



Technical Paper

Managing SAS® Web Infrastructure Platform Data Server High-Availability Clusters on UNIX

Cover: Version 3.0 March 2017

Contents

- 1 Introduction 1**
 - 1.1 Overview 2
 - 1.2 Background 3
 - 1.3 Enhancements to the Process of Configuring the Web Infrastructure Platform Data Server for High Availability 4

- 2 Steps to Configure for High Availability 5**
 - 2.1 Gather Information about the SAS Web Infrastructure Platform Data Server High-Availability Cluster 5
 - 2.2 Install and Configure the SAS Web Infrastructure Platform Data Server within the SAS 9.4 Intelligence Platform Solution 7
 - 2.3 Stop the Middle-Tier and Compute-Tier Servers 7
 - 2.4 Prepare the High Availability Host Machines for Clustering 7
 - 2.5 Install PostgreSQL and pgpool-II on the High Availability Host machines 8
 - 2.6 Configure the High Availability Cluster 10
 - 2.7 Restart All SAS Servers 14
 - 2.8 Install and Configure the PGPOOL-II Administration Tool 14
 - 2.9 Maintain the SAS Web Infrastructure Platform Data Server High-Availability Cluster Using PgpoolAdmin Tool 16
 - 2.9.1 Log into to PgpoolAdmin Tool 16
 - 2.9.2 Monitor the Status of the Cluster 17

- 3 Conclusion 18**

- 4 Appendix 19**
 - 4.1 Cluster Initialization Scripts 19
 - 4.2 Pgpool-II Scripts and Configuration Files 21
 - 4.3 PostgreSQL Scripts and Configuration Files 25

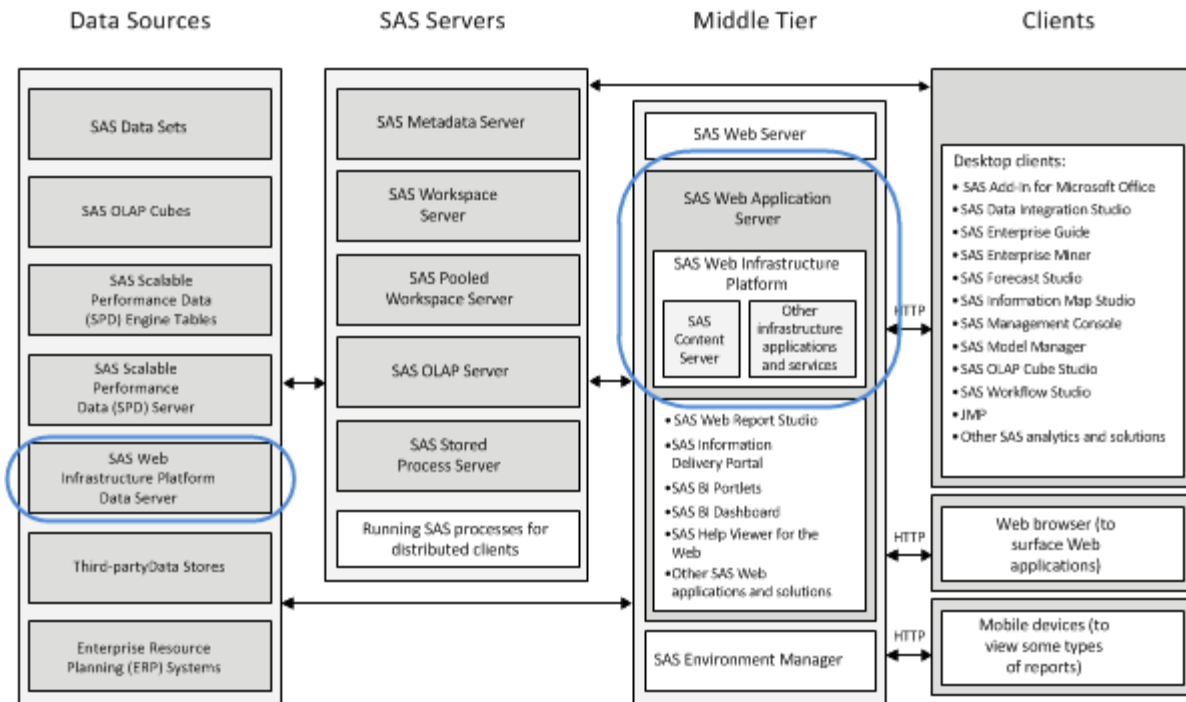
- 5 References and Recommended Reading 29**

1 Introduction

The SAS 9.4 Web Application Server is a lightweight server that provides enterprise-class features for running SAS middle-tier web applications. The SAS Web Infrastructure Platform Data Server, which runs on UNIX and Windows, is used by the SAS Web Infrastructure Platform component of the SAS Web Application Server to provide transactional storage. If you want to configure the data server for high availability on Windows, see [Configuring the SAS Web Infrastructure Platform Data Server for High Availability](#). SAS Environment Manager also uses the SAS Web Infrastructure Platform Data Server for data storage.

Figure 1 shows the relationship among SAS Web Application Server, the SAS Web Infrastructure Platform (WIP), and the SAS WIP Data Server in the SAS Intelligence Platform. It is important to note that there are other SAS solutions that use their own instances of PostgreSQL data servers. High availability for these data servers is possible using the techniques described in this paper. Contact SAS Technical Support for the list of supported data servers.

Figure 1: Middle Tier and Related Components



You can meet the high-availability data requirement in your business plan by implementing a SAS Web Infrastructure Platform Data Server cluster. This paper focuses on how the SAS Web Infrastructure Platform Data Server can be configured for failover and data replication involving multiple nodes. SAS Environment Manager and pgpool-II are used to enable these high-availability strategies, to monitor the server status, and to initiate failover as needed.

1.1 Overview

Why is high availability important in the SAS Web Application Server? The SAS Web Infrastructure Platform is a collection of services and applications that provide common infrastructure features for use by SAS web applications. By default, the SAS Web Infrastructure Platform uses the SAS Web Infrastructure Platform Data Server to store alerts, comments, workflows, and SAS Content Server data. If the SAS Web Infrastructure Platform Data Server becomes unavailable, then the SAS Web Infrastructure Platform services will likewise become unavailable. This effect will cause the entire middle-tier node to effectively go offline, since none of the web server applications will be able to log on successfully. For SAS 9.4, the SAS Web Infrastructure Platform Data Server is version 9.1.9 of the PostgreSQL server that is configured to be used with SAS.

This high-availability configuration enables the primary data server to replicate data to one or more standby data servers. These standby servers are configured to take over as primary server if a failure occurs. High availability is provided for all databases in the SAS Web Infrastructure Platform Data Server that are used by the SAS Web Application Server, including the SharedServices, EVManager, and Administration databases. These databases are described in the *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide*.

This high-availability configuration is summarized in the following table:

SAS Web Infrastructure Platform Data Server	Configuration	Hosted Databases
SAS Web Infrastructure Platform Data Server 1	Set up for replication as the primary data server. This server sends all data to one or more standby servers.	SharedServices, EVManager, Administration, vdbadm, transportsvcs_db, others.
SAS Web Infrastructure Platform Data Server 2+	One or more servers set up for replication as standby. The standby servers receive all data from the primary server but remain in a read-only state.	Read-only copies of the databases, which are replicated from SAS Web Infrastructure Platform Data Server 1 in real time.

The installation and configuration process involves converting the installed PostgreSQL Data Server to be the primary data server and installing and configuring a second PostgreSQL server to be the standby data server. Pgpool-II is also installed as the administration tool.

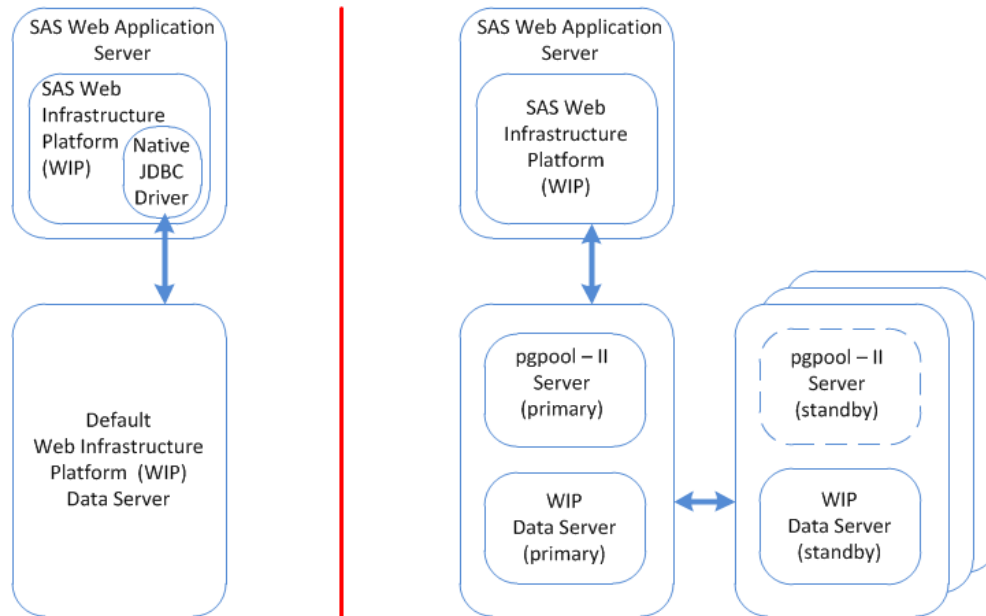
This paper provides the steps to configure and manage the SAS 9.4 Web Infrastructure Platform Data Server for High Availability.

1.2 Background

The SAS Web Infrastructure Platform is a collection of middle-tier services and applications. This collection provides basic integration services and infrastructure that support SAS applications and solutions. The collection is delivered as part of the SAS Integration Technologies package. As such, all business intelligence applications, data integration applications, and SAS solutions have access to the SAS Web Infrastructure Platform as part of their standard product bundling.

Figure 2 shows the default data server configuration and the new configuration described in this paper that replaces the default on UNIX platforms. This new configuration includes a primary data server and a second, standby data server. Additional standby data servers can be added, if necessary.

Figure 2: Default WIP Data Server Configuration (left) and the High-Availability Configuration (right)



1.3 Enhancements to the Process of Configuring the Web Infrastructure Platform Data Server for High Availability

The process described in this paper corrects most of the limitations of the process described in the previously published SAS Technical Paper “Configuring the Web Infrastructure Platform Data Server for High Availability.” Specifically, the following features are now supported:

- Multiple standby high-availability servers can be configured.
- High availability is configured on the server side only. There are no longer any client-tier or middle-tier high-availability dependencies to manage.
- All SAS Web Infrastructure Platform databases can be configured for high availability, including the EVManager and Administration databases.
- All SAS Web Infrastructure Platform Data Server interfaces (for example, JDBC, ODBC, and pg_ctl) are supported.
- The SAS JDBC Driver for High Availability is no longer used.
- SAS Environment Manager has been removed as a single point of failure for controlling data server failover.
- An existing SAS Web Infrastructure Platform Data Server can be updated without SAS Deployment Wizard reconfiguration.
- The primary SAS Web Infrastructure Platform Data Server uses the same data directory as the deployed data server, so the SAS Deployment Backup and Recovery Tool can continue to be used.

2 Steps to Configure for High Availability

The steps to configure and manage the SAS Web Infrastructure Platform for High Availability are:

1. Gather information about the cluster that you will need to complete the install and configure process.
2. Install and configure the SAS Web Infrastructure Platform Data Server within the SAS 9.4 Intelligence Platform solution.
3. Stop the middle-tier and compute-tier servers.
4. Prepare the high availability host machines for clustering.
5. Install PostgreSQL and pgpool-II on the high availability host machines.
6. Configure the high availability cluster.
7. Restart all SAS servers.
8. Install and configure the pgpool-II Administration Tool (pgpoolAdmin).
9. Maintain the SAS Web Infrastructure Platform Data Server high-availability cluster using pgpoolAdmin.

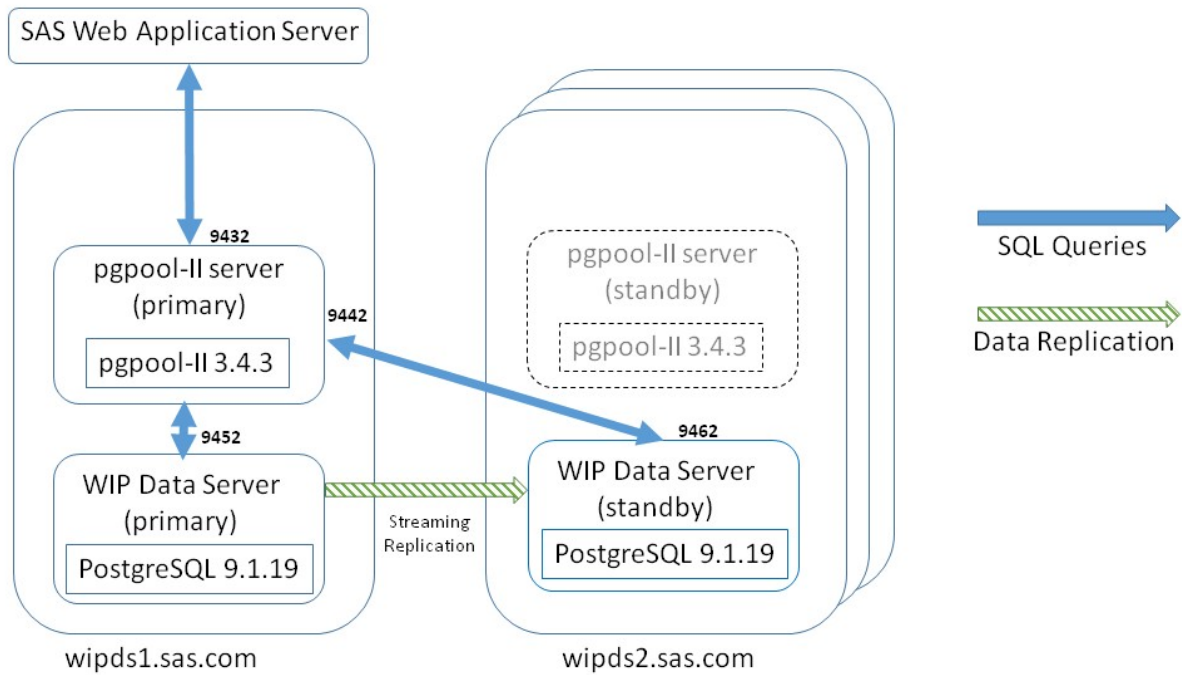
Caution: There is a potential for data loss or corruption if the failover components are configured improperly. Never configure more than one primary SAS Web Infrastructure Platform Data Server for use simultaneously. Doing so can result in data loss or corruption that can be very difficult to recover from without full backups of the databases. Make regular backups of the databases as a best practice. The high-availability configuration is not a substitute for making regular backups.

2.1 Gather Information about the SAS Web Infrastructure Platform Data Server High-Availability Cluster

These instructions assume that you want to deploy one primary data server and one standby data server. However, you can deploy additional standby data servers by configuring more hosts in Steps 4–6. These instructions are limited to deploying a single pgpool-II server, as this configuration is supported by SAS. However, additional pgpool servers (shown in gray in Figure 3) can be deployed on each standby server node. The pgpool-II watchdog, dynamic IP switching, and quorum functions will be needed to support multiple pgpool-II servers.

Before deploying the SAS Web Infrastructure Platform Data Servers, record the IP address, host name, and port number for each server. In our example, the pgpool-II server and primary data server host is wipds1.sas.com. The standby data server host is wipds2.sas.com. Figure 3 shows how these elements relate to one another.

Figure 3: Data Flow between the SAS Web Application Server, the pgpool-II Server, and the SAS Web Infrastructure Platform Data Servers



The following table shows the example values that are used in these instructions. Substitute your site's values for these values:

Host Value	Example Values Used in These Instructions
Hostnames	wipds1.sas.com (pgpool-II and primary data server) wipds2.sas.com (standby data server)
pgpool-II data port	9432 ¹
SAS Web Infrastructure Platform Data Server ports	9452, 9462
SASHome	/install/SASHome
SASCONFIG²	/install/SAS/config/Lev1
WIPCONFIG² (SAS Web Infrastructure Platform Data Server data and log locations)	\$\$SASCONFIG/WebInfrastructurePlatformDataServer
WIPINSTALL² (Data Server installation directory)	\$\$SASHome/SASWebInfrastructurePlatformDataServer/9.4
SAS Web Infrastructure Platform Data Server administrator account (internal)	dbmsowner
Apache HTTP server and pgpool-II account (external)	centos (this account needs root or SUDO privileges)
SAS installer account (external)	cfigsas1 ³

Notes:

¹ Clients of the SAS Web Infrastructure Platform Data Server connect to the pgpool-II data port (9432), not to the port for the SAS Web Infrastructure Platform Data Server (9452). Port 9442 is used internally by pgpool-II.

² SASCONFIG, WIPCONFIG, and WIPINSTALL are UNIX environment variables. Set these to the appropriate values used in your deployment.

³The Linux user account used to run the High Availability PostgreSQL deployment must begin with a letter or underscore. Specifically, it must not have a number as the first character.

2.2 Install and Configure the SAS Web Infrastructure Platform Data Server within the SAS 9.4 Intelligence Platform Solution

The first step is to install and configure the SAS Web Infrastructure Platform Data Server within the SAS 9.4 Intelligence Platform solution. This deployment is performed normally, but take note of the data server specific parameters while executing the SAS Deployment Wizard. This paper assumes that after the deployment is completed, the SAS Web Infrastructure Platform Data Server is installed on a data-tier host (wipds1.sas.com) and that all of the middle-tier products are installed on middle-tier hosts.

2.3 Stop the Middle-Tier and Compute-Tier Servers

In order to reconfigure the SAS Web Infrastructure Platform Data Server for high availability, the middle-tier and compute-tier servers must be shut down in an orderly fashion. Use the `sas.servers` script in the configuration directory to stop the servers on each machine. For example, on each middle-tier host and on `wipds1.sas.com`, execute:

```
> $SASCONFIG/sas.servers stop
```

Execute this command on each machine that contains this script in the configuration directory. As a best practice, stop the middle-tier services before stopping the compute-tier services. For more information about stopping and starting SAS servers, please refer to the [SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide](#).

2.4 Prepare the High Availability Host Machines for Clustering

On each host machine that will contain either a pgpool-II server or a SAS Web Infrastructure Platform Data Server, set up a shared Secure Socket Shell (SSH) key environment. This environment is needed by pgpool-II in order to maintain streaming replication between the primary data server and its standby data servers. Do not change the names provided for the key files, as they are hardcoded in the configuration scripts. To create the key files, perform the following steps:

1. From the SAS installer account, create a public and private key on the primary host.
2. Update the `authorized_key` file with the public key.
3. Change the permissions as appropriate to comply with SSH guidelines.
4. Copy the `.ssh` directory from the primary host to all standby hosts.

For example, open a command shell and perform the following commands:

- a. Generate the keys on the primary host and update the `authorized_keys` file.

```
> rlogin wipds1.sas.com -l cfigsas1
> ssh-keygen -N "" -t rsa -f ~/.ssh/id_rsa_ha_rsync_conn
> cat ~/.ssh/id_rsa_ha_rsync_conn.pub >> ~/.ssh/authorized_keys
> chmod 700 ~/.ssh
> chmod 600 ~/.ssh/authorized_keys
> chmod 600 ~/.ssh/id_rsa_ha_rsync_conn*
> ssh -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no -tt
$(whoami)@$(hostname) 'hostname'
```

- b. Copy the `.ssh` directory from the primary host to the standby host. In the `scp` UNIX command below, replace the `<home directory>` parameter with the actual home directory of the standby host.

```
> rlogin wipds1.sas.com -l cfigsas1
> cd .ssh
> scp -p -i id_rsa_ha_rsync_conn * cfigsas1@wipds2.sas.com:<home directory>/.ssh
```

To verify that the SSH environment has been set up correctly, test a simple UNIX command using SSH on each host. In the following example, the `hostname` command should display the remote host name for specified machine. Executing the SSH command for the first time on each host will require an interactive acknowledgment. Subsequent executions will not prompt for a response. From `wipds1.sas.com`, execute:

```
> ssh -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no -tt $(whoami)
@wipds2.sas.com 'hostname'
```

```
wipds2.sas.com
```

2.5 Install PostgreSQL and pgpool-II on the High Availability Host machines

Even though the SAS Web Infrastructure Platform Data Server contains a working version of PostgreSQL, a new PostgreSQL version must be installed. This is needed because the pgpool-II extensions must be installed into a YUM-deployed version of PostgreSQL.

This step describes how to install PostgreSQL and pgpool-II on the host machines that will be used in the high availability cluster. The latest version of PostgreSQL (currently 9.1.24) will be used instead of the version deployed with SAS Web Infrastructure Platform Data Server so that pgpool-II can be seamlessly integrated. This process does not involve using the SAS Deployment Wizard. Instead, the PostgreSQL and pgpool-II are installed from existing open-source community code repositories.

Using the `yum` (or `rpm`) UNIX system command, install PostgreSQL and pgpool-II on all high availability hosts. In our example, these hosts are `wipds1.sas.com` and `wipds2.sas.com`.

1. Connect (via SSH, Remote Desktop, or whatever is appropriate for your environment) to the machine that will host the standby data server. Log on as a user with root or SUDO privileges.
2. Open a command shell.
3. Install the latest PostgreSQL 9.1 data server version:

```
> sudo rpm -ivh http://yum.postgresql.org/9.1/redhat/rhel-6-x86_64/pgdg-redhat91-9.1-7.noarch.rpm
> sudo yum install postgresql91-devel postgresql91
> sudo yum install postgresql91-server postgresql91-contrib
```

4. Add PostgreSQL into the `PATH` and `LD_LIBRARY_PATH` environment variables. The pgpool-II installation in the next step will install extensions into the PostgreSQL installation location.

```
> export LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib:$LD_LIBRARY_PATH
> export PATH=/usr/pgsql-9.1/bin:$PATH
```

5. Install pgpool-II. Even though a pgpool-II instance will NOT be run on `wipds2.sas.com`, the pgpool-II extensions need to be added to the PostgreSQL installation. A full installation of pgpool-II will also make the standby hosts ready to support multiple pgpool-II servers in the future.

```
> sudo yum install http://yum.postgresql.org/9.1/redhat/rhel-6-x86_64/pgpool-II-91-3.4.3-2.rhel6.x86_64.rpm
> sudo yum install http://yum.postgresql.org/9.1/redhat/rhel-6-x86_64/pgpool-II-91-devel-3.4.3-2.rhel6.x86_64.rpm
> sudo yum install http://yum.postgresql.org/9.1/redhat/rhel-6-x86_64/pgpool-II-91-extensions-3.4.3-2.rhel6.x86_64.rpm
```

6. To verify the correct installation, execute the following commands to show the installed PostgreSQL and pgpool-II packages:

```
> rpm -qa | grep postgresql91
postgresql91-devel-9.1.19-1PGDG.rhel6.x86_64
postgresql91-contrib-9.1.19-1PGDG.rhel6.x86_64
postgresql91-libs-9.1.19-1PGDG.rhel6.x86_64
postgresql91-server-9.1.19-1PGDG.rhel6.x86_64
postgresql91-9.1.19-1PGDG.rhel6.x86_64

> rpm -qa | grep pgpool-II
pgpool-II-91-devel-3.4.3-2.rhel6.x86_64
pgpool-II-91-extensions-3.4.3-2.rhel6.x86_64
pgpool-II-91-3.4.3-2.rhel6.x86_64
```

2.6 Configure the High Availability Cluster

The SAS Web Infrastructure Platform Data Server deployed by the SAS Deployment Wizard on `wipds1.sas.com` needs to be designated as the primary data server for the high availability cluster. To reconfigure this server, the `pg_hba.conf` file, which is located in the configuration directory, needs to be updated to support streaming replication to the standby hosts.

To set up the high availability cluster, there are three cluster initialization scripts available from SAS Technical Support:

1. **ha-pg_init0.sh** (contains the environment variables needed to set up the cluster)
2. **pgpool_configure_multi.sh** (specifies the parameters pgpool-II setup)
3. **pgpool_setup.sh** (initializes the cluster and generates the back-end management scripts)

Place these scripts in the `$WIPCONFIG` directory. These scripts will perform the configuration steps to upgrade the existing data server to primary status, as well as configure all standby data servers. The scripts will also generate the back-end cluster management scripts listed in the appendix.

On `wipds1.sas.com`, manually configure SAS Web Infrastructure Platform Data Server to be the primary high-availability server, as follows:

1. Open a command shell and navigate.
2. **Make a backup copy of the data directory while keeping the existing permissions**, since the existing data directory will now be updated PostgreSQL for cluster use.

```
> cd $WIPCONFIG
> cp -rp data data_orig
```

3. Copy the `pgpool etc` directory into the main `pgpool` installation directory. You will need root privileges to perform this step.

```
> sudo mkdir /usr/pgpool-9.1/etc
> sudo cp -rp /etc/pgpool-II-91/* /usr/pgpool-9.1/etc
```

4. Edit the remote access permissions configuration file, `$WIPCONFIG/data/pg_hba.conf`. Make two sets of changes to this file:
 - a. Temporarily comment out the two lines that specify the MD5 encryption method.
 - b. Add three trusted user lines for replication and general access.

These changes are needed for `pgpool_setup.sh` to generate the cluster. MD5 encryption will be re-enabled once the cluster has been generated. The `pg_hba.conf` should look similar to this:

```
#TYPE DATABASE USER IP-ADDRESS IP-MASK METHOD
# Temporarily disable MD5 encryption so the cluster can be setup
#local all all md5
#host all all all md5
# Set streaming replication to the centos role and trusted host access
host replication centos all trust
host all all all trust
local all all trust
```

Note: You can substitute your actual IP values in the IP-ADDRESS and IP-MASK fields. This allows the cluster to be configured with the trusted database user (“centos”, in this case). Replace this user with the Linux user account that will be used to execute the `pgpool_configure_multi.sh` script. The md5 encryption method must be disabled (commented out) during cluster initialization by the `pgpool_setup.sh` script. After the cluster has been initialized, the data server needs to be secured again with MD5 authentication.

5. The cluster is now ready to be initialized. Set the environment variables in `$WIPCONFIG/ha-pg_init0.sh` on `wipds1.sas.com` to the values you need for your cluster. Pay close attention to ensuring that the `PGHOME` and `PGPOOLHOME` environment variables are set correctly. On a command line, source this script from the `$WIPCONFIG` directory.
 - a. Customize `ha-pg_init0.sh` to your environment. For this example, set the environment values in this file as shown below. See the appendix for more details about setting these environment variables in `ha-pg_init0.sh`.

```
export CLUSTER_PG_NODE_COUNT=2
export CLUSTER_PG_NODE_DIR_LIST="data data2"
export CLUSTER_PG_HOST_LIST="wipds1.sas.com wipds2.sas.com"
export CLUSTER_PG_PORT_LIST="9452 9462"
export CLUSTER_POOL_PRIMARY_NODE_INDEX=0
export CLUSTER_POOL_NODE_COUNT=1
export CLUSTER_POOL_DIR_LIST="pool0"
export CLUSTER_POOL_HOST_LIST="wipds1.sas.com"
export CLUSTER_POOL_PORT_LIST="9432"
export CLUSTER_POOL_PCP_PORT_LIST="9442"
```

- b. Source the script from the top-level cluster directory.

```
> cd $WIPCONFIG
> source ha-pg_init0.sh
```

6. Convert the existing SAS Web Infrastructure Platform Data Server from a single PostgreSQL server to a primary PostgreSQL data server and to create the standby server. Execute `pgpool_configure_multi.sh` on `wipds1.sas.com` as follows. Replace `<password>` with the actual password specified when setting up the SAS Web Infrastructure Platform Data Server using the SAS Deployment Wizard.

```
pgpool_configure_multi.sh dbmsowner <password> | tee results.wipds_2_nodes
```

You should see output similar to the following:

```
> ./pgpool_configure_multi.sh dbmsowner mypassword | tee results.pgpool_2_nodes
```

```
Starting setup in streaming replication mode
creating startall and shutdownall
creating failover script
hostarray set to wipds1.sas.com wipds2.sas.com
Using existing database directory:
/install/SAS/config/Lev1/WebInfrastructurePlatformDataServer/data
updating remote postgresql.conf...
creating remote pgpool_remote_start...
creating remote basebackup.sh...
creating remote recovery.conf...
```

```

creating new database directory:
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data...
The files belonging to this database system will be owned by user "centos".
This user must also own the server process.

The database cluster will be initialized with locale "C".
The default text search configuration will be set to "english".

Data page checksums are disabled.
creating directory /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
creating template1 database in /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data
/base/1 ... ok
initializing pg_authid ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating collations ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
loading PL/pgSQL server-side language ... ok
vacuuming database template1 ... ok
copying template1 to template0 ... ok
copying template1 to postgres ... ok
syncing data to disk ...
WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.
ok

```

Success. You can now start the database server using:

```

/usr/pgsql-9.1/bin/postgres -D
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data
or
/usr/pgsql-9.1/bin/pg_ctl -D /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data
-l logfile start

```

```

done.
updating remote postgresql.conf...
creating remote pgpool_remote_start...
creating remote basebackup.sh...
creating remote recovery.conf...
temporarily start data cluster to create extensions
temporarily start pgpool-II to create standby nodes
 node_id |      hostname      | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
 0       | wipds1.sas.com    | 9452 | 2      | 0.500000 | primary
 1       | wipds2.sas.com    | 9462 | 3      | 0.500000 | standby
(2 rows)

```

```

recovery node 1...done.
creating follow master script
 node_id |      hostname      | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
 0       | wipds1.sas.com    | 9452 | 2      | 0.500000 | primary

```

```
1          | wipds2.sas.com      | 9462 | 2          | 0.500000 | standby
(2 rows)
```

```
shutdown all
```

```
pgpool-II setting for streaming replication mode is done.
To start the whole system, use
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/startall.
To shutdown the whole system, use
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/shutdownall.
pcp command user name is "centos", password is "xxxxxx".
Each PostgreSQL, pgpool-II and pcp port is as follows:
#1 port is 9452
#2 port is 9462
pgpool port is 9432
pcp port is 9442
The info above is in README.port.
updating remote postgresql.conf...
pgpool-II cluster configuration is complete. Execute ./startall to start the servers.
```

7. The cluster has now been initialized. As shown in the preceding output, the data server cluster consists of two data server nodes. The data server running on wipds1.sas.com is configured as the primary data server and the data server on wipds2.sas.com is configured as the standby data server. A status of 2 indicates that the data server is running and actively part of the cluster. By default, the cluster is left in a stopped state after initialization. This set of steps is best performed by an automated script. The `pgpool_setup.sh` script has re-enabled MD5 encryption for the cluster on all nodes. The following back-end scripts and configuration files were either generated or modified by `pgpool_setup.sh`. See the appendix for more details.

- **startall.sh** – starts the pgpool-II server and all PostgreSQL servers
- **shutdownall.sh** – stops the pgpool-II server and all PostgreSQL servers
- **failover.sh** – removes the primary data server from the cluster and promotes the standby data server to primary status.
- **pgpool_remote_start.sh** – remotely starts a primary or standby data server.
- **pgpool_reload** – triggers the pgpool-II server to reload its configuration.
- **basebackup.sh** -- adds a PostgreSQL server to the cluster as a standby data server.
- **recovery.conf** – PostgreSQL configuration file to enable failover.
- **follow_master.sh** – reinitializes streaming replication from the newly promoted primary data server to the standby data servers after failover has occurred.
- **pcp.conf** – configuration file for internal pgpool-II functions.
- **pgpool.conf** – pgpool-II server configuration file.

8. On wipds1.sas.com, back up and modify the `webinfdsvr.sh` script in the `WIPCONFIG` directory to reference the new top level `startall` and `shutdownall` cluster management scripts. No additional changes are needed on wipds2.sas.com or other standby server hosts.

```
> cd $WIPCONFIG
> cp webinfdsvrc.sh webinfdsvrc.sh.orig
> pgpool_update_webinfdsvrc.sh
```

Verify that the changes made to `webinfdsvrc.sh` are correct. On wipds1.sas.com, execute the following:

```
> webinfdsvrc.sh start
> webinfdsvrc.sh status
> webinfdsvrc.sh stop
```

You should see output similar to the following:

```
node_id | hostname | port | status | lb_weight | role
-----+-----+-----+-----+-----+-----
0       | wipds1.sas.com | 9452 | 2      | 0.500000 | primary
1       | wipds2.sas.com | 9462 | 2      | 0.500000 | standby
(2 rows)
```

9. In the SAS Environment Manager middle-tier server configuration file, change the JDBC protocol version from “2” to “3”. Protocol version “3” is the default, if not specified. Specifically, remove the “?protocolVersion=2” option in the JDBC connection string in `config/Lev1/Web/SASEnvironmentManager/server-5.8.-EE/conf/hq-server.conf`.

It should now look similar to the following:

```
#
# Property: server.database-url
#
# The JDBC URL to connect to.
Server.database-url=jdbc:postgresql://wipds1.sas.com:9432/EVManager
```

2.7 Restart All SAS Servers

The middle-tier and compute-tier servers can now be started in an orderly fashion. Use the `sas.servers` script in the configuration directory to start the servers on each machine. For example, on `middle-tier.sas.com` and `wipds1.sas.com`, execute:

```
> export LD_LIBRARY_PATH=/install/SAS/SASHome/
SASWebInfrastructurePlatformDataServer/9.4/lib:$LD_LIBRARY_PATH
```

Execute this command on each machine where the SAS servers were previously stopped. Be sure to run the scripts in the appropriate sequence, as documented in the [SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide](#).

2.8 Install and Configure the PGPOOL-II Administration Tool

The pgpool-II Administration Tool (`pgpoolAdmin`) is a web-based management tool for pgpool-II. It is used to monitor, start, and stop pgpool-II. This tool also controls pgpool-II settings. You can use either YUM (Yellowdog Updater Modifier) or the RPM package manager to retrieve and install UNIX packages. Most of the installation packages need to be retrieved and installed with root system privileges. This is typically done using the `apache` user account. The Apache HTTP server and PHP must be installed on the primary SAS Web Infrastructure Platform Data Server high-availability node (`wipds1.sas.com`). Follow these steps:

1. Install the latest version of PHP, using the following command. This version must be 4.4.2 or higher.

```
> sudo yum install php, php -ini
```

2. Install the PostgreSQL extension for PHP, using the following command:

```
> sudo yum install php-pgsql
```


3. Install the latest version of Apache HTTP Server, using the following command:

```
> sudo yum install httpd
```

4. Install pgpoolAdmin using the following command:

```
> sudo yum install http://www.pgpool.net/yum/rpms/3.4/redhat/rhel-6-x86_64/pgpoolAdmin-3.4.0-1.pgdg.noarch.rpm
```

To verify the correct installation of pgpoolAdmin, execute the following command:

```
> rpm -qi pgpoolAdmin
```

Messages similar to the following appear:

```
Version : 3.4.0 Vendor: (none)
Release : 1.pgdg Build Date: Fri 07 Nov 2014 01:12:45 AM EST
Install Date: Fri 05 Dec 2014 03:16:04 PM EST Build Host: 93eb297ac0f2
Group : Applications/Databases Source RPM: pgpoolAdmin-3.4.0-1.pgdg.src.rpm
Size : 2565595 License: BSD
Signature : (none)
URL : http://pgpool.net/
Summary : PgpoolAdmin - web-based pgpool administration
```

5. Configure pgpoolAdmin to run with the Apache HTTP Server. File ownership of key pgpoolAdmin files need to be changed to the apache user.

```
> sudo chmod 777 /var/www/html/pgpoolAdmin/templates_c
> sudo chown apache:apache /usr/local/etc/pcp.conf
> sudo chmod 644 /usr/local/etc/pcp.conf
> sudo chown apache:apache /var/www/html/pgpoolAdmin/conf/pgmgt.conf.php
> sudo chmod 644 /var/www/html/pgpoolAdmin/conf/pgmgt.conf.php
```

6. Start the Apache HTTP Server, using the following command:

```
> sudo /etc/init.d/httpd start
```

To verify the correct installation of the Apache HTTP Server, execute the following command:

```
> sudo /etc/init.d/httpd status
```

The following message should appear indicating that the server is running:

```
httpd (pid 9655) is running..
```

7. To verify the correct integration and operation of pgpoolAdmin, you should be able to access the PHP information page and the pgpoolAdmin login screen at the following URLs:

```
http://wipds1.sas.com/pgpoolAdmin/install/phpinfo.php
http://wipds1.sas.com/pgpoolAdmin/login.php
```

2.9 Maintain the SAS Web Infrastructure Platform Data Server High-Availability Cluster Using PgpoolAdmin Tool

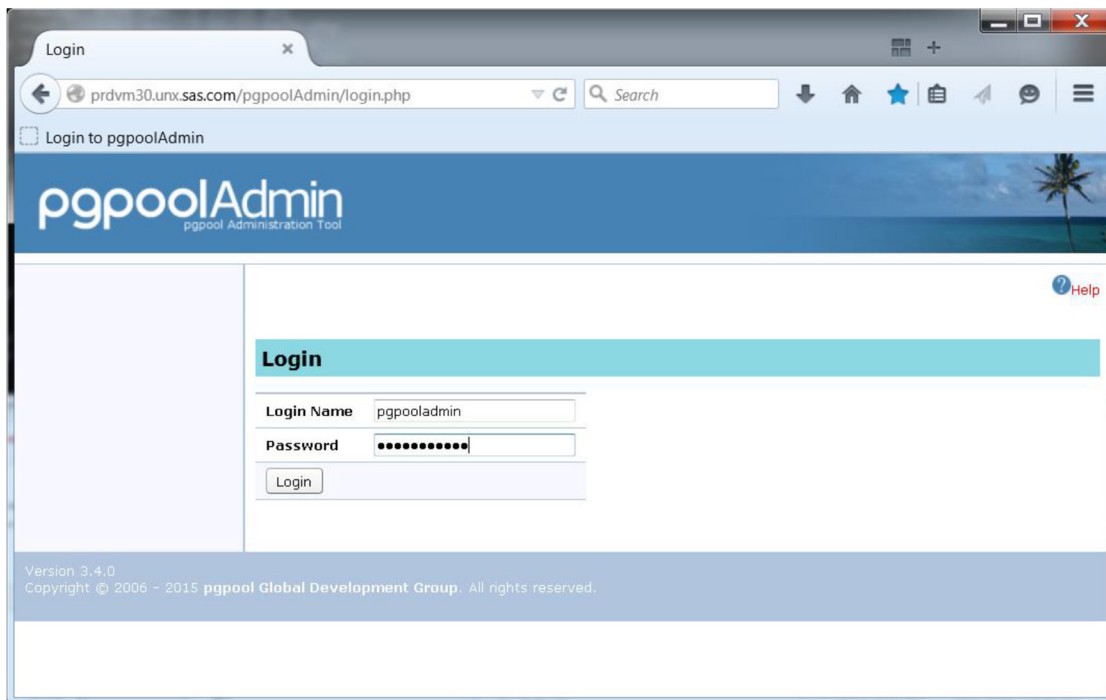
Using the PgpoolAdmin tool, you can do the following cluster management tasks:

- Log in to pgPoolAdmin tool.
- Monitor the status of the server cluster.
- Manually trigger a failover.
- Initiate a failback to a recovered SAS Web Infrastructure Platform Data Server.

2.9.1 Log into to PgpoolAdmin Tool

1. Navigate to the pgpoolAdmin tool login screen.
2. Provide the pgpoolAdmin login name and password as shown in Figure 8.

Figure 4: Log In to pgpoolAdmin Tool



The resulting pgpoolAdmin screen provides tools to manage the data server instances and pgpool-II instances, change various settings, and to trigger failover and failback. The default view is pgpool status information. The panel on the left of the screen provides the other selections.

2.9.2 Monitor the Status of the Cluster

The pgpool Status screen will show the state of the data server nodes, as seen by pgpool-II. There should be one node that is “up, connected, and running as the primary server.” There should be one (or more) nodes that are “up, connected, and running as the standby server.” An example is shown in Figure 9. You can also view the pgpool-II option summary by selecting the Summary button and information about each pgpool-II process by selecting the Process Info. button.

Figure 5: Pgpool-II Status Screen Showing PostgreSQL Node Information

Backend info (PostgreSQL)

Summary Process Info. **Node Info.**

Node Info.

	IP Address	Port	Status	Weight						
node 0	wipds1	9432	Up, Connected, Running as primary server	postgres: Up 0.500	Disconnect	Stop	Restart	Reload	Remove	
node 1	wipds2	9432	Up, Connected, Running as standby server	postgres: Up 0.500	Disconnect	Promote	Stop	Restart	Reload	Remove

Add a new backend node

3 Conclusion

A robust high-availability solution for the SAS Web Infrastructure Platform Data Server can ensure that the middle tier's SAS Web Infrastructure Platform data will always be accessible. This is critical in maintaining a smoothly running SAS Web Application Server and SAS Web Server for your business. This software-based solution should be coupled with hardware-based solutions, such as Error Correction Code (ECC) RAM and redundant disk storage, to achieve a superior high-availability SAS Web Server experience.

4 Appendix

In order to properly manage the SAS Web Infrastructure Platform Data Server High Availability cluster, a complete set of back-end scripts and configuration files are needed. The scripts execute a mixture of UNIX shell commands, pgpool-II commands, and SQL queries to enable cluster management. These management functions include placing the primary server in base backup mode, copying data directories, and promoting a standby to primary status. The set of scripts and configuration files are divided into three categories:

- cluster initialization files
- pgpool-II files
- PostgreSQL files

All of the scripts listed in this section are generated by `pgpool_setup.sh`. The following scripts and configuration files were generated for a two data server cluster, but can be generated for a cluster from 1 to 10 data servers. Replace `$WIPCONFIG` with your SAS Web Infrastructure Platform Data Server configuration directory.

4.1 Cluster Initialization Scripts

All of the pgpool-II scripts reside on the pgpool-II host.

`$WIPCONFIG/pgpool_configure_multi.sh`

```
#!/bin/sh
#
# usage: pgpool_configure.sh <port> <userid> <encrypted password>
# Prerequisites: ha-pg_init0.sh, ha-pginit1.sh and ha-pginit2.sh must be run first.

# Uncomment for debugging
#set -v
#set -x

CLUSTER_PG_PRIMARY_NODE_INDEX=${CLUSTER_PG_PRIMARY_NODE_INDEX:-"0"}
CLUSTER_PG_NODE_COUNT=${CLUSTER_PG_NODE_COUNT:-"1"}
CLUSTER_PG_NODE_DIR_LIST=${CLUSTER_PG_NODE_DIR_LIST:-"data0"}
CLUSTER_POOL_DIR_LIST=${CLUSTER_POOL_DIR_LIST:-"localhost"}
CLUSTER_PG_HOST_LIST=${CLUSTER_PG_HOST_LIST:-"localhost"}
CLUSTER_PG_PORT_LIST=${CLUSTER_PG_PORT_LIST:-"9432 9433 9434"}
```

```

CLUSTER_POOL_PORT_LIST=${CLUSTER_POOL_PORT_LIST:-"9432"}
CLUSTER_POOL_PCP_PORT_LIST=${CLUSTER_POOL_PCP_PORT_LIST:-"9433"}

./pgpool_setup.sh -n $CLUSTER_NODES -p $BASEPORT -u $2 --pass $3 --upgrade

pghostarray=($CLUSTER_PG_HOST_LIST)
pooldirarray=($CLUSTER_POOL_DIR_LIST)

echo "updating remote postgresql.conf..."
n=0
for i in $CLUSTER_PG_NODE_DIR_LIST # "data0 data1 data2 ..."
do
  SSHCMD="ssh -i $HOME/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no
$(whoami)@${pghostarray[$n]} "
  POSTGRES_DATA_DIR=$BASEDIR
  CONF=$POSTGRES_DATA_DIR/$i/postgresql.conf
  $$SSHCMD "echo \"local all all md5\" >>
$POSTGRES_DATA_DIR/$i/pg_hba.conf"
  $$SSHCMD "echo \"host all all md5\" >>
$POSTGRES_DATA_DIR/$i/pg_hba.conf"
  if [ $n != 0 ]; then
    $$SSHCMD "echo \"# Medium usage model settings\" >> $CONF"
    #$$SSHCMD "echo \"log_truncate_on_rotation = on\" >> $CONF" (done by pgpool-II)
    #$$SSHCMD "echo \"log_line_prefix = '%m'\" >> $CONF" (done by pgpool-II)
    $$SSHCMD "echo \"shared_buffers = 256MB\" >> $CONF"
    $$SSHCMD "echo \"work_mem = 2MB\" >> $CONF"
    $$SSHCMD "echo \"maintenance_work_mem = 64MB\" >> $CONF"
    $$SSHCMD "echo \"wal_buffers = 16MB\" >> $CONF"
    $$SSHCMD "echo \"checkpoint_segments = 30\" >> $CONF"
    $$SSHCMD "echo \"checkpoint_completion_target = 0.9\" >> $CONF"
    $$SSHCMD "echo \"effective_cache_size = 2GB\" >> $CONF"
    $$SSHCMD "echo \"max_connections = 256\" >> $CONF"
    #$$SSHCMD "echo \"max_prepared_transactions = 256\" >> $CONF"
    #$$SSHCMD "echo \"log_directory = '$PGHOME/logs'\" >> $CONF"
    #$$SSHCMD "echo \"logging_collector = on\" >> $CONF" (done by pgpool-II)
  fi
  $$SSHCMD "echo \"wal_keep_segments = 200\" >> $CONF"
  n=`expr $n + 1`
done
POOLCONF=$BASEDIR/${pooldirarray[0]}/etc/pgpool.conf
echo "max_pool = 1" >> $POOLCONF
echo "num_init_children = 256" >> $POOLCONF

# ./startall
echo pgpool-II cluster configuration is complete. Execute \"${pooldirarray[0]}/startall\"
to start the servers.

```

\$WIPCONFIG/pgpool_setup.sh

pgpool_setup.sh is not included in this document because it contains over 1000 lines of code. The file can be obtained from SAS Technical Support.

4.2 Pgpool-II Scripts and Configuration Files

All of the pgpool-II scripts reside on the pgpool-II host.

\$WIPCONFIG/ha-pg_init0.sh

This table shows you the environment variables that must be updated in the script to apply to your specific environment. The environment variables to change are shown in ***boldface italics*** in the script.

Variable	Description
<i>CLUSTER_PG_NODE_COUNT</i>	Total number of data servers, including the primary.
<i>CLUSTER_PG_NODE_DIR_LIST</i>	List of data directory names, one for each data server.
<i>CLUSTER_PG_HOST_LIST</i>	List of host machines, one for each data directory.
<i>CLUSTER_PG_PORT_LIST</i>	List of ports, one for each data server.
<i>CLUSTER_POOL_HOST_LIST</i>	List of host machines, one for each pgpool-II server. (Currently, only one host is supported.)
<i>CLUSTER_POOL_PORT_LIST</i>	List of ports, one for each pgpool-II server.
<i>CLUSTER_POOL_PCP_PORT_LIST</i>	(Currently, only one port is supported.)

Note that the variable `CLUSTER_POOL_PRIMARY_NODE_INDEX` should typically remain "0". The variable `CLUSTER_POOL_NODE_COUNT` must be "1".

```
#!/bin/sh

# Set environment variables
export PGHOME=/usr/pgsql-9.1
export PGPOOLHOME=/usr/pgpool-9.1
export PATH=$PGHOME/bin:$PATH
export LD_LIBRARY_PATH=$PGHOME/lib:$PGHOME/lib64:$PGPOOLHOME/lib
export BASHPROFILE=.bash_profile
export PGBIN=$PGHOME/bin
export PGLIB=$PGHOME/lib
export PGPOOL_INSTALL_DIR=$PGPOOLHOME
export PGPOOLDIR=$PGPOOLHOME/etc

export CLUSTER_PG_PRIMARY_NODE_INDEX=0
export CLUSTER_PG_NODE_COUNT=2
export CLUSTER_PG_NODE_DIR_LIST="data data2"
export CLUSTER_PG_HOST_LIST="wipds1.sas.com wipds2.sas.com"
export CLUSTER_PG_PORT_LIST="9452 9462"
export CLUSTER_POOL_PRIMARY_NODE_INDEX=0
export CLUSTER_POOL_NODE_COUNT=1
export CLUSTER_POOL_DIR_LIST="pool0"
export CLUSTER_POOL_HOST_LIST="wipds1.sas.com"
export CLUSTER_POOL_PORT_LIST="9432"
export CLUSTER_POOL_PCP_PORT_LIST="9442"
export BASEDIR=`pwd`
export PGPOOL_LOG_CONF=$BASEDIR/pool0
```

\$WIPCONFIG/pool0/startall

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib
PGPOOL_LOG_CONF=/install/SAS/config/Levl/WebInfrastructurePlatformDataServer
PGPOOL_INSTALL_DIR=/usr/pgpool-9.1
ssh -f -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no cfigsas1@wipds1.sas.com
"export LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib; nohup /usr/pgsql-
9.1/bin/pg_ctl -w -D /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data
start"
sleep 4
ssh -f -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no cfigsas1@wipds2.sas.com
"export LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib; nohup /usr/pgsql-
9.1/bin/pg_ctl -w -D /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data
start"
sleep 4
/usr/pgpool-9.1/bin/pgpool -D -n -f
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/pgpool.conf -F
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/pcp.conf -a
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/pool_hba.conf >
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/log/pgpool.log 2>&1 &
```

\$WIPCONFIG/pool0/shutdownall

```
#!/bin/sh
PGPOOL_LOG_CONF=/install/SAS/config/Levl/WebInfrastructurePlatformDataServer
PGPOOL_INSTALL_DIR=/usr/pgpool-9.1
/usr/pgpool-9.1/bin/pgpool -f
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/pgpool.conf -m f stop
ssh -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no cfigsas1@wipds1.sas.com
"export LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib; /usr/pgsql-9.1/bin/pg_ctl
-D /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data -m f stop"
ssh -i ~/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no cfigsas1@wipds2.sas.com
"export LD_LIBRARY_PATH=/usr/pgsql-9.1/lib:/usr/pgpool-9.1/lib; /usr/pgsql-9.1/bin/pg_ctl
-D /install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data -m f stop"
```

\$WIPCONFIG/pool0/pgpool_reload

```
PGPOOL_LOG_CONF=/install/SAS/config/Levl/WebInfrastructurePlatformDataServer
PGPOOL_INSTALL_DIR=/usr/pgpool-9.1
/usr/pgpool-9.1/bin/pgpool -f
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/pgpool.conf reload
```

\$WIPCONFIG/pool0/etc/failover.sh

```
#!/bin/sh
# Execute command by failover.
# special values: %d = node id
#                 %h = host name
#                 %p = port number
#                 %D = database cluster path
#                 %m = new master node id
#                 %M = old master node id
#                 %H = new master node host name
#                 %P = old primary node id
#                 %R = new master database cluster path
#                 %r = new master port number
#                 %% = '%' character

failed_node_id=$1
failed_host_name=$2
failed_port=$3
```



```

failed_db_cluster=$4
new_master_id=$5
old_master_id=$6
new_master_host_name=$7
old_primary_node_id=$8
new_master_port_number=$9
new_master_db_cluster=${10}
mydir==/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/pool0
log=$mydir/log/failover.log
pg_ctl=/usr/pgsql-9.1/bin/pg_ctl

SSHCMD="ssh -i $HOME/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no
$(whoami)@$new_master_host_name "
EXPLIST="export LD_LIBRARY_PATH=$PGLIB:$PGPOOLHOME/lib"

date >> $log
echo "failed_node_id $failed_node_id failed_host_name $failed_host_name failed_port
$failed_port failed_db_cluster $failed_db_cluster new_master_id $new_master_id
old_master_id $old_master_id new_master_host_name $new_master_host_name
old_primary_node_id $old_primary_node_id new_master_port_number $new_master_port_number
new_master_db_cluster $new_master_db_cluster" >> $log

if [ a"$failed_node_id" = a"$old_primary_node_id" ];then          # master failed
!       new_primary_db_cluster=${mydir}/data"$new_master_id"
        $SSHCMD "mkdir -p $mydir/archivedir" >> $log
        echo "$SSHCMD \"${EXPLIST}\"; $pg_ctl -D $new_primary_db_cluster promote\" \" >> $log
# let standby take over
        $SSHCMD "${EXPLIST}; $pg_ctl -D $new_primary_db_cluster promote" >> $log
fi

```

\$WIPCONFIG/pool0/etc/follow_master.sh

```

#!/bin/sh
# Execute command by failover.
# special values: %d = node id
#                 %h = host name
#                 %p = port number
#                 %D = database cluster path
#                 %m = new master node id
#                 %M = old master node id
#                 %H = new master node host name
#                 %P = old primary node id
#                 %R = new master database cluster path
#                 %r = new master port number
#                 %% = '%' character

failed_node_id=$1
failed_host_name=$2
failed_port=$3
failed_db_cluster=$4
new_master_id=$5
old_master_id=$6
new_master_host_name=$7
old_primary_node_id=$8
new_master_port_number=$9
new_master_db_cluster=${10}
mydir=/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/pool0
log=$mydir/log/failover.log
pg_ctl=/usr/pgsql-9.1/bin/pg_ctl
PCP_PORT=9442
pcp_super_user=centos

```

```

date >> $log
echo "failed_node_id $failed_node_id failed_host_name $failed_host_name failed_port
$failed_port failed_db_cluster $failed_db_cluster new_master_id $new_master_id
old_master_id $old_master_id new_master_host_name $new_master_host_name
old_primary_node_id $old_primary_node_id new_master_port_number $new_master_port_number
new_master_db_cluster $new_master_db_cluster" >> $log

# Stop standby node if it's running
if $pg_ctl -D $failed_db_cluster status|grep "is running" >/dev/null 2>&1
then
    $pg_ctl -w -m f -D $failed_db_cluster stop >> $log 2>&1
    sleep 10
    # recovery the node
    pcp_recovery_node -d 1 localhost $PCP_PORT $pcp_super_user $pcp_super_user
$failed_node_id >> $log 2>&1
else
    echo "$failed_db_cluster is not running. skipping follow master command." >> $log

```

\$WIPCONFIG/pool0/etc/pcp.conf

```

PCP Client Authentication Configuration File
# =====
#
# This file contains user ID and his password for pgpool
# communication manager authentication.
#
# Note that users defined here do not need to be PostgreSQL
# users. These users are authorized ONLY for pgpool
# communication manager.
#
# File Format
# =====
#
# List one UserID and password on a single line. They must
# be concatenated together using ':' (colon) between them.
# No spaces or tabs are allowed anywhere in the line.
#
# Example:
# postgres:e8a48653851e28c69d0506508fb27fc5
#
# Be aware that there will be no spaces or tabs at the
# beginning of the line! although the above example looks
# like so.
#
# Lines beginning with '#' (pound) are comments and will
# be ignored. Again, no spaces or tabs allowed before '#'.

# USERID:MD5PASSWD
centos:<md5 encrypted password>

```

\$WIPCONFIG/pool0/etc/pgpool.conf (changes only)

```
listen_addresses = '*'
sr_check_user = 'centos'
recovery_user = 'centos'
recovery_password = ''
recovery_1st_stage_command = 'basebackup.sh'
health_check_period = 10
health_check_user = 'centos'
memqcache_oiddir =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/log/pgpool/oiddir'
log_per_node_statement = on
failover_command =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/failover.sh %d %h %p %D
%m %M %H %P %r %R'
port = 9432
pcp_port = 9442
pid_file_name =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/run/pgpool.pid'
logdir = '/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/log'
backend_hostname0 = 'wipds1.sas.com'
backend_port0 = 9452
backend_weight0 = 1
backend_data_directory0 =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data'
backend_hostname1 = 'wipds2.sas.com'
backend_port1 = 9462
backend_weight1 = 1
backend_data_directory1 =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/data'
follow_master_command =
'/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/etc/follow_master.sh %d %h %
p %D %m %M %H %P %r %R'
max_pool = 1
num_init_children = 256
```

\$WIPCONFIG/pool0/etc/pool_passwd

```
centos:<md5 encrypted password>
```

4.3 PostgreSQL Scripts and Configuration Files

The PostgreSQL scripts reside in the data directory of each SAS Web Infrastructure Platform Data Server. For reference, this set of scripts is for wipds1.sas.com.

\$WIPCONFIG/data/basebackup.sh

```
#!/bin/sh
psql=/usr/pgsql-9.1/bin/psql
DATADIR_BASE=/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/pool0
PGSUPERUSER=centos

master_db_cluster=$1
recovery_node_host_name=$2
DEST_CLUSTER=$3
PORT=$4
SSHCMD="ssh -i $HOME/.ssh/id_rsa_ha_rsync_conn -o StrictHostKeychecking=no
centos@${recovery_node_host_name} "

log=$DATADIR_BASE/log/recovery.log
```

```

$psql -h wipdsl.sas.com -p $PORT -c "SELECT pg_start_backup('Streaming Replication',
true)" postgres

echo "source: $master_db_cluster dest: $DEST_CLUSTER" >> $log
echo "recovery_node_host_name: $recovery_node_host_name port: $PORT" >> $log

rsync -C -a -c --delete -e "ssh -i $HOME/.ssh/id_rsa_ha_rsync_conn -o
StrictHostKeyChecking=no " --exclude postgresql.conf --exclude postmaster.pid --exclude
postmaster.opts --exclude pg_log --exclude recovery.conf --exclude recovery.done --exclude
pg_xlog $master_db_cluster/ $PGSUPERUSER@${recovery_node_host_name}:${DEST_CLUSTER}/

$SSHCMD "rm -fr $DEST_CLUSTER/pg_xlog"
$SSHCMD "mkdir $DEST_CLUSTER/pg_xlog"
$SSHCMD "chmod 700 $DEST_CLUSTER/pg_xlog"
$SSHCMD "rm $DEST_CLUSTER/recovery.done"
$SSHCMD "echo 'standby_mode          = '\''on'\''' > $DEST_CLUSTER/recovery.conf"
$SSHCMD "echo 'primary_conninfo      = '\''host=wipdsl.sas.com port=$PORT
user=$PGSUPERUSER'\''' >> $DEST_CLUSTER/recovery.conf"
$SSHCMD "echo 'recovery_target_timeline='latest'" >> $DEST_CLUSTER/recovery.conf"
$SSHCMD "echo 'restore_command = '\''cp /archivedir/%f "%p" 2> /dev/null'\''' >>
$DEST_CLUSTER/recovery.conf"

$psql -h wipdsl.sas.com -p $PORT -c "SELECT pg_stop_backup()" postgres

```

\$WIPCONFIG/data/pgpool_remote_start

```
#!/bin/sh
#
# start postmaster on the recovered node
#
if [ $# -ne 2 ]
then
    echo "pgpool_remote_start remote_host remote_datadir"
    exit 1
fi

DEST=$1
DESTDIR=$2
PGCTL=/usr/pgsql-9.1/bin/pg_ctl
ssh -i $HOME/.ssh/id_rsa_ha_rsync_conn -T $DEST "export LD_LIBRARY_PATH=/usr/pgsql-
9.1/lib:/usr/pgpool-9.1/lib; $PGCTL -w -D $DESTDIR start 2>/dev/null 1>/dev/null <
/dev/null &"
```

\$WIPCONFIG/data/postgresql.conf (changed settings only)

```
listen_addresses = '*'
port = 9452
logging_collector = on
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_line_prefix = '%m '
log_truncate_on_rotation = on
log_statement = 'all'
max_prepared_transactions = 10
hot_standby = on
wal_level = hot_standby
max_wal_senders = 2
archive_mode = on
archive_command = 'cp %p
/install/SAS/config/Levl/WebInfrastructurePlatformDataServer/archivedir/%f </dev/null'
```

\$WIPCONFIG/data/pg_hba.conf

```
# TYPE  DATABASE          USER          ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local  all              all           trust
# IPv4 local connections:
host   all              all           127.0.0.1/32    trust
# IPv6 local connections:
host   all              all           ::1/128         trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
host   replication      centos        all              trust
host   all              all          all              trust
host   all              all          all              md5
```

\$WIPCONFIG/data/recover.done (recover.conf on the standby host)

```
standby_mode          = 'on'
primary_conninfo      = 'host=wipds1.sas.com port=9452 user=centos'
recovery_target_timeline='latest'
restore_command       = 'cp /archivedir/%f %p 2> /dev/null'
```

5 References and Recommended Reading

Doninger, Cheryl, Zhiyong Li, and Bryan Wolfe. 2014. "Best Practices for Implementing High Availability for SAS® 9.4." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available at https://support.sas.com/rnd/scalability/grid/305_Doninger_FinalPaper.pdf.

PgPool Global Development Group. 2014. "pgpool-II." Accessed February 9, 2015. Available at <http://www.pgpool.net/docs/pgpool-II-3.4.3/doc/pgpool-en.html>.

PgPool Global Development Group. "pgpool-II Tutorial [watchdog in master-slave mode]." Accessed February 9, 2015. http://www.pgpool.net/pgpool-web/contrib_docs/watchdog_master_slave_3.3/en.html.

PgPool Global Development Group. "pgpoolAdmin: pgpool Administration Tool." Accessed February 9, 2015. http://pgpool.projects.pgfoundry.org/pgpoolAdmin/doc/index_en.html.

SAS® 9.4 Intelligence Platform Administration documentation available at <http://support.sas.com/documentation/onlinedoc/intellplatform/tabs/admin94.html>, in particular these references:

- SAS Institute Inc. 2016. *SAS® 9.4 Intelligence Platform: Middle-Tier Administration Guide*, Fourth Edition. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/bimtag/69826/HTML/default/viewer.htm#titlepage.htm>.
- SAS Institute Inc. 2016. *SAS® 9.4 Intelligence Platform: Web Application Administration Guide*, Sixth Edition. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/biwaag/69837/HTML/default/viewer.htm#titlepage.htm>.

SAS® 9.4 Intelligence Platform Installation, Configuration, and Migration documentation available at <http://support.sas.com/documentation/onlinedoc/intellplatform/tabs/install94.html>, in particular this reference:

- SAS Institute Inc. 2016. *SAS® 9.4 Intelligence Platform: Installation and Configuration Guide*, Second Edition. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/biig/69172/HTML/default/viewer.htm#titlepage.htm>.

SAS Institute Inc. 2016. "Configuring the Web Infrastructure Platform Data Server for High Availability." Available at <http://support.sas.com/resources/papers/configure-wip-4-availability.pdf>

