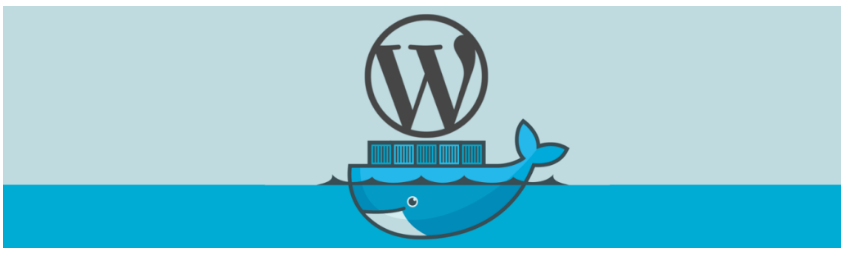
IBM Emerging Technologies Blog

thoughts, inquiries, observations, hacking and test results

WordPress on Bluemix Containers

May 22, 2015 Miguel Clement 16 comments



A guide to Using app binding to connect a WordPress instance to a database.

Why?

Docker Containers are ideal for hosting WordPress sites for three reasons: official images, Volumes, and scalability.

Search …

Recent Posts

- IBM Emerging Technologies Contributes Notebook Tech to Project Jupyter
- IBM Containers Launch in London
- Gridplumb: a grid-based diagrammer for the web
- Apache Spark – Utilizing Access Point Wi-Fi Data
- TrueNorth – Next Generation

- The Official WordPress Image: WordPress maintains the WordPress docker image on docker hub. This is an official image so you can be sure it is stable and set up properly.
- Docker Volumes: Volumes are used as persistent file storage. We use volumes as a file system for WordPress. If you stop, restart or even delete your wordpress container the volume will persist. This allows us to save media and install plugins/themes directly to the volume. Using volumes also makes migrating and scaling your WordPress site very easy!
- Scalability: With Bluemix docker containers you can increase the Memory of your containers, and Number of CPU's. You can also create a cluster of identical containers to host your site. This is great for load balancing! We also have the added benefit of being able to scale the database services independently. Your WordPress Database can come from any of the MySQL services in the Bluemix catalog. (this tutorial will use clearDB)

How?

This tutorial assumes you have already installed the cf "ic" plugin and docker CLI. For instructions on how to install the CLIs see the bluemix docs. The first three steps are done with docker and the cf ic clis.

Step 1: Pull the official image locally

```
$ docker pull ibmjstart/bluemix-wordpress
```

Step 2: Tag the wordpress image for pushing up to your bluemix registry

you'll need to replace [namespace] with your namespace.

Cognitive Computing

Twitter Widget

Tweets Follow

IBM jStart Team 14h
@IBMjStart
RT: @RHancock9
#ibminsight bit.ly/1jznUAt pic.twitter.com

Expand

IBM jStart Team 17h
@IBMjStart
Right NOW: #Spark and #IBMSPSS

Tweet to @IBMjStart

Other Emerging Tech Blogs

- Randy Wilcox
- Nick Heidloff
- Ryan Baxter
- Jon Kaufman

Archives

- September 2015
- August 2015

```
$ docker tag ibmjstart/bluemix-wordpress registry.n
```

Step 3: Push [namespace]/wordpress up to your registry

```
$ docker push registry.ng.bluemix.net/[namespace]/w
```

note: you may need to use “cf ic login” before pushing up to your bluemix registry.

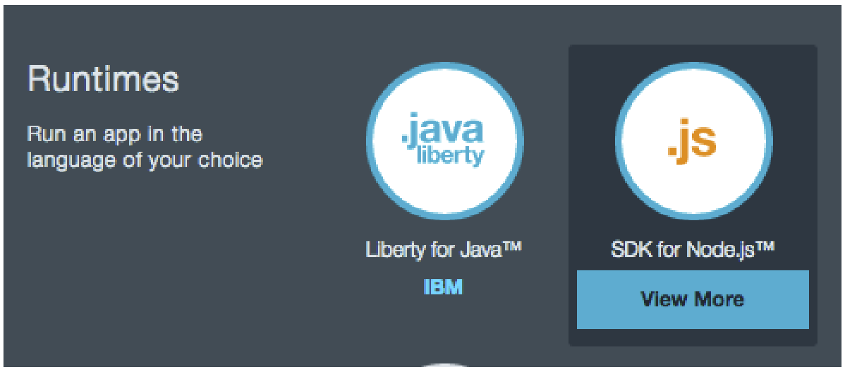
Step 4: Create the volume that will serve as the WordPress file system

```
$ cf ic volume create [Volume Name]
```

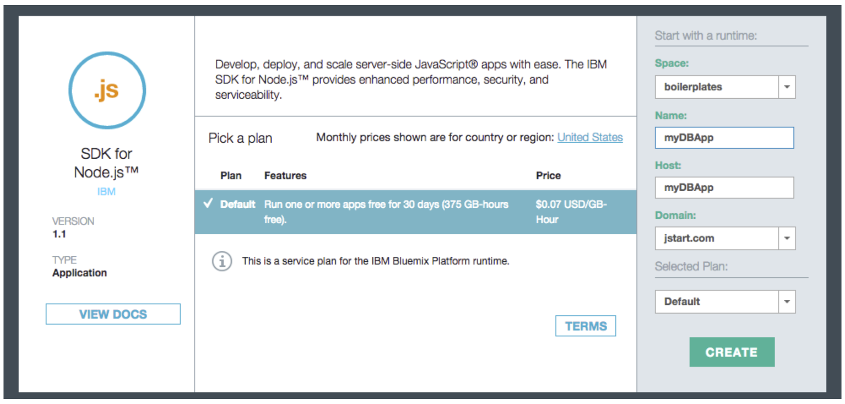
You can name the volume whatever you want but remember the name for when you run the container in step 6.

Step 5: Create a MySQL Service

Lets head over to Bluemix for the rest of the tutorial. In bluemix, you will need to create a new app to bind a MySQL DB to. I recommend using the **SDK for node.js runtime** from the catalog.



Go ahead and create it:



VERSION 1.1
TYPE Application

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

Pick a plan   Monthly prices shown are for country or region: United States

Plan	Features	Price
✓ Default	Run one or more apps free for 30 days (375 GB-hours free).	$0.07 USD/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Start with a runtime:
Space: boilerplates
Name: myDBApp
Host: myDBApp
Domain: jstart.com
Selected Plan:
Default

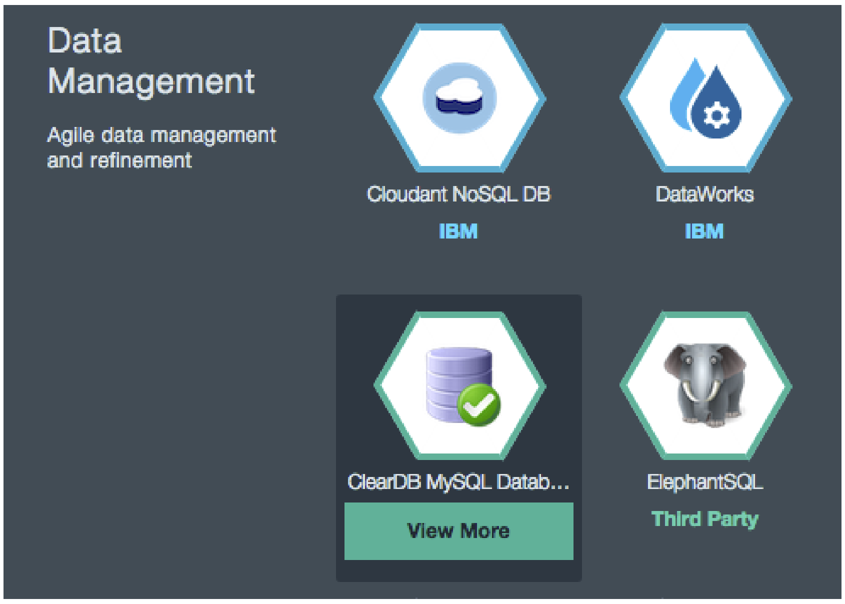
Make sure to choose a unique name for your app.

July 2015
June 2015
May 2015

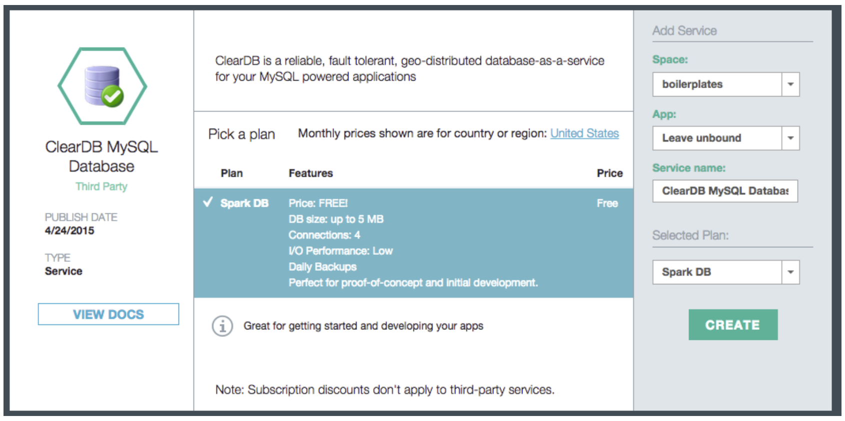
Meta

- Log in
- Entries RSS
- Comments RSS
- WordPress.org

Once your app is running go ahead and create a MySQL Service instance in the catalog:



Finally make sure to bind it to the node.js app we just created. This is done in the App section.


ClearDB MySQL Database
Third Party
PUBLISH DATE
4/24/2015
TYPE
Service
VIEW DOCS

ClearDB is a reliable, fault tolerant, geo-distributed database-as-a-service for your MySQL powered applications

Pick a plan Monthly prices shown are for country or region: United States

Plan	Features	Price
✓ Spark DB	Price: FREE! DB size: up to 5 MB Connections: 4 I/O Performance: Low Daily Backups Perfect for proof-of-concept and initial development.	Free

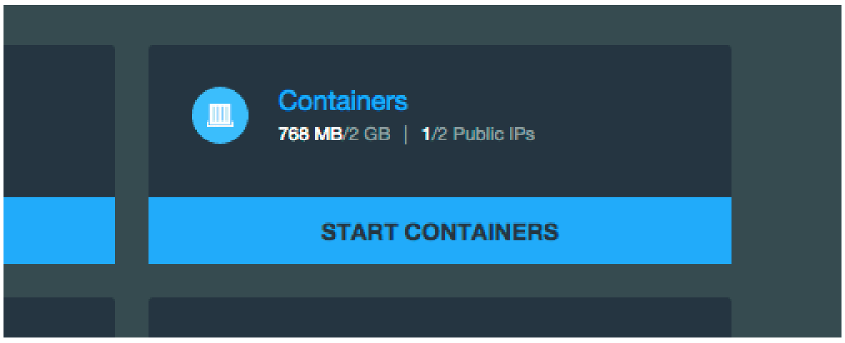
Great for getting started and developing your apps

Note: Subscription discounts don't apply to third-party services.

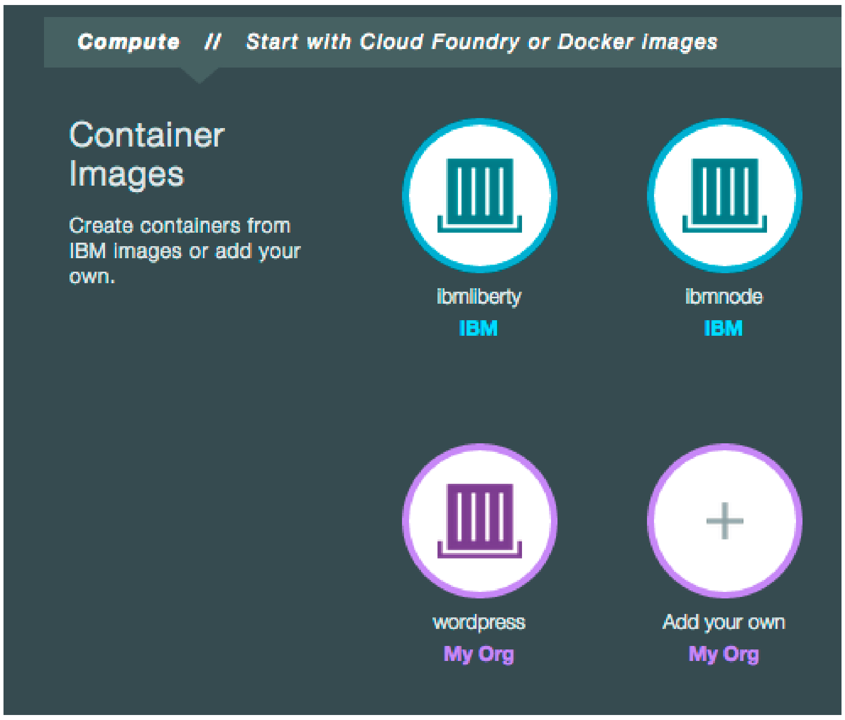
Add Service
Space: boilerplates
App: Leave unbound
Service name: ClearDB MySQL Databa:
Selected Plan: Spark DB
CREATE

Step 6: Run the Container with the WordPress Image

Go on to your Dashboard and select Start Containers:



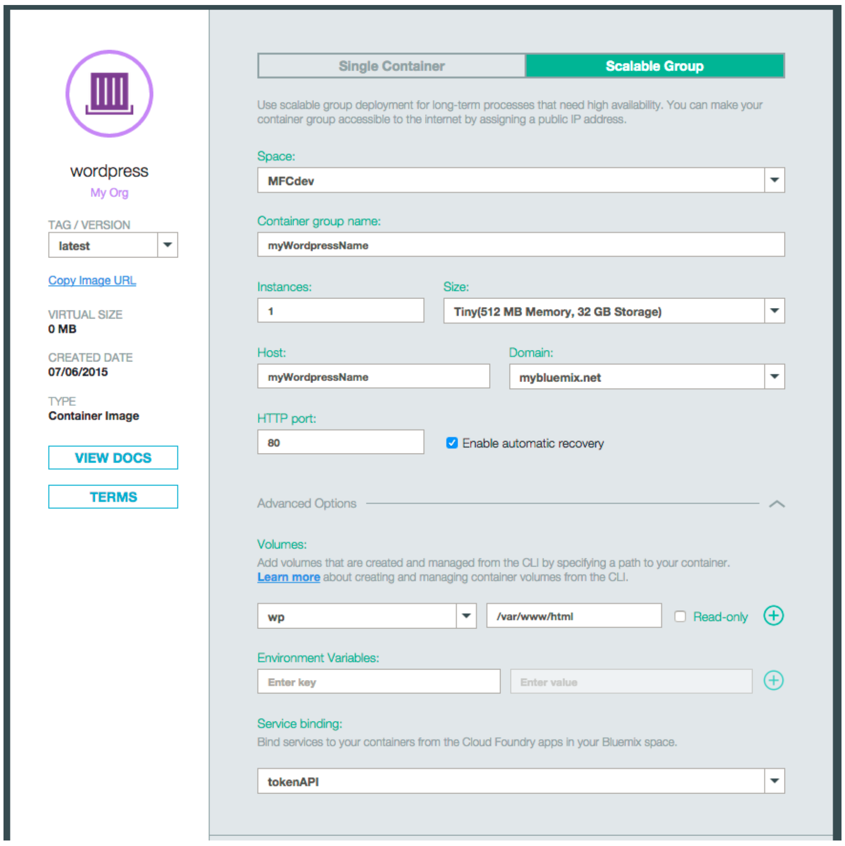
Next you will see this (your wordpress image should be listed):


Compute // Start with Cloud Foundry or Docker Images
Container Images
Create containers from IBM images or add your own.
ibmliberty
IBM
ibmnode
IBM
wordpress
My Org
Add your own
My Org

Click on the wordpress image. That will take you to the following page. there a quite a few details on this page that we need to get right.

1. Make sure you are under the scalable group tab (Important!)
2. Give your container group a unique name
3. choose mybluemix.net as the route domain.
4. choose as many instances as you would like (1 should be fine)
5. choose a unique hostname
6. open HTTP port 80 (very important)
7. Click advanced options
8. Select the volume you created in step 4 from the dropdown. Set the mount path to "/var/www/html"
9. Bind the bridge app we made in step 5. by selecting the bridge app from the dropdown.

The rest can be left as default.



wordpress
My Org

TAG / VERSION: latest

VIRTUAL SIZE: 0 MB

CREATED DATE: 07/06/2015

TYPE: Container Image

Use scalable group deployment for long-term processes that need high availability. You can make your container group accessible to the internet by assigning a public IP address.

Space: MFCdev

Container group name: myWordpressName

Instances: 1

Size: Tiny(512 MB Memory, 32 GB Storage)

Host: myWordpressName

Domain: mybluemix.net

HTTP port: 80

[✓] Enable automatic recovery

Volumes:
Add volumes that are created and managed from the CLI by specifying a path to your container.
Learn more about creating and managing container volumes from the CLI.

wp /var/www/html [ ] Read-only

Environment Variables:
Enter key Enter value

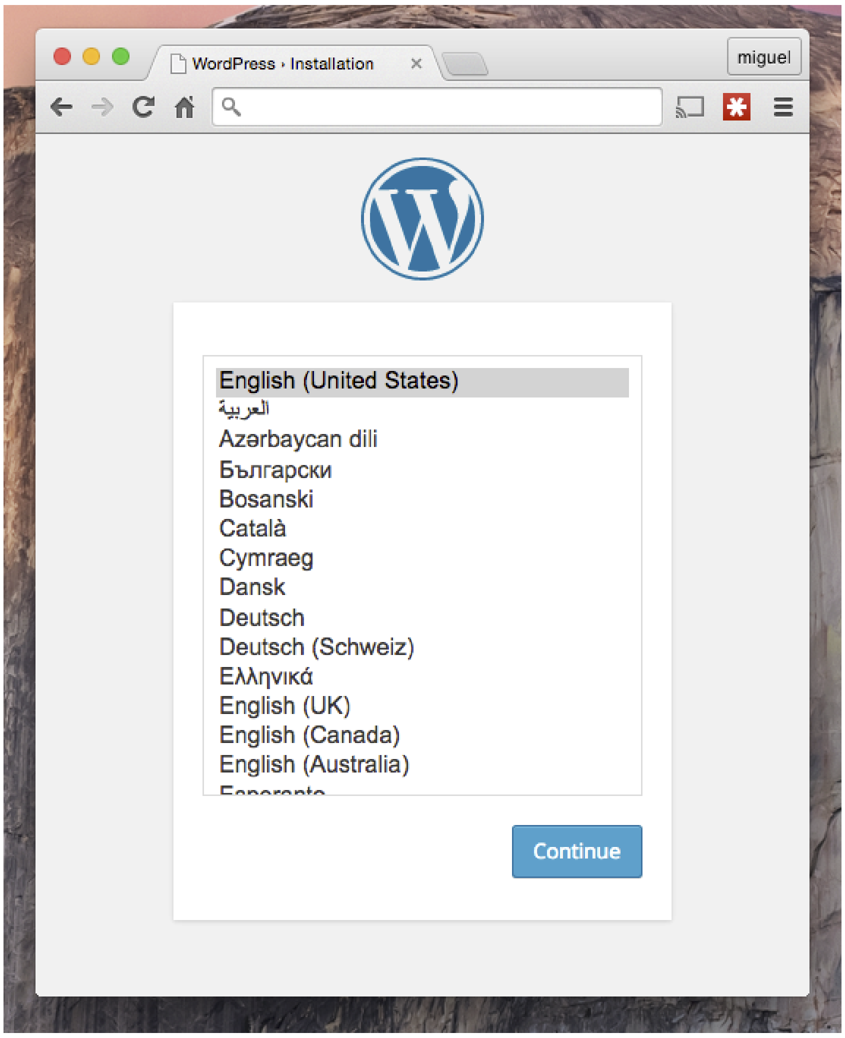
Service binding:
Bind services to your containers from the Cloud Foundry apps in your Bluemix space.

tokenAPI

hit **create.**

Step 7: Hello WordPress

visit your page using the [Hostname].mybluemix.net url you selected in step 5 (note: this may take a few minutes to deploy, so give it a while). Your page should look like this:



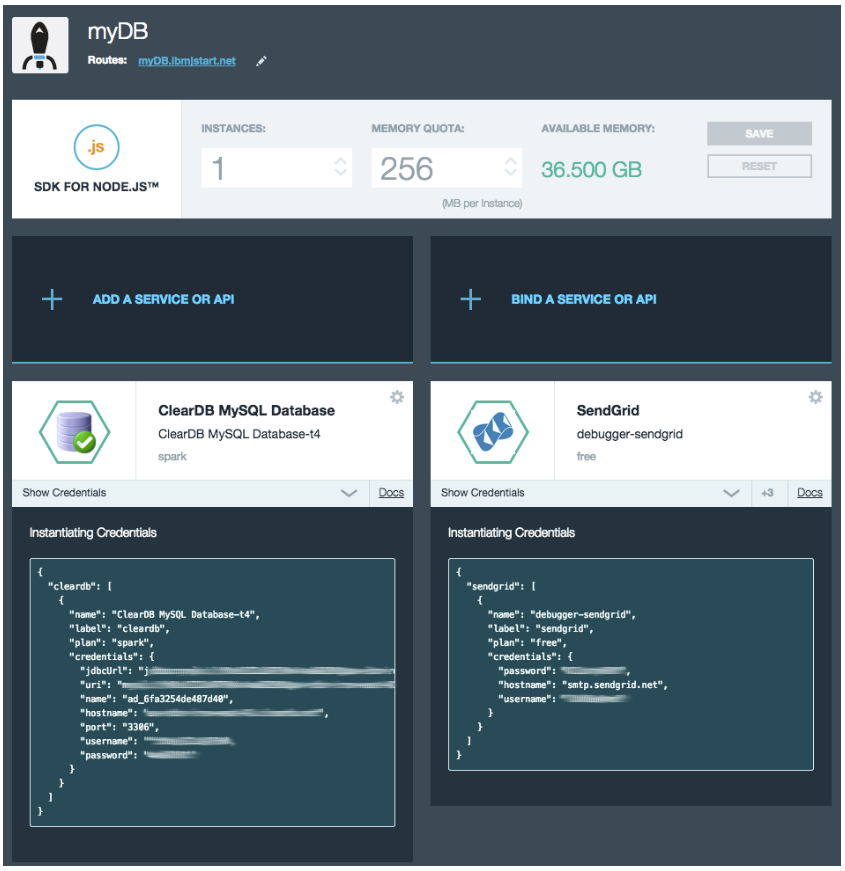
You should now have a fully functional WordPress install on containers! run through the setup and you will be good to go. **If you would like to learn more about how our WordPress image works, you can read about it HERE**

Setting up email in your WordPress Container (optional):

The container does not have a SMTP server configuration so the WordPress email functionality will not work. We can use the SendGrid plugin to remedy this.

Step 0: Create a SendGrid service in bluemix. Bind it to your MyDB application.

SendGrid can be found in the catalog. This is what your app's dashboard will look like when you have successfully bound your SendGrid service instance.



Instantiating Credentials

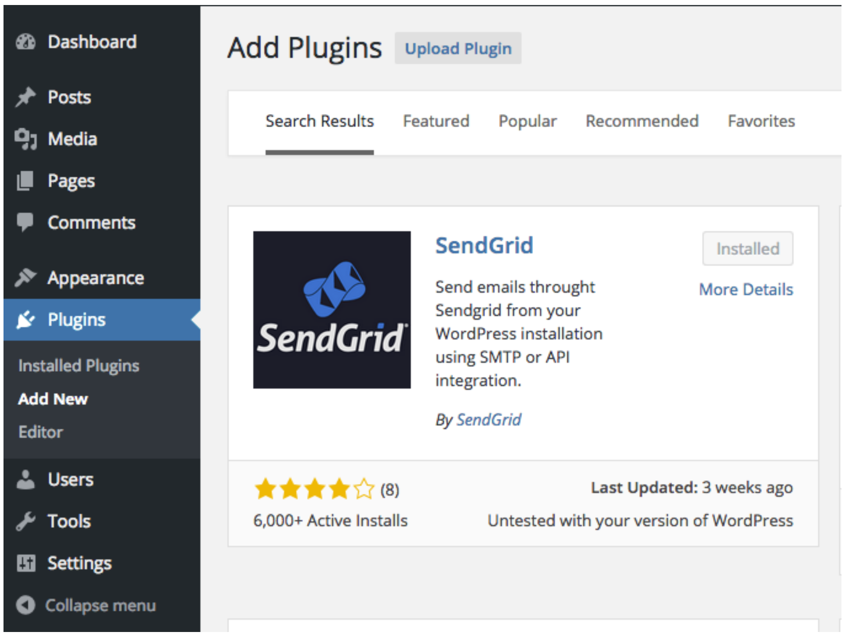
```
{
  "cleardb": [
    {
      "name": "ClearDB MySQL Database-t4",
      "label": "cleardb",
      "plan": "spark",
      "credentials": {
        "jdbcUrl": "j[redacted]
        "uri": "m[redacted]
        "name": "ad_6fa3254de487d40",
        "hostname": [redacted],
        "port": "3306",
        "username": [redacted]
        "password": [redacted]
      }
    }
  ]
}
```

Instantiating Credentials

```
{
  "sendgrid": [
    {
      "name": "debugger-sendgrid",
      "label": "sendgrid",
      "plan": "free",
      "credentials": {
        "password": [redacted],
        "hostname": "smtp.sendgrid.net",
        "username": [redacted]
      }
    }
  ]
}
```

Step 1. Install the SendGrid plugin in your WordPress instance.

The plugin can be installed in the wordpress ui (Plugins > AddNew > Search (Sendgrid))
Once it is installed, make sure to activate it.



SendGrid

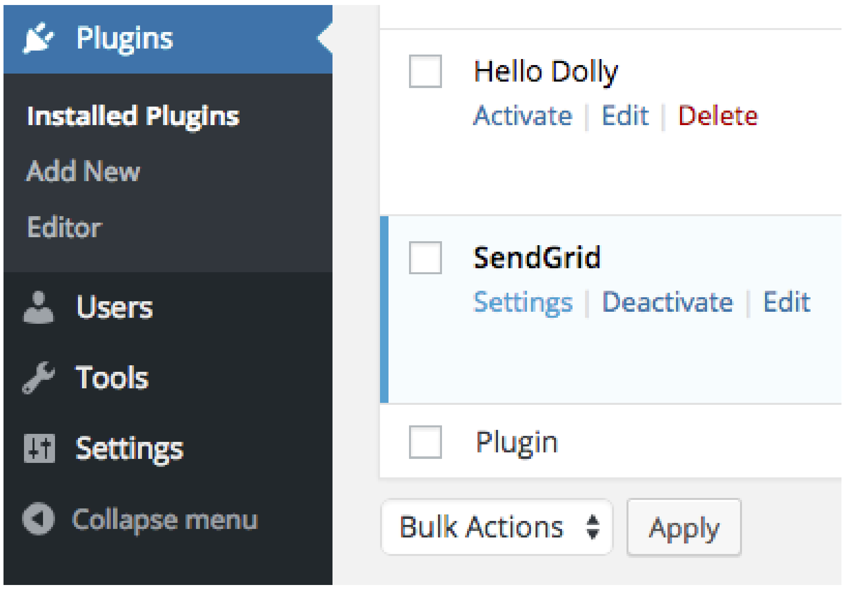
Send emails throught Sendgrid from your WordPress installation using SMTP or API integration.

By SendGrid

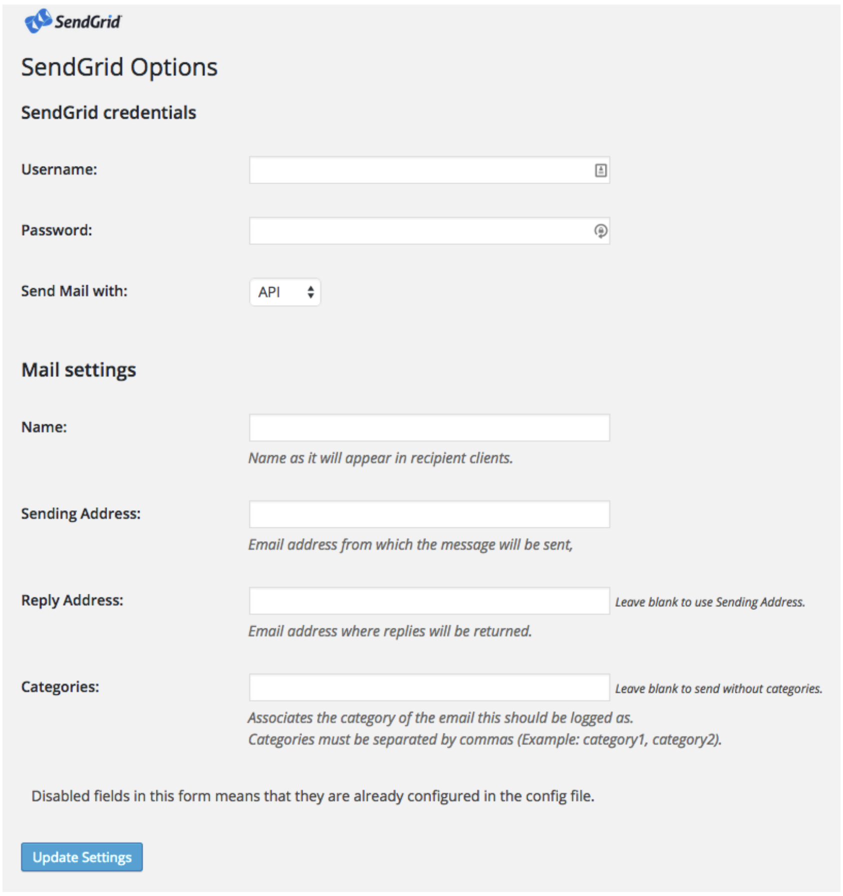
★★★★☆ (8)
6,000+ Active Installs
Last Updated: 3 weeks ago
Untested with your version of WordPress

Step 2. Set Credentials

Go to Plugins > Installed Plugins > SendGrid Settings



Once there go ahead and fill out the credentials.



SendGrid

SendGrid Options

SendGrid credentials

Username:	
Password:	
Send Mail with:	API

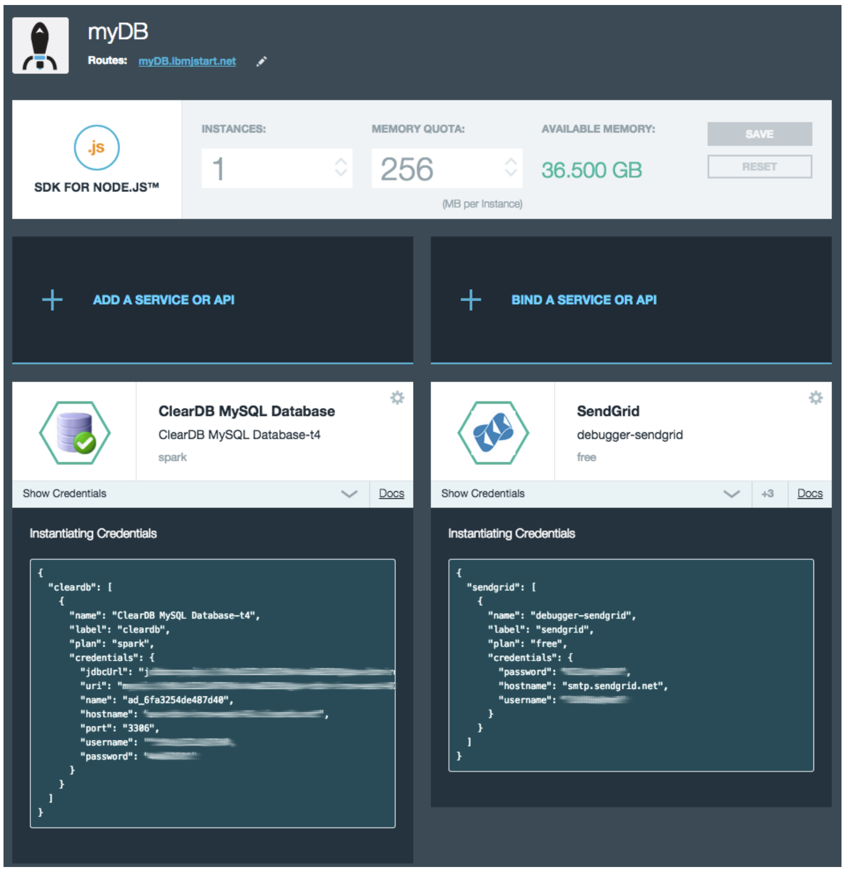
Mail settings

Name:	Name as it will appear in recipient clients.
Sending Address:	Email address from which the message will be sent,
Reply Address:	Leave blank to use Sending Address. Email address where replies will be returned.
Categories:	Leave blank to send without categories. Associates the category of the email this should be logged as. Categories must be separated by commas (Example: category1, category2).

Disabled fields in this form means that they are already configured in the config file.

Update Settings

The credentials can be found back on your dashboard. From this screen (Click "Show Credentials" under SendGrid logo)



Instantiating Credentials

```
{
  "cleardb": [
    {
      "name": "ClearDB MySQL Database-t4",
      "label": "cleardb",
      "plan": "spark",
      "credentials": {
        "jdbcUrl": "j[redacted]
        "uri": "m[redacted]
        "name": "ad_6fa3254de487d40",
        "hostname": [redacted],
        "port": "3306",
        "username": [redacted]
        "password": [redacted]
      }
    }
  ]
}
```

Instantiating Credentials

```
{
  "sendgrid": [
    {
      "name": "debugger-sendgrid",
      "label": "sendgrid",
      "plan": "free",
      "credentials": {
        "password": [redacted],
        "hostname": "smtp.sendgrid.net",
        "username": [redacted]
      }
    }
  ]
}
```

Fill out all the info and Save. You can try sending a test email on that same page. If the email sends properly, then your WordPress install is complete!

Bio | in LinkedIn | Latest Posts | GitHub

Miguel Clement

Miguel is a Computer Science Senior at Texas A&M Univeristy. He joined the jStart Emerging Technology Team in January 2015 and has been exploring the cutting edge ever since.

Category: Containers **Tags:** Containers, Wordpress

16 comments

Pingback: Accessing Bluemix Services In Containers -

Jorge Flor
July 7, 2015 at 1:11 pm

Watch this error:
MacBook-Pro-de-Jorge:BluemixWordpress jorgeflor$ docker pull ibmjstart/bluemix-wordpress
Post http:///var/run/docker.sock/v1.19/images/create?fromImage=ibmjstart%2Fbluemix-wordpress%3Alatest: dial unix /var/run/docker.sock: no such file or directory. Are you trying to connect to a TLS-enabled daemon without TLS?
MacBook-Pro-de-Jorge:BluemixWordpress jorgeflor$

What can i do?

Reply

Miguel Clement
July 7, 2015 at 1:21 pm

Hi Jorge,
this looks like an issue with your boot2docker vm. Try pulling another image, and I'm sure it will give the same error. The best solution I've found has been to restart the VM. this can be done by calling:

"boot2docker down"
(followed by)
"boot2docker up"

if any errors appear after "up" they should tell you more about what's going on.

Reply

Morten Borklund
August 6, 2015 at 12:02 pm

Hi
I followed your example, but I got en error when I tried to access my “wordpress-site” (step 7).

Error establishing a database connection

Any hint will do

Best regards
Morten

Reply

Morten Borklund
August 7, 2015 at 7:02 am

Note to myself: Now it works fine. The problem must have been in Bluemix…

Reply

Miguel Clement
August 6, 2015 at 1:27 pm

There must be an issue with your clearDB connection. Here is how it should work:

1. create an empty node app that will be used as a bridge between ClearDB and the container.
2. bind a clearDB service to the bridge node app.
3. go through the container image creation steps and when you get to this page: http://blog.ibmjstart.net/wp-content/uploads/2015/05/Screen-Shot-2015-07-06-at-1.27.21-PM.png Make sure that you selected the bridge application in the advanced options > service bindings dropdown, this is very important.

If you are sure all of these things are set and configured properly and it still gives the error let me know an I will help you debug. Thank you for reading!

Reply

Murad Korejo
August 7, 2015 at 3:38 am

Miguel,

This is a great post! I was able to get WordPress running in a container in my Bluemix org, and I retrieved the DB creds from the Bluemix dashboard, but when I try to configure WordPress to use the DB (via the CF bridge application), I get a WordPress error which says:

Sorry, but I can’t write the wp-config.php file.

You can create the wp-config.php manually and paste the following text into it.

Any ideas? Thanks again for the post regardless.

Reply

Murad Korejo
August 7, 2015 at 7:13 am

Actually, I was able to get past the issue by launching the site in an incognito window. I have another question though and will post a comment for that.

Reply

Pingback: Hello world! | Head in the Cloud

Murad Korejo

August 12, 2015 at 1:32 pm

Miguel,

I am trying to update the WordPress site in my container, but I'm not sure how. There is no FTP access to the site to my knowledge, and I can't seem to be able to place new files on the persistent volume directly. Any thoughts?

Reply

Lee Surprenant
September 2, 2015 at 9:59 pm

Because the site is installed on a persistent volume, you should be able to rely on WordPress's Automatic Background Updates for minor core updates and you should be able to click the "Update Now" button for major core updates as described at https://codex.wordpress.org/Updating_WordPress.
If you are having trouble with either of these, we'd like to hear more about it.

Reply

Rene Meyer
August 26, 2015 at 8:56 am

Hello Miguel,
thank you for your tutorial. in your Dockerfile you referenced a version of WordPress Docker base image which was changed in the meantime in a way making your tutorial not working anymore.
If you look on old revision of the WP Docker github project you see that many changes happened in the [docker]entrypoint.sh:

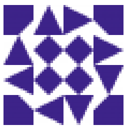
https://github.com/docker-library/wordpress/tree/1420b4c44ba0cf13d2d1d1e41bbc3bc74e83ce9f/apache
https://github.com/docker-library/wordpress/tree/master/apache
I have now an example which works, including supervisor based ENTRYPOINT and ssh support:
https://github.com/cloud-dach/wp-bluemix-container
So I created a combination between the older revision of the base WP Dockerfile and your Dockerfile and scripts.

Reply

Alex Lewitt
August 26, 2015 at 4:25 pm

Thank you for bringing up the issue. We are looking into the problem to see if we can fix the tutorial.

Reply



Rene Meyer
September 3, 2015 at 8:50 pm

Hello Miguel,

I had an issue in my Bluemix space, so there is nothing wrong with your tutorial. After I created a new space I can now repeat the tutorial each time without errors. The cf env of the bound application and MySQL service was not passed to the container, when I created container groups. This is now working. So I learned a lot about Docker and the Container service. Thank you very much for that tutorial.

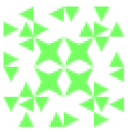
Reply

Alex Lewitt

September 4, 2015 at 1:22 pm

Glad to hear its all working for you! Thanks for taking the time to work through the tutorial!

Reply



Lee Surprenant

September 2, 2015 at 9:43 pm

The steps in the article worked for me with the existing ibmjstart/bluemix-wordpress image. Additionally, steps 1-3 can be simplified to the following one-liner by using the IBM Containers "cpi" command:

```
 cf ic cpi ibmjstart/bluemix-wordpress
registry.ng.bluemix.net/[namespace]/wordpress
```

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

Post Comment