DESTINATION CLOUD
DEPLOYING APPLICATIONS TO THE CLOUD WITH DOCKER

Ryan Baxter - @ryanjbaxter - 11.16.15
GO PRO
Giveaway
Stop By Bluemix Booth
DEPLOYMENT CYCLES ARE GETTING FASTER
CONTINUOUS DELIVERY IS AWE-SOME
DEVELOPMENT
TESTING
STAGING
PRODUCTION
CLOUD
“…it works for me.”
Build your apps, your way.

Use a combination of the most prominent open-source compute technologies to power your apps. Then, let Bluemix handle the rest.

**Instant Runtimes**
App-centric runtime environments based on Cloud Foundry.

[Check out runtimes on Bluemix](#)

**IBM Containers**
Portable and consistent delivery of your app without having to manage an OS.

[Check out containers on Bluemix](#)

**Virtual Machines**
Get the most flexibility and control over your environment with VMs.

[Check out VMs on Bluemix](#)
Data and Analytics

Essential data services; limitless possibilities

- Analytics for Apache Hadoop
  IBM BETA

- Apache Spark
  IBM

- BigInsights for Apache Hadoop
  IBM

- Cloudant NoSQL DB
  IBM

- dashDB
  IBM

- DataWorks
  IBM

- Elasticsearch by Compose
  IBM

- Geospatial Analytics
  IBM

- IBM DB2 on Cloud
  IBM

- Insights for Twitter
  IBM

- MongoDB by Compose
  IBM

- Object Storage
  IBM BETA

- Object Storage (v2)
  IBM BETA

- PostgreSQL by Compose
  IBM

- Predictive Analytics
  IBM BETA

- Redis by Compose
  IBM

- SQL Database
  IBM

- Streaming Analytics
  IBM
PRIVATE REGISTRY
GUI or CLI
100+ SERVICES
MONITOR CPU
CONTAINER GROUPS
DEPLOYMENT PIPELINE
MACHINE SIZES
VULNERABILITY SCANNING
BlueChatter

kauffecup
why hello there

jon
how goes it

Type something insightful

Built By The BlueMix Dev Advocate Team
https://github.com/IBM-Bluemix/bluechatter
var redisService = appEnv.getService('redis-chatter');
var credentials;
if (!redisService || redisService == null) {
  if (isDocker) {
    credentials = {
      "hostname": "redis", "port": 6379
    };
  } else {
    credentials = {
      "hostname": "127.0.0.1", "port": 6379
    };
  }
} else {
  if (isDocker) {
    // This works around a problem with networking when deployed to Bluemix in a docker container
    // For some reason it takes about 30 seconds for the networking to come up on the container
    // so we sleep here before we continue on and use these credentials to connect
    console.log('The app is running in a Docker container on Bluemix so we are ' +
                'sleeping for 90 seconds waiting for the networking to become active.');
    require('sleep').sleep(90);
  }
  credentials = redisService.credentials;
}
FILE STRUCTURE

bluechatter
|-- node_modules
|-- public
|-- app.js
|-- admin.js
|-- docker-compose.yml
|-- Dockerfile
|-- package.json
RUNNING
LOCALLY
STEP 1: GET THE CODE
Workspace  git clone https://github.com/IBM-Bluemix/bluechatter.git
Cloning into 'bluechatter'...
remote: Counting objects: 101, done.
remote: Total 101 (delta 0), reused 0 (delta 0), pack-reused 101
Receiving objects: 100% (101/101), 29.25 KiB | 0 bytes/s, done.
Resolving deltas: 100% (52/52), done.
Checking connectivity... done.

Workspace  cd bluechatter
bluechatter  git:(master)
WITHOUT DOCKER
NODE + REDIS

1. Install dependencies
2. Kick off Redis server
3. Start Node app
npm WARN package.json BlueChatter@0.0.1 No license field.

> sleep@3.0.0 install /Users/jkaufman/Workspace/bluechatter/node_modules/sleep
> node-gyp rebuild

CXX(target) Release/obj.target/node_sleep/sleep.o
SOLINK_MODULE(target) Release/node_sleep.node

> BlueChatter@0.0.1 install /Users/jkaufman/Workspace/bluechatter
> node admin.js track

...
redis-server

2609:C 15 Oct 13:03:44.119 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf

2609:M 15 Oct 13:03:44.121 * Increased maximum number of open files to 10032 (it was originally set to 256).

Redis 3.0.5 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 2609
http://redis.io

2609:M 15 Oct 13:03:44.122 # Server started, Redis version 3.0.5
2609:M 15 Oct 13:03:44.122 * DB loaded from disk: 0.000 seconds
2609:M 15 Oct 13:03:44.122 * The server is now ready to accept connections on port 6379
bluechatter git:(master) npm start
> BlueChatter@0.0.1 start /Users/jkaufman/Workspace/bluechatter
> node app.js

Express server listening on port 6002
The subscriber redis client has connected!
The publisher redis client has connected!
BlueChatter

Enter a username to get started

Username

Go!

Built By The BlueMix Dev Advocate Team
TO SHARE (and guarantee behavior):

1. Have same OS
2. Have same version of Node
3. Have same version of Redis
WITH DOCKER
1. Build the app into an image
2. Run the image in a container
FROM registry.ng.bluemix.net/ibmnode:latest
COPY ./ bluechatter
WORKDIR bluechatter
RUN npm install -d --production
EXPOSE 80
ENV PORT 80
ENV DOCKER true
CMD ["node", "app.js"]
```bash
    ➜  bluechatter  git:(master)  docker build -t bluechatter:test .
Sending build context to Docker daemon 142.8 kB
Step 0 :  FROM registry.ng.bluemix.net/ibmnode:latest
       --->  8f962f6afc9a
Step 1 :  COPY ./ bluechatter
       --->  51605b64b015
Removing intermediate container 7b0f6e22fb99
Step 2 :  WORKDIR bluechatter
       ---> Running in e1c0436890d5
       --->  6f607172a3ff
Removing intermediate container e1c0436890d5
Step 3 :  RUN npm install -d --production
       ---> Running in 278bf3713ef6
cfenv@1.0.3  node_modules/cfenv
    └── ports@1.1.0
    └── underscore@1.8.3
     └── js-yaml@3.4.3 (esprima@2.6.0, argparse@1.0.2)
npm info ok
```
Removing intermediate container 58b5613bfe6f
Step 4 : EXPOSE 80
    ---> Running in f18f213e2bc6
Removing intermediate container f18f213e2bc6
Step 5 : ENV PORT 80
    ---> Running in 214f959e23ca
Removing intermediate container 214f959e23ca
Step 6 : ENV DOCKER true
    ---> Running in 701f4a794e70
Removing intermediate container 701f4a794e70
Step 7 : CMD node app.js
    ---> Running in 74c50bc04811
Removing intermediate container 74c50bc04811
Successfully built 250ab2421de16
IDENTIFY AND RUN THE IMAGE
<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>VIRTUAL SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>bluechatter</td>
<td>test</td>
<td>250ab421de16</td>
<td>12 seconds ago</td>
<td>673.2 MB</td>
</tr>
<tr>
<td>node</td>
<td>0.10-onbuild</td>
<td>f8a19007bc68</td>
<td>2 days ago</td>
<td>633.4 MB</td>
</tr>
</tbody>
</table>

```bash
➜ bluechatter git:(master) docker run -t bluechatter:test
Express server listening on port 80
There was an error with the subscriber redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
There was an error with the publisher redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
There was an error with the subscriber redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
There was an error with the publisher redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
There was an error with the subscriber redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
There was an error with the publisher redis client Error: Redis connection to redis:6379 failed - getaddrinfo ENOTFOUND
```
DEFINE AND RUN MULTI-CONTAINER APPLICATIONS
docker-compose.yml

```yaml
web:
  build: .
  ports:
    - "80:80"
    - "8080:8080"
  links:
    - redis
redis:
  image: redis
```
RUNS WITH ONE COMMAND:
docker-compose up
Pulling redis (redis:latest)...
latest: Pulling from library/redis
library/redis:latest: The image you are pulling has been verified. Important: image verification is a tech preview feature and should not be relied on to provide security.

Digest: sha256:ea780255d8b42745f14e61c75c83eba2c7254f6ccc9f97f975a64490f5d3bf7e
Status: Downloaded newer image for redis:latest
Creating bluechatter_redis_1...
Building web...
Step 0 : FROM registry.ng.bluemix.net/ibmnode:latest
... Successfully built 5a9d88a42af2
Creating bluechatter_web_1...
Attaching to bluechatter_redis_1, bluechatter_web_1
Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf

Redis 3.0.5 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 1

http://redis.io

The server is now ready to accept connections on port 6379
Express server listening on port 80
The subscriber redis client has connected!
The publisher redis client has connected!
IDENTIFY
OUR MACHINE’S IP
AND CONNECT
<table>
<thead>
<tr>
<th>NAME</th>
<th>ACTIVE</th>
<th>DRIVER</th>
<th>STATE</th>
<th>URL</th>
<th>SWARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td></td>
<td>virtualbox</td>
<td>Stopped</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```bash
-> bluechatter  git:(master)  docker-machine ls
```

```bash
192.168.99.101

-> bluechatter  git:(master)  docker-machine ip bluechatter
```

```bash
192.168.99.101

-> bluechatter  git:(master)
```
TO SHARE (and guarantee behavior):

1. Have Docker
SOLVED:
INCONSISTENT ENVIRONMENTS
TO THE CLOUD
STEPS:
1. BUILD + PUSH IMAGE TO REGISTRY
2. CREATE BRIDGE APP + BIND REDIS
3. START CONTAINER FROM IMAGE
1. BUILD + PUSH IMAGE TO REGISTRY

cf ic build -t bluechatter
➡️  bluechatter  git:(master)  cf ic build  -t  bluechatter  .
Sending build context to Docker daemon 142.8 kB
Step 0 : FROM registry.ng.bluemix.net/ibmnoded:latest
...
Successfully built e0d9661a2c3f
The push refers to a repository [registry.ng.bluemix.net/jdkaufma/bluechatter]
Sending image list
Pushing repository registry.ng.bluemix.net/jdkaufma/bluechatter (1 tags)
Image 83e4d6e6b9cf already pushed, skipping
Image d2a0ecffe6fa already pushed, skipping
e288e2539065: Image successfully pushed
803521c35f1b: Image successfully pushed
Pushing tag for rev [e0d9661a2c3f] on {https://registry.ng.bluemix.net/v1/repositories/jdkaufma/bluechatter/tags/latest}
➡️  bluechatter  git:(master)
Boilerplates
Get started with a new app, now

- Apache Spark Starter
- Internet of Things Foundation Starter
- Java Cloudant Web Starter
- Java DB Web Starter
- LoopBack Starter
- MobileFirst Services Starter
- Personality Insights Java Web Starter
- StrongLoop Arc
- Node-RED Starter
- Python Flask
- Ruby Sinatra
- Vaadin Rich Web Starter

Compute
Start with Cloud Foundry or Docker images

- Runtimes
  Run an app in the language of your choice

- Liberty for Java™
- SDK for Node.js™
- PHP
- Python
- Ruby
- Community buildspacks
2. MAKE A BRIDGE + BIND REDIS
3. START CONTAINER FROM 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.

**Single Container**

**Scalable Group**

**Space:**

dev

**Container group name:**

Enter a container group name

**Instances:**

Examples: 2

**Size:**

Micro (256 MB Memory, 16 GB Storage)

**Host:**

Enter host

**Domain:**

mybluemix.net

**HTTP port:**

Enter an HTTP port or select a default

- Enable automatic recovery

**Advanced Options**

**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.

**Environment Variables:**

Add a new environment variable

**Service binding:**

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

Select a Cloud Foundry app
Use single container deployment for short-term processes. You can make your container accessible to the Internet by assigning a public route.

**Space:**
- dev

**Container name:**
- b

**Size:**
- Micro (256 MB Memory, 16 GB Storage)

**Public IP address:**
- Leave unassigned

**Public IP Address:**
- 22.80

**Container Pricing**

<table>
<thead>
<tr>
<th>Plan</th>
<th>Features</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Use one or more containers free for 365 GB-hours free each month, use 20 GB free external storage and 2 static public IPs free each month with your containers</td>
<td>$0.028 USD/GB-Hour</td>
</tr>
</tbody>
</table>

Monthly prices shown are for country or region: United States
SOME SETUP REQUIRED
HOOK UP SHELL TO DOCKER
```bash
➜ ~ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM
default                virtualbox   Stopped

➜ ~ docker-machine env bluechatter
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.101:2376"
export DOCKER_CERT_PATH="/Users/jkaufman/.docker/machine/machines/bluechatter"
export DOCKER_MACHINE_NAME="bluechatter"
# Run this command to configure your shell:
# eval "$(docker-machine env bluechatter)"
➜ ~ eval "$(docker-machine env bluechatter)"
➜ ~
```
CLI DECISIONS
<table>
<thead>
<tr>
<th></th>
<th>Cloud Foundry Plug-in</th>
<th>ICE (IBM Containers Extension)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs native Docker CLI commands</td>
<td>Yes</td>
<td>Local image-development only</td>
</tr>
<tr>
<td>Requires Docker and Cloud Foundry CLI</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Requires Python, Pip, and Setuptools</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
https://www.ng.bluemix.net/docs/
CLICK "CONTAINERS"
CLICK "SETTING UP THE IBM CONTAINERS CLI"
https://www.ng.bluemix.net/docs/containers/container_cli_ov.html
Setting up the IBM Containers CLI

Some container creation and management tasks can be done from the Bluemix™ Dashboard, but for maximum capabilities, you can set up one of the IBM® Containers CLIs.

Choosing a CLI

If you are just getting started with IBM Containers, use the Cloud Foundry plug-in for IBM Containers. With the plug-in, you are required to install fewer prerequisites than ICE, but all of the same functions are available. Plus, you have the ability to run native Docker commands.

If you used ICE in the past, you can continue to use it to manage your containers in Bluemix.
I PREFER
CLOUD FOUNDRY
PLUGIN
cf ic build    docker build
cf ic images   docker images
cf ic inspect  docker inspect
cf ic restart  docker restart
    cf ic run    docker run
    cf ic start  docker start
    cf ic stop   docker stop
cf ic version  docker version
<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>registry.ng.bluemix.net/jdkaufma/bluechatter</td>
<td>latest</td>
<td>e0d9661a2c3f</td>
<td>3 days ago</td>
<td>448.1 MB</td>
</tr>
<tr>
<td>registry.ng.bluemix.net/ibm-node-strong-pm</td>
<td>latest</td>
<td>ef21e9d1656c</td>
<td>13 days ago</td>
<td>528.7 MB</td>
</tr>
<tr>
<td>registry.ng.bluemix.net/ibmliberty</td>
<td>latest</td>
<td>2209a9732f35</td>
<td>13 days ago</td>
<td>492.8 MB</td>
</tr>
<tr>
<td>registry.ng.bluemix.net/ibmnodnode</td>
<td>latest</td>
<td>8f962f6afc9a</td>
<td>13 days ago</td>
<td>429 MB</td>
</tr>
<tr>
<td>registry.ng.bluemix.net/ibm-mobilefirst-starter</td>
<td>test</td>
<td>97513e56aaa7</td>
<td>13 days ago</td>
<td>769.1 MB</td>
</tr>
<tr>
<td>registry.ng.bluemix.net/ibm-mobilefirst-starter</td>
<td>7.1</td>
<td>97513e56aaa7</td>
<td>13 days ago</td>
<td>769.1 MB</td>
</tr>
</tbody>
</table>
INSTALL IBM CONTAINER CLOUD FOUNDRY CLI PLUGIN
cf install-plugin https://static-ice.ng.bluemix.net/ibm-containers-mac

Attempting to download binary file from internet address...
9086880 bytes downloaded...
Installing plugin /var/folders/h9/00ngb57132bgxqs_jvzpbw40000gn/T/ibm-containers-mac...
OK
Plugin IBM-Containers v0.8.723 successfully installed.
cf plugins
Listing Installed Plugins...
OK

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Version</th>
<th>Command Name</th>
<th>Command Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-Containers</td>
<td>0.8.723</td>
<td>ic</td>
<td>IBM Containers Plugin</td>
</tr>
</tbody>
</table>

~
→ bluechatter git:(master) cf login api.ng.bluemix.net

API endpoint: https://api.ng.bluemix.net

Email> jdkaufma@us.ibm.com

Password>
Authenticating...
OK

API endpoint: https://api.ng.bluemix.net (API version: 2.27.0)
User: jdkaufma@us.ibm.com
Org: jdkaufma@us.ibm.com
Space: dev

** Retrieving client certificates from IBM Containers
** Storing client certificates in /Users/jkaufman/.ice/certs
Successfully retrieved client certificates

bluechatter git:(master)
NOT COVERED
RESILIENT CONTAINER GROUPS
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:**

```
dev
```

**Container group name:**

```
Enter a container group name
```

**Instances:**

```
Example: 2
```

**Size:**

```
Micro (256 MB Memory, 16 GB Storage)
```

**Host:**

```
Enter host
```

**Domain:**

```
mybluemix.net
```

**HTTP port:**

```
Enter an HTTP port or select a default
```

- [ ] Enable automatic recovery

**Advanced Options**

**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.

**Environment Variables:**

[Add a new environment variable](#)

**Service binding:**

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

[Select a Cloud Foundry app](#)
DEPLOYMENT PIPELINES
MONITORING
<table>
<thead>
<tr>
<th>SERVICES</th>
<th>MEMORY</th>
<th>STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>bluechatter-zendcon</td>
<td>256 MB</td>
<td>16 GB</td>
</tr>
</tbody>
</table>

**MEMORY USED**

- Memory Usage Over Time (MB)

- Memory Usage: 33.48 MB

**Services bound from**: zendcon-bridge:

- **Redis Cloud**
  - redis-chatter
  - 30mb

**Show Credentials**
### LOG MESSAGES OVER TIME

![Graph showing log messages over time](image)

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Machine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/20/2015 11:02</td>
<td>prod-da09-vizio2-host-07</td>
<td>&quot;log&quot;: &quot;The app is running in a Docker container on Bluemix so we are sleeping for 90 seconds waiting for the networking to become active.&quot;</td>
</tr>
<tr>
<td>10/20/2015 11:04</td>
<td>prod-da09-vizio2-host-07</td>
<td>&quot;log&quot;: &quot;The publisher redis client has connected!&quot;</td>
</tr>
<tr>
<td>10/20/2015 11:04</td>
<td>prod-da09-vizio2-host-07</td>
<td>&quot;log&quot;: &quot;The subscriber redis client has connected!&quot;</td>
</tr>
<tr>
<td>10/20/2015 11:04</td>
<td>prod-da09-vizio2-host-07</td>
<td>&quot;log&quot;: &quot;Express server listening on port 80&quot;</td>
</tr>
</tbody>
</table>

Showing 1 to 4 of 4 entries
LET'S RECAP
YOUR APP

DOCKER
IBM Bluemix

YOUR APP

DOCKER

YOUR APP

DOCKER

YOUR APP

DOCKER

IBM BLUEMIX
RESOURCES
Docker Toolbox

IBM Containers

Bluechatter Sample App

https://www.docker.com/toolbox

https://www.ng.bluemix.net/docs/containers/container_index.html

https://github.com/IBM-Bluemix/bluechatter
QUESTIONS

Ryan Baxter - @ryanjbaxter - 11.16.15