

Pivotal Greenplum Database Azure Marketplace v4.2 Release Notes

Updated: July 2019

ARM template version 4.2 is based on Pivotal Greenplum Database version 5.20.1.

Overview

Pivotal Greenplum is deployed on Azure using an Azure Resource Manager (ARM) template that has been optimized for efficiency and performance. Pivotal has worked with Microsoft to ensure reliability and strong security for running Greenplum on Azure.

The licensing options on Azure are either Bring Your Own License (BYOL) or billed at an Hourly rate. BYOL does include a 90 day evaluation period but without support and after 90 days, a license must be obtained directly from Pivotal Software, Inc. For customers with a license, BYOL support is treated the same as an on-premise installation. The Hourly rate adds a Pivotal Software cost in addition to the cloud platform cost and does include email support.

Deployment Sizing

Greenplum on Azure BYOL is licensed by the number of cores deployed and it is important to note that in Azure, **1 vCPUs equals 1 core**. Customers will purchase x number of subscription cores and then examine which Instance Type to use and how many.

Instance Count

If possible, use 1 Master + 32 Segments for your cluster. This will give you the highest throughput possible in Azure. Next, pick the Instance Type with the most number of cores that your license allows.

Disk Size

Starting with template version 4.1, you can choose the data disk size and that is a factor in determining the raw and usable storage amount.

To determine the storage size of the Deployment, multiply the number of segment hosts in the cluster times the number of disks per host, and times the size of disk chosen to get the raw storage. Divide the raw storage by two because of mirroring and then multiply this by .7 to leave space for transaction and temp files.

Instance Type	Instance Count	Disk Size
Standard_H16	1-Master-32-Segments	4TB

(32 Segment Hosts * 4 Disks Per Host * 4TB Disks) = 512 TB Raw Storage

(512 / 2) * .7 = 179.2 TB Usable Storage

Instance Types

Azure supports many different Instance Types but all aren't necessarily optimal for Greenplum. After thorough testing of various instance types, Pivotal has found that the HPC instance types work the best for Greenplum. Therefore, we recommend using either the Standard_H16 or Standard_H8 instance type. Standard_D13_v2 and Standard_D14_v2 can also be used but is only recommended if the HPC instance types are not available in your region.

Other instance types in Azure will work but the performance will be less than the HPC instance types using the same number of cores.

Instance Type	Memory	vCPUs	Data Disks	Segs Per Host
Standard_H16	112	16	4	4
Standard_H8	56	8	2	2
Standard_D14_v2	112	16	4	4
Standard_D13_v2	56	8	2	2

Note: on multi-node deployments, the Master node will only have one data disk.

Greenplum on Cloud Utilities and Features

Utility/Feature	Description
Self-Healing	Automatic node replacement and recovery in a failure event.
gpsnap	Automates the execution of snapshot backups and restores using the cloud vendor's native snapshot utilities.
gpcronsnap	Scheduling tool for gpsnap.
gprelease	Automates the upgrade of Pivotal Greenplum, any optional components installed, and the cloud utilities.
gpcronrelease	Scheduling tool for gprelease.
gpmaintain	Automates routine maintenance of Pivotal Greenplum such as vacuum and analyzing tables.
gpcronmaintain	Scheduling tool for gpmaintain.

gpooptional	Installation tool for installing optional components such as MADlib and Command Center.
bouncer	Command line utility that simplifies using pgBouncer.

Overview	1
Deployment Sizing	1
Instance Count	1
Disk Size	1
Instance Types	2
Greenplum on Cloud Utilities and Features	2
Deploying Greenplum on Azure	7
Parameters - Basics	7
Deployment Name	7
gpadmin (Admin user) SSH Public Key	7
Subscription	7
Resource Group	7
Location	8
Parameters - Network	8
Internet Access	8
Subnet Mask	8
Parameters - Instances	8
Instance Type	8
Instance Count	9
Disk Size	9
Parameters - Optional	9
Install Command Center	10
Install Data Science Python	10
Install Data Science R	10
Install MADlib	10
Install PL/R	10
Install PostGIS	10
ARM Template	10
Connecting	12
phpPgAdmin	12
SSH Access	13
Client Tool	14
Additional Resources	14
Deployment Logs	14
Validation	14
Greenplum on Azure Additional Features	14
Self Healing	14
Segment Healing	15
Standby-Master Healing	15

Master Healing	15
Snapshots	15
Steps Creating a Snapshot	15
Steps Restoring a Snapshot	15
Important Considerations and Features for Snapshots	16
gpsnap	16
gpcronsnap	16
Greenplum Maintenance	17
gpmaintain	17
gpcronmaintain	17
Greenplum Upgrades	17
gprelease	17
gpcronrelease	17
pgBouncer	17
bouncer start	18
bouncer stop	18
bouncer pause	18
bouncer resume	18
Optional Installs	19
gpoptional	19
The PXF Extension Framework (PXF)	19
Stopping / Starting Virtual Machines	19
Stopping	20
Step 1 - Stop the database	20
Step 2 - Stop the Virtual Machines	20
Starting	20
Step 1 - Start the Virtual Machines	20
Step 2 - Start the Database and pgBouncer	20
Patching Linux on Azure	20
Yum Update	21
Greenplum on Azure Technical Details and History	21
Azure Resources	21
Image	21
Resource Group	21
Virtual Network	21
Subnet	21
Network Security Group	22
Storage Account	22
Public IP Address	22
Network Interface	22

Availability Set	22
Virtual Machines	22
Storage	22
Root and Swap	22
Data Storage	22
Data Storage	23
Master Storage	23
Diagram	23
Version History	24
Version 4.2	24
Enhancements	24
Version 4.1	24
Fixes	24
Enhancements	24
Version 4.0.0	25
Fixes	25
Enhancements	25
Version 3.5	25
Enhancements	25
Version 3.4	25
Fixes	25
Enhancements	25
Version 3.3	25
Enhancements	25
Version 3.2	26
Enhancements	26
Version 3.1	26
Fixes	26
Enhancements	26
Version 3.0	26
Fixes	26
Enhancements	26

Deploying Greenplum on Azure

Parameters - Basics

The screenshot shows the 'Create Pivotal Greenplum (BYOL)' deployment wizard in the 'Basics' step. The left sidebar contains five steps: 1. Basics (Configure basic settings), 2. Network (Network Configuration), 3. Instances (Type and Count of Nodes), 4. Optional (Optional Installs), and 5. Summary (Pivotal Greenplum (BYOL) by Pivo...). The main area is titled 'Basics' and contains the following fields:

- * Deployment name (3-10 letters or numbers)**: A text input field containing 'greenplum'.
- * gpadmin (Admin user) SSH Public Key**: An empty text input field.
- Subscription**: A dropdown menu with a white selection bar.
- * Resource group**: A dropdown menu with 'Select existing...' and a 'Create new' link below it.
- * Location**: A dropdown menu with '(US) East US' selected.

Deployment Name

This identifies the Pivotal Greenplum Deployment. The default is "greenplum".

gpadmin (Admin user) SSH Public Key

This is your public key used to ssh to the Master node. Password authentication has been disabled on all nodes so the only way to connect via ssh is by creating a key-pair. Use a utility such as ssh-keygen to create your public and private key pairs. Once created, provide the public key as the parameter and reference the private key when connecting via ssh.

Subscription

Pick from one of your existing Azure Subscriptions.

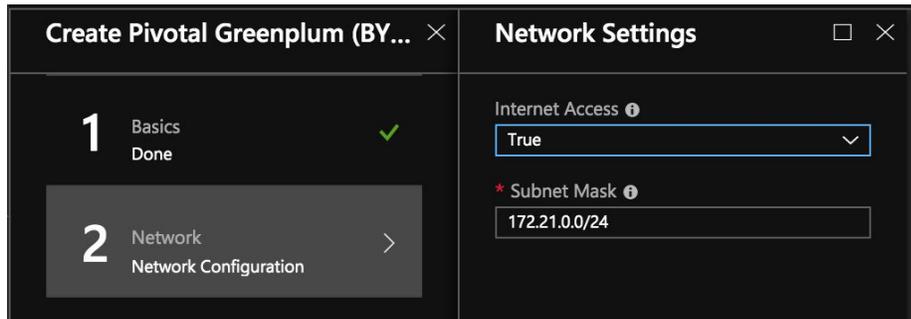
Resource Group

Microsoft only allows a Deployment to be created in an empty Resource Group so either create a new one or pick an existing, empty Resource Group for your deployment.

Location

Provide the location in Azure where you want your cluster to be deployed.

Parameters - Network



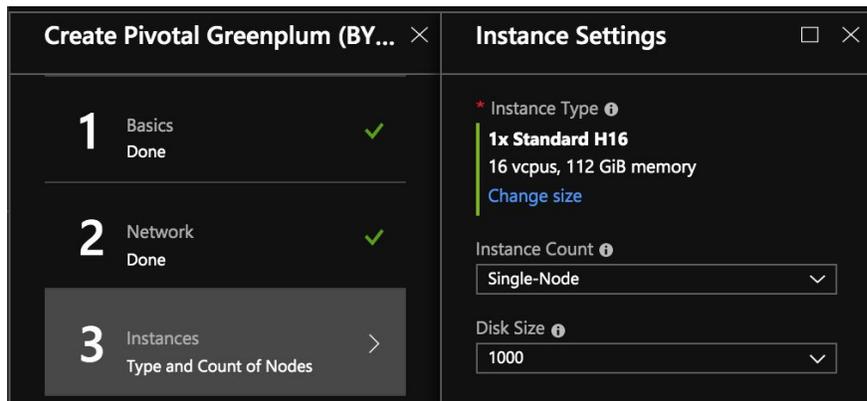
Internet Access

True means a Public IP address will be created for the Master node with ports 22, 5432, 28080, and 28090 open to the Internet. False means the Master will not have a Public IP address created and a jump box will be needed to access the cluster.

Subnet Mask

The Subnet CIDR block that the subnet will cover. You can typically use the default since the deployment is using a dedicated vnet but this added feature gives you more control over how the IP addresses are assigned in the subnet.

Parameters - Instances



Instance Type

Pick from one of the available instance types in the chosen Azure Location.

Instance Count

The number of Instances in the Deployment can be up to 32 segment nodes. When deploying with a Single Node, database mirroring and the Standby-Master will be disabled. The default is Single-Node.

Nodes	Description
Single-Node	Single Node
1-Master-2-Segments	1/16 Rack
1-Master-4-Segments	1/8 Rack
1-Master-8-Segments	1/4 Rack
1-Master-16-Segments	1/2 Rack
1-Master-24-Segments	3/4 Rack
1-Master-32-Segments	1 Rack

Disk Size

Specify the size of each Data Disk attached to the nodes. The number of disks per node is determined by the Instance Type.

Parameters - Optional

Create Pivotal Greenplum (BYOL) by Pivotal

- 1 Basics ✓
Done
- 2 Network ✓
Done
- 3 Instances ✓
Done
- 4 Optional >
Optional Installs
- 5 Summary >
Pivotal Greenplum (BYOL) by Pivotal

Optional Installs

Greenplum Command Center ⓘ
Skip

Data Science Python ⓘ
Skip

Data Science R ⓘ
Skip

MADlib ⓘ
Skip

PL/R ⓘ
Skip

PostGIS ⓘ
Skip

Install Command Center

Indicates if you would like the optional Command Center package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

Install Data Science Python

Indicates if you would like the optional Data Science Python package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

Install Data Science R

Indicates if you would like the optional Data Science R package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

Install MADlib

Indicates if you would like the optional MADlib package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

Install PL/R

Indicates if you would like the optional PL/R package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

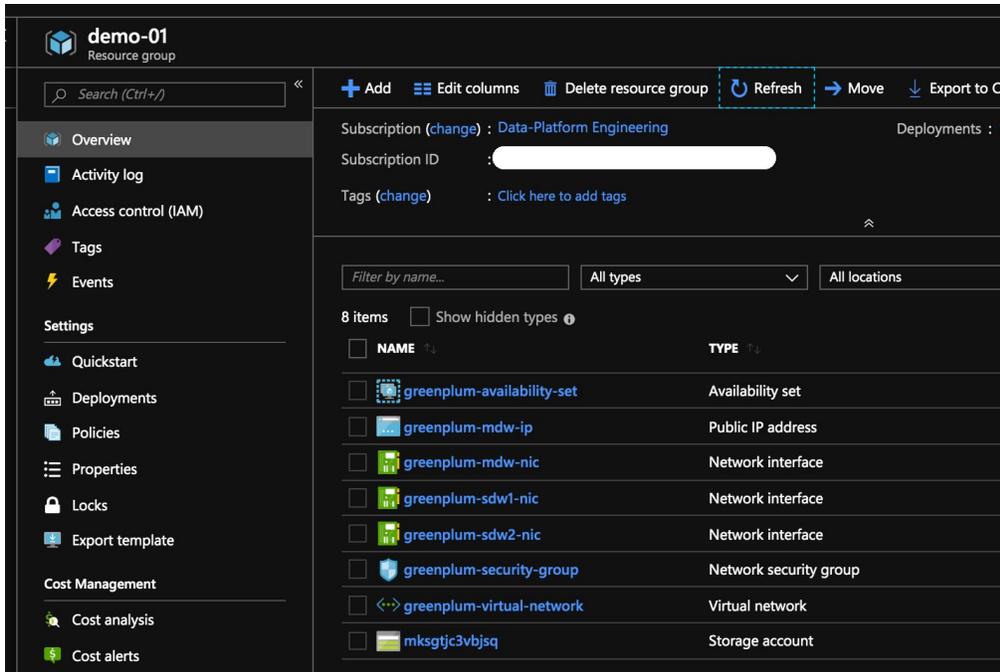
Install PostGIS

Indicates if you would like the optional PostGIS package to be installed or not. If you choose to skip this install initially, you can still run the optional install to install this package. Use `goptional` to install this package.

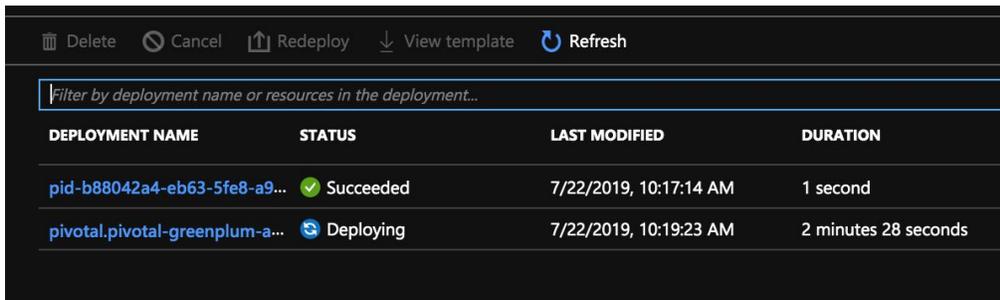
ARM Template

Deployment is very simple in the Azure Marketplace. Simply provide the parameters in the user interface and then submit the ARM template to create the deployment.

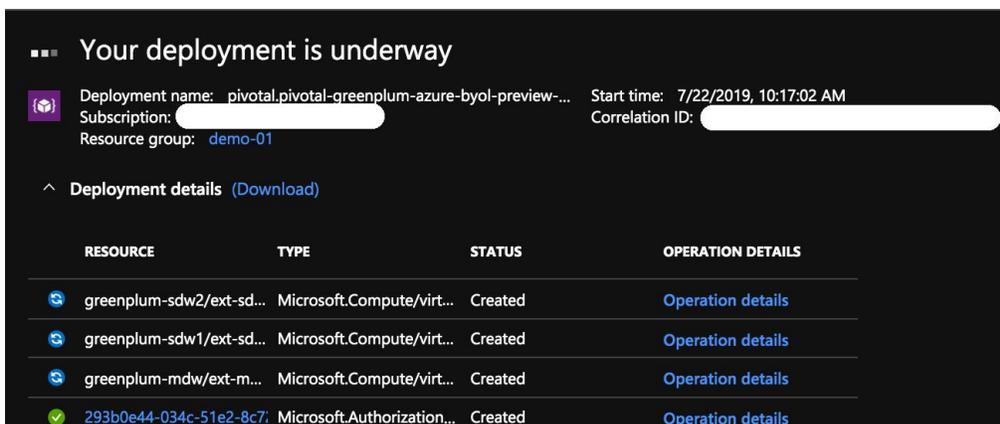
Once the deployment has been submitted, go to the new Resource Group that you created for this deployment.



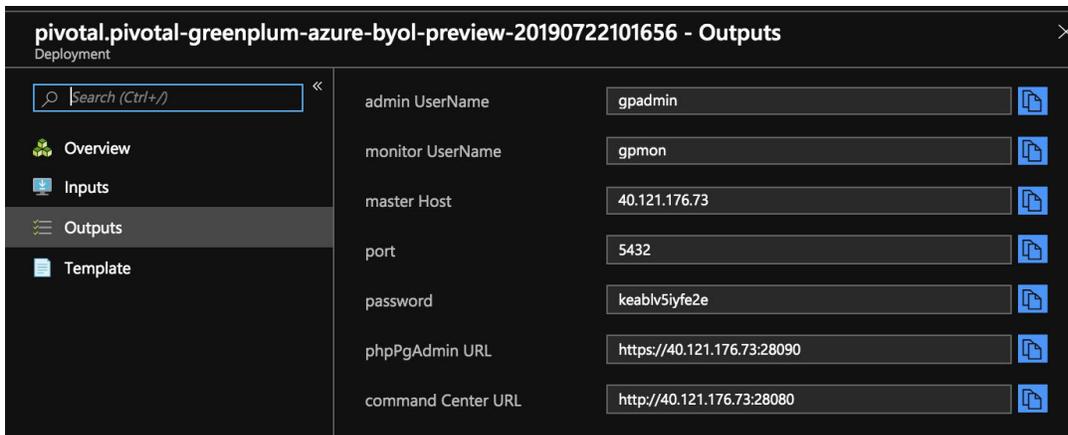
Next, click on Deployments in your Resource Group.



There are two Deployments that are created. One is used by Microsoft to track deployments while the second is the Greenplum cluster. Click on the second deployment (the one that doesn't start with "pid-").



Once complete, Click on the other Deployment and then "Outputs".



The above shows the information needed to get started using the new Pivotal Greenplum on Azure cluster.

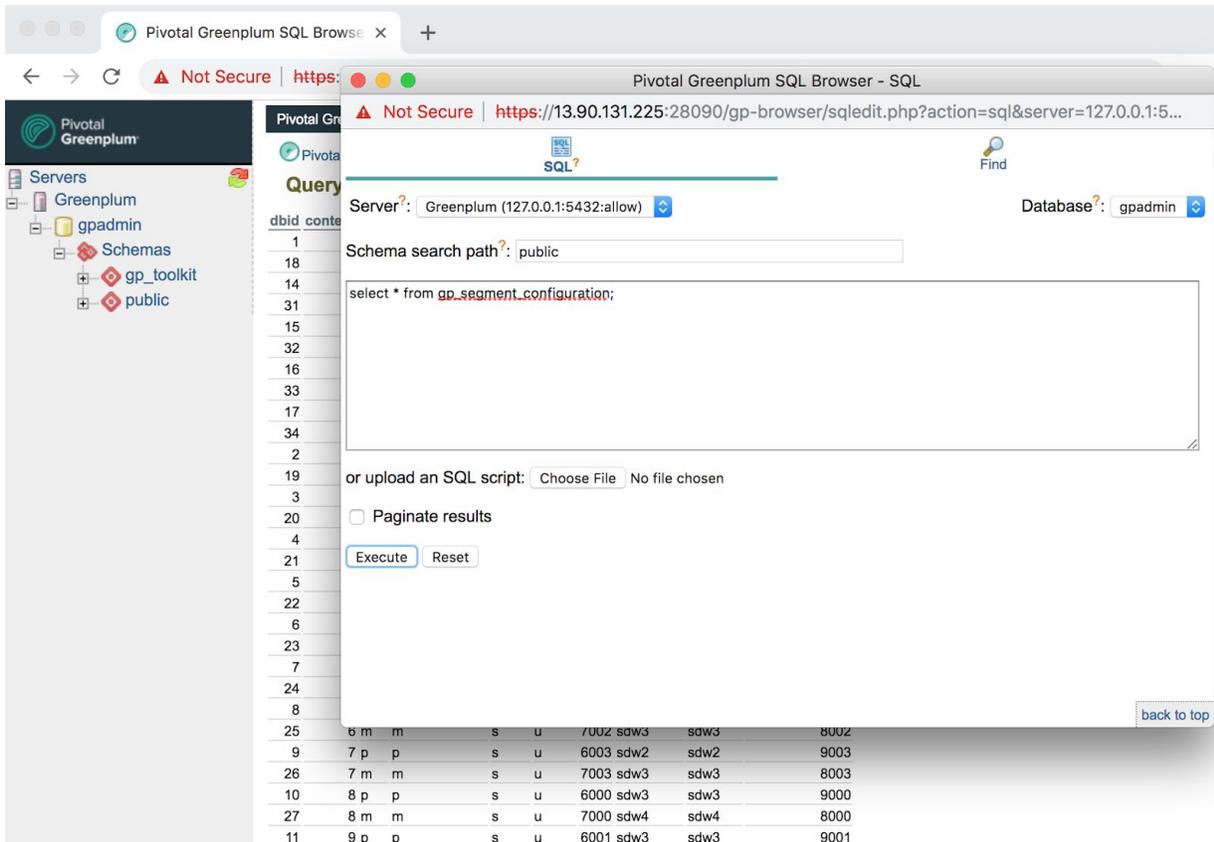
Connecting

Connecting can be done with the web based GP Browser database client, ssh or with an external database client tool like pgAdmin 4. The Deployment Output for Master Host, Port, Admin UserName, and Password used to connect to Greenplum. Note the Password in the Output is the database password for user "gpadmin" and not the password for ssh.

phpPgAdmin

This is a Pivotal enhanced version of the popular phpPgAdmin web based SQL tool. It has been configured to start automatically on the Master node. The URL is provided in the Deployment Output. Connection is simple by using the admin "gpadmin" user and provided database password.

A self-signed SSL certificate is created automatically for you so you will see a web browser warning that the Certificate Authority is invalid. It is safe to proceed with the self-signed certificate.



SSH Access

Password authentication has been disabled so you must use the Key Pair you created for the deployment. To connect with ssh, use your private key that matches the public key you provided when creating the deployment. Connect as "gpadmin" which is the administrator user for Greenplum. The message of the day provides detailed information about the deployment as shown below.

```

*****
Greenplum Version: 5.20.1
Master Data Directory: /data1/master/gpseg-1
Master: mdw
Standby: sdw1
Segment Hosts: sdw1, sdw10, sdw11, sdw12, sdw13, sdw14, sdw15, sdw16, sdw17,
sdw18, sdw19, sdw2, sdw20, sdw21, sdw22, sdw23, sdw24, sdw25, sdw26, sdw27,
sdw28, sdw29, sdw3, sdw30, sdw31, sdw32, sdw4, sdw5, sdw6, sdw7, sdw8, sdw9
Total Segments: 128
Computed storage in GB: 187795
*****
Note: Be sure to switch to gpadmin:
sudo su - gpadmin

[gpadmin@mdw ~]$
    
```

Client Tool

Connecting with a remote client tool like pgAdmin 4 is also very easy to do using the Master public IP address and password provided in the Deployment Output.

Additional Resources

Installation of Pivotal Greenplum on Azure includes detailed logs plus supplemental installs and validation scripts that can be executed after the initial installation is complete.

Deployment Logs

Logs for the deployment of the deployment can be found in: `/opt/pivotal/greenplum/rollout.log`. These logs are on every node but the Master node will have more detailed logs of the database initialization.

Validation

Validation includes scripts to run industry standard benchmarks of TPC-H and TPC-DS. It also includes scripts to validate the disk and network performance of the deployment using the Pivotal Greenplum utility "gpcheckperf".

Greenplum on Azure Additional Features

Self Healing

Azure automatically has "service healing" which is a process where a bad node gets replaced automatically. The bad node will be shutdown and a new node will be brought online automatically. The new node will retain all disks as well.

Pivotal Greenplum on Azure will automatically run the commands needed to heal the database. It is done by the startup script `/opt/pivotal/greenplum/azure_heal.sh`.

Monitoring of the execution of the healing script on a each node can be done with this operating system command:

```
sudo journalctl -u greenplum.service
```

Segment Healing

When a segment fails, the node on restart will automatically detect that the database is up but needing to recover this one node. It will run `gprecoverseg` automatically and when ready to rebalance, it will then pause `pgBouncer` so current queries can complete, and then it will rebalance the cluster. Lastly, it will resume `pgBouncer`.

Standby-Master Healing

In the event that the first segment host were to fail which also has the Standby-Master process, the self-healing process will recover automatically. The Standby-Master process is restarted automatically. Database activity may continue during the Standby-Master Healing process.

Master Healing

In the event that the Master were to fail, the new node executes a few Greenplum utilities. The process first will fail over to the Standby-Master process running on the first segment host followed immediately by a database shutdown. The Standby-Master is then returned back to a Standby-Master role and the new Master node is set as the Master.

The database is up and operational at this point but the database statistics were lost in this process so users are still not allowed to connect. The Master Healing process next executes the `analyzedb` command on every database in the Greenplum cluster to gather the needed statistics. The `pgBouncer` load balancer is restarted and normal database activity may resume.

Snapshots

Deployments in Azure have a snapshot feature called ***gpsnap*** which is very useful for quickly creating a database backup. The snapshot utility automates the execution of creating, listing, deleting and most importantly, restoring a collection of Disk snapshots using Azure commands.

Steps Creating a Snapshot

- Stop the database
- Create snapshots in parallel (one process per host)
- Start the database

Steps Restoring a Snapshot

- Stop `pgBouncer`
- Stop the database
- Umount volumes
- Detach volumes
- Delete volumes
- Create new volumes from Snapshots

- Attach new volumes
- Mount new volumes
- Start the database
- Start pgBouncer

Each snapshot must be labeled correctly so that when a restore is desired, the volumes get attached to the right hosts and mounted as the right volumes. This is done automatically with `gpsnap`.

Important Considerations and Features for Snapshots

- Because of limitation in Azure, Snapshots are not a solution for Disaster Recovery
- Snapshots are stored in the same Resource Group as your Greenplum on Azure Deployment

gpsnap

This utility manages creating, listing, deleting, and restoring snapshots. Please note that creating or restoring a snapshot will restart the database. Here are the list of parameters used with `gpsnap`:

- `gpsnap list`: lists snapshots
- `gpsnap create`: creates a new snapshot
- `gpsnap delete <snapshot_id>`: deletes a specific snapshot
- `gpsnap restore <snapshot_id>`: restores a specific snapshot

gpcronsnap

This utility manages the automatic execution of `gpsnap`. By default, there will be a cron job that runs every 10 minutes and using the configuration file:

`/usr/local/greenplum-cloud/conf/gpcronsnap.conf` to determine if a snapshot is needed or not.

gpcronsnap.conf

```
#maximum number of snapshots; delete the oldest when max reached
max_snapshots=4
```

```
#snapshot day of week (1..7); 1 is Monday
#to specify daily, use (1 2 3 4 5 6 7)
snapshot_dow=(7)
```

```
#time of day to run the snapshot
#do not schedule a time where the snapshot may not finish until the next day
snapshot_time=04:00
```

As shown above, the default schedule is a weekly snapshot on Sunday at 4:00 AM in the local timezone. Four snapshots will be retained before the oldest snapshot will be automatically deleted.

Greenplum Maintenance

gpmaintain

This utility automates basic maintenance of Greenplum. It analyzes all tables using `analyze`, vacuums tables that are in need, and maintains the catalog.

gpcronmaintain

This is the command that is executed via cron and looks at the schedule file in `/usr/local/greenplum-cloud/conf/gpcronmaintain.conf`. By default, this runs at 3:00 AM every Saturday.

Greenplum Upgrades

gprelease

This utility upgrades a Greenplum on Azure cluster to the latest database release available. The tool automatically downloads the binaries, copies it to the hosts in the cluster, stops the cluster, installs the new version, and then starts the cluster again. The tool automatically executes `gproptional` so that optionally installed packages are re-installed or upgraded to a compatible version.

gpcronrelease

This utility checks to see if a new release is available. By default, this runs in cron weekly on Sunday at 12:00 AM in the local timezone. If a new version is available, the message of the day is updated to indicate a new version is available.

pgBouncer

It is highly encouraged to use pgBouncer with deployments in Azure because of "flow table" limitations that are unique to Azure.

This is a load balancing utility that is included with Greenplum. This utility allows far greater connections to the database with less impact on resources. It is recommended to use pgBouncer

instead of connecting directly to the database. More information on pgBouncer is available in the Greenplum documentation.

pgBouncer is configured to listen on port 5432 which is the default port usually used by Greenplum. Greenplum has been configured to listen on port 6432.

Authentication has been configured to use "md5" which is encrypted password. Create users and assign passwords in Greenplum as normal and pgBouncer will authenticate users with the database passwords you set. Other authentication schemes such as LDAP can be configured with pgBouncer post-installation.

Pooling has been configured for "transaction" with max client connections of 1000. These settings can be changed but these defaults provide a good starting point for most installations.

Configuration and logs for pgBouncer are located in /data1/master/pgbouncer on the Master node.

Note that for JDBC connections, you may need to "search_path" to the ignore_startup_parameters configuration item in the ini file.

Connections can optionally be made with SSL to secure connections from your client to the database.

Lastly, the "bouncer" utility has been added to make it easier to start and stop pgBouncer.

bouncer start

Starts pgBouncer. Run this on the Master host.

bouncer stop

Stops pgBouncer. Run this on the Master host.

bouncer pause

Pauses pgBouncer. Run this on the Master host.

bouncer resume

Resumes pgBouncer. Run this on the Master host.

Optional Installs

Many of the commonly used packages are included as optional installs. These packages can be installed during the initial deployment or after the deployment has completed.

gpooptional

This utility simplifies installing optional components during the initial deployment and also after the deployment has been completed. Simply run "gpooptional" to see the optional installation options.

This tool is also used in conjunction with gprelease to upgrade or reinstall already installed optional packages.

The PXF Extension Framework (PXF)

The PXF Extension Framework (PXF) provides parallel, high throughput data access and federated queries across heterogeneous data sources via built-in connectors that map a Greenplum Database external table definition to an external data source. This Greenplum Database extension is based on PXF from Apache HAWQ (incubating).

PXF has been pre-configured in your cluster with settings stored in `/usr/local/greenplum-pxf/`. To start PXF in your cluster, simply execute this command as `gpadmin` on the Master (`mdw`) node.

```
pxf cluster start
```

And to stop PXF:

```
pxf cluster stop
```

Self-Healing and upgrades via `gprelease` have been enhanced to restart PXF if the process was found to be running.

More information about PXF can be found here:
https://gpdb.docs.pivotal.io/5170/pxf/overview_pxf.html

Stopping / Starting Virtual Machines

In order to stop the virtual machines in your cluster to save on infrastructure costs or to apply patches, you first need to stop the database and then deallocate the virtual machines.

Stopping

Step 1 - Stop the database

Login as gpadmin to the Master node and then execute "bouncer stop" to stop pgBouncer. Next, execute the "gpstop" command to stop the database.

```
gpstop -a
bouncer stop
```

Step 2 - Stop the Virtual Machines

You can do this from a terminal window or from the Azure console. To save on infrastructure costs, you need to not only stop the virtual machines but also deallocate the nodes. The Azure console does this automatically for you when you issue a stop command so use this instead of stopping from a terminal window.

Starting

Step 1 - Start the Virtual Machines

Use the Azure console to start all virtual machines. These can be identified by looking at the virtual machines in your Resource Group.

Step 2 - Start the Database and pgBouncer

Wait a few minutes and then connect to the Master node as gpadmin with ssh. Use gpssh to verify that all hosts are up.

```
gpssh -f /usr/local/greenplum-cloud/conf/all_hosts.txt "uptime"
```

After confirming all hosts are up, use "gpstart" and "bouncer start" to start the database and the connection pooler.

```
gpstart -a
bouncer start
```

Patching Linux on Azure

After you have deployed your cluster in Azure, you may need to patch the operating system to address vulnerabilities and in order to do that, you have to take a few extra steps compared to an on-premise installation.

Follow the steps outlined for stopping and starting the Azure cluster above but before you stop the Virtual Machines, use yum to patch the system.

Yum Update

Logged into the Master nodes as gadmin, use "gpssh -f all_hosts.txt" to connect to all hosts via ssh. The all_hosts.txt file is in the gadmin home directory on the Master node. Execute the command to run the yum update.

```
gpssh -f /usr/local/greenplum-cloud/conf/all_hosts.txt "sudo yum
update --security -y"
exit
```

Note: Be patient as this executes on all nodes in parallel. You will not see any output until the command completes on all nodes.

Greenplum on Azure Technical Details and History

Azure Resources

Image

The Image is based off of CentOS 7.6 with Accelerated Networking enabled. The image has all the software packages and pre-reqs for installing Pivotal Greenplum and necessary add-ons including Intel-enhanced networking drivers.

Resource Group

This is a collection of resources in Azure that you want to manage as a single entity. It also specifies the Region. Only one Greenplum Deployment is allowed in the same Resource Group.

Virtual Network

A dedicated Virtual Network is used when deploying Greenplum on Azure. This isolates traffic for Greenplum and helps to ensure it remains secure.

Subnet

Specifies the IP address range for nodes deployed in the Virtual Network. This is the IP address range used by the Interconnect traffic for Greenplum. DHCP is used to assign IP addresses to nodes deployed in the Subnet.

Network Security Group

This specifies which ports and protocols are allowed between nodes in the cluster as well as from the Internet. All traffic is allowed between all nodes on all protocols. TCP Ports 28080 (HTTP), 22 (ssh), and 5432 (Greenplum) are exposed on the 1 to 2 Master nodes.

Storage Account

This is used to enable boot diagnostics in Azure. Boot diagnostics are needed to enable serial access to the virtual machines.

Public IP Address

Static public IP Addresses are assigned to the 1 to 2 Master nodes.

Network Interface

All nodes in the cluster have a Network Interface, or NIC attached to it. The Master nodes have NICs with Public IP Addresses while the Segment Nodes do not.

Availability Set

An Availability Set is defined for the deployment and it specifies the number of Fault and Update domains. This is done to minimize the impact to Greenplum in the event of Azure restarting a node.

Virtual Machines

These are the nodes deployed in the cluster. It has Disks attached to each node and specified in the ARM template. It also includes the deployment execution of the Pivotal scripts executed on each node.

Console access has been enabled by setting the root password. The default root password is the same as the randomly generated password for gpadmin but with the additional "#" character at the end.

Storage

Root and Swap

Storage for the root partition is fixed at 32 GB each. The swap partition uses the automatic temp disk created by Azure and is sized the same as RAM unless RAM is greater than 32 GB and then swap is set to 32 GB.

Data Storage

Disks are mounted with "rw,noatime,nobarrier,nodev,inode64,allocsize=16m 0 2" and blockdev read ahead of 16385. The scheduler is set to deadline.

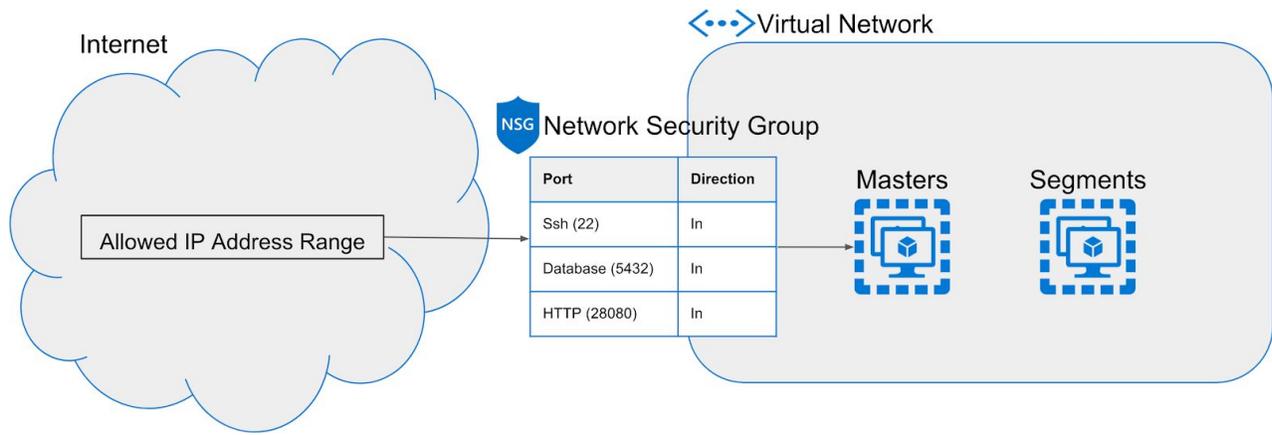
Data Storage

The storage option has been configured with Standard_LRS disks which are optimized for throughput instead of IOPs. Disk configuration has been optimized based on VM Type.

Master Storage

The data storage needs on the Master is much less than on the Segment nodes so there will be only 1 "data" mount on the Master while the Segment nodes will have up to 4.

Diagram



Version History

Version 4.2

Enhancements

- Upgraded to Pivotal Greenplum 5.20.1 and Command Center 4.7.0.
- Upgraded CentOS kernel to 7.6.
- Added deployment parameter to specify if the Master node will have a public ip address or not.
- The gproptional tool now displays which database packages will be installed.
- Snapshot name enhancements so that a snapshots will be filtered by the database version (GP5 vs GP6).

Version 4.1

Fixes

- gpdelete maintains n snapshots determined in the gpcronsnap.conf file. The delete process would delete snapshots from other clusters to maintain the determined number of snapshots. Starting with 4.1, gpcronsnap will only delete snapshots that exceed the number and have been created by the currently running cluster.
- Changed disk entries in fstab file to use UUID which ensures proper mounting on reboots and snapshot restores.

Enhancements

- Created a new binary installer for the greenplum-cloud utilities. This makes it easier to upgrade these utilities.
- GP Browser renamed back to phpPgAdmin. It has enhancements for a Pivotal template and handling of partitioned tables. Installation is done via an RPM.
- Add caching of ssh keys between the segments back to the master.
- Added SSL to pgBouncer
- gprelease now updates cloud utilities
- Added new sysctl.conf entries for better memory management
- Added disk size parameter
- Changed drop down list of Subnet CIDRs to a text box to give more flexibility
- Reboots now reset the blockdev read ahead for data disks
- Enable Resource Groups by default instead of Resource Queues
- Automated maintenance with gpmaintain and gpcronmaintain
- Enabled console access by setting root password

Version 4.0.0

Fixes

- Disabled unnecessary `nf_contrack` which could cause queries to fail under heavy concurrent load.

Enhancements

- Upgraded Greenplum to 5.17.0 and Greenplum Command Center to 4.6.0.
- Added support for PXF and included OpenJDK.
- Enhanced Greenplum Command Center installer to better handle future upgrades.
- Added Subnet parameter.
- Improved `check_disk` and `check_network` scripts in `/opt/pivotal/greenplum/validation/performance`

Version 3.5

Enhancements

- Upgraded Greenplum to 5.16.0 and Greenplum Command Center to 4.5.1.
- Enhanced `gprelease/gpoptional` to migrate existing packages that are installed and upgrade packages if needed.
- Improved database initializing performance so deployments are completed faster.

Version 3.4

Fixes

- Resolved issue where `gprelease` failed if a package directory exists but no packages are installed.

Enhancements

- Upgraded Greenplum to 5.13.0
- Removed standby-master host and process now runs on first segment host
- Added GP Browser SQL utility
- Added `Standard_D14_v2` as an option for regions that don't support HPC instance types

Version 3.3

Enhancements

- Upgraded Greenplum to 5.12.0
- Upgraded Greenplum Command Center to 4.4.2
- Added `gpsnap` and `gpcronsnap` for executing database backups using Disk snapshots

Version 3.2

Enhancements

- Upgraded Greenplum to 5.10.2.
- Upgraded Greenplum Command Center to 4.3.1
- Added bouncer pause and resume functions
- Added Self-Healing
- Renamed gpupgrade and gpconupgrade to gprelease and gpconrelease
- gprelease and gpconrelease enhanced for better integration to optionally installed components
- gpooptional tool created to make it easier to install optional components and also upgrade existing components.

Version 3.1

Fixes

- Identified and disabled UDP port conflict in Azure

Enhancements

- Upgraded Greenplum to 5.9.0.
- Upgraded Greenplum Command Center to 4.2.0.
- Added Data Science Python and R packages.
- Increased the number of segment nodes to 32.
- Optimized and standardized offering to Standard_H16 and Standard_H8 instance types.

Version 3.0

Fixes

- Patched the operating system for the Meltdown and Spectre vulnerabilities.

Enhancements

- Rewrite of Template and Scripts to align scripts, tools, and user experience of Greenplum running on other clouds.
- Manage Greenplum Database upgrades with two new tools; gpupgrade and gpconupgrade
- Optional installs are now available as parameters to the ARM Template and visible in the Marketplace. Optional installs are still available post-installation.