

Pivotal Postgres on Azure Marketplace v10.7 Release Notes

Updated: March 2019

Overview

Pivotal Postgres is deployed on Azure using a Virtual Machine Image. The Image uses CentOS 7 and includes MADlib, PostGIS, phpPgAdmin, pgBouncer, and automated backups.

This offering is ideal for quick and easy evaluations via the End User License Agreement that includes a 90 day free evaluation period. Support is provided to licensed customers just as it would if Postgres was installed independently of this offering.

Pivotal Postgres on Azure Marketplace Version 10.7 includes:

- Postgresql v10.7
- pgBouncer v1.9.0
- MADlib v1.15.1
- PostGIS v2.5.1
- phpPgAdmin v5.1.0

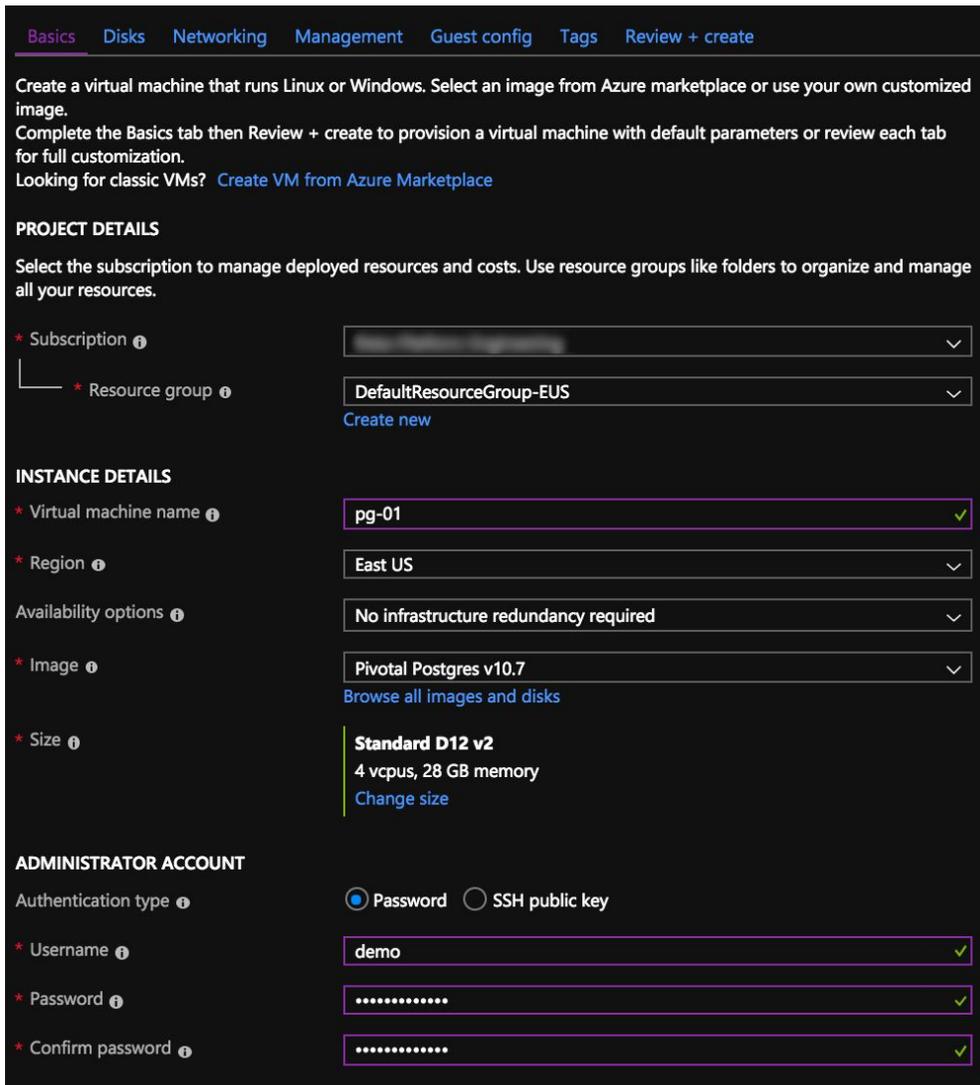
Overview	1
Deploying Pivotal Postgres	3
Basics	3
Disks	4
Networking	4
Management	5
Guest Config	5
Tags	5
Review and Create	5
Connecting	6
Connect with ssh	6
Message of the Day	6
PhpPgAdmin	6
Client Tool	7
Pivotal Postgres Additional Features	7
Backups	7
MADlib	8
PostGIS	8
pgBouncer	8
bouncer start	9

bouncer stop	9
bouncer pause	9
bouncer resume	9
Additional Resources	9
Installation Logs	9
Operating System Patches	9
Version History	9
Version 10.7	9

Deploying Pivotal Postgres

Microsoft provides an easy to use web console which can deploy Pivotal Postgres for you. First, find the Image in the Marketplace and then click Create.

Basics

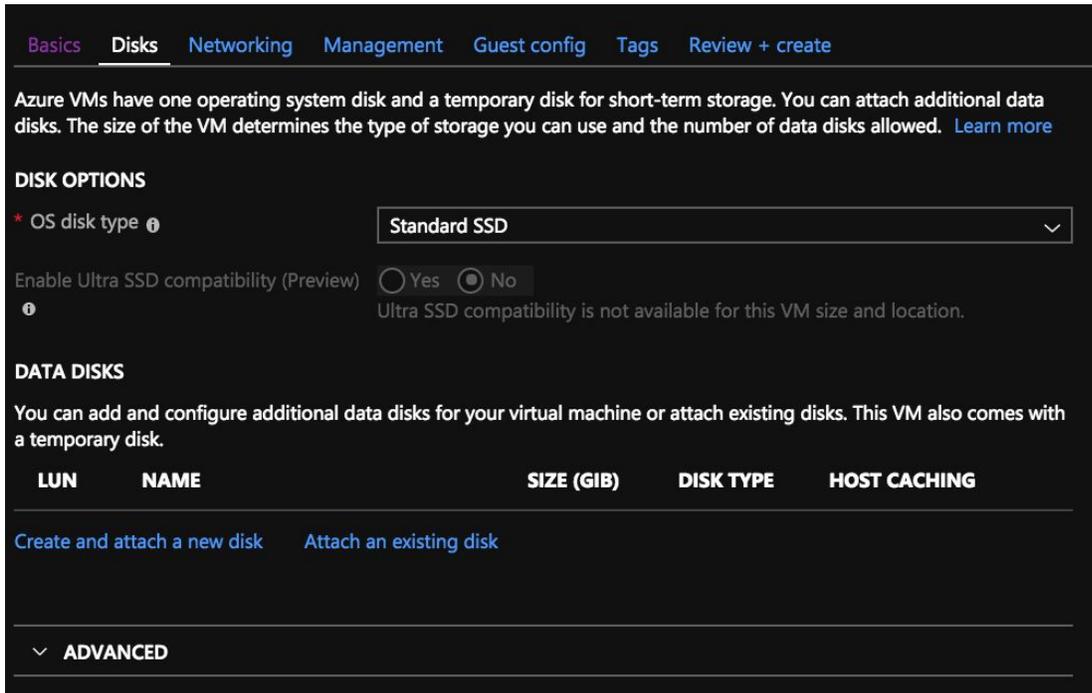


The screenshot shows the 'Basics' tab of the Azure portal's virtual machine creation wizard. The interface is dark-themed with white text. At the top, there are navigation tabs: Basics (selected), Disks, Networking, Management, Guest config, Tags, and Review + create. Below the tabs, there is introductory text: 'Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Looking for classic VMs? [Create VM from Azure Marketplace](#)'. The form is divided into three sections: PROJECT DETAILS, INSTANCE DETAILS, and ADMINISTRATOR ACCOUNT. PROJECT DETAILS includes fields for Subscription (a dropdown menu), Resource group (set to 'DefaultResourceGroup-EUS' with a 'Create new' link below it), and a 'Create new' link. INSTANCE DETAILS includes fields for Virtual machine name (set to 'pg-01'), Region (set to 'East US'), Availability options (set to 'No infrastructure redundancy required'), Image (set to 'Pivotal Postgres v10.7' with a 'Browse all images and disks' link below it), and Size (set to 'Standard D12 v2' with '4 vcpus, 28 GB memory' and a 'Change size' link below it). ADMINISTRATOR ACCOUNT includes 'Authentication type' (radio buttons for 'Password' and 'SSH public key', with 'Password' selected), 'Username' (set to 'demo'), 'Password' (masked with dots), and 'Confirm password' (masked with dots). Each field has a small information icon to its left and a green checkmark to its right.

Provide Subscription, Resource group, Virtual machine name, Region, Availability options, Image (Pivotal Postgres), Size, and Credentials.

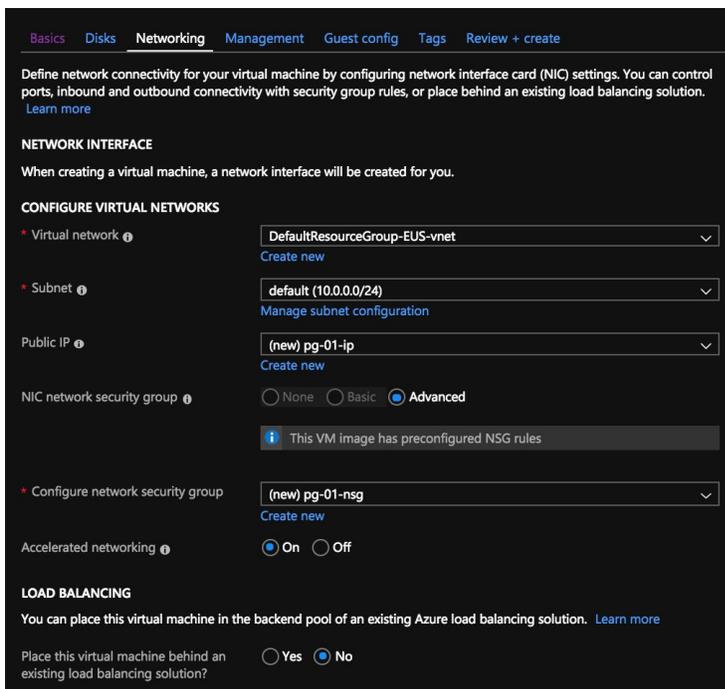
Note: It is recommended to use SSH public key authentication. You can also use the "pivotal" username which is the default database administrator name.

Disks



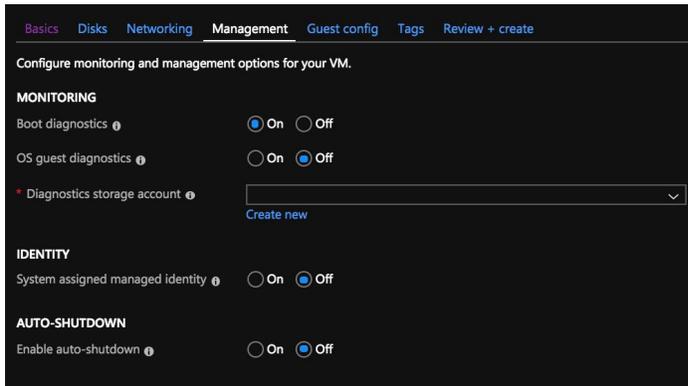
Add additional disks as needed. Note that additional disks will need to be formatted before use and Pivotal Postgres configured to use the additional storage.

Networking



Configure the Network as needed for your Pivotal Postgres deployment. The default NSG rules allows ssh access and access to port 28090 for phpPgAdmin.

Management

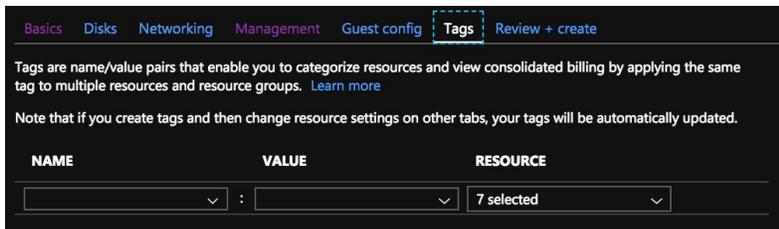


It is generally a good idea to enable Boot diagnostics which requires creating a Diagnostics storage account.

Guest Config

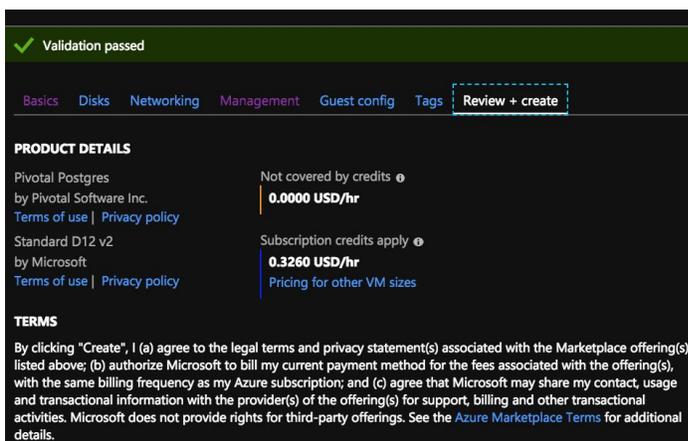
This is not supported by this VM so continue to Tags.

Tags



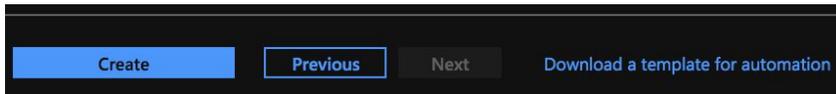
Add Tags to help identify your new VM.

Review and Create



A summary will be provided and Azure will validate the options you have chosen.

Next, simply click Create to deploy your VM. You can also Download the template to make it easier to automate future deployments.



Connecting

Connecting can be done with ssh, the web based phpPgAdmin database client, or with an external database client tool like pgAdmin 4.

Connect with ssh

After the VM has launched, use your credentials (username and either password or SSH key you defined) to connect to the new instance. Use the Public ip address to connect.

```
ssh <username>@<ip address>
```

```
escmac:postgres-azure jon$ ssh demo@52.179.0.68
Password:
*****
Welcome to your Pivotal Postgres on Azure!
*****
PostgreSQL 10.7; MADlib: 1.15.1; PostGIS: 2.5.1
pgBouncer Port: 5432; Database Port: 6432
Admin User: "pivotal"; Password in: /opt/pivotal/postgres/variables.sh
phpPgAdmin Public URL: https://52.179.0.68:28090
phpPgAdmin Local URL: https://10.0.0.4:28090
*****
Note: Be sure to switch to "pivotal"
sudo su - pivotal
[demo@pg-01 ~]$
```

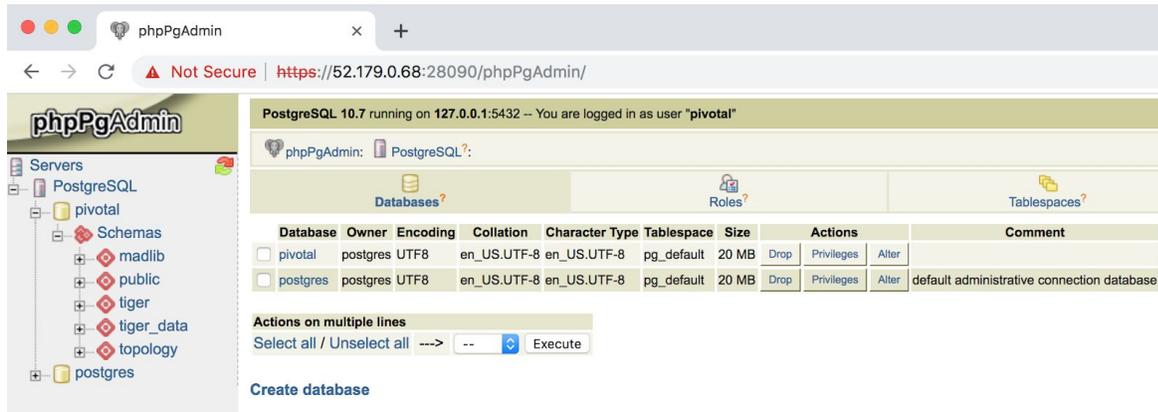
Message of the Day

When connecting with ssh, Unix will output the Message of the Day as shown above. This provides information about Pivotal Postgres such as versions and URLs for phpPgAdmin. If your VM does not have a public IP address, then there will not be a public URL listed.

PhpPgAdmin

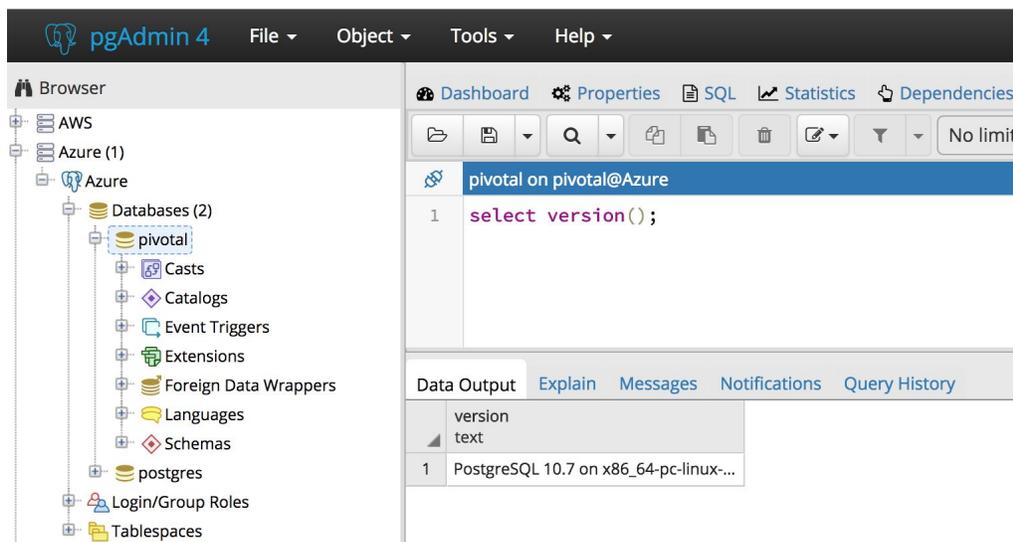
This web based SQL tool has been configured to start automatically. A self-signed SSL certificate is created automatically for you so you will see a web browser warning that the Certificate Authority is invalid. It is safe to proceed with the self-signed certificate.

Note: Connect with user "pivotal" and use the password found in /opt/pivotal/postgres/variables.sh. This file is readable by OS user "pivotal".



Client Tool

Connecting with a remote client tool like pgAdmin 4 is also very easy to do using the public IP address as user "pivotal" and password provided in /opt/pivotal/postgres/variables.sh.



Pivotal Postgres Additional Features

Backups

A backup script runs via cron every day at 4:00 AM local time. A new complete database is created and stored in /var/lib/pgsql/10/backups. The script will maintain the 3 latest backups.

Both the location of the backup and retention are managed in `/opt/pivotal/postgres/variable.sh`. You can also change the schedule using crontab.

```
crontab -e

0 4 * * * source /home/pivotal/.bashrc;
/opt/pivotal/postgres/backup.sh > /home/pivotal/backup.log 2>&1 <
/home/pivotal/backup.log
```

The backup log is stored in `/home/pivotal/backup.log`

MADlib

The analytical package MADlib is pre-installed and ready to use. More information about MADlib can be found here: <https://madlib.apache.org/>

PostGIS

The spatial database extender is pre-installed and ready to use. More information about PostGIS can be found here: <https://postgis.net/>

pgBouncer

This is a connection pooler that is included with Pivotal Postgres. 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 here: <https://pgbouncer.github.io/>

pgBouncer is configured to listen on port 5432 which is the default port usually used by Postgres. Postgres 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 Postgres 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 "session" with max client connections of 100 and default pool size of 20. These settings can be changed but these defaults provide a good starting point for most installations.

Configuration for pgBouncer are located in `/etc/pgbouncer/pgbouncer.ini` and logs are in `/var/log/pgbouncer/pgbouncer.log`.

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

bouncer start

Starts pgBouncer.

bouncer stop

Stops pgBouncer.

bouncer pause

Pauses pgBouncer.

bouncer resume

Resumes pgBouncer.

Additional Resources

Installation Logs

Installation of Pivotal Postgres includes detailed logs using journalctl.

```
sudo journalctl -u pivotal
```

Operating System Patches

A simple script has been provided which updates the deployed VM to be updated with the latest patches. This script should be executed as root and can be found in `/opt/pivotal/postgres/patch_image/`

Version History

Version 10.7

Initial release.