

# Pivotal Greenplum Database Azure Marketplace v4.0 Release Notes

Updated: February 2019

## 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.

ARM template version 4.0 is based on Pivotal Greenplum Database version 5.17.0.

<b>Overview</b>	<b>1</b>
<b>Deploying Greenplum on Azure</b>	<b>4</b>
Parameters - Basics	4
Cluster Name	4
Admin User 'gpadmin' SSH Public Key	4
Allowed IP Address Range	4
Subnet	5
Subscription	5
Resource Group	5
Location	5
Parameters - Instances	5
Instance Type	5
Instance Count	6
Parameters - Optional	7
Install Command Center	7
Install Data Science Python	7
Install Data Science R	7
Install MADlib	7
Install PL/R	8
Install PostGIS	8
ARM Template	8

Connecting	10
GP Browser	11
SSH Access	11
Client Tool	12
Additional Resources	12
Deployment Logs	12
Validation	12
<b>Greenplum on Azure Additional Features</b>	<b>12</b>
Self Healing	12
Segment Healing	13
Standby-Master Healing	13
Master Healing	13
Snapshots	13
Steps Creating a Snapshot	14
Steps Restoring a Snapshot	14
Important Considerations and Features for Snapshots	14
gpsnap	14
gpcronsnap	15
Greenplum Upgrades	15
gprelease	15
gpcronrelease	15
pgBouncer	15
bouncer start	16
bouncer stop	16
bouncer pause	16
bouncer resume	16
Optional Installs	16
gpoptional	17
The PXF Extension Framework (PXF)	17
<b>Greenplum on Azure Technical Details and History</b>	<b>17</b>
Azure Resources	17
Image	17
Resource Group	18
Virtual Network	18
Subnet	18
Network Security Group	18
Storage Account	18
Public IP Address	18
Network Interface	18
Availability Set	18

Virtual Machines	18
Storage	19
Root and Swap	19
Data Storage	19
Data Storage	19
Master Storage	19
Diagram	19
Version History	20
Version 4.0.0	20
Fixes	20
Enhancements	20
Version 3.5	20
Enhancements	20
Version 3.4	20
Fixes	20
Enhancements	20
Version 3.3	21
Enhancements	21
Version 3.2	21
Enhancements	21
Version 3.1	21
Fixes	21
Enhancements	21
Version 3.0	21
Fixes	21
Enhancements	21

# Deploying Greenplum on Azure

## Parameters - Basics

The screenshot shows the 'Basics' configuration step of the 'Create Pivotal Greenplum (BYOL)' deployment wizard. The left sidebar contains five steps: 1. Basics (selected), 2. Instances, 3. Optional, 4. Summary, and 5. Buy. The main area displays the following fields:

- Cluster name (3-10 letters or numbers):** greenplum
- Admin User 'gpadmin' SSH Public Key:** (Empty text box)
- Allowed IP Address Range:** 0.0.0.0/0
- Subnet:** 172.21.0.0/24
- Subscription:** (Dropdown menu)
- Resource group:** Select existing... (Dropdown menu) with a 'Create new' link below it.
- Location:** East US (Dropdown menu)

### Cluster Name

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

### Admin User 'gpadmin' SSH Public Key

This is your public key used to ssh to the Master nodes. 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.

### Allowed IP Address Range

This is the IP Address range that is allowed to connect to your deployment. You can use "0.0.0.0/0" but that means that every address on the Internet will be able to ssh to the Master nodes in the cluster so long as they have the private key. Consider using a more restrictive mask to prevent unwanted attempts to connect to the deployment.

## Subnet

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.

## Subscription

Pick from one of your existing Azure Subscriptions.

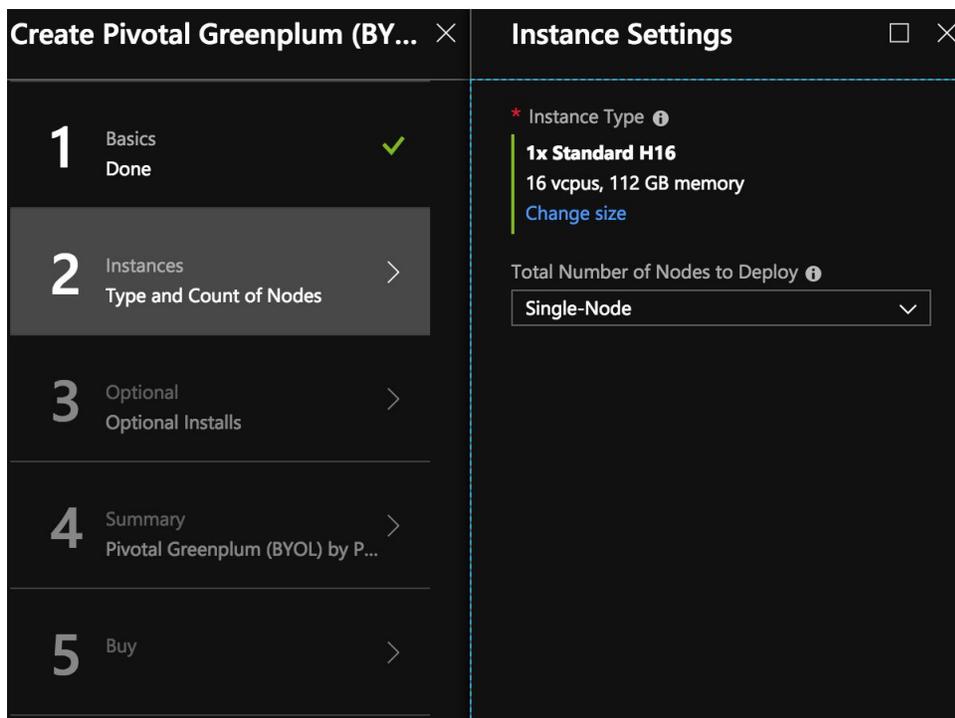
## Resource Group

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 - Instances



## Instance Type

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\_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 Count

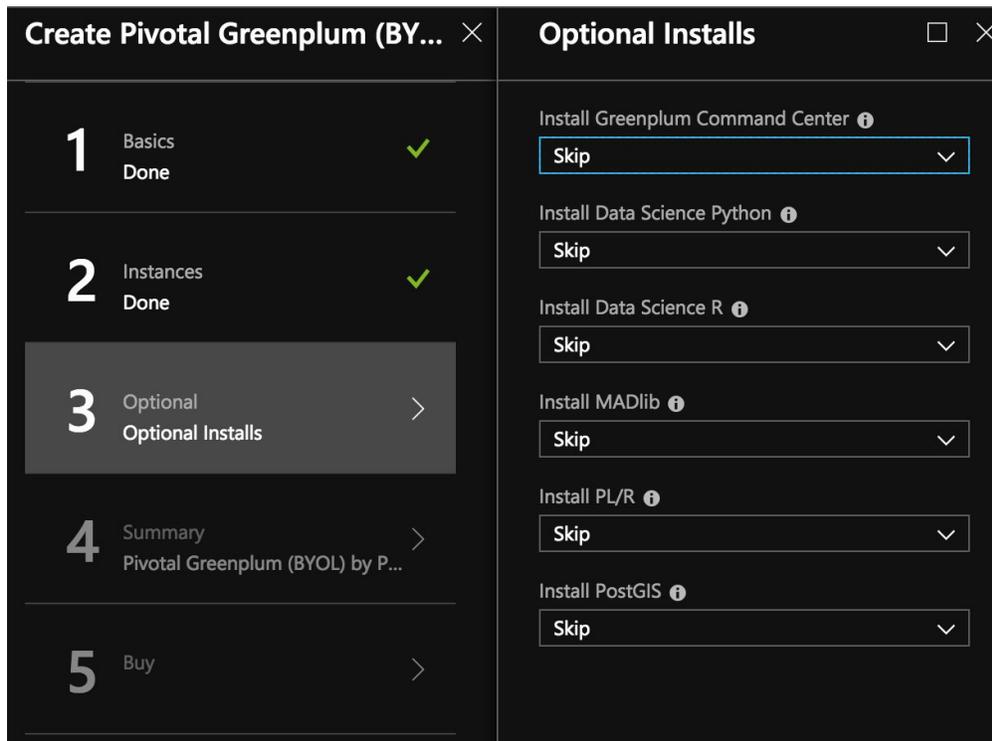
The number of Instances in the Deployment can be up to 32 segment nodes. When deploying with a Single Node, database mirroring and the Standby-Master will be disabled. The default is Single-Node.

Below is a table which provides the amount of raw data storage for each configuration.

<b>Nodes</b>	<b>Description</b>	<b>Standard_H16 &amp; Standard_D14_v2</b>	<b>Standard_H8</b>
Single-Node	Single Node	16 TB	8 TB
1-Master-2-Segments	1/16 Rack	34 TB	18 TB
1-Master-4-Segments	1/8 Rack	66 TB	34 TB
1-Master-8-Segments	1/4 Rack	130 TB	66 TB
1-Master-16-Segments	1/2 Rack	258 TB	130 TB
1-Master-24-Segments	3/4 Rack	386 TB	194 TB
1-Master-32-Segments	1 Rack	514 TB	258 TB

Storage size indicates how the disk size per Segment node. For multi-node deployments, the Master will have a single 2 TB drive.

## Parameters - Optional



### 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 `gpooptional` to install this package.

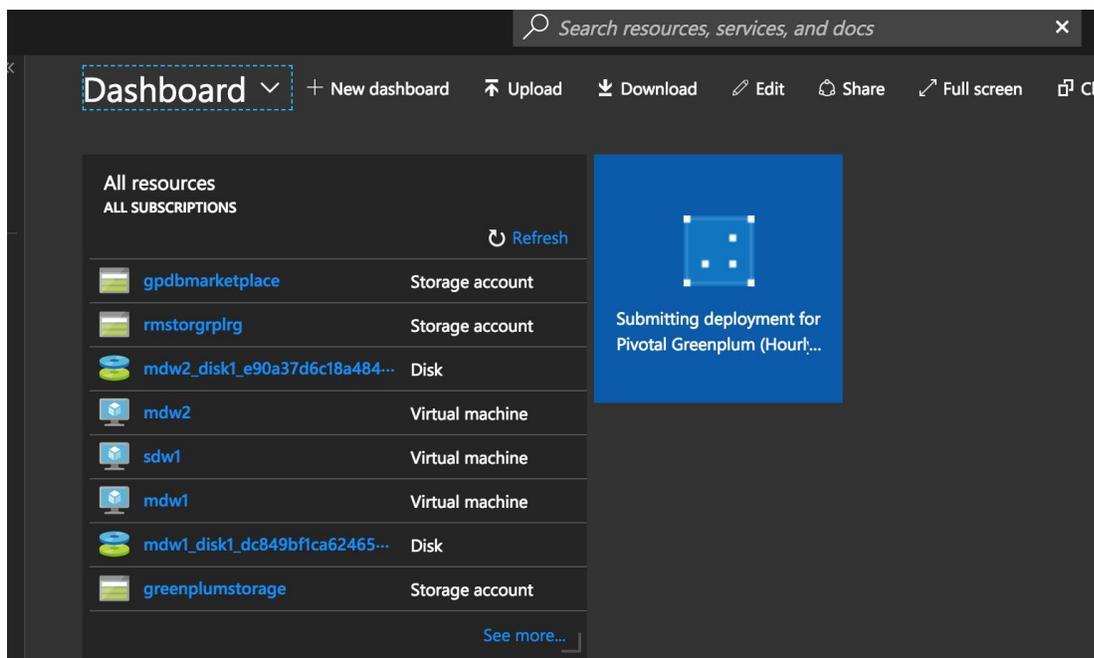
## Install PostGIS

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 `gpooptional` to install this package.

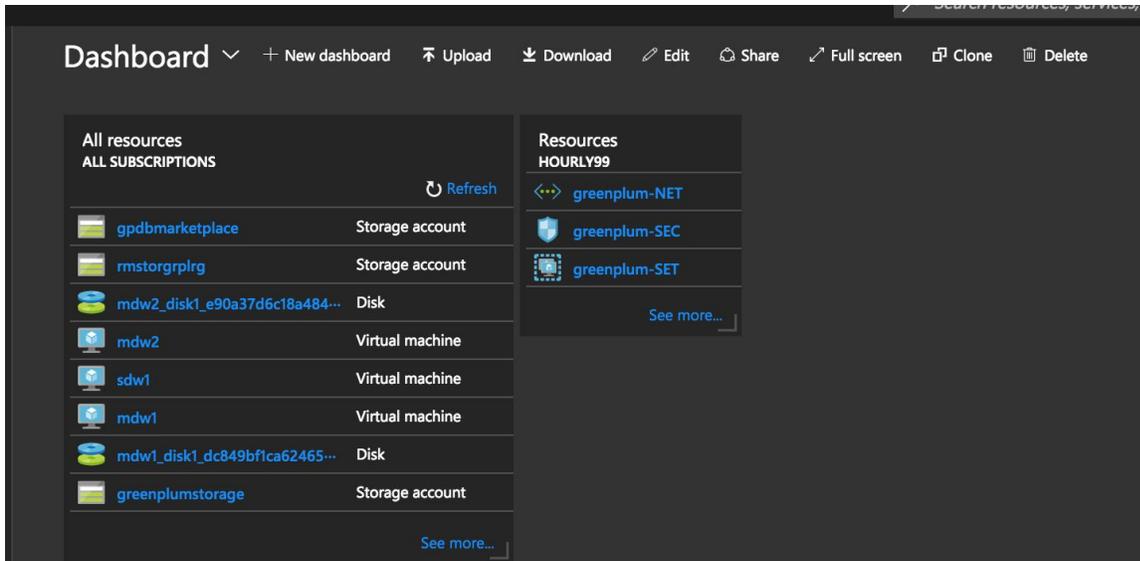
---

## ARM Template

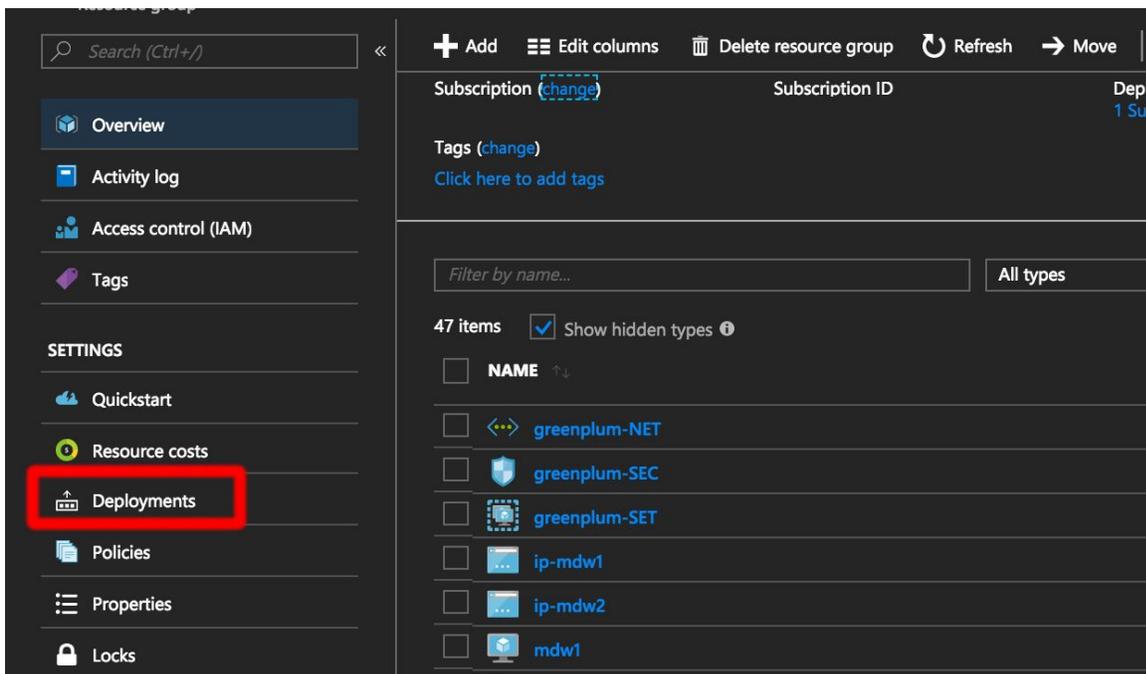
Deployment is very simple in the Azure Marketplace. Simply provide the parameters in the user interface and then submit the ARM template to create the deployment.



During the deployment, you will see the above image during the deployment.

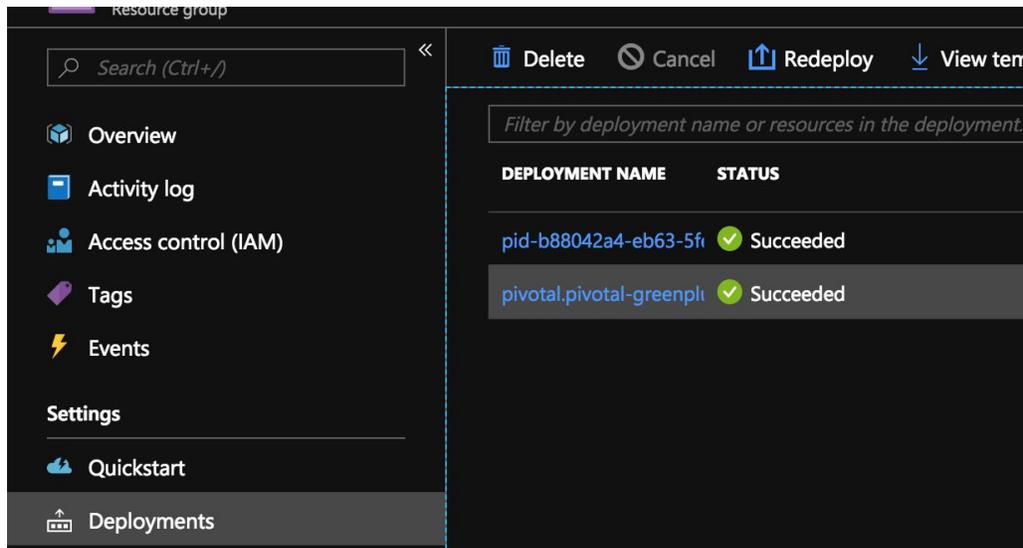


Once complete, the image will change to show a few resources and a "See more..." link. Click on "See more..."

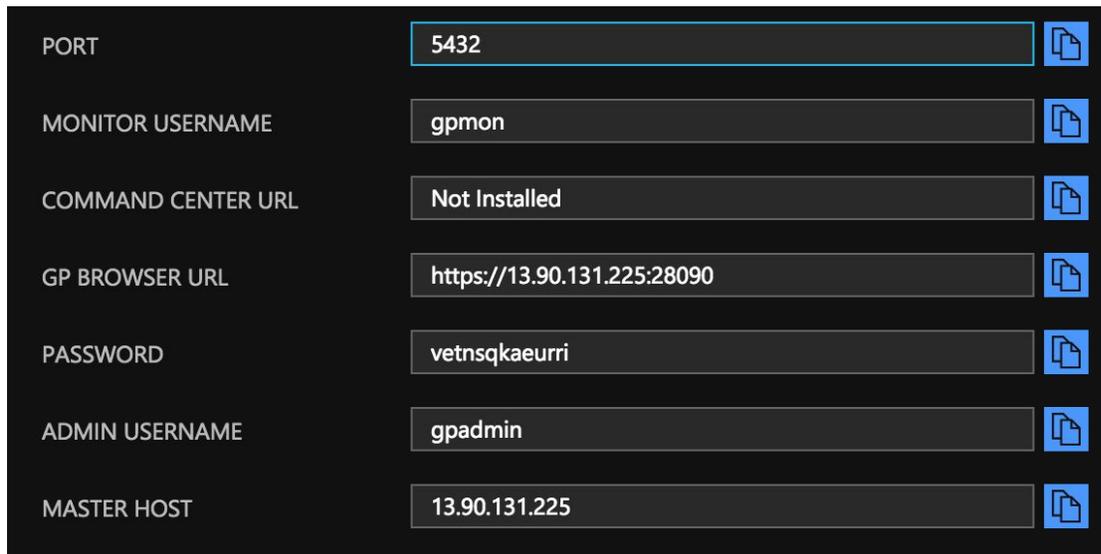


Click on "Deployments".

You will see two Deployments. Ignore the Deployment starting with "pid-".



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.

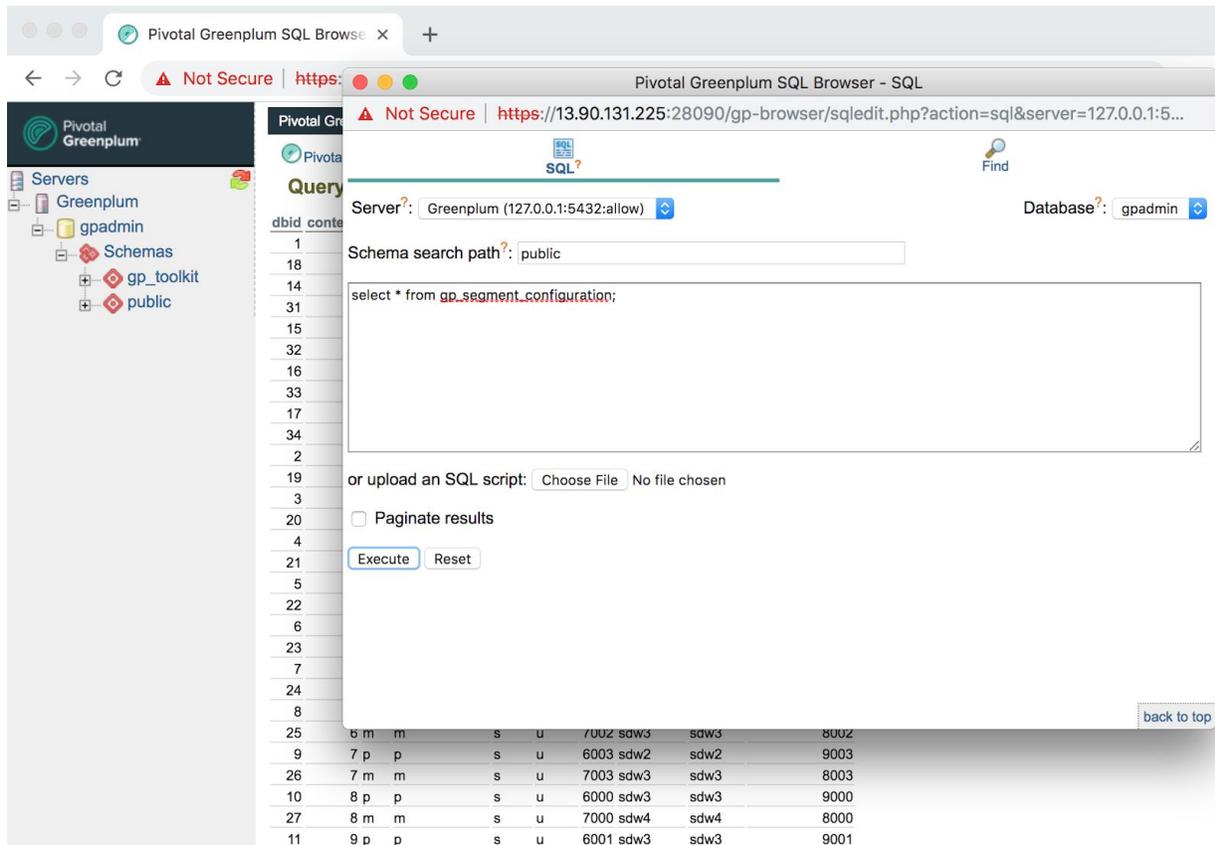
## Connecting

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.

## GP Browser

This web based SQL tool is based on phpPgAdmin and 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 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: 5.17.0
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: 187783
*****
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.

There are multiple 2TB Disks used in Greenplum on Azure Deployments which maximizes performance but also means there are more Disks per Segment host than there are Segments. Software RAID is used stripe the Disks so that there is one volume per Segment but consequently, this also means the volumes must be unmounted in order to take a consistent snapshot. All of this is automated with `gpsnap`.

## Steps Creating a Snapshot

- Stop pgBouncer
- Stop the database
- Umount volumes
- Create snapshots in parallel (one process per host)
- Mount volumes
- Start the database
- Start pgBouncer

## 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 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

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.

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.

## gpoptional

This utility simplifies installing optional components during the initial deployment and also after the deployment has been completed. Simply run "gpoptional" 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](https://gpdb.docs.pivotal.io/5170/pxf/overview_pxf.html)

## Greenplum on Azure Technical Details and History

---

### Azure Resources

#### Image

The Image is based off of CentOS 7.4 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.

## 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.

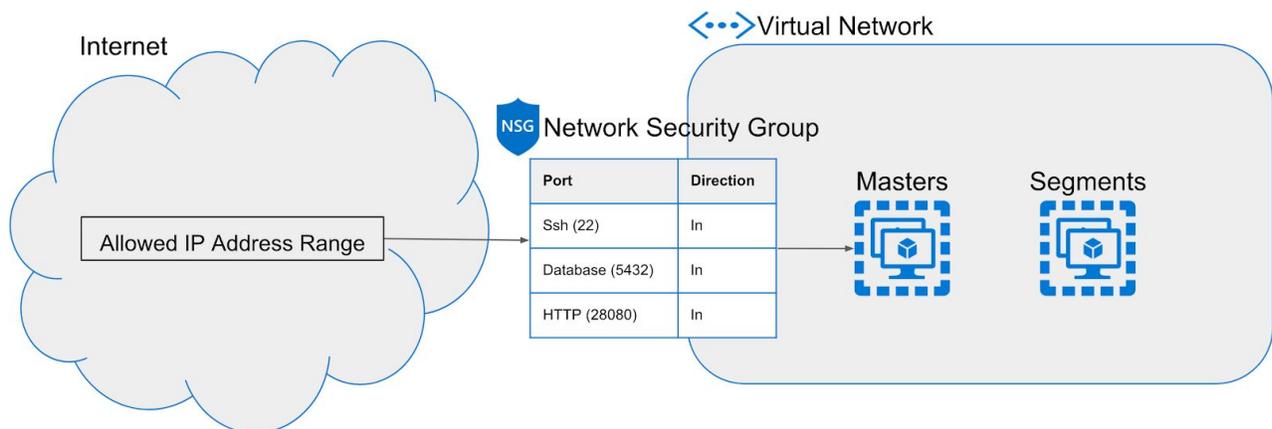
### Data Storage

The storage option has been configured with Standard\_LRS disks which are optimized for throughput instead of IOPs. Disk configuration has been optimized in terms of the number and size of the disks based on the 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 4.0.0

#### Fixes

- Disabled unnecessary nf\_conntrack 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/gproptional 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 gpconsnap 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
- gpoptional 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.