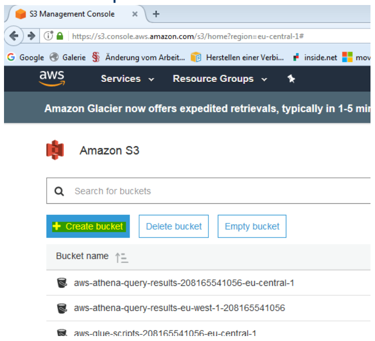
Anlegen eines neues AWS EMR Clusters

- Anhang:

Zuerst müssen zwei S3 Buckets bei AWS in der Region eu-central-1 angelegt werden.
Die URL https://s3.console.aws.amazon.com/s3/home?region=eu-central-1# aufrufen und mit „Create Bucket" ein s3 Bucket erzeugen.


S3 Management Console
https://s3.console.aws.amazon.com/s3/home?region=eu-central-1#
Google Galerie Änderung vom Arbeit... Herstellen einer Verbi... inside.net mov

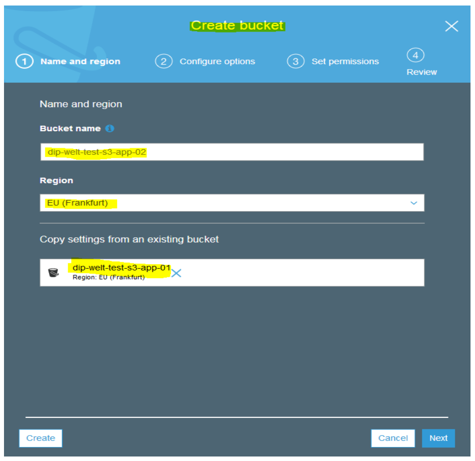
Amazon Glacier now offers expedited retrievals, typically in 1-5 mi

Amazon S3

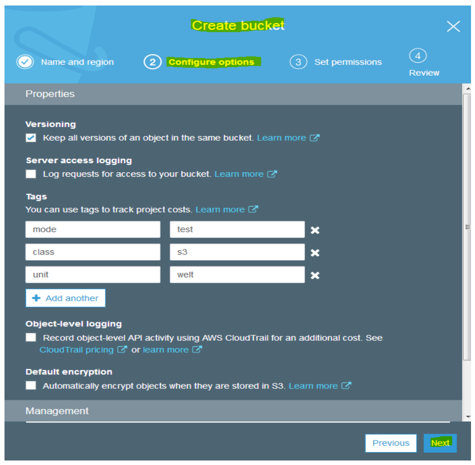
Search for buckets

Bucket name
aws-athena-query-results-208165541056-eu-central-1
aws-athena-query-results-eu-west-1-208165541056
aws-glue-scripts-208165541056-eu-central-1

1. dip-xxx-test-s3-app-01: Hier werden alle Skript- und Konfiguration-Dateien abgelegt. Das Bucket wird mit einer Versionierung-Option angelegt. Dadurch sind alle Skripte etc. versioniert.
2. dip-xxx-test-s3-data-01: Hier werden alle Daten (Rohdaten, Warehouse-Daten, Log-Daten etc.) abgelegt. Es erfolgt keine Historisierung der Dateien.


Create bucket

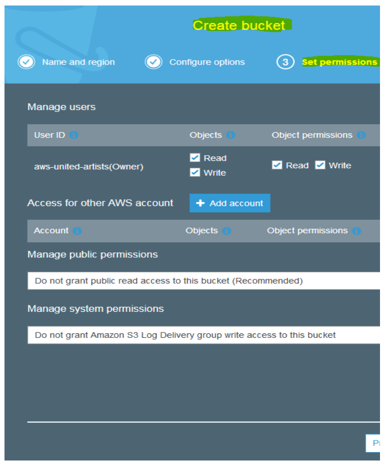
Name and region
Bucket name: dip-welt-test-s3-app-02
Region: EU (Frankfurt)
Copy settings from an existing bucket: dip-welt-test-s3-app-01
Region: EU (Frankfurt)


Create bucket

Properties
Versioning
[✓] Keep all versions of an object in the same bucket. Learn more
Server access logging
[ ] Log requests for access to your bucket. Learn more
Tags
You can use tags to track project costs. Learn more

mode	test
class	s3
unit	welt

Object-level logging
[ ] Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more
Default encryption
[ ] Automatically encrypt objects when they are stored in S3. Learn more
Management


Create bucket

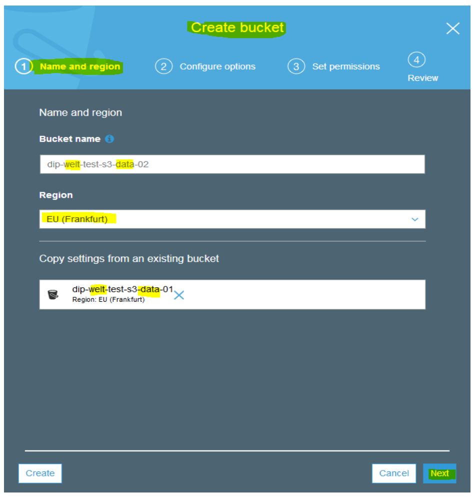
Manage users

User ID	Objects	Object permissions
aws-united-artists(Owner)	✓ Read ✓ Write	✓ Read ✓ Write

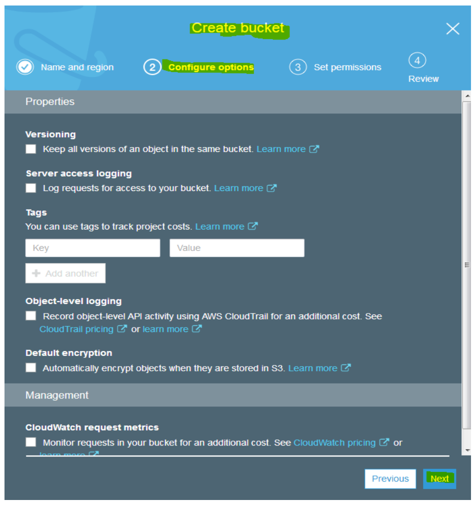
Access for other AWS account

Account	Objects	Object permissions

Manage public permissions
Do not grant public read access to this bucket (Recommended)
Manage system permissions
Do not grant Amazon S3 Log Delivery group write access to this bucket


Create bucket

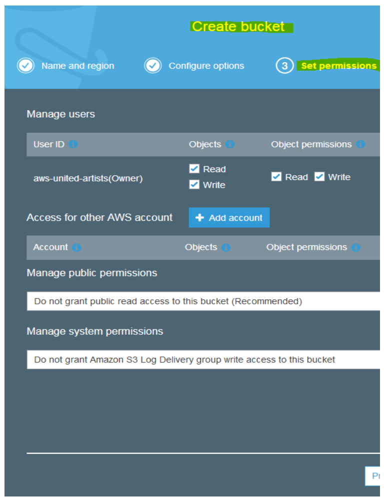
Name and region
Bucket name: dip-welt-test-s3-data-02
Region: EU (Frankfurt)
Copy settings from an existing bucket: dip-welt-test-s3-data-01
Region: EU (Frankfurt)


Create bucket

Properties
Versioning
[ ] Keep all versions of an object in the same bucket. Learn more
Server access logging
[ ] Log requests for access to your bucket. Learn more
Tags
You can use tags to track project costs. Learn more

Key	Value

Object-level logging
[ ] Record object-level API activity using AWS CloudTrail for an additional cost. See CloudTrail pricing or learn more
Default encryption
[ ] Automatically encrypt objects when they are stored in S3. Learn more
Management
CloudWatch request metrics
[ ] Monitor requests in your bucket for an additional cost. See CloudWatch pricing or learn more


Create bucket

Manage users

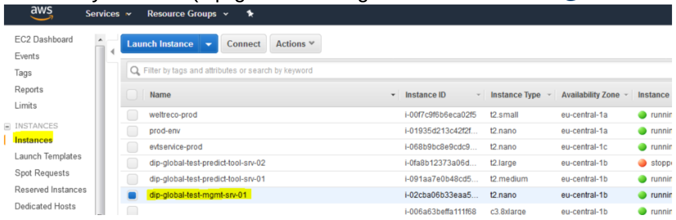
User ID	Objects	Object permissions
aws-united-artists(Owner)	✓ Read ✓ Write	✓ Read ✓ Write

Access for other AWS account

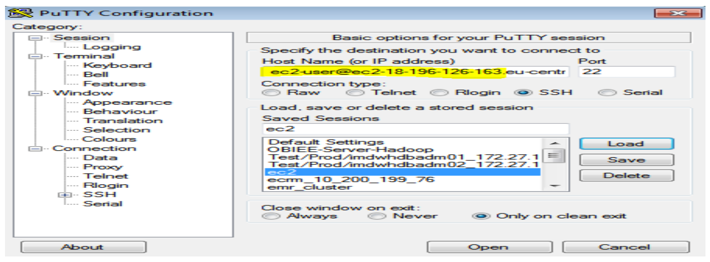
Account	Objects	Object permissions

Manage public permissions
Do not grant public read access to this bucket (Recommended)
Manage system permissions
Do not grant Amazon S3 Log Delivery group write access to this bucket

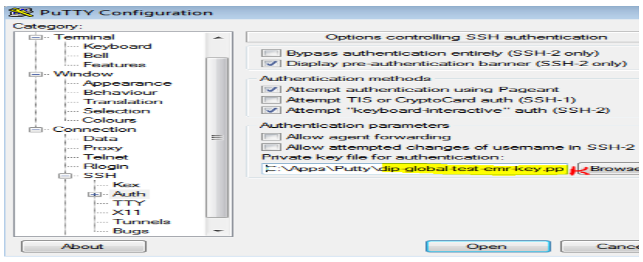
Über Putty auf EC2 (dip-global-test-mgmt-srv-01 - ec2-user@ec2-18-196-126-163.eu-central-1.compute.amazonaws.com) einloggen.



Name	Instance ID	Instance Type	Availability Zone	Instance
weltreco-prod	i-00f7c9f6b6eca02f5	t2.small	eu-central-1a	runnir
prod-env	i-01935d213c42f2f...	t2.nano	eu-central-1a	runnir
evtservice-prod	i-068b9bc8e9cdc9...	t2.nano	eu-central-1c	runnir
dip-global-test-predict-tool-srv-02	i-0fa8b12373a06d...	t2.large	eu-central-1b	stopp
dip-global-test-predict-tool-srv-01	i-091aa7e0b48cd5...	t2.medium	eu-central-1b	runnir
dip-global-test-mgmt-srv-01	i-02cba06b33eaa5...	t2.nano	eu-central-1b	runnir
	i-006a63beffa111f68	c3.8xlarge	eu-central-1b	runnir



Specify the destination you want to connect to
Host Name (or IP address): ec2-user@ec2-18-196-126-163.eu-centr
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Load, save or delete a stored session
Saved Sessions: ec2
Default Settings
OBIEE-Server-Hadoop
Test/Prod/imdwhdbadm01_172.27.1
Test/Prod/imdwhdbadm02_172.27.1
ec2
ecrm_10_200_199_76
emr_cluster
Close window on exit: Always Never [✓ Only on clean exit]

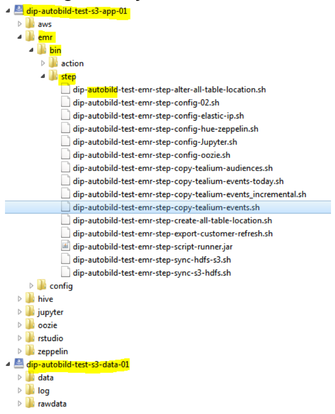


☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)
Authentication methods
☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)
Authentication parameters
☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2
Private key file for authentication:
C:\Apps\Putty\dip-global-test-emr-key.pp k

dip-global-test-emr-key.ppk

Der Management-Server ist so mit Rechten ausgestattet, dass die nachfolgenden Befehle ausgeführt werden können.
aws s3 cp -~~recursiv s3://dip~~welt-test-s3-app-01/ s3://dip-autobild-test-s3-app-01/
aws s3 cp -~~recursiv s3://dip~~xxx-test-s3-data-01/ s3://dip-autobild-test-s3-data-01/
Für die Applikationsdaten wird ein bestehender Bucket (welt) verwendet und für die Struktur des Daten Bucket wird der Template Bucket (xxx) verwendet.

Anschließend müssen die Dateinamen und die Datei-Inhalte im neuen App-Bucket angepasst werden. Kann mit Cyberduck durchgeführt werden.
In den Skriptnamen im Verzeichnis - dip-autobild-test-s3-app-01/emr/bin/step/ stehen noch in den Dateinamen …~~welt~~…(da von diesem vorhandenen Verzeichnis die Dateien herüberkopiert wurden) und „welt" muss durch „autobild" ersetzt werden.
Auszug aus Cyberduck:



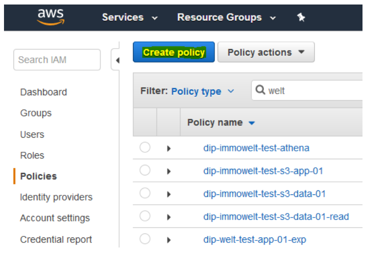
```
dip-autobild-test-s3-app-01
  aws
  emr
    bin
      action
      step
        dip-autobild-test-emr-step-alter-all-table-location.sh
        dip-autobild-test-emr-step-config-02.sh
        dip-autobild-test-emr-step-config-elastic-ip.sh
        dip-autobild-test-emr-step-config-hue-zeppelin.sh
        dip-autobild-test-emr-step-config-Jupyter.sh
        dip-autobild-test-emr-step-config-oozie.sh
        dip-autobild-test-emr-step-copy-tealium-audiences.sh
        dip-autobild-test-emr-step-copy-tealium-events-today.sh
        dip-autobild-test-emr-step-copy-tealium-events_incremental.sh
        dip-autobild-test-emr-step-copy-tealium-events.sh
        dip-autobild-test-emr-step-create-all-table-location.sh
        dip-autobild-test-emr-step-export-customer-refresh.sh
        dip-autobild-test-emr-step-script-runner.jar
        dip-autobild-test-emr-step-sync-hdfs-s3.sh
        dip-autobild-test-emr-step-sync-s3-hdfs.sh
    config
  hive
  jupyter
  oozie
  rstudio
  zeppelin
dip-autobild-test-s3-data-01
  data
  log
  rawdata
```

/emr/config Konfiguration des Clusters
/emr/bin/step Skripte, die beim Starten des Clusters laufen sollen
/emr/action Skripte, um Bootstrap-Aktionen auszuführen
/emr/aws/bin Skript, um die Cluster von der Konsole aus zu starten und zu stoppen
/jupyter/user Ordner auf die aktuellen User anpassen (optional)
/jupyter/config Konfiguration von Jupyterhub (optional)
/zeppelin/config Konfiguration von Zeppelin (optional)
/rstudio/user Ordner auf die aktuellen User anpassen (optional)
/oozie/workflows Workflows des Clusters (optional)
/oozie/dataflows Dataflows des Clusters (optional)
dip-welt-test-emr-step-copy-tealium-events.sh:
Tealium Events werden vom s3 Tealium Bucket abgeholt und im CDP Bucket gespeichert. Die Zugangsdaten für die Tealium s3 Buckets befinden sich im Password Safe und müssen in dem entsprechenden Skript hinterlegt werden (Target/Source). Es müssen die Pfade der Verzeichnisse angepasst werden und in der GroupBy-Option muss beim „s3-dist-cp" Befehl der neue Unitname gesetzt werden. Aus „welt" muss hier „autobild" werden.
Auszug aus Skript dip-welt-test-emr-step-copy-tealium-events.sh:

- ## Verzeichnisse setzen
    - LOCAL_TMPPATH=/tmp/transfer/eventstore/
    - S3_SRC_PATH=s3://dataaccess-eu-central-1.tealiumiq.com/axelspringer/welt/events/all_events/
    - S3_TRANS_PATH=s3://dip-welt-test-s3-data-01/tmp/transfer/tealium/eventstore/
    - S3_DATA_PATH=s3://dip-welt-test-s3-data-01/rawdata/tealium/eventstore/

- s3-dist-cp ~~-src ${S3_TRANS_PATH}${dateStart}/ --dest ${S3_DATA_PATH}file_date=${dateFs}/ --groupBy='./(**axelspringer**weltevents-all_events~~)('${dateStart}').' --targetSize=1024 –deleteOnSuccess

Für jedes Cluster müssen die Zugriffrechte auf die S3-Buckets freigeschaltet werden. Dies kann durch Einrichtung eines eigenen Service-User (IAM) z.B. dip-autobild-test-emr-srv ohne Zugriff über die Konsole geschehen.

- Create Policy 

Search IAM

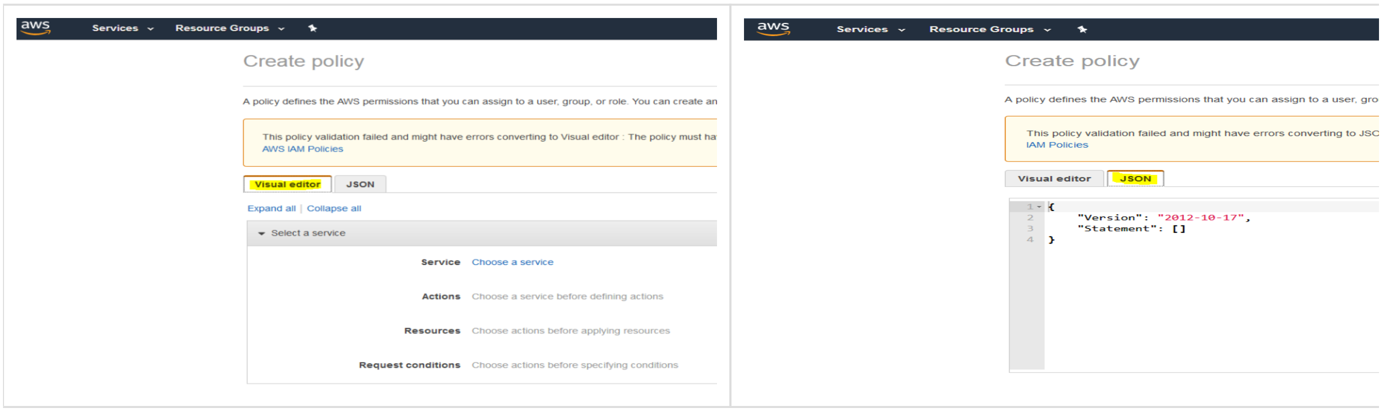
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report

Create policy | Policy actions

Filter: Policy type | welt

Policy name:
- dip-immowelt-test-athena
- dip-immowelt-test-s3-app-01
- dip-immowelt-test-s3-data-01
- dip-immowelt-test-s3-data-01-read
- dip-welt-test-app-01-exp

Die Zugriffsrechte können mit einem bereits vorhandenen json-Skript direkt in den json-Reiter kopiert werden oder über den Visual editor-Reiter werden die entsprechenden Services ausgewählt.



Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create an

This policy validation failed and might have errors converting to Visual editor : The policy must ha
AWS IAM Policies

Visual editor | JSON

Expand all | Collapse all

Select a service

Service	Choose a service
Actions	Choose a service before defining actions
Resources	Choose actions before applying resources
Request conditions	Choose actions before specifying conditions

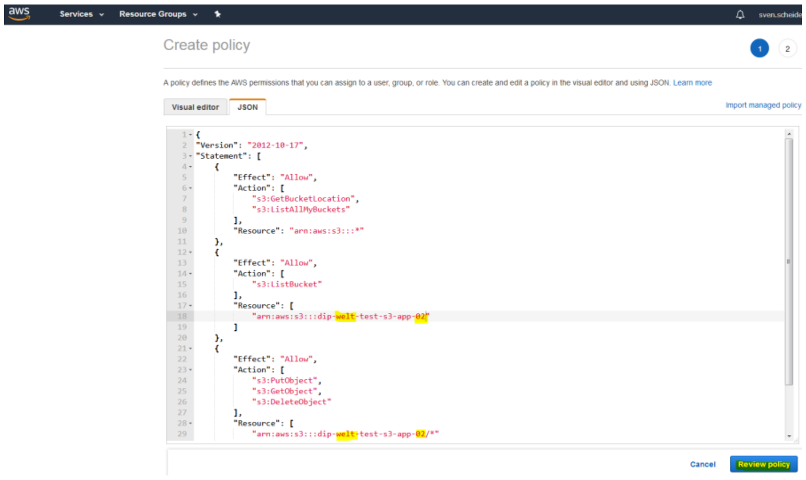
Create policy

A policy defines the AWS permissions that you can assign to a user, gro

This policy validation failed and might have errors converting to JSC
IAM Policies

Visual editor | JSON

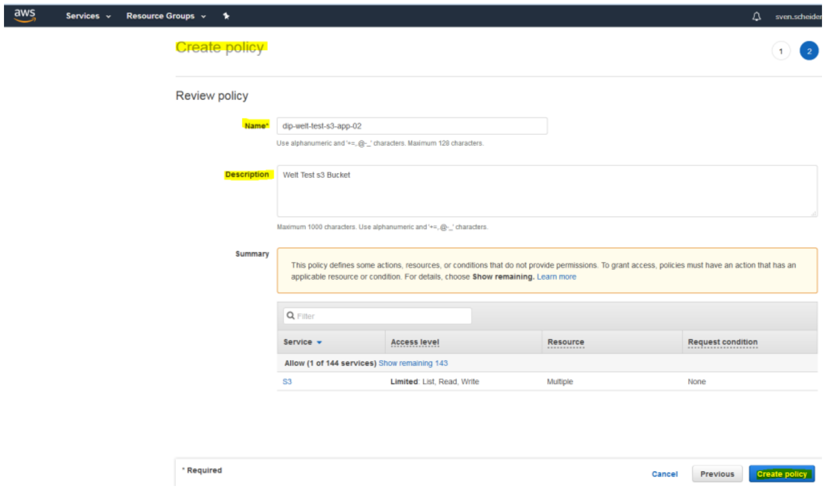
```
{
    "Version": "2012-10-17",
    "Statement": []
}
```



Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketLocation",
            "s3:ListAllMyBuckets"
        ],
        "Resource": "arn:aws:s3:::*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::dip-welt-test-s3-app-02"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject"
        ],
        "Resource": [
            "arn:aws:s3:::dip-welt-test-s3-app-02/*"
```



Create policy

Review policy

Name* dip-welt-test-s3-app-02

Use alphanumeric and '+=,.@-_' characters. Maximum 128 characters.

Description Welt Test s3 Bucket

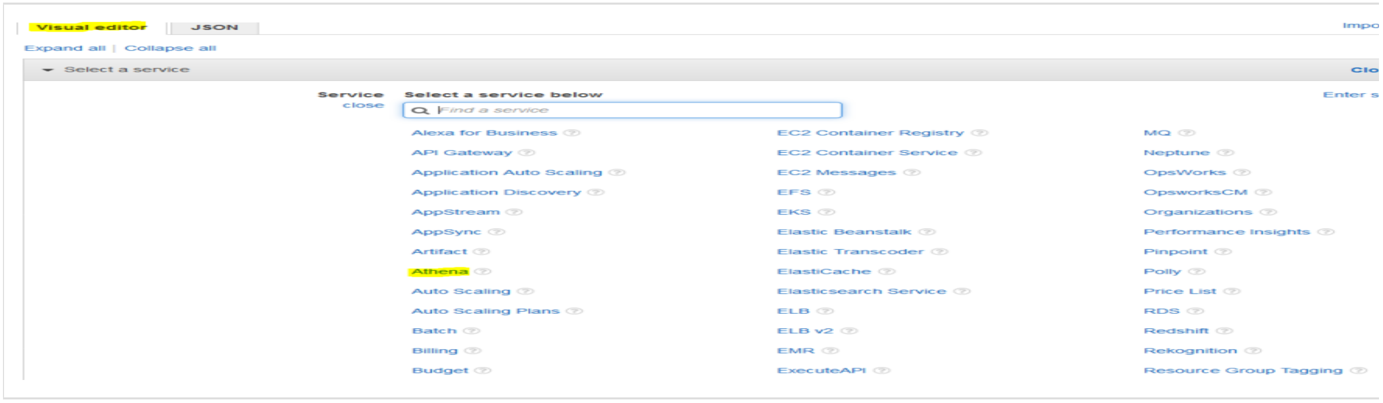
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Summary

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining.** Learn more

Service	Access level	Resource	Request condition
Allow (1 of 144 services) Show remaining 143			
S3	**Limited**: List, Read, Write	Multiple	None

* Required



- s3 – App – Zugriff Enable AWS Management Console access to an Amazon S3 bucket (dip-autobild-test-s3-app-01) – (PolicyName=dip-autobild-test-s3-app-01)

{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"s3:GetBucketLocation",
"s3:ListAllMyBuckets"
],
"Resource": "arn:aws:s3:::*"

```
},
{
"Effect": "Allow",
"Action": [
"s3:ListBucket"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-app-01"
]
},
{
"Effect": "Allow",
"Action": [
"s3:PutObject",
"s3:GetObject",
"s3:DeleteObject"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-app-01/*"
]
}
]
}
```

- s3 – Data – Zugriff Enable AWS Management Console access to an Amazon S3 bucket (dip-autobild-test-s3-data-01) - (PolicyName=dip-autobild-test-data-01-exp) {

```
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"s3:GetBucketLocation",
"s3:ListAllMyBuckets"
],
"Resource": "arn:aws:s3:::*"
},
{
"Effect": "Allow",
"Action": [
"s3:ListBucket"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01"
]
},
{
"Effect": "Allow",
"Action": [
"s3:PutObject",
"s3:GetObject",
"s3:DeleteObject"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01/*"
]
}
]
}
```

- Athena / Glue – Zugriff Autobild Rechte um Athena und das entsprechende Bucket zu lesen und Staging Area zu schreiben - (PolicyName=dip-autobild-test-app-01-exp)

```
{
"Version": "2012-10-17",
"Statement": [
{
```

```
"Effect": "Allow",
"Action": [
"athena:BatchGetQueryExecution",
"athena:CancelQueryExecution",
"athena:GetCatalogs",
"athena:GetExecutionEngine",
"athena:GetExecutionEngines",
"athena:GetNamespace",
"athena:GetNamespaces",
"athena:GetQueryExecution",
"athena:GetQueryExecutions",
"athena:GetQueryResults",
"athena:GetTable",
"athena:GetTables",
"athena:ListQueryExecutions",
"athena:RunQuery",
"athena:StartQueryExecution",
"athena:StopQueryExecution"
],
"Resource": [
"*"
]
},
{
"Effect": "Allow",
"Action": [
"glue:GetDatabase",
"glue:GetDatabases",
"glue:GetTable",
"glue:GetTables",
"glue:GetPartition",
"glue:GetPartitions",
"glue:BatchGetPartition"
],
"Resource": [
"*"
]
},
{
"Effect": "Allow",
"Action": [
"s3:GetBucketLocation",
"s3:GetObject",
"s3:ListBucket",
"s3:ListBucketMultipartUploads",
"s3:ListMultipartUploadParts",
"s3:AbortMultipartUpload",
"s3:PutObject"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01*"
]
},
{
"Effect": "Allow",
"Action": [
"s3:GetObject"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01/athena/*"
]
}
]
}
```

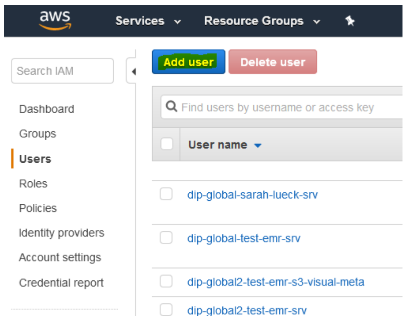
- jupyter – Zugriff Allow access to jupyter user hadoop - (PolicyName=dip-autobild-test-app-01-exp)

```
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "AllowRootAndHomeListingOfCompanyBucket",
"Action": [
"s3:ListBucket"
],
"Effect": "Allow",
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-app-01"
],
"Condition": {
"StringEquals": {
"s3:prefix": [
"",
"jupyter/user/hadoop/"
],
"s3:delimiter": [
"/"
]
}
}
},
{
"Sid": "AllowListingOfUserFolder",
"Action": [
"s3:ListBucket"
],
"Effect": "Allow",
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-app-01"
],
"Condition": {
"StringLike": {
"s3:prefix": [
"jupyter/user/hadoop/*"
]
}
}
},
{
"Sid": "AllowAllS3ActionsInUserFolder",
"Effect": "Allow",
"Action": [
"s3:*"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-app-01/jupyter/user/hadoop/*"
]
}
]
}
```

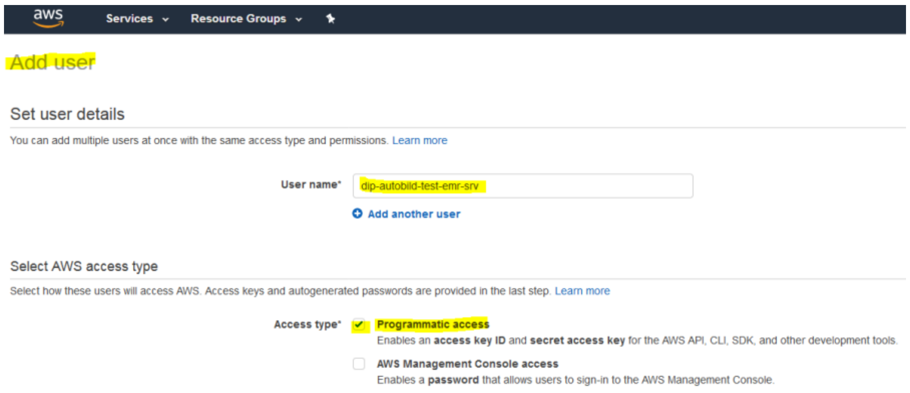
- lesender Zugriff auf das Autobild-Bucket (PolicyName=dip-autobild-test-s3-data-01-read)

```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"s3:GetBucketLocation",
"s3:ListAllMyBuckets"
],
"Resource": "arn:aws:s3:::*"
},
{
"Effect": "Allow",
"Action": [
"s3:ListBucket"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01"
]
},
{
"Effect": "Allow",
"Action": [
"s3:GetObject"
],
"Resource": [
"arn:aws:s3:::dip-autobild-test-s3-data-01/*"
]
}
]
}
```

- Add User



dip-global-sarah-lueck-srv
dip-global-test-emr-srv
dip-global2-test-emr-s3-visual-meta
dip-global2-test-emr-srv



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* dip-autobild-test-emr-srv

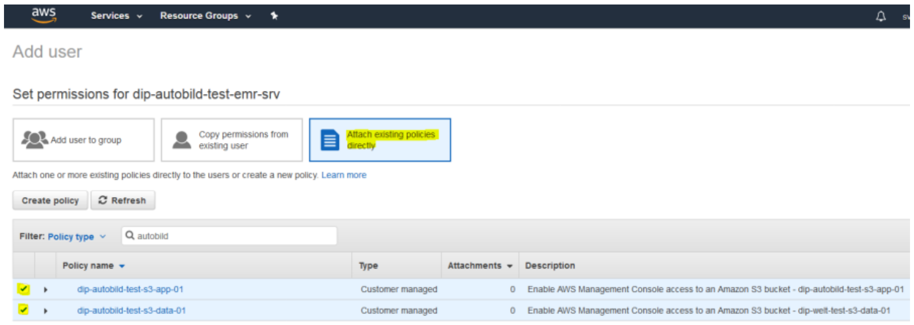
Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* [✓ Programmatic access]
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ AWS Management Console access
Enables a **password** that allows users to sign-in to the AWS Management Console.

- User name: dip-autobild-test-emr-srv
- Access type: Programmatic access (ohne Zugriff über die Konsole – (https://aws.amazon.com/de/ Mein Konto AWS Management Konsole))


aws
Services
Resource Groups
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Refresh

Add user

Set permissions for dip-autobild-test-emr-srv

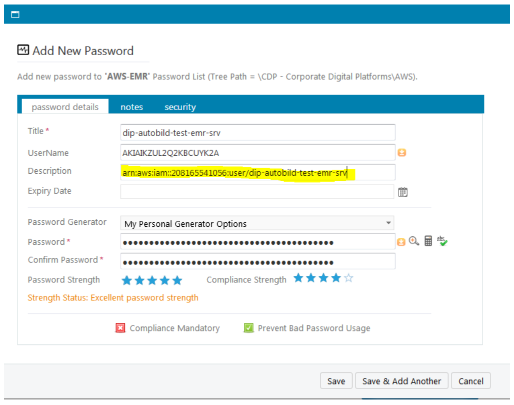
Attach one or more existing policies directly to the users or create a new policy. Learn more

Filter: Policy type autobild

	Policy name	Type	Attachments	Description
✓	dip-autobild-test-s3-app-01	Customer managed	0	Enable AWS Management Console access to an Amazon S3 bucket - dip-autobild-test-s3-app-01
✓	dip-autobild-test-s3-data-01	Customer managed	0	Enable AWS Management Console access to an Amazon S3 bucket - dip-welt-test-s3-data-01

Nach dem man diese oben gezeigte Seite abgeschickt hat, werden einem nochmal die angelegten Infos gezeigt und eine Seite weiter wird ein „Download.csv" bereitgestellt, mit denen die Credentials als Excel heruntergeladen werden MÜSSEN - wenn man diese Infos verliert hat man ein Problem!!!
Die Access Key Id und der Secret Key Id werden dann noch im „AS Passwort Safe" hinterlegt.

- AS Passwort Safe: https://passwords.asv.local/


Add New Password

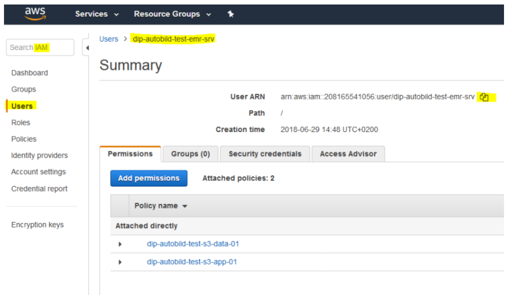
Add new password to '**AWS-EMR**' Password List (Tree Path = \CDP - Corporate Digital Platforms\AWS).

Title *	dip-autobild-test-emr-srv
UserName	AKIAIKZUL2Q2KBCUYK2A
Description	arn:aws:iam::208165541056:user/dip-autobild-test-emr-srv
Expiry Date	
Password Generator	My Personal Generator Options
Password *	••
Confirm Password *	••
Password Strength	★★★★★
Compliance Strength	★★★★☆

Strength Status: Excellent password strength

Compliance Mandatory
Prevent Bad Password Usage

- der gelb markierte Eintrag ist der interne Name, der

zu finden ist unter IAM – Users:


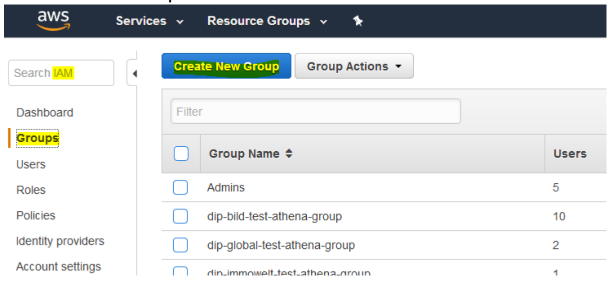
Summary

User ARN	arn:aws:iam::208165541056:user/dip-autobild-test-emr-srv
Path	/
Creation time	2018-06-29 14:48 UTC+0200

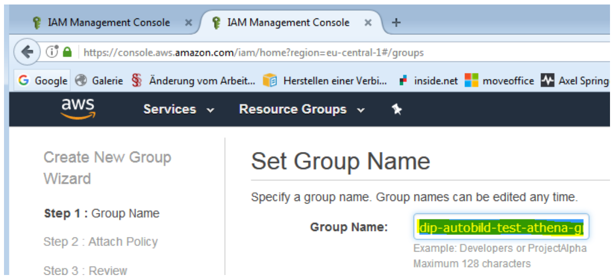
Attached policies: 2

Policy name
Attached directly
dip-autobild-test-s3-data-01
dip-autobild-test-s3-app-01

Eine andere Möglichkeit ist eine neue EMR-Rolle für jede Unit von EMR_DefaultRole abzuleiten und dieser Rolle die S3 Rechte zu geben.
Für beide Fälle müssen vorher die Policies angelegt werden.
Zurzeit müssen jedoch wenigstens der IAM User angelegt werden und wenn möglich auch die Rolle.
Als erstes wird der Cluster für die Analyse durch Duplizieren angelegt und auf die neuen Skripte, Namen, Logfiles etc. angepasst.

Create New Group:


Group Name	Users
Admins	5
dip-bild-test-athena-group	10
dip-global-test-athena-group	2
dip-immowelt-test-athena-group	1



Set Group Name

Specify a group name. Group names can be edited any time.

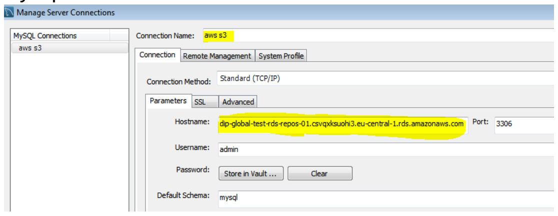
Group Name: dip-autobild-test-athena-g

Example: Developers or ProjectAlpha
Maximum 128 characters

- dip-autobild-test-athena-group

Als nächstes wird dann der ETL Cluster eingerichtet. Dazu müssen dann auch die externen Metastores für Hue, Hive und Jupyter im der MySQL

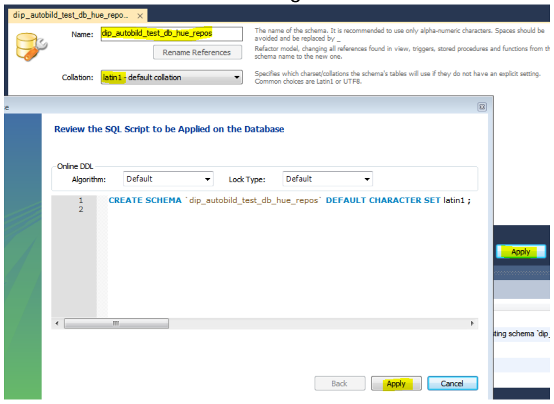
Datenbank angelegt werden.
MySql Workbench öffnen



Connection Name: aws s3
Connection Method: Standard (TCP/IP)
Hostname: dip-global-test-rds-repos-01.csvqxksuohi3.eu-central-1.rds.amazonaws.com Port: 3306
Username: admin
Default Schema: mysql

- dip-global-test-rds-repos-01.csvqxksuohi3.eu-central-1.rds.amazonaws.com

und entweder Schema hinzufügen :

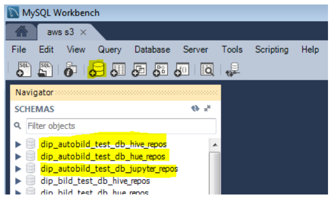


Name: dip_autobild_test_db_hue_repos
The name of the schema. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be replaced by _
Refactor model, changing all references found in view, triggers, stored procedures and functions from th schema name to the new one.
Collation: latin1 - default collation
Specifies which charset/collations the schema's tables will use if they do not have an explicit setting. Common choices are Latin1 or UTF8.

Review the SQL Script to be Applied on the Database

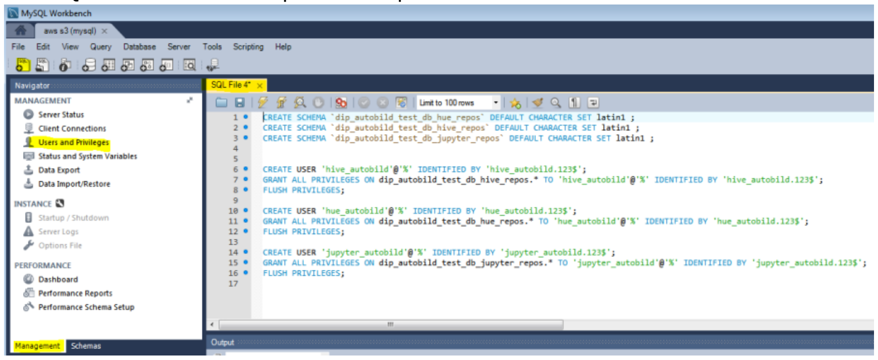
Online DDL
Algorithm: Default Lock Type: Default

```
CREATE SCHEMA `dip_autobild_test_db_hue_repos` DEFAULT CHARACTER SET latin1 ;
```



MySQL Workbench
aws s3
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
dip_autobild_test_db_hive_repos
dip_autobild_test_db_hue_repos
dip_autobild_test_db_jupyter_repos
dip_bild_test_db_hive_repos

oder SQL File öffnen und Skripte hineinkopieren und ausführen.



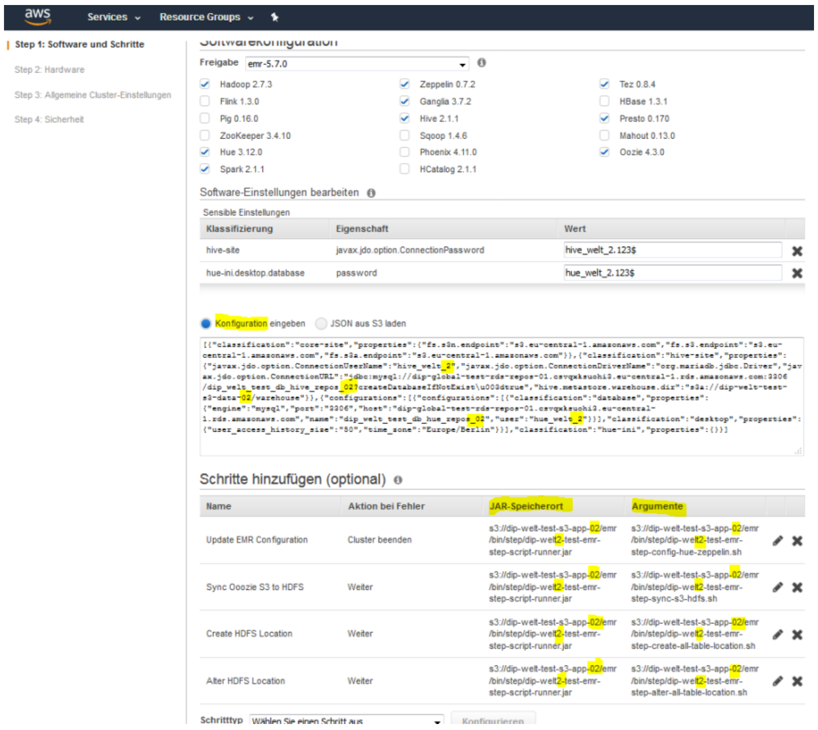
```
CREATE SCHEMA `dip_autobild_test_db_hue_repos` DEFAULT CHARACTER SET latin1 ;
CREATE SCHEMA `dip_autobild_test_db_hive_repos` DEFAULT CHARACTER SET latin1 ;
CREATE SCHEMA `dip_autobild_test_db_jupyter_repos` DEFAULT CHARACTER SET latin1 ;


CREATE USER 'hive_autobild'@'%' IDENTIFIED BY 'hive_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_hive_repos.* TO 'hive_autobild'@'%' IDENTIFIED BY 'hive_autobild.123$';
FLUSH PRIVILEGES;

CREATE USER 'hue_autobild'@'%' IDENTIFIED BY 'hue_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_hue_repos.* TO 'hue_autobild'@'%' IDENTIFIED BY 'hue_autobild.123$';
FLUSH PRIVILEGES;

CREATE USER 'jupyter_autobild'@'%' IDENTIFIED BY 'jupyter_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_jupyter_repos.* TO 'jupyter_autobild'@'%' IDENTIFIED BY 'jupyter_autobild.123$';
FLUSH PRIVILEGES;
```

```
CREATE SCHEMA `dip_autobild_test_db_hue_repos` DEFAULT CHARACTER SET latin1 ;
CREATE SCHEMA `dip_autobild_test_db_hive_repos` DEFAULT CHARACTER SET latin1 ;
CREATE SCHEMA `dip_autobild_test_db_jupyter_repos` /*!40100 DEFAULT CHARACTER SET latin1 */;
CREATE USER 'hive_autobild'@'%' IDENTIFIED BY 'hive_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_hive_repos.* TO 'hive_autobild'@'%' IDENTIFIED BY 'hive_autobild.123$';
FLUSH PRIVILEGES;
CREATE USER 'hue_autobild'@'%' IDENTIFIED BY 'hue_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_hue_repos.* TO 'hue_autobild'@'%' IDENTIFIED BY 'hue_autobild.123$';
FLUSH PRIVILEGES;
CREATE USER 'jupyter_autobild'@'%' IDENTIFIED BY 'jupyter_autobild.123$';
GRANT ALL PRIVILEGES ON dip_autobild_test_db_jupyter_repos.* TO 'jupyter_autobild'@'%' IDENTIFIED BY 'jupyter_autobild.123$';
FLUSH PRIVILEGES;
```

Die Tabellenstruktur von Hive und Hue wird automatisch beim Erstellen des ersten Clusters angelegt. Die von Jupyter muss manuell angelegt werden. Dazu als jupyter_autobild User einloggen und das Dump im Ordner von Sourcetree (Dump20170926) importieren.
Die noch zu bearbeiten Skripte ergeben sich aus den Steps. Nachdem alle angepasst sind, wird der Cluster gestartet.
Wenn er läuft, meldet man sich als Hue und dem Standard Passwort an. Im Home Ordner findet man alle Skripte die nun auf das neue Profil angepasst und ausgeführt werden müssen.
Danach müssen die Datenbanken noch manuell angelegt werden.

Danach einen EMR-Cluster auswählen und klonen.

- Anpassungen (gelb markiert) unter der Konfiguration machen
- sämtliche Schritte löschen, bis auf die 4 in der Abbildung und den JAR-Speicherort und die Verzeichnisse unter Argumenteauch anpassen



Freigabe: emr-5.7.0

✓ Hadoop 2.7.3	✓ Zeppelin 0.7.2	✓ Tez 0.8.4
Flink 1.3.0	✓ Ganglia 3.7.2	HBase 1.3.1
Pig 0.16.0	✓ Hive 2.1.1	✓ Presto 0.170
ZooKeeper 3.4.10	Sqoop 1.4.6	Mahout 0.13.0
✓ Hue 3.12.0	Phoenix 4.11.0	✓ Oozie 4.3.0
✓ Spark 2.1.1	HCatalog 2.1.1	

Software-Einstellungen bearbeiten

Sensible Einstellungen

Klassifizierung	Eigenschaft	Wert
hive-site	javax.jdo.option.ConnectionPassword	hive_welt_2.123$
hue-ini.desktop.database	password	hue_welt_2.123$

[● Konfiguration eingeben] JSON aus S3 laden

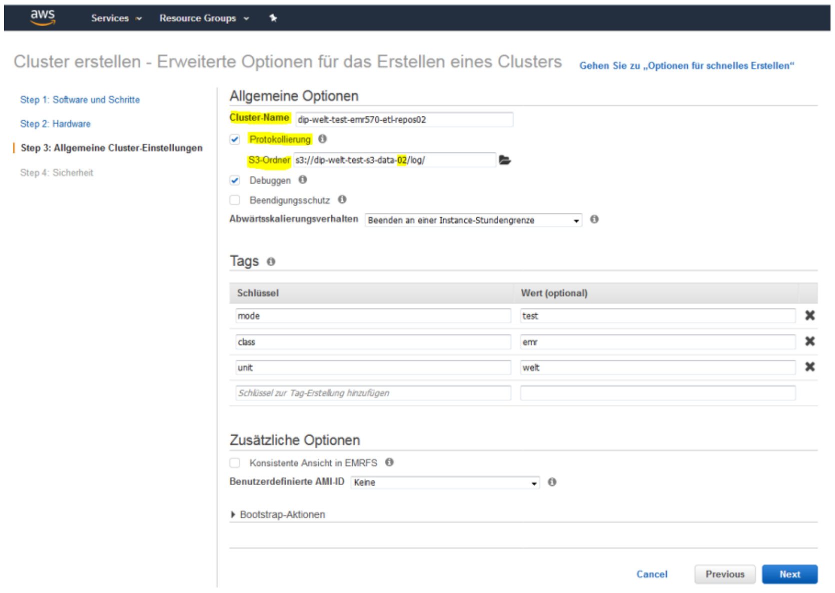
```
[{"classification":"core-site","properties":{"fs.s3n.endpoint":"s3.eu-central-1.amazonaws.com","fs.s3.endpoint":"s3.eu-
central-1.amazonaws.com","fs.s3a.endpoint":"s3.eu-central-1.amazonaws.com"}},{"classification":"hive-site","properties":
{"javax.jdo.option.ConnectionUserName":"hive_welt_2","javax.jdo.option.ConnectionDriverName":"org.mariadb.jdbc.Driver","jav
ax.jdo.option.ConnectionURL":"jdbc:mysql://dip-global-test-rds-repos-01.csvqxksuohi3.eu-central-1.rds.amazonaws.com:3306
/dip_welt_test_db_hive_repos_02?createDatabaseIfNotExist\u003dtrue","hive.metastore.warehouse.dir":"s3a://dip-welt-test-
s3-data-02/warehouse"}},{"configurations":[{"configurations":[{"classification":"database","properties":
{"engine":"mysql","port":"3306","host":"dip-global-test-rds-repos-01.csvqxksuohi3.eu-central-
1.rds.amazonaws.com","name":"dip_welt_test_db_hue_repos_02","user":"hue_welt_2"}}],"classification":"desktop","properties":
{"user_access_history_size":"50","time_zone":"Europe/Berlin"}}],"classification":"hue-ini","properties":{}}]
```

Schritte hinzufügen (optional)

Name	Aktion bei Fehler	JAR-Speicherort	Argumente
Update EMR Configuration	Cluster beenden	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-script-runner.jar	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-config-hue-zeppelin.sh
Sync Ooozie S3 to HDFS	Weiter	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-script-runner.jar	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-sync-s3-hdfs.sh
Create HDFS Location	Weiter	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-script-runner.jar	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-create-all-table-location.sh
Alter HDFS Location	Weiter	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-script-runner.jar	s3://dip-welt-test-s3-app-02/emr/bin/step/dip-welt2-test-emr-step-alter-all-table-location.sh

Schritttyp: Wählen Sie einen Schritt aus

- den Cluster-Name anpassen
- und ggf. den Protokollierung-s3-Ordner


aws
Services
Resource Groups
Gehen Sie zu „Optionen für schnelles Erstellen“
Step 1: Software und Schritte
Step 2: Hardware
Step 3: Allgemeine Cluster-Einstellungen
Step 4: Sicherheit

Cluster erstellen - Erweiterte Optionen für das Erstellen eines Clusters

Allgemeine Optionen

Cluster-Name: dip-welt-test-emr570-etl-repos02

✓ Protokollierung

S3-Ordner: s3://dip-welt-test-s3-data-02/log/

✓ Debuggen

Beendigungsschutz

Abwärtsskalierungsverhalten: Beenden an einer Instance-Stundengrenze

Tags

Schlüssel	Wert (optional)
mode	test
class	emr
unit	welt
Schlüssel zur Tag-Erstellung hinzufügen	

Zusätzliche Optionen

Konsistente Ansicht in EMRFS

Benutzerdefinierte AMI-ID: Keine

Bootstrap-Aktionen

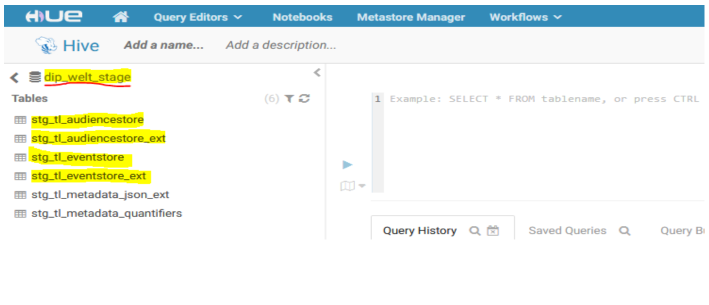
Wenn der Cluster hochgefahren ist, dann müssen noch auf der Hive MySql-Datenbank die Stage-DB und Cleanse-DB, sowie die Core-DB, Mart_01-DB auf dem s3 und die Datenbanktabellen einmal manuell angelegt werden.

- create database dip_welt_stage location '/user/hive/warehouse/dip_welt_stage.db/';
- create database dip_welt_cleanse location '/user/hive/warehouse/dip_welt_cleanse.db/';
- create database dip_welt_core location 's3a://dip-welt-test-s3-data-02/warehouse/dip_welt_core.db/';

- create database dip_welt_mart_01 location 's3a://dip-welt-test-s3-data-02/warehouse/dip_welt_mart_01.db/';

Datenbanktabellen Stand 20180816:

dip_welt_stage:



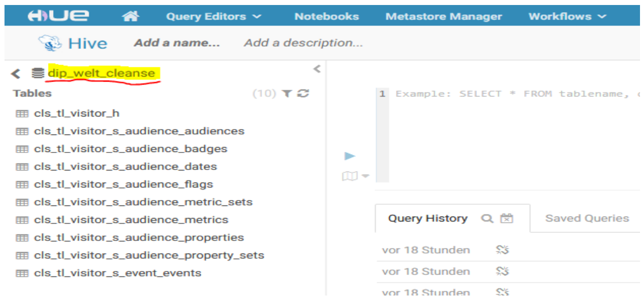
dip_welt_stage

Tables (6)

- stg_tl_audiencestore
- stg_tl_audiencestore_ext
- stg_tl_eventstore
- stg_tl_eventstore_ext
- stg_tl_metadata_json_ext
- stg_tl_metadata_quantifiers

```
1 Example: SELECT * FROM tablename, or press CTRL
```

dip_welt_cleanse:



dip_welt_cleanse

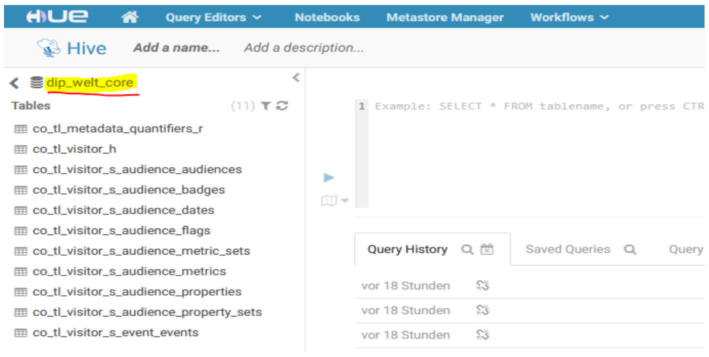
Tables (10)

- cls_tl_visitor_h
- cls_tl_visitor_s_audience_audiences
- cls_tl_visitor_s_audience_badges
- cls_tl_visitor_s_audience_dates
- cls_tl_visitor_s_audience_flags
- cls_tl_visitor_s_audience_metric_sets
- cls_tl_visitor_s_audience_metrics
- cls_tl_visitor_s_audience_properties
- cls_tl_visitor_s_audience_property_sets
- cls_tl_visitor_s_event_events

```
1 Example: SELECT * FROM tablename, 
```

vor 18 Stunden
vor 18 Stunden
vor 18 Stunden

dip_welt_core:



dip_welt_core

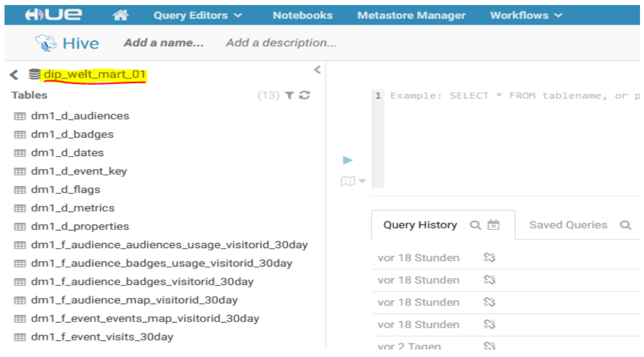
Tables (11)

- co_tl_metadata_quantifiers_r
- co_tl_visitor_h
- co_tl_visitor_s_audience_audiences
- co_tl_visitor_s_audience_badges
- co_tl_visitor_s_audience_dates
- co_tl_visitor_s_audience_flags
- co_tl_visitor_s_audience_metric_sets
- co_tl_visitor_s_audience_metrics
- co_tl_visitor_s_audience_properties
- co_tl_visitor_s_audience_property_sets
- co_tl_visitor_s_event_events

```
1 Example: SELECT * FROM tablename, or press CTR
```

vor 18 Stunden
vor 18 Stunden
vor 18 Stunden

dip_welt_mart_01:



dip_welt_mart_01

Tables (13)

- dm1_d_audiences
- dm1_d_badges
- dm1_d_dates
- dm1_d_event_key
- dm1_d_flags
- dm1_d_metrics
- dm1_d_properties
- dm1_f_audience_audiences_usage_visitorid_30day
- dm1_f_audience_badges_usage_visitorid_30day
- dm1_f_audience_badges_visitorid_30day
- dm1_f_audience_map_visitorid_30day
- dm1_f_event_events_map_visitorid_30day
- dm1_f_event_visits_30day

```
1 Example: SELECT * FROM tablename, or p
```

vor 18 Stunden
vor 18 Stunden
vor 18 Stunden
vor 18 Stunden
vor 2 Tagen

Create-Skripte befinden sich im Anhang.
WORKFLOW:

- danach, wenn in einem anderen Cluster enthalten, kann noch der Workflow exportiert werden und in das neue Cluster importiert werden.

Anhang:

ddl-Skripte:
dip_global_stage: --------------------- INIT SESSION ---------------------- ADD JAR s3://dip-${Unit}~~test-s3-app-01/hive/libs/brickhouse-0.7.1-SNAPSHOT.jar;CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; SET hive.execution.engine=mr; ---------------- DATABASE ------------------ CREATE DATABASE IF NOT EXISTS dip_${Unit}~~~~*stage --COMMENT "Stage Area für die Unit ${Unit}" --LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*stage.db' --WITH DBPROPERTIES(– 'created_date' = '2017-03-07', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_stage; ------------------------ AUDIENCE-STORE ------------------------ CREATE EXTERNAL TABLE `stg_tl_audiencestore_ext`(
`json_string` string)
PARTITIONED BY (
`file_date` date)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/rawdata/tealium/audiencestore'
; DROP TABLE `dip_${Unit}_stage.stg_tl_audiencestore`; CREATE TABLE `stg_tl_audiencestore`(
`visitor_id` string,
`last_visit` string,
`metrics_json` string,
`dates_json` string,
`properties_json` string,
`flags_json` string,
`audiences_json` string,
`badges_json` string,
`preloaded_json` string,
`property_sets_json` string,
`metric_sets_json` string,
`creation_ts_json` string,
`id_json` string,
`partition_json` string,
`shard_token_json` string,
`expire_at_json` string,
`dctrace_json` string,
`audiences_joined_at_json` string,
`file_date` string,
`file_name` string,
`event_date_str` string,
`visit_number` int)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_stage.db/stg_tl_audiencestore'
; --------------------- EVENT-STORE --------------------- DROP TABLE dip_${Unit}_stage.stg_tl_eventstore_ext; CREATE EXTERNAL TABLE `stg_tl_eventstore_ext`(
`json_string` string)
PARTITIONED BY (
`file_date` date)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/rawdata/tealium/eventstore'
;
MSCK REPAIR TABLE dip_${Unit}*stage.stg_tl_eventstore_ext;show partitions dip*${Unit}*stage.stg_tl_eventstore_ext;SELECT * FROM dip*${Unit}*stage.stg_tl_eventstore_ext where file_date = '2018-03-15' LIMIT 100; DROP TABLE dip*${Unit}_stage.stg_tl_eventstore; CREATE TABLE `stg_tl_eventstore`(
`event_id` string,
`visitor_id` string,
`eventtime` string,
`profile` string,
`json_string` string,
`file_date` string,

`file_name` string,
`event_date_str` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_stage.db/stg_tl_eventstore'
;

dip_global_cleanse:
--ROW FORMAT SERDE
– 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
--STORED AS INPUTFORMAT
– 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
--OUTPUTFORMAT
– 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
--------------------- INIT SESSION ---------------------- ADD JAR s3://dip-${Unit}~~test-s3-app-01/hive/libs/brickhouse-0.7.1-SNAPSHOT.jar;CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; SET hive.execution.engine=mr; ---------------- DATABASE ---------------- CREATE DATABASE IF NOT EXISTS dip_${Unit}*cleanse --COMMENT "Stage Area für die Unit ${Unit}" --LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*cleanse.db' --WITH DBPROPERTIES(– 'created_date' = '2017-03-07', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_cleanse; ------------------------ AUDIENCE-STORE ------------------------ DROP TABLE `cls_tl_visitor_h`;
CREATE TABLE `cls_tl_visitor_h`(
`dwh_id` string,
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_bk_visitor_id` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_h'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `cls_tl_visitor_s_audience_audiences`;
CREATE TABLE `cls_tl_visitor_s_audience_audiences`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`audiences_arr` array<string>,
`audiences_arr_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
<ac:structured-macro ac:name="anchor" ac:schema-version="1" ac:macro-id="2dad2597-22be-46a5-af3c-c16421e40180"><ac:parameter ac:name="">_Hlk524000704</ac:parameter></ac:structured-macro>ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_audiences'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true', 'PARQUET.COMPRESS'='SNAPPY')
;
DROP TABLE `cls_tl_visitor_s_audience_badges`;
CREATE TABLE `cls_tl_visitor_s_audience_badges`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`badges_arr` array<string>,
`badges_arr_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_badges'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true', 'PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE cls_tl_visitor_s_audience_dates;
CREATE TABLE `cls_tl_visitor_s_audience_dates`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`dates_map` map<string,double>,
`dates_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_dates'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true', 'PARQUET.COMPRESS'='SNAPPY')
;


DROP TABLE `cls_tl_visitor_s_audience_flags`;
CREATE TABLE `cls_tl_visitor_s_audience_flags`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`flags_map` map<string,string>,
`flags_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_flags'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
DROP TABLE `cls_tl_visitor_s_audience_metric_sets`;
CREATE TABLE `cls_tl_visitor_s_audience_metric_sets`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`metric_sets_map` map<string,map<string,double>>,
`metric_sets_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_metric_sets'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
DROP TABLE `cls_tl_visitor_s_audience_metrics`;
CREATE TABLE `cls_tl_visitor_s_audience_metrics`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`metrics_map` map<string,double>,
`metrics_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_metrics'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `cls_tl_visitor_s_audience_properties`;
CREATE TABLE `cls_tl_visitor_s_audience_properties`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`properties_map` map<string,string>,
`properties_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_properties'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `cls_tl_visitor_s_audience_property_sets`;
CREATE TABLE `cls_tl_visitor_s_audience_property_sets`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`property_sets_map` map<string,array<string>>,
`property_sets_map_cnt` int,
`last_visit` string,
`visit_number` int,
`effectiv_date` string,
`expire_date` string)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.
ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_audience_property_sets'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
---------------------- EVENT-STORE ----------------------
--CREATE TABLE `cls_tl_visitor_s_event_events`(
– `dwh_cr_load_id` string,
– `dwh_cr_job_id` string,
– `dwh_id_hub` string,
– `dwh_load_date` timestamp,
– `dwh_load_src` string,
– `dwh_hash_diff` string,
– `event_id` string,
– `event_time` string,
– `visitor_id` string,
– `visit_id` string,
– `pageurl_full_url` string,
– `referrerurl_full_url` string,
– `event_map` map<string,string>,
– `event_map_cnt` int,
– `effectiv_date` string)
--ROW FORMAT SERDE
– 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
--STORED AS INPUTFORMAT
– 'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
--OUTPUTFORMAT
– 'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
--LOCATION
– '/user/hive/warehouse/dip_global_cleanse.db/cls_tl_visitor_s_event_events';
```

DROP TABLE dip_global_cleanse.cls_tl_visitor_s_event_events;CREATE TABLE dip_global_cleanse.cls_tl_visitor_s_event_eventsSTORED AS ORC TBLPROPERTIES ('ORC.COMPRESS'='SNAPPY')ASSELECT '${LOAD_ID}' AS DWH_CR_LOAD_ID, – Die Lade Id wird im Workflow vergeben'${JOB_ID}' AS DWH_CR_JOB_ID, – Die Job Id wird im Workflow vergebenevent_id AS DWH_ID_HUB , – FK auf den HUB. Noch nicht verfügbarCURRENT_TIMESTAMP AS DWH_LOAD_DATE, – Das LadedatumFILE_NAME AS DWH_LOAD_SRC, – Als Quelle wird hier der Filename hinterlegtcast('' as string) AS DWH_HASH_DIFF, – Array wird in String konvertiert und gehasht. Kann als Änderungskennzeichen verwendet werden.event_id AS event_id , – Eindeutige Event-IdFROM_UNIXTIME(CAST(SUBSTR(eventtime,1,10) AS BIGINT)) AS event_time , – Event-Time asl Timestampvisitor_id AS visitor_id , – Visitor-ID get_json_object(json_string, '$.firstpartycookies_utag_main_ses_id') AS visit_id, get_json_object(json_string, '$.pageurl_full_url') AS pageurl_full_url, get_json_object(json_string, '$.referrerurl_full_url') AS referrerurl_full_url,from_json(json_string,map("", "")) AS event_map, – alle event attribute als mapsize(from_json(json_string,map("", ""))) AS event_map_cnt, – Anzahl der Werte im Map,event_date_str AS EFFECTIV_DATE – Das Datum auf das sich der Inhalt der Datensätze beziehtFROM dip_global_stage.stg_tl_eventstore a limit 100; show create table dip_global_cleanse.cls_tl_visitor_s_event_events;

dip_global_core:

--------------------- INIT SESSION ---------------------- ADD JAR s3://dip-${Unit}~~test-s3-app-01/hive/libs/brickhouse-0.7.1-SNAPSHOT.jar;CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; SET hive.execution.engine=mr; ------------------- DATABASE ------------------- CREATE DATABASE IF NOT EXISTS dip_${Unit}~~~~*core --COMMENT "Stage Area für die Unit ${Unit}" --LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*core.db' --WITH DBPROPERTIES(– 'created_date' = '2017-03-07', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_core; ------------------------- AUDIENCE-STORE -------------------------

```
DROP TABLE co_tl_visitor_h;
CREATE TABLE `co_tl_visitor_h`(
`dwh_id` string,
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_bk_visitor_id` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_h'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `co_tl_visitor_s_audience_audiences`;
CREATE TABLE `co_tl_visitor_s_audience_audiences`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`audiences_arr` array<string>,
`audiences_arr_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_audiences'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `co_tl_visitor_s_audience_badges`;
```

```
CREATE TABLE `co_tl_visitor_s_audience_badges`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`badges_arr` array<string>,
`badges_arr_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_badges'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
DROP TABLE ` co_tl_visitor_s_audience_dates`;
CREATE TABLE `co_tl_visitor_s_audience_dates`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`dates_map` map<string,double>,
`dates_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_dates'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE ` co_tl_visitor_s_audience_flags`;
CREATE TABLE `co_tl_visitor_s_audience_flags`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`flags_map` map<string,string>,
`flags_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_flags'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
```

```
DROP TABLE `co_tl_visitor_s_audience_metric_sets`;
CREATE TABLE `co_tl_visitor_s_audience_metric_sets`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`metric_sets_map` map<string,map<string,double>>,
`metric_sets_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_metric_sets'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `co_tl_visitor_s_audience_metrics`; CREATE TABLE `co_tl_visitor_s_audience_metrics`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`metrics_map` map<string,double>,
`metrics_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_metrics'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE ; CREATE TABLE `co_tl_visitor_s_audience_properties`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`properties_map` map<string,string>,
`properties_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_properties'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
```

```
;

DROP TABLE `co_tl_visitor_s_audience_property_sets`; CREATE TABLE `co_tl_visitor_s_audience_property_sets`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` timestamp,
`dwh_load_src` string,
`dwh_hash_diff` string,
`property_sets_map` map<string,array<string>>,
`property_sets_map_cnt` int,
`last_visit` string,
`visit_number` int,
`expire_date` string)
PARTITIONED BY (
`effectiv_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_audience_property_sets'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

--CREATE TABLE `dip_${Unit}_core.co_tl_visitor_s_event_audience_list`( --`dwh_cr_load_id` string, --`dwh_cr_job_id` string, --`dwh_id_hub`
string, --`dwh_load_date` timestamp, --`dwh_load_src` string, --`dwh_hash_diff` string, --`audience_arr` array<string>, --`audience_arr_cnt` int, --
`event_number` int) --PARTITIONED BY ( – `effectiv_date` string) --ROW FORMAT SERDE – 'org.apache.hadoop.hive.ql.io.parquet.serde.
ParquetHiveSerDe' --STORED AS INPUTFORMAT – 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' --OUTPUTFORMAT –
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
--LOCATION
– 's3a://dip-global-test-s3-data-01/warehouse/dip_global_core.db/co_tl_visitor_s_event_audience_list'
--TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY', 'last_modified_by'='hive'); --CREATE TABLE `dip_${Unit}_core.
co_tl_visitor_s_event_badge_list`( --`dwh_cr_load_id` string, --`dwh_cr_job_id` string, --`dwh_id_hub` string, --`dwh_load_date` timestamp, --
`dwh_load_src` string, --`dwh_hash_diff` string, --`badge_arr` array<string>, --`badge_arr_cnt` int, --`event_number` int) --PARTITIONED BY ( –
`effectiv_date` string) --ROW FORMAT SERDE – 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' --STORED AS
INPUTFORMAT – 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' --OUTPUTFORMAT – 'org.apache.hadoop.hive.ql.io.
parquet.MapredParquetOutputFormat'
--LOCATION
– 's3a://dip-global-test-s3-data-01/warehouse/dip_global_core.db/co_tl_visitor_s_event_badge_list'
--TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY', 'last_modified_by'='hive');




---------------------- EVENT-STORE ---------------------- DROP TABLE `co_tl_visitor_s_event_events`; CREATE TABLE `co_tl_visitor_s_event_events`
(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` string,
`dwh_load_src` string,
`dwh_hash_diff` string,
`event_id` string,
`event_time` string,
`visitor_id` string,
`visit_id` string,
`pageurl_full_url` string,
`referrerurl_full_url` string,
`event_map` map<string,string>,
`event_map_cnt` int)
PARTITIONED BY (
`effectiv_date` date)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
```

```
's3a://dip-global-test-s3-data-02/warehouse/dip_global_core.db/co_tl_visitor_s_event_events'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
```

--------------------- INIT SESSION ----------------------~~-ADD JAR s3://dip~~${Unit}-test-s3-app-01/hive/libs/brickhouse-0.7.1-SNAPSHOT.jar; --CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; --SET hive.execution.engine=mr; ------------------------ AUDIENCE-STORE --------------------------ALTER TABLE dip_global_core.co_tl_visitor_s_audience_audiences ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.co_tl_visitor_s_audience_badges ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.
co_tl_visitor_s_audience_dates ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.co_tl_visitor_s_audience_flags ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.co_tl_visitor_s_audience_metric_sets ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.co_tl_visitor_s_audience_metrics ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.
co_tl_visitor_s_audience_properties ADD COLUMNS (expire_date string); --ALTER TABLE dip_global_core.
co_tl_visitor_s_audience_property_sets ADD COLUMNS (expire_date string);

dip_global_mart_01:

--------------------- INIT SESSION ---------------------- ADD JAR s3://dip-${Unit}~~test-s3-app-01/hive/libs/brickhouse-0.7.1-SNAPSHOT.jar;CREATE TEMPORARY FUNCTION from_json AS 'brickhouse.udf.json.FromJsonUDF'; SET hive.execution.engine=mr; -------------------- DATABSE -------------------- CREATE DATABASE IF NOT EXISTS dip_${Unit}*mart_01 - COMMENT "Stage Area für die Unit ${Unit}" - LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*MART.db' --WITH DBPROPERTIES(– 'created_date' = '2017-03-07', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_mart_01;

```
— AUDIENCE-STORE ----------------------
DROP TABLE `dm1_d_audiences`;
CREATE TABLE `dm1_d_audiences`(
`audience` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_audiences'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_d_badges`;
CREATE TABLE `dm1_d_badges`(
`badge` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_badges'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_d_dates`;
CREATE TABLE `dm1_d_dates`(
`dates` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_dates'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_d_flags`;
CREATE TABLE `dm1_d_flags`(
`flag` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_flags'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
```

```
;


DROP TABLE `dm1_d_metrics`;
CREATE TABLE `dm1_d_metrics`(
`metric` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_metrics'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_d_properties`;
CREATE TABLE `dm1_d_properties`(
`property` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_d_properties'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
DROP TABLE `dm1_f_audience_audiences_visitorid_30day`;
CREATE TABLE `dm1_f_audience_audiences_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`audience` string,
`browser` string,
`operating_system` string,
`last_visit_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_audiences_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_f_audience_badges_visitorid_30day`;
CREATE TABLE `dm1_f_audience_badges_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`badge` string,
`browser` string,
`operating_system` string,
`last_visit_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_badges_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_f_audience_flags_visitorid_30day`;
CREATE TABLE `dm1_f_audience_flags_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`flag` string,
`browser` string,
`operating_system` string,
`last_visit_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_flags_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_f_audience_map_visitorid_30day`;
CREATE TABLE `dm1_f_audience_map_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`last_visit_date` string,
`expire_date` string,
`audiences_map` map<string,string>,
`audiences_map_cnt` int,
`badges_map` map<string,string>,
`badges_map_cnt` int,
`properties_map` map<string,string>,
`properties_map_cnt` int,
`dates_map` map<string,double>,
`dates_map_cnt` int,
`flags_map` map<string,string>,
`flags_map_cnt` int,
`metrics_map` map<string,double>,
`metrics_map_cnt` int)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_map_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_f_audience_metrics_visitorid_30day`;
CREATE TABLE `dm1_f_audience_metrics_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`metric` string,
`browser` string,
`operating_system` string,
`last_visit_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_metrics_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

DROP TABLE `dm1_f_audience_properties_visitorid_30day`;
CREATE TABLE `dm1_f_audience_properties_visitorid_30day`(
`date_str` string,
`visitor_id` string,
`property` string,
`browser` string,
`operating_system` string,
`last_visit_date` string,
`expire_date` string)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
```

```
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-global-test-s3-data-02/warehouse/dip_global_mart_01.db/dm1_f_audience_properties_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;

---------------------- EVENT-STORE ---------------------- DROP TABLE `dm1_f_event_events_map_visitorid_30day`; CREATE TABLE
`dm1_f_event_events_map_visitorid_30day`(
`dwh_cr_load_id` string,
`dwh_cr_job_id` string,
`dwh_id_hub` string,
`dwh_load_date` string,
`dwh_load_src` string,
`dwh_hash_diff` string,
`event_id` string,
`event_time` string,
`visitor_id` string,
`visit_id` string,
`pageurl_full_url` string,
`referrerurl_full_url` string,
`event_map` map<string,string>,
`event_map_cnt` int)
PARTITIONED BY (
`effectiv_date` date)
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
LOCATION
's3a://dip-welt-test-s3-data-02/warehouse/dip_welt_mart_01.db/dm1_f_event_events_map_visitorid_30day'
TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true','PARQUET.COMPRESS'='SNAPPY')
;
```
Export:
~~-CREATE DATABASE IF NOT EXISTS dip_${Unit}*exp COMMENT "Export Area für die Unit ${Unit}" LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*exp.db' --WITH DBPROPERTIES(– 'created_date' = '2017-11-07', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_exp;

```
CREATE TABLE `dip_${Unit}_exp.exp_tl_page_url`( `page_id` string COMMENT 'In dieser Spalte der Satellitentabelle wird der Primärschlüssel der Hubtabelle gespeichert. Damit sind die Hub- und Satellitentabelle über eine 1:n-Beziehung miteinander verknüpft.', `page_url` string COMMENT 'The Url the data was collected with', `page_protocol` string COMMENT 'The Protocol the data was collected with', `page_host` string COMMENT 'The Host the data was collected with', `page_path` string COMMENT 'The Path the data was collected with', `page_query` string COMMENT 'The Query the data was collected with')PARTITIONED BY ( `effectiv_date` string)ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe' STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat' OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'LOCATION 's3a://dip-global-test-s3-data-02/warehouse/dip_visual_meta_stage.db/stg_tl_page_url'TBLPROPERTIES ('COLUMN_STATS_ACCURATE'='true', 'PARQUET.COMPRESS'='SNAPPY');
```
Mining:
------------------- DATABASE -------------------~~CREATE DATABASE IF NOT EXISTS dip_${Unit}*mine COMMENT "Mining Area für die Unit ${Unit}" LOCATION 's3a://dip*~~${Unit}-test-s3-data-01/warehouse/dip${Unit}*mine.db' --WITH DBPROPERTIES(– 'created_date' = '2017-10-02', – 'created_by' = 'Stephan Semmler', – 'Email' = 'stephan.semmler@axelspringer.de' –); --USE dip*${Unit}_mine;

– wird als Datflow eingebaut über Oozie
– CONNECT DATABASE
– Es wird ein generischen DDL-Statement generiert, welches eine Tabelle mit allen Badges als einer Spalte erstellt.
– Diese wird im Verzeichnis Dataflow/Mine abgelegt und durch den Ooozie aufgerufen.
– CONNECT DATABASE
– Es wird ein generischen DDL-Statement generiert, welches eine Tabelle mit allen Badges als einer Spalte erstellt.
– Diese wird im Verzeichnis Dataflow/Mine abgelegt und durch den Ooozie aufgerufen.
```
USE dip_global_mine;
set hive.execution.engine=mr;
INSERT OVERWRITE DIRECTORY '/user/hue/oozie/dataflow/mine/df_mine_audience_badges_visitorid_30day.sql'
SELECT sql
FROM
(
SELECT row_num, sql
FROM mine_audience_badges_visitorid_30day_ddl_v x
ORDER BY row_num
) x;
CREATE OR REPLACE VIEW `mine_audience_badges_visitorid_30day_ddl_v` AS SELECT cast(0 as int) as `row_num`, "USE dip_global_mine;
drop table dip_global_mine.mine_audience_badges_visitorid_30day; create table dip_global_mine.mine_audience_badges_visitorid_30day
STORED AS PARQUET TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY') as select '' as `sql`
union all
```

```
SELECT ROW_NUMBER() OVER () AS `row_num`, concat('if(array_contains(badges_arr, '"', batch_arr_exploded.batch_object,'"'),1,0) AS `',
LOWER(REGEXP_REPLACE(batch_arr_exploded.batch_object, '[^a-zA-Z0-9]+', '_')),'`,') as `sql`
FROM `dip_global_core`.`co_tl_visitor_s_audience_badges`
lateral view explode(`co_tl_visitor_s_audience_badges`.`badges_arr`) `batch_arr_exploded` as `batch_object`
WHERE `co_tl_visitor_s_audience_badges`.`effectiv_date` >= date_sub(CURRENT_DATE,30)
GROUP BY concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `', LOWER(REGEXP_REPLACE
(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,')
union all
SELECT cast(99999 as int) as `row_num`, "visitor_id as `visitor_id` from dip_global_mart_01.dm1_f_audience_badges_arr_visitorid_30day;" as
`sql`
– OLD –
USE dip_bild_mine;
set hive.execution.engine=mr;
INSERT OVERWRITE DIRECTORY '/user/hue/oozie/dataflow/mine/df_mine_audience_badges_visitorid_30day.sql'
SELECT sql
FROM
(
SELECT row_num, sql
FROM
( – SQL Header CREATE AS
(
SELECT cast(0 as int) as `row_num`, "USE dip_bild_mine;drop table dip_bild_mine.mine_audience_badges_visitorid_30day; create table
dip_bild_mine.mine_audience_badges_visitorid_30day STORED AS PARQUET TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY') as
select " as `sql`
)
UNION ALL – SQL Body Listing of each Badge
(
SELECT ROW_NUMBER() OVER () AS `row_num`, concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `',
LOWER(REGEXP_REPLACE(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,') as `sql`
FROM `dip_bild_core`.`co_tl_visitor_s_audience_badges`
lateral view explode(`co_tl_visitor_s_audience_badges`.`badges_arr`) `batch_arr_exploded` as `batch_object`
WHERE `co_tl_visitor_s_audience_badges`.`effectiv_date` >= date_sub(CURRENT_DATE,30)
GROUP BY concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `', LOWER(REGEXP_REPLACE
(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,')
)
UNION ALL --SQL-Footer From-Statement
(
SELECT cast(99999 as int) as `row_num`, "visitor_id as `visitor_id` from dip_bild_mart_01.dm1_f_audience_badges_arr_visitorid_30day;" as `sql`
)
) y
ORDER BY row_num
) x;
– NEBENRECHNUNGEN
USE dip_bild_mine;
set hive.execution.engine=mr;
DROP VIEW dip_bild_mine.`mine_audience_badges_visitorid_30day_ddl_v`;
CREATE VIEW dip_bild_mine.`mine_audience_badges_visitorid_30day_ddl_v`
AS
SELECT cast(0 as int) as `row_num`, "USE dip_bild_mine;drop table dip_bild_mine.mine_audience_badges_visitorid_30day; create table
dip_bild_mine.mine_audience_badges_visitorid_30day STORED AS PARQUET TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY') as
select " as `sql`
union all
SELECT ROW_NUMBER() OVER () AS `row_num`, concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `',
LOWER(REGEXP_REPLACE(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,') as `sql`
FROM `dip_bild_core`.`co_tl_visitor_s_audience_badges`
lateral view explode(`co_tl_visitor_s_audience_badges`.`badges_arr`) `batch_arr_exploded` as `batch_object`
WHERE `co_tl_visitor_s_audience_badges`.`effectiv_date` >= date_sub(CURRENT_DATE,30)
GROUP BY concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `', LOWER(REGEXP_REPLACE
(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,')
union all
SELECT cast(99999 as int) as `row_num`, "visitor_id as `visitor_id` from dip_bild_mart_01.dm1_f_audience_badges_arr_visitorid_30day;" as
`sql`;
– wird als Datflow eingebaut über Oozie
– CONNECT DATABASE
USE dip_bild_mine;
set hive.execution.engine=mr;
DROP TABLE dip_bild_mine.mine_audience_badges_visitorid_30day_ddl;
CREATE TABLE dip_bild_mine.mine_audience_badges_visitorid_30day_ddl
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
AS
SELECT sql FROM (SELECT row_num,sql FROM dip_bild_mine.mine_audience_badges_visitorid_30day_ddl_v order by row_num) x;
SELECT * FROM mine_audience_badges_visitorid_30day_ddl;
INSERT OVERWRITE DIRECTORY '/user/hue/oozie/dataflow/mine/df_mine_audience_badges_visitorid_30day.sql'
```

```
SELECT sql
FROM
(
SELECT cast(0 as int) as `row_num`, "USE dip_bild_mine;drop table dip_bild_mine.mine_audience_badges_visitorid_30day; create table
dip_bild_mine.mine_audience_badges_visitorid_30day STORED AS PARQUET TBLPROPERTIES ('PARQUET.COMPRESS'='SNAPPY') as
select " as `sql`
union all
SELECT ROW_NUMBER() OVER () AS `row_num`, concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `',
LOWER(REGEXP_REPLACE(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,') as `sql`
FROM `dip_bild_core`.`co_tl_visitor_s_audience_badges`
lateral view explode(`co_tl_visitor_s_audience_badges`.`badges_arr`) `batch_arr_exploded` as `batch_object`
WHERE `co_tl_visitor_s_audience_badges`.`effectiv_date` >= date_sub(CURRENT_DATE,30)
GROUP BY concat('if(array_contains(badges_arr, '"', `batch_arr_exploded`.`batch_object`,'"'),1,0) AS `', LOWER(REGEXP_REPLACE
(`batch_arr_exploded`.`batch_object`, '[^a-zA-Z0-9]+', '_')),'`,')
union all
SELECT cast(99999 as int) as `row_num`, "visitor_id as `visitor_id` from dip_bild_mart_01.dm1_f_audience_badges_arr_visitorid_30day;" as `sql`
) x;
– Beispiel DDL
USE dip_bild_mine;
set hive.execution.engine=mr;
drop table dip_bild_mine.mine_audience_badges_visitorid_30day;
create table dip_bild_mine.mine_audience_badges_visitorid_30day STORED AS PARQUET TBLPROPERTIES ('PARQUET.
COMPRESS'='SNAPPY')
as
select
if(array_contains(badges_arr, "visit duration 30 medium"),1,0) AS `visit_duration_30_medium`,
if(array_contains(badges_arr, "visit duration 30 low"),1,0) AS `visit_duration_30_low`,
if(array_contains(badges_arr, "visit duration 30 high"),1,0) AS `visit_duration_30_high`,
if(array_contains(badges_arr, "video plays 30 medium"),1,0) AS `video_plays_30_medium`,
if(array_contains(badges_arr, "video plays 30 low"),1,0) AS `video_plays_30_low`,
if(array_contains(badges_arr, "video plays 30 high"),1,0) AS `video_plays_30_high`,
if(array_contains(badges_arr, "unterhaltung fav channel // no BILDplus"),1,0) AS `unterhaltung_fav_channel_//_no_BILDplus`,
if(array_contains(badges_arr, "social shares 30 medium"),1,0) AS `social_shares_30_medium`,
if(array_contains(badges_arr, "social shares 30 low"),1,0) AS `social_shares_30_low`,
if(array_contains(badges_arr, "social shares 30 high"),1,0) AS `social_shares_30_high`,
if(array_contains(badges_arr, "regional main channel - Berlin with min regio views "),1,0) AS `regional_main_channel_-Berlin_with_min_regio_vie
ws`,
if(array_contains(badges_arr, "ratgeber fav channel // no BILDplus"),1,0) AS `ratgeber_fav_channel_//_no_BILDplus`,
if(array_contains(badges_arr, "iPad Fans"),1,0) AS `iPad_Fans`,
if(array_contains(badges_arr, "Video Fans"),1,0) AS `Video_Fans`,
if(array_contains(badges_arr, "Unterhaltung Fans"),1,0) AS `Unterhaltung_Fans`,
if(array_contains(badges_arr, "Sport Fans"),1,0) AS `Sport_Fans`,
if(array_contains(badges_arr, "Spiele Fans"),1,0) AS `Spiele_Fans`,
if(array_contains(badges_arr, "Rezepte Leser"),1,0) AS `Rezepte_Leser`,
if(array_contains(badges_arr, "Reise Fans"),1,0) AS `Reise_Fans`,
if(array_contains(badges_arr, "Regio Fans"),1,0) AS `Regio_Fans`,
if(array_contains(badges_arr, "Ratgeber Fans"),1,0) AS `Ratgeber_Fans`,
if(array_contains(badges_arr, "Preference for BVB with min. views last week"),1,0) AS `Preference_for_BVB__with_min-_views_last_week`,
if(array_contains(badges_arr, "Politik Fans"),1,0) AS `Politik_Fans`,
if(array_contains(badges_arr, "News Fans"),1,0) AS `News_Fans`,
if(array_contains(badges_arr, "Lotto interest"),1,0) AS `Lotto_interest`,
if(array_contains(badges_arr, "Lifestyle Fans"),1,0) AS `Lifestyle_Fans`,
if(array_contains(badges_arr, "Interest - used cars"),1,0) AS `Interest_-_used_cars`,
if(array_contains(badges_arr, "Interest - conversion page 30 medium"),1,0) AS `Interest_-_conversion_page_30_medium`,
if(array_contains(badges_arr, "Interest - conversion page 30 low"),1,0) AS `Interest_-_conversion_page_30_low`,
if(array_contains(badges_arr, "Interest - conversion page 30 high"),1,0) AS `Interest_-_conversion_page_30_high`,
if(array_contains(badges_arr, "Interest - DIY market"),1,0) AS `Interest_-_DIY_market`,
if(array_contains(badges_arr, "Interest - unterhaltung 30 high"),1,0) AS `Interest_-__unterhaltung_30_high`,
if(array_contains(badges_arr, "Hertha Fan - Buli Tally"),1,0) AS `Hertha_Fan__-_Buli_Tally`,
if(array_contains(badges_arr, "HSV fan"),1,0) AS `HSV_fan`,
if(array_contains(badges_arr, "Gladbach fan"),1,0) AS `Gladbach_fan`,
if(array_contains(badges_arr, "Gladbach Fan - Buli Tally"),1,0) AS `Gladbach_Fan__-_Buli_Tally`,
if(array_contains(badges_arr, "Frequent visitor"),1,0) AS `Frequent_visitor`,
if(array_contains(badges_arr, "Frequency paid 30 medium"),1,0) AS `Frequency_paid_30_medium`,
if(array_contains(badges_arr, "Frequency paid 30 low"),1,0) AS `Frequency_paid_30_low`,
if(array_contains(badges_arr, "Frequency paid 30 high"),1,0) AS `Frequency_paid_30_high`,
if(array_contains(badges_arr, "Frequency 30 very high"),1,0) AS `Frequency_30_very_high`,
if(array_contains(badges_arr, "FC Bayern Munchen fan"),1,0) AS `FC_Bayern_Munchen_fan`,
if(array_contains(badges_arr, "Digital Fans"),1,0) AS `Digital_Fans`,
if(array_contains(badges_arr, "Borussia Dortmund"),1,0) AS `Borussia_Dortmund`,
if(array_contains(badges_arr, "Bayer 04 Leverkusen fan"),1,0) AS `Bayer_04_Leverkusen_fan`,
if(array_contains(badges_arr, "Auto Fans"),1,0) AS `Auto_Fans`,
if(array_contains(badges_arr, "3 conversion pages in 5 days"),1,0) AS `3_conversion_pages_in_5_days`,
if(array_contains(badges_arr, "travel"),1,0) AS `travel`,
if(array_contains(badges_arr, "socio - male"),1,0) AS `socio_-_male`,
if(array_contains(badges_arr, "socio - female"),1,0) AS `socio_-_female`,
if(array_contains(badges_arr, "regional main channel - Ruhrgebiet"),1,0) AS `regional_main_channel_-_Ruhrgebiet`,
if(array_contains(badges_arr, "regional main channel - Muenchen"),1,0) AS `regional_main_channel_-_Muenchen`,
if(array_contains(badges_arr, "regional main channel - Hamburg"),1,0) AS `regional_main_channel_-_Hamburg`,
if(array_contains(badges_arr, "regional main channel - Berlin"),1,0) AS `regional_main_channel_-_Berlin`,
if(array_contains(badges_arr, "digital_itk"),1,0) AS `digital_itk`,
if(array_contains(badges_arr, "channel reise one touch 30"),1,0) AS `channel_reise_one_touch_30`,
if(array_contains(badges_arr, "aida_travel"),1,0) AS `aida_travel`,
if(array_contains(badges_arr, "aida"),1,0) AS `aida`,
if(array_contains(badges_arr, "Wrestling"),1,0) AS `Wrestling`,
if(array_contains(badges_arr, "Wolfsburg Fan - Buli Tally"),1,0) AS `Wolfsburg_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Unbadged"),1,0) AS `Unbadged`,
if(array_contains(badges_arr, "Stuttgart Fan - Buli Tally"),1,0) AS `Stuttgart_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Schalke Fan - Buli Tally"),1,0) AS `Schalke_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Schalke 04 fan"),1,0) AS `Schalke_04_fan`,
if(array_contains(badges_arr, "Mainz Fan - Buli Tally"),1,0) AS `Mainz_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Leverkusen Fan - Buli Tally"),1,0) AS `Leverkusen_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Leipzig Fan - Buli Tally"),1,0) AS `Leipzig_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Koeln Fan - Buli Tally"),1,0) AS `Koeln_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Interest - weather"),1,0) AS `Interest_-_weather`,
if(array_contains(badges_arr, "Interest - video 30 medium"),1,0) AS `Interest_-_video_30_medium`,
if(array_contains(badges_arr, "Interest - video 30 low"),1,0) AS `Interest_-_video_30_low`,
if(array_contains(badges_arr, "Interest - video 30 high"),1,0) AS `Interest_-_video_30_high`,
if(array_contains(badges_arr, "Interest - unterhaltung 30 medium"),1,0) AS `Interest_-_unterhaltung_30_medium`,
if(array_contains(badges_arr, "Interest - unterhaltung 30 low"),1,0) AS `Interest_-_unterhaltung_30_low`,
if(array_contains(badges_arr, "Interest - sport 30 medium"),1,0) AS `Interest_-_sport_30_medium`,
if(array_contains(badges_arr, "Interest - sport 30 low"),1,0) AS `Interest_-_sport_30_low`,
if(array_contains(badges_arr, "Interest - sport 30 high"),1,0) AS `Interest_-_sport_30_high`,
if(array_contains(badges_arr, "Interest - spiele 30 medium"),1,0) AS `Interest_-_spiele_30_medium`,
if(array_contains(badges_arr, "Interest - spiele 30 low"),1,0) AS `Interest_-_spiele_30_low`,
if(array_contains(badges_arr, "Interest - spiele 30 high"),1,0) AS `Interest_-_spiele_30_high`,
if(array_contains(badges_arr, "Interest - reise 30 medium"),1,0) AS `Interest_-_reise_30_medium`,
if(array_contains(badges_arr, "Interest - reise 30 low"),1,0) AS `Interest_-_reise_30_low`,
if(array_contains(badges_arr, "Interest - reise 30 high"),1,0) AS `Interest_-_reise_30_high`,
if(array_contains(badges_arr, "Interest - regional 30 medium"),1,0) AS `Interest_-_regional_30_medium`,
if(array_contains(badges_arr, "Interest - regional 30 low"),1,0) AS `Interest_-_regional_30_low`,
if(array_contains(badges_arr, "Interest - regional 30 high"),1,0) AS `Interest_-_regional_30_high`,
if(array_contains(badges_arr, "Interest - ratgeber 30 medium"),1,0) AS `Interest_-_ratgeber_30_medium`,
if(array_contains(badges_arr, "Interest - ratgeber 30 low"),1,0) AS `Interest_-_ratgeber_30_low`,
if(array_contains(badges_arr, "Interest - ratgeber 30 high"),1,0) AS `Interest_-_ratgeber_30_high`,
if(array_contains(badges_arr, "Interest - politik 30 medium"),1,0) AS `Interest_-_politik_30_medium`,
if(array_contains(badges_arr, "Interest - politik 30 low"),1,0) AS `Interest_-_politik_30_low`,
if(array_contains(badges_arr, "Interest - politik 30 high"),1,0) AS `Interest_-_politik_30_high`,
if(array_contains(badges_arr, "Interest - paid 30 medium"),1,0) AS `Interest_-_paid_30_medium`,
if(array_contains(badges_arr, "Interest - paid 30 low"),1,0) AS `Interest_-_paid_30_low`,
if(array_contains(badges_arr, "Interest - paid 30 high"),1,0) AS `Interest_-_paid_30_high`,
if(array_contains(badges_arr, "Interest - news 30 medium"),1,0) AS `Interest_-_news_30_medium`,
if(array_contains(badges_arr, "Interest - news 30 low"),1,0) AS `Interest_-_news_30_low`,
if(array_contains(badges_arr, "Interest - news 30 high"),1,0) AS `Interest_-_news_30_high`,
if(array_contains(badges_arr, "Interest - lifestyle 30 medium"),1,0) AS `Interest_-_lifestyle_30_medium`,
if(array_contains(badges_arr, "Interest - lifestyle 30 low"),1,0) AS `Interest_-_lifestyle_30_low`,
if(array_contains(badges_arr, "Interest - lifestyle 30 high"),1,0) AS `Interest_-_lifestyle_30_high`,
if(array_contains(badges_arr, "Interest - home 30 medium"),1,0) AS `Interest_-_home_30_medium`,
if(array_contains(badges_arr, "Interest - home 30 low"),1,0) AS `Interest_-_home_30_low`,
if(array_contains(badges_arr, "Interest - home 30 high"),1,0) AS `Interest_-_home_30_high`,
if(array_contains(badges_arr, "Interest - geld 30 medium"),1,0) AS `Interest_-_geld_30_medium`,
if(array_contains(badges_arr, "Interest - geld 30 low"),1,0) AS `Interest_-_geld_30_low`,
if(array_contains(badges_arr, "Interest - geld 30 high"),1,0) AS `Interest_-_geld_30_high`,
if(array_contains(badges_arr, "Interest - digital 30 medium"),1,0) AS `Interest_-_digital_30_medium`,
if(array_contains(badges_arr, "Interest - digital 30 low"),1,0) AS `Interest_-_digital_30_low`,
if(array_contains(badges_arr, "Interest - digital 30 high"),1,0) AS `Interest_-_digital_30_high`,
if(array_contains(badges_arr, "Interest - byou 30 medium"),1,0) AS `Interest_-_byou_30_medium`,
if(array_contains(badges_arr, "Interest - byou 30 low"),1,0) AS `Interest_-_byou_30_low`,
if(array_contains(badges_arr, "Interest - byou 30 high"),1,0) AS `Interest_-_byou_30_high`,
if(array_contains(badges_arr, "Interest - bundesliga 30 medium"),1,0) AS `Interest_-_bundesliga_30_medium`,
if(array_contains(badges_arr, "Interest - bundesliga 30 low"),1,0) AS `Interest_-_bundesliga_30_low`,
if(array_contains(badges_arr, "Interest - bundesliga 30 high"),1,0) AS `Interest_-_bundesliga_30_high`,
if(array_contains(badges_arr, "Interest - auto 30 medium"),1,0) AS `Interest_-_auto_30_medium`,
if(array_contains(badges_arr, "Interest - auto 30 low"),1,0) AS `Interest_-_auto_30_low`,
if(array_contains(badges_arr, "Interest - auto 30 high"),1,0) AS `Interest_-_auto_30_high`,
if(array_contains(badges_arr, "Horoskop"),1,0) AS `Horoskop`,
if(array_contains(badges_arr, "Hoffenheim Fan - Buli Tally"),1,0) AS `Hoffenheim_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Hannover Fan - Buli Tally"),1,0) AS `Hannover_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "HSV Fan - Buli Tally"),1,0) AS `HSV_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Frequency 30 medium"),1,0) AS `Frequency_30_medium`,
if(array_contains(badges_arr, "Frequency 30 low"),1,0) AS `Frequency_30_low`,
if(array_contains(badges_arr, "Frequency 30 high"),1,0) AS `Frequency_30_high`,
if(array_contains(badges_arr, "Freiburg Fan - Buli Tally"),1,0) AS `Freiburg_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Frankfurt Fan - Buli Tally"),1,0) AS `Frankfurt_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Favorite Channel Changer"),1,0) AS `Favorite_Channel_Changer`,
if(array_contains(badges_arr, "Fan"),1,0) AS `Fan`,
if(array_contains(badges_arr, "Dortmund Fan - Buli Tally"),1,0) AS `Dortmund_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Bremen Fan - Buli Tally"),1,0) AS `Bremen_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "Borussia Dortmund fan"),1,0) AS `Borussia_Dortmund_fan`,
if(array_contains(badges_arr, "Bayern Muenchen "),1,0) AS `Bayern_Muenchen_`,
if(array_contains(badges_arr, "Bayern Fan - Buli Tally"),1,0) AS `Bayern_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "BILDplus customer (session)"),1,0) AS `BILDplus_customer__session-`,
if(array_contains(badges_arr, "BILDplus cart abandoner"),1,0) AS `BILDplus_cart_abandoner`,
if(array_contains(badges_arr, "Augsburg Fan - Buli Tally"),1,0) AS `Augsburg_Fan_-_Buli_Tally`,
if(array_contains(badges_arr, "5877"),1,0) AS `5877`,
if(array_contains(badges_arr, "5875"),1,0) AS `5875`,
if(array_contains(badges_arr, "5873"),1,0) AS `5873`,
if(array_contains(badges_arr, "5871"),1,0) AS `5871`,
if(array_contains(badges_arr, "5869"),1,0) AS `5869`,
if(array_contains(badges_arr, "5865"),1,0) AS `5865`,
if(array_contains(badges_arr, "5246"),1,0) AS `5246`,
visitor_id as `visitor_id` from dip_bild_mart_01.dm1_f_audience_badges_arr_visitorid_30day;
```