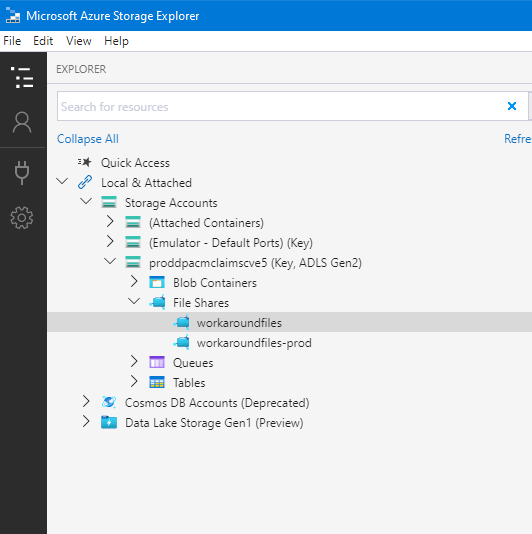
**MIS Claims Documentation**

1. Fileshare
2. Postgres
3. DQ Checks
4. Kafka PipeLines
5. Jenkins Pipelines
6. tibco

**Azure fileshare**

****

Obtain read write access to the file share

proddpacmclaimscve5

[4:15](https://dse-world.slack.com/archives/D01V9F72LD8/p1645445703995429)

Resource group ():[alz-gdp-dpacmclaims-gdp-base-customer-rg](https://portal.azure.com/" \l "blade/HubsExtension/ResourceMenuBlade/id/%2Fsubscriptions%2F66dad72a-9dc3-4cfe-8b07-1c5f4c02885a%2FresourceGroups%2Falz-gdp-dpacmclaims-gdp-base-customer-rg)Location :West Europe Subscription () : [aztecse-gdp-06-prod](https://portal.azure.com/#_blank)  
Subscription ID:66dad72a-9dc3-4cfe-8b07-1c5f4c02885aDisk state

**Postgre database**

Posgre database is a located at

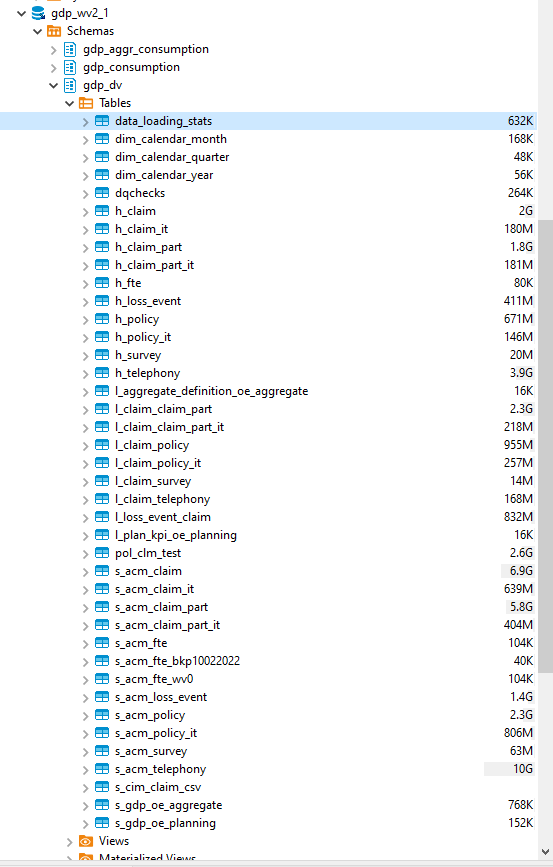
[aztecse-gdp-06-prod](_blank)

Subscription ID: 66dad72a-9dc3-4cfe-8b07-1c5f4c02885a

you can obtain credentials from keyvault

<https://portal.azure.com/#@allianzms.onmicrosoft.com/resource/subscriptions/66dad72a-9dc3-4cfe-8b07-1c5f4c02885a/resourceGroups/alz-gdp-dpacmclaims-gdp-base-customer-rg/providers/Microsoft.KeyVault/vaults/dpacmclaimscve5/secrets>

database contians two workstream databases gdp\_wv2\_0 and gdp\_wv2\_1



tables and views are build accordence to Datamodel V125

to view dat amode schemas obtain sqldbm.com access

**DQ Checks**

Dqchecks is not GDP Allianz Approved and managed tool you can skipp it and use Kafka directly at your convenience by configuring Kafka source connectors to your desired folders.

You can read DQchecks documentation and solution architecture in confluence

[Claims MIS DQ Checks - Global Data Platform (GDP) - JMP Confluence (allianz.net)](https://cmp.allianz.net/display/GDPG/Claims+MIS+DQ+Checks)

Dqchecks sroucecode is in github [global-data-platform/rollout-dpclaims (allianz.io)](https://github.developer.allianz.io/global-data-platform/rollout-dpclaims)

WV2.1 [rollout-dpclaims/wv21dqchecks at main · global-data-platform/rollout-dpclaims (allianz.io)](https://github.developer.allianz.io/global-data-platform/rollout-dpclaims/tree/main/wv21dqchecks)

WV2.0 [rollout-dpclaims/dqchecks at main · global-data-platform/rollout-dpclaims (allianz.io)](https://github.developer.allianz.io/global-data-platform/rollout-dpclaims/tree/main/dqchecks)

Checkout the repository

by default there no passwords in sourcecode ,ask your manager

To build and deploy DQ application run build.sh command done.

How it works

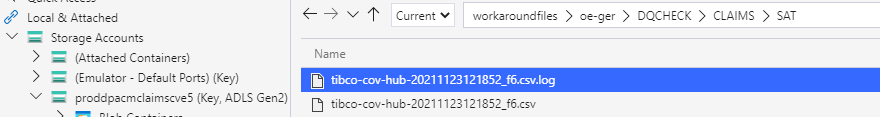
obtain Azure GDP-developer Contributor role

DQ deployed in dqchecks namespace and runs every 10 hours, you can suspend/resume/trigger any time .

Dqcronjob will check files in entry folder workaroundfiles/oe-ger/ passes it for technical DQ checks and if check was successful it will move to the workaroundfiles/kafka folder for kafka import

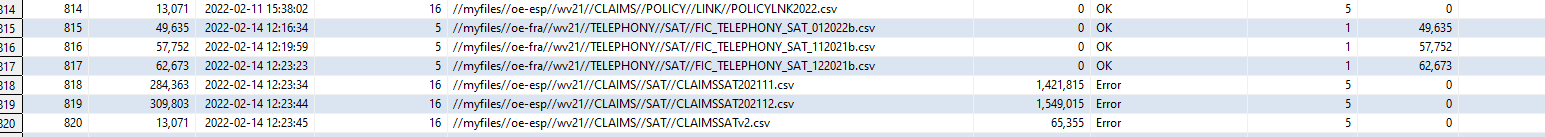
In case of any error file will be copied in dqcheck subfolder with corresponding error log file

So that OE can view their error list and react on it



Dq populates the gdp.wv21 or gdp.wv20 data base DQCHECK table depending which worksream you are running wv2.1 or wv2.0

Dqcheck table describes file dqcheck load history and details in short form



But error log file in fileshare describes detailed error occurred for specific row

This is how it looks

error in 0ed90a87-02f5-11eb-abdf-005056990158-500f7f38-1767-11e1-9561-030bbf36030b -> cim\_load\_to\_ts wrong date null

error in 0ed90a87-02f5-11eb-abdf-005056990158-500f7f38-1767-11e1-9561-030bbf36030b -> cim\_load\_to\_ts exception Unparseable date: "null" null

error in 0ed90a87-02f5-11eb-abdf-005056990158-500f7f38-1767-11e1-9561-030bbf36030b -> cim\_invalid\_ts wrong date null

error in 0ed90a87-02f5-11eb-abdf-005056990158-500f7f38-1767-11e1-9561-030bbf36030b -> cim\_invalid\_ts exception Unparseable date: "null" null

error in 546cf892-cb19-11ea-908b-00505699c336-a697edee-738c-11e3-b771-030bbf36044f -> cim\_load\_to\_ts wrong date null

error in 546cf892-cb19-11ea-908b-00505699c336-a697edee-738c-11e3-b771-030bbf36044f -> cim\_load\_to\_ts exception Unparseable date: "null" null

error in 546cf892-cb19-11ea-908b-00505699c336-a697edee-738c-11e3-b771-030bbf36044f -> cim\_invalid\_ts wrong date null

error in 546cf892-cb19-11ea-908b-00505699c336-a697edee-738c-11e3-b771-030bbf36044f -> cim\_invalid\_ts exception Unparseable date: "null" null

OE can fix this and uploadagain

build DQ checks application

clone from git clone https://github.developer.allianz.io/global-data-platform/rollout-dpclaims.git

change directory to rollout-dpclaims/wv21dqchecks/wv21dqchecks

run build.sh

script will automatically deploy dqchecks crojob to the kubernetes cluster

**Kafka Pipelines**

Kafka piplines are already deployied for all Oes Oe-ger,oe-fr,oe-it,oe-aut,oe-sp and for both work streams wv20 and wv21

├───CLAIMS  
 │  ├───COVERAGE  
 │  │  ├───HUB  
 │  │  ├───LINK  
 │  │  └───SAT  
 │  ├───HUB  
 │  └───SAT  
 ├───FTE  
 │  ├───HUB  
 │  └───SAT  
 └───TELEPHONY  
   ├───HUB  
   ├───LINK  
   └───SAT

dq checks will parse files for errors and place it in the corresponding kafka folder for loading

for example

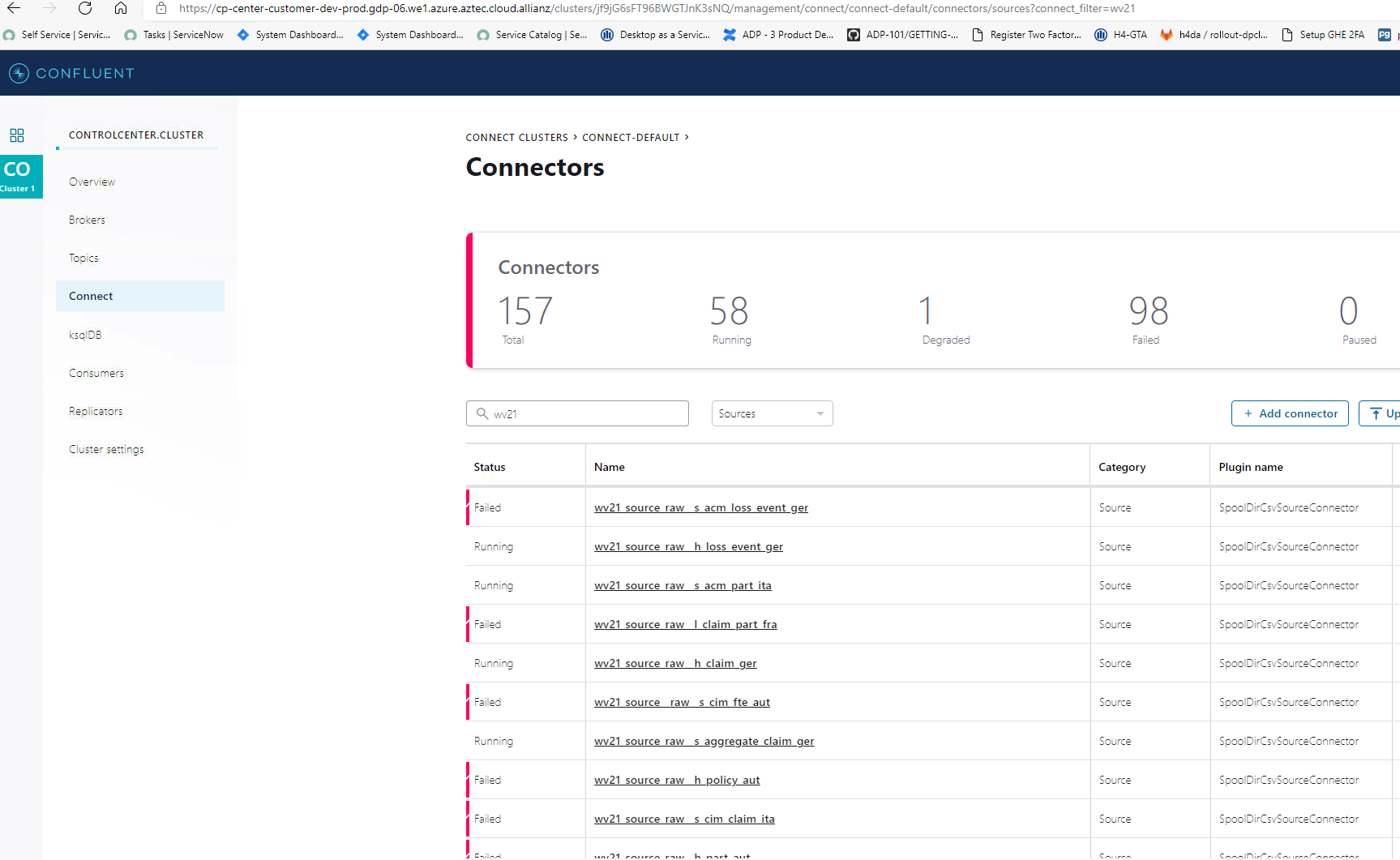
checking //myfiles//oe-ger//wv21//AGGREGATE//SAT//AGG\_DE\_JAN\_2022.csv

error\_count=0

file\_moved //myfiles//oe-ger//wv21//AGGREGATE//SAT//AGG\_DE\_JAN\_2022.csv to //myfiles//kafka//oe-ger//wv21//AGGREGATE//SAT//AGG\_DE\_JAN\_2022.csv

an dwill update the DQCHECKS table with a status OK or ERROR

each OE has own kafka source connector



it is ok to see failed source connector meaning that file is not yet available

inital connector run requires manual one time restart when file is evailable then it is permanently initialized

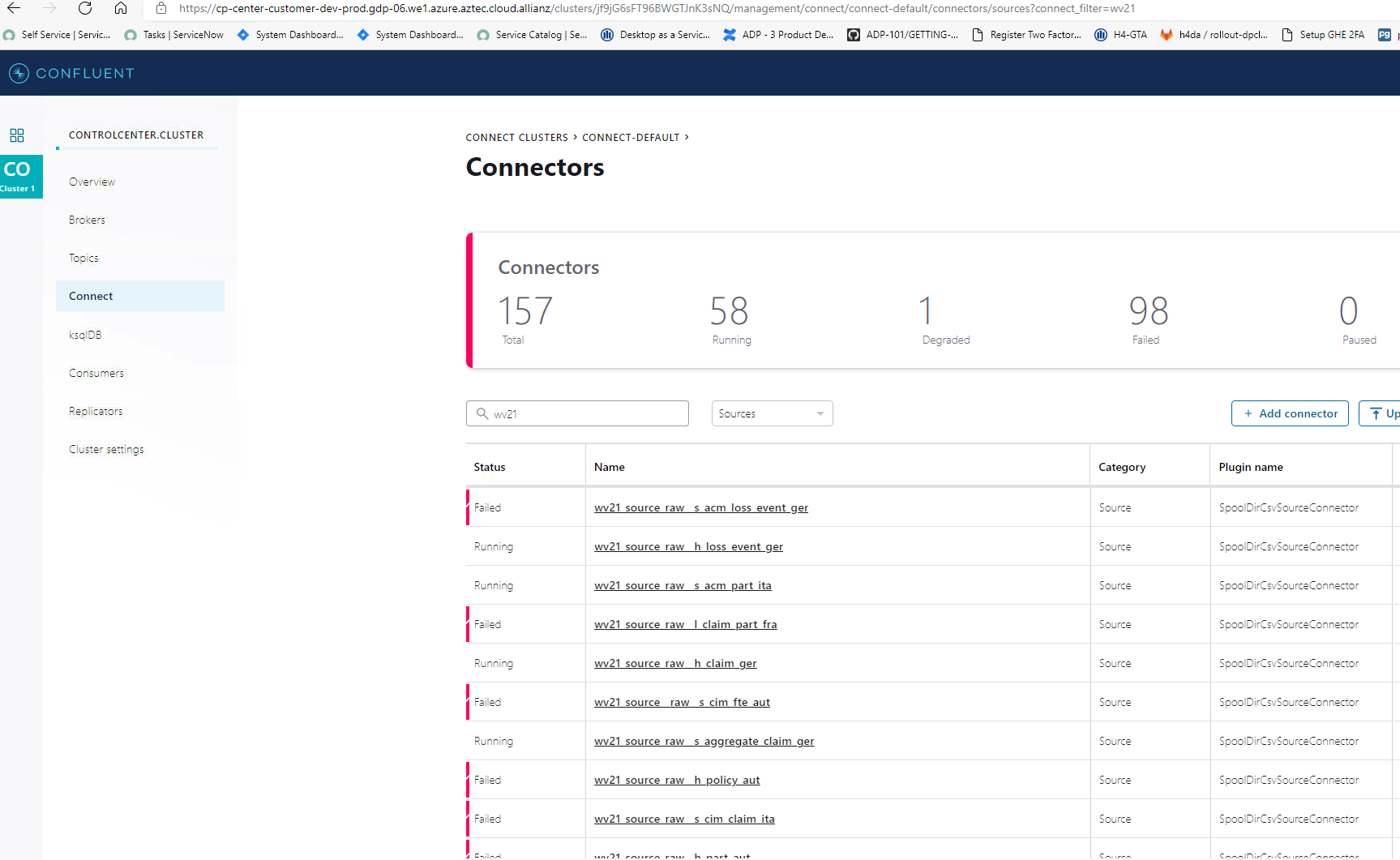
first manual run is also important for kafka topic schema gneration

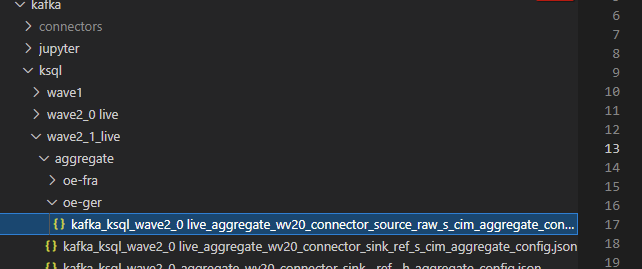
retention perios is set 1 week for small files and 30 minutes for big files

**Kafka Pipeline deployment**

all source /sink connector and stream transformers are stored in https://github.developer.allianz.io/global-data-platform/rollout-dpclaims/tree/main/kafka/ksql

to deploy source connector use KafkaControl center

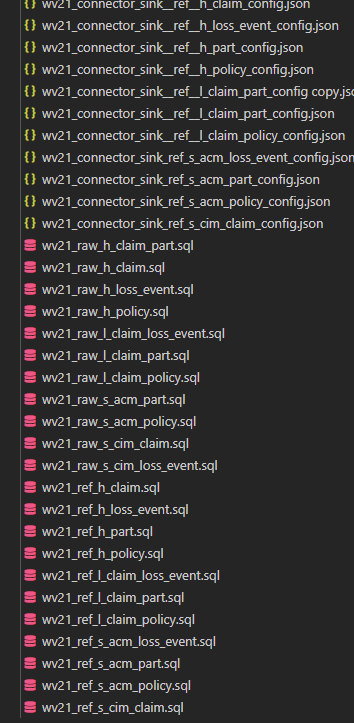
and press upload Connector button then choose corersponding connector from file system and deploy it



connect to the KSQL engine and deploy Transofmrations

kubectl run my-ksql-cli -it --rm -n customer-dev --requests cpu=100m,memory=256Mi --limits cpu=200m,memory=512Mi --image confluentinc/ksqldb-cli:0.7.1 ksql http://kafka-cp-ksql-server:8088

tranformers are located in ksql folder



paste content of the transofmration in commandline or use any http client at your convenience described in ksql documentation https://docs.ksqldb.io/en/latest/

to make kafka pipeline working you have to deply sourc econnector,RAW transofmrer,REF transofmrer and Sink connectors

raw and ref transformers can be easily distinguished by its nameing convention wv21\_raw\* wv21\_ref\*

**debuging trouble shooting**

**sink connector failed or DRAGGED status**

solution clean kafka topic , drop sink connecto rand deploy again

**field changed in database**

solution drop all transformeters and connectors make changes in source and deploy again topic names must be different than old ones

**Jenkins Pipeline for SFTP**

https://github.developer.allianz.io/global-data-platform/customer-dpacmclaims/blob/master/infrastructure/customer-resources/sftp-servers.tf

this is instructions we give OE

1.      Generate the public and the private key executing using this command: ssh-keygen -b 4096  
2.      Keys are generated under your home directors ~/.ssh/  
3.      Send the public key to the GDP Claims Solution Engineer [extern.robakidze\_nikoloz@allianz.de](mailto:extern.robakidze_nikoloz@allianz.de) (keep the private key for yourself)    
4.      Connect using your private key  
a.      Host:                  sftp.oe-aut.gdp-06.we1.azure.aztec.cloud.allianz  
b.      Username:       sftp  
c.      Port Number:  225.      You are free to generate multiple keys if you have multiple upload locations.  
6.      Optionally: you can always use WINSCP sftp client from your AVC.  
   
   
SFTP upload folder structure:├───CLAIMS  
 │  ├───COVERAGE  
 │  │  ├───HUB  
 │  │  ├───LINK  
 │  │  └───SAT  
 │  ├───HUB  
 │  └───SAT  
 ├───FTE  
 │  ├───HUB  
 │  └───SAT  
 └───TELEPHONY  
   ├───HUB  
   ├───LINK  
   └───SAT

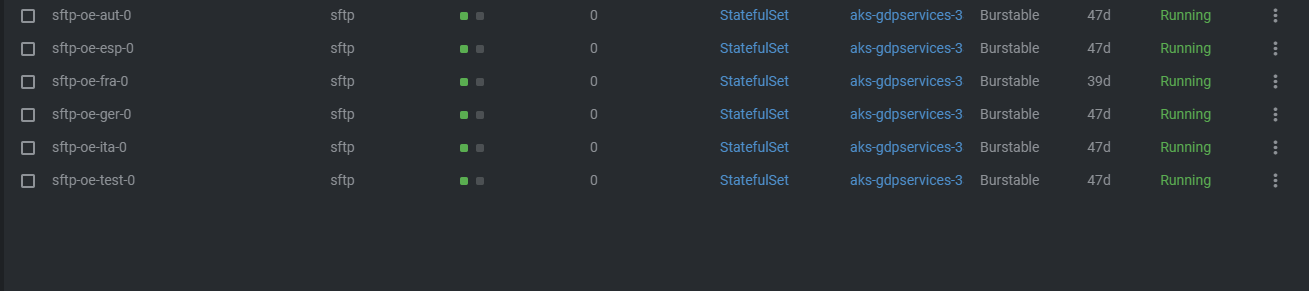
then after you receive Public key from OE creat branch and add key to the sftp-servers.tf

and build NOW jenkins pipeline https://jenkins-dsp.devops-services.we1.azure.aztec.cloud.allianz/job/GDP\_customer/job/customer-dpacmclaims/

choose gdp 04 and finish the build

after successfull jenkins build goto the Kubernetes cluster and select SFTP namespace

and restart SFTP instance in order to apply changes

make pull request to DSP to merge your branch change to the master branch

References

[Claims MIS Technical Solution - Global Data Platform (GDP) - JMP Confluence (allianz.net)](https://cmp.allianz.net/display/GDPG/Claims+MIS+Technical+Solution)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool Name** | **Used for** | **Link to tool/repository** | **Mandatory permissions and or necessary group membership** | **etc…** |
| GitHub | Source code for pipelines uptodate | [global-data-platform/rollout-dpclaims (allianz.io)](https://github.developer.allianz.io/global-data-platform/rollout-dpclaims) |  |  |
| GitLab | Source code for pipelines old | <https://gitlab.gda.allianz/h4da/rollout-dpclaims> |  |  |
| kafka production | kafka connect admin interface | <https://cp-center-customer-dev-prod.gdp-06.we1.azure.aztec.cloud.allianz/> |  |  |
| kafka dev | kafka connect admin interface | [https://cp-center-customer-prod-prod.gdp-06.we1.azure.aztec.cloud.allianz/](https://cp-center-customer-dev-prod.gdp-06.we1.azure.aztec.cloud.allianz/) |  |  |
| Postgre | postgre secrets are protected in Azure secrets | key vault dpacmclaimscve5 | postgres-endpoint  postgres-password  postgres-username |  |
| Kubernete kafka prod namspace |  | namespace | customer-dev |  |
| Kubernete kafka dev namspace |  | namespace | customer-prod |  |
| kubernetes sftp |  | namespace | sftp |  |
| kubernetes dqchecks |  | namespace | dqchecks |  |
| claimsdchecks-job |  | deployment | dqchecks kubernetes jobs |  |
| claimstibcojobs |  | deployment | tibco kubernetes job |  |
| azure fileshare |  | [proddpacmclaimscve5 - Microsoft Azure](https://portal.azure.com/#@allianzms.onmicrosoft.com/resource/subscriptions/66dad72a-9dc3-4cfe-8b07-1c5f4c02885a/resourceGroups/alz-gdp-dpacmclaims-gdp-base-customer-rg/providers/Microsoft.Storage/storageAccounts/proddpacmclaimscve5/overview) | proddpacmclaimscve5 |  |
| azure subscription |  |  | [aztecse-gdp-06-prod](https://portal.azure.com/#blade/HubsExtension/ResourceMenuBlade/id/%2Fsubscriptions%2F66dad72a-9dc3-4cfe-8b07-1c5f4c02885a) |  |
| jenkins repository | used for SFTP deployment | <https://github.developer.allianz.io/global-data-platform/customer-dpacmclaims> |  |  |
| jenkins installation |  | [Branches (10) [GDP\_customer » customer-dpacmclaims] [Jenkins] (cloud.allianz)](https://jenkins-dsp.devops-services.we1.azure.aztec.cloud.allianz/job/GDP_customer/job/customer-dpacmclaims/) |  |  |
| Jupyterhub |  | <https://jupyterhub-prod.gdp-06.we1.azure.aztec.cloud.allianz/hub> |  |  |
| Storag Explorer |  | https://gdp-user-documentation.dsp1.we1.azure.aztec.cloud.allianz/dataingestion/storage-explorer/ |  |  |

**Tibco**

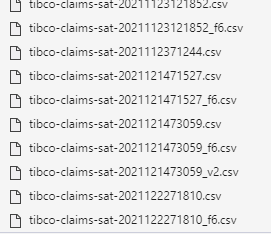
Jus tlike DQChecks Tibco kubernetes job is executed on schedule and pulls data from Germany Tibco server and saves in DQ checks entry folder for quality checks

https://github.developer.allianz.io/global-data-platform/rollout-dpclaims/tree/main/tibco

execute build.xml it ill deploy it on the kubernetes cluster

Tibco job will download data and save it as a CSV file in DQ entry folder for example *workaroundfile/oe-ger/ so that dq script can pick it up*

*tibco files have namin conventions*

**