Updated: March 2022

ARM template version 6.5 is based on:

- VMware Tanzu Greenplum Database version 6.13.0

Or

- VMware Tanzu Greenplum Database version 5.28.4

Or

- VMware Tanzu Greenplum Database version 4.3.33

Overview

VMware Tanzu Greenplum on Azure uses Azure Resource Manager (ARM) template to automate the deployment of a cluster in the Azure cloud.

Cloud Native and Cloud Managed Tools and Features

Utility/Feature	Description
bouncer	Command line utility that simplifies using pgBouncer connection pooler.
gpcompute	Grow or shrink compute independently of storage.
gpgrow	Grow storage independently of compute.
gpmaintain / gpcronmaintain	Automates routine maintenance of VMware Tanzu Greenplum such as vacuum and analyzing tables.
gpoptional	Installation tool for installing optional components such as MADlib and Command Center.
gppower	Automates the Pausing and Resuming a cluster to save on infrastructure costs.
gprelease / gpcronrelease	Automates the upgrade of VMware Tanzu Greenplum, any optional components installed, and the cloud utilities.
gpsnap / gpcronsnap	Automates the execution of snapshot backups and restores using the cloud vendor's native snapshot utilities.
Self-Healing	Automatic instance replacement and recovery in a failure event.

Licensing

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 total the number of vCPUs deployed in the cluster and then purchase that many subscription cores from VMware Tanzu Software.

Calculating Usable Storage

To determine the storage size of the Deployment, multiply the number of segment instances in the cluster times the number of disks per instance, 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.

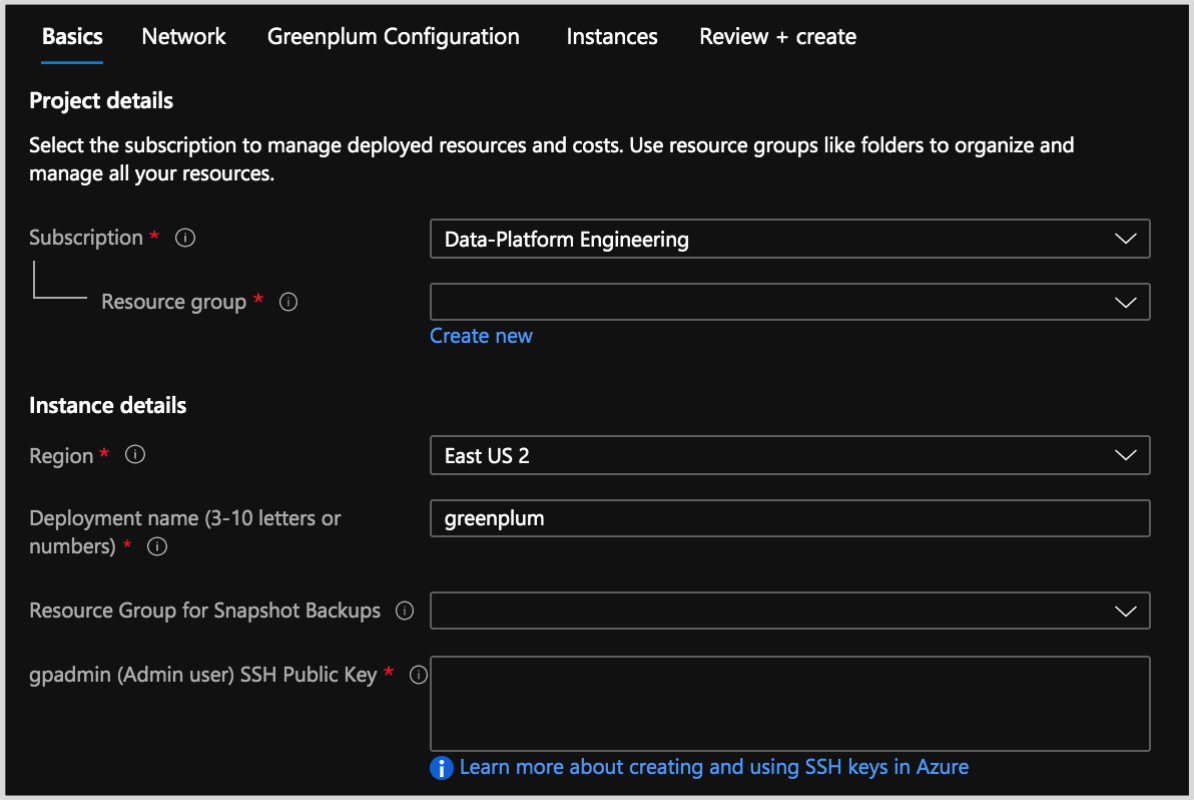
Instance Type	Instance Count	Disk Size
Standard_H8	64	2TB

```
( 64 Segment Instances * 2 Disks * 2TB Disks ) = 256 TB Raw Storage

( 256 / 2 ) * .7 = 89.6 TB Usable Storage
```

Deploying Greenplum on Azure

Parameters - Basics



Subscription

Pick from one of your existing Azure Subscriptions.

Resource Group

This is the Resource Group that will contain all of the Azure resources related to this Greenplum cluster. 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.

Region

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

Deployment Name

This identifies the VMware Tanzu Greenplum Deployment. The default is "greenplum".

Resource Group for Snapshot Backups

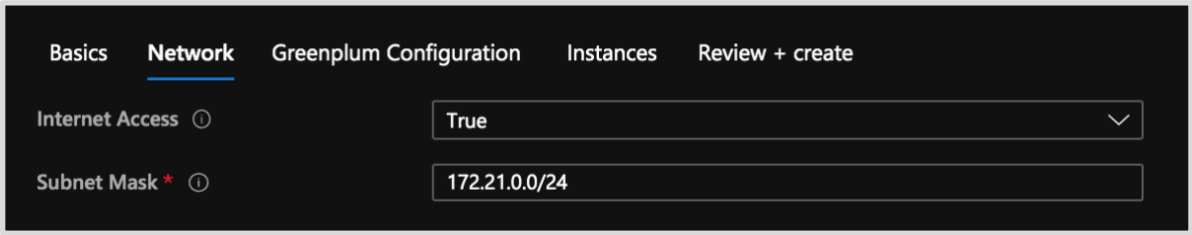
Snapshot backups are stored in this Resource Group. It is recommended that you pick a different Resource Group than the Greenplum cluster Resource Group. This will allow you to retain snapshot backups if you delete your Greenplum cluster.

This Resource Group also enables you to copy snapshots from one region to another. This is ideal for Disaster Recovery as you can use the copied snapshots in other regions to restore services in the event that Azure has a region failure.

gpadmin (Admin user) SSH Public Key

This is your public key used to ssh to the Master instance. Password authentication has been disabled on all instances 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.

Parameters - Network



Internet Access

True means a Public IP address will be created for the Master instance 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 - Greenplum Configuration

Database Version

Pick the major database version you wish to use. GP4, GP5 and GP6 are currently available with this release.

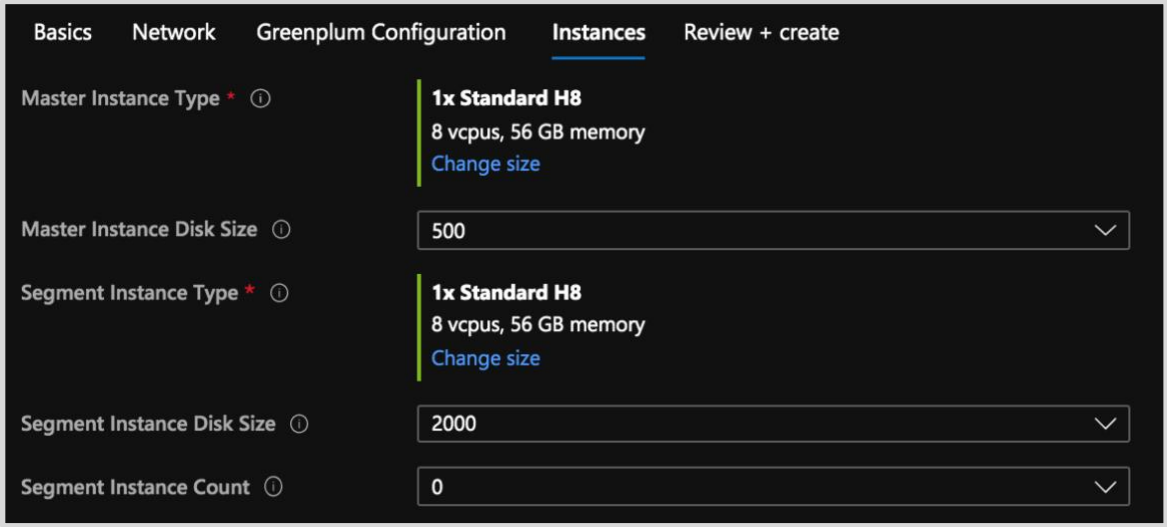
Database Name

This defines the default database name for your Greenplum cluster. Optional components will be installed in this database.

Timezone

Pick the time zone you wish to use for the virtual machines in your cluster.

Parameters - Instances



Master Instance Type

It is recommended to use the HPC instance types if available in your region. The Master node typically does not need a large number of vCPUs or RAM so picking the smallest instance type available is usually sufficient.

Master Instance Disk Size

Specify the size of the Data Disk(s) attached to the Master instance. In a Single-Instance deployment, the Master will have two disks while in a Multi-Instance deployment, there will only be one.

Segment Instance Type

It is recommended to use the HPC instance types if available in your region. Furthermore, pick the same "family" of instance type. If you decide to use the HPC instance type for your Master, also use HPC for the Segment Instances.

Use the Standard_H8 or Standard_D13_v2 instance types for moderate concurrency needs and use the Standard_H16 or Standard_D14_v2 instance type when the cluster has higher concurrency demands.

Instance Type	RAM	vCPUs	Disk Type	Data Disks	Segs Per Instance
Standard_H8	56	8	StandardSSD_LRS	3	3
Standard_H16	112	16	StandardSSD_LRS	3	3
Standard_D13_v2	56	8	Standard_LRS	3	3
Standard_D14_v2	112	16	Standard_LRS	3	3

Helpful Tip
You can leverage the ***gpcompute*** utility to change the instance type after your deployment has been completed.

Segment Instance Disk Size

Each Segment instance will have two data disks and this specifies how large each disk is. The sizes listed are based on the cost structure in Azure.

Segment Instance Count

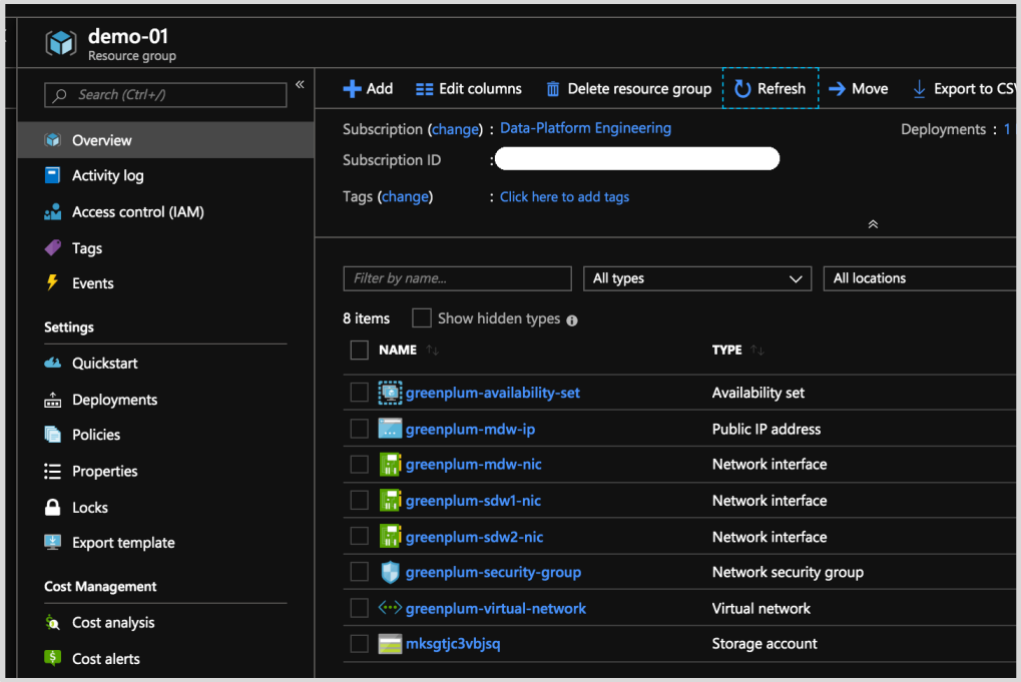
The number of Segment instances in the Deployment can range from 0 up to 64. When picking 0, the Master will act as a Single-Instance cluster with Master and Segment processes in one virtual machine.

Helpful Tips
If deploying a single-instance cluster, some cloud utilities will not work fully.
- gpcompute will not change the size of the Master instance
- gppower will not pause the Master instance
- gpgrow will not grow the Master instance

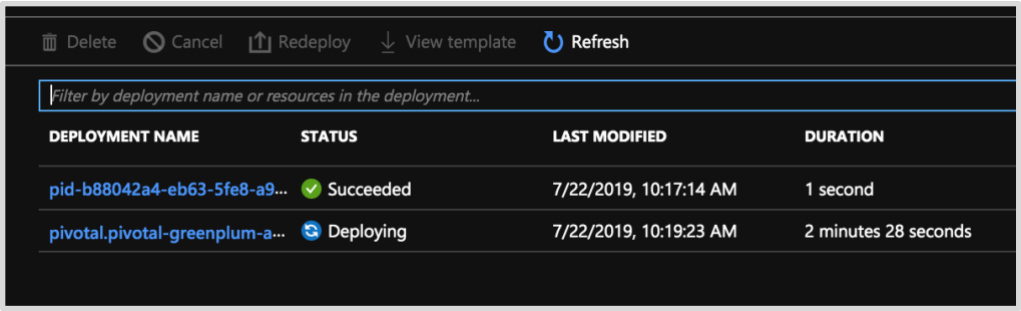
ARM Template

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.

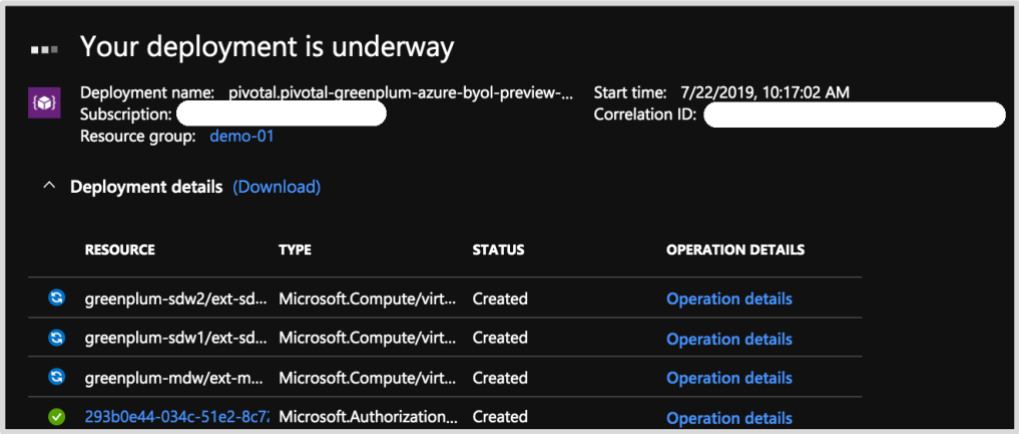
Once the deployment has been submitted, go to the new Resource Group that you created for this deployment.



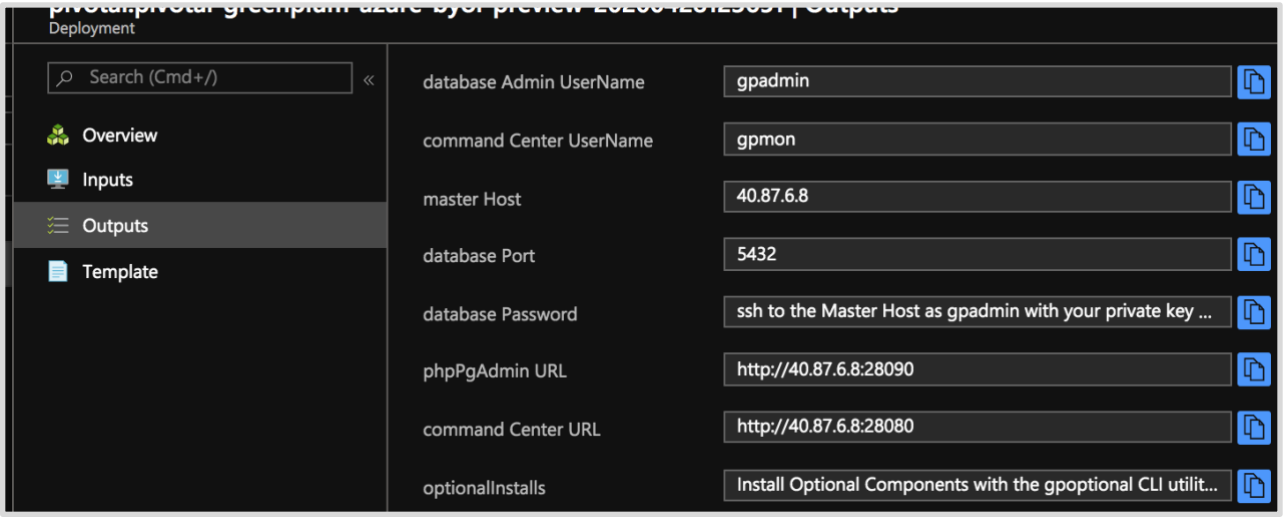
Next, click on Deployments in your Resource Group.



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 VMware Tanzu Greenplum on Azure cluster.

Security

The randomly generated password can be changed for gpmon and gpadmin. It is recommended to keep these passwords in sync too. This is done inside a database session as shown below.

```
ALTER USER gpadmin PASSWORD '<new_password>';
ALTER USER gpmon PASSWORD '<new_password>'
```

After updating the database passwords, you need to update configuration files.

1. /home/gpadmin/.pgpass
   The format for this file is as follows:

   ```
   localhost:5432:pgbouncer:gpadmin:<new password>
   mdw:6432:*:gpadmin:<new password>
   *:6432:gpperfmon:gpmon:<new password>
   ```

2. /opt/pivotal/greenplum/variables.sh

Find the GPADMIN_PASSWORD variable and update the value set as follows:

```
GPADMIN_PASSWORD="<new password>"
```

3. /data1/master/pgbouncer/userlist.txt
   Use the encrypted value for the password from pg_shadow for the gpadmin user.

   ```
   psql -t -A -c "SELECT passwd FROM pg_shadow \
   WHERE usename = 'gpadmin'"
   ```
   The encrypted value will begin with "md" and use this in the file with the following format:

   ```
   "gpadmin" "<encrypted password>"
   ```

The reason for the storing of the password in an /opt/pivotal/variables.sh is for a snapshot backup restore to work properly. If you take a backup of your cluster and restore it to another cluster, the password stored inside the database likely won't match the pgBouncer and the .pgpass configuration. Another scenario is, you take a snapshot of the database, update the database password, and then restore your snapshot which again causes a conflict of configuration files not matching the database.

During a snapshot restore, the gpsnap utility uses the password stored in `/opt/pivotal/greenplum/variables.sh` to set the database passwords for gpmon and gpadmin as well as update the pgBouncer and .pgpass configuration files. This ensures that a snapshot restore works properly.

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 Instance, Port, Admin UserName, and Password information used to connect to Greenplum.

The database password is no longer available in the Deployment output and must be retrieved from the Master host. Use your private key and ssh as gpadmin to the Master host and then "cat ~/.pgpass".

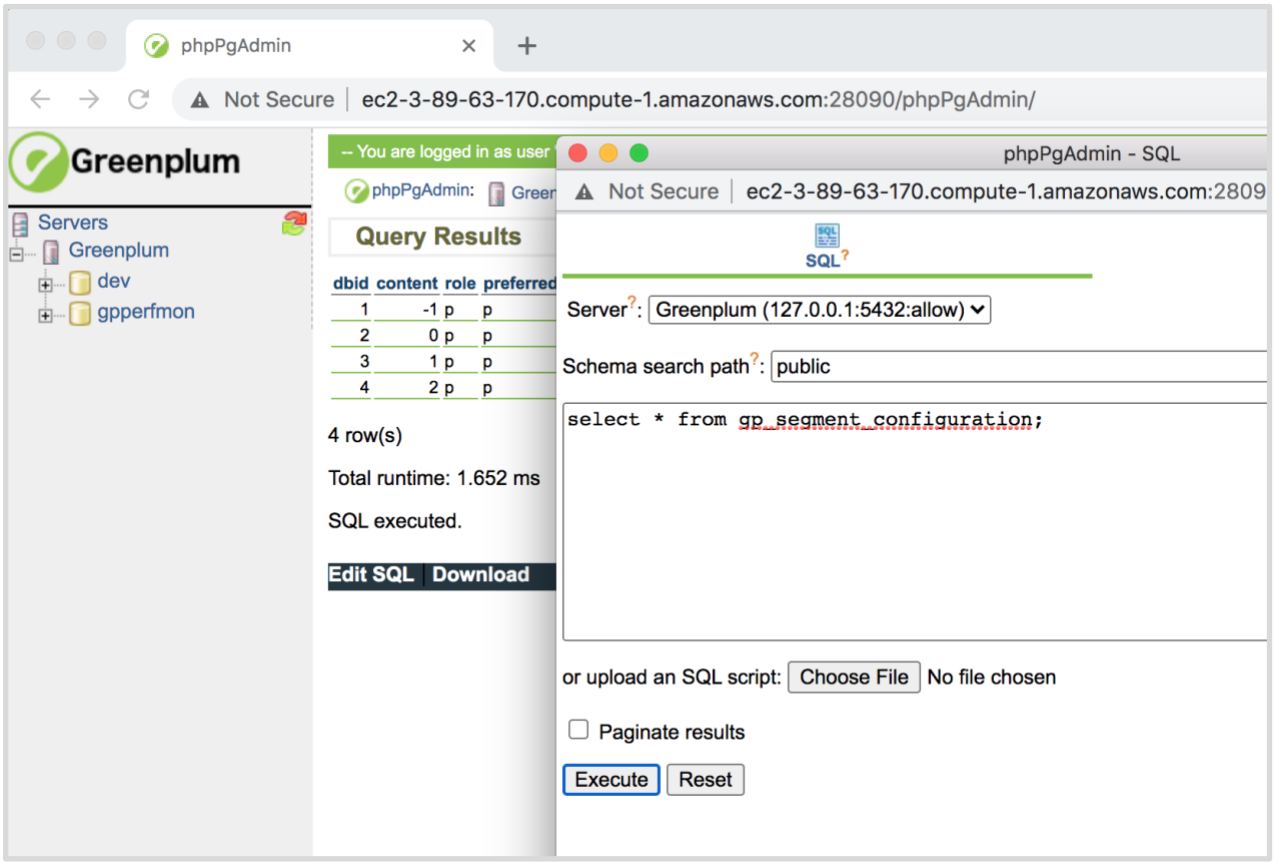
Example:

```
cat ~/.pgpass
localhost:5432:pgbouncer:gpadmin:vmcbigojqen2e
mdw:6432:*:gpadmin:vmcbigojqen2e
*:6432:gpperfmon:gpmon:vmcbigojqen2e
```

phpPgAdmin

This is a VMware Tanzu enhanced version of the popular phpPgAdmin web based SQL tool. It has been configured to start automatically on the Master Instance. The URL is provided in the

Deployment Output. Connection is simple by using the admin "gpadmin" user and the provided database password.



Helpful Tip
phpPgAdmin uses Apache HTTP server and does not include an SSL certificate. If you plan on using this tool across the Internet, it is highly recommended that you configure Apache with an SSL 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: 6.13.0
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, sdw33, sdw34, sdw35, sdw36, sdw37,
sdw38, sdw39, sdw4, sdw40, sdw41, sdw42, sdw43, sdw44, sdw45, sdw46, sdw47,
sdw48, sdw49, sdw5, sdw50, sdw51, sdw52, sdw53, sdw54, sdw55, sdw56, sdw57,
sdw58, sdw59, sdw6, sdw60, sdw61, sdw62, sdw63, sdw64, sdw7, sdw8, sdw9
Total Segments: 192
Computed storage in GB: 140825
****************************************************************************
Note: Be sure to switch to gpadmin:
sudo su - gpadmin

[gpadmin@mdw ~]$
```

Client Tool

Connecting with a remote client utility 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 VMware Tanzu 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 of the deployment can be found in: /opt/pivotal/greenplum/rollout.log. These logs are on every instance but the Master instance 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 VMware Tanzu Greenplum utility "gpcheckperf".

Greenplum on Azure Additional Features

bouncer

pgBouncer 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.

The bouncer utility automates the starting, stopping, pausing, and resuming of pgBouncer. It is recommended to use this utility to manage pgBouncer on your cluster.

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 an encrypted password. Create users and assign passwords in Greenplum as normal and pgBouncer will authenticate users with the database passwords you set.

Pooling has been configured for "transaction" with max client connections of 1000 and max database connections to 10. 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 Instance.

Note that for JDBC connections, you may need to add "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.
Actions available with bouncer include start, stop, pause, and resume.

```
bouncer <action>
```

Note: pgBouncer may not be compatible with some applications. You can disable pgBouncer by making these changes.

1. Stop pgBouncer with "bouncer stop"
2. Stop Greenplum Command Center with "gpcc stop"
3. Stop the database with "gpstop -a"
4. On the mdw, change $MASTER_DATA_DIRECTORY/postgresql.conf port=6432 to port=5432.
5. On mdw, change /home/gpadmin/.bashrc PGPORT environment variable to 5432.
6. On mdw, change all 6432 values to 5432 in /home/gpadmin/.pgpass
7. Source /home/gpadmin/.bashrc
8. Start the database with "gpstart -a"
9. Start Greenplum Command Center with "gpcc start"

gpcompute

The "gpcompute" utility enables you to add or reduce the compute capacity of your cluster independently of your storage. The command completely automates and integrates with Azure to change the instance type of the segment instances.

The status command will show you the current instance type of your segment instances and also show the instance type of the current launch template.

```
gpcompute status
```

The command will accept these two different instance types in Azure: Standard_H8 and Standard_H16.

```
gpcompute <instance_type>
```

The Standard_H8 instance type is ideal for single-user and moderate concurrency workloads while the Standard_H16 instance type provides nearly the same single-user performance but an increase in concurrency performance.

The "gpcompute" utility enables you to add or reduce the compute capacity of your cluster independently of your storage but is not yet available for Azure.

gpgrow

Increase the segment instances' disk size in your cluster completely online with this utility. Segment Instances have 2 data disks per each and the data size is specified in GB. Azure only allows increasing a disk size and this is not an online activity. The command will restart the segment instances automatically to complete the task.

Note: This utility does not support growing the Master Instance disk size.

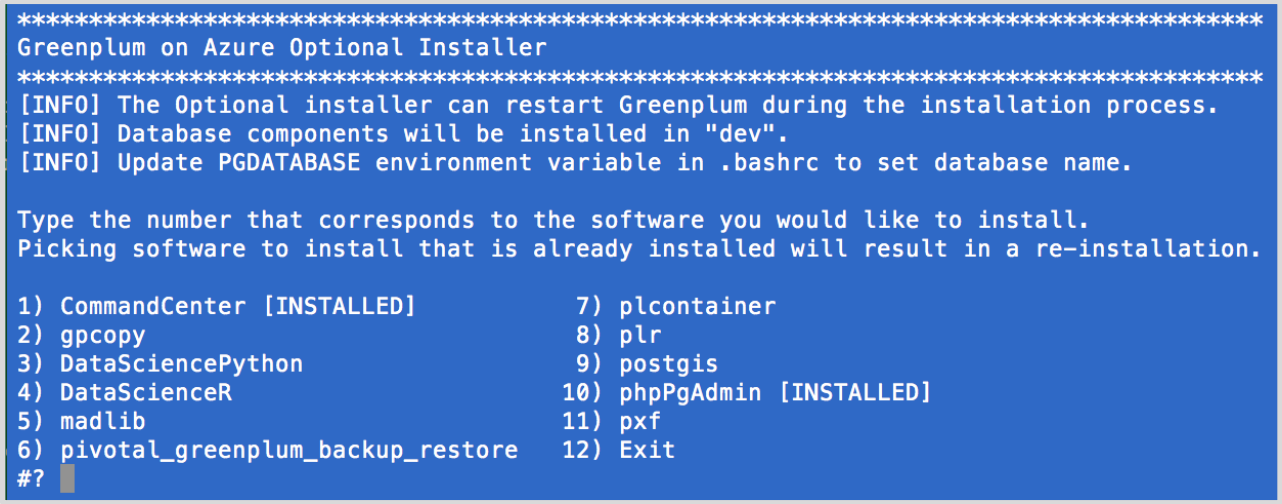
```
gpgrow <new disk size>
```

gpmaintain / gpcronmaintain

The gpmaintain utility automates basic maintenance of Greenplum. It analyzes all tables using analyzedb, vacuums tables that are in need, and maintains the catalog.

The gpcronmaintain utility is executed via cron and looks at the schedule file in /usr/local/greenplum-cloud/conf/gpcronmaintain.conf. By default, this runs at 4:00 AM every Saturday and can be configured to run more or less often.

gpoptional



This utility simplifies installing optional components 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.

phpPgAdmin and Command Center are installed with every deployment automatically while MADlib, Data Science Python, Data Science R, PL/R, PostGIS, PL/Container, Backup/Restore, PXF, and gpcopy can optionally be installed with gpoptional.

gppower

The gppower command makes it easy to pause (stop) a cluster to save on infrastructure costs. The command stops the database and all segment instances with gppower pause and resume a cluster with gppower resume.

Pausing and suspending with gppower:

```
gppower pause
```

```
gppower resume
```

This command will stop/start and deallocate/allocate the segment instances, stop/start the database, stop/start PXF (if it was running when paused), stop/start Command Center (if it was running when paused), and pause/resume the Segment Auto-Scaling-Group.

gprelease / gpcronrelease

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

This gpcronrelease utility is the scheduling component of gprelease. It checks to see if a new release is available and updates the message of the day to indicate a new version is available. By default, this runs in cron weekly on Sunday at 1:00 AM in the local timezone.

gpsnap / gpcronsnap

Note: Greenplum 4 (GP4) does not support gpsnap and gpcronsnap.

The gpsnap utility leverages Azure disk snapshots to quickly create a database backup. In order to get a consistent view of the database, the command first stops the database and then executes Azure commands in parallel to create a snapshot for each disk volume. After the disk snapshots are created, the database is restarted. This usually only requires a database outage of 30 seconds to 2 minutes.

Each snapshot is tagged so that when a restore is desired, the disk volumes get attached and mounted to the correct instances. This is done automatically with gpsnap.

You can restore a snapshot from one cluster to a different one so long as the snapshot is in the same Region and the cluster has the same configuration (number of nodes and disks). Valid snapshots can be found using "gpsnap list".

You can copy snapshots from one Region to another which is very useful for a Disaster Recovery scenario. Simply copy your snapshots with "gpsnap copy" to a different Region and in case of a disaster, you can create a new cluster in the region where you copied the snapshots with the same configuration and then recover the snapshots using gpsnap.

This utility manages creating, listing, deleting, and restoring snapshots. Please note that creating or restoring a snapshot will restart the database.

The actions available with gpsnap are list, create, delete, and restore. Delete requires an additional parameter of the snapshot_id you wish to delete, and restore requires the snapshot_id and the disk type you wish to use.

```
gpsnap list
gpsnap create
gpsnap delete <snapshot_id>
gpsnap restore <snapshot_id>
gpsnap copy <snapshot_id> <region>
```

Helpful Tip
You can leverage the ***gpsnap*** utility to change the disk type after your deployment has been completed. This is done by creating a snapshot with gpsnap and then restoring that snapshot but using a different disk type.

`gpsnap restore` sets the gpadmin and gpmon passwords to the value found in `/opt/pivotal/greenplum/variables.sh`. A snapshot could be from another cluster which has a different password which would cause problems unless the passwords are set. The .pgpass and pgBouncer configuration files are updated with the correct password too.

The gpcronsnap utility manages the automatic execution of gpsnap. By default, there will be a cron job that runs every 10 minutes and uses the configuration file: /usr/local/greenplum-cloud/conf/gpcronsnap.conf to determine if a snapshot is needed or not.

gpcronsnap.conf

```
# This utility will only execute gpsnap once on the day(s) of the week you
specify below.

#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
#do not schedule a time that conflicts with gpcronmaintain
snapshot_time=03:00
```

As shown above, the default schedule is a weekly snapshot on Sunday at 3:00 AM in the local timezone. Four snapshots will be retained before the oldest snapshot will be automatically deleted.

Self Healing

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

VMware Tanzu 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 each instance can be done with this operating system command:

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

Segment Healing

When a segment fails, the instance on restart will automatically detect that the database is up but needing to recover the one instance. 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 instance 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 instance executes a few Greenplum utilities. The process will restart the Master processes and rejoin the cluster.

The PXF Extension Framework (PXF)

Note: Greenplum 4 (GP4) does not support PXF.

Starting with template version 6.3, PXF is installed via gpoptional.

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.

After installing PXF with gpoptional, you can use the following commands on the Master (mdw) to manage the cluster.

Start the cluster:

```
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.

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.

First stop the database, pgBouncer, PXF (if started), and Command Center (if started).

```
gpstop -a
bouncer stop
pxf cluster stop
gpcc stop
```

Next, use "gpssh -f all_hosts.txt" to connect to all instances via ssh. The all_hosts.txt file is in the gpadmin home directory on the Master Instance. Execute the command to run the yum update.

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

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

Once complete, you can restart the virtual machines.

Now you can start the database, pgBouncer, PXF (if desired), and Command Center (if desired).

```
gpstart -a
bouncer start
pxf cluster start
gpcc start
```

Core Dumps

Core dumps are enabled and it makes it easier for Support to debug issues with the database.

Each virtual machine instance has a /var/coredumps directory owned by the gpadmin user. These dumps are retained for up to 7 days and then will get automatically purged by the cronjob script /opt/pivotal/greenplum/dump_maintenance.sh.

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 VMware Tanzu 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 instances 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 instances deployed in the Subnet.

Network Security Group

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

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 instances.

Network Interface

All instances in the cluster have a Network Interface, or NIC attached to it. The Master instance as a NIC with Public IP Address while the Segment instances 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 an instance.

Virtual Machines

These are the instances deployed in the cluster. It has Disks attached to each instance and specified in the ARM template. It also includes the deployment execution of the VMware Tanzu scripts executed on each instance.

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.

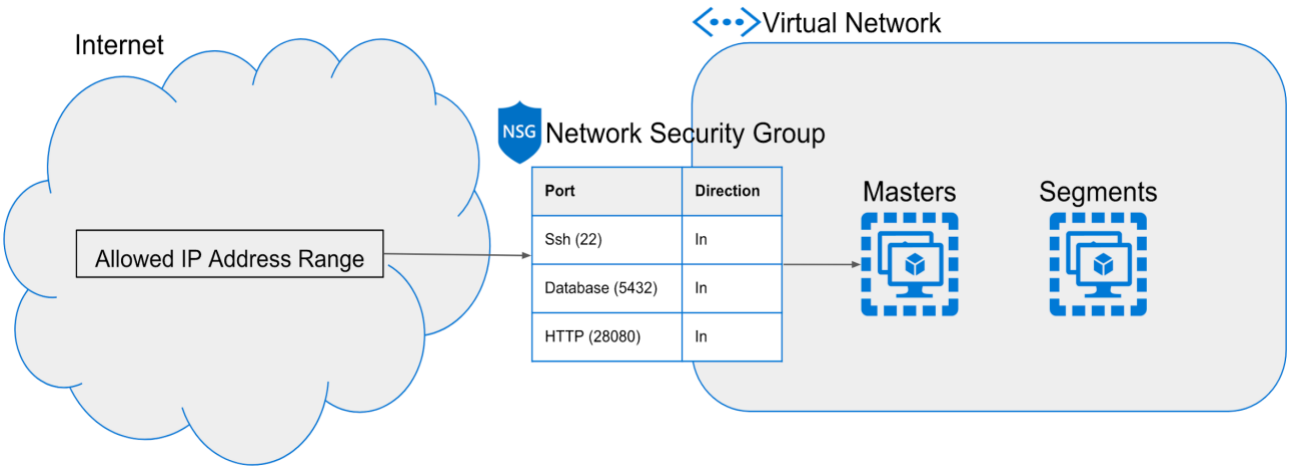
Data Storage

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 needed on the Master is much less than on the Segment instances so there will be only 1 "data" mount on the Master while the Segment instances will have 4.

Diagram



Version History

GPDB version update

- GP5 to 5.29.4 support using gprelease
    - VMware Tanzu Greenplum Database version 5.29.4

- GP6 to 6.19.2 support using gprelease
    - VMware Tanzu Greenplum Database version 6.19.2

Fixes

- A Fixed gpcronsnap for deleting old snapshots.

Enhancements

- Upgraded GP6 to version 6.19.2 and GP5 to version 5.29.4.
- Upgraded PXF version to 6.2.3 for GP6 and GP5.
- Upgraded pivotal_greenplum_backup_restore version to 1.24.0 for GP6 and GP5.
- Upgraded plcontainer version to 2.1.5 for GP6.
- Upgraded Command Center versions to 6.6.0 and 4.14.0 for GP6 and GP5 respectively.
- Upgraded GPCopy version to 2.3.1 for GP6 and GP5.
- Upgraded PostGIS version to postgis-2.5.4_pivotal.5.build.1 and postgis-2.1.5_pivotal.3.build.3 for GP6 and GP5 respectively.
- Upgraded plr version to plr-3.0.4 for GP6
- Upgraded madlib version to madlib-1.18.0 for GP6 and GP5
- Upgraded DataScience to version DataSciencePython-2.0.4 for GP6

Version 6.5

Fixes

- A gpsnap restore could fail if the restored database cluster does not have a database specified by the environment variable PGDATABASE. During a restore, gpsnap updates the .bashrc file if the PGDATABASE specified doesn't exist in the cluster. It first tries to use "gpadmin" and if that database also doesn't exist, it defaults to template1 which is always available.
- During an upgrade with gprelease, the optional components may fail to upgrade properly if a package was installed in the database in which gpoptional didn't have the package files for. These missing packages are now skipped by gpoptional during an upgrade.
- Command Center upgrades no longer drop the gpmon user. This could cause problems if the gpmon user owned objects outside of the gpperfmon database.

Enhancements

- Upgrade PHP to 7.4
- Upgraded GP6 to 6.13.0 and GP5 to 5.28.4.
- Upgraded phpPgAdmin which uses a new Greenplum theme.
- Upgraded gpview utility.
- The gpoptional utility will now download any missing packages during an upgrade process to ensure the correct packages are installed with the matching database version.

Version 6.4

Enhancements

- Upgraded GP6 to 6.12.0 and GP5 to 5.28.2.

- Changed Greenplum Interconnect type to "Proxy" for Greenplum version 6. This is specifically designed to work in the Azure environment which minimizes the number of Azure "flows" which increases performance.
- Changed the number of segments per host from 2 to 3 for Greenplum version 6. This was possible because of the Interconnect performance increase.
- Added additional Resource Group parameter used to store snapshot backups.
- Snapshot backups can now be retained after deleting a cluster and can be used for other clusters with the same configuration.
- Snapshot backups can now be copied to other regions to be used for Disaster Recovery.
- Removed Disk Type parameter as tests have indicated that Standard disk performs best with the D series instance types and the SSD disk works best with the HPC series.
- Added Standard_D14_v2 instance type again. This was possible after optimizing the disk configuration and using the Proxy Interconnect.
- Simplified the self-healing steps for Master recovery.

Version 6.3

Enhancements

- Upgraded GP6 to 6.10.1 and GP5 to 5.28.0.
- Added Greenplum version GP4 (version 4.3.33) as a parameter.
- Added xfs_defrag to gpmaintain.
- Changed memory management by using STATEMENT_MEM instead of spill_ratio in Resource Groups.
- Patched PHP for phpPgAdmin to the latest stable version.
- Added TimeZone parameter.
- Many improvements for gprelease to handle various edge cases.
- PXF is now installed via gpoptional.

Version 6.2

Fixes

- Fixed condition where gpmaintain might fail due to a catalog lock.

Enhancements

- Updated documentation to reflect VMware Tanzu instead of Pivotal Software.
- Upgraded VMware Tanzu Greenplum GP6 to version 6.8.1 and GP5 to 5.27.1.
- Changed default database version to GP6.
- Upgraded Command Center versions to 6.2 and 4.10 for GP6 and GP5 respectively.
- Upgraded PostGIS version to 2.5.4 build 3 for GP6.
- Upgraded PL/Container to version 2.1.2 for GP6.
- Upgraded MADlib to version 1.17 for GP5 and GP6.
- Upgraded Cloud gpoptional to now install Backup/Restore utility.
- Added operating system tools for debugging any problems that might happen (gdb, strace, pstack, sysstat, lsof, dstat, iotop and atop).

- Updated operating system to the latest version.

Version 6.1

Fixes

- Installation of PL/Container during the initial deployment of a cluster could cause a timeout because of the time it takes to install all of the docker images. To prevent this, PL/Container can only be installed with the command line utility gpoptional rather than as a parameter in the template.
- Some web browsers prevent the use of self-signed SSL certificates so phpPgAdmin has switched to http rather than https. It is recommended that you install your own certificate and use SSL post-installation.
- The bouncer utility was displaying the wrong program name in the INFO messages. This issue has been resolved.
- The wrong database version would be displayed in the message of the day after a database upgrade with gprelease. This issue has been resolved.

Enhancements

- Upgraded Pivotal Greenplum GP6 to version 6.6.0 and GP5 to 5.26.
- Updated kernel to latest version available.
- Command Center is now installed by default.
- Azure now supports the gpcompute utility with the HPC (Standard_H8 and Standard_H16) instance types.

Version 6.0

Enhancements

- Upgraded Pivotal Greenplum GP6 to version 6.4.0 and GP5 to 5.25.
- Added Database Name parameter so customers can specify the default database name.
- Added core file generation to make it easier to debug issues with Support.

Version 5.1

Fixes

- Self-healing now restarts Command Center after a segment instance failure.

Enhancements

- Added PL/Container optional install.
- Added ability to specify the disk type when restoring a snapshot.
- Simplified the disk formatting process during deployment.
- Ran a yum update to update for new security patches.
- Upgraded Pivotal Greenplum GP6 to version 6.3.0 and GP5 to 5.24.
- Changed the count of segment instances from a multiple of 4 to 2.

Version 5.0

Fixes

- Updated logic for scheduled maintenance to ensure it doesn't execute while it is already running.
- Changed when the gpmon password is set during Command Center installation so that it prevents fatal login messages in the GUI.
- Updated pg_hba.conf file settings automatically during deployment and recovery so that it always has hostnames and not ip addresses. This makes snapshot restoration much more reliable.
- "set-hostname" was executed twice during a deployment for each instance.
- Added "create extension" when initializing PXF.
- Swap file would be deleted on rebooting machines.

Enhancements

- Added DatabaseVersion parameter to indicate either GP5 or GP6 major version to install.
- Added Pivotal Greenplum version 6.1.0 and upgraded GP5 to 5.23.
- Added ability to grow storage independently from compute with gpgrow.
- Added ability to pause / resume a cluster with gppower.
- Removed Standard_H16 and Standard_D14_v2 instance types and doubled the number of instances supported.
- Ran yum update to update to CentOS 7.7.
- Removed analyzedb when Master Instance is recovered as this is no longer recommended.
- Disabled gpperfmon for Command Center as starting with version 4.8, it is recommended to not use gpperfmon_install.
- Updated operating system limits settings.
- Added SSD disk option for data disks.

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 gpoptional 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

- gpcrondelete 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_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/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 gpcronupgrade to gprelease and gpcronrelease
- gprelease and gpcronrelease 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 gpcronupgrade
- Optional installs are now available as parameters to the ARM Template and visible in the Marketplace. Optional installs are still available post-installation.