

Teil 1:
**Erstellung eines Big Data Clusters
& Einführung in die Big Data Verarbeitung**

daniela.nicola@mt-ag.com, 14. Mai 2018

Einleitung

Das Projekt „Data Lake meets KI“ zeigt eine Big Data & KI Musterlösung in der Oracle Cloud. In Teil 1 der Projektbeschreibung zeigen wir Ihnen, wie Sie den Big Data Service in der Oracle Cloud erstellen und dort Daten laden und verarbeiten können.

Was ist der Big Data Service?

Der Big Data Cloud Service ist ein von Oracle gemanagtes Cluster, vorkonfiguriert mit den wichtigsten Komponenten des Apache Hadoop Ökosystems, z.B. Spark, HDFS, Yarn, Tez, Hive, Pig, Ambari, Zeppelin Notebooks. Weitere Komponenten können bei Bedarf hinzugefügt werden.

Anlegen des Big Data Service

1. Loggen Sie sich bei der Oracle Cloud ein. Nutzen Sie Ihr existierendes Konto oder erstellen Sie ein kostenloses Probekonto unter https://cloud.oracle.com/de_DE/tryit
2. Erstellen Sie eine Subscription für die Cloud Storage Container Classic. Optional können Sie einen Testcontainer anlegen.
3. Erstellen Sie eine Instanz des Oracle Big Data Services. Hier einige Tipps für die Konfigurationsparameter:

Create New Instance

← Zurück Abbrechen

Instance **Details** Bestätigung

Weiter →

Service Details

Provide additional configuration parameters for Big Data Cloud instance.

📄 Selection Summary

Cluster Configuration

Legen Sie fest, wie viele Rechner („Knoten“) das Cluster und wie viele virtuelle CPUs jeder Knoten haben soll.

Das kostenlose Cloud Probekonto hat eine Limitierung von 8 OCPUs pro Cluster

Deployment Profile: Full

Number of Nodes: 3

* Compute Shape: OC2m - 2.0 OCPU, 30.0GB R

Queue Profile: Preemption On

Spark Version: 2.1

Cloud Storage Credentials

* Cloud Storage Container: storage-analyticsmtag/container1

* Username: cloudstorageuser

* Password:

Create Cloud Storage Container:

REST Endpoint beim Cloud Storage Service nachgucken. Der Container muss leer sein.

Nutzername darf kein @ enthalten. Unter User Management kann man Nutzernamen ohne Sonderzeichen anlegen.

Credentials

SSH Schlüssel zum Einloggen auf der Shell der Clusterknoten. Kann hier generiert werden.

* SSH Public Key: ssh-rsa AAAAB3NzaC1yc2EAAA

Administrative User: bdcsc_admin

* Password:

* Confirm Password:

Block Storage Settings

Use High Performance Storage:

* Usable HDFS Storage (GB): 50

* Usable BDFS Cache (GB): 50

Total Allocated Storage (GB): 152.5

Associations

Database Cloud Service:

MySQL Cloud Service:

Event Hub Cloud Service:

Praktisch: hier verbinden Sie den Big Data Service mit anderen Cloud Services.

Das war's! In ca. 30 Minuten haben Sie ein Hadoop/Spark Cluster erstellt. So schnell und leicht kann man ein Big Data System im eigenen Rechenzentrum bei weitem nicht erstellen.

Hier ein Beispiellcluster mit 2 Knoten:

ORACLE CLOUD My Services

< bigdataservice2

Big Data Cloud / bigdataservice2

Ab 13.05.2018 03:37 Uf

Überblick

2 Knoten

Administration

2 Patches verfügbar ⓘ

Instance Overview

2 Knoten 4 OCPUs 60 GB Speicher 310 GB Speicherung

Status: Ready Version: 18.1.2-20
 Administrative User: bdcscs_admin Ambari Server Host: 144.2
 Compute Shape: oc2m Deployment Profile: Full
 Spark Thrift Server: jdbc:hive2://bigdataservice2-ana... Spark Version: 2.1

Show more...

Resources

Host Name:	bigdataservice2-bdcscs-1	OCPUs:	2
Öffentliche IP:	144.2	Speicher:	30 GB
Instance:	Runs MASTER-1	Speicherung:	155 GB
Host Name:	bigdataservice2-bdcscs-2	OCPUs:	2
Öffentliche IP:	144.;	Speicher:	30 GB
Instance:	Runs SLAVE-2	Speicherung:	155 GB

Dokumentation

<https://docs.oracle.com/en/cloud/paas/big-data-compute-cloud/csspc/creating-cluster.html>

Was kostet ein Cluster?

Eine Schätzung der Kosten bietet Oracle unter <https://cloud.oracle.com/cost-estimator>. Um Kosten zu sparen, kann das Cluster bei Nichtbenutzung heruntergefahren werden, oder bei geringem Nutzungsgrad herunterskaliert werden (z.B. weniger Knoten).

Datentransfer

Kleinere Dateien können über die Big Data Service Webconsole hochgeladen werden. Für größere Dateien bietet Oracle einen speziellen [File Transfer Manager](#) an. Für das Laden von Daten in Echtzeit kann der Event Hub Service verwendet werden, der auf Apache Kafka basiert.

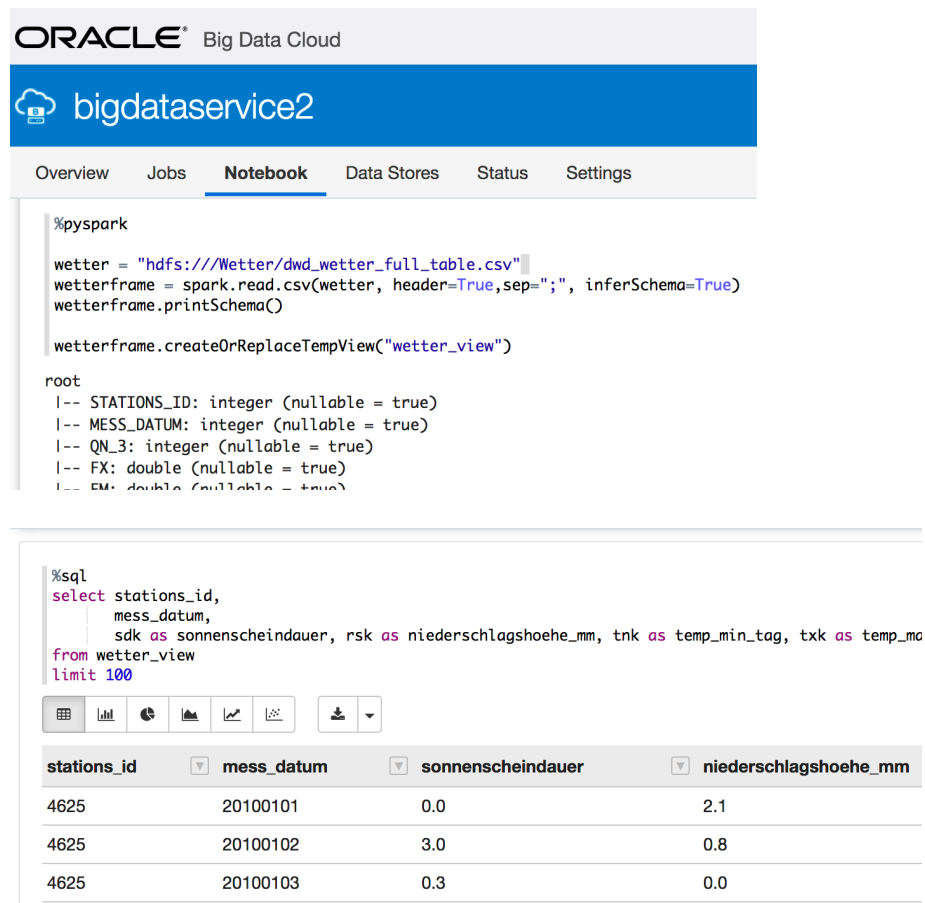
Datenspeicherung

Sie haben die Möglichkeit, Daten lokal auf dem verteilten Dateisystem des Clusters (HDFS) zu speichern. Das hat den Vorteil, dass die Daten direkt auf den Knoten liegen, auf denen sie später per Spark oder MapReduce verarbeitet werden („data locality“). Alternativ kann man Daten im Oracle Cloud Storage ablegen. Letzteres hat den Vorteil, dass die Daten verfügbar sind, auch wenn das Cluster heruntergefahren oder gelöscht wird. Außerdem ist es günstiger als HDFS. Die Performance von Cloud Storage im Vergleich zu HDFS muss von Fall zu Fall untersucht werden. Schnelle Zugriffsmethoden wie das Oracle Big Data Filesystem ([BDFS](#)) sollten dabei berücksichtigt werden.

Datenverarbeitung

Für explorative Transformationen von großen Datenmengen empfehlen wir *Zeppelin Notebooks*. Notebooks sind praktische, browserbasierte Editoren, mit denen man Python, Spark, Scala, SQL und Shell Kommandos ausführen, speichern und dokumentieren kann. Notebooks sind Bestandteil des Big Data Services und benutzen Spark als darunterliegende, hoch-performante Ausführungsumgebung.

In der folgenden Abbildung sieht man, wie man in einem Notebook per Spark SQL Anfragen auf einer Textdatei in HDFS ausführen kann:



The screenshot shows the Oracle Big Data Cloud interface for a Zeppelin Notebook. The top navigation bar includes 'Overview', 'Jobs', 'Notebook', 'Data Stores', 'Status', and 'Settings'. The main content area displays a Python script using %pyspark to read a CSV file from HDFS and create a temporary view. Below the script, the output shows the schema of the view. A second section shows a %sql query being executed, with the results displayed in a table format.

```
%pyspark

wetter = "hdfs://Wetter/dwd_wetter_full_table.csv"
wetterframe = spark.read.csv(wetter, header=True, sep=";", inferSchema=True)
wetterframe.printSchema()

wetterframe.createOrReplaceTempView("wetter_view")

root
 |-- STATIONS_ID: integer (nullable = true)
 |-- MESS_DATUM: integer (nullable = true)
 |-- QN_3: integer (nullable = true)
 |-- FX: double (nullable = true)
 |-- FM: double (nullable = true)
```

```
%sql
select stations_id,
       mess_datum,
       sdk as sonnenscheindauer, rsk as niederschlagshoehe_mm, tnk as temp_min_tag, txk as temp_max_tag
from wetter_view
limit 100
```

stations_id	mess_datum	sonnenscheindauer	niederschlagshoehe_mm
4625	20100101	0.0	2.1
4625	20100102	3.0	0.8
4625	20100103	0.3	0.0

Nächste Schritte

In den folgenden Teilen der Projekt Beschreibung zeigen wir Ihnen:

- ETL in der Cloud: Cleansing und Transformation der Daten im Detail
- KI in der Cloud: Machine Learning Algorithmen auf verteilten Cloud Umgebungen
- BI in der Cloud: Visualisierung & Reporting

Kontakt

Kontaktieren Sie die uns, wenn Sie Fragen haben oder Unterstützung bei einem Projekt im Umfeld von Big Data oder Cloud brauchen.



Balcke-Dürr-Allee 9
40882 Ratingen
+49 2102 30961-0
info@mt-ag.com