Zenoss Analytics Installation and Administration Guide

Release 5.1.1

Zenoss, Inc.

www.zenoss.com

Zenoss Analytics Installation and Administration Guide

Contents

About this guide

Zenoss Analytics Installation and Administration Guide provides detailed procedures for installing Zenoss Analytics (Analytics), and for administering the product.

Related Resource Manager publications

Title	Description
Zenoss Resource Manager Administration Guide	Provides an overview of Resource Manager architecture and features, as well as procedures and examples to help use the system.
Zenoss Resource Manager Configuration Guide	Provides required and optional configuration procedures for Resource Manager, to prepare your deployment for monitoring in your environment.
Zenoss Resource Manager Installation Guide	Provides detailed information and procedures for creating deployments of Control Center and Resource Manager.
Zenoss Resource Manager Planning Guide	Provides both general and specific information for preparing to deploy Resource Manager.
Zenoss Resource Manager Release Notes	Describes known issues, fixed issues, and late-breaking information not already provided in the published documentation set.
Zenoss Resource Manager Upgrade Guide	Provides detailed information and procedures for upgrading deployments of Resource Manager.

Additional information and comments

If you have technical questions about this product that are not answered in this guide, please visit the *Zenoss Support* site or contact Zenoss Support.

Zenoss welcomes your comments and suggestions regarding our documentation. To share your comments, please send an email to docs@zenoss.com. In the email, include the document title and part number. The part number appears at the end of the list of trademarks, at the front of this guide.

Change history

The following list associates document part numbers and the important changes to this guide since the previous release. Some of the changes involve features or content, but others do not. For information about new or changed features, refer to the *Zenoss Analytics Release Notes*.

1100.18.290.8 (5.1.1)
    Update version information and dates.

1100.17.321 (5.1.0)
    Update installation and upgrade procedures.

Introduction

Zenoss Analytics (Analytics) provides an enterprise ETL capability and data analysis capability for Resource Manager instances, enabling improved business intelligence and knowledge management. Fundamentally, Analytics is three distinct and unrelated capabilities:

- An ETL (extract, transform and load) capability that populates a fully dynamic, data warehousing industry standard, star-schema mySQL-based data warehouse with Resource Manager model, event, and performance data information.
- Data warehouse capabilities that perform user-configurable and recurring analysis on that data.
- An embedded BI/Reporting capability (TIBCO™ Jaspersoft Enterprise Edition, Release 6.0.0) that can be used as one way to access data in the warehouse and a showcase of sample ways to do so to provide monitoring and capacity planning insight in a Resource Manager context. For details of how to use the Jaspersoft reporting capabilities, consult the *TIBCO™ JasperReports® Server User Guide, Release 6.0.0* which can be found with all the *Jaspersoft 6.0.0 documentation*.

Installing Analytics

<div style="text-align: right; font-size: 3em;">2</div>

The topics in this section provide information and procedures to help you install and set up Analytics.

Note The following considerations apply:

- Before you install Analytics, review compatibility and other information in the Analytics release notes.
- If you are upgrading from Analytics version 4.4.0 or 5.0.x, skip the installation instructions in this chapter and proceed to *Upgrading Analytics* on page 21.
- The instructions in this section are written for CentOS/RHEL 7.2, which is the officially supported and tested version. However, field evidence indicates that Analytics functions without issue on CentOS/RHEL 5 and 6. If you cannot deploy CentOS/RHEL 7.2, consult Zenoss Professional Services for assistance. (Using CentOS/RHEL 5 or 6 might not prevent a successful installation; however, it is important to plan to move your Analytics installation to CentOS/RHEL 7.2.)

Preparing to install

Analytics data is stored in a mySQL database. Like any database server, compute speed, memory usage, and disk speed are critical for the Analytics server. The installation is performed on a single server that must have adequate CPU, memory, and disk resources. For Zenoss Resource Manager customers, your Zenoss professional services architect will provide specific sizing for your current needs and that allows for anticipated growth of the data warehouse.

In all scenarios, Zenoss recommends provisioning the Analytics server as a virtual machine (VM) to allow flexible rescaling of CPU and memory resources on an ongoing basis. Provision storage to be extensible also. Best results are achieved when the Analytics server is a stand-alone server that is dedicated to the installation of Analytics.

Other than the operating system (OS) and the external dependencies explicitly listed by this document, run no third-party software on the server. In particular, anti-virus software is known to significantly interfere with the speed of all database systems, and Analytics is no exception. Before starting the installation procedure, audit your OS environment to check for and disable third-party software.

Verifying candidate host resources

The minimum requirements for resources are as follows:

- CentOS/RHEL 7.2
- 50GB of available storage at `/opt/zenoss_analytics`

- 500GB of available storage at `/var/lib/mysql`
- 10GB of available storage at `/tmp`
- 8GB RAM
- 8 CPU cores

To determine whether a host's hardware resources and operating system are sufficient to serve as an Analytics server, complete the following procedures.

Validating the OS environment

1 Log in to the candidate host as `root` or as a user with superuser privileges.
2 Verify that the host implements the 64-bit version of the x86 instruction set:

```
uname -m
```

- If the output is `x86_64`, the architecture is 64-bit. Proceed to the next step.
- If the output is `i386/i486/i586/i686`, the architecture is 32-bit. Stop this procedure and select a different host for the Analytics server.

3 Verify that name resolution works on this host:

```
hostname -i
```

- If the result is not a valid IPv4 address, add an entry for the host to the network nameserver, or to `/etc/hosts`.
- Ensure that the hostname also resolves to the loopback interface locally by editing `/etc/hosts` to include an entry for the hostname of the server after the listing for `127.0.0.1`.

4 Update the operating system to the latest CentOS/RHEL version (currently 7.4), if necessary.
   a Determine which release is installed:

```
cat /etc/redhat-release
```

   b If you are not running the latest version (currently 7.4), before starting the Analytics installation process, update the operating system and then reboot the server.

```
yum update -y
reboot
```

> **Note**   There is an issue with file locking when using NFS 4.1 with CentOS/RHEL 7.4. Updating the OS to version 7.4 requires that you force the use of NFS 4.0 to avoid this issue. For detailed steps, see the following Zenoss KB article: *https://support.zenoss.com/hc/en-us/articles/115005085763-Potential-issues-running-with-RHEL-7-4-or-CentOS-7-4*

5 Verify that UID 1337 is available to Analytics:

```
getent passwd 1337
```

6 Determine whether the `firewalld` service is enabled, and if necessary, disable the firewall.

```
systemctl status firewalld.service
```

- If the result includes `Active: inactive (dead)`, the service is correctly disabled. Proceed with the next step.
- If the result includes `Active: active (running)`, the service is enabled and must be disabled now and on the next boot:

```
systemctl stop firewalld && systemctl disable firewalld
```

7 Determine whether Security-Enhanced Linux (SELinux) is installed and enabled, and if so, disable it:

```
test -f /etc/selinux/config && grep '^SELINUX=' /etc/selinux/config
```

If the command does not return a result, proceed with the next step. If the command returns a result, SELinux is installed. Disable it and reboot the server:

1 Open `/etc/selinux/config` in a text editor and change the value of the `SELINUX` variable to `disabled`.

2 Confirm the new setting:

```
grep '^SELINUX=' /etc/selinux/config
```

3 Reboot the server:

```
reboot
```

8 Install and configure the NTP package:

```
ntpd -gq
```

9 Enable the `ntpd` daemon:

```
systemctl enable ntpd
```

Checking memory and CPU

Determine whether the available memory and swap is sufficient.

The Analytics server requires a minimum of 8GB of available memory and a minimum of 8 cores.

1 Log in to the candidate host as `root` or as a user with superuser privileges.

2 Display the available memory:

```
free -h
```

3 Compare the available memory with the amount required for the Analytics server.

4 Execute `top`, then press `1`. Note the number of CPU cores present.

Provisioning storage

Determine whether the available, unused storage is sufficient and provision it. The required minimums are as follows:

- **50GB for /opt/zenoss_analytics**. This is where the Analytics software is installed, and is also used as temporary storage by the ETL processes.

- **500GB for `/var/lib/mysql`**. This is where mySQL stores its data for both the data analytics database and the data warehouse.
- **10GB for `/tmp`**. This is used as temporary storage by both the Analytics ETL application and mySQL.

1 Log in to the candidate host as `root` or as a user with superuser privileges.
2 Display the available storage devices:

```
lsblk --output=NAME,SIZE
```

3 Compare the available storage with the amount required for an Analytics server, identify an appropriate partition, and then create the appropriate file system.
4 For each type of storage that requires provisioning, identify an appropriate partition, and create and mount the appropriate file system.

   a Identify the target primary partition for the file system to create.

```
lsblk --output=NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT
```

   b Create an EXT4 file system of the partition.

```
mkfs -t ext4 <partition>
```

   c Create the mount point for the file system.

```
mkdir -p <filesystem path>
```

   For example, `mkdir -p /opt/zenoss_analytics`.

   d Add an entry for the file system to the `/etc/fstab` file. Replace `<partition>` with the path of the partition from step b and the `<filesystem path>` from that used in the previous step:

```
echo "<partition> <filesystem path> ext4 defaults 0 0" >> /etc/fstab
```

   e Mount the file system, and then verify that it mounted correctly. Replace the `<file system path>` with the actual path.

```
mount -a && mount | grep <filesystem path>
```

Open network ports

All Resource Manager 4.x collector hosts, a Resource Manager 4.x master node and all Resource Manager 5.x and 6.x hosts (including collector pool hosts) must be able to access the Analytics server HTTP address. Zenoss strongly recommends that, before installing Analytics, you add the FQDN of the Analytics server to your DNS.

The standard SSL port of `443` is used for all Analytics-related communication between Resource Manager hosts and the Analytics server.

Optionally, to enable the use of third-party tools with Analytics, the following TCP ports can be opened on the Analytics server:

- mySQL server port, `3306`, which allows direct mySQL access to the data warehouse and Jaspersoft databases.
- jasperserver port, `7070`, which allows direct connection from Jaspersoft development tools to the Jaspersoft installation.

Installing the OpenJDK

Analytics requires the OpenJDK 7 Development package. Analytics cannot be installed when just the JRE is installed.

Prior to installing the OpenJDK 7 Development package, you should remove all other versions of Java that are on the system:

```
yum -y remove $(rpm -qa | egrep -i '(jdk|jre|java)')
```

1 Log in to the Analytics server as `root`, or as a user with superuser privileges.
2 Install the OpenJDK 7.

```
yum -y install java-1.7.0-openjdk-devel
```

3 Verify that the installation succeeded.

```
java -version
```

If the command returns output similar to the following, continue to the next procedure. The `<_version>` patch version does not need to match

```
java version "1.7.0_75"
OpenJDK Runtime Environment (rhel-2.5.4.7.el7_1.x86_64 u75-b13)
OpenJDK 64-Bit Server VM (build 24.75-b04, mixed mode)
```

Setting the JAVA_HOME environment variable

After installing the OpenJDK, you must set `$JAVA_HOME` for the root user.

1 Log in to the Analytics server as `root`, or as a user with superuser privileges and check that the variable is not currently set:

```
echo $JAVA_HOME
```

2 Locate and note the path to your OpenJDK directory. Typically, this is `/usr/lib/jvm/java-1.7.0-openjdk-<version>`.

```
ls -l /usr/lib/jvm/java-1.7.0-openjdk-<version>
```

3 Navigate to the `/etc/default` directory.

```
cd /etc/default
```

4 Open a new `zenoss_analytics` file in an editor.

```
vi zenoss_analytics
```

5 Edit the `JAVA_HOME` line in `zenoss_analytics` to point to your OpenJDK location and save the file. The resulting entry will look like this:

```
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-<version>
```

6 Source this file:

```
source zenoss_analytics
```

7 Test the new setting of the JAVA_HOME variable:

```
echo $JAVA_HOME
```

Installing MariaDB

MariaDB is used as the mySQL database software for Analytics databases. Version 10.0.x is the currently supported version.

Make sure there is no existing installation of mysql server in place. As the root user, execute the following:

```
rpm -qa | grep -i mysql-server
```

If one exists, remove it before proceeding.

If it isn't already installed, install the Perl Database Interface Module (DBI) with the following command as the root user:

```
yum -y install perl-DBI
```

1 Create a custom MariaDB version 10.0.x YUM repository. The following example shows information for a CentOS 7 installation. For more information, see *https://downloads.mariadb.org/mariadb/repositories*

a As the root user, navigate to the /etc/yum.repos.d directory.

```
cd /etc/yum.repos.d
```

b Edit a file with vi called MariaDB.repo and cut and paste the following text into the file.

```
# MariaDB 10.0 CentOS repository list
#http://mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.0/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

2 When the MariaDB.repo file is in place, install the MariaDB server:

```
yum -y install mariadb-server
```

Note   If you have not accepted the MariaDB GPG key, you will be prompted to do so.

3 Add the following configuration settings:

```
echo "[mysqld]" >> /etc/my.cnf
echo "table_open_cache=16K" >> /etc/my.cnf
echo "table_definition_cache=16K" >> /etc/my.cnf
echo "tmp_table_size=2G" >> /etc/my.cnf
echo "max_heap_table_size=2G" >> /etc/my.cnf
```

```
echo "join_buffer_size=512K" >> /etc/my.cnf
echo "open_files_limit=200000" >> /etc/my.cnf
echo "tmpdir=/tmp" >> /etc/my.cnf
echo "wait_timeout=86400" >> /etc/my.cnf
echo "innodb_adaptive_hash_index=OFF" >> /etc/my.cnf
echo "innodb_buffer_pool_size=6G" >> /etc/my.cnf
echo "innodb_log_file_size=1892M" >> /etc/my.cnf
echo "innodb_log_buffer_size=128M" >> /etc/my.cnf
```

Note   The values for the innodb settings should reflect your system. Edit the file in `vi` and change the default innodb_buffer_pool_size of 16G to 50% of the total memory of the server and set innodb_log_file_size default size of 1892M to 25% of the innodb_buffer_pool_size value.

4   Start the MariaDB mysql server for the first time:

```
/etc/init.d/mysql start
```

You can confirm success by connecting to the database server as follows:

```
mysql -u root
```

If mysql fails to start, it is highly likely you have a `/etc/my.cnf` configuration file misconfiguration or environmental preparation was not completed correctly. You can check the `/var/lib/mysql/<server fqdn>.err` file for details of why startup was not successful. You should correct any issues and confirm successful server start before proceeding further with the installation.

5   Set MariaDB to start on boot:

```
systemctl enable mysql
```

6   Optionally, if you wish to enable remote access to the database server (that is, you have opened port 3306), you can grant remote access to the `root` and `reporting_read` users (Analytics installation will create the later as a read-only user otherwise.)

```
mysql -u root
grant all privileges on *.* to 'root'@'%' with grant option;
grant all privileges on *.* to 'reporting'@'%' with grant option;
flush privileges;
```

Installing the Analytics server

Install the Analytics server by using an `.rpm` file on a separate server.

1   To install the Analytics ETL and embedded Jaspersoft software, enter the following command, as the `root` user, based on the RPM version that you are installing:

```
rpm -ivh zenoss_analytics-<version>.noarch.rpm
```

Replace `<version>` with the version number you want to install; for example, `5.1.0-1`.

2   Change to the zenoss user and execute the initial database creation procedure. This procedure creates the jaspersoft database (zenoss_analytics), populates it with Zenoss examples and creates a blank data warehouse database (reporting):

```
su - zenoss
/opt/zenoss_analytics/bin/upgrade_db.py
```

If this process ends in errors, an environmental issue exists. Typically, the environmental procedure was not followed. In the first instance, make sure localhost resolves to `127.0.0.1` and that you do not have files in `/tmp` left over from a previous installation attempt. Correct issues and then rerun the above process as the `zenoss` user.

3   After this process completes, change to the `root` user and enable Analytics to start on boot and then start the service.

```
exit
systemctl enable zenoss_analytics
systemctl start zenoss_analytics
```

4   Tail the Analytics application log (this is a Tomcat log) and watch the log as Analytics starts up for the first time:

```
tail -F /opt/zenoss_analytics/logs/catalina.out
```

5   On first startup, the data warehouse schema is adjusted and populated with configuration information that is appropriate to the current version of Analytics. This process usually takes about 5 minutes, during which the Analytics application is unavailable. Wait for the following message in this log and then continue with the next step.

```
INFO: Server successfully started in <time in ms>
```

6   If this first startup ends in errors, an environmental issue exists. Typically, the environmental procedure was not followed. In the first instance, make sure you have the correct version of mariadb installed. Version 10.0.x is required.

```
rpm -qa | grep -i maria
```

Installing the Zenoss provided user defined percentile function to MariaDB

The mysql SQL windowing functions are rather limited. To enable a sql database function for calculating percentiles (used by the Analytics percentile aggregation capabilities), Zenoss has written a mysql User Defined Function (UDF) and this needs to be installed into mysql. To install this function, log in to the Analytics server as root, or as a user with superuser privileges and execute the following procedure to install this function and restart the sql server (required by installation of any UDF):

1   Log in as the `root` user or a user with superuser permissions and execute the following commands:

```
su - zenoss
/opt/zenoss_analytics/bin/setup_zenoss_extensions install
exit
service zenoss_analytics stop
/etc/init.d/mysql restart
service zenoss_analytics start
```

Securing Analytics to use SSL

Analytics uses Apache and `mod_ssl` to provide SSL for all of the different types of communication needed. It is assumed that if you are running Resource Manager 4.x you already have secured the Resource Manager

behind SSL (this is in place as a matter of course in Resource Manager 5.x and 6.x). This is a required prerequisite for securing Analytics.

The following procedure will install Apache and `mod_ssl` and set it to use a self-signed SSL certificate. You may also choose to purchase a SSL certificate signed by a third-party Certificate Authority or to generate your own SSL certificate.

Log in to the Analytics server as the `root` user, or as a user with superuser privileges. Install Apache and `mod_ssl`, configure Apache to start on server boot and start it for the first time:

```
yum -y install httpd
yum -y install mod_ssl
systemctl enable httpd
systemctl start httpd
```

You can check this was successful by visiting both `http://<analytics server fqdn>/` and `https://<analytics server fqdn/` in a web browser.

To support potential use of Internet Explorer 8 (IE8), Apache must be configured to strip out the "Pragma" statements from the headers of HTTP files. To do this, navigate to the following Apache configuration folder and edit the config file as follows:

```
cd /etc/httpd/conf
# Backup the existing config file
cp httpd.conf original_httpd.conf_original
# Edit the file
vi httpd.conf
# Add the following line right at the top of the file.
Header unset Pragma
```

Save the file and exit the editor.

Next, we configure SSL to add an internal proxy rule for Apache to proxy any request to the Analytics server and to turn on the Rewrite Engine. Navigate to the Apache SSL configuration folder and edit the SSL config file as follows:

```
cd /etc/httpd/conf.d
# Backup the existing config file
cp ssl.conf original_ssl.conf_original
# Edit the file
vi ssl.conf
```

The last line of the file should be the closing tag `</VirtualHost>`. Add the following text just above this closing `</VirtualHost>` tag:

```
#Internal proxy rules instructing Apache to proxy any request to the
#Analytics server and data warehouse on 7070
ProxyPass /reports http://127.0.0.1:7070/reports
ProxyPassReverse /reports http://127.0.0.1:7070/reports
ProxyPass /etl http://127.0.0.1:7070/etl
ProxyPassReverse /etl http://127.0.0.1:7070/etl
#Turn on the RewriteEngine
RewriteEngine On
#Redirect any just / over to /reports
RewriteRule ^/+$ https://%{SERVER_NAME}:443/reports/ [R]
```

Save and close the `ssl.conf` file and then restart Apache.

```
systemctl restart httpd
```

Next we lockdown tomcat to localhost only so that the Analytics server will not respond to requests on its internal port (7070). An alternate solution is to simply close port 7070 altogether via firewall configuration. Note that if you are intending to use 3rd party tools with Jaspersoft you should NOT lockdown this port or make this server level config change.

Log in to the Analytics server as the `root` user, or as a user with superuser privileges and navigate to the server configuration file and edit it as follows:

```
cd /opt/zenoss_analytics/conf
# Make a backup of the server.xml file.
cp server.xml original_server.xml_original
# Edit the file
vi server.xml
```

Locate the following section in the file (`/7070` in vi will locate it).

```
<Connector port="7070" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"/>
```

Change it to add in `address="127.0.0.1"` so that the section looks like the following:

```
<Connector port="7070" address="127.0.0.1" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"/>
```

Save and close the file and restart tomcat by restarting the service to pick up the changes.

```
service zenoss_analytics stop
service zenoss_analytics start
```

This completes the Analytics Server installation. Proceed with the next section to install extraction daemon services in Resource Manager and to connect the Analytics server into your Resource Manager deployments.

Installing the Analytics ZenETL ZenPack in Resource Manager

Analytics-related UI in Resource Manager, and ETL data extraction daemon services are provided by an Analytics-specific Resource Manager ZenPack, `ZenETL`. This is installed like any other ZenPack, and you should refer to the instructions specific to your version of Resource Manager for doing so if you are not already familiar with the process. In summary the process is the following:

In Resource Manager 4.x:

```
zenpack --install ZenPacks.zenoss.ZenETL<version>.py2.7.egg
```

If you are using a `daemons.txt`/`DAEMONS_TX_ONLY` and/or a `collectordaemons.txt` to control which daemons start on the master and collectors respectively, edit these files as appropriate add the new ETL daemons to these files. Add the following to `daemons.txt`:

- `zenmodeletl`
- `zeneventetl`
- `zenperfetl`

Add the following to `collectordaemons.txt`:

- `zenperfetl`

Then perform a full stop and start of Resource Manager:

```
zenoss stop
pkill -f ${ZENHOME}
zenoss start
```

At this point, you should NOT update all hubs and collectors just yet to push the new ZenPack code out. Complete the section below entitled *Connecting Analytics server to Resource Manager* on page 16 first and then update all hubs and collectors.

In Resource Manager 5.x and 6.x, place `ZenPacks.zenoss.ZenETL<version>.py2.7.egg` in the `/tmp/zenpacks` directory on the Control Center master host, then perform the following:

```
chmod -R 777 /tmp/zenpacks
cd /tmp/zenpacks
serviced service run zope zenpack-manager install
 ZenPacks.zenoss.ZenETL<version>.py2.7.egg
```

To speed up performance data extractions in Resource Manager 5.2.x or later (or any Resource Manager instance that has OpenTSDB v2.2 or later installed), the following change should be made. The configuration option, `tsd.query.skip_unresolved_tagvs` should be set to `True`. This option can be set in the `/opt/zenoss/etc/opentsdb/opentsdb.conf` file that is editable in the "opentsdb reader" service in Control Center.

Finally, issue the following commands to stop and start Resource Manager so that `zenperfetl`, `zenmodeletl`, and `zeneventetl` initialize properly:

```
serviced service stop Zenoss.resmgr
# Wait for containers to stop
serviced service start Zenoss.resmgr
```

Connecting Analytics server to Resource Manager

To connect Resource Manager to the completed Analytics server installation is accomplished through the Resource Manager user interface. Log in to Resource Manager as an admin or "Manager" user.

Note   For Resource Manager 5.x and 6.x only: Performance data extraction is performed by the Analytics server by calls to the Zenoss QueryService API. As such, a privileged user account dedicated to this purpose should be used. Before proceeding with the next paragraph, create such an account with a "Manager" role in Resource Manager and note the username and password. Suggested username is "analytics_etl". Note that the built in "admin" account cannot be used for this purpose.

Navigate to **Reports > Configuration** in the Resource Manager UI. The Analytics configuration page appears. Click the gear icon in the lower-left corner to display the Configure Analytics Base URLs page.

In the Internal URLs area, enter the URLs that Resource Manager services will use to communicate with Analytics. Note that if the two Resource Manager URLs are not displayed, you may need to shift-refresh your browser to drop the UI cache post-ZenETL installation. Do not proceed with configuration on this page unless both Resource Manager URLs are displayed correctly.

Paste the fully qualified domain name (including SSL port number 443) for the Analytics server into both the "Internal Analytics" and "External Analytics" URL fields:

```
https://<analytics_server_fqdn>:443
```

Note   For Resource Manager 5.x and 6.x only: In the QueryService area, enter the complete URL of the Resource Manager virtual host (that is, the URL for your Resource Manager UI) and the access credentials to the QueryService of this Resource Manager instance created above.

Click **Submit** to save the configuration. A success flare should confirm that configuration information was written to both Resource Manager and Analytics servers. If this process fails, make sure your DNS is valid in your client browser and on the Resource Manager master server (5.x and 6.x) or within all Zope containers (5.x and 6.x) and that environmental set up on the Analytics server was completed appropriately (that is, the firewall is disabled and appropriate ports are open).

Once this configuration has been successfully saved, the `zenmodeletl`, `zeneventetl`, and the `zenperfetl` service (all found in the Analytics ETL group) should be restarted in the Control Center UI so that the extractor services pick up the newly saved configuration for the Analytics server.

Note   For Resource Manager 4.x only: At this point, you should update all hubs and collectors to push the new ZenPack code out. This can be achieved via the **Advanced > Collectors** screen in Resource Manager, or from the command line using `dc-admin`.

Fixing syntax errors in aliases

Perform this step after successfully installing or upgrading Analytics.

In the LinuxMonitor ZenPack, there are syntax errors that need to be manually fixed in order for Analytics to perform data extraction. Perform the following procedure:

1 Log in to the Resource Manager master as `root` or a user with superuser permissions and execute the following command:

```
serviced service attach zope/0
```

2 Change to the `zenoss` user and navigate to the indicated folder:

```
su - zenoss
cd /opt/zenoss/ZenPacks/ZenPacks.zenoss.ZenETL-<Version>-py2.7.egg/
ZenPacks/zenoss/ZenETL/bin
```

where `<Version>` is the version of the ZenETL ZenPack that was installed.

3 Execute the following commands to run the `dumpAliases` script, make the necessary changes, and commit the changes:

```
./dumpAliases.py --aliases=with

egrep "(mem_buffers__pct|mem_cached__pct|mem_free__pct|
mem_swap_free__pct|lvm_pv_free__pct|lvm_vg_free__pct)" aliases.txt >
 aliases_changes.txt
```

```
sed -i 's/,\*,100/,100,\*/' aliases_changes.txt

./manageAliases.py --action=add --inputFile=aliases_changes.txt --
commit

exit
exit
```

4   Execute the following commands to stop the `zenperfetl` service, remove the disk cache that does not get cleared on restart, and then start the `zenperfetl` service.

```
serviced service stop zenperfetl

rm -rf /opt/serviced/var/volumes/<tenantid>/etl-analytics/zenperfetl/
perf-config-cache

serviced service start zenperfetl
```

5   Wait until the new PERFORMANCE extractor has registered with Analytics. You can check this by an ssh connection to the Analytics server and then opening the mysql console and entering the following command.

```
select * from meta_extractor where extractor_name = 'PERFORMANCE';
```

You will know the extractor has registered properly when the previous command returns only one row.

6   Get the key of the new extractor by executing the following command:

```
select extractor_key into @extractor_key from meta_extractor where
 extractor_name = 'PERFORMANCE';
```

7   Update all existing performance batches to have the new key:

```
update meta_batch set extractor_key = @extractor_key where
 extractor_key != @extractor_key and extractor_key in (select
 extractor_key from meta_extractor where extractor_type =
 'PERFORMANCE');
```

8   Remove all the old performance extractors:

```
delete from meta_extractor where extractor_name like '%/_%';
```

Verifying installation success and correct ETL operation

Installation success should be immediately verified as follows. If any of the verification steps fail, refer to the Analytics configuration documentation for specific troubleshooting procedures:

1   Create a user account (or update an existing one) with BOTH the "Manager" and "ReportingUser" roles in Resource Manager and note the username and password. Log out of Resource Manager, and then log in to Resource Manager again using this user account. Navigate to **Reporting > Advanced** in the Resource Manager UI and successfully confirm that you were redirected to the Analytics UI and automatically logged into analytics. This confirms that single-sign on between Resource Manager and Analytics is working correctly. Note that the built-in Resource Manager "admin" account cannot be used for this purpose.

2   From the **Reporting > Configuration** screen in the Resource Manager UI, click the **Add Batches** button. Simply verify the MODEL, EVENTS and an entry matching the name of each collector (for Perf ETL extractors) are listed in the **Extractor** drop-down. This confirms that the extractor daemons have successfully registered with the Analytics server and that therefore the Analytics server will automatically schedule work for them on an ongoing basis.

3 Wait until at least midday local time on the day AFTER installation before proceeding with this step. This gives Analytics time to have scheduled and executed MODEL, EVENT, and PERFORMANCE extracts for each collector and to have aggregated that data which is necessary before any out-of-the-box data analytics capabilities in Analytics will show data. Then verify the following:

a That all batches in **Reporting > Configuration** screen with an end time of within 15 mins prior to the current time are marked "COMPLETED".

b Log in to Resource Manager using the test user account created in verification step 1 above. Navigate to **Reporting > Advanced** in the Resource Manager UI and then **View > Repository** in the Analytics UI. Verify that executing the following reports with a date range of "DAY-1" to "DAY-1" returns some data.

■ **Zenoss > Current Reports > Event Reports > Events by Class Report**
■ **Zenoss > Current Reports > Device Performance > Availability, CPU Usage, and Memory Usage Exceptions Report**. You may need to adjust all the input parameters specifically to match your environment to get results from this report, for example, Availability < 101%, Memory/CPU Usage > 0%.

If all three of these verification steps succeeded, congratulations, your Analytics installation was successful! You should proceed to read the documentation on *Working with Analytics* on page 26 and perform further initial ETL configuration activities before you are ready to start using the capabilities provided by Zenoss Analytics.

Removing an Analytics installation

To remove an installation of Analytics that was installed by RPM on a separate server:

1 Remove the ZenETL ZenPack from the Resource Manager master:

a Log in to the Control Center browser interface.

b Stop Resource Manager by clicking **Stop** in the Actions column for `Zenoss.resmgr`.

c In the Stop Services dialog, click **Stop Services**.

d Display the child services of Resource Manager to ensure they have stopped.
In the **Application** column of the **Applications** table, click `Zenoss.resmgr`, then scroll down to the **Services** table. Stopped services have a grey circle icon in the **Status** column.

e Create a snapshot of the Resource Manager service by logging in to the Control Center master host as a user with `serviced` CLI privileges.

Execute the following command to create a snapshot.

```
serviced service snapshot Zenoss.resmgr
```

The snapshot ID is displayed upon completion.

f Restart the following required services.

■ `ZooKeeper`
■ The modelling and event database services:

■ `mariadb-events`
■ `mariadb-model`
■ `memcached`
■ `RabbitMQ`
■ `redis`
■ `zencatalogservice`
■ `zeneventserver`
■ `Zope`

> **Note**   In the **Services** table, the Failing icon (a red circle with an exclamation point) in the **Status** column represents the cumulative result of one or more customized health checks. To view the status of individual health checks, move the pointer over the icon, which displays a pop-up.

When the Failing icon is present, a service is unable to support the normal operations of Resource Manager. For this task, the `Zope` health checks includes failing health checks of `zproxy_answering`, which does not affect this procedure.

g Remove the `ZenPacks.zenoss.ZenETL` ZenPack by executing the following command.

```
serviced service run zope zenpack-manager uninstall
  ZenPacks.zenoss.ZenETL
```

2 In the Control Center browser interface, restart Resource Manager.

3 Log in to the Analytics server and change to the root user:

```
su - root
```

4 Stop the Analytics server:

```
service zenoss_analytics stop
```

5 Remove the Analytics program files:

```
rpm -e zenoss_analytics-version-build
```

This command removes the program files, but does not remove configuration, log files, data files, or databases.

To remove configuration files, enter:

```
rm -rf /etc/zenoss_analytics
```

To remove data and log files, enter:

```
rm -rf /opt/zenoss_analytics
```

To remove databases (this will erase all your data), enter the following command as the root user:

```
mysql -u root -e 'drop database reporting; drop database
  zenoss_analytics';
```

Upgrading Analytics

3

This chapter provides instructions for upgrading Analytics. Resource Manager will not be upgraded as part of this process. If Resource Manager is at version 4.x, then it will remain at its 4.x version after the upgrade to Analytics version 5.x.

Based on your current installed version of Analytics, read the corresponding section for information about the upgrade process.

- *Upgrading an Analytics 5.0.x instance* on page 21
- *Upgrading to Analytics 5.x from Analytics 4.x* on page 24

Upgrading an Analytics 5.0.x instance

If you are upgrading from Analytics 4.4.0, see *Upgrading to Analytics 5.x from Analytics 4.x* on page 24.

Upgrading from Analytics 5.0.x to Analytics 5.1.x is a straightforward process that requires approximately 15 minutes of downtime for Analytics and Resource Manager. No data warehouse "schema" changes affect your data, so any existing reports work without modification post-upgrade, which is done "in-place".

To perform an upgrade to Analytics 5.1.x from an earlier version of Analytics 5.0.x, perform the following process:

1 Ensure that the `/tmp` directory has 50G of space or adjust your `/etc/my.conf` tmpdir setting appropriately.

2 In the Control Center UI, stop the following services (found in Analytics ETL group):

- `zenmodeletl`
- `zeneventetl`
- `zenperfetl`

3 Stop the Analytics service by logging in to the Analytics server as the `root` user and enter the commands:

```
sudo -s
service zenoss_analytics stop
```

4 On the Analytics server, upgrade the RPM for Analytics.

```
rpm -Uvh zenoss_analytics-5.1.x-<version>.noarch.rpm
```

Ignore the suggestion to run the `/opt/zenoss_analytics/bin/upgrade_db.py` process, which is not applicable to this type of upgrade.

5 Proceed as follows based on the Analytics version to which you are upgrading:

- To 5.0.5 or later: Skip this step.
- To Analytics 5.0.3 or an earlier 5.0.x version, and you have previously connected Analytics to a Resource Manager 5.x installation:

  Execute the following command as `root` on the Analytics server to work around a known issue (ZEN-24476) concerning renaming performance extractors prior to first service start:

```
mysql -u root reporting -e
   "update meta_extractor set extractor_name =
      CONCAT(extractor_name,'/0')
   where extractor_type = 'PERFORMANCE'
   and zenoss_instance_key in
      (select zenoss_instance_key from dim_zenoss_instance
      where query_service_account is not null);"
```

6 Start the Analytics service again:

```
service zenoss_analytics start
```

7 Immediately tail the Analytics application log (a Tomcat log) and watch it as Analytics starts up for the first time post upgrade.

```
tail -F /opt/zenoss_analytics/logs/catalina.out
```

On first startup post upgrade, the data warehouse schema is adjusted, if necessary. This process usually takes about 5 minutes, during which the Analytics application is unavailable. Wait for the following message, and then proceeding.

```
INFO: Server successfully started in <time in ms>
```

This completes the Analytics server upgrade.

8 Upgrade the client side by upgrading the `ZenETL` ZenPack following the normal procedure for installing a ZenPack, and then restart services.

For Resource Manager 4.x, perform the following:

1 Log in as the `zenoss` user.

```
sudo su - zenoss
```

2 Install the `ZenETL` ZenPack:

```
zenpack --install ZenPacks.zenoss.ZenETL<version>.py2.7.egg
```

3 Issue the `stop` command:

```
zenoss stop
```

4 Issue the `start` command:

```
zenoss start
```

5 Update all hubs and collectors to push the new ZenPack code out using `dc-admin`.

For Resource Manager 5.x and 6.x, perform the following steps:

1 Copy the `ZenETL` ZenPack `egg` file to a local directory on the Control Center master host, such as `/tmp/zenpacks`.

2 Change the file permissions.

```
chmod -R 777 /tmp/zenpacks
```

3 Change directory to the directory in which the ZenPack `egg` file is located.

```
cd /tmp/zenpacks
```

4 Install the `ZenETL` ZenPack.

```
serviced service run zope zenpack-manager install
  ZenPacks.zenoss.ZenETL<version>.py2.7.egg
```

5 To speed up performance data extractions in Resource Manager 5.2.x or later (or any Resource Manager instance that has OpenTSDB v2.2 or later installed):

Locate the `/opt/zenoss/etc/opentsdb/opentsdb.conf` file that is editable in the "opentsdb reader" service in Control Center.

Set configuration option `tsd.query.skip_unresolved_tags` to `True`.

6 Initialize the `zenmodeletl`, `zeneventetl`, and `zenperfetl` services:

```
serviced service stop Zenoss.resmgr
# Wait for containers to stop
serviced service start Zenoss.resmgr
```

Fixing syntax errors in aliases

Perform this step after successfully installing or upgrading Analytics.

In the LinuxMonitor ZenPack, there are syntax errors that need to be manually fixed in order for Analytics to perform data extraction. Perform the following procedure:

1 Log in to the Resource Manager master as `root` or a user with superuser permissions and execute the following command:

```
serviced service attach zope/0
```

2 Change to the `zenoss` user and navigate to the indicated folder:

```
su - zenoss
cd /opt/zenoss/ZenPacks/ZenPacks.zenoss.ZenETL-<Version>-py2.7.egg/
ZenPacks/zenoss/ZenETL/bin
```

where `<Version>` is the version of the ZenETL ZenPack that was installed.

3 Execute the following commands to run the `dumpAliases` script, make the necessary changes, and commit the changes:

```
./dumpAliases.py --aliases=with
```

```
egrep "(mem_buffers__pct|mem_cached__pct|mem_free__pct|
mem_swap_free__pct|lvm_pv_free__pct|lvm_vg_free__pct)" aliases.txt >
 aliases_changes.txt

sed -i 's/,\*,100/,100,\*/' aliases_changes.txt

./manageAliases.py --action=add --inputFile=aliases_changes.txt --
commit

exit
exit
```

4   Execute the following commands to stop the `zenperfetl` service, remove the disk cache that does not get cleared on restart, and then start the `zenperfetl` service.

```
serviced service stop zenperfetl

rm -rf /opt/serviced/var/volumes/<tenantid>/etl-analytics/zenperfetl/
perf-config-cache

serviced service start zenperfetl
```

5   Wait until the new PERFORMANCE extractor has registered with Analytics. You can check this by an ssh connection to the Analytics server and then opening the mysql console and entering the following command.

```
select * from meta_extractor where extractor_name = 'PERFORMANCE';
```

You will know the extractor has registered properly when the previous command returns only one row.

6   Get the key of the new extractor by executing the following command:

```
select extractor_key into @extractor_key from meta_extractor where
 extractor_name = 'PERFORMANCE';
```

7   Update all existing performance batches to have the new key:

```
update meta_batch set extractor_key = @extractor_key where
 extractor_key != @extractor_key and extractor_key in (select
 extractor_key from meta_extractor where extractor_type =
 'PERFORMANCE');
```

8   Remove all the old performance extractors:

```
delete from meta_extractor where extractor_name like '%/_%';
```

Upgrading to Analytics 5.x from Analytics 4.x

Analytics 5.x is fully compatible with Resource Manager 4.2.x and later. Analytics 5.x is officially supported on CentOS/RHEL 7.2. However, anecdotal evidence from the field strongly indicates that the system works without issue on CentOS/RHEL 5 and 6. Thus, an upgrade "in-place" is technically possible; however, CentOS/RedHat do not recommend an in-place OS upgrade to CentOS/RHEL 7.2. Zenoss strongly recommends that you transition to CentOS/RHEL 7.2, if possible, as part of an Analytics 5.x upgrade.

If you cannot deploy CentOS/RHEL 7.2 at your organization, contact Professional Services to ensure that you have a plan in place for an eventual move to CentOS/RHEL 7.2.

Upgrading from Analytics 4.x might involve moving an Analytics 4.x installation in CentOS/RHEL 5 or 6 to a new CentOS/RHEL 7.2-based server as part of the upgrade process or an in-place upgrade without movement.

At a high level, the upgrade process from Analytics 4.x to Analytics 5.x involves the following steps:

- Preparing the old or new environment for upgrade
- Taking an export of the Jaspersoft reporting capabilities in your current install
- Copying/moving the database storage for the underlying data warehouse database to a CentOS/RHEL 7.2 server
- Installing MariaDB 10.0.x on the new server
- Removing any existing Java and replacing it with the Java OpenJDK
- Installing the Analytics server software
- Reimporting Jaspersoft reporting capabilities
- Upgrading the `ZenETL` ZenPack on the Resource Manager master

Contact Zenoss Professional Services for assistance with this process. As a "one-time" event, this upgrade scenario does not warrant detailed legacy documentation. We are happy to assist you with the upgrade to ensure your success.

No data warehouse "schema" changes affect your data with this upgrade scenario, so any existing reports work without modification post-upgrade and are transitioned to the new database as part of the upgrade.

Working with Analytics

<div style="text-align: right; font-size: 3em;">4</div>

The following sections provide information about working with Analytics:

- *Starting and stopping the Analytics server*
- *Logging in to Analytics*
- *Creating batches*
- *Working with the Analytics Repository*

Analytics architecture

Analytics combines extract, transform, and load (ETL) procedures with a data warehouse and reporting tools to process report data.

Analytics creates two databases:

- **zenoss_analytics (Analytics database)**: Stores report design, execution information, and the results of the reports, including PDFs. You can also access previous reports stored here.
- **reporting**: Warehouses all data retrieved from Resource Manager. Also referred to as the "etl database".

Analytics uses the following functionality:

- **Jaspersoft**

  - A robust, ad-hoc reporting feature to facilitate self-management
  - A built-in report scheduler
  - A wide range of report export formats, including PDF, CSV, Excel, Word, HTML, and JPG.
  - Analytics reports and features are accessed through an integrated interface. Reports are distributed by email.
- **Resource Manager**

  - Out-of-the-box (OOTB) example domains, views, and "reports"
  - Additional capabilities installable from ZenPack Analytics bundles
- **Data Warehouse**

  - A data warehouse used for enterprise reporting. The data warehouse schema is driven by meta data provided by the ZenETL ZenPack, providing a flexible platform for report creation.
  - The following three data sources are used:

    - Zope object database (ZODB)

- MariaDB event database (EventDB)
- Centralized performance data storage (HBase cluster)

ETL is not a continuous process; it extracts and processes data periodically (according to user-defined parameters) in *batches*. Model, event, and performance data are independently extracted.

As part of Resource Manager, three daemons run to export data from its sources of data to the Analytics server:

- **zeneventetl** - Extracts event information from ZEP, which is then sent to the Analytics server for subsequent transform and load. There is no necessary configuration to be done. All event information is automatically ETLed to the data warehouse.
- **zenmodeletl** - Extracts ZODB model information, which is then sent to the Analytics server for subsequent transform and load. There is no necessary configuration to be done. All event information is automatically ETLed to the data warehouse.
- **zenperfetl** - Performance data for the subset of data points with aliases is sent to the Analytics server for subsequent transform and load. This daemon also runs on each collector. Typically, this is run on the Resource Manager server and in the collector instances in the various resource pools.

All data points are available for performance data extraction. You can select which data points are extracted by assigning them an alias.

Figure 1: Analytics components
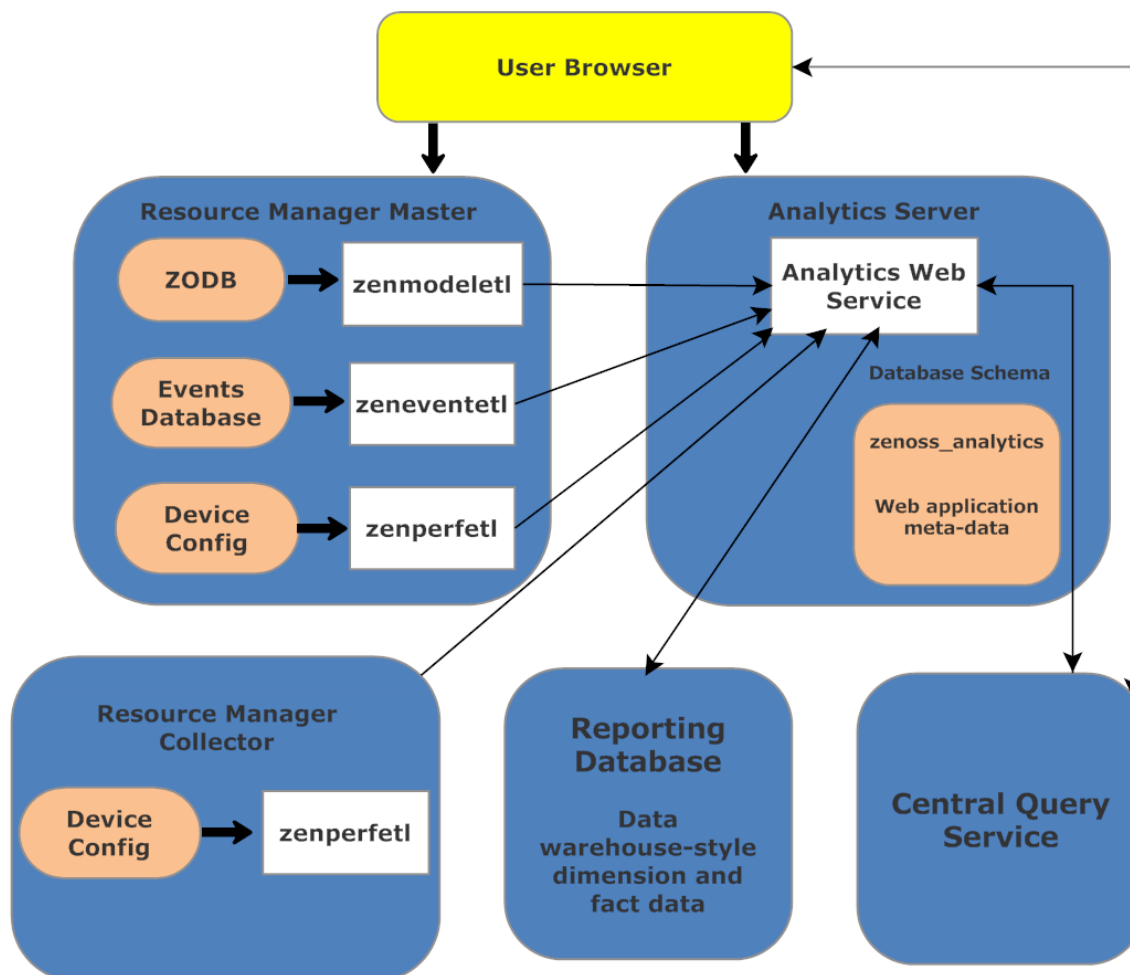
Export data point data by creating aliases

This section provides information about how Analytics and Resource Manager share performance data, and how to manually modify data points.

All monitored data points are available for performance data extraction. However, this configuration is something that needs to be done knowingly, proactively, and maintained carefully ON A PER DATA POINT BASIS and on an ongoing basis using configuration management processes appropriate for your organization. The initial configuration in this section is required before you start to try to report on any performance data in Analytics.

Data point ETL configuration is attained by adding "Aliases" to data points. It is EXACTLY the presence of a least one alias on a monitored data point, that causes it to be ETLed to the Analytics data warehouse. In other words, if a data point does not have an alias, that data is not being extracted and therefore is not available for Analytics. Note that it is perfectly valid to have more than one alias on a data point, and indeed, you should do so if you want that one value available in different units in the data warehouse. Duplicating data in the data warehouse is strongly encouraged instead of having to unit convert data at reporting time in real time.

Resource Manager data points represent raw, collected metrics. For example, view the `/Server/Linux` monitoring template. The ssCpuIdle data point is collected by the ssCpuIdle data source. ssCpuIdle represents the percentage of idle time for all CPUs on this box.

Edit the data point, and you can examine its defined aliases. The cpu__pct alias represents the percentage of CPU used by the box. The alias value is calculated by using the data point and a reverse Polish notation (RPN) expression. Aliases are used to normalize raw data point values across data point types.

Note To avoid issues with percentile and projections, limit alias names to 31 characters. Do not use periods in alias names.

You can provide a formula to be used during extraction of data. This allows you to convert data points with different units to the same units during extraction. For example, if you are monitoring disk space on a Unix system (bytes) and a Windows system (bits), you can use a formula to convert both to kilobytes and have them use the same alias but different RPNs.

On the `/Server/Windows` monitoring template, open the ProcessorTotalProcessorTime data point in the ProcessorTotalProcessorTime data source. An alias "cpu__pct" is defined with no RPN formula. In this case, the raw data point represents the desired quantity ("Percentage of Consumed CPU Time"). The alias, cpu__pct, allows you to compare two different data sources.

The same principle applies to other data points. Some devices give network bandwidth by using kilobytes; others use megabits. Aliases provide a method to normalize these data points to a common denominator.

There are two ways you can add aliases to a data point:

- Manually adding and modifying them a data point at a time using the Resource Manager UI data point screen as mentioned previously. This is highly discouraged. Not only is it tedious and error prone, but there is no record of what was done.
- Running the scripts provided in the ZenETL ZenPack to manage them in bulk. This is highly recommended.

Note Periods are not allowed in aliases since the alias names are used to name the database tables in Analytics that store them. Indeed, while the structure of individual tables will be identical on all Analytics installs, the schema in total for performance data in Analytics is dynamically controlled by exactly these alias definitions and therefore never the same on any two installs. Tables are dynamically created in the data warehouse as needed if new aliases are added.

Required initial configuration post installation

After installing or upgrading Analytics, there are several configuration steps that are REQUIRED to be completed before attempting to use Analytics.

Adding self monitoring and administration scripts

Download the latest version of the `ZenPacks.zenoss.ZenossAnalyticsMonitoring` ZenPack from Leapfile. Once you have saved, unzip the `.egg` file and read the `README.md`. Follow all the instructions to get the ZenPack installed in Resource Manager and the Analytics server management scripts put in place in `/opt/zenoss_analytics/bin` on the Analytics server.

Performing initial performance data ETL configuration

Since monitoring templates are a function of the ZenPacks that provide them, the default set of aliases provided for any particular ZenPack is a function of what is provided by that ZenPack. Details of default aliases provided by a particular ZenPack are provided in the "Zenoss Analytics" section of the webpage for that ZenPack on *https://www.zenoss.com/product/zenpacks*.

Where it exists, such default supplied configuration should be thought of as definitive documentation of a fully comprehensive set of ETL configuration for all the data points monitored by that ZenPack and this ETL configuration will evolve with future ZenPack development. Thus, it is likely significant overkill to ETL all such data points to your Analytics data warehouse without any actual requirement for reporting on them. Doing so is likely to add more ETL load and use a lot of disk space on the Analytics server, which makes managing Analytics much harder than it would otherwise be.

The following procedure should be run immediately after initial install and repeated after every ZenPack upgrade or Resource Manager upgrade to re-audit the ETL configuration so that it is not inadvertently changed as a result of such activities.

If you are on Resource Manager 4.x, enter the following on the Resource Manager master:

```
su - zenoss
```

If you are on Resource Manager 5.x or 6.x, enter the following on the Control Center master:

```
serviced service attach zope/0
su - zenoss
```

Then, regardless of your Resource Manager version, change directory to the ZenETL ZenPack directory. In the following command, replace `<version>` with the current ZenETL ZenPack version:

```
cd /opt/zenoss/ZenPacks
cd ZenPacks.zenoss.ZenETL-<version>-py2.7.egg
cd ZenPacks/zenoss/ZenETL/bin
```

In this directory, there are two relevant python scripts:

- `dumpAliases.py`

    - This script can be used to traverse your entire monitoring template hierarchy and dump out details of data points and aliases.
    - Run `dumpAliases.py --help` to read the detailed help.

- The script takes one required parameter `--aliases=` with a value of either `with` or `without`. The former dumps details of all aliases on data points, the later dumps details of all data points with no aliases.
- The optional parameter `-f` or `--outputFile` can use to specify a filename to write the information to. If you exclude this file, the default filename of `aliases.txt` is used. The output format of this file is in the exact required input format for `manageAliases.py`.
- `manageAliases.py`

  - This script can be used to bulk or remove aliases from data points in your monitoring template hierarchy. Run `manageAliases.py --help` to read the detailed help on options. Run `manageAliases.py --printhelp` to read the detailed help on input file format and output results.
  - The script takes one required parameter `--action=` with a value of either `add` or `remove`. The former adds aliases to data points, the later removes existing aliases from data points.
  - By default it is "read-only", that is it only reports what it could/would do regarding changing aliases on monitoring templates. You must call it with the `--commit` option to actually make changes effective on your monitoring templates.

Use these scripts as follows to get your ETL configuration to an initial known state with metrics for device-level CPU and memory usage, and component-level filesystem and IP interface metrics ETL to Analytics. This will allow out-of-the-box sample reports to execute successfully. In the following steps, replace <YYYYMMDD> with today's date.

1   Dump out the current state of aliases in your install:

```
./dumpAliases.py --aliases=with --
outputFile=original_aliases_<YYYYMMDD>.txt
```

2   Make a copy of this file and keep it somewhere safe for future reference:

```
scp original_aliases_<YYYYMMDD>.txt myuser@myserver:/pathtosafeplace/
```

3   Use this file to remove ALL aliases currently in place:

```
./manageAliases.py --action=remove --
inputFile=original_aliases_<YYYYMMDD>.txt 2>&1 | tee removal_test.log
```

Check that this *would* remove all aliases successfully. The following `grep` should return no alias lines:

```
grep -iv "^-" removal_test.log | grep -v ControlCenter
```

If not, run `manageAliases.py --printhelp` to get information of what the error codes at the start of each line indicate.

If successful, repeat with `--commit` to actually make the changes:

```
./manageAliases.py --action=remove --
inputFile=original_aliases_<YYYYMMDD>.txt --commit 2>&1 | tee
 aliases_removed_<YYYYMMDD>.log
```

4   Confirm this was successful, that is you now have no aliases at all:

```
./dumpAliases.py --aliases=with
```

A file named `aliases.txt` with no contents should have been created. To check that no contents were created, execute the following `grep` command:

```
grep -v ControlCenter aliases.txt
```

5 Create a desired subset of aliases from this file for the above mentioned device- and component-level metrics:

```
head -1 original_aliases_<YYYYMMDD>.txt >
 current_aliases_<YYYYMMDD>.txt
 egrep "\|(cpu__pct|mem__pct|fs__pct|in__pct|out__pct)\|"
 original_aliases_<YYYYMMDD>.txt >> current_aliases_<YYYYMMDD>.txt
```

6 Keep a copy of this file safe for future reference. It documents what your ETL configuration is always supposed to be. Zenoss highly recommends you put it under configuration management, and incrementally add to it over time as you add more data to the data warehouse to meet Analytics reporting needs.

7 Add these aliases to your monitoring templates:

```
./manageAliases.py --action=add --
inputFile=current_aliases_<YYYYMMDD>.txt  2>&1 | tee add_test.log
```

Check that this *would* add all aliases successfully. The following `grep` should return no alias lines:

```
grep -iv "^+" add_test.log | grep -v ControlCenter
```

If not, run `manageAliases.py --printhelp` to get information of what the error codes at the start of each line indicate.

If successful, repeat with `--commit` to actually make the changes:

```
./manageAliases.py --action=add --
inputFile=current_aliases_<YYYYMMDD>.txt  --commit 2>&1 | tee
 aliases_added_<YYYYMMDD>.log
```

8 Confirm this was successful, that is you now only have exactly the aliases you added:

```
./dumpAliases.py --aliases=with --outputFile=aliases.txt
        sort -u aliases.txt > a
        sort -u current_aliases_<YYYYMMDD>.txt > b
        diff a b
```

The diff should show no differences between the two files.

Verifying your ETL configuration

To verify that your ETL configuration is correct, perform the following:

1 Log in to the Analytics database directly and execute the following query:

```
select distinct metric_name from reporting.meta_metric order by
 metric_name;
```

The aliases you created will appear in the returned list after the next scheduled `zenperfetl` batch post aliases add has successfully completed for a collector that has devices monitored using the monitoring template that aliases have been added to.

Starting and stopping the Analytics server

To stop, start, or restart the Analytics server, such as when troubleshooting problems, use the following commands. While the server is stopped, no data is extracted.

- To start Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics start
```

- To stop and then start Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics restart
```

- To stop Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics stop
```

Logging in to Analytics

To access Analytics, you can:

- Log in to the Resource Manager interface using your Resource Manager user name and password (single sign on).
- Log in directly to Analytics using your Resource Manager user name and password or as an Analytics superuser.

Logging in through the Resource Manager browser interface

Access to Analytics through single sign on (SSO) on the Resource Manager browser interface requires Manager, ZenManager, or ReportingUser privileges, which are assigned through roles. (Manager and ZenManager provide Analytics administrative privileges; ReportingUser provides user-level privileges.) You cannot log in to Analytics solely with Resource Manager administrator privileges. For more information about assigning roles, see the "Managing users" chapter in *Zenoss Resource Manager Administration Guide*.

1 Log in to the Resource Manager browser interface with your Resource Manager user name and password.
2 Select **Reports** > **Advanced**.
   The Getting Started page appears.

Figure 2: Getting Started

From the Getting Started page, you can

- View your reports.
- Create a report.
- Create an ad hoc view.
- Change the Analytics server settings.

Alternatively, make a selection from one of the Analytics menu options:

- **View**: View search results, the Analytics repository, or messages.
- **Manage**: Manage organizations, users, roles, or server settings (accessible by administrators). Do not create organizations in the system because they are linked to Zenoss instances.
- **Create**: Create ad hoc reports, dashboards, or domains.

Logging directly in to Analytics

You can log in directly to Analytics by using your Resource Manager user name and password (with appropriate Resource Manager privileges), or as an Analytics superuser.

1 Browse to the URL of your Analytics server. (`http://<your_server_or_IP_address>:7070` or `https://<your_server_or_IP_address>:443` for SSL).
   The Analytics login page appears.
2 Log in as one of four roles:
   a With your Resource Manager user name and password:

      1 From the Resource Manager Instance list of options, select the Resource Manager instance for which you will provide a user name and password.
      2 Enter your Resource Manager user name and password in the `User ID` and `Password` fields.

Figure 3: Analytics Login - Resource Manager Credentials

b As an Analytics superuser:

1 From the Resource Manager Instance list of options, select `Internal Authentication.`

2 If the user is located in an Analytics organization other than Resource Manager, enter the organization name. (Typically, you would not enter a value in this field.)

3 Enter your Analytics User ID and password in the `User ID` and `Password` fields.

Figure 4: Analytics Login - Analytics Superuser

3 Click **Login**.

Creating batches

You create batches to populate the reporting database. Every eight hours, the ETL processes a batch and extracts information from the previous eight-hour window. The main types of batches match the primary data types (Performance, Model, and Event).

Existing batches and associated status information appear in the batch list.

To access the batch list, choose **Reports** > **Configuration** from the Resource Manager browser interface. The batch list appears.

Figure 5: Batch list

The list shows each batch and its status: UNSTARTED, CONFIGURING, READY, EXTRACTING, STAGING, FAILED, COMPLETED, or CANCELLED. The CANCELLED state occurs when the collector goes offline for a long period of time.

Note   The CONFIGURING and READY states are currently only valid for Performance batches.

In general, you do not need to create manual batches. The only reason a manual batch should be created is as a result of a previous failed scheduled batch.

To add a manual batch:

1   From the batch list, click **Add Batches** and make selections as follows:

   a   **Extractor** - Choose `MODEL`, `EVENTS`, or `PERFORMANCE`. This process may take a few minutes after you connect the Resource Manager to the Analytics server and update the hubs and collectors.

   b   **Begin Date** - Optionally adjust the begin date and time. By default, the value is set to two weeks before the current Analytics server date and time.

   c   **End Date** - Optionally adjust the end date and time. By default, the value is set to the current Analytics server date and time.

   Note   Creating a batch that finishes in the future leaves a batch with a status of UNSTARTED. The batch will not start until that future date. Setting dates for MODEL is ignored; you will just get the current model.

2   Click **Submit**. The batch is added to the batch list.

Configuring repeat intervals

When configuring batch intervals (in Resource Manager), you must set an Analytics repeat interval for each Analytics server (rather than for each Resource Manager instance). This means that the same Analytics repeat interval setting will appear across Resource Manager instances.

You can configure repeat intervals separately for each extractor type. The default values are:

- Event Jobs - 8 hours
- Model Jobs - 24 hours
- Performance Jobs - 2 hours

To edit the repeat interval:

1   Expand the Jobs folder, and then select one of the extractor types.

Figure 6: Edit Repeat Interval

2   Select a repeat interval (.25 hours to 24 hours), and then click **Submit**.

Filtering performance batches to a list of metrics

When you schedule a batch for a collector, all aliases that are present for the devices being monitored are extracted. If you want to add a new alias to an existing batch at a later date, you will only get future data when the scheduled batches are run. To collect past data, you need to create a filtered list of metrics to collect the past data. The following workflow should be used when adding a new alias to an existing scheduled batch:

1   Add the new alias in ZODB. See *Adding aliases*.
2   Extract one batch normally, or wait for the next scheduled performance extraction to complete. This ensures that the future batches will contain the new metric and that Analytics has registered the new metric.
3   Create a new manual batch to collect past data for the new metric by inserting the metric names into the `meta_batch_metric_assoc` table.

Setting a delay on performance batches

A configuration option for the `zenperfetl` daemon allows you to delay the execution of a performance batch until it reaches a specified amount of time into the past. This defaults to fifteen minutes and should not be changed without consulting Zenoss Support.

To change the value of the delay on performance batches:

1   Log in to the Control Center master host as a user with `sudo` privileges.
2   Attach to the zenperfetl service.

```
serviced service attach zenperfetl
```

3 Change to the `opt/zenoss/etc` directory.

```
cd opt/zenoss/etc
```

4 Open `zenperfconfigetl.conf` in an editor and uncomment the line containing the setting for `batch-delay` and set it to the value you want (in minutes).

Parallelization of performance loads

To increase loading of performance data, a new configuration has been added which loads performance data in parallel. Based on your I/O performance, you may want to increase or decrease the default number of performance threads that Analytics uses. The default value is 10, which is the equivalent of 5 cores.

To change the value, you must edit the `meta_setting` table directly:

1 Log in to the Analytics data warehouse database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following to lower the `concurrent_threads` to a value of 6. (If you want to increase parallelization, set the value to 12.)

```
update meta_setting set setting_value = '6' where setting_name =
  'concurrent_threads';
```

Limiting concurrent performance extractions

This topic only applies to Resource Manager 4.x systems. Skip this topic if you are running a Resource Manager 5.x or 6.x system.

Since multiple logical collectors can be installed on a single host, there will be multiple `zenperfetl` daemons running on the same host. Under normal conditions, these daemons will begin extraction nearly simultaneously, which can saturate disk I/O for a period of time. To manage this behavior, Analytics tracks the hosts where each of its performance extractors are running. There is a `concurrent_extractions_limit` column in the `meta_perf_extractor_limits` table. By default, there is no limit. If you have an environment where there are many logical collectors installed on a single host, you may want to change the value of `concurrent_extractions_limit` to limit the number of concurrent performance extractions that can take place at the same time on the host.

To change the value, you must edit the `meta_perf_extractor_limits` table directly:

1 Log in to the Analytics data warehouse database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following to set the `concurrent_extractions_limit` to a value of 4. The actual value you will want to use depends on your disk I/O.

```
update meta_perf_extractor_limits set setting_value = '4' where
  setting_name = 'concurrent_extractions_limit';
```

Alternate metric naming conventions for performance data

Analytics users can opt-in to a new alternate metric name for OpenTSDB data for certain types of devices, but this action will reduce Analytics ability to perform bulk data retrievals from the Query service. Alias names for Analytics are defined separately, so there are no changes needed to alias names.

If you do opt-in to the new alternate metric name and do not migrate your OpenTSDB data, there will be a gap in the Analytics data warehouse data. To mitigate this, you will need to run an Analytics ETL just prior to the metric name change.

Any gap will be limited to any data related to the affected opt-in metrics renamed and that was collected after the last ETL and before the execution of the opting-in metric rename. After the opt-in occurs, only data collected for the affected renamed metrics after the opt-in was executed will be ETL'ed. All metrics not affected by the rename will ETL as normal and NOT result in an ETL gap.

For more information on the alternate metric name process and migrating OpenTSDB data, see the *Zenoss Resource Manager Administration Guide*.

Configuring the Analytics data retention policy

Based on your business needs, you may want to change the data retention policy for certain types of data. The data retention policies are configured through direct manipulation of the `meta_setting` table in the data warehouse ("reporting") database.

Column `setting_value` specifies the number of days for which a given type of data is kept.

Values in column `setting_name` are as follows:

To see the data retention setting in the `meta_setting` table, perform the following:

1 Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following command:

```
select * from meta_setting where setting_name in
('daily_retention_days','raw_retention_days','event_retention_days',
'hourly_retention_days');
```

A table similar to the following is displayed:

setting_key	setting_name	setting_value	setting_description
1	daily_retention_days	365	Days to keep daily aggregation data.
2	raw_retention_days	30	Days to keep raw fact data.
3	event_retention_days	365	Days to keep event data.
10	hourly_retention_days	90	Days to keep hourly aggregate data.

The value displayed in the `setting_value` column specifies the number of days for which the data is kept for the given type of data.

- Raw, Hourly, and Daily retention refer to performance data only.

  - Raw indicates the lowest resolution of extracted performance data (typically 5 minute data).
  - Hourly and Daily represent hourly and daily aggregates of that data.
- The values shown above are Zenoss recommendations, which are satisfactory for typical reporting and analysis needs.
- Event retention applies to all events extracted to Analytics.

3 Data is automatically purged by the system on a daily basis (typically runs at midnight) based on these settings. Updating them takes effect immediately at the next data purge. To change one of these settings, update the value in the table.

For example, to change the retention policy for raw performance data from 30 days to 45 days, issue the following database update:

```
update meta_setting set setting_value = '45' where setting_name =
  'raw_retention_days';
```

Working with the Analytics repository

The repository is the Analytics internal storage for all reports, dashboards, analysis views, and related files. The repository is organized as a tree structure of folders containing those components, much like a file system. Zenoss provides a set of folders that are available upon installation (such as ad hoc components, dashboards, domains, reports, resources, etc.), as shown below.

To access the Repository Folder view, select **View** > **Repository**.

Figure 7: Zenoss folder in repository
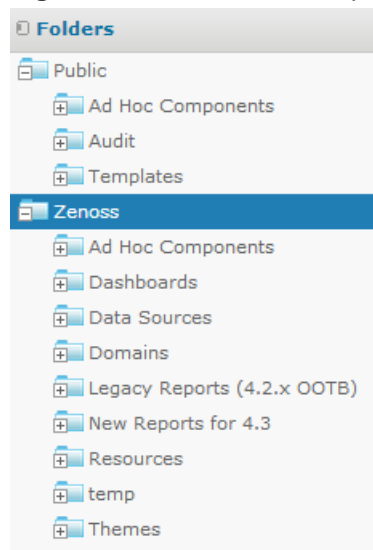
Since the offerings in the Zenoss-provided folders may change in a future update, it is important to note that any changes you make to the Zenoss folder may be overwritten in the future. Therefore, it is important to populate a folder specifically for your organization. You can use the contents of the Zenoss folder as a guide to determine which folders you would like to copy to your organization's space in the repository (typically a folder on the same level as the existing Zenoss folder).

To copy a folder, simply right-click on the folder and select **Copy**, then right-click on the destination parent folder and select **Paste**.

Searching the repository

Once you have set up your organization's folder within the repository, you can easily search for your information for quick access. To search the repository, select **View** > **Search Results**. The **Repository Search Results** page displays search results, as well as lists of reports and ad hoc views.

Figure 8: Analytics repository search results

The results page features:

- **Search controls**: Find resources by searching for text in their names or descriptions.
- **Filters for refining results**: Provides a quick way to find resources based on other criteria, such as type or access time.
- **Search expression and sorting**: Interactive status bar that shows search criteria, filters, and sorting criteria.
- **List of resources**: Resources in the repository that meet the search criteria. Click or right-click a resource in the list to view it or see available actions.

The **Repository Search Results** page allows you to refine or enlarge your set of results. Analytics remembers your settings so that your most commonly needed resources remain visible when you return to the page.

The following sections provide more information about:

- *Searching by name* on page 40
- *Refining with filters* on page 41
- *Changing the search expression* on page 42
- *Interacting with the list of resources* on page 42

Searching by name

Use the search field at the top of any Analytics page to begin a search. Search terms are not case-sensitive. When you click the **search** icon, Analytics opens the repository page with the search term and displays the results. Analytics matches the search term, even word fragments, anywhere it appears in the display name or description string of the results. Analytics does not match folder names or display folders in the results. If you enter multiple words, Analytics searches for resources that match all words, but not necessarily in order. Do not enter quotes around terms or any symbols between terms. If you enter a search term and click **Search** on

the **Repository** page, the search uses the current settings on the page. If you enter a search term and click the **search** icon at the top of any other page, the search uses the following default settings:

- Include subfolders
- Start at the top-most visible folder
- Sort alphabetically by name

Refining with filters

The filters below the folder tree let you refine your search using several other resource attributes. For example, these filters can help you find your most recently viewed reports. You can set each filter independently of the others.

The user filter has the following settings:

Filter Setting	Description
All available (default)	Does not take into account the user's access to the resource.
Modified by me	Selects only resources that have been last modified by the currently logged in user.
Viewed by me	Selects only resources that have been run and viewed by the currently logged in user. This filter applies not only to visualization types, but also to resources that are included in reports (such as images).

The resource type filter has the following settings:

Filter Setting	Description
All types (default)	Does not take into account the resource type; as a result, every type is displayed.
Reports	Displays only reports.
Report outputs	Displays only the output from reports that were scheduled or run in the background. Report output can be any of the supported export types (such as HTML or PDF).
Ad Hoc views	Displays only the output from reports that were created using an ad hoc view into an available data source type (Topics, Domains, or OLAP Connections).
Dashboards	Displays only dashboards.
OLAP views	Displays only OLAP views.
Domains	Displays only domains.
Data sources	Displays only data sources.
Deprecated Reports	Displays only deprecated reports.

The access time filter has the following settings. All times are relative to your effective time zone:

Filter Setting	Description
Any time (default)	Does not take into account the last modification time of a resource.
Today	Resources viewed or modified since the previous midnight.
Yesterday	Resources viewed or modified during the previous day ending at midnight.
Past week	Resources viewed or modified during the past 7 days, including today.
Past month	Resources viewed or modified during the past 30 days, including today.

The scheduled report filter has the following settings:

Filter Setting	Description
Any schedule (default)	Does not take into account the existence of scheduled jobs.
Scheduled	Only reports that have scheduled jobs.
Scheduled by me	Only reports that have jobs scheduled by the currently logged in user.
Not scheduled	Only reports that do not have scheduled jobs, and all other resource types.

Changing the search expression

Above the list of resources, the search expression shows you all of the criteria that contribute to the search. It contains the following elements, always in this order from left to right:

- The search term, if any, or the word `All`
- Any and all filter settings

The search expression provides a quick visual summary of the search criteria for the list of results that appear right below it. The search expression is also interactive, allowing you to easily remove some of the filters. It supports the following actions:

- If there is more than one filter, clicking any of them removes all those to its right.
- You can click the `search term` or the word `All` to remove any folders to the right.

After any of these actions, the search controls displayed on the left are refreshed, and the results are updated.

To the right of the search expression. the sort criteria lets you change the order of the results. Analytics supports the following sorting:

- Click **Name** to sort alphabetically from A-Z. This is the default sort order.
- Click **Modified Date** to sort by the latest modified time and date (most recent at top).

Interacting with the list of resources

- **Multiple Types Condensed**: When there are more than two types of resources listed, the number of resources of each type is limited to a certain number, by default 5. When there are more resources of that

type, there is a link to see all of that type. You can quickly scroll through the condensed results and find the type of resource you want. Click **See all** to display the results of that type.

■ **Single Type Expanded**: When there is a single type of resource in the results, either by circumstance, by clicking **See all**, or because a single-type filter is selected, all resources are listed, with a scroll bar if necessary. If the list of results is still too long, enter or refine the search term, or select additional filters.

Note   When you click **See all** for a type that has a single-type filter, for example Domains, that filter is automatically applied. The filter display and the search expression refresh to reflect this. Use the search expression to remove the filter if necessary. When no filter exists for that type, for example input controls, the filter display and search expression are not updated. Use the type filter list of options to change the types displayed.

Once the resource or resources you want are displayed, you can interact with them in several ways:

■ Click the name of a report or dashboard to run and view it.
■ Right-click the name of a resource to access other operations on the context menu, for example **Open in Designer**. Items appear on the context menu according to your permissions.
■ Select the resource name or click anywhere in the row to select it. (Control-click anywhere in the rows to select multiple resources.) You can drag-and-drop selected items to move them or press Control while you drag-and-drop items to copy them.

Two icons may appear between a report name and its selection box:

■ + indicates that the report has saved options for is input controls. Click + to list the saved options below the report.
■ The **clock icon** indicates that the report has been scheduled to run or that it currently is running in the background. Click this icon to view the list of jobs scheduled for the report.

Workflow for creating reports

The following sections describe the process to follow for creating your own domains, then creating ad hoc views based on those domains, and finally creating and scheduling reports to be run from those ad hoc views.

Creating domains

A *domain* is a metadata layer that provides a business view of the data accessed through a data source. A domain presents the data in business terms appropriate to your audience, and can limit the access to data based on the security permissions of the user who runs the report. You can use a domain to create reports and ad hoc views. Analytics ships with several out-of-the-box domains, and you can create your own domains based on business needs.

Before you create a domain, review the following considerations and recommendations. For more information about creating domains, see the *JasperReports Server User Guide, Release 6.x*.

In practice, you should plan domain features before you begin the creation process. You will use the zenoss_reporting data source for your domain creation. Decide on the elements of your design:

■ tables and columns to choose
■ joins to perform
■ filters to define
■ sets you need
■ item properties to expose to users

Finally, determine a security policy for the data retrieved through the domain. To help you in your creation of domains, we have included an *ERD diagram for Analytics 5.1.x* on page 58 that shows the relationships

between tables in the database. You will also find a sample of some dynamically created dimension tables as well as a flow of performance fact data as it goes through hourly and daily aggregation.

To create a new domain:

1 Log in to Analytics as Administrator or superuser.
2 Click **Create > Domain.**. The Add New Domain screen is displayed.
3 Enter the `name`, `resource ID`, and `description`.
4 Select a **Save Location** by browsing to your organization's `Domain` folder.
5 In the Data Source field, browse to the `zenoss_reporting` data source located at `/organizations/ zenoss/Data_Sources/zenoss_reporting`.
6 In the **Domain Design** section, click **Create with Domain Designer**. The **Domain Designer** screen is displayed. On this screen, you will configure the various aspects of the design.
7 Click the corresponding tab and enter information for the following:
    a **Tables** - Select all tables whose columns you want to use in the domain, either directly or indirectly.
    b **Derived Tables** - Enter queries whose results appear as derived tables in the domain.
    c **Joins** - Define inner and outer joins between all tables and derived tables.
    d **Calculated Fields** - Enter expressions whose results appear as calculated fields. An example of a calculated field can be found in the **IP Interface Inventory** domain in the `Zenoss/Domains` folder. Note that the `_deleted` calculated fields are based on whether the time stamp is present.
    e **Pre-filters** - Specify conditions on field values to limit the data that is accessed through the domain. An example of a pre-filter can be found in the **Daily Device KPI** domain in the `Zenoss/Domains` folder. You cannot use this domain if you have components. Thus, there is a pre-filter defined as `component_key = 0`.
    f **Display** - Organize the visual aspects of the domain and change the display properties of tables, columns, sets, and items exposed to domain users. The properties are located in the far-right column. The **Label** is the friendly text display, while the **ID** is the system name that should not be changed unless there are similar IDs coming from different input controls. In the **Summary Function** section, all performance metrics are defined as measures, while everything else in the system (including time) is defined as a field.
8 As you create your domain, click **Check Design** to validate your domain design. If you have issues, click **Export Design**, which will create an XML file of your schema that can be shared with Zenoss Support.
9 Click **OK** once you have finished with the **Domain Designer**. You will return to the Edit Domain screen.
10 If all the information is correct including your design, click **Submit** to add the domain to the repository.

Creating ad hoc views

The Ad Hoc editor supports the creation of views for various types of reports: tables, crosstabs, and charts. You interact with the editor to create these views by dragging and dropping elements. You can add and summarize fields, define groups, label and title the report, and format data for each field.

To open the Ad Hoc editor:

1 Log in to Analytics as an administrator or superuser.
2 Click **Create > Ad Hoc View.** The Select Data screen is displayed.
3 Select your data source from the list.

> **Note**   You can also navigate through the data source by clicking on the **View as tree** icon. (**root > Organization > Zenoss > Ad Hoc Components**)

■ **Topics** - These can be either a JRXML file that is associated with a data source on the server or can be created from a domain with one or more filters applied to it. Using a topic as your source generates an empty view, which allows you to begin adding data to your view right away, without choosing, pre-

filtering, or changing display names of the data (all of which are required steps when creating a Domain-based view).

- **Domains** - They specify tables in the database, join clauses, calculated fields, display names, and default properties, all of which define items and sets of items for creating Ad Hoc views.

4 Fill out the New Ad Hoc View screen based on the data you selected. For more details on the functionality and toolbar icons, see the *JasperReports Server User Guide*.

5 Select a view type:

- **Tables** - Used to view values in the database and to summarize the values in columns.
- **Charts** - Used to compare one or more measures across multiple sets of related fields. Charts summarize data graphically. Some types of charts include bar chart, line chart, and pie chart, among others.
- **Crosstabs** - Used to aggregate data across multiple dimensions. Crosstabs are more compact representations than tables; they show only computed values, rather than individual database values. Columns and rows specify the dimensions for grouping; cells contain the summarized measurements.

6 When completed, save your Ad Hoc View. This view is saved in the repository. You can use this view to create reports or you can save the view as a report itself so that you can embed the data content into a dashboard or see the data in the interactive report viewer.

Note   When you create a report from an Ad Hoc view, the report is considered "dependent" on that view. If you later edit the Ad Hoc view, the dependent reports are not updated with the new changes to the view.

Creating and running reports

Zenoss Analytics makes it easy to create and run reports. When you run a report, it opens in the interactive Report Viewer. Using this Viewer, you can personalize and refine the displayed report data. If your report has input controls, you can run the report with one set of data and then another. You can then export the output to another format or you can schedule the report to create an output repeatedly and unattended during off hours or at other times.

Creating a report and editing the output

1 Log in to Analytics.
2 From the Home page, click **Create** in the Reports section; or from any page, click **Create** > **Report**.
3 From the **Create Report** window, select an ad hoc view and click **OK**.
4 If you are prompted for input controls, provide them in the displayed window. If not, the report output appears.
5
  Edit the report output by clicking the Options 🖾 icon in the upper-right corner.

Note   By default, a dynamic date range is used to define the date filter. The range is defined by the unit (DAY, WEEK, MONTH) and a numerical unit that can be positive or negative. For example a range of DAY-7 to DAY is equivalent to the last seven days from when the report is run.

---

6 (optional) While you cannot change the fundamental type of output (you cannot change a chart into a crosstab), you can change various elements of the output based on the type of report. For example, you can resize columns or filter by a column's value. If you are using a chart, you can change the type of chart you want to display. Refer to the *JasperReports Server User Guide* for even more ways to customize your reports.
7 Once you have the report output displaying the way you want, save the report by clicking the **Save** icon.

Scheduling a report job to run

1 Log in to Analytics.
2 On the Home page, click **View list** in the Reports area; or from any page, click **View > Repository**.

3   Use the search field or browse the list of reports to find the report you want to schedule.

4   Right-click the report name and select **Schedule** from the context menu. The Schedule Jobs page appears.

5   Click **Create Schedule**. The Schedule tab appears. Fill out the information requested and select the recurrence of the job. The default value is None, which will run the report only once. Select **Simple** from the drop-down list to schedule the job to recur at an hourly, daily, or weekly interval or select **Calendar** to schedule the job to recur on certain days of the week or days of the month.

6   Set the other values on the appropriate tabs for **Parameters**, **Output Options**, and **Notifications**. Click **Save** to save the schedule. On the Repository view page, you will see a clock icon to indicate that a report has been scheduled to run.

Note   The permissions of the user who schedules a job determines the data that the report exposes. Be aware that sensitive data could be exposed if you schedule a job as an administrative user with no data restrictions because the report output will contain all the requested data in the data source. Any user who receives the report output can view all the data regardless of the user's access restrictions.

Entity-Relationship Diagrams for aggregation-related tables

This section describes percentiles and aggregates and the way to understand how projections are created.

Percentiles operate against the raw data, i.e., `raw_v2_tables`, not against an aggregate. Think of the `nth_percentile_configuration` tables as defining what subset of all the raw data has percentiles calculated, i.e., which rows within a specific `raw_v2_table` (specific `device_key` - group filter, specific `fct_ts` values - business hours filter) and which raw tables are actually read at all (specific metrics).

Projections operate against already calculated daily aggregates or against nth percentile calculations that have already been performed on daily_ or percentile_ tables. You can think of a projection of nth percentiles as a dependency, since nth-percentile configurations can limit the data produced, the projections on that nth percentile have, at a minimum, the same limits. You cannot have projections that specify fewer overall restrictions than the data source it depends on.

Projections groups, business hours, and metric configuration can be thought of as being used to decide which existing aggregate tables the values to be projected are read from, i.e., they only configure which table's values are projected and what additional group filters to apply to select devices.

Some configuration override scenarios:

■   The specified nth-percentile metric configuration overrides the projection metric configuration
■   The nth-percentile configuration overrides the projection configuration for business hour selection on a per metric basis
■   If both nth-percentile and projection have group filters configured, only the intersection of the two sets is used. Otherwise, the configured group filter from either is used, if any are specified at all.

The following Entity-Relationship Diagrams (ERD) show the relationships between aggregation-related tables in the database.

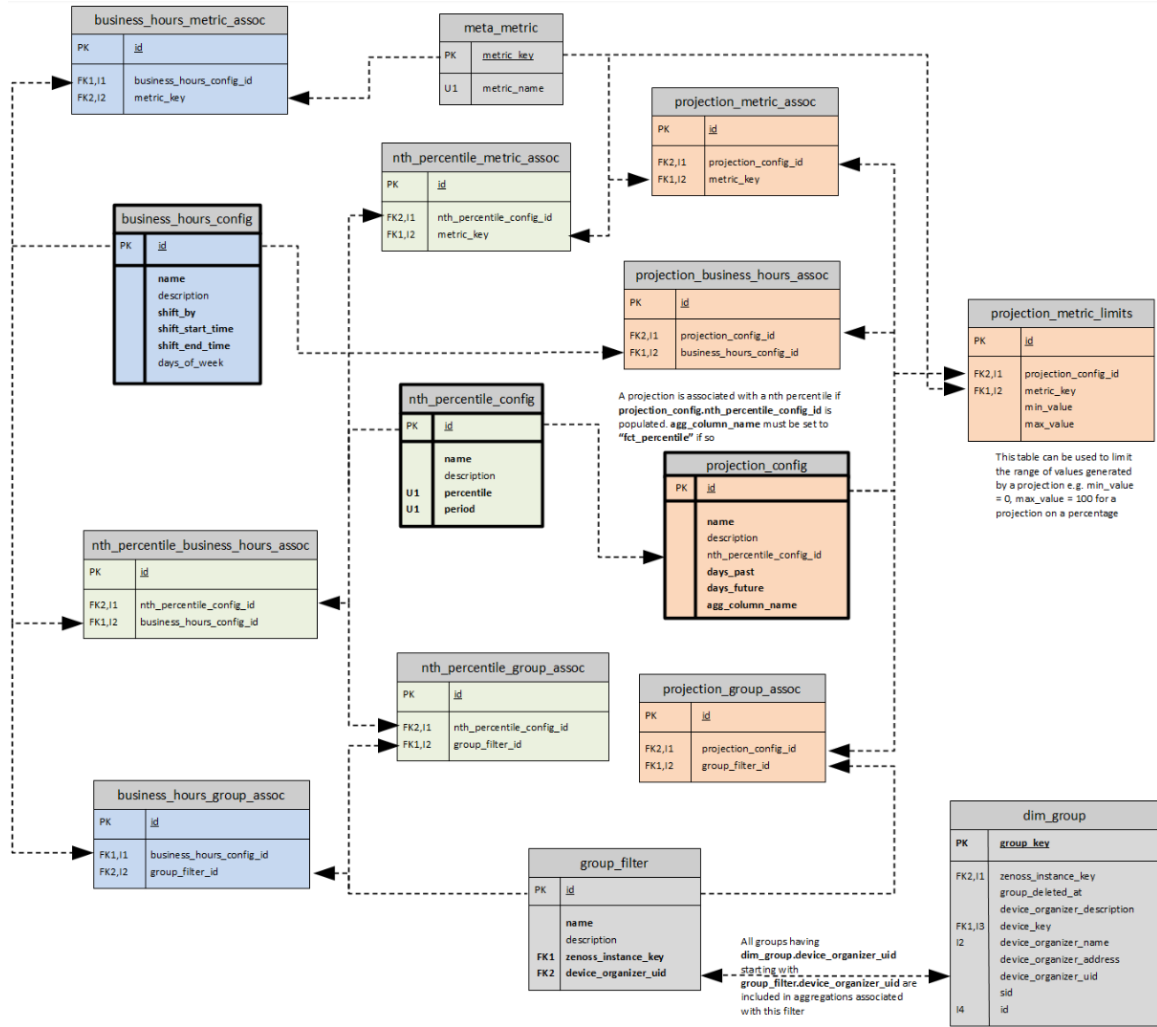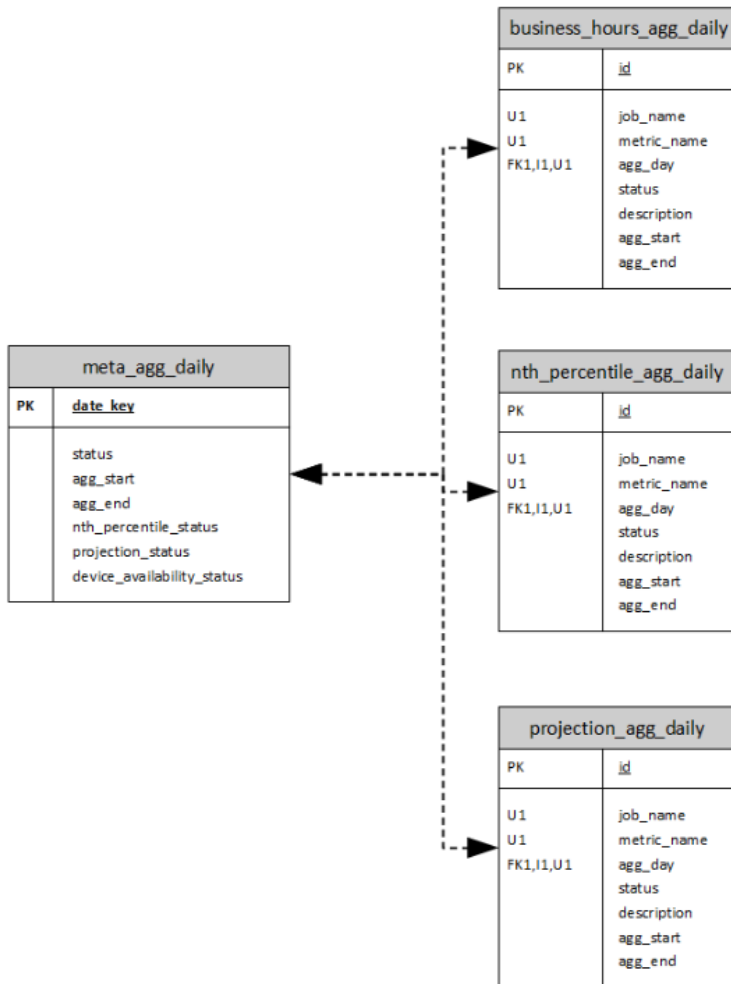**Figure 9:** Configuration ERD of aggregation-related tables

Figure 10: Status tracking ERD of the aggregation execution

Creating business hours aggregation

You can prepare reports showing information relative to the business hours of operation based on a user-specified time offset, which allows for more accurate reporting of how applications and services are performing relative to different user communities. For example, you can aggregate data for a specified time range that has been defined as a "business day" or "shift". Each row is named for the shift or business hours it represents, e.g., `austin`, `london`, `singapore` as locations or shifts like `austin_day`, `austin_swing`, `austin_night`.

The creation of rows for the business hour aggregation is performed manually by inserting records into the reporting database. The tables of interest are described below:

- `business_hours_config`: main table defining the business hours

  - `name`: name of the shift or business hours
  - `description`: up to 255 characters to describe the row (optional)
  - `shift_by`: difference in time between the raw data time stamp and the time specified by the `shift_start` and `shift_end` fields. This value can be negative and is in the format `hh:mm`.
  - `shift_start_time`: time of day when the shift (or business day) begins. The format is `hh:mm`.

zenoss

- shift_end_time: time of day when the shift (or business day) ends. This value can be less than shift_start which indicates that the end of the shift is the following day. The shift begins at the shift_start time and goes up to but does not include the shift_end time. The format is hh:mm.

shift_start_time	shift_end_time	shift_by	Selected range of raw data
08:00	16:00	04:00	12:00 PM through 8:00 PM of the current day
17:00	02:00	00:00	Assuming today is 1/17/2017: 1/17/2017 5:00 PM through 1/18/2017 2:00 AM
06:00	14:00	-11:30	Assuming today is 1/17/2017: 1/16/2017 6:30 PM through 1/17/2017 2:30 AM

- group_filter: optional filter used to reduce the set of devices included in the aggregation. If a location is used, such as London, you may only want to include devices in the London data center. This filter uses a zenoss_instance_key from dim_group, so groups need to be created and populated using the grouping procedure from the user interface. If this filter is not provided for a business hour row (see business_hours_group_assoc below), all devices are included in the aggregation.

  - filter_name: name to identify the filter
  - description: up to 255 characters to describe the filter (optional)
  - zenoss_instance_key: key of the zenoss instance, used along with device_organizer_uid to identify a device grouping in a particular zenoss instance
  - device_organizer_uid: id of the device group, used along with zenoss_instance_key to identify a device grouping in a particular zenoss instance

- meta_metric: optional filter used to reduce the set of metrics included in the aggregation. If this filter is not provided for a business hour row (see business_hours_group_assoc below), all metrics are included in the aggregation.

  - metric_key: primary key
  - metric_name: name of the metric to aggregate

- business_hours_group_assoc: each business hour row may have multiple associated job filters. Each job filter may be used by many business hour rows. For each job filter associated with a business hour row, an association record is created.

  - business_hours_config_id: id of the row in the business hours table
  - group_filter_id: id of the job filter

- business_hours_metric_assoc: each business hour row may have multiple associated metric filters. Each metric filter may be used by many business hour rows. For each metric filter associated with a business hour row, an association record is created.

  - business_hours_config_id: id of the row in the business hours table
  - metric_key: id of the metric filter

Inserting records for business hours aggregation

Perform the following steps to insert records into the reporting database. The code in this procedure is for demonstration purposes only. This example assumes that you have two data centers (Austin and Bangalore) already defined and the metrics you want to filter on are called analytics_test1 and analytics_test2:

1 Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following commands at the prompt substituting values that apply to your company's situation. The first section of the following inserts groups. If you already have groups, ignore this section and customize the commands to your situation:

```
insert into dim_group
    (group_key, sid, zenoss_instance_key, id, device_organizer_name,
 device_key,
        device_organizer_uid)
values
    (1001, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_1', 1,
 'AustinDataCenter',
        'AustinDataCenter', 1, '/zport/dmd/Groups/AustinDataCenter'),
    (1002, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_3', 1,
 'AustinDataCenter',
        'AustinDataCenter', 3, '/zport/dmd/Groups/AustinDataCenter'),
    (1003, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_2', 1,
 'BangaloreDataCenter',
        'BangaloreDataCenter', 2, '/zport/dmd/Groups/
BangaloreDataCenter'),
    (1004, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_4', 1,
 'BangaloreDataCenter',
        'BangaloreDataCenter', 4, '/zport/dmd/Groups/
BangaloreDataCenter');

insert into group_filter
    (name, description, zenoss_instance_key, device_organizer_uid)
values
    ('austin', 'Devices in the Austin data center', 1, '/zport/dmd/
Groups/AustinDataCenter'),
    ('bangalore', 'Devices in the Bangalore data center', 1,
        '/zport/dmd/Groups/BangaloreDataCenter');


insert into meta_metric (metric_name)
values ('analytics_test1'), ('analytics_test2');


insert into business_hours_config
    (name, description, shift_by, shift_start_time, shift_end_time)
values
    ('austin', 'Austin data center', '-06:00', '8:00', '16:00'),
    ('bangalore', 'Bangalore data center', '-09:30', '7:00', '17:00');

insert into business_hours_group_assoc
    (business_hours_config_id, group_filter_id)
values
    (
        (select id from business_hours_config where name='austin'),
        (select id from group_filter where name='austin')
    ),
    (
        (select id from business_hours_config where name='bangalore'),
        (select id from group_filter where name='bangalore')
    );

insert into business_hours_metric_assoc
    (business_hours_config_id, metric_key)
```

```
values
   (
      (select id from business_hours_config where name='austin'),
      (select metric_key from meta_metric where
metric_name='analytics_test1')
   ),
   (
      (select id from business_hours_config where name='austin'),
      (select metric_key from meta_metric where
metric_name='analytics_test2')
   ),
   (
      (select id from business_hours_config where name='bangalore'),
      (select metric_key from meta_metric where
metric_name='analytics_test1')
   ),
   (
      (select id from business_hours_config where name='bangalore'),
      (select metric_key from meta_metric where
metric_name='analytics_test2')
   );
```

Creating *n*th percentile calculations

Analytics allows you to create *n*th percentile calculations that can be added to a report. The *n*th percentile is the smallest value in the set of raw data with the property that *n*% of the data values are less than or equal to it. You need to define several parameters in order for the calculation to provide the targeted information you need:

- Value of percentile desired (e.g., 95th percentile)
- Number of days of data to aggregate (e.g., prior 30 days, this value is bounded by the `raw_retention_days` value)
- Business hours to calculate within (if defined, otherwise the default 24-hour "shift" is used)
- Job filter (also known as a group filter) to limit the calculation to certain devices (otherwise all devices are included)
- Metrics filter to limit the calculation to certain metrics (otherwise all metrics are used)

The creation of rows for the *n*th percentile calculation is performed manually by inserting records into the reporting database. The tables of interest are as follows:

- `nth_percentile_config`

  The main table for defining the percentile jobs.

  - `percentile` - Specifies the *n*th value to calculate; e.g., the 95th.
  - `period` - Specifies the number of days of data to use for the *n*th percentile calculation
  - `name` - Not used for the *n*th percentile; only exists for integration with an interface.
  - `description` - Not used for the *n*th percentile; only exists for integration with an interface.
- `business_hours_config`

  Specifies the time ranges that has been defined as "business day" or "shift".
- `nth_percentile_business_hours_assoc`

  Relates `nth_percentile_config` records with `business_hours_config` records.

  - `nth_percentile_config_id` - Specifies the `nth_percentile_config` foreign key.
  - `business_hours_config_id` - Specifies the `business_hours` foreign key.

Note If `nth_percentile_business_hours_assoc` is not defined,
`projection_business_hours_assoc` is used.

- `group_filter`

  Optional filter used to reduce the set of devices included in the aggregation.
- `nth_percentile_group_assoc`

  Relates `nth_percentile_config` records with `group_filter` records.

  - `nth_percentile_config_id` - Specifies the `nth_percentile_config` foreign key.
  - `group_filter_id` - Specifies the `group_filter` foreign key.

  Note If `projection_group_assoc` is also defined, only the intersection is used.

- `meta_metric`

  Optional filter used to reduce the set of metrics included in the aggregation.
- `nth_percentile_metric_assoc`

  Relates `nth_percentile_config` records with `meta_metric` records.

  - `nth_percentile_config_id` - Specifies the `nth_percentile_config` foreign key.
  - `metric_key` - Specifies the `meta_metric` foreign key.

  Note If `nth_percentile_metric_assoc` is not defined for a metric,
  `projection_metric_assoc` is used.

For each aggregate table to be projected, the table name from which to read the data is constructed as follows:

```
nth_percentile_<nth_percentile_config.name><business_hours_config.name,
  if any><meta_metric.metric_name>
```

The table name to which to write data is constructed as follows:

```
proj_<projection_config.name><nth_percentile_config.name, if
  any><business_hours_config.name, if any>_<meta_metric.metric_name>
```

Inserting records for *n*th percentile calculations

This example procedure inserts records into the reporting database and is for demonstration purposes only. For this example, assume that two data centers are already defined, Austin and Bangalore. The metrics on which to filter are called `analytics_test1` and `analytics_test2`.

1 Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following command at the prompt, substituting values that apply to your company's situation:

```
insert into nth_percentile_config
    (name, description, percentile, period)
values
    ('percentile_test_1', 'first percentile test', 90, 10),
```

```
    ('percentile_test_2', 'second percentile test', 80, 20);

insert into nth_percentile_business_hours_assoc
    (nth_percentile_config_id, business_hours_config_id)
values
    (
        (select id from nth_percentile_config where percentile = 90 and
 period = 10),
        (select id from business_hours_config where name='austin')
    ),
    (
        (select id from nth_percentile_config where percentile = 80 and
 period = 20),
        (select id from business_hours_config where name='bangalore')
    );

insert into nth_percentile_group_assoc
    (nth_percentile_config_id, group_filter_id)
values
    (
        (select id from nth_percentile_config where percentile = 90 and
 period = 10),
        (select id from group_filter where name='austin')
    ),
    (
        (select id from nth_percentile_config where percentile = 80 and
 period = 20),
        (select id from group_filter where name='bangalore')
    );

insert into nth_percentile_metric_assoc
    (nth_percentile_config_id, metric_key)
values
    (
        (select id from nth_percentile_config where percentile = 90 and
 period = 10),
        (select metric_key from meta_metric where
 metric_name='analytics_test1')
    ),
    (
        (select id from nth_percentile_config where percentile = 90 and
 period = 10),
        (select metric_key from meta_metric where
 metric_name='analytics_test2')
    ),
    (
        (select id from nth_percentile_config where percentile = 80 and
 period = 20),
        (select metric_key from meta_metric where
 metric_name='analytics_test1')
    ),
    (
        (select id from nth_percentile_config where percentile = 80 and
 period = 20),
        (select metric_key from meta_metric where
 metric_name='analytics_test2')
    );
```

Projecting other aggregates

You can project any numeric metric into the future by defining the number of days of past data to analyze along with the number of days into the future you want to predict. For example, you want to predict the next 30 days of an *n*th percentile calculation based on the last 60 days of data. This prediction returns a value for every day for the next 30 days. In order to accomplish this, a table (`projection_config`) is defined in the reporting database for the projection configuration of the future values.

Projections are performed using a polynomial function, which is beneficial when data follows a curved pattern toward a maximum or minimum value, such as percentage of disk storage used over time.

The `projection_config` table has the following form:

- `name` - Name to identify the projection; e.g., `bangalore_60_30`, `austin_95th`.
- `description` - Up to 255 characters to describe the projection.
- `nth_percentile_config_id` - ID of the row in the `nth_percentile_config` table on which to base a projection (only used for *n*th percentile projections.
- `days_past` - Integer indicating the number of past days to use in making the calculation.
- `days_future` - Integer indicating the number of days into the future to make projections.
- `agg_column_name` - Aggregation column name.

Projections run each day at the completion of aggregation and *n*th percentile processing. Resultant tables from projection computations have the same structure as aggregation tables with the addition of a column with the timestamp of when the projection was performed. For each aggregate table to be projected, the name of the table from which to read data is constructed as follows:

```
daily_<business_hours_config.name, if any>_<meta_metric.metric_name>
```

The name of the table to which to write data is constructed as follows:

```
proj_<projection_config.name><business_hours_config.name, if
  any><meta_metric.metric_name>
```

Each projection row may have one or more associated job filters and metric filters. A job filter reduces the device sample set by allowing you to create a device group and assigning devices to that group. Several job filters may be defined for a given projection. Each filter is additive and increases the number of the devices included in the sample set. If no job filters are defined, then all devices are included in the sample set.

In addition to the group filters, you may also define and associate metric filters which restrict the metrics processed in the projection. If no metric filters are defined, all metrics are processed. Each metric processed is run in a separate thread. The projection calculations use data from the `raw_v2` tables. To accommodate the maximum flexibility in which data can be used, the metric name in this context is the name of the daily table without the preceding "`daily_`". For normal aggregated metrics, this equates to the metric name (e.g., `cpu__pct`). For specially aggregated (computed) data, such as business hours or *n*th percentile, the metric name includes the computed name (such as `austin_cpu__pct`).

Inserting records for daily aggregation projections

Perform the following steps to insert records into the reporting database. The code in this procedure is for demonstration purposes only. This example assumes that two data centers (Austin and Bangalore) already defined and the metrics you want to filter on are called `analytics_test1` and `analytics_test2`.

1 Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

```
mysql -u root reporting
```

2 Enter the following command at the prompt substituting values that apply to your company's situation:

```
insert into projection_config
    (name, days_past, days_future, agg_column_name)
values
    ('austin_60_30', 60, 30, 'fct_avg'),
    ('bangalore_60_30', 60, 30, 'fct_avg'),

insert into projection_business_hours_assoc
    (projection_config_id, business_hours_config_id)
values
    (
       (select id from projection_config where name='austin_60_30'),
       (select id from business_hours_config where name='austin')
    ),
    (
       (select id from projection_config where name='bangalore_60_30'),
       (select id from business_hours_config where name='bangalore')
    );

insert into projection_group_assoc
    (projection_config_id, group_filter_id)
values
    (
       (select id from projection_config where name='austin_60_30'),
       (select id from group_filter where name='austin')
    ),
    (
       (select id from projection_config where name='bangalore_60_30'),
       (select id from group_filter where name='bangalore')
    );

insert into projection_metric_assoc
    (projection_config_id, metric_key)
values
    (
       (select id from projection_config where name='austin_60_30'),
       (select metric_key from meta_metric where
 metric_name='analytics_test1')
    ),
    (
       (select id from projection_config where name='austin_60_30'),
       (select metric_key from meta_metric where
 metric_name='analytics_test2')
    ),
    (
       (select id from projection_config where name='bangalore_60_30'),
       (select metric_key from meta_metric where
 metric_name='analytics_test1')
    ),
    (
       (select id from projection_config where name='bangalore_60_30'),
       (select metric_key from meta_metric where
 metric_name='analytics_test2')
    );
```

Inserting records for *n*th percentile projections

Similar to the previous section, you can insert records that allow *n*th percentile calculations. The following is a sample insertion to the `projection_config` table for a 95th percentile projection:

```
insert into projection_config
    (name, nth_percentile_config_id, days_past, days_future,
 agg_column_name)
values
    ('austin_95th', 1, 60, 30, 'fct_percentile')
```

where `1` is the ID of the row in the `nth_percentile_config` table where the Austin 95th percentile calculation is defined.

Third party licenses

A

Zenoss Analytics incorporates third party components whose licenses require us to mention their inclusion. For more information, see the following sites:

- Antlr - *http://www.antlr.org/*
- AspectJ weaver - *https://eclipse.org/aspectj/*
- Java Specification Request (JSR) 250 - *https://jcp.org/en/jsr/detail?id=250*
- connector-api-1.5

ERD diagram for Analytics 5.1.x

B

The following image shows the Entity-Relationship Diagram (ERD) for Analytics 5.1.x. You will find the relationships shown between tables in the database as well as a sample of dynamically created dimension tables.

Figure 11: Analytics ERD