Installation Guide

This installation guide explains how to set up Datameer for enterprise and productions environments. If you are upgrading from a previous version, see the Upgrade Instructions.

Following this step-by-step guide also prepares you for later unattended installation, integration into Ansible, Chef, Puppet, or Saltstack, and creating a log of changes. To achieve this, configuration and property changes in files are made using sed.

Prerequisites

Complete the following prerequisites before installing Datameer:

- Install the Hadoop client
    - Datameer application server, as well all data nodes, are configured properly with host names, DNS, datetime, NTP, and other details
        - Datameer application server, as well all data nodes, have Java 1.8 (Oracle recommended)
            - Check this installation using the following commands: `java -version` and `echo $JAVA_HOME`
        - Datameer application server has Oracle Java Cryptography Extension (JCE) already installed. See Java SE Security for more information.
        - Commands such as `hadoop`, `yarn`, and `mysql` can be executed
- Install MySQL client
    - For Datameer's application database, the MySQL server must be prepared with necessary access
- Grant administrative rights or `root` access
- Ensure Internet access to download packages and plug-ins or have necessary ZIP files downloaded and available
- If using Kerberos, configure a Kerberos Secured cluster for secure impersonation

Create the Datameer User

Administrative rights are required to create the Datameer user on the machine where Datameer is being installed. This can be accomplished under the `root` account. Make sure the user ID is above 500 and that the account has enough resources and file descriptors available.

- Create the user account under which the Datameer service will be started and running later:

Create user

```
./usr/sbin/groupadd --system datameer
./usr/sbin/useradd --system --create-home --gid datameer datameer
```

These commands also create the directory `/home/datameer`.

- Check the max number of open files - global level or per-user limits (or both) - and set it to 64K if it isn't set already. This configuration needs to be done on all nodes within the cluster and might require a reboot.

Create Directories for Application, Cache, Logs, and Temporary Files

For performance reasons and to have better control about where space on the file systems and on disks is used, create separate directories for application, cache, logs, and temporary files. Do this according the Linux Filesystem Hierarchy Standard (FHS). To create the directories and change the permissions you need administrative rights. Complete this task under the user account `root`.

- Create the directories for application, cache, logs, and temporary files:

Create directories

```
mkdir -p /opt/datameer
chown -R datameer:datameer /opt/datameer
mkdir -p /var/cache/datameer
chown -R datameer:datameer /var/cache/datameer
mkdir -p /var/log/datameer
chown -R datameer:datameer /var/log/datameer
mkdir -p /tmp/datameer
chown -R datameer:datameer /tmp/datameer
```

Switch the User and Change the Working Directory

This should be the last task to which administrative rights are necessary.

- Switch to the new Datameer user and change to the working directory where Datameer is being installed:

Switch user and directory

```
su - datameer
cd /opt/datameer
```

Proceed from within the Datameer installation directory and under the user account `datameer` only.

Download and Unzip Datameer

Download the appropriate Datameer package for your Hadoop cluster distribution. If you have already a Datameer installation you can also start from here.

- Download and unzip.

Switch user and directory

```
curl -s -k -o Datameer-<package>.zip
"https://download.datameer.com.s3.amazonaws.com/releases/Datameer-<version>/<dist>/Datameer-<package>.zip?<AWSproperties>" ; unzip
Datameer*
```

If you are an authorized Enterprise customer, you can get the download link for the latest public available package from https://my.datameer.com/workspace/downloads, or request one through your Customer Success Manager (CSM).

Best Practice: Create a symlinks and change the working and log directory

To be prepared for future upgrades, create symlinks to the current (or latest) package as well as for the log directory.

- Create symlink and change the working directory:

Create symlink and change working directory

```
ln -s Datameer-<package> current
cd current
```

By default, all Datameer logs are in the installation subdirectory `logs/` . For logs, there is no single property to specify the location, but many depending on the type of log. The main configuration file where you can change the location for most of the log files is `conf/log4j-production.properties`. To keep the change fast and simple, log in a central location according the Linux Filesystem Hierarchy Standard (FHS).

- Move the log directory:

Create symlink to log directory

```
mv logs/.donotdelete /var/log/datameer
rm -rf logs
ln -s /var/log/datameer logs
```

Download and Install the MySQL Database JDBC Connector

By default, the Datameer application runs with an HSQL file database that is created on the local filesystem under `das-data/database/hsql-db`. If you are setting up Datameer for production use, Datameer strongly recommends using MySQL instead of the HSQL file database.
As of Datameer 7.4: MariaDB is supported as an alternative to MySQL.

- Download the official MySQL JDBC driver ZIP file, extract the driver from the archive file, and copy it into the correct destination:

Download and install JDBC

```
# Lookup latest JDBC driver version
JDBCDRV="$(curl -s -k 'https://dev.mysql.com/downloads/connector/j/' |
grep -o -m 1 'mysql-connector-java.*zip')"
# Download latest JDBC driver version
curl -s -L0 -k -O
"https://dev.mysql.com/get/Downloads/Connector-J/${JDBCDRV}"
# Unzip driver package
unzip mysql-connector* -d etc/custom-jars
# Move only the necessary JAR file
mv etc/custom-jars/mysql-connector*/*bin.jar etc/custom-jars
# Clean up
rm -rf etc/custom-jars/mysql-connector-java-?.?.??
```

- Double-check if `etc/custom-jars` contain the latest `mysql-connector-java-<version>-bin.jar:`

Check installation

```
echo $JDBCDRV
ll etc/custom-jars
```

Configure Datameer for MySQL Database

Datameer service depends on the MySQL database. The MySQL database is used for writing to workbooks, permission changes, job execution, scheduling, and more. To function properly, a response time should be between ten and twenty milliseconds. To run the application in MySQL mode, the following changes need to be implemented. **As of Datameer 7.4:** MariaDB is supported as an alternative to MySQL.

- Check database connection:

Connection check

```
mysqladmin version
mysqladmin ping
mysqladmin status
echo q | telnet -e q `hostname` 3306
nc -z -w1 `hostname` 3306
```

You can follow up later with using the Check if the Datameer Application Database is Running and Accessible article.

- Initialize application database:

Initialize database

```
mysql -uroot -p < bin/mysql-init.sql
mysql -uroot -p dap < bin/create-tables.sql
```

- Configure for an enterprise environment:

etc/das-env.sh

```
# Create a backup of the original configuration file
cp etc/das-env.sh etc/das-env.sh.original
# Change the deploy mode
sed -i "s/\(DAS_DEPLOY_MODE=\).*\$/\1live/" etc/das-env.sh
# Uncomment the database name you will be using
sed -i '/#.*DATAMEER_DB_NAME=/s/^#//' etc/das-env.sh
# Uncomment the user that the application should be started at
sed -i '/#.*DAS_USER=/s/^#//' etc/das-env.sh
# Specify the maximum size of the memory allocation pool
sed -i 's/Xmx2048m/Xmx4096m/' etc/das-env.sh
# Create a log of changes made
diff -e etc/das-env.sh.original etc/das-env.sh > changes.das-env.sh
```

If you aren't using `sed`, changes can also be made using other editors.
Example:
Defining the DAS_USER is commented out in *etc/das-env.sh*. To use this property, uncomment the line and add for the correct value.

Installing the License

INFO

If you don't have a license, email the application's product ID to license@datameer.com and request the key. Find the product ID displayed at the 'Welcome' page.

See 'License Information' for information on how to update the license and for details about volume-based licensing.

If you have already received a Datameer license:

1. Launch the Datameer application and open the UI. *The welcome page with all available licensing options is loaded.*
2. Press the button "Activate" and upload the key you received from Datameer. *The license is being activated. You will be redirected to the login page.*

Start Datameer

Start the Datameer service.

- Working within the `current` installation directory, use the following commands:

Start Datameer

```
# Start the Datameer service
./bin/conductor.sh start
# Check the process ID (PID)
ps -ef | grep -i "java.*jetty.*datameer" | grep -v grep | tr -s " " |
cut -d " " -f2
# Monitor the process booting and the log files
cat logs/jvm-stdout.log; sleep 3; tail -F logs/`date
+"%Y_%m_%d"`.stderrout.log logs/conductor.log
```

Stop Datameer

Stop the Datameer service.

- Working within the `current` installation directory, use the following commands:

Start Datameer

```
# Stop the Datameer service
./bin/conductor.sh stop
# Monitor the process shutting down
cat logs/jvm-stdout.log; sleep 3; tail -F logs/`date +"%Y_%m_%d"`.stderrout.log logs/conductor.log
```

Restart Datameer

Restart the Datameer service.

- Working within the `current` installation directory, use the following commands:

Restart Datameer

```
# Restart the Datameer service
./bin/conductor.sh restart
# Monitor the process booting and the log files
cat logs/jvm-stdout.log; sleep 3; tail -F logs/`date +"%Y_%m_%d"`.stderrout.log logs/conductor.log
```

Datameer Graceful Shutdown

Gracefully shut down the Datameer service.

1. Pause the Job Scheduler located under the **Admin** tab in Datameer.
2. Wait for current jobs to be marked as completed.
3. When all jobs have been completed, use the "stop" command on *conductor.sh*
4. After the Datameer application has been stopped, perform needed maintenance.
5. With all maintenance completed, resume Datameer using the "start" command on *conductor.sh*.
6. Under Datameer's **Admin** tab, resume the **Job Scheduler**.

Service Check

Check if the Datameer service is running and accessible.

- Working within the `current` installation directory, use the following commands:

Check service

```
./bin/conductor.sh check
ps -ef | grep -i "java.*datameer" | grep -v grep
lsof -i tcp@`hostname`:8080
lsof -i tcp@`hostname`:8443
lsof -i tcp | grep 'datameer'
echo -e "GET /login \n\n" | openssl s_client -connect `hostname`:8443 -quiet | grep -i -m 1 'datameer'
```

- Monitor if the service is running and accessible later:

Monitor service

```
curl -k "https://`hostname`:8443/watchdog"
lsof -i -p ps -ef | grep -i "java.*jetty.*datameer" | grep -v grep |
tr -s " " | cut -d " " -f2
```

- You can also monitor the Datameer core directory size in HDFS.

Configure Datameer for Kerberos Secured Cluster

Before configuring Datameer for a Kerberos Secured cluster, test Kerberos authentication and job execution on CLI.

- Send a test job to the cluster:

Test job execution

```
hadoop jar
/<distribution-specific-path>/hadoop-mapreduce-examples-*.jar pi
-Dmapreduce.job.queuename=root.default 3 10
```

To configure Datameer for a Kerberos-secured cluster follow the Secure Mode Configuration instructions.

Secure Hadoop Distributed Filesystem (HDFS)

You must have a properly configured connection to a Kerberos-secured cluster to use the tool to secure the Hadoop Distributed Filesystem (HDFS) .

- For initial setup of secure impersonation, execute the following commands:

Secure HDFS

```
# Check current available access rights
hadoop fs -ls /user/datameer
# Configure Datameer Core Directories (aka Private Folder)
./bin/secure_hdfs_tool.sh -u -g <dasuser>
hadoop fs -chown -R datameer:<dasuser> /user/datameer/.staging
hadoop fs -chmod -R 770 /user/datameer/.staging
# Check if changes are made correctly
hadoop fs -ls /user/datameer
```

Start Testing

Start the Datameer service to do final testing.

- Working within the `current` installation directory, use the following commands:

Start Datameer

```
# Start the Datameer service
./bin/conductor.sh start
# Check the process ID (PID)
ps -ef | grep -i "java.*jetty.*datameer" | grep -v grep | tr -s " " |
cut -d " " -f2
# Monitor the process booting and the log files
cat logs/jvm-stdout.log; sleep 3; tail -F logs/`date
+"%Y_%m_%d"`.stderrout.log logs/conductor.log
```

Best Practices for Installing Datameer

Implement frequent database backups

Datameer service depends on the MySQL database, it is used for writing to workbooks, permission changes, job execution, scheduling, and more. It is highly recommend to backup the application database frequently.

Backup via crontab

```
0 * * * * mysqldump -u'dap' -p'dap' dap | gzip >
/home/datameer/<company>_<system>_<datameer-version>_`date
+\%Y\%m\%d_\%H\%M`.sql.gz
```

Don't leave the backup unattended for long time. Monitor the directory `/home/datameer` for its size!

Verify backup

```
# Check from time to tome how long the database dump will take and if it
fits into the timeslot
time mysqldump -u'dap' -p'dap' dap | gzip >
/home/datameer/<company>_<system>_<datameer-version>_`date
+\%Y\%m\%d_\%H\%M`.sql.gz
# Verify from time to time if the files are OK
gzip -d
/home/datameer/company>_<system>_<datameer-version>_<date>_<time>.sql.gz
head /home/datameer/<company>_<system>_<datameer-version>_<date>_<time>.sql
```

Validate the content. Don't leave backup files on the application server. Move backup files from `/home/datameer` to a safe and secure remote location.

Change your stored data directory

Use a path that doesn't depend on a Datameer installation directory. Because the `das-data` folder is stored inside of your installation directory by default, you need to make a backup of your stored data every time you create a new distribution or upgrade.

Learn how to set a different path.

Change the default admin password

Log in and change the default admin password following the instructions on managing user accounts.

Download and install plug-ins

If you are setting up Datameer for production use, it is most likely in a Kerberos Secured environment. To use Kerberos, an additional plug-in is necessary. This Datameer plug-in is part of the Advanced Governance module.

- Download the Kerberos plug-in and install it.

Download and install Kerberos plugin

```
# Look up for corresponding Kerberos plug-in version before
curl -s -k -o plugin-kerberos-<version>.zip
"https://download.datameer.com.s3.amazonaws.com/releases/Datameer-<ve
rsion>/plug-ins_Advanced_Governance/plugin-kerberos-<version>.zip?<AW
Sproperties>" ; mv plugin-kerberos* etc/custom-plugins
```

If you are an authorized Enterprise customer, you can request the download link from your Customer Success Manager (CSM).

Configure Datameer for enterprise

By default, the application runs with settings where files are created on the local filesystem under the `current` directory. To address enterprise requirements, some changes need to be implemented.

> To avoid any mismatch in the configuration files or incompatibility with different versions, don't copy over configuration files from other versions. Make changes every time based on the originally delivered versions.

- **Configure defaults**

conf/default.properties

```
# Create a backup of the original configuration file
cp conf/default.properties conf/default.properties.original
# Move the cache for workbook-previews and dfs
sed -i "s/\(localfs.cache-root=\).*\$/\1\/var\/cache\/datameer/"
conf/default.properties
# Move the temp folder for local-execution
sed -i "s/\(localfs.temporary-files=\).*\$/\1\/tmp\/datameer/"
conf/default.properties
# Provide REST API access by setting failed.login.attempts.max=0
sed -i "s/\(failed.login.attempts.max=\).*\$/\10/"
conf/default.properties
# Switch off tutorial bar
sed -i
"s/\(system.property.integratedTutorial.enabled=\).*\$/\1false/"
conf/default.properties
# Depending on your infrastructure and data set the timezone UTC
sed -i "s/\(system.property.das.default-timezone=\).*\$/\1UTC/"
conf/default.properties
# Create a log of changes made
diff -e conf/default.properties.original conf/default.properties >
changes.default.properties
# Do not expose stack traces to end users
sed -i "s/\(verbose.error.reporting=\).*\$/\1false/"
conf/default.properties
```

- **Configure system properties**

conf/live.properties

```
# Create a backup of the original configuration file
cp conf/live.properties conf/live.properties.original
# Name the address and port used to connect to Datameer UI
# The value given will be used in email notification only
EXT_DM_URL="<hostname>.<domain>.<tld>"
sed -i
"s/\(system.property.server.address=\).*\$/\1https:\/\/${EXT_DM_URL}:
8443/" conf/live.properties
# Comment out temporary file directory since changes were implemented
in default
sed -i '/localfs.temporary-files=/ s/^#*/# /' conf/live.properties
# Create a log of changes made
diff -e conf/live.properties.original conf/live.properties >
changes.live.properties
```

- **Configure UI behavior**

conf/skin-default.properties

```
# Create a backup of the original configuration file
cp conf/skin-default.properties conf/skin-default.properties.original
# Make UI access faster
sed -i '/#.*menu.show-welcome.visibility=/s/^#//'
conf/skin-default.properties
sed -i '/#.*dialog.welcome.visibility=/s/^#//'
conf/skin-default.properties
sed -i '/#.*page.home.visibility=/s/^#//' conf/skin-default.properties
# Provide REST API access by setting force.license-agreement=false
sed -i "s/\(force.license-agreement=\).*\$/\1false/"
conf/skin-default.properties
# Create a log of changes made
diff -e conf/skin-default.properties.original
conf/skin-default.properties > changes.skin-default.properties
```

Review the changes implemented by accessing the change log

- **Review and back up change log**

changes.*

```
more changes.*
cp changes.* /home/datameer
```

- **Back up the installation history**

history

```
history > /home/datameer/install_command.log
cp ~/.bash_history /home/datameer
```

> Validate the changes made. Move files from `/home/datameer` to a safe and secure remote location.

Enable and configure transport layer security (TLS)

> Before the next steps, consider reverse proxies or a load balancer to offload the SSL traffic or to use wild card certificates. In that case, you only need to configure rewrite handling.

Enable TLS for use with Datameer in production environments. As Datameer is packed with Jetty 9, you only need to enable modules.

- Enable TLS:

Enable modules

```
# Check default configuration
java -jar start.jar --list-config | grep -i 'etc/jetty*'
# Add SSL and HTTPS to the startup modules
java -jar start.jar --add-to-start=ssl,https
# Check final configuration
java -jar start.jar --list-config
```

- Redirect all HTTP requests to HTTPS
- Configure TLS for Embedded Jetty (for more security)
- To change the HTTPS port follow the instructions under Configure TLS

  > All port changes should be made in the `start.ini` file, which overrides `jetty.port`.

- Use your own custom certificate

> You can proceed further with Enabling SSL for MySQL service as well.

Configure bash for operations

Set up shell aliases for most common commands to make work easier, faster, and less error prone.

- Working within the `current` Datameer installation directory, add the following aliases:

Create Aliases

```
# Edit your profile file
nano ~/.bash_profile
```

~/.bash_profile

```
# Add aliases
alias dmpid='ps -ef | grep -i "java.*jetty.*datameer" | grep -v grep | tr -s " " | cut -d " " -f2'
alias dmver='ps -ef | grep -i "java.*datameer" | grep -v grep'
alias dmstart='./bin/conductor.sh start'
alias dmstop='./bin/conductor.sh stop'
alias dmcheck='./bin/conductor.sh check'
alias dmkill="kill `dmpid`"
alias dmpath='readlink `pwd`; pwd'
alias dmdap='mysql -udap -pdap dap -Bse'
alias dmsqlping='for ((i=1; i<=5; i++)); do time -p dmdap "START TRANSACTION; INSERT INTO test_entity2 (version) VALUES ('1'); UPDATE test_entity2 SET version = 2 WHERE version = 1; DELETE FROM test_entity2 WHERE version = 2; ROLLBACK;"; sleep 1; done'
alias jettyconf='java -jar start.jar --list-config'
alias classpath='yarn classpath | tr ":" ","'
alias dminit="kinit datameer@<DOMAIN>.<TLD> -k -t /home/datameer/datameer.keytab"
alias dmfs="hadoop fs -du -h /user/datameer"
# Most important alias to set, since this will cover all three phases of Datameer's boot process
alias dmlog='cat logs/jvm-stdout.log; sleep 3; tail -F logs/`date +"%Y_%m_%d"`.stderrout.log logs/conductor.log'
```

Load profile

```
# Load your profile file
source ~/.bash_profile
```

Conductor.sh commands and parameters

Usage: conductor.sh <command> <option>

Commands:

- `start` - Starts the conductor
- `stop` - Stops the conductor
- `restart` - Restarts the conductor
- `check` - Checks if the conductor is already running

Options:

- `--injectExamples` - Injects example import jobs and workbooks on start-up. (This option only works the first time when starting Datameer)
- `--resetPassword` - Resets the admin password to default value.
- `--jobschedulerPaused` - Previously scheduled jobs are paused until re-enabling the job scheduler.
- `--jmx` - Starts JMX management extension for managing and monitoring DAS.
- `--profile` - Runs conductor with attached profiling agent.
- `--profile-sampling` - Runs conductor with cpu profiling (sampling) activated.
- `--profile-tracing` - Runs conductor with cpu profiling (tracing) activated.
- `--profile-memory` - Runs conductor with cpu and memory profiling (sampling) activated.
- `--help` - Opens the help dialog.

Examples:

- How to Monitor Datameer via Java Management Extensions (JMX)
- How to Check if a Datameer Service is Running and Accessible

Where to go from here

If you are a Datameer system administrator, see the Administrator's Guide.