6) in the plug-in.



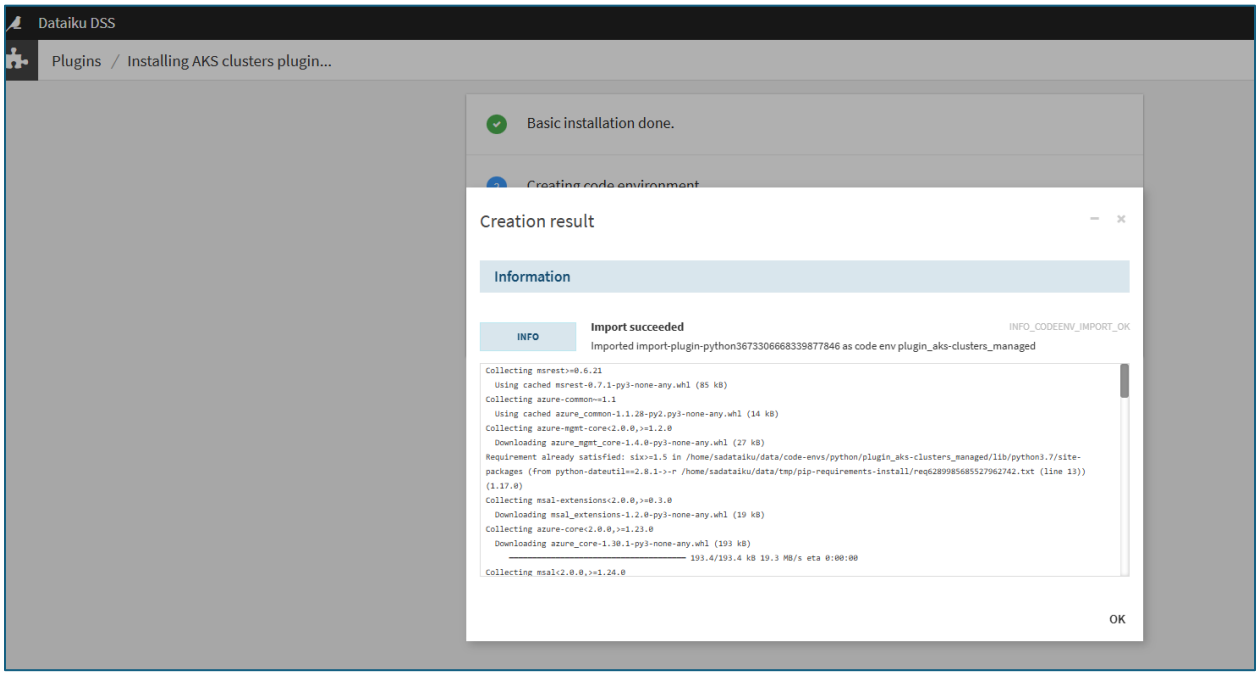
Resolution:

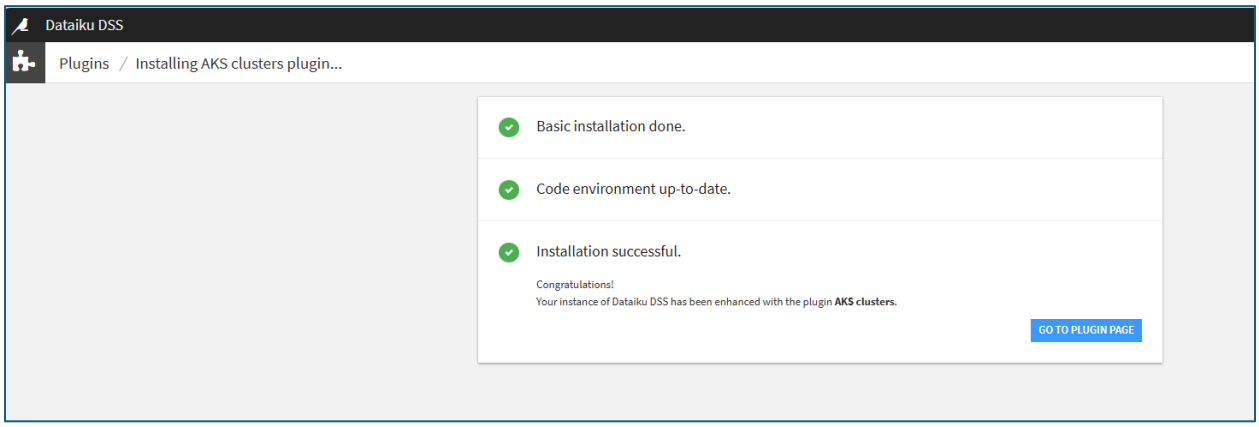
Following steps were taken to resolve this issue:

1. Downloaded and installed Python 3.7 and made it a default version on the host.

```
#Install Python 3.7:
sudo yum update -y
sudo yum groupinstall "Development Tools" -y
sudo yum install gcc openssl-devel bzip2-devel libffi-devel -y

cd /usr/src
sudo curl -O https://www.python.org/ftp/python/3.7.17/Python-3.7.17.tgz

sudo tar xzf Python-3.7.17.tgz
cd Python-3.7.17

sudo ./configure --enable-optimizations
sudo make altinstall

python3.7 --version

sudo alternatives --install /usr/bin/python3 python3
/usr/local/bin/python3.7 1
sudo alternatives --config python3
```

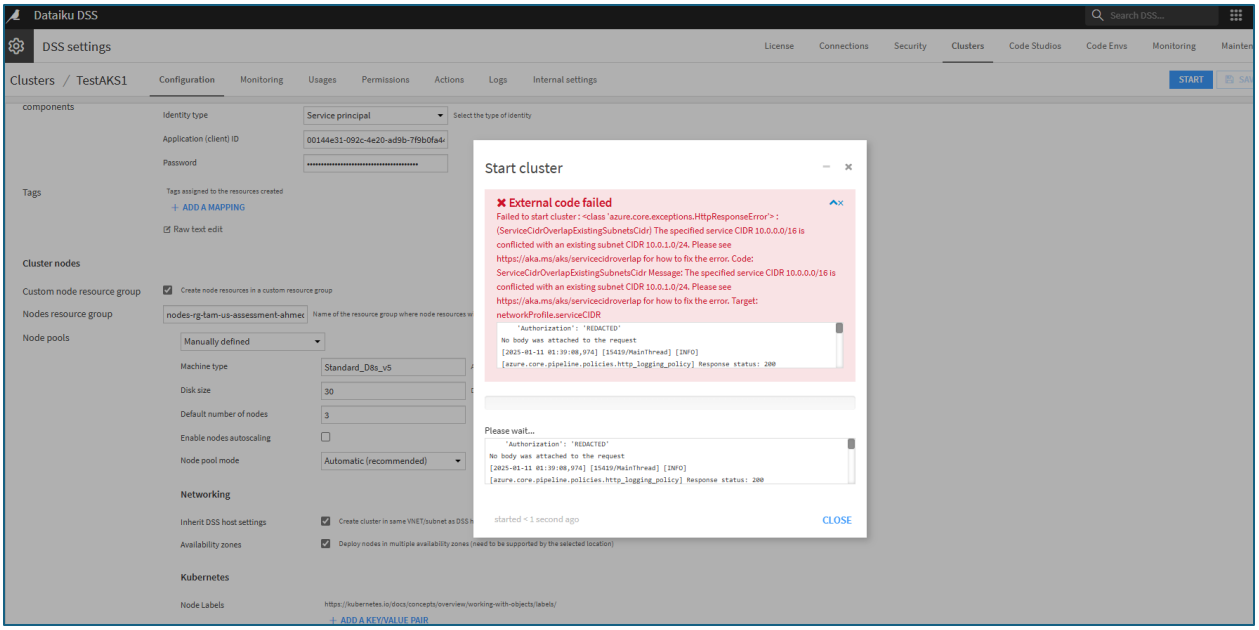
2. Installed AKS plug-in with python 3.7.





Issue no. 6: Service CIDR overlapping existing subnet CIDR

When tried to create the AKS cluster, it complained about the conflict of the service CIDR with the subnet CIDR.



Resolution:

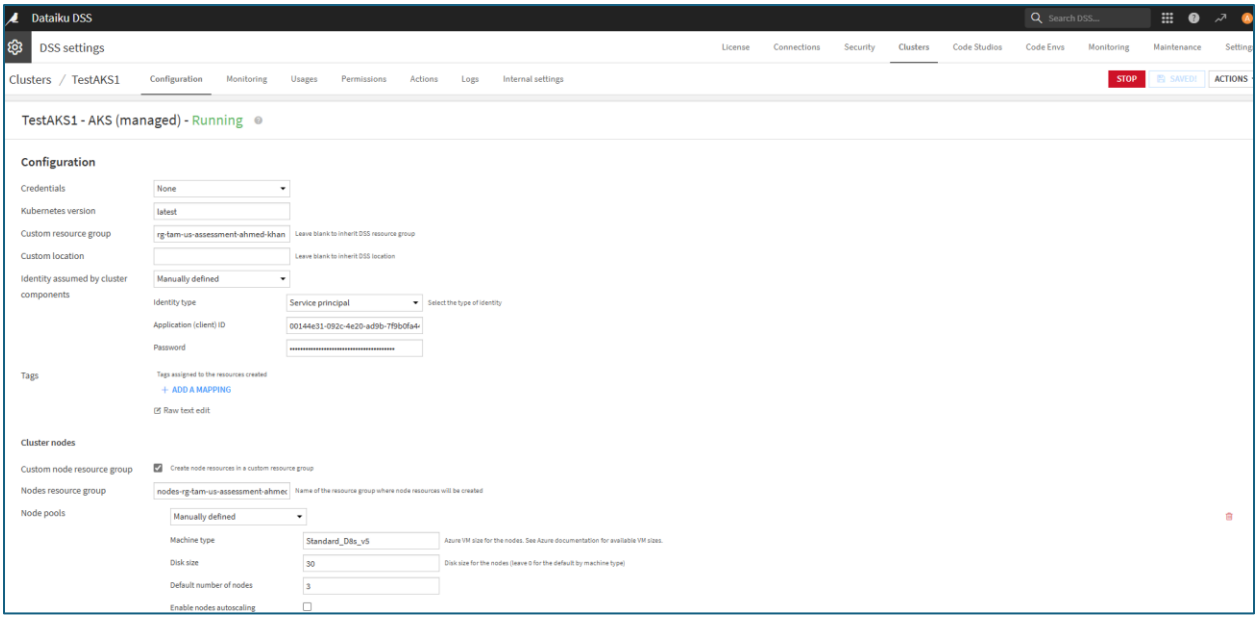
Based on this [Troubleshoot the ServiceCidrOverlapExistingSubnetsCidr error code - Azure | Microsoft Learn](), created a new vnet (vnet2-tam-us-assessment-ahmed-khan) and a subnet (subnet3-tam-us-assessment-ahmed-khan) for the AKS cluster.

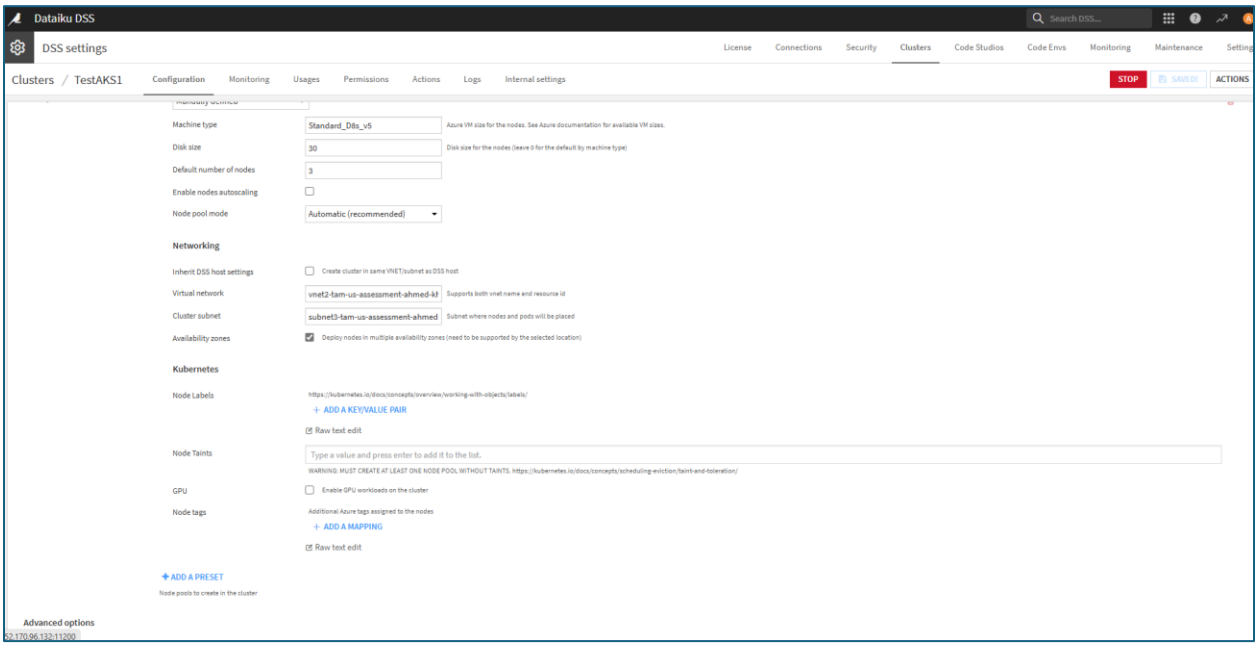
```
az network vnet create \
  --resource-group rg-tam-us-assessment-ahmed-khan \
  --name vnet2-tam-us-assessment-ahmed-khan \
  --address-prefix 10.1.0.0/16 \
  --location eastus

az network vnet subnet create \
  --resource-group rg-tam-us-assessment-ahmed-khan \
  --vnet-name vnet2-tam-us-assessment-ahmed-khan \
  --name subnet3-tam-us-assessment-ahmed-khan \
  --address-prefix 10.1.1.0/24
```

Created vnet pairing between the two vnets:

```
# Peering from vnet-tam-us-assessment-ahmed-khan to vnet2-tam-us-assessment-
ahmed-khan
az network vnet peering create \
  --name vnet-to-vnet2-peer \
  --resource-group rg-tam-us-assessment-ahmed-khan \
  --vnet-name vnet-tam-us-assessment-ahmed-khan \
  --remote-vnet /subscriptions/82852abb-55ae-44d8-9bac-
4632a4173215/resourceGroups/rg-tam-us-assessment-ahmed-
khan/providers/Microsoft.Network/virtualNetworks/vnet2-tam-us-assessment-ahmed-
khan \
  --allow-vnet-access

# Peering from vnet2-tam-us-assessment-ahmed-khan to vnet-tam-us-assessment-
ahmed-khan
az network vnet peering create \
```

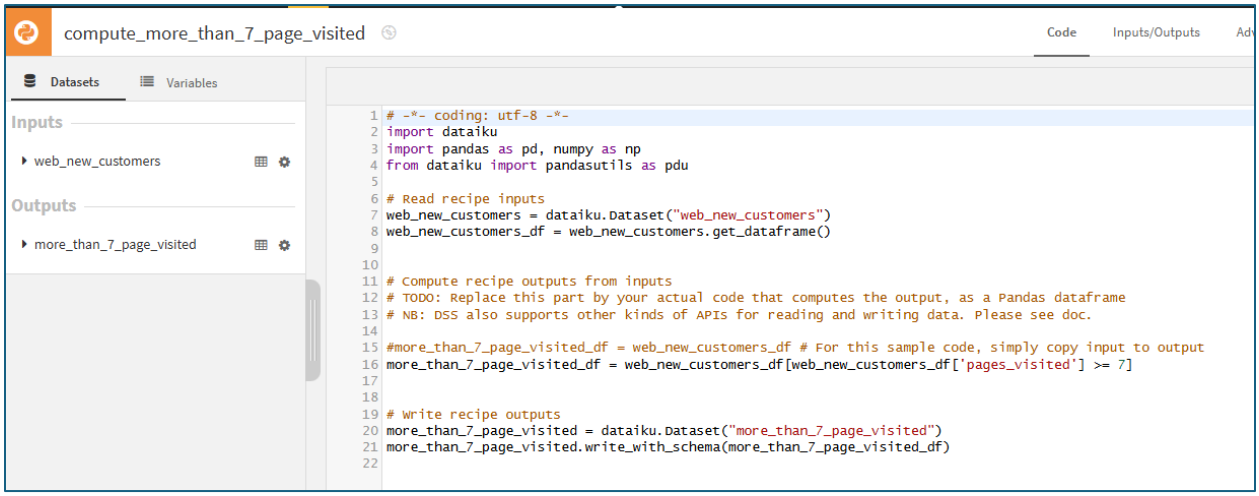
```
  --name vnet2-to-vnet-peer \
  --resource-group rg-tam-us-assessment-ahmed-khan \
  --vnet-name vnet2-tam-us-assessment-ahmed-khan \
  --remote-vnet /subscriptions/82852abb-55ae-44d8-9bac-
4632a4173215/resourceGroups/rg-tam-us-assessment-ahmed-
khan/providers/Microsoft.Network/virtualNetworks/vnet-tam-us-assessment-ahmed-
khan \
  --allow-vnet-access
```



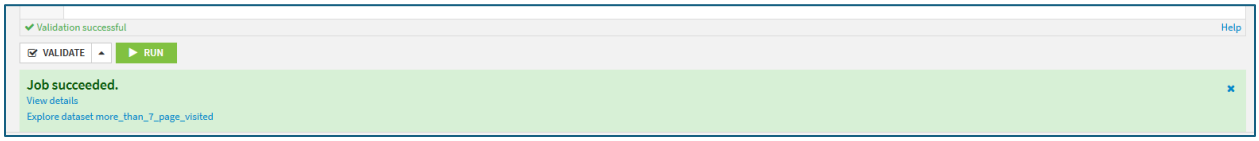


Check point : In the TShirt project, create a simple Python recipe and run it in a container

1. Creates a recipe using python called ‘compute_more_than_7_page_visited’.



2. Successfully completed the job.



Task 4: Expose DSS on the standard HTTPS port instead of HTTP

Followed the instructions in this link [Customizing DSS installation — Dataiku DSS 13 documentation]

1. Generated SSL certificate and key files.
   ```
   sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout selfsigned.key
   -out selfsigned.crt
   ```

2. Stopped DSS
   ```
   ./bin/dss stop
   ```

3. Modified /home/sadataiku/data/install.ini file
   ```
   [general]
   nodetype = design
   installid = HGjVFWb0Ne6dOrQ6tfNur4C3

   [server]
   port = 11200
   ssl = true
   ssl_certificate = /home/sadataiku/selfsigned.crt
   ssl_certificate_key = /home/sadataiku/selfsigned.key
   ssl_ciphers = recommended

   [git]
   mode = project

   [javaopts]
   backend.xmx = 4g
   ```

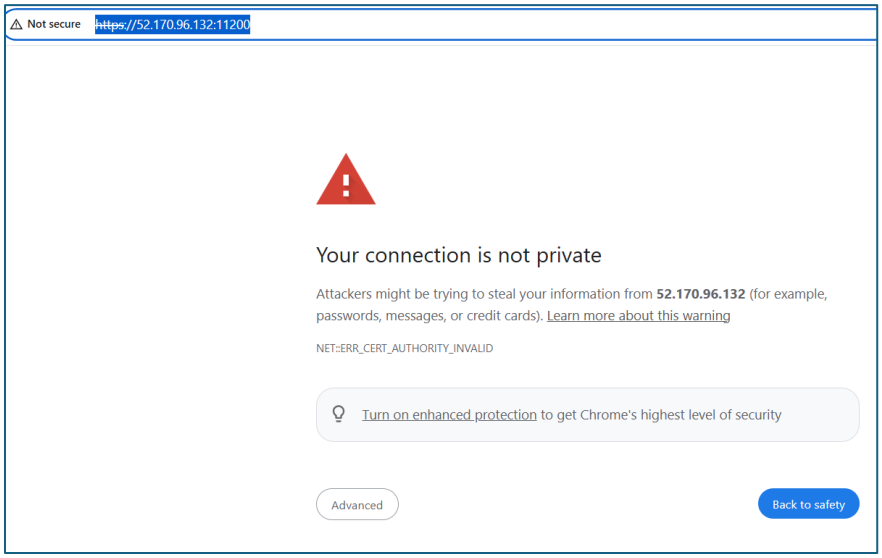
4. Regenerated the configuration
   ```
   ./bin/dssadmin regenerate-config
   ```

5. Started DSS
   ```
   ./bin/dss start
   ```

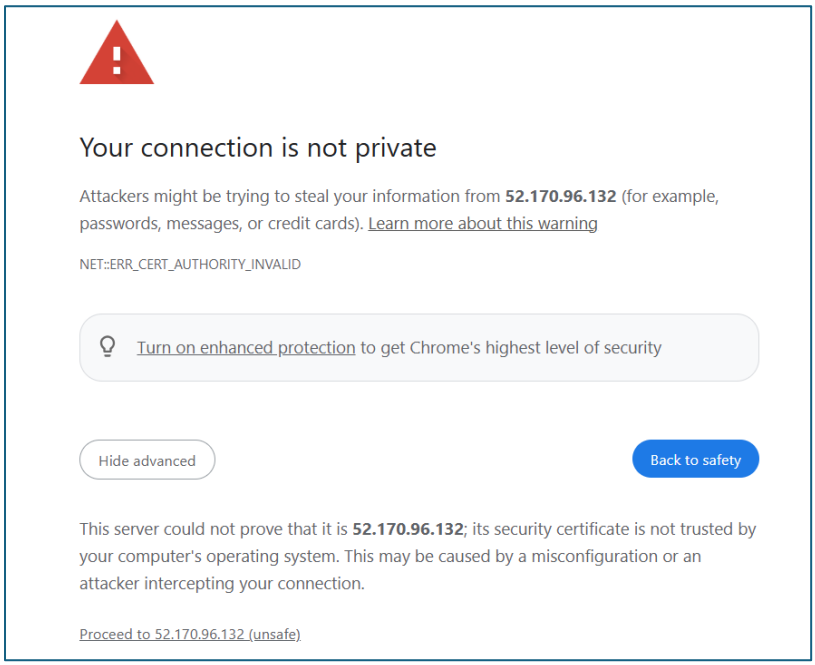
Checkpoint: Confirm that you can reach DSS instance with HTTPS

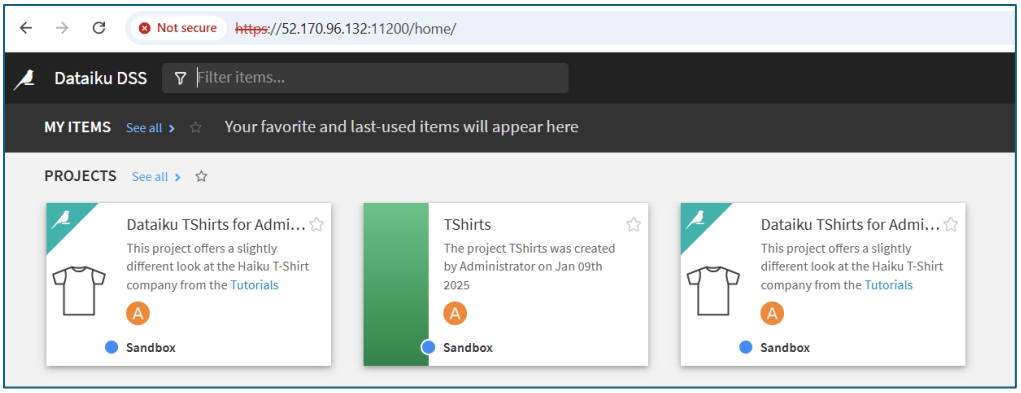
Visited DSS using HTTPS https://52.170.96.132:11200

Since the certificate was self-signed, there was a warning initially.



Go to Advanced and click 'Proceed to 52.170.96.132'




Not secure
https://52.170.96.132:11200/home/
Dataiku DSS
Filter items...
MY ITEMS
See all
Your favorite and last-used items will appear here
PROJECTS
See all
Dataiku TShirts for Admi...
This project offers a slightly different look at the Haiku T-Shirt company from the Tutorials
Sandbox
TShirts
The project TShirts was created by Administrator on Jan 09th 2025
Sandbox
Dataiku TShirts for Admi...
This project offers a slightly different look at the Haiku T-Shirt company from the Tutorials
Sandbox

Task 5: Setup your DSS instance so it can run Spark on AKS and interact with Azure Blob Storage (managed Spark on K8S recommended)

1. Followed the instructions to setup Spark using this link
   [Initial setup — Dataiku DSS 13 documentation]

   a. Downloaded Hadoop and Spark binary files in /home/sadataiku/softwares directory
      ```
      wget https://cdn.downloads.dataiku.com/public/dss/12.2.3/dataiku-dss-hadoop-standalone-libs-generic-hadoop3-12.2.3.tar.gz

      wget https://cdn.downloads.dataiku.com/public/dss/12.2.3/dataiku-dss-spark-standalone-12.2.3-3.4.1-generic-hadoop3.tar.gz
      ```

   b. Ran Hadoop and Spark integration
      ```
      ./bin/dssadmin install-hadoop-integration -standaloneArchive /home/sadataiku/softwares/dataiku-dss-hadoop-standalone-libs-generic-hadoop3-12.2.3.tar.gz

      ./bin/dssadmin install-spark-integration -standaloneArchive /home/sadataiku/softwares/dataiku-dss-spark-standalone-12.2.3-3.4.1-generic-hadoop3.tar.gz -forK8S
      ```

2. Created base images following the instructions in this link [Initial setup — Dataiku DSS 13 documentation]

   ```
   ./bin/dssadmin build-base-image --type container-exec
   ```

Issue no. 7: incorrect CentOS 7 mirror referenced in CentOS-Base.repo file

Where tried to run the command to build base image, the Dockerfile generated would use CentOS 7 with yum.repos.d pointing to deprecated mirror i.e., mirrorlist.centos.org

```
#8 1.061 Loaded plugins: fastestmirror, ovl
#8 1.263 Determining fastest mirrors
#8 1.276 Could not retrieve mirrorlist http://mirrorlist.centos.org/?release=7&arch=x86_64&repo=os&infra=container error was
#8 1.276 14: curl#6 - "Could not resolve host: mirrorlist.centos.org; Unknown error"
#8 1.279
#8 1.279
#8 1.279  One of the configured repositories failed (Unknown),
#8 1.279  and yum doesn't have enough cached data to continue. At this point the only
#8 1.279  safe thing yum can do is fail. There are a few ways to work "fix" this:
#8 1.279
#8 1.279      1. Contact the upstream for the repository and get them to fix the problem.
#8 1.279
#8 1.279      2. Reconfigure the baseurl/etc. for the repository, to point to a working
#8 1.279         upstream. This is most often useful if you are using a newer
#8 1.279         distribution release than is supported by the repository (and the
#8 1.279         packages for the previous distribution release still work).
#8 1.279
#8 1.279      3. Run the command with the repository temporarily disabled
#8 1.279             yum --disablerepo=<repoid>  ...
#8 1.279
#8 1.279      4. Disable the repository permanently, so yum won't use it by default. Yum
#8 1.279         will then just ignore the repository until you permanently enable it
#8 1.279         again or use --enablerepo for temporary usage:
#8 1.279
#8 1.279             yum-config-manager --disable <repoid>
#8 1.279         or
#8 1.279             subscription-manager repos --disable=<repoid>
#8 1.279
#8 1.279      5. Configure the failing repository to be skipped, if it is unavailable.
#8 1.279         Note that yum will try to contact the repo. when it runs most commands,
#8 1.279         so will have to try and fail each time (and thus. yum will be be much
#8 1.279         slower). If it is a very temporary problem though, this is often a nice
#8 1.279         compromise:
#8 1.279
#8 1.279             yum-config-manager --save --setopt=<repoid>.skip_if_unavailable=true
#8 1.279
#8 1.279 Cannot find a valid baseurl for repo: base/7/x86_64
#8 ERROR: process "/bin/sh -c yum -y update     && yum -y install epel-release     && . /etc/os-release && case \"$VERSION_ID\" in
       7*) yum -y install procps python3-devel python-devel;;         8*) yum -y install procps-ng python36-devel glibc-langpack-en
```

Resolution:

Modified build-image.py in /home/sadataiku/softwares/dataiku-dss-12.2.3/resources/container-exec directory to add the following lines for replacing the strings in /etc/yum.repo.d/CentOS* files

```
        self.dockerfile += """
RUN sed -i 's|^mirrorlist=|#mirrorlist=|g' /etc/yum.repos.d/CentOS-* && \\
    sed -i 's|^#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-* && \\
    yum clean all
"""
```

Issue no. 8: R package dependencies not found

Image build failed due to R dependencies not found.

```
712.9 ** testing if installed package keeps a record of temporary installation path
713.0 * DONE (dplyr)
714.2 ERROR: dependency 'gtable' is not available for package 'ggplot2'
714.2 * removing '/opt/dataiku/R/R.lib/3.x/ggplot2'
714.8 ERROR: dependency 'cpp11' is not available for package 'tidyr'
714.8 * removing '/opt/dataiku/R/R.lib/3.x/tidyr'
715.4 ERROR: dependency 'tidyr' is not available for package 'dbplyr'
715.4 * removing '/opt/dataiku/R/R.lib/3.x/dbplyr'
716.3 ERROR: dependencies 'dbplyr', 'httr', 'tidyr' are not available for package 'sparklyr'
716.3 * removing '/opt/dataiku/R/R.lib/3.x/sparklyr'
716.3
```

Resolution:

Ran the base image build without R

```
./bin/dssadmin build-base-image --type container-exec --without-r
```

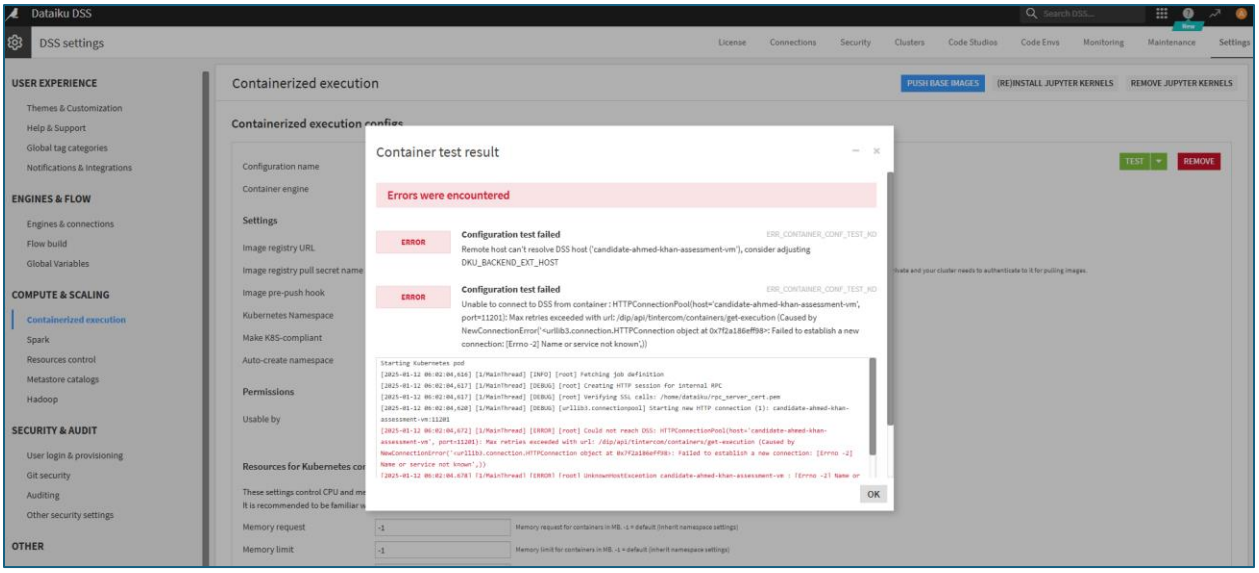
Similarly, created a base image for Spark

```
./bin/dssadmin build-base-image --type spark --without-r
```

3. Configured Containerized Execution and pushed the base image

Issue no. 9: Container unable to connect to host on port 11201

When tried to test the connection, container was not able to connect to the host at port 11201

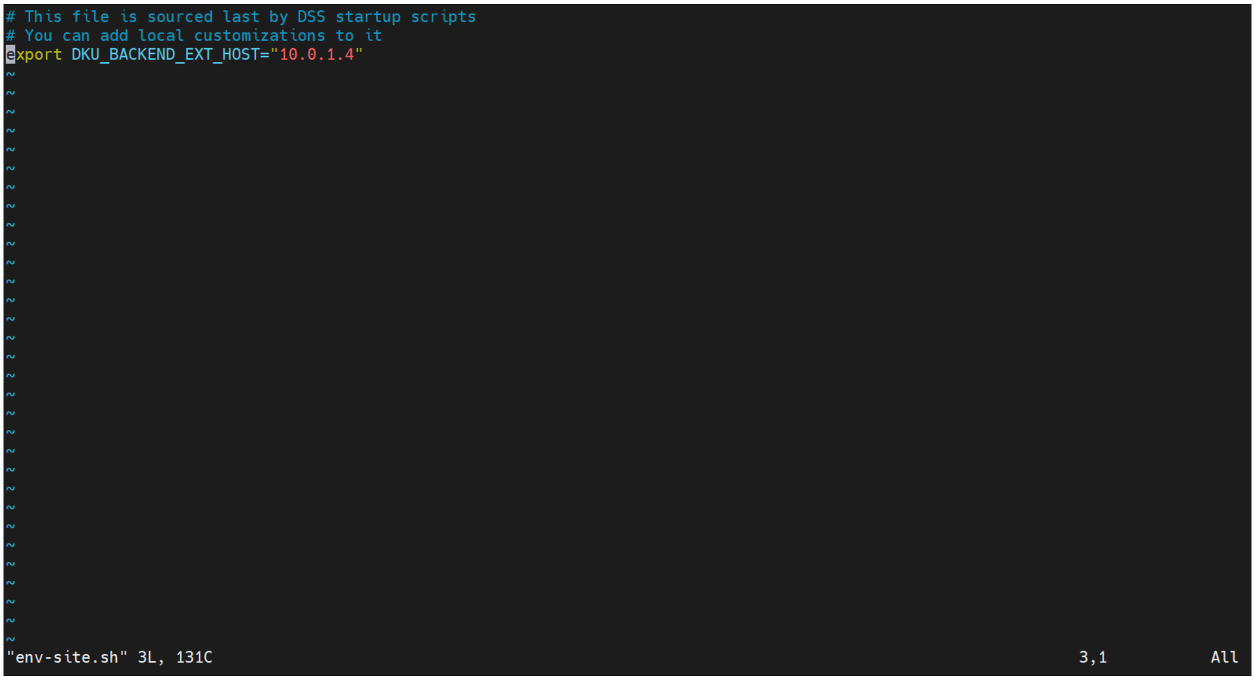


<u>Resolution</u>:

1. Opened the port 11201 in the security group

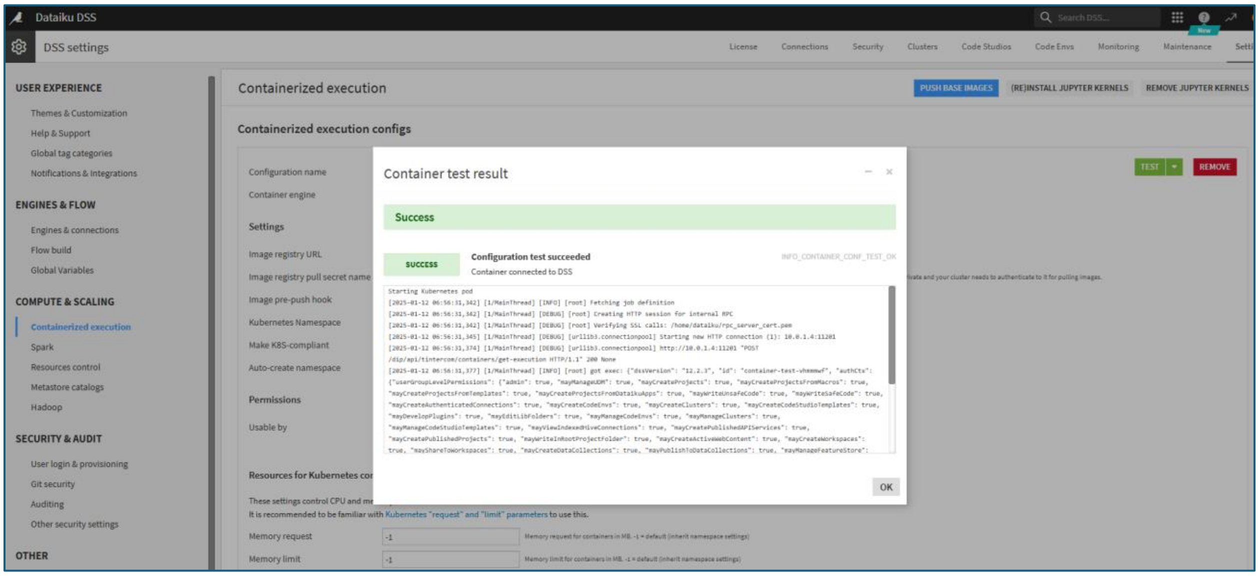
```
az network nsg rule create \
  --resource-group rg-tam-us-assessment-ahmed-khan \
  --nsg-name nsg-tam-us-assessment-ahmed-khan \
  --name AllowPort11201 \
  --priority 1000 \
  --direction Inbound \
  --access Allow \
  --protocol Tcp \
  --destination-port-ranges 11201
```

2. Added host ip in bin/env-site.sh



```
# This file is sourced last by DSS startup scripts
# You can add local customizations to it
export DKU_BACKEND_EXT_HOST="10.0.1.4"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"env-site.sh" 3L, 131C                                                                        3,1           All
```

Tested the connection successfully.



Container test result

Success

SUCCESS **Configuration test succeeded**
Container connected to DSS
INFO_CONTAINER_CONF_TEST_OK

```
Starting Kubernetes pod
[2025-01-12 06:56:31,342] [1/MainThread] [INFO] [root] Fetching job definition
[2025-01-12 06:56:31,342] [1/MainThread] [DEBUG] [root] Creating HTTP session for internal RPC
[2025-01-12 06:56:31,342] [1/MainThread] [DEBUG] [root] Verifying SSL calls: /home/dataiku/rpc_server_cert.pem
[2025-01-12 06:56:31,345] [1/MainThread] [DEBUG] [urllib3.connectionpool] Starting new HTTP connection (1): 10.0.1.4:11201
[2025-01-12 06:56:31,374] [1/MainThread] [DEBUG] [urllib3.connectionpool] http://10.0.1.4:11201 "POST
/dip/api/tintercom/containers/get-execution HTTP/1.1" 200 None
[2025-01-12 06:56:31,377] [1/MainThread] [INFO] [root] got exec: {"dssVersion": "12.2.3", "id": "container-test-vhmmwf", "authCtx":
{"userGroupLevelPermissions": {"admin": true, "mayManageUDM": true, "mayCreateProjects": true, "mayCreateProjectsFromMacros": true,
"mayCreateProjectsFromTemplates": true, "mayCreateProjectsFromDataikuApps": true, "mayWriteUnsafeCode": true, "mayWriteSafeCode": true,
"mayCreateAuthenticatedConnections": true, "mayCreateCodeEnvs": true, "mayCreateClusters": true, "mayCreateCodeStudioTemplates": true,
"mayDevelopPlugins": true, "mayEditLibFolders": true, "mayManageCodeEnvs": true, "mayManageClusters": true,
"mayManageCodeStudioTemplates": true, "mayViewIndexedHiveConnections": true, "mayCreatePublishedAPIServices": true,
"mayCreatePublishedProjects": true, "mayWriteInRootProjectFolder": true, "mayCreateActiveWebContent": true, "mayCreateWorkspaces":
true, "mayShareToWorkspaces": true, "mayCreateDataCollections": true, "mayPublishToDataCollections": true, "mayManageFeatureStore":
```

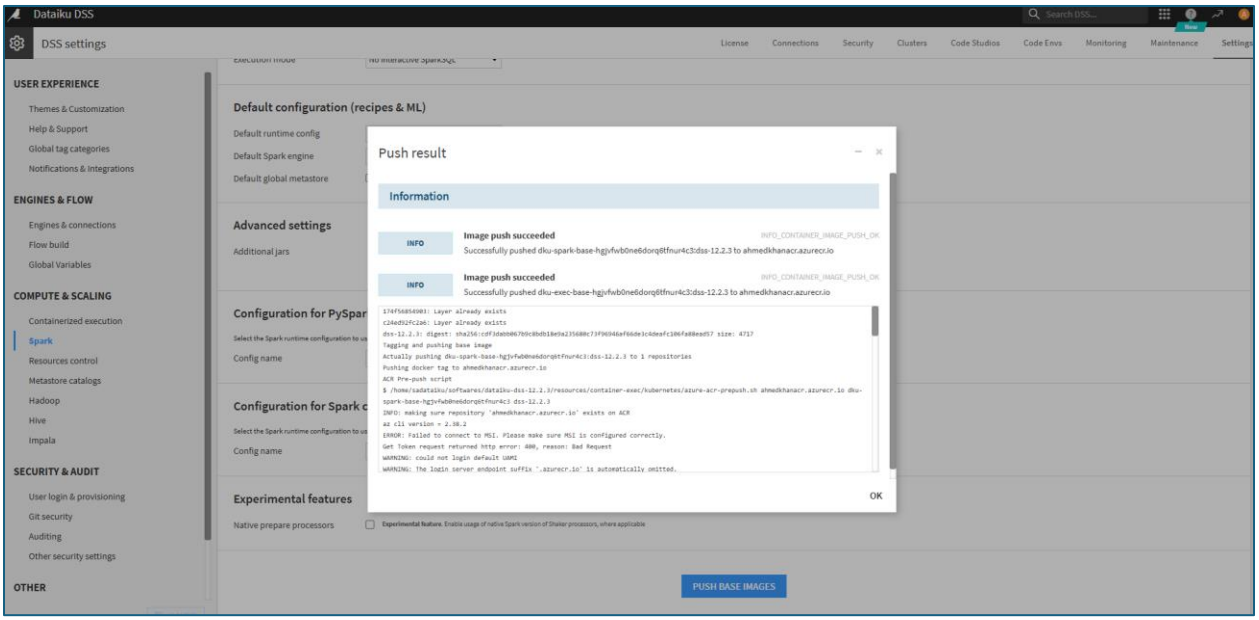
OK

These settings control CPU and me
It is recommended to be familiar with Kubernetes "request" and "limit" parameters to use this.

Memory request -1 Memory request for containers in MB. -1 = default (inherit namespace settings)

Memory limit -1 Memory limit for containers in MB. -1 = default (inherit namespace settings)

Pushed the spark image successfully.

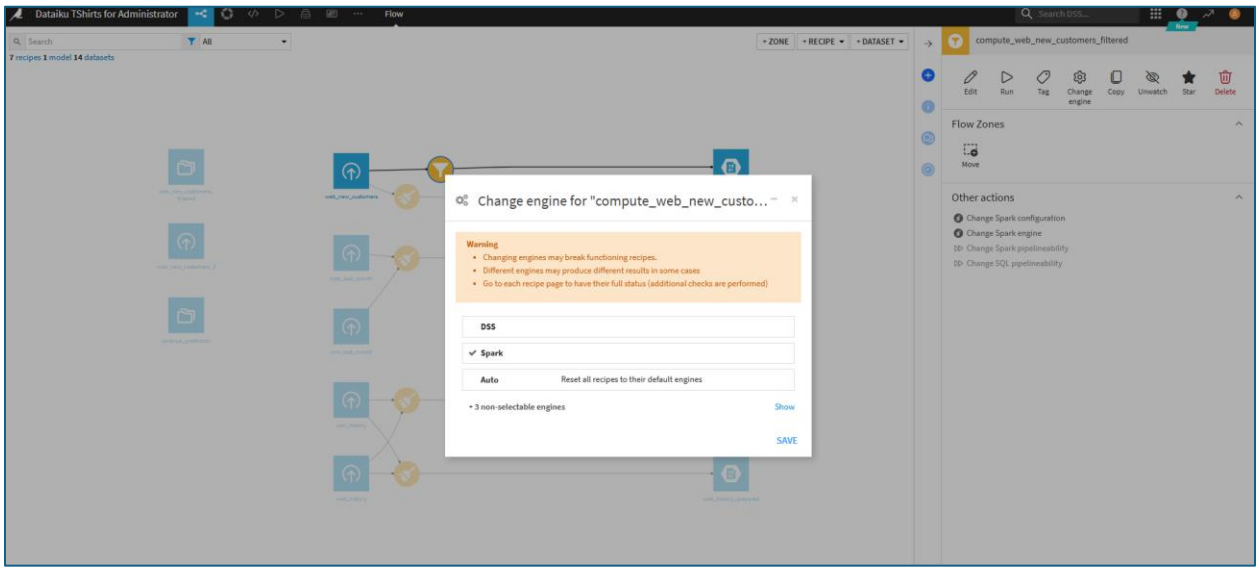


Verified the docker images.

```
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$ docker images
REPOSITORY                                                      TAG          IMAGE ID       CREATED         SIZE
ahmedkhanacr.azurecr.io/dku-spark-base-hgjvfwb0ne6dorq6tfnur4c3 dss-12.2.3   aa7c7e13b306   12 hours ago    2.78GB
dku-spark-base-hgjvfwb0ne6dorq6tfnur4c3                         dss-12.2.3   aa7c7e13b306   12 hours ago    2.78GB
ahmedkhanacr.azurecr.io/dku-exec-base-hgjvfwb0ne6dorq6tfnur4c3  dss-12.2.3   79cd7baaf8ae   12 hours ago    1.8GB
dku-exec-base-hgjvfwb0ne6dorq6tfnur4c3                          dss-12.2.3   79cd7baaf8ae   12 hours ago    1.8GB
centos7.5-test                                                  latest       429f5ef90c11   15 hours ago    623MB
hello-world                                                     latest       d2c94e258dcb   20 months ago   13.3kB
[sadataiku@candidate-ahmed-khan-assessment-vm ~]$
```

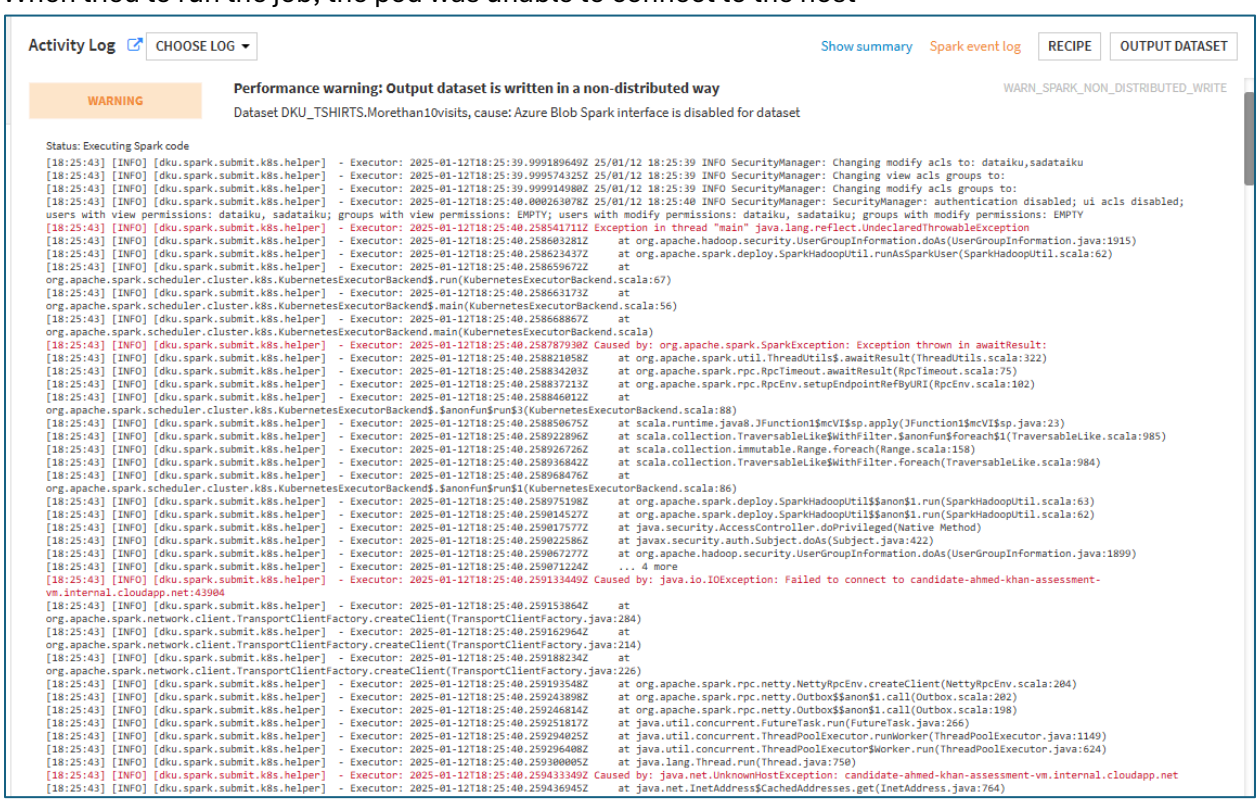
Check point : In the TShirt project, change the execution engine of the visual recipes to Spark and run it the AKS cluster

1. Changed the execution engine to Spark



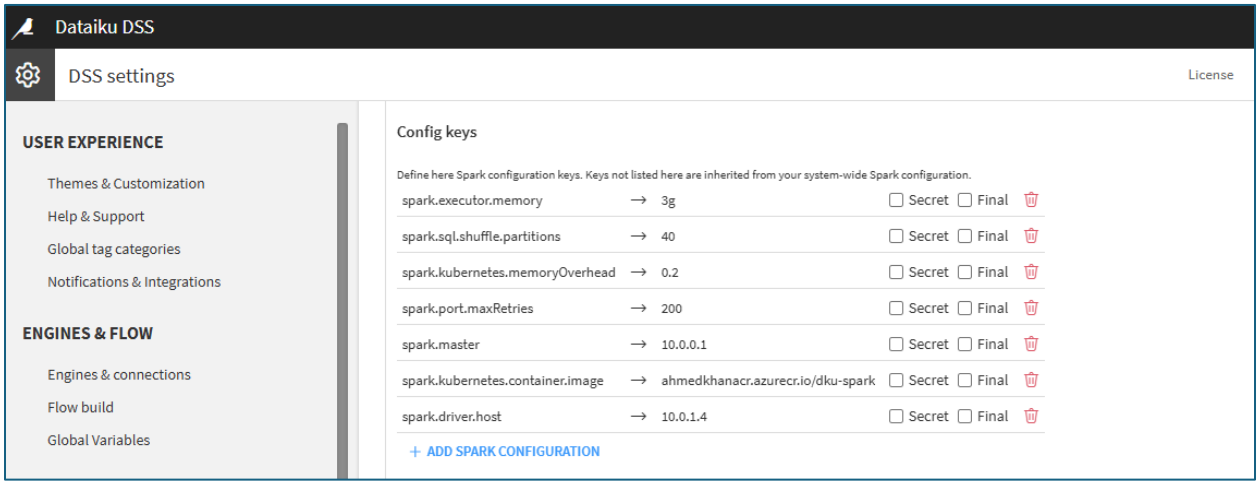
Issue no. 10: K8s pod failed to connect to the host

When tried to run the job, the pod was unable to connect to the host



Resolution:

Added spark.driver.host key in the Spark configuration:



Spark job completed successfully.