Smarter Balanced Reporting (RFP 15)
Runbook, Deployment, Administration

Prepared for:

by:

Approvals

Representing	Date	Author	Status
Consortium		Joe Willhoft	
Consortium	2014.09.24	Brandt Redd	Approved for Milestone 5
PMP	2014.09.24	Kevin King	Endorsed for Milestone 5
Workgroup	2014.09.23	Henry King	Endorsed for Milestone 5

Revision History

Revision Description	Author/Modifier	Date
Initial Release (DRAFT)	Anna Grebneva (Amplify)	2014.07.09

Table of Contents

1 Summary

This document is for system administrators who will be operating the system. It contains instructions on how to install, scale, and maintain Smarter Balanced Data Warehouse and Reporting instance.

2 Data Capacity Planning Assumptions

1. Smarter Balanced Reporting Database Storage
    a. Student Registration: 0.5kb per student
    b. Student Assessments: 2kb per assessment and 0.5kb per student

   c. Allocate similar size to whole postgresql database for postgresql point in time recovery
   d. Keep disk usage less than 50% to help Database Administrator to perform postgres vacuum operations efficiently.
2. PDF file Storage
   Smarter web application will pre-generate grayscale PDF, and color PDF will be generated when a user requests. Therefore, it is good practice to estimate disk space for 100% disk storage for grayscale PDF files and 50% disk storage for color PDF.
   a. color - 85kb/assessment outcome.
   b. grayscale - 75kb/assessment outcome.
3. Historical Assessment Archive
   a. 50 mb assessment csv for 100k per assessment
4. Individual Item Response (a.k.a. Item-Level) Data Storage
   a. Average size of file after compression is 200kb.
   b. A student generates around 5 item-level files per school year.
   c. SBAC requires storing of 10 years of student item level xml raw data.
5. HTTP Pickup Zone Storage
   a. [PLACEHOLDER: performance and load testing scheduled as part of M6]
   b. Database for pickup zone operation.
6. Landing Zone Storage:
   a. 0.5kb per student per assessment.
7. Loader Database Storage:
   a. [PLACEHOLDER: performance and load testing scheduled as part of M6]

3 Hardware Minimum Requirements

The number of instances required for each machine type is correlated to three independent criterias: the number of tenants/states in the system, the number of users accessing the system, and the number of students in the tenant/state.

Table 1 - Sample configuration for data warehouse and reporting for 1 tenant containing 2M students and 100K users.

Machine	Avg Usage (3% user concurrency) instances	Peak Usage (10% user concurrency) instances	Cores (vCPU per instance)	Memory (GB per instance)	non-OS portion Storage (GB)
web servers[2]	9	30	4	8	
load balancers[2]	2	2	2	2	
database	1	1	8	64	1000

masters[1]					
database replicas for web front end[2]	2	7	8	64	1000
database replicas for extraction[2]	1	3	8	64	1000
db load balancer[1]	2	2	8	64	
db pool[2]	2	3	4	8	
cache[3]	2	2	8	64	
pdf messenger[2]	2	2	4	8	
pdf workers[3]	2	2	8	64	
pdf generator[1]	1	1	2	8	
extract messaging[2]	2	2	4	8	
extract worker[2]	2	7	8	32	
cache warmer[1]	1	1	2	8	
landing zone[1]	2	2	4	8	50
pickup zone[5]	3	9	4	8	
pickup zone database[5]	1	1	4	8	10
loader[3]	2	2	4	8	
loader messenger[1]	2	2	4	8	
loader database[1]	1	1	8	32	1000
migrator[1]	1	1	4	32	
database staging[1]	1	1	8	32	1000
gluster (storage)[3]					26480[4]

[1] The number of instances for this machine type is proportional to every one Tenant/State.
[2] The number of instances for this machine type is proportional to every 100,000 Users.
[3] The number of instances for this machine type is proportional to every 2,000,000 Students.
[4] The number of storage for this is proportional to 10 years of item level data storage plus 1 years of pdf.
[5] The number of storage for this is proportional to proportional to every 100,000 Users.

4 Software Requirements

The system has been tested with servers running on CentOS 6.4. A brief summary of software requirements for each machine type is listed in the table below. Please refer to other sections of this document for more details.

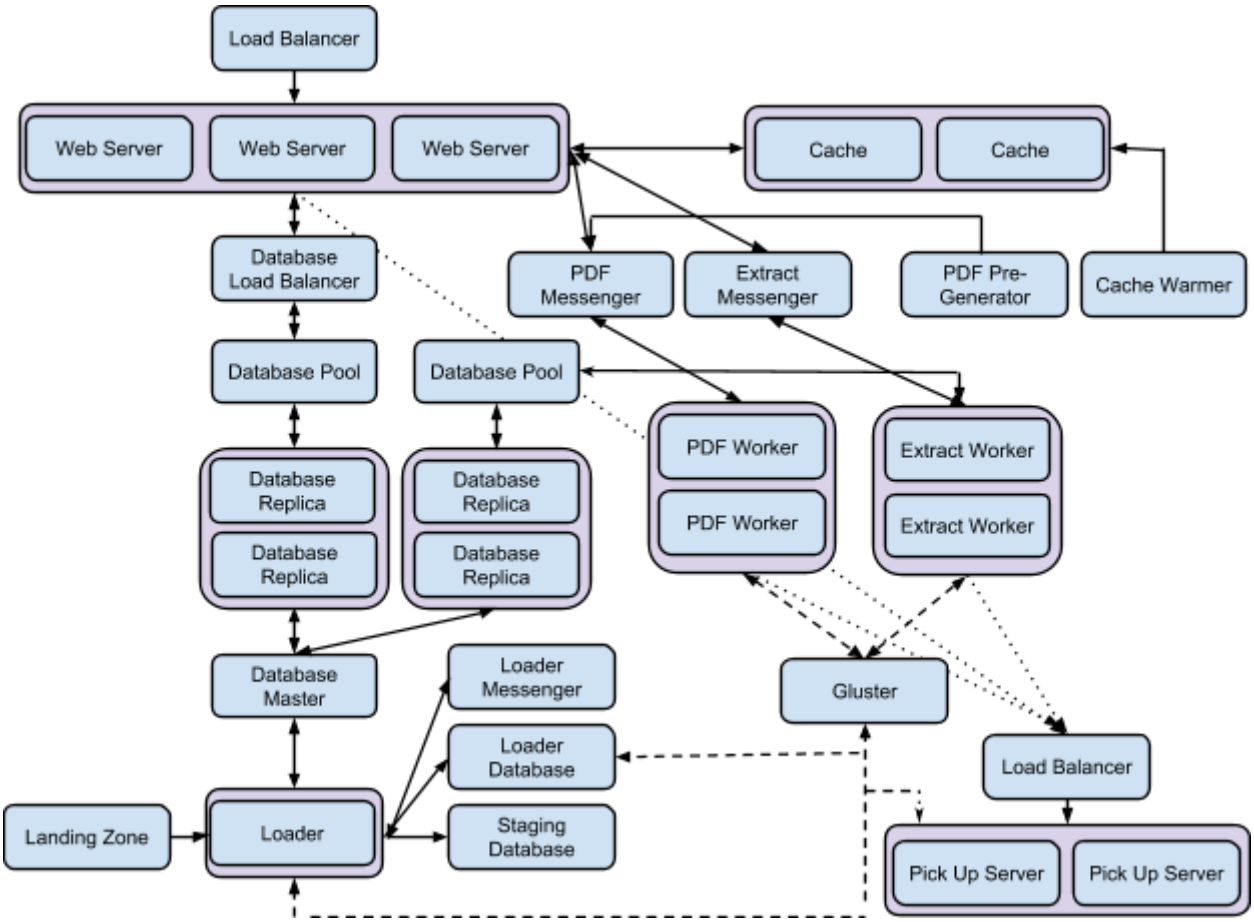
Table 2 - Machine Types with their software requirements

Machine	External Software (Available from Linux distributors)	Software provided by Amplify
web servers	httpd xmlsec1 xmlsec1-openssl-devel	python3-3 python3-mod_wsgi smarter
http landing zone	httpd mod_xsendfile xmlsec1-openssl-devel xmlsec1 glusterfs glusterfs-fuse	python3-mod_wsgi3-3 hpz
http landing zone database	postgresql92 postgresql92-libs postgresql92-server postgresql92-contrib	
load balancers		
database masters	postgresql92 postgresql92-libs postgresql92-server postgresql92-contrib repmgr	
database replicas for web reporting	postgresql92 postgresql92-libs postgresql92-server postgresql92-contrib repmgr xmlsec1 xmlsec1-openssl-devel	python3-3 python3-mod_wsgi smarter
database replicas	postgresql92	

for extraction	postgresql92-libs postgresql92-server postgresql92-contrib repmgr	
db load balancer	pgbouncer postgres92 postgres92-server xinetd	
db pool with failover for web front end	pgpool-II	
db pool without failover for bulk extract	pgpool-II	
cache	memcached	
pdf messenger	rabbitmq-server	
pdf workers	glusterfs glusterfs-fuse fuse-encfs urw-fonts	wkhtmltopdf wkhtmltopdf-qt pdfunite
pdf generator	httpd xmlsec1 xmlsec1-openssl-devel	python3-3 python3-mod_wsgi smarter
extract messaging	rabbitmq-server	
extract worker	glusterfs glusterfs-fuse fuse-encfs xmlsec1 xmlsec1-openssl-devel	python3-3 python3-mod_wsgi smarter
cache warmer	httpd xmlsec1 xmlsec1-openssl-devel	python3-3 python3-mod_wsgi smarter
landing zone		python3-3 edsftp
loader	glusterfs glusterfs-fuse fuse-encfs	python3-3 edudl
loader messenger	rabbitmq-server	
loader database	postgresql92	

	postgresql92-libs postgresql92-server postgresql92-contrib glusterfs glusterfs-fuse fuse-encfs	
migrator	xmlsec1 xmlsec1-openssl-devel rabbitmq-server	python3-3 python3-mod_wsgi smarter
database staging	postgresql92 postgresql92-libs postgresql92-server postgresql92-contrib	
gluster (storage)	glusterfs-geo-replication glusterfs glusterfs-server glusterfs-fuse	

The following diagram illustrates the architecture of and relationships between the components listed above:


Load Balancer
Web Server
Web Server
Web Server
Cache
Cache
Database Load Balancer
PDF Messenger
Extract Messenger
PDF Pre-Generator
Cache Warmer
Database Pool
Database Pool
PDF Worker
PDF Worker
Extract Worker
Extract Worker
Database Replica
Database Replica
Database Replica
Database Replica
Database Master
Loader Messenger
Loader Database
Staging Database
Gluster
Load Balancer
Landing Zone
Loader
Pick Up Server
Pick Up Server

5 Planning for Availability

The reporting application is designed for high availability and each installation can be setup to provide the service with minimal interruption.

Following architectural diagram is a sample configuration intended to minimize administrative involvement to failover process.

Table 1. Sample Configuration for Smarter Balanced Reporting

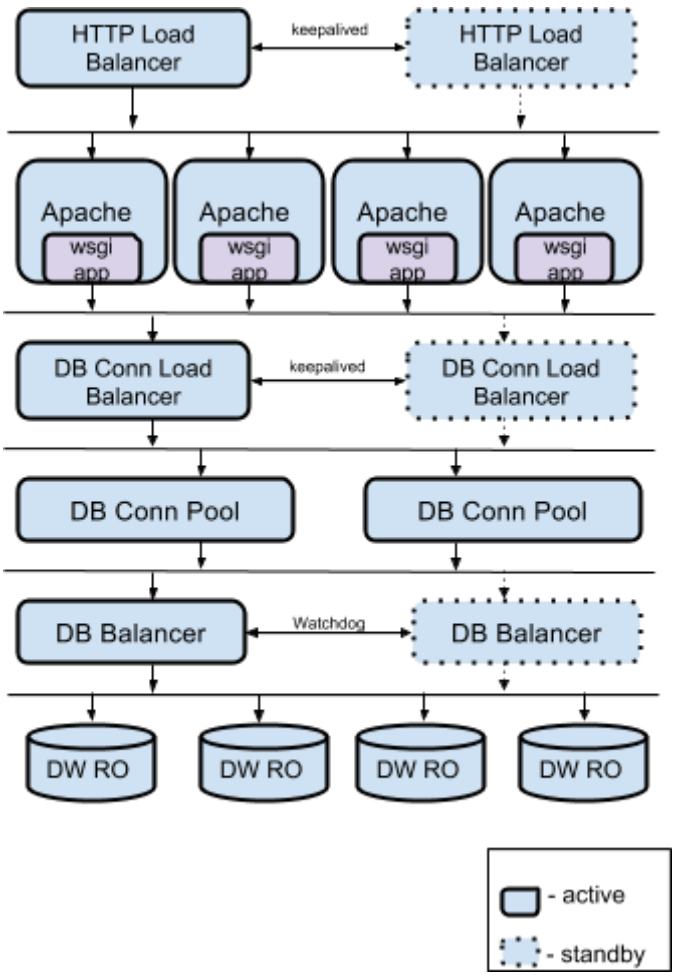
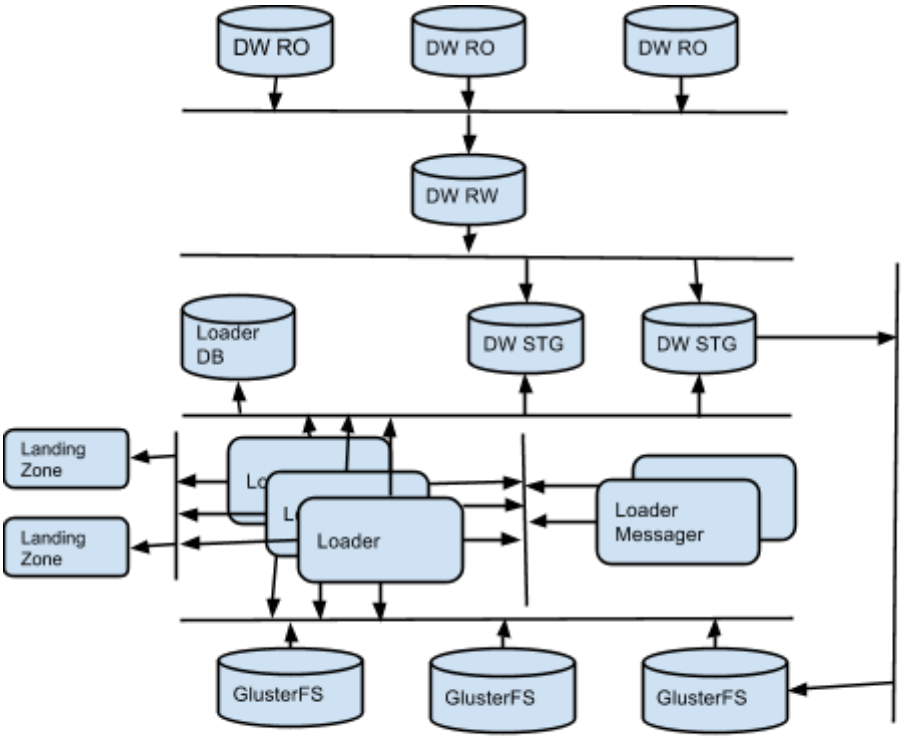


Table 2. Sample Configuration for Smarter Data Loader



1. HA Proxy
    a. HTTP Load Balancer
    b. DB Conn Load Balancer
    Using HA-Proxy as Load balancers for both HTTP and database. When active HTTP Load balancer goes down, another load balancer will take over the virtual IP address and start serving as new active load balancer. Load balancers monitor each other when keepalived is enabled.
        i. keepalived rpm and virtual IP are required to setup.
        ii. when one load balancer goes down, other load balancer automatically takes over virtual IP.

2. Apache
    All web servers are registered to HA Proxy. When a web server goes down, HA Proxy will not route a request. When a web server comes back online, HA Proxy will resume routing a request.

3. DB Connection Pool

   a. PgBouncer
      All PgBouncer servers are registered in DB Connection Load Balancer configuration. When DB Connection Pool server is down, DB Connection Load Balancer will route within active DB Connection Pool servers.

4. DB Bouncer
   a. PgPool-II
      Version of PgPool-II must be 3.3 or greater. PgPool-II uses “watchdog” to monitor each other. An active PgPool-II server uses virtual IP address to receive all database requests and forward to specific PostgreSQL servers by roundrobin. Standby PgPool-II becomes active and takes over virtual IP when an active PgPool-II goes down.

5. Memcached:
   a. Memcached supports clustering. By setting up clustering, it provides load balancing and redundancy. Configuration guidance cna be found in the following external documentation:
      https://code.google.com/p/memcached/wiki/NewConfiguringServer

6. PostgresQL 9.2
   a. The Smarter Balanced Data Warehouse uses multiple read-only replication database to provide redundancy. The failover for database is via PgBouncer and PgPool in previous sections.
   b. The data loader can be configured with multiple staging databases to parallelize data injection. This also provides failover for data loading pipeline when a subset of staging database is not in available.

7. Data Loader
   a. The data loader can be configured with multiple processing servers to process data loading job in parallel. This provides redundancy and load balancing in data loading process.

8. Data Loader Messenger
   a. The data loader can be configured with RabbitMQ clustering to provide redundancy and failover and load balancing for data loading process.

9. Landing Zone:
   a. The Landing Zone(s) can be configured with multiple landing zone servers with a load balancer in front of them to provide redundancy and failover.

10. NTP servers for Smarter Balanced Reporting Cluster network

a. Redundant NTP server is important to guarantee integrity of data migration and backup, by ensuring all clocks and machines are in sync. For more information, please refer to: http://blog.meinbergglobal.com/2013/11/27/ntp-network-redundancy/

11. DNS server for HA Proxy Cluster network
    a. Please set up redundant DNS servers for internal network DNS lookups. This can be achieved with common Name server software such as BIND.
    b. You can refer https://www.digitalocean.com/community/tutorials/how-to-install-the-bind-dns-server-on-centos-6

12. Storage Area Network:
    a. DRBD will be useful for setting up redundant storage area network for Smarter Balanced Reporting cluster.
    b. if your system uses iSCSI, please refer here for redundancy for SANs. http://porky.linuxjournal.com:8080/LJ/217/11275.html
    c. if your system use Fiber Channel, you still can refer to iSCSI section plus

13. RAID on disk drives
    a. If cost is not an issue. Please try to implement RAID 1+0, otherwise, the next best options if hardware or software supports is RAID 6
    b. If RAID disk controllers are too costly, Please use Linux Software RAID to create redudancy. Please refer to https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/System_Administration_Guide/Software_RAID_Configuration.html

6 Archival, Backup and Recovery

1. Backup and restore PostgreSQL (**manual processes**)
   Use PostgreSQL master database to backup database.
   Depending on the size of the database, it may take several hours.
   1. Stop EdMigrate service
      `/etc/init.d/edmigrate-conductor stop`
   2. extract PostgreSQL data (Please refer man for pg_dump)
      `pg_dump -f /path/to/backup.sql.gz -Z 9 -U edware -W edware`
   3. Resume Edmigrate serivce
      `/etc/init.d/edmigrate-conductor start`

   To restore Smarter Balanced Reporting System database
   1. Stop EdMigrate service
      `/etc/init.d/edmigrate-conductor stop`

2. restore PostgreSQL data (Please refer man for pg_restore)
   `pg_restore -U edware -W -d edware /path/to/backup.sql.gz`
3. Resume Edmigrate service
   `/etc/init.d/edmigrate-conductor start`

2. Please check http://www.slideshare.net/InesSombra/data-antipatterns-nycdevops for more information on the following process:
   a. Provisioning enough disk space for database backup.
   b. Proper backup schedule for full backup and incremental backup, see https://wiki.postgresql.org/wiki/Incrementally_Updated_Backups, http://stackoverflow.com/questions/5529603/best-method-for-postgres-incremental-backup and http://www.postgresql.org/docs/9.2/static/continuous-archiving.html for good incremental backup for PostgresQL
   c. Set up good backup retention time.
   d. Periodically practice restore database backups to detect corrupted backup files.
   e. Alwasy test restore backup with smarter app to make sure restoration from backup is proper.
   f. Set up recovery protocols, plans and practice disaster recovery.
3. Historical Assessments archive:
   a. Periodically back up history assemssment files in landing zone history to tapes/optical media/amazon glacier
4. Item-Level raw data:
   a. We suggest setting up rsync and redundant item level storage cluster as backup plan due to number of files and size of data.
   b. Periodically backup item-level files to tape/optical media/amazon glacier.
   c. Item level data file may be updated/modified, so it is advisable to plan incremental backup on whole file repository periodically. Make sure the gap for missed item-level raw data as minimal as possible
   d. Tar may not be usuable in backup item-level data. see http://serverfault.com/questions/329273/store-and-backup-200-million-small-files

7 Installation & Configurations

7.1 Web Servers

Web Servers are responsible for running Smarter Balanced Reporting Web Application. This application is written in Python, and runs on Apache using WSGI.

7.1.1 Installation

The following table lists the RPMs that must be installed on web servers.

Package	Tested Version	RPM Name	Description
Python	3.3	python3	Python 3 RPM is generally not available from Linux distributions, hence Amplify has supplied a customized Python 3 RPM
Apache	2.2.15	httpd	Apache hosts Smarter Web Application. It can be installed from the standard distributed RPM. We highly recommend using customized configuration. Apache should run on worker MPM mode.
WSGI	3.4	python3-mod_wsgi	WSGI defines a simple and universal interface between web servers and web applications or frameworks for Python. This is a customized RPM supplied by Amplify.
Smarter	TBD	smarter	Amplify's Smarter Web Application
XMLSec	1.2.16	xmlsec1 xmlsec1-openssl-devel	XMLSec can be installed from standard distributed RPM. It is used for verification of SAML Responses.

7.1.2 Configuration

7.1.2.1 Apache

We highly recommend using our customized configurations for Smarter Web Application. The nature of the application is CPU intensive, and requires Apache to run in worker MPM (Multi-Processing Module) mode.

7.1.2.1.1 Configuration for worker MPM Mode

Modify **/etc/sysconfig/httpd** to set Apache to use worker MPM mode.

```
# Configuration file for the httpd service.
HTTPD=/usr/sbin/httpd.worker
```

7.1.2.1.2 Configuration for Apache Web Server

Modify **/etc/httpd/conf/httpd.conf** to set the main configuration for Apache

```
ServerTokens Prod
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
```

```
Timeout 120
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 15
Listen 80

LoadModule authz_host_module modules/mod_authz_host.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule logio_module modules/mod_logio.so
LoadModule env_module modules/mod_env.so
LoadModule deflate_module modules/mod_deflate.so
LoadModule headers_module modules/mod_headers.so
LoadModule mime_module modules/mod_mime.so
LoadModule status_module modules/mod_status.so
LoadModule vhost_alias_module modules/mod_vhost_alias.so
LoadModule dir_module modules/mod_dir.so
LoadModule alias_module modules/mod_alias.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule rewrite_module modules/mod_rewrite.so
Include conf.d/*.conf
User apache
Group apache
ServerAdmin root@localhost
UseCanonicalName Off
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b %>D \"%{Referer}i\" \"%{User-Agent}i\""
combined_with_duration
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log combined env=!dontlog
ServerSignature Off
AddDefaultCharset UTF-8
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
AddHandler type-map var
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[0123]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully
```

7.1.2.1.3 Configuration for security certification

Modify **/etc/httpd/conf.d/edware.conf** to configure security certification

```
RewriteEngine        on
RewriteCond %{HTTPS}            !=on
```

```
RewriteCond %{REQUEST_URI}        !^/6e34dffe3b7f3f7b182489a67fe12acf1e3a24ff
RewriteCond %{HTTP_USER_AGENT}    !^haproxy-check
RewriteCond %{HTTP_USER_AGENT}    !^check_http
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}

NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/pki/tls/certs/server.crt
    SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt
    SSLCertificateKeyFile /etc/pki/tls/private/server.key
    SSLCipherSuite RC4-SHA:AES128-SHA
    SSLHonorCipherOrder on
    MaxKeepAliveRequests 500
    KeepAliveTimeout 2
    Servername <webServerHost>
    ServerAdmin <admin>

    LogLevel info

    RewriteEngine On
    RewriteRule ^/$ /assets/html/index.html [R]
</VirtualHost>
```

7.1.2.1.4 Configuration for SSL

Modify **/etc/httpd/conf.d/ssl.conf** to configure SSL

```
LoadModule ssl_module modules/mod_ssl.so
Listen 443
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
SSLPassPhraseDialog  builtin
SSLSessionCache         shmcb:/var/cache/mod_ssl/scache(512000)
SSLSessionCacheTimeout  300
SSLMutex default
SSLRandomSeed startup file:/dev/urandom  256
SSLRandomSeed connect builtin
SSLCryptoDevice builtin
SSLProtocol all -SSLv2
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/private/server.key
SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt
SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0
<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
```

7.1.2.2 WSGI

mod_wsgi is an Apache module which can host any Python applications which supports Python WSGI interface. Smarter is deployed as an application using mod_wsgi, and mod_wsgi must be configured to load with Apache.

7.1.2.2.1 Configuration for loading module

Modify **/etc/httpd/conf.d/wsgi.conf** to load mod_wsgi for Apache

```
LoadModule wsgi_module modules/mod_wsgi.so
```

7.1.2.2.2 Configuration for Smarter application specific WSGI configuration

Modify **/etc/httpd/conf.d/wsgi_edware.conf** to add WSGI configuration for Smarter Web Application.

```
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
WSGIDaemonProcess pyramid user=apache group=apache processes=2 threads=30 python-
path=/opt/virtualenv/smarter/lib/python3.3/site-packages
WSGIScriptAlias / /opt/edware/smarter/smarter.wsgi
WSGIImportScript /opt/edware/smarter/smarter.wsgi process-group=pyramid application-
group=%{GLOBAL}
WSGISocketPrefix run/wsgi
<Directory /opt/virtualenv/smarter>
   WSGIProcessGroup pyramid
   Order allow,deny
   Allow from all
</Directory>
WSGIPythonPath /opt/virtualenv/smarter/lib/python3.3/site-packages
```

7.1.2.2.3 Configuration for WSGI to load Smarter's Configuration

Modify **/opt/edware/smarter/smarter.wsgi** to assign the path of Smarter's INI configuration file.

```
import site
from pyramid.paster import get_app, setup_logging
# The path below is used for symbolic link to development.ini
ini_path = '/opt/edware/conf/smarter.ini'
setup_logging(ini_path)
```

7.1.2.3 Smarter

Smarter is the main web application and provides HTML, CSV and PDF data access to end users. The applications run on Apache web server via WSGI with Python Pyramid Framework.

The Smarter RPM, provided by Amplify, packages an entire Python 3.3 virtual environment and all Python dependencies used by the application. Smarter RPM installs the virtual environment in /opt/virtualenv/smarter. An utility, packaged within the RPM, is provided to generate INI configuration file for Smarter Web Application. The generation of the INI file is the administrator/operator's responsibility.

7.1.2.3.1 Generating smarter.ini

Smarter needs to read an INI configuration upon start-up. The default location of this file is /opt/edware/conf/smarter.ini, and the configuration of the path is specified in /opt/edware/smarter/smarter.wsgi.

Each environment is unique, hence, the configuration of each configuration is unique. The operator must generate the INI file for each server/environment. Within the directory, /opt/edware/conf/, a settings.yaml file exists that defines default and environment specific key/value pairs of configuration. For permanent changes to configurations for an environment, changes should be made in /opt/edware/conf/settings.yaml.

Here are the steps to generating the INI file for the environment, “uat”,

```
shell>  . /opt/virtualenv/smarter/bin/activate
(virtualenv)  cd /opt/edware/conf
(virtualenv)  python generate_ini.py -e uat
(virtualenv)  mv uat.ini smarter.ini
```

7.1.2.3.2 Configuration File for Authentication between OpenAM SAML and Smarter

Smarter Web Application needs to verify the authentication with OpenAM SAML by IDP Metadata XML file. This metadata file doesn’t come with the RPM. The operator must generate and reference the path of this file inside smarter.ini. By default, Smarter expects this file to be located in **/opt/edware/conf/idp_metadata.xml.**

7.1.2.3.2 Adding Custom Metadata Configuration

Every tenant/state has the option to insert custom metadata into their production database’s custom_metadata table. This table has three columns:

Column Name	Description	Example Value
state_code	The state code of a particular tenant.	NC
asmt_subject	The subject for a particular assessment	Math
asmt_custom_metadata	A JSON formatted string that contains the minimum cell size, and colors used for assessment representation within reports.	{"min_cell_size":30, "colors":[{"text_color":"#ffffff","bg_color":' 5B","end_gradient_bg_color":"# {"text_color":"#000","bg_color":' 3","end_gradient_bg_color":"#e {"text_color":"#ffffff","bg_color":' 3","end_gradient_bg_color":"#6 {"text_color":"#ffffff","bg_color":' ","end_gradient_bg_color":"#3a

The insertion of custom metadata is optional, as default values will be used instead.

7.1.2.3.3 Configuring syslog

Optional rsyslog configuration file, **/etc/rsyslog.d/wgen.conf**, can be created by an operator. All regular files in the /etc/rsyslog.d directory are included as additional rsyslog server setting for /etc/rsyslogd.conf. If syslog local facilities are not specified, they are sent to syslog message facility.

```
$FileCreateMode 0644
local0.* /opt/edware/log/audit.log
local1.* /opt/edware/log/smarter.log
local2.* /opt/edware/log/security_event.log
```

7.1.2.3.4 Creating a Database Schema for Production Database

You must manually create an empty schema for production database. A script is provided to perform this task. This step requires that Database Master is installed and configured first.

```
shell>  . /opt/virtualenv/smarter/bin/activate
(virtualenv)  cd /opt/virtualenv/smarter/lib/python3.3/site-packages/edschema-0.1-py3.3.egg/edschema
(virtualenv) python metadata_generator.py -s edware -d edware -m edware --host [dbMastHostName] -p [password]
```

Note: The command is in the format of:

```
python metadata_generator.py -s [schemaName] -d [databaseName] -m edware --host [dbMastHostName] -p [password]
```

7.2 Load Balancer

Load Balancers are used to load balance traffic to web servers. Any standard load balancing solution is sufficient for this purpose.

7.3 PDF Messenger

PDF Messenger hosts the broker for PDF tasks that are requested by Smarter Web Application. We have chosen to use RabbitMQ as the message broker. RabbitMQ, written in Erlang, implements the Advanced Message Queuing Protocol (AMQP) standard.

7.2.1 Installation

The following table lists the RPMs that must be installed on PDF Messenger server.

Package	Tested Version	RPM Name	Description
RabbitMQ	2.6.1	rabbitmq-server	RabbitMQ RPM is available from Extra Packages for Enterprise Linux (EPEL)

7.2.2 Configuration

RabbitMQ

RabbitMQ RPM comes with default built-in configurations which are sufficient for running the service effectively. We recommend using RabbitMQ cluster to prevent a single point of failure. Please see http://www.rabbitmq.com/clustering.html for more details.

The main configuration file for RabbitMQ is located in **/etc/rabbitmq/rabbitmq.config**. Please refer to http://www.rabbitmq.com/configure.html for more details.

Please remember that any configurations set here (such as user name, passwords, etc.) needs to be reflected back in your smarter.ini file in your web server, pdf worker, and pdf generator machines.

Adding a New User

A user needs to be created to authenticate with applications with RabbitMQ. To add a new user, execute the commands below with root access, please refere to smarter.ini for user name and password. We default use user 'edware' and password 'edware1234'

```
shell>  rabbitmqctl add_user <user> <password>
```

Adding a New vhost

A new virtual host needs to be added so that PDF Generator has a separation with other applications that may utilize the same broker. By default, Smarter Web Applications expects this virtual host to be named, services. To add a new virtual host, execute the following commands below with root access,

```
shell> rabbitmqctl add_vhost services
```

Granting User Permission

The user that you have created above needs to have permission to your virtualhost. To grant permission, execute the following command below, replace <user> with the one with proper permission

```
shell> rabbitmqctl set_permissions -p services <user> ".*" ".*" ".*"
```

Enabling SSL (Optional)

If you decide to run RabbitMQ on SSL, you will need to enable Celery on SSL in your PDF Worker (described later).

Generate self signed certificate and key with OpenSSL

You will need to generate the certificate with the following commands,

```
shell> cd ~/
shell> mkdir testca
shell> mkdir certs private
shell> chmod 700 private
shell> echo 01 > serial
shell> touch index.txt
shell> vi testca/openssl.cnf
        [ ca ]
        default_ca = testca
        [ testca ]
        dir = .
        certificate = $dir/cacert.pem
        database = $dir/index.txt
        new_certs_dir = $dir/certs
        private_key = $dir/private/cakey.pem
        serial = $dir/serial
        default_crl_days = 7
        default_days = 365
        default_md = sha1
        policy = testca_policy
        x509_extensions = certificate_extensions
        [ testca_policy ]
        commonName = supplied
        stateOrProvinceName = optional
        countryName = optional
        emailAddress = optional
        organizationName = optional
        organizationalUnitName = optional
        [ certificate_extensions ]
        basicConstraints = CA:false
        [ req ]
        default_bits = 2048
        default_keyfile = ./private/cakey.pem
        default_md = sha1
        prompt = yes
        distinguished_name = root_ca_distinguished_name
        x509_extensions = root_ca_extensions
        [ root_ca_distinguished_name ]
        commonName = hostname
        [ root_ca_extensions ]
        basicConstraints = CA:true
        keyUsage = keyCertSign, cRLSign
        [ client_ca_extensions ]
        basicConstraints = CA:false
        keyUsage = digitalSignature
        extendedKeyUsage = 1.3.6.1.5.5.7.3.2
        [ server_ca_extensions ]
        basicConstraints = CA:false
        keyUsage = keyEncipherment
        extendedKeyUsage = 1.3.6.1.5.5.7.3.1
```

Generate self-signed CA

You will need to generate a self-signed certificate authority with the following command,

```
shell> openssl req -x509 -config openssl.cnf -newkey rsa:2048 -days 365 -out cacert.pem -outform
PEM -subj /CN=MyTestCA/ -nodes
```

Generate Server Certificate

You will need to generate a server certificate with the following commands,

```
shell>  cd ~/
shell>  mkdir server
shell>  cd server
shell>  openssl genrsa -out key.pem 2048
shell>  openssl req -new -key key.pem -out req.pem -outform PEM -subj /CN=$(hostname)/O=server/ -
nodes
shell>  cd ../testca
shell>  openssl ca -config openssl.cnf -in ../server/req.pem -out ../server/cert.pem -notext -
batch -extensions server_ca_extensions
```

Generate Client Certificate

You will need to generate a client certificate, which be installed by clients such as PDF Worker.

```
shell>  cd ~/
shell>  mkdir client
shell>  cd client
shell>  openssl genrsa -out key.pem 2048
shell>  openssl req -new -key key.pem -out req.pem -outform PEM -subj /CN=$(hostname)/O=client/ -
nodes
shell>  cd ../testca
shell>  openssl ca -config openssl.cnf -in ../client/req.pem -out ../client/cert.pem -notext -
batch -extensions client_ca_extensions
```

Configure and Enable SSL on RabbitMQ

1. Update /etc/rabbitmq/rabbitmq.config

```
[
        {rabbit, [
                {tcp_listeners, []},
                {ssl_listeners, [5671]},
                {ssl_options, [{cacertfile,"/etc/rabbitmq/testca/cacert.pem"},
                               {certfile,"/etc/rabbitmq/server/cert.pem"},
                               {keyfile,"/etc/rabbitmq/server/key.pem"},
                               {verify,verify_peer},
                               {fail_if_no_peer_cert,false}]}
]}]
```

2. Restart rabbitmq server

```
/etc/rc.d/init.d/rabbitmq-server restart
```

3. Configure Firewall, change **/etc/sysconfig/iptables**, add following lines before REJECT rules

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5671 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 15672 -j ACCEPT
```

7.3 PDF Worker

PDF Worker is responsible for producing PDF versions of reports. The worker picks up messages/tasks from the PDF Messager via RabbitMQ. PDF Worker is written using a Python framework, Celery. Celery is an asynchronous task queue/job queue based on distributed message passing. PDFs are stored on a volume on glusterFS encrypted with encFS.

7.3.1 Installation

PDF Messenger, internally known as celeryd-services, is packaged insider Smarter RPM, hence, the prerequisites for Smarter is also required in PDF Worker servers.

The following table lists the RPMs that must be installed on PDF Worker servers.

Package	Tested Version	RPM Name	Description
Python	3.3	python3-3.3.0	Python 3 RPM is generally not available from Linux distributions, hence Amplify has supplied a customized Python 3 RPM
WSGI	3.4	python3-mod_wsgi	WSGI defines a simple and universal interface between web servers and web applications or frameworks for Python. This is a customized RPM supplied by Amplify.
Smarter	TBD	smarter	Amplify's Smarter Web Application
XMLSec	1.2.16	xmlsec1 xmlsec1-openssl-devel	XMLSec can be installed from standard distributed RPM. It is used for verification of SAML Responses.
wkhtmltopdf	0.11.0	wkhtmltopdf	A third party utility for PDF generation of reports using webkit rendering engine and Qt. Amplify provides a customized RPM of wkhtmltopdf optimized for our needs.
wkhtmltopdf-qt	4.8.4	wkhtmltopdf-qt	Amplify provides a customized RPM of wkhtmltopdf-qt.
PDFUnite	0.26.1	pdfunite	A third party utility for merging PDFs. Amplify provides a customized RPM for PDFUnite.
urw-fonts	2.4	urw-fonts	Free versions of 35 standard

			PostScript fonts. This RPM is readily available from Linux Distributions.
PDF Fonts	<placeholder>	<placeholder>	Fonts for other languages are required so that PDFs are generated properly for such languages.
glusterfs	3.3.1	glusterfs	We suggest installing a newer version of the RPM from GlusterFS' site other than the RPM from Linux distributors. See below for more details.
glusterfs-fuse	3.3.1	glusterfs-fuse	It provides support to FUSE based clients. We suggest installing a newer version of the RPM from GlusterFS' site other than the RPM from Linux distributors. See below for more details.
encfs	1.7.4	fuse-encfs	encFS client to encrypt a volume This RPM is readily available from Linux Distributions.

Installing GlusterFS

GlusterFS is a clustered file-system capable of scaling to several petabytes. As noted above, we recommend installing a newer version of glusterfs RPM directly from GlusterFS' website.

Setting up yum repo configuration for GlusterFS

Please execute the following command,

```
shell> wget -P /etc/yum.repos.d
http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo
```

7.3.2 Configuration

7.3.2.1 GlusterFS

At this step, it's expected that you have a gluster that is ready to be mounted. You will need to configure PDF worker servers to automount the gluster.

Automount the Gluster

Modify **/etc/fstab** to automount when the OS boots up.

```
glusterServer.example.net:/gv0   /mnt/gluster    glusterfs    defaults     1 2
```

There is a network timing issue with automount when the OS boots up. To work around this issue, use rc.local to mount GlusterFS. Append the following line to **/etc/rc.d/rc.local**,

```
mount -a
```

7.3.2.2 EncFS

EncFS provides an encrypted Filesystem in Userspace (FUSE). It runs without any special permission and uses FUSE. PDF Worker uses EncFS to protect Personally Identifiable Information (PII) with PDF files.

Initial Setup (Mounting for the first time)

1. Create pdf directory under gluster mount point.

```
shell> sudo su
shell> mkdir -p /mnt/gluster/pdf
shell> chown -R celery.celery /mnt/gluster/pdf
shell> chmod 700 /mnt/gluster/pdf
shell> mkdir -p /opt/edware/pdf
shell> chown -R celery.celery /opt/edware/pdf
shell> chmod 700 /opt/edware/pdf
```

2. Add celery user to the fuse group. See here if you plan to run celeryd with a different user.

```
shell> id celery
uid=500(celery) gid=500(celery) groups=500(celery)

shell> usermod -G fuse celery

shell> id celery
uid=500(celery) gid=500(celery) groups=500(celery),494(fuse)
```

3. Mount EncFS
   As **`celery`** user, when prompt for configuration options, choose "p" for pre-configured paranoia mode. By default, PDF Worker is configured to read/write PDFs to /opt/edware/pdf

```
shell> encfs /mnt/gluster/pdf /opt/edware/pdf
Creating new encrypted volume.
```

```
Please choose from one of the following options:
 enter "x" for expert configuration mode,
 enter "p" for pre-configured paranoia mode,
 anything else, or an empty line will select standard mode.
?> p

Paranoia configuration selected.
Configuration finished.  The filesystem to be created has
the following properties:
Filesystem cipher: "ssl/aes", version 3:0:2
Filename encoding: "nameio/block", version 3:0:1
Key Size: 256 bits
Block Size: 1024 bytes, including 8 byte MAC header
Each file contains 8 byte header with unique IV data.
Filenames encoded using IV chaining mode.
File data IV is chained to filename IV.
File holes passed through to ciphertext.
-------------------------- WARNING --------------------------
The external initialization-vector chaining option has been
enabled.  This option disables the use of hard links on the
filesystem. Without hard links, some programs may not work.
The programs 'mutt' and 'procmail' are known to fail.  For
more information, please see the encfs mailing list.
If you would like to choose another configuration setting,
please press CTRL-C now to abort and start over.
Now you will need to enter a password for your filesystem.
You will need to remember this password, as there is absolutely
no recovery mechanism.  However, the password can be changed
later using encfsctl.
New Encfs Password:
Verify Encfs Password:
```

Important:
Keep the password in safe place. There are no recovery procedures if the password is lost.

To mount EncFS, a password is always pass with the enfs command. For security reasons, a system administrator must manually mount EncFS when the server boots up.

The administrator must be logged in as celery user to execute encfs command to remount.

7.3.2.3 celeryd-services

Similar to Smarter Web Application, PDF Workers, reads the same INI file (located in /opt/edware/conf/smarter.ini). Please refer here to generate INI file.

Changing Celery process' user/group (Optional)

If you need to run celery process with a different user or group other than the default, you will need to modify **/opt/edware/conf/celeryd-services.conf**.

You'll need to modify the values for CELERYD_USER and/or CELERYD_GROUP.

Changing PDF Worker's Celery Configuration (Optional)

In **/opt/edware/conf/smarter.ini**, the following configurations are relevant and most likely needs to be changed in PDF Worker,

Configuration Name	Description	Example Va
services.celery.BROKER_URL	The Broker URL that PDF Worker will bind to	amqp://edwar

Enabling SSL (Optional)

This section is required only if you've enabled SSL on your PDF Messenger.

Copy Client Certificates

Copy the client certificates and testca generated from here and place the CA to /opt/edware/conf/testca and certificates to /opt/edware/conf/client.

Configure Celery to use SSL

1. Update broker URL to RabbitMQ with SSL port number

```
services.celery.BROKER_URL = amqp://user:pwd@broker:5671/services
```

2. Add the following lines into smarter.ini

```
services.celery.BROKER_USE_SSL = {'ca_certs': '/opt/edware/conf/testca/cacert.pem', 'keyfile':
'/opt/edware/conf/client/key.pem', 'certfile': '/opt/edware/conf/client/cert.pem', 'cert_reqs':
True }
```

3. Restart celeryd

```
shell> /etc/init.d/celeryd-services restart
```

7.4 PDF Pre-Generator

PDF Pre-Generator is used to pre-generate PDFs when new data batches are loaded into the system. This mechanism is done via cron job that is scheduled to run and check the database for newly ingested data. We can consider this mechanism as an offline Smarter Web Application. By offline, it implies that the server is running Smarter Web Application via Apache, but is not serving web requests from end user. Its sole responsibility is to trigger PDF pregeneration on a scheduled basis.

PDF Generator needs to write and update the edware_stats database, therefore the database connection must be directly to the database master. We recommend this database to be on Loader Database server.

7.4.1 Installation

Please refer to here for installing PDF Pre-Generator server. The install is almost identical to installing and configuring web servers, except that wsgi needs to be configured to be single process. This is described in configuration section.

7.4.2 Configuration

7.4.2.1 smarter

Generating Smarter.ini

Similarly, PDF Generator needs to read from INI file. Please refer to here for generating this file.

Note: In settings.yaml, you can check if there exists a section dedicated to this server type. To generate INI for PDF Generator, you can/should set the environment value as 'uat.pdf_generator'.
ex. python generate_ini.py -e uat.pdf_generator

Configurations specific to PDF Generator in Smarter.ini

Configuration	Description	Example Value
cache.session.expire	The special pdf generation batch user's session expiration in seconds. This should be set to the value that you expect the duration of the PDFs to be generated for the current batch of data.	300000
batch.user.session.timeout	The special pdf generation batch user's cookie expiration in seconds. This should align with the session expiration.	300000
trigger.pdf.enable	Enable pdf pre-generation	True
trigger.pdf.schedule.cron.hour	The hour in which to schedule pdf-generation	20
trigger.pdf.schedule.cron.minute	The minute in which to schedule pdf-generation	15

7.4.2.2 WSGI

The configuration for wsgi is almost identical to the configuration of wsgi in web servers. Please refer to that section for the configurations needed. The only configuration difference is described below.

Configuration for Smarter application specific WSGI configuration

Modify **/etc/httpd/conf.d/wsgi_edware.conf** to add WSGI configuration for PDF Pre-Generator. The notable difference compared to web server is that we're running on a single process.

```
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
WSGIDaemonProcess pyramid user=apache group=apache processes=1 threads=30 python-
path=/opt/virtualenv/lib/python3.3/site-packages
WSGIScriptAlias / /opt/edware/smarter/smarter.wsgi
WSGIImportScript /opt/edware/smarter/smarter.wsgi process-group=pyramid application-
group=%{GLOBAL}
WSGISocketPrefix run/wsgi
<Directory /opt/virtualenv>
   WSGIProcessGroup pyramid
   Order allow,deny
   Allow from all
</Directory>
WSGIPythonPath /opt/virtualenv/lib/python3.3/site-packages
```

7.5 Extract Messenger

Extract Messenger hosts a message system between Smarter Web Application and Extract Worker. It uses RabbitMQ as its message broker and extract tasks are sent from Web Servers and are received by Extract Workers. The installation and configuration is very similar to PDF Messenger.

7.5.1 Installation

Please refer to the installation of RabbitMQ from here.

Note: The only configuration difference is that the name of the virtual host for extracts should be different than the one defined for PDF Messenger. By default, we expect and recommend the virtual host for Extract Messenger to be named, **edextract**.

7.5.2 Configuration

Please refer to the configuration of RabbitMQ from here.

7.6 Extract Worker

Extract worker is responsible for receiving extract tasks from the queue and generating bulk raw extracts in CSV format. The installation and configuration is very similar to PDF Worker.

There are a few key differences for Extract Worker:

- Celery service name is **celeryd-edextract**
- Celery configuration file is **celeryd-edextract.conf**
- Mount EncFS to **/opt/edware/extraction** (Be sure to mount it as celery user)
- **extract.celery.BROKER_URL** is the broker URL used by Extract Worker

7.6.1 Installation

Please refer to the installation of Extract Worker from here.

Note: Please remember the notable installation differences between PDF and Extract Worker, namely, the celery service.

7.6.2 Configuration

Please refer to the configuration of Extract Worker from here.

Note: Please remember the notable configuration differences between PDF and Extract Worker, namely, the celeryd configuration file, the EncFS mount point, and the broker URL configuration in smarter.ini.

7.7 Cache

Smarter Web Application uses cache servers to persist data for reports and user sessions for some configurable amount of duration. It uses memcached, an in-memory key-value store for small chunks of arbitrary data (strings, objects).

7.7.1 Installation

The following table lists the RPMs that must be installed on cache servers.

Package	Tested Version	RPM Name	Description
memcached	1.4.4	memcached	Available on standard Linux Distributions

7.7.2 Configuration

memcached

The standard install of memcached should be sufficient.

Main configuration for memcached

Modify /etc/sysconfig/memcached for main configuration for memcached

```
PORT="11211"
USER="memcached"
MAXCONN="1024"
```

```
CACHESIZE="64"
OPTIONS=""
```

7.8 Cache Warmer

The Cache Warmer re-populates memcached used by the Application Server for results of popular data requests. Cache Warmer is triggered on a schedule. The data requests are configurable for the types of demographics filters to apply to the data requests.

Similar to PDF Pre-Generator, Cache Warmer is considered as an offline Smarter Web Application. It's a dedicated server used to handle the flushing and re-caching of reports data when new data has been ingested into the system.

7.8.1 Installation

Please refer to the installation guide from here. The install is almost identical to installing and configuring web servers, except that the wsgi needs to be configured to be a single process. This is described in configuration section.

7.8.2 Configuration

WSGI

The configuration for wsgi is almost identical to the configuration of wsgi in web servers. Please refer to that section for the configurations needed. The only configuration difference is described below.

Configuration for Smarter application specific WSGI configuration

Modify **/etc/httpd/conf.d/wsgi_edware.conf** to add WSGI configuration for Cache Warmer. The notable difference compared to web server is that we're running on a single process.

```
WSGIApplicationGroup %{GLOBAL}
WSGIPassAuthorization On
WSGIDaemonProcess pyramid user=apache group=apache processes=1 threads=30 python-
path=/opt/virtualenv/lib/python3.3/site-packages
WSGIScriptAlias / /opt/edware/smarter/smarter.wsgi
WSGIImportScript /opt/edware/smarter/smarter.wsgi process-group=pyramid application-
group=%{GLOBAL}
WSGISocketPrefix run/wsgi
<Directory /opt/virtualenv>
  WSGIProcessGroup pyramid
  Order allow,deny
  Allow from all
</Directory>
WSGIPythonPath /opt/virtualenv/lib/python3.3/site-packages
```

smarter

Generating Smarter.ini

Similarly, Cache Warmer needs to read from INI file. Please refer to here for generating this file.

Note: In settings.yaml, you should check if there exists a section dedicated to this server type. To generate INI for Cache Warmer, you can/should set the environment value as 'uat.cache.
ex. python generate_ini.py -e uat.cache

Configurations specific to Cache Warmer in Smarter.ini

Configuration	Description	Example Value
trigger.recache.enable	Enable cache warmer	True
trigger.recache.schedule.cron.hour	The hour in which to schedule cache warmer	20
trigger.recache.schedule.cron.minute	The minute in which to schedule cache warmer	15

Configuration for Demographics Filters

Cache Warmer re-caches comparing populations reports, and the demographic filters for the report configuration. The state and district comparing populations reports are both candidates for caching, the configuration file contains state and district specific sections.

Modify **/opt/edware/conf/comparing_populations_precache_filters.json** to configure filters specific to your tenant/state. You can override a particular tenant's configuration by adding a section in the JSON file by prepending the tenant name in front of state or district.

Ex. The following configuration has 2 filters for State View, 1 filter for District View, and overwrites ES tenant State View with a different filter.

```
{"state":
      [{"grade: ["3"], "dmgPrgIep": ["Y"]},
      {"grade": ["4"]}],
 "district": [{"grade": ["5"]}],
 "ES.state": [{"grade": ["6"]}]}
```

7.9 Database Master

Database Master server is used for storing all reporting data. Database Master allows insert/update/delete/select queries. Master sends replication data to one or more standby servers.

In order to protect PII, database data should be encrypted. Please see the Encrypting section to prepare to encrypt your data in PostgreSQL.

7.9.1 Installation

The following lists the RPM required by Database Master

Package	Tested Version	RPM Name	Description
Database	9.2.4	postgresql postgresql92-server postgresql92-libs postgresql92-contrib postgresql92-devel	These RPMs are available from most Linux distributions
Replication Manager	1.2.0	repmgr	RPM is available from most Linux distributions.

Encrypting PostgreSQL data and WAL logs

EncFS should be used to encrypt Postgres data.

1. Install encfs
2. Plan ahead about directory for raw data and encrypted data.
3. Configure encfs partition to use expert mode. Postgresql uses hard link when writing WAL log to disk, so operator needs to enable Encfs to support hard links. During configure encfs, you need to disable "`filename initialization vector chaining`". See the following transcript of configuration.

```
shell> su - postgres
shell> encfs /opt/wgen/postgres/pg_raw/  /opt/wgen/postgres/pg_encfs/
Creating new encrypted volume.
Please choose from one of the following options:
enter "x" for expert configuration mode,
enter "p" for pre-configured paranoia mode,
anything else, or an empty line will select standard mode.
?> x
Manual configuration mode selected.
The following cipher algorithms are available:
1. AES : 16 byte block cipher
-- Supports key lengths of 128 to 256 bits
-- Supports block sizes of 64 to 4096 bytes
2. Blowfish : 8 byte block cipher
-- Supports key lengths of 128 to 256 bits
-- Supports block sizes of 64 to 4096 bytes
```

```
Enter the number corresponding to your choice: 1
Selected algorithm "AES"
Please select a key size in bits.  The cipher you have chosen
supports sizes from 128 to 256 bits in increments of 64 bits.
For example:
128, 192, 256
Selected key size: 256
Using key size of 256 bits
Select a block size in bytes.  The cipher you have chosen
supports sizes from 64 to 4096 bytes in increments of 16.
Or just hit enter for the default (1024 bytes)
filesystem block size:4096
Using filesystem block size of 4096 bytes
The following filename encoding algorithms are available:
1. Block : Block encoding, hides file name size somewhat
2. Null : No encryption of filenames
3. Stream : Stream encoding, keeps filenames as short as possible
Enter the number corresponding to your choice: 1
Enable filename initialization vector chaining?
This makes filename encoding dependent on the complete path,
rather then encoding each path element individually.
The default here is Yes.
Any response that does not begin with 'n' will mean Yes:no
Enable per-file initialization vectors?
This adds about 8 bytes per file to the storage requirements.
It should not affect performance except possibly with applications
which rely on block-aligned file io for performance.
The default here is Yes.
Any response that does not begin with 'n' will mean Yes: Yes
External chained IV disabled, as both 'IV chaining'
and 'unique IV' features are required for this option.
Enable block authentication code headers
on every block in a file?  This adds about 12 bytes per block
to the storage requirements for a file, and significantly affects
performance but it also means [almost] any modifications or errors
within a block will be caught and will cause a read error.
The default here is No.
Any response that does not begin with 'y' will mean No:Yes
Add random bytes to each block header?
This adds a performance penalty, but ensures that blocks
have different authentication codes.  Note that you can
have the same benefits by enabling per-file initialization
vectors, which does not come with as great of performance
```

```
penalty.
Select a number of bytes, from 0 (no random bytes) to 8: 8
Enable file-hole pass-through?
This avoids writing encrypted blocks when file holes are created.
The default here is Yes.
Any response that does not begin with 'n' will mean Yes: Yes
Configuration finished.  The filesystem to be created has
the following properties:
Filesystem cipher: "ssl/aes", version 3:0:2
Filename encoding: "nameio/block", version 3:0:1
Key Size: 256 bits
Block Size: 4096 bytes, including 8 byte MAC header
Each file contains 8 byte header with unique IV data.
File holes passed through to ciphertext.
Now you will need to enter a password for your filesystem.
You will need to remember this password, as there is absolutely
no recovery mechanism.  However, the password can be changed
later using encfsctl.
```

7.9.2 Configuration

postgres

Configuring postgres master

1. If this is the first time you install postgres. Run
   `shell> service postgresql-92 initdb`
2. We need to set up archive directory for WAL log for replication. And put the right archive directory into archive_command options. For example in the runbook we have /var/lib/pgsql/9.2/archive as archive directory.
3. Generate ssh key pair. Executing the following commands with **`postgres`** user.

```
shell> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/pgsql/.ssh/id_rsa):
Created directory '/var/lib/pgsql/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/pgsql/.ssh/id_rsa.
Your public key has been saved in /var/lib/pgsql/.ssh/id_rsa.pub.
The key fingerprint is:
58:66:3b:d1:97:f4:b9:fc:1f:66:dc:28:3f:8f:bf:8f postgres@dbpgdw0.qa.dum.edwdc.net
The key's randomart image is:
+--[ RSA 2048]----+
|            .    |
|         . . o . |
|        = . o o  |
|       = o . . . |
|      . S     o  |
|         .    .o.|
|           . .=o|
```

```
|             o+oo|
|              E=B|
+-----------------+
```

4. Distribute public key to postgresql database replica servers
   Append generated public key(**/var/lib/pgsql/.ssh/id_rsa.pub**) to replica servers(**/var/lib/pgsql/.ssh/authorized_keys**)
5. Modify **/var/lib/pgsql/9.2/data/postgresql.conf**
   Update following options. Keep other options untouched.

Configuration	Description	Example Value
listen_addresses	address for postgres server, we use all IP on the server. so '*'	'*'
shared_buffers	This depends on your server memory. See postgres manual for details.	8192MB
work_mem	work_mem for each connection,	100MB
log_filename	file name for log file	'postgresql-%a.log'
log_truncate_on_rotation		on
log_rotation_size		0
log_timezone	Time Zone for log file. We use 'UTC'	'UTC'
timezone	Time Zone	'UTC'
wal_level	We are doing replication. So host_standbay	host_standby
checkpoint_segments		30
archive_mode	We need to run replication. so we set it on	on
max_wal_senders	number of processes to send WAL log to remote	10
wal_keep_segments	WAL files that are kept under pg_xlog before it is archived.	5000
hot_standby		on
archive_command	command to be executed by celery to move WAL log to archive directory. We prefer keep pg_xlog small and move compress WAL log to archive and delete the original WAL log under pg_xlog	'gzip -9 < %p > /var/lib/pgsql/9.2/archiv && rm %p'

6. Update **/var/lib/pgsql/9.2/data/pg_hba.conf,**
   add following line into pg_hba.conf, keep the original pg_hba.conf content

```
host all all ###.###.###.###/## trust
host replication all ###.###.###.###/## trust
host all all 127.0.0.1/32 trust
```

###.###.###.### is your postgresql slave server IP/network.

7. Create edware user and database psql prompt

```
CREATE DATABASE edware;
CREATE DATABASE edware_stats;
CREATE USER edware WITH PASSWORD 'edware2013';
GRANT ALL PRIVILEGES ON DATABASE edware to edware;
GRANT ALL PRIVILEGES ON DATABASE edware_stats to edware;
```

8. Create /var/lib/pgsql/repmgr.conf

Configuration	Description	Example Value
cluster	group of cluster name. Use the same name for all servers.	edware_pg_cluster
node	node number must be unique for each servers.	1
node_name	value must be its own server hostname	fully qualified domain name of t machine
conninfo	repmgr to connect to PostgreSQL server, host value must be its server host name, and the proper user name and database name for replicated database	'host=<my_FQDN_.com> user= dbname=edware'
pg_bindir	path to pgsql installation	/usr/pgsql-9.2/bin

9. Create repmgr user for PostgreSQL psql prompt

```
CREATE ROLE repmgr SUPERUSER LOGIN;
```

10. Grant privilege to repmgr_edware_pg_cluster to edware user

```
shell> su - postgres
shell> psql -d edware
edware=#  grant usage on schema repmgr_edware_pg_cluster to edware
```

11. Restart PostgreSQL
12. Register repmgr

```
PATH=$PATH:/usr/pgsql-9.2/bin repmgr -f /var/lib/pgsql/repmgr.conf master register
```

13. To Be added: Create schema for smarter

7.10 Database Replica

Database Replica servers are responsible for replicating data from master database. The server is a read-only server, therefore, only SELECT queries can be executed.

In order to protect PII, database data should be encrypted. Please see the Encrypting section to prepare to encrypt your data in PostgreSQL.

In Smarter Balanced Reporting, we allocated ⅓ of replica servers for data extract use and ⅔ replica servers for web frontend reporting. The data extraction replica server doesn't require smarter to be installed.

7.10.1 Installation

The following lists the RPM required by Database Replica for web frontend reporting

Package	Tested Version	RPM Name	Description
Database	9.2.4	postgresql postgresql92-server postgresql92-libs postgresql92-contrib postgresql92-devel	These RPMs are available from most Linux distributions.
Replication Manager	1.2.0	repmgr	RPM is available from most Linux distributions.
smarter	<TBD>	smarter	An Amplify supplied RPM. edmigrate-celerdy is package within smarter RPM.

The following lists the RPM required by Database Replica for extract

Package	Tested Version	RPM Name	Description
Database	9.2.4	postgresql postgresql92-server postgresql92-libs postgresql92-contrib postgresql92-devel	These RPMs are available from most Linux distributions.
Replication Manager	1.2.0	repmgr	RPM is available from most Linux distributions.

7.10.2 Configuration

postrgres

1. Generate ssh key pair. Executing commands must be **`postgres`** user.

```
shell> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/pgsql/.ssh/id_rsa):
Created directory '/var/lib/pgsql/.ssh'.
Enter passphrase (empty for no passphrase):
```

```
Enter same passphrase again:
Your identification has been saved in /var/lib/pgsql/.ssh/id_rsa.
Your public key has been saved in /var/lib/pgsql/.ssh/id_rsa.pub.
The key fingerprint is:
58:66:3b:d1:97:f4:b9:fc:1f:66:dc:28:3f:8f:bf:8f postgres@dbpgdw0.qa.dum.edwdc.net
The key's randomart image is:
+--[ RSA 2048]----+
|            .    |
|         . . o . |
|        = . o o  |
|       = o . . . |
|      . S     o  |
|         .    .o.|
|            . .=o|
|             o+oo|
|              E=B|
+-----------------+
```

2. Distribute public key to postgresql slave servers
   append generated public key(**/var/lib/pgsql/.ssh/id_rsa.pub**) to slave servers(**/var/lib/pgsql/.ssh/authorized_keys**)
3. Stop PostgreSQL
4. Delete all files and directories
   ```
   cd /var/lib/pgsql/9.2/data
   rm -rf *
   ```
5. Start PostgreSQL
6. Clone the master database
   a. This must be **postgres** user to execute, replace master.server.com to your master databse fully qualified domain name.
      ```
      shell> su - postgres
      shell> PATH=$PATH:/usr/pgsql-9.2/bin repmgr -D
      /var/lib/pgsql/9.2/data -d edware -p 5432 -R postgres --verbose
      standby clone master.server.com
      ```
7. After clone. Modify recovery.conf to add following two lines, you need to change the archive path to what you have on your replica server.
   ```
   restore_command = 'gunzip < /var/lib/pgsql/9.2/archive/%f > %p'
   archive_cleanup_command = 'pg_archivecleanup /var/lib/pgsql/9.2/archive %r'
   ```
8. After clone. if there are files under archive directory in master database. scp them under postgres user to replicated servers. for example. this copy from master to slave replica's archive directory
   ```
   shell> scp * postgres@[fqdn of replica]:/var/lib/pgsql/9.2/archive/
   ```
9. Update pg_hba.conf, make sure this line is in pg_hba.conf
   ```
   host all all 127.0.0.1/32 trust
   ```
10. After clone. Modify postgresql.conf for replication server

Configuration	Description	Example Value
listen_addresses	address for postgres server, we use all IP on the server. so '*'	'*'

shared_buffers	This depends on your server memory. See postgres manual for details.	8192MB
work_mem	work_mem for each connection,	100MB
log_filename	file name for log file	'postgresql-%a.log'
log_truncate_on_rotation		on
log_rotation_size		0
log_timezone	Time Zone for log file. We use 'UTC'	'UTC'
timezone	Time Zone	'UTC'
wal_level	We are doing replication. So host_standbay	host_standby
checkpoint_segments		30
archive_mode	We are doing replication. So turn it on	on
max_wal_senders	Number of processes that sends WAL logs.	10
wal_keep_segments	WAL files that are kept under pg_xlog before it is archived.	5000
hot_standby		on
archive_command	command to be executed by celery to move WAL log to archive directory. We prefer keep pg_xlog small and move compress WAL log to archive and delete the original WAL log under pg_xlog	'gzip -9 < %p > /var/lib/pgsql/9.2/archive/%f && r %p'

11. Restart database after modified postgresql.conf
12. Create /var/lib/pgsql/repmgr.conf

Configuration	Description	Example Value
cluster	group of cluster name. Use the same name for all servers.	edware_pg_cluster
node	node number must be unique for each servers.	1
node_name	value must be its own server hostname	fully qualified domain name of the machine
conninfo	repmgr to connect to PostgreSQL server, host value must be its server host name, and the proper user name and database name for replicated database	'host=<my_FQDN_.com> user=repm dbname=edware'
pg_bindir	path to pgsql installation	/usr/pgsql-9.2/bin

13. make sure /var/log/repmgrd and /var/run/repmgrd exists and are owned by postgres user
14. Register repmgr as postgres use
    ```
    PATH=$PATH:/usr/pgsql-9.2/bin repmgr -f /var/lib/pgsql/repmgr.conf standby register
    ```
15. Start replication
    a. If used repmgr 1.2 then sudo as postgres
       ```
       PATH=$PATH:/usr/pgsql-9.2/bin repmgrd -f /var/lib/pgsql/repmgrd.conf
       ```
    b. if used repmgr, then run as root
       ```
       service repmgrd start
       ```

celeryd-edmigrate

celeryd-edmigrate is a service used for our migration process. The role of replicas in migration process is known as a player. We only need to enable celeryd-edmigrate for Database replicas that are used for Smarter Balanced Reporting Web Front End.

Configure iptables for celeryd-edmigrated

1. Reset iptables rules.
   shell> `service iptables stop`
2. Create minimum iptables rule
   shell> `/sbin/iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT`
   shell> `/sbin/iptables -A INPUT -p tcp --dport 22 -j ACCEPT`
   shell> `/sbin/iptables -A INPUT -p tcp --dport 5432 -j ACCEPT`
   shell> `/sbin/iptables -P INPUT DROP`
3. Create new user-defined custom chain “EDMIGRATE_PGSQL”
   shell> `/sbin/iptables -N EDMIGRATE_PGSQL`
4. Create new rule for the chain
   shell> `/sbin/iptables -A EDMIGRATE_PGSQL -p tcp -d 127.0.0.1 --dport 5432 -j ACCEPT`
   shell> `/sbin/iptables -A EDMIGRATE_PGSQL -p tcp --dport 5432 -j REJECT`
5. Save iptables rules. Caution: when you execute a following command, it will overwrite existing iptables saved settings.
   shell> `service iptables save`
   or
   shell> `iptables-save > /etc/sysconfig/iptables`

Configure sudoers for celeryd-edmigrate

1. Run visudo
   shell> `visudo`
2. Add the following lines. Caution: the second line is a single line.
   ```
   Defaults:celery !requiretty
   ```

```
celery ALL=NOPASSWD: /sbin/iptables -t filter -I INPUT -j EDMIGRATE_PGSQL, /sbin/iptables
-t filter -D INPUT -j EDMIGRATE_PGSQL, /sbin/iptables -t filter -I OUTPUT -j
EDMIGRATE_PGSQL, /sbin/iptables -t filter -D OUTPUT -j EDMIGRATE_PGSQL, /sbin/iptables-
save
```

Configure Player

1. Register celery task
   shell> `chkconfig --add celeryd-edmigrate`
2. Start service
   shell> `service celeryd-edmigrate start`
3. Stop service
   shell> `service celeryd-edmigrate stop`

7.11 Database Load Balancer

PgBouncer is a lightweight connection pooler for PostgreSQL. Smarter web application uses PgBouncer in order to handle massive database connections by web service requests efficiently.

7.11.1 Installation

The following lists the RPM required by Database Load Balancer

Package	Tested Version	RPM Name	Description
Database Middleware	1.5.4	pgbouncer	RPM is available from most Linux distributions.

7.11.2 Configuration

1. Edit **`/etc/pgbouncer/pgbouncer.ini.`** Modify following options and keep others unchanged.

```
[databases]
* = host=dwrouter1.qa.dum.edwdc.net port=9999 user=edware password=edware2013 pool_size=450
connect_query='select 1'

[pgbouncer]
logfile = /var/log/pgbouncer/pgbouncer.log
pidfile = /var/run/pgbouncer/pgbouncer.pid
listen_addr = *
listen_port = 6432
auth_type = md5
auth_file = /etc/pgbouncer/userlist.txt
admin_users = user2, postgres
stats_users = stats, postgres
pool_mode = session
server_reset_query = DISCARD ALL
ignore_startup_parameters = client_min_messages
max_client_conn = 500
default_pool_size = 100
reserve_pool_size = 10
```

```
log_connections = 0
log_disconnections = 0
log_pooler_errors = 1
query_timeout = 180
```

2. Create **/etc/pgbouncer/userlist.txt**
   PgBouncer requires its own userlist file. The client authenticates with PgBouncer service first, this is independent of the PostgreSQL Database Authentication. The authentication file contains pairs of double-quote enclosed username and passwords that a client application uses to access PgBouncer. The location of **userlist.txt** is specified at **auth_file** in **/etc/pgbouncer/pgbouncer.ini**.
   a.
   ```
   "postgres" "md5cf8e80c6852c634a6e00613455d34189"
   "edware" "md50927d04170fc5ebc2a14e662d5425c9c"
   ```
   b. Encrypting the password
   The authentication file takes both clear text passwords and MD5-encrypted passwords by settings in pgbouncer.ini (**auth_type**). For security reasons, we strongly encourage to use of MD5-encrypted passwords.
      i. How to generate MD5-encrypted password
      From PostgreSQL Query Prompt
      ```
      select 'md5' || md5 ('edware2013' || 'edware');
      ```
      Copy the resulting string into the userlist.txt file:
      ```
      “edware" "md5d305b538896b9a9ea5086cc126bcc09f"
      ```
3. Log file
   The location of log file can be specified in pgbouncer.ini (**logfile**).

7.12 Database Pool

Pgpool-II is a middleware that works between PostgreSQL servers and PostgreSQL database client. Pgpool-II is mainly used by Smarter to load balance the distribution of SELECT queries among multiple servers, thus improving the system’s overall throughput.

In Smarter Balanced Reporting. We have two sets of Database Pool. One set of Database Pool is for Smarter Balanced Reporting Web Front End. Another set of Database Pool is for Smarter Balanced Bulk Extract and Bulk Printing.

7.12.1 Installation

The following lists the RPM required by Database Pool

Package	Tested Version	RPM Name	Description
Database	9.2.8	postgresql92-server postgresql92-libs postgresql92	RPM for postgres user for pgpool
Database	3.3.1	pgpool-II	RPM is available from most Linux

Middleware			distributions.

7.12.2 Configuration

pgpool for Smarter Balanced Reporting web frontend

1. Make sure postgres user exists
2. Make sure firewall is opened for port 9999 for pgpool.
3. Disable postgresql server running
   ```
   shell> service postgresql-9.2 stop
   ```
4. Make sure /var/run/pgpool is owned by postgres user
5. Edit **/etc/pgpool-II/pool_hba.conf,** Add following to allow access to pgpool.
   ```
   host all all ###.###.###.###/## trust
   host replication all ###.###.###.###/## trust
   host all all 127.0.0.1/32 trust
   ```
   ###.###.###.### is your pgbouncer server IP/network.
6. Edit **/etc/pgpool-II/pcp.conf**
   Configuration for pcp.conf user. We need to use pcp_attach_node command to recover pg_pool connection during edmigrate.
   a. `generate md5 password`
      ```
      # USERID:MD5PASSWD
      postgres:{the md5 passphrase from postgres}
      ```
7. Edit **/etc/pgpool-II/pgpool.conf**
   Main configuration file for Pgpool-II. Default configurations that comes with RPM is almost sufficient for Smarter. There are few changes need in the file.
   a. Updating pgpool.conf, keep other options unchanged.
      Assuming load balancing 2 servers
      ```
      port = 9999
      listen_addresses = '*'
      backend_hostname0 = 'dbpgdwr0s0.qa.dum.edwdc.net'
      backend_port0 = 5432
      backend_weight0 = 1
      backend_data_directory0 = '/mnt/san-postgres/9.2/data'
      backend_flag0 = 'ALLOW_TO_FAILOVER'
      backend_hostname1 = 'dbpgdwr0s1.qa.dum.edwdc.net'
      backend_port1 = 5432
      backend_weight1 = 1
      backend_data_directory1 = '/mnt/san-postgres/9.2/data'
      backend_flag1 = 'ALLOW_TO_FAILOVER'
      load_balance_mode = on
      replication_mode = on
      ```
   b. Update pgpool.conf to add failover_command for edmigrate, if the slaves that pgpool points doesn't need edmigrate. then no need for this step and step 8.
      ```
      failover_command = '/usr/local/bin/pgpool_failover.sh %d %h %p'
      ```

3. Create **/usr/local/bin/pgpool_failover.sh** script by copying following text

   ```
   #!/bin/bash
   node_id=$1
   host_name=$2
   port=$3
   pcp_user=postgres  # replace this with your pcp.conf user name if
   ```

```
necessary
pcp_pass=postgres  # replace this with your pcp.conf password if
necessary
pcp_host=localhost
pcp_port=9898     # replace this with your pcp.conf pgpool.conf pcp port
if necessary
attach_timeout=100
pgpool_status_file=/var/log/pgpool-II/pgpool_status # replace with your
pgpool_status file location

# command to check pgpool parent is running or not
COMMAND_PS="ps -C pgpool|wc -l"
# command to check failover host name
COMMAND_TEST="nc -w 1 $host_name $port"
# command to reattach node to pgpool. change port number and user name,
password
COMMAND_ATTACH="/usr/bin/pcp_attach_node $attach_timeout localhost
$pcp_port $pcp_user $pcp_pass $node_id"

wait_for_host_to_recover() {
    eval $COMMAND_TEST
    while [ $? -ne 0 ]; do
       PARENT=`eval $COMMAND_PS`
       if [ $PARENT -eq "1" ]; then

          rm $pgpool_status_file
          exit
       fi
       eval $COMMAND_TEST
    done
    eval $COMMAND_ATTACH
}

wait_for_host_to_recover &
```

4. Logfile
   create /var/log/pgpool if it doesn't exist, and change ownership to postgres.postgres
   a. /var/log/pgpool.log
   b. /var/log/pgpool/pgpool_status.

pgpool for Extracts

1. Make sure postgres user exists
2. Make sure firewall is opened for port 9999
3. Disable postgresql server running
   `shell> service postgresql-9.2 stop`
4. Make sure /var/run/pgpool is owned by postgres user
5. Edit **`/etc/pgpool-II/pool_hba.conf,`** Add following to allow access to pgpool.
   ```
   host all all ###.###.###.###/## trust
   host replication all ###.###.###.###/## trust
   host all all 127.0.0.1/32 trust
   ```
   ###.###.###.### is your pgbouncer server IP/network.
6. Edit **`/etc/pgpool-II/pgpool.conf`**

Main configuration file for Pgpool-II. Default configurations that comes with RPM is almost sufficient for Smarter. There are few changes need in the file.

a. Updating pgpool.conf
   Assuming load balancing 2 servers

```
port = 9999
listen_addresses = '*'
backend_hostname0 = 'dbpgdwr0s2.qa.dum.edwdc.net'
backend_port0 = 5432
backend_weight0 = 1
backend_data_directory0 = '/mnt/san-postgres/9.2/data'
backend_flag0 = 'ALLOW_TO_FAILOVER'
backend_hostname1 = 'dbpgdwr0s3.qa.dum.edwdc.net'
backend_port1 = 5432
backend_weight1 = 1
backend_data_directory1 = '/mnt/san-postgres/9.2/data'
backend_flag1 = 'ALLOW_TO_FAILOVER'
load_balance_mode = on
replication_mode = on
```

5. Logfile
   create /var/log/pgpool if it doesn't exist, and change ownership to postgres.postgres
   a. /var/log/pgpool.log
   b. /var/log/pgpool/pgpool_status.

7.13 Smoke test for smarter functioning

After setting up above software. There is a smoke test to test the whole system is functioning correctly.

1. open a browser
2. pick a web server for smarter. for example: web1.example.com
3. type following url into browser tab: http://web1.example.com/services/heartbeat
4. if it shows 200 Ok. Then the whole Smarter Balanced Reporting is running correctly now. If not. you need to troubleshooting the whole chained of reporting.

7.14 Landing Zone

Landing Zone is a tenant-based drop-off zone for incoming batched data files. Files are dropped into its dedicated tenant space, and are transferred to the Loader server for processing. A File Watcher sits on this machine and watches for newly arrived files

7.14.1 Installation

The following lists the RPM required by Landing Zone

Package	Tested Version	RPM Name	Description
Python	3.3	python3-3.3.0	Python 3 RPM is generally not available from Linux distributions,

			hence Amplify has supplied a customized Python 3 RPM
EdSFTP	TBD	edsftp	Amplify's EdSFTP RPM for Landing Zone server.

7.14.2 Configuration

edsftp

EdSFTP is a RPM supplied by Amplify that contains scripts for setting up Landing Zone server for tenants.

Configuring chroot

1. Add the following section in **/etc/ssh/sshd_config**. Make sure the content is placed at the very bottom of the file.

```
Match Group edwaredataadmin
  X11Forwarding no
  AllowTcpForwarding no
  ForceCommand internal-sftp
  ChrootDirectory /sftp/%h
```

2. Replace existing system override in **/etc/ssh/sshd_config**

```
override default of no subsystems
#Subsystem      sftp    /usr/libexec/openssh/sftp-server
Subsystem       sftp    internal-sftp
```

3. Restart sshd service

```
shell> service sshd restart
```

Generating Smarter INI file

EdSFTP also reads configuration from INI file. You can find the instructions of how to generate the INI file from here.

Creating Groups for SFTP users

By default, all the users will belong to the group, edwaredataadmin. If you want to change the group that the SFTP users belong to, you can modify **/opt/edware/conf/smarter.ini** and change the value for sftp.group

To create the group, you will need to execute the following commands,

```
shell> source /opt/virtualenv/edsftp/bin/activate
(virtualenv) cd /opt/virtualenv/edsftp
(virtualenv) sftp_driver.py --init
```

Creating Tenant Accounts

For each tenant, you will need to set up an account. Each tenant may have more than one SFTP user if required.

```
# adding tenant "ca"
(virtualenv) sftp_driver.py -s -t ca
# adding user "ca_user1"
(virtualenv) sftp_driver.py -a -u ca_user1 -t ca -r sftparrivals
# set password for the user
(virtualenv) passwd ca_user1
```

You can test whether SFTP works with jailroot for the above user

```
Shell> sftp ca_user1@landingZoneServer
ftp> cd /
ftp> cd file_drop
ftp> put </path/to/test/file/to/be/sftped>
ftp> cd /etc   # access will be restricted
```

Starting EdSFTP watcher service

On Landing Zone machine, you will need to run the SFTP watcher service to monitor incoming files that are being dropped off.

The following instructions requires the Loader server to be installed and configured (udl2 user must be created in Loader machine). Please proceed if that is completed.

As root user,

1. start service
   `service edsftp-watcher start`
2. Verify the service is running

```
ps -ef | grep sftp_driver
```

7.15 Loader

The Data Loader is responsible for processing newly arrived data and loading it into a Staging database.

The Loader has the ability to call a callback URL for notification purposes, please make sure that the server is able to make outgoing HTTP and/or HTTPS to such URLs (port 80 and 443).

7.15.1 Installation

The following lists the RPM required by Loader

Package	Tested Version	RPM Name	Description
Python	3.3	python3-3.3.0	Python 3 RPM is generally not available from Linux distributions, hence Amplify has supplied a customized Python 3 RPM
EdUDL	TBD	edudl	Amplify's RPM for Data Loader
smarter	TBD	smarter	Amplify's RPM for Smarter. We currently need this installed in the Loader for logging purposes.

7.15.2 Configuration

celeryd-udl2

Generate udl2_conf.ini

This configuration file is needed and read by celeryd-udl2

```
shell>  . /opt/virtualenv/udl2/bin/activate
(virtualenv) /opt/edware/conf
(virtualenv) python generate_ini.py -i udl2_conf.yaml -o
/opt/edware/conf/ud2_confi.ini
```

This file contains configurations for tenants that are supported, the UDL database, the Staging and Production Database servers.

Generate smarter.ini

Currently, we require smarter.ini for configuring logging in the Loader machine.

Please follow the instructions from here.

Initialize UDL2 database

You will need to manually initialize the Loader Database schema

```
shell>  . /opt/virtualenv/udl2/bin/activate
(virtualenv) cd /opt/virtualenv/udl2
(virtualenv) python -m edudl2.database.database --action setup
```

Ensure GPG keys are copied to /opt/edware/keys

Please make sure the encryption keys are copied to /opt/edware/keys

Mount Work Zones directories from Gluster

On every Loader servers, please mount the gluster as root user. Make sure id of udl2 user and group are same across central UDL DB and pipeline machines (use group id 501 and user id of 502)

```
usermod -G fuse udlchown -R udl2.udl2 /opt/edware
2
```

Mount encrypted zones from gluster as udl2 user, execute the following,

```
encfs /mnt/gluster/udl/zones /opt/edware/zones -o umask='007'
```

Ensure Outgoing HTTP and HTTPs ports are opened

Loader makes GET requests to a configurable URL specified in data files for notifications. Please make sure ahead of time that the URLs are reachable.

Start edudl2-file-grabber and edudl2-trigger service to watch for incoming files being copied to work zone

Note: If multiple Loader servers exist, only one instance needs to be running edudl2-file-grabber and edudl2-trigger service.

edudl2-file-grabber is to move file (except .partial file extension) from Landing Zone to Loader by rsync for edudl2-trigger service.

To start edudl2-file-grabber,

```
service edudl2-file-grabber start
```

edudl2-trigger is used to monitor and watch for files that have arrived in the Loader machine and triggering the Loader to process the new data file.

To start edudl2-trigger,

```
service edudl2-trigger start
```

The logs can be found based on the logging configs defined in /opt/edware/conf/smarter.ini

7.16 Loader Messenger

Loader Messenger hosts the broker for Loader tasks that are requested by Data Loader. We have chosen to use RabbitMQ as the message broker. RabbitMQ, written in Erlang, implements the Advanced Message Queuing Protocol (AMQP) standard.

7.16.1 Installation

Please refer to the installation of RabbitMQ from here.

Note: By default, we expect and recommend the virtual host for Loader Messenger to be named, **edudl**. Please make the appropriate configuration changes.

7.16.2 Configuration

Please refer to the configuration of RabbitMQ from here.

7.17 Loader Database

Loader Data hosts an internal centralized database used for temporary storage used by the Loader. When data files get dropped off in the Landing Zone, the data gets transformed and are temporarily stored in Staging Database waiting to be migrated. The Loader Database is shared amongst all tenants.

In order to protect PII, database data should be encrypted. Please see the Encrypting section to prepare to encrypt your data in PostgreSQL.

7.17.1 Installation

The following lists the RPM required by Loader Database

Package	Tested Version	RPM Name	Description
Database	9.2.4	postgresql postgresql92-server postgresql92-libs postgresql92-contrib postgresql92-devel	RPMs for postgres are readily available from Linux distributions
glusterfs	3.3.1	glusterfs	We suggest installing a newer version of the RPM from GlusterFS' site other than the RPM from Linux distributors. See below for more details.
glusterfs-fuse	3.3.1	glusterfs-fuse	It provides support to FUSE based clients.

			We suggest installing a newer version of the RPM from GlusterFS' site other than the RPM from Linux distributors. See below for more details.
encfs	1.7.4	fuse-encfs	encFS client to encrypt a volume This RPM is readily available from Linux Distributions.

7.17.2 Configuration

postgres

Allow username/password based Authentication

Update **/var/lib/pgsql/9.2/data/pg_hba.conf**

```
host all all ###.###.###.###/## trust
host replication all ###.###.###.###/## trust
```

###.###.###.### is your postgresql server IP/network

Allow client configuration

Update **/var/lib/pgsql/9.2/data/postgresql.conf**

```
listen_addresses =  '*'# what IP address(es) to listen on;
port = 5432
max_connections = 100
```

Restart PostgreSQL after this configuration change

```
shell> service postgresql-9.2 restart
```

Create Database User

We recommend creating a dedicated user for database operations.

```
shell>  sudo -u postgres createuser -s -e -E -d -P  udl2
```

Create UDL database

You will need to create a database to host the data and grant the user created above permission to the database. We recommend that you name this database, udl2.

```
shell> su - postgres
shell> createdb -e -E utf-8 -O udl2 -W udl2
```

Create udl2 User and Group

The creation of udl2 user and group is required for the Loader Database to change the files in the GlusterFS as the same user being used in the Loader. We recommend using 501 id for group and 502 id for the user, though you just need to make sure it's consistent with the user created in the Loader server(s).

To add the group and user, please execute the following commands,

```
groupadd udl2 -f -g 501
useradd udl2 -g udl2 -u 501
```

Prepare Work Zone Directory

Incoming files into Landing Zone are copied over to an encrypted volume sitting on a gluster. You will set this up with the steps below

1. Mount gluster onto Loader Database server

   ```
   mkdir /mnt/gluster
   mount -t glusterfs <glusterServer>:/gv0 /mnt/gluster
   ```

2. Mount work zone directory from gluster (This is needed for Foreign Data Wrapper to locate the same path for zones folder as defined in INI). Please execute the following commands as root,

   ```
   usermod -G fuse udl2
   echo "user_allow_other" | sudo tee -a /etc/fuse.conf
   usermod -G udl2 postgres
   mkdir /mnt/gluster/udl/zones /opt/edware/zones
   chown -R udl2.udl2 /mnt/gluster/udl/zones/ /opt/edware/zones/
   service postgresql-9.2 restart
   ```

3. As udl2 user, mount the encfs root

   ```
   encfs /mnt/gluster/udl/zones /opt/edware/zones -o allow_other -o umask='007'
   ```

4. Create udl arrivals directories under the enfs root /opt/edware/zones

   ```
   mkdir  /opt/edware/zones/landing/arrivals
   ```

```
mkdir /opt/edware/zones/landing/work
mkdir /opt/edware/zones/landing/history
```

5. as udl2 user, generate ssh key pair if one does not exist. Alternatively, you can specify the file location of a private key in udl2_rsync.args.private_key
6. copy “udl2” user public key to LZ server “root” authorized_keys (/root/.ssh/authorized_keys)
7. try to ssh to LZ server without password as root user.

7.18 Database Staging

Database Staging machine hosts a staging database for each tenant. The staging database gets populated by the Loader, and contains the data delta only (Data since the last migration from staging to production database).

In order to protect PII, database data should be encrypted. Please see the Encrypting section to prepare to encrypt your data in PostgreSQL.

7.18.1 Installation

The following lists the RPM required by Database Staging

Package	Tested Version	RPM Name	Description
Database	9.2.4	postgresql postgresql92-server postgresql92-libs postgresql92-contrib postgresql92-devel	RPMs for postgres are readily available from Linux distributions

7.18.2 Configuration

postgres

Allow username/password based Authentication

Update **/var/lib/pgsql/9.2/data/pg_hba.conf**

```
host all all ###.###.###.###/## trust
host replication all ###.###.###.###/## trust
```

###.###.###.### is your postgresql server IP/network

Allow client configuration

Update **/var/lib/pgsql/9.2/data/postgresql.conf**

```
listen_addresses =  '*'# what IP address(es) to listen on;
port = 5432
max_connections = 100
```

Restart PostgreSQL after this configuration change

```
shell> service postgresql-9.2 restart
```

Create Database User

We recommend creating a dedicated user for database operations. We recommend creating a user named, edware.

```
shell>  sudo -u postgres createuser -s -e -E -d -P  edware
```

Create edware database

You will need to create a database to host the data and grant the user created above permission to the database. We recommend that you name this database, edware.

```
shell> su - postgres
shell> createdb -e -E utf-8 -O edware -W edware
```

You can validate that the user has access by running the following commands,

```
shell> su - postgres
shell> psql -U edware -W
postgres#> \l
postgres#> \du
```

7.19 Gluster (Storage)

GlusterFS is distributed file system capable to scaling to several petabytes and handling thousands of clients. We use GlusterFS to store large number of files including PDFs, CSVs, and landing zone files.

The installation and configuration for GlusterFS is relatively standard, but we have listed our installation recommendation below.

7.19.1 Installation

The following lists the recommended RPM required by Gluster machine.

Note: Even though Linux distributors provide RPM for GlusterFS server, Amplify suggests to use newer version of RPM directly from GlusterFS site.

You will need to set up yum repo configuration for GlusterFS by:

```
wget -P /etc/yum.repos.d
http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo
```

Package	Tested Version	RPM Name	Description
glusterfs	3.3.1	glusterfs-geo-replication glusterfs glusterfs-server glusterfs-fuse	RPMs for GlusterFS server

7.20 Migrator

Migration is a batch process to move records from Staging Database (pre-prod) to Database Master (prod). The process is designed to provide minimal interruption to user reporting.

It's designed to have two roles - a conductor and player(s). Conductor orchestrates the process of removing players from the Database Pool such that they can sync with the latest data to be migrated.

7.20.1 Installation

EdMigrate is part of smarter package. Installing Smarter RPM is required. Also:

Package	Tested Version	RPM Name	Description
RabbitMQ	2.6.1	rabbitmq-server	RabbitMQ RPM is available from Extra Packages for Enterprise Linux (EPEL)

7.20.2 Configuration

RabbitMQ

Please refer to this section in RabbitMQ configuration. For the virtual host name, we recommend that you name it, edmigrate.

edmigrate

Setting up INI file

INI file from Smarter will be used. Please see here for more details.

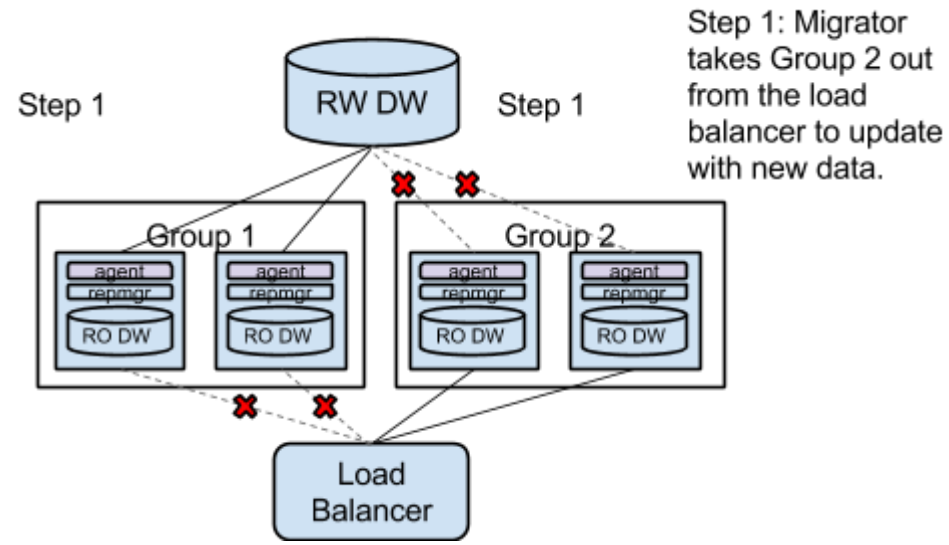
Configuration Name	Description	Example Value
migrate.broadcast.queue	Name of Queue used by Celery from the Conductor to the Players	edmigrate_players
migrate.celery.BROKER_URL	URL of the Message Queue server used by celery	amqp://edware:edware1234@edwappsrv1.poc.dum.edwdc.net/edmigrate
migrate.celery.CELERYBEAT_SCHEDULE	celery task scheduling	[{'seconds': 100, 'task': 'task.edmigrate.master.prepare_edware_data_refresh', 'schedule': 'timedelta', 'name': 'prepare-migration'}, {'seconds': 200, 'task': 'task.edmigrate.master.start_edware_data_refresh', 'schedule': 'timedelta', 'name': 'start-migration'}]
migrate.celery.CELERY_QUEUES	Celery queue used by EdMigrate	[{'exchange': 'fanout', 'key': 'edmigrate_players', 'durable': False, 'name': 'edmigrate_players'}]
migrate.celery.CELERY_RESULT_BACKEND	Type of Queue server	amqp
migrate.celery.CELERY_ROUTES	name of celery route for EdMigrate	[{'edmigrate.tasks.player': {'queue': 'edmigrate_players'}}]
migrate.conductor.enable	Set to True if the server is conductor	True
migrate.conductor.find_player.timeout	How long in second the conductor waits to find the players	5
migrate.conductor.schedule.cron.day	How often conductor runs	*/1
migrate.iptables.chain	Name of chain to use for iptable	EDMIGRATE_PGSQL
migrate.iptables.command	path to iptables	/sbin/iptables
migrate.iptables.mock	for testing purpose, mock iptable for the players	False
migrate.iptables.sudo	path to sudo	/usr/bin/sudo
migrate.master.hostname	hostname of PostgreSQL master server	edwdbsrv1.poc.dum.edwdc.net
migrate.pgpool.hostname	hostname of PgPool server	edwdbsrv4.poc.dum.edwdc.net
migrate.replication_monitor.admin.apply_lag_tolerance	replication tolerance	100
migrate.replication_monitor.admin.check_interval		1000
migrate.replication_monitor.admin.replicatio	replication tolerance	100

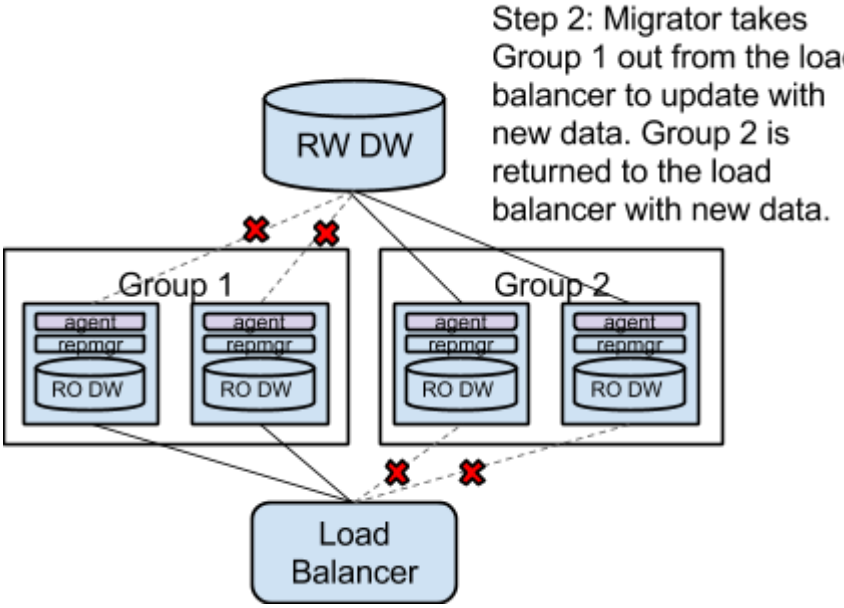
n_lag_tolerance		
migrate.replication_monitor.admin.time_lag_tolerance	replication tolerance	100
migrate.replication_monitor.apply_lag_tolerance	replication tolerance	100
migrate.replication_monitor.monitor_timeout	replication tolerance	28800
migrate.replication_monitor.replication_lag_tolerance	replication tolerance	100
migrate.replication_monitor.time_lag_tolerance	replication tolerance	100
migrate.timeout		5
migrate_dest.db.[tenant].schema_name	schema name production server	edware_prod
migrate_dest.db.[tenant].url	production PostgreSQL server	postgresql+psycopg2://edware:edware2013@localhost:5432/edware
migrate_source.db.[tenant].url	pre-production PostgreSQL server	postgresql+psycopg2://edware:edware2013@localhost:5432/edware
edware_stats.db.schema_name	The schema name of the stats database server	edware_stats
edware_stats.db.url	the database url of the stats database server	postgresql+psycopg2://edware:edware2013@localhost:5432/edware_stats

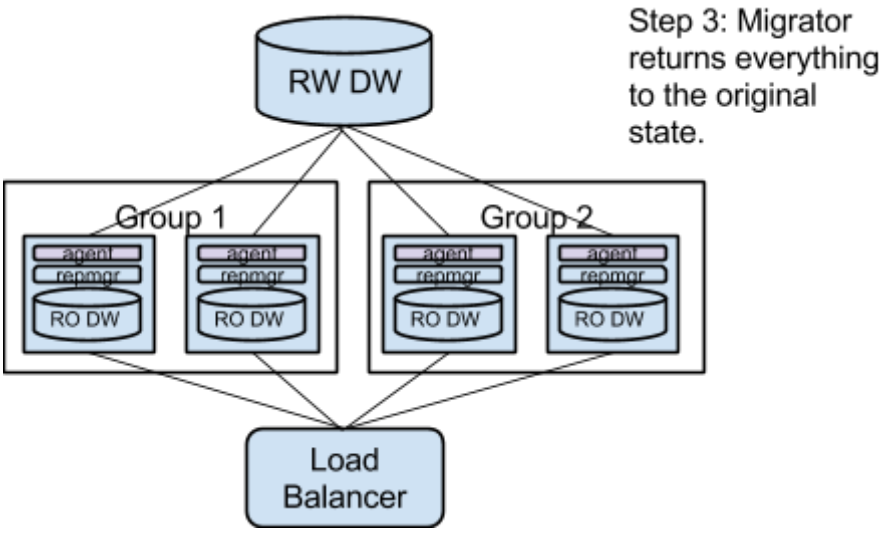
Configure Conductor

1. Register startup service
   shell> `chkconfig --add edmigrate-conductor`
2. Start service
   shell> `service edmigrate-conductor start`
3. Stop service
   shell> `service edmigrate-conductor stop`

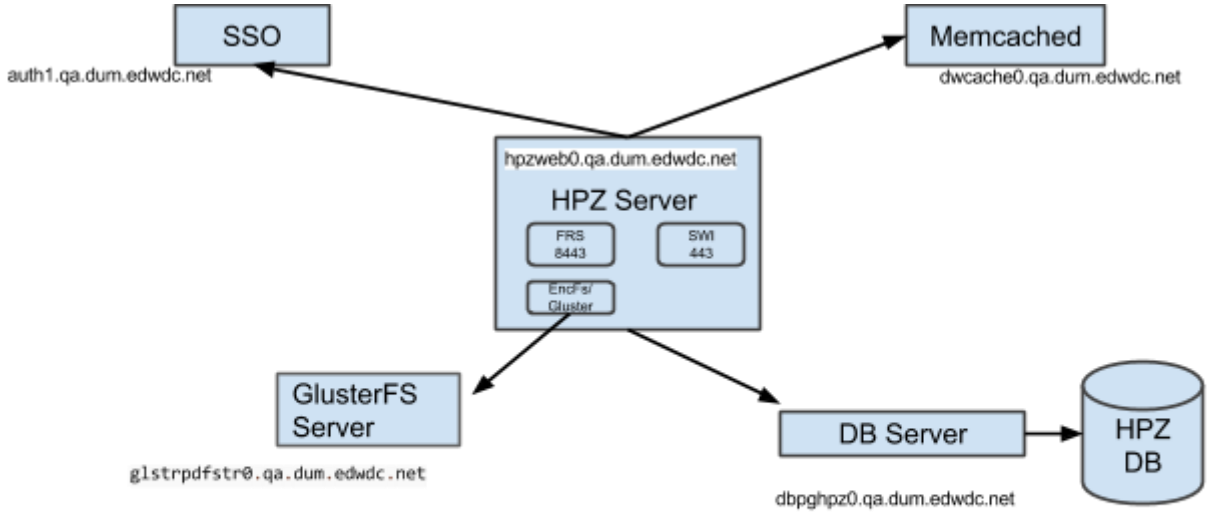
Diagrams to illustrate migration process


Step 1
RW DW
Step 1
Step 1: Migrator takes Group 2 out from the load balancer to update with new data.
Group 1
Group 2
agent
repmgr
RO DW
Load Balancer


Step 2: Migrator takes Group 1 out from the load balancer to update with new data. Group 2 is returned to the load balancer with new data.
RW DW
Group 1
Group 2
agent
repmgr
RO DW
Load Balancer



7.21 HTTPS Pickup Zone Server



7.21.1 Installation

The following RPMs are required for installing hpz

Package	Tested Version	RPM Name	Description

apache module	0.12	mod_xsendfile	RPMs for apache server to process X-SENDFILE headers
xmlsec1	1.2.16	xmlsec1 xmlsec1-openssl	RPMs for XML security
encfs	1.7.4	fuse-encfs	RPMS for encrypt/decrypt hpz data on glusterfs
fuse	2.8.3	fuse fuse-libs	RPMs for fuse file system interface to be used by encfs and glusterfs
gluster	3.3.1	glusterfs glusterfs-fuse	RPMs for hpz to use gluster file server
mod_wsgi	3.4.0	python3-mod_wsgi	RPMs for apache server uses Python wsgi server
python	3.3.0	python3-3.3.0 python3-libs python3-psycopg2	RPMs for python runtime
postgresql	9.2.8	postgresql92 postgresql92-libs	RPMs for hpz server to contact its working database
hpz	0.1	hpz	RPM for http pickup zone

The following RPMs are required for installing hpz database server

Package	Tested Version	RPM Name	Description
postgresql	9.2.8	postgresql92 postgresql92-libs postgresql92-server postgresql92-contrib postgresql92-devel	RPMS for hpz database to keep track of files.

Install IDP Metadata File

7.22.2 Configuration

hpz

Generate .ini file

1. cd /opt/edware/conf
2. source /opt/virtualenv/hpz/bin/activate
3. python generate_ini.py -e qa -o /opt/edware/conf/hpz.ini

Install IDP Metadata File

1. Grab the IDP's metadata file from OpenAM server in your installation) for example https://auth1.qa.dum.edwdc.net/openam/saml2/jsp/exportmetadata.jsp?entityid=https://auth1.qa.dum.edwdc.net/openam then save it as /opt/edware/conf/idp_metadata.xml

postgres

Configure Postgresql

1. Update config file: /var/lib/pgsql/9.2/data/postgresql.conf

Configuration Name	Description	Example Value
listen_address		'*'
shared_buffers		8192MB
work_mem		100MB
log_filename		'postgresql-%a.log'
log_truncate_on_rotation		on
log_rotation_size		0
log_timezone		'UTC'
timezone		'UTC'

2. Updating /var/lib/pgsql/9.2/data/pg_hba.conf
```
host all all ###.###.###.###/## trust
host replication all ###.###.###.###/## trust
host all all 127.0.0.1/32 trust
```
###.###.###.### is your postgresql slave server IP/network.
3. Start the Postgres Database
```
shell> services postgresql-9.2 start
```
4. If postgresql has not used. Initialize postgres,
```
shell> services postgresql-9.2 initdb
```
5. If HPZ datatabase is not created yet, create HPZ Database
```
shell> su - postgres
shell> psql
psql> CREATE DATABASE hpz
```
6. If HPZ user is not created yet, create DB user

```
psql> CREATE USER hpz WITH PASSWORD 'hpz2014';
psql> GRANT ALL PRIVILEGES ON DATABASE hpz to hpz;
```

7. Teardown old HPZ Schemas

```
shell> python3.3 -m hpz.database.metadata_generator --metadata
hpz -a teardown -s hpz -d hpz --host=dbpghpz0.qa.dum.edwdc.net:5432 -u
hpz -p hpz2014
```

8. Initialize HPZ Schema

```
shell> python3.3 -m hpz.database.metadata_generator --metadata
hpz -s hpz -d hpz --host=dbpghpz0.qa.dum.edwdc.net:5432 -u hpz -p
hpz2014
```

hpz

Configure hpz

You can adjust hpz's number of connections by change following options in /opt/edware/conf/hpz.ini, /etc/httpd/conf.d/wsgi_frs.conf, /etc/httpd/conf.d/wsgi_swi.conf and postgresql.conf

hpz.ini

Configuration Name	Description	Example Value
hpz.db_pool_size	For optimal configuration, the pool_size should be equal to number of threads	5

wsgi_swi.ini

Configuration Name	Description	Example Value
WSGIDaemonProcess	number of process and threads for external facing enpoint	swi user=apache group=apache processes=2 threads=30 python-path=/opt/virtualenv/hpz/lib/python3.3/site-packages
MaxClients	must be at least as large as ThreadsPerChild	90

wsgi_frs.ini

Configuration Name	Description	Example Value

WSGIDaemonProcess	number of process and threads used by internal facing endpoint	`frs user=apache group=apache processes=2 threads=30 python-path=/opt/virtualenv/hpz/lib/python3.3/site-packages`
MaxClients	must be at least as large as ThreadsPerChild	`90`

postgres.ini

Configuration Name	Description	Example Value
max_connection	max number of db connections for postgres server	`100`

apache

Configure apache

1. Update apache config file /etc/sysconfig/httpd

Configuration Name	Description	Example Value
HTTPD	*Configuration file for the httpd service.*	`/usr/sbin/httpd.worker`

2. Add /etc/httpd/conf.d/wsgi_frs.conf
3. Add /etc/httpd/conf.d/wsgi_swi.conf

glusterfs

Configure glusterfs

1. Mount glusterfs as root
   ```
   shell> mount -t glusterfs  glstrpdfstr0.qa.dum.edwdc.net:/gv0 /mnt/gluster
   ```
2. Create hpz uploads directory
   ```
   shell> cd /mnt/gluster
   shell> mkdir hpz
   shell> cd hpz
   shell> mkdir uploads
   ```
3. Make apache user owner of the hpz directory
   ```
   shell> cd /mnt/gluster/
   shell> chown -R apache:apache hpz
   ```

encfs

Configure encfs

1. Make the /opt/edware/hpz/uploads, make sure it is owned by apache
   ```
   shell> mkdir /opt/edware/hpz/uploads
   shell> chown apache.apache /opt/edware/hpz/uploads
   ```
2. Make the apache user a member of the fuse group
   ```
   shell> usermod -G fuse apache
   ```
3. Mount EncFs as the apache user, and given a password for encfs
   ```
   shell> sudo su -s /bin/sh apache -c "encfs
   /mnt/gluster/hpz/uploads /opt/edware/hpz/uploads"
   ```

Smoke Test For HPZ

1. Register a file
   a. `curl -X PUT -H "Content-Type: application/json" -d '{"uid":"shall"}' https://hpzweb0.qa.dum.edwdc.net:8443/registration -k`
   b. Capture the registration id and download url from the response
2. Copy a file to HPZ
   a. `curl -X POST -H "File-Name: test.txt" -F "file=@<Path to file>" https://hpzweb0.qa.dum.edwdc.net:8443/files/<registration id> -k`
3. Download the file from HPZ
   a. Use the download url captured in the first step to login as shall and download the test.txt file

8 Starting Applications

The relevant services for each machine type are listed below. You can use the commands below to start and/or stop the services.

Type	Component	Command
Web Servers	Smarter	/etc/init.d/httpd [start|stop]
HTTPS Pickup Zone Servers	https pickup zone server	/etc/init.d/httpd [start|stop]
HTTPS Pickup Zone Servers	encfs	sudo su -s /bin/sh apache -c "encfs /mnt/gluster/hpz/upl /opt/edware/hpz/uploads"
PDF Messenger	RabbitMQ	/etc/init.d/rabbitmq-server [start|stop]

PDF Worker	celeryd-services	/etc/init.d/celeryd-services [start|stop]
PDF Pre-Generator	Smarter	/etc/init.d/httpd [start|stop]
Extract Messenger	RabbitMQ	/etc/init.d/rabbitmq-server [start|stop]
Extract Worker	celeryd-edextract	/etc/init.d/celeryd-services [start|stop]
Cache	memcached	/etc/init.d/memcached [start|stop]
Cache Warmer	Smarter	/etc/init.d/httpd [start|stop]
Database Master	PostgreSQL	/etc/init.d/postgres-9.2 [start|stop]
Database Replica	PostgreSQL	/etc/init.d/postgres-9.2 [start|stop]
Database Load Balancer	PgBouncer	/etc/init.d/pgbouncer [start|stop]
Database Pool	Pgpool-II	/etc/init.d/pgpool [start|stop]
Landing Zone	edsftp-watcher	service edsftp-watcher [start|stop]
Loader	edudl2-trigger	service edudl2-trigger start
Loader	edudl2-file-grabber	service edudl2-file-grabber [start|stop]
Loader	edudl	service celeryd-edudl [start|stop]
Loader Messenger	RabbitMQ	/etc/init.d/rabbitmq-server [start|stop]
Loader Database	PostgreSQL	/etc/init.d/postgres-9.2 [start|stop]
Migrator	Conductor	/etc/init.d/edmigrate-conductor [start|stop]
Migrator	Player	/etc/init.d/celeryd-edmigrate [start|stop]
Database Staging	PostgreSQL	/etc/init.d/postgres-9.2 [start|stop]
GlusterFS	glusterfsd	service glusterd [start|stop]

9 Logging & Monitoring

9.1 Web Server

Apache

The Apache server provides very comprehensive and flexible logging capabilities.

Log File	Description
/var/log/httpd/access_log	Refer to acces
/var/log/httpd/error_log	Refer to error_

Smarter

Smarter has three log files - **`smarter.log`**, **`audit.log`** and **`security_event.log`**. Smarter utilizes syslog service for logging and uses syslog local facilities. By default, **`local0`** is used by audit.log, **`local1`** is used by smarter.log, and **`local2`** is used by audit.log. Using syslog facilities and log levels can be changed inside the ini file. The location of log files is managed by rsyslogd.

Log File	Description
/opt/edware/log/smarter.log	This log contains application leve
/opt/edware/log/audit.log	This log contains information on r
/opt/edware/log/security_event.log	This log contains information on s system related to login, logout, fo SAML2 responses.

9.2 HTTP Pickup Server

Apache

Log File Directory	Description
/var/log/httpd/access.log	The log contains information for http pickup server' access rec
/var/log/httpd/error.log	The log contains information for errors in hpz wsgi server.

HPZ

Log File Directory	Description
/var/log/hpz/log	The log contains information on hpz working status and errors

9.3 PDF Messenger

RabbitMQ

Log File Directory	Description
/var/log/rabbitmq/	All RabbitMQ logging are saved i

9.4 PDF Worker

celeryd-services

Log File	Description
/var/log/celery-services/default_worker.log	Logs for PDF Worker tasks
/var/log/celery-services/batch_worker.log	Logs for PDF Pre-Generator tasks
/var/log/celery-services/health_check_worker.log	Logs for health check worker. This our health check queue.

9.5 PDF Pre-Generator

Smarter

Log File	Description
/opt/edware/log/smarter.log	This log contains application leve

9.6 Extract Messenger

RabbitMQ

Log File Directory	Description
/var/log/rabbitmq/	All RabbitMQ logging are saved

9.7 Extract Worker

celeryd-edextract

Log File	Description
/var/log/celery-edextract/extract_sync_worker.log	Logs for synchronous extraction
/var/log/celery-edextract/extract_worker.log	Logs for asynchronous extractio
/var/log/celery-edextract/extract_archive_worker.log	Logs for archiving step of extrac

9.8 Cache

memcached

Log File	Description
/var/log/memcached.log	Logs related to memcached. B disabled

9.9 Cache Warmer

Smarter

Log File	Description
/opt/edware/log/smarter.log	This log contains application level lo

9.10 Database Master

PostgreSQL

Log File	Description
/var/log/postgres/postgres.log	PostgreSQL application level logs

9.11 Database Replica

PostgreSQL

Log File	Description
/var/log/postgres/postgres.log	PostgreSQL application level logs

9.12 Database Load Balancer

PgBouncer

Log File	Description
/var/log/pgbouncer/pgbouncer.log	Logging of PgBouncer status

9.13 Database Pool

Pgpool-II

Log File	Description
/var/log/pgpool.log	Logging of Pgpool
/var/log/pgpool/pgpool_status	Logging of Pgpool status

9.14 Landing Zone

edsftp-watcher

Log File	Description
/opt/edware/log/smarter.log	This log contains application level lo

9.15 Loader

edudl2-trigger
edudl2-file-grabber

Log File	Description
/opt/edware/log/smarter.log	This log contains application level l

edudl

Log File	Description
/var/log/celeryd-udl2/udl2_worker.log	Logging of edudl service

9.16 Loader Messenger

RabbitMQ

Log File Directory	Description
/var/log/rabbitmq/	All RabbitMQ logging are saved in

9.17 Loader Database

PostgreSQL

Log File	Description
/var/log/postgres/postgres.log	PostgreSQL application level log

9.18 Database Staging

PostgreSQL

Log File	Description
/var/log/postgres/postgres.log	PostgreSQL application level log

9.19 Migrator

Log File	Description
/opt/edware/log/smarter.log	This log contains application level logging.

10 Monitoring

1. Set up Nagios to monitor Smarter Balanced Reporting cluster. Please check http://nagios.sourceforge.net/docs/nagioscore/3/en/, we only tested on Nagios version 3.x. Please check and provide share your feedback and experience for using Nagios 4 to monitor Smarter Balanced Reporting clusters.
2. Set up Nagios monitors for common system
    a. Please monitor all system related, such as disk space, network connectivity, cpu load. Those monitor plugins are available in http://exchange.nagios.org/.
    Use your best judgement and consult with other Nagios users in community

for suitable plugins.

3. Set up Nagios NRPE monitor script for SBAC components:
   a. In order to use Nagios to monitor and alert SBAC, please customize sample Nagios scripts in smarter repo for your use case.
   b. Create a directory /var/spool/nagios/sbac to store sbac alerts to be sent for nagios.
   c. Set up alert-service-by-sbac and alert-host-by-sbac:
      i. defined command 'alert-service-by-sbac' in /etc/nagios/objects/commands.cfg

```
define command {
    command_name alert-service-by-sbac
    command_line /opt/edware/bin/alert-sbac.py --queue-dir \
/var/spool/nagios/sbac enqueue -f sbac_nagios_object=service
 }
```

      ii. defined command 'alert-host-by-sbac' in /etc/nagios/objects/commands.cfg

```
define command {
 command_name alert-host-by-sbac
 command_line /opt/edware/bin/alert-sbac.py --queue-dir
/var/spool/nagios/sbac enqueue -f sbac_nagios_object=service
 }
```

      iii. defined contact configuration in /etc/nagios/object/sbac.cfg

```
define contact {
 contact_name sbac
 alias SBAC-ALERTING
 service_notification_period 24x7
 host_notification_period 24x7
 service_notification_options w,u,c,r
 host_notification_options d,u,r
 service_notification_commands alert-service-by-sbac
 host_notification_commands alert-host-by-sbac
}
```

   d. set up NPRE for check scripts for related services on each machine. Add those configurations into /etc/nagios/npre.cfg. The command name matches the name you will use in nagiosgen.cfg's service check command

```
command[check_smarter]=/opt/edware/bin/check_smarter.sh localhost 80
```

   e. set up nagiosgen.cfg, we use monitoring smarter service as the example

```
define service{
        service_description             SMARTER:Services
        check_command                   check_smarter
        use                             application-service
        hostgroup_name                  smarter-host
```

```
        servicegroups                   smarter-service
}
define hostgroup{
        hostgroup_name                  udl-pgsql-server
        alias                           PGSQL Servers
        notes                           Hostgroup for PGSQL servers
}

define service{
        service_description             UDL-PGSQL
        check_command                   check_tcp!5432
        use                             application-service
        hostgroup_name                  udl-pgsql-server
        servicegroups                   udl-pgsql-service
}
```

4. Set up Nagios notification script to alert SBAC monitoring server.
    a. Set up cron job that sends alerts to sbac

```
* * * * * nagios /opt/edware/bin/sbac_alert.py --queue-dir
/var/spool/nagios/sbac flush
```

11 Troubleshooting

How do I know if the smarter web application is up and running?

When Smarter is up and running, you can navigate to /services/heartbeat endpoint and make sure a 200 OK is returned.

Ex. http://[hostname]/services/heartbeat

A 200 OK is returned only if smarter is able to connect to all the databases that are configured in smarter.ini and that it's heartbeat task is processed in the queue.

How do I tell if memcached is up and running?

```
telnet [server] 11211
stats items
```

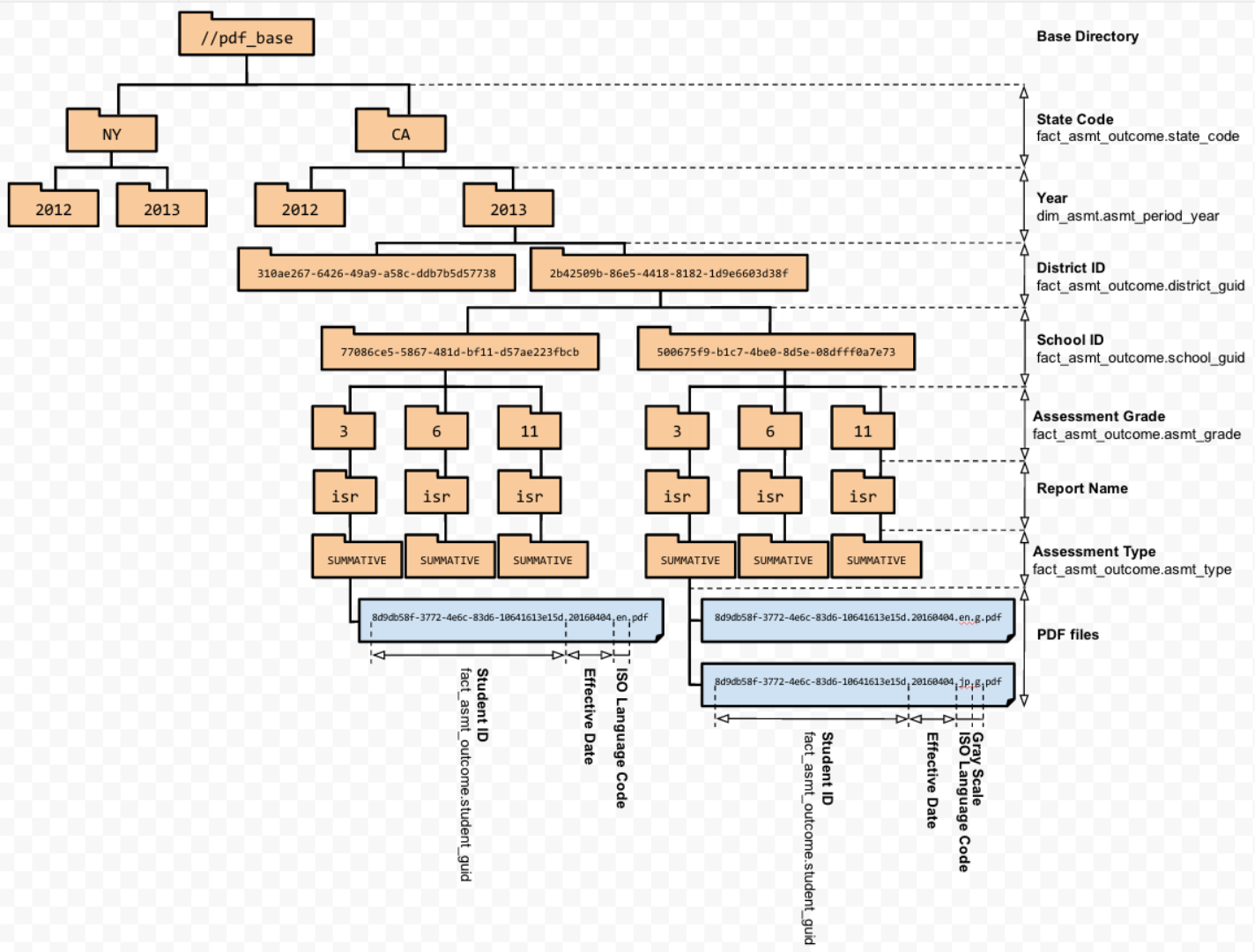
How do I clear the contents in memcache?

The quickest way is to restart memcached service on cache server.

A bad PDF was generated. How do I remove it?

Login as celery user and then change to /opt/edware/pdf directory. Find the PDF and delete it.

The diagram below describes the directory structure in which PDFs are stored.



How do I delete all PDF files?

The quickest way to delete all the PDF files is to delete the directory directory within the volume.

```
rm -rf /export/brick1/vdb1/smarter/[dir]
```

Important Warning: You will see a "**`/export/brick1/vdb1/smarter/.encfs6.xml`**" file. **DO NOT DELETE THE FILE**. It contains EncFS information. Deleting file causes PDF files unrecoverable by EncFS.

How do I clear all queues in RabbitMQ?

Restarting RabbitMQ does not clear all queues. Executing following commands will clear all queues.

```
rabbitmqctl stop_app
rabbitmqctl reset
rabbitmqctl start_app
```

How do I monitor Celery Tasks?

Flower is a real-time web based monitor and administration tool for Celery. It provides graphs and statistics on task details, as well as remote control. An important note is that Celery Flower runs on Python 2.7 (Reference: https://github.com/mher/flower)

How do I list the queues or users configured in RabbitMQ?

To get the list of users, you can execute the following as root user,

```
rabbitmqctl list_users
```

To get the list of queues, you can execute the following as root user,

```
rabbitmqctl list_queues [-p vhostpath]
```

12 Maintenance

12.1 Cache

Clearing Cache

Cache can be flushed only by a user with super admin rights. Using a Rest Client (ex. Chrome Advanced Rest Client), the super admin user can send REST API calls to interface with the cache services.

To delete all data in memcached,
Send a `DELETE` request to `http://[hostname]/services/flush/all`

To delete sessions in memcached,
Send a `DELETE` request to `http://[hostname]/services/flush/session`

To delete reports data in memcached,
Send a `DELETE` request to `http://[hostname]/services/flush/data`

12.2 Database

Cleanup Maintenance [Database, Repmgr and pg_xlogs]

Vacuum and Analyze

Vacuum can be performed on the Loader Database, Staging Database and Production Database Master to compact and release space back to the OS. The frequency of doing this can be different for each databases

Example of Running vacuum on Loader Database:

Determine DB size

> psql -h localhost -U udl2 -d udl2 -c "select pg_size_pretty(pg_database_size('udl2'))"

Run vacuum command while udl is running. This frees up space of the deleted records, but will not release space to OS. This command is non-blocking

> **psql -h localhost -U udl2 -d udl2 -c "VACUUM (VERBOSE, ANALYZE)"**

Run vacuum command while udl2 is not running. This command blocks the database but releases the space to OS and is more efficient than previous

> **psql -h localhost -U udl2 -d udl2 -c "VACUUM (VERBOSE, FULL)"**

http://www.postgresql.org/docs/9.2/static/sql-vacuum.html
http://www.postgresql.org/docs/9.2/static/sql-analyze.html

Transaction log Cleanup

Transaction logs under pg_xlog can be archived continuously by the following settings

```
# To prevent the primary server from removing the WAL segments required for
# the standby server before shipping them, set the minimum number of segments
# retained in the pg_xlog directory. At least wal_keep_segments should be
# larger than the number of segments generated between the beginning of
# online-backup and the startup of streaming replication. If you enable WAL
# archiving to an archive directory accessible from the standby, this may
# not be necessary.
wal_keep_segments = 32

# Enable WAL archiving on the primary to an archive directory accessible from
# the standby. If wal_keep_segments is a high enough number to retain the WAL
# segments required for the standby server, this is not necessary.
archive_mode    = on
archive_command = 'gzip -9 < %p > /mnt/server/archivedir/%f && rm %p'
```

The archive file path should be accessible to the master and the slaves for recovery. In this case we can keep the value of **wal_keep_segments** to minimal such as 32.

We also need to set up recovery.conf to enable postgres recover from gzipped archive wal files.

In recover.conf

```
restore_command = 'gunzip < /mnt/server/archivedir/%f > %p'
archive_cleanup_command = 'pg_archivecleanup /mnt/server/archivedir %r'
```

http://www.mkyong.com/database/postgresql-point-in-time-recovery-incremental-backup/
http://wiki.postgresql.org/wiki/Streaming_Replication

12.3 HTTP Pickup Zone

Cleanup Maintenance

HPZ will store files internally indefinitely. As HPZ accumulates files, cleaning up old files may be needed.

Cleanup via script

Requirement

1. Make sure pickup_zone_cleanup.py is installed on hpz server.

2. The user that the script runs as needs to be able to delete both entries in HPZ's internal database and the files in HPZ's filesystem
3. Python and SQLAlchemy installed

Usage

Parameter	Description
-c, --config	The local path to HPZ's config file. If not provided, will default to /opt/edware/conf/hpz.ini
-e, --expiration	Files (and their related metadata) older than the number of days passed in (as an integer) will be removed

For example, when that HPZ's config file is located at "/the/path/to/my/file.ini" and that all files older than a week that HPZ is storing should be removed.

```
python pickup_zone_cleanup.py -c "/the/path/to/my/file.ini" -e 7
```

HPZ is meant to be used with extract files that can take a lot time to generate, as a general rule, expiration values should be in the range of 3 to 10 days. Anything less, end users may not always be able to receive the file that they requested. Anything more, large files may be hanging around long after they've been used.

The clean up script can be triggered via crond.

Clean up manually

1. Check the download url that corresponds to that file.
2. The url contains an id that corresponds to that file.
3. The format of the url is <hpz host>/download/<file id>
4. On the machine that is running HPZ, navigate to the location where HPZ stores files. (Refer to HPZ's config file if uncertain where HPZ stores them). Remove the file that has the same name as the file id.