Windows Server Containers for Cloud Foundry

Sunjay Bhatia, Matthew Horan, Pivotal
Introductions

Sunjay Bhatia
Software Engineer, Pivotal
sunjayBhatia

Matthew Horan
Software Engineer, Pivotal
mhoran
What have we been working on?

- Maintaining the Windows 2012R2 stack
  - Buildpacks (thanks @stefanschneider)
  - Configurable HTTP/TCP health check mode (instead of forcing HTTP)
  - Improved “bind mount” implementation
  - Enabled the use of Instance Identity Credentials
  - “Security enhancements”
  - cf-deployment opsfile
What have we been working on?

- Windows 2016 stack

```
dorchester:/Volumes/SSD/workspace [alfredo] $ cf stacks
Getting stacks in org dev / space s as admin...
OK

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cflinuxfs2</td>
<td>Cloud Foundry Linux-based filesystem</td>
</tr>
<tr>
<td>windows2012R2</td>
<td>Windows Server 2012 R2</td>
</tr>
<tr>
<td>windows2016</td>
<td>Windows Server 2016</td>
</tr>
</tbody>
</table>
```

- Now with 2x more stacks than Linux!
... but why?

- We envision a first class .NET developer experience on CF
- Reduce the burden of leveraging Windows on CF
- Windows 2012R2 stack gets us part of the way, but many advanced platform features are still infeasible
Shortcomings of Windows Server 2012R2 Stack

- Containers pieced together using job objects, Windows firewall, filesystem ACLs
Shortcomings of Windows Server 2012R2 Stack

- Lack of true isolation and resource limiting
  - Shared host registry
  - Shared filesystem
  - Shared network interface (and unable to firewall localhost)
  - No network bandwidth limits
  - No CPU limits
Shortcomings of Windows Server 2012R2 Stack

- Isolation primitives are ... primitive
  - `conhost.exe` exploit

```c
SetProcessExecutable(&attrib_list, L"\SystemRoot\System32\Conhost.exe");
SetProcessToken(&attrib_list, token_handle);

result = ZwCreateUserProcess(
    handle,
    &thread_handle,
    ...,
    PROCESS_BREAKAWAY_JOB, // Process Flags
    CREATE_SUSPENDED,        // Thread Flags
    ...
    &attrib_list);

if ( result < 0 )
    *handle = NULL;
else
    ObCloseHandle(thread_handle, 0);
```
How do we improve the experience?

- For app developers?
- For CF component teams?
How do we improve the experience?

- Windows Server 2016 stack
- Adopting established Cloud Foundry development patterns
- Adopting industry standards (OCI)
- Move towards closer collaboration with the core Garden team
Design Goals

Windows Server 2012R2 Parity

● Only supporting buildpack lifecycle (no Docker yet)
● Application Security Groups
● Resource limits
  ○ Memory
  ○ Disk
● Same class of apps can be pushed to both stacks
Design Goals

Pragmatic Parity with Linux

● Improved experience for .NET developers
● Enabling more existing platform features
● Set ourselves up for ability to enable new platform features
Windows Server 2016 Stack Improvements

- Complete filesystem isolation
- Container rootfs
  - Simplifies BOSH stemcell
The RootFS
Multiple layers as packaged in windows2016fs-release.

Windows Server Core container base image

Windows Server Container

Windows Features (.NET, HWC, etc.)

.NET modules (URL Rewrite module), etc.

User config

Utilities (git, tar)

Buildpack

Droplet

Application

App Staging

Resulting container image at runtime.
Windows Server 2016 Stack Improvements

- CPU limits (shares)
- Users are unique to each container
- Each container has its own registry instance
- Individual network “compartment” per container
- Container processes cannot communicate with host
- Bind Mounts, not “Bind Mounts”
Windows Server 2016 Stack Architecture

- The following were drawn by a literal architect. Thanks @awmartin.
Guardian

BOSH Agent

Diego Rep

Metron Agent

Consul Client

Local Route Emitter

Guardian

Garden Server

container plugin

network plugin

rootfs plugin
Demo

- cf-deployment
- cf push
- cf ssh
Windows Server Semi-Annual Channel (1709)

- Actively working with Windows Server Containers team
- Major networking and performance improvements
- Improved process isolation
- Smaller rootfs
- “Sidecar” containers (shared container network compartment)
- Microsoft moving to a 6 month release cycle
Needs Improvement

- Memory limits do not constrain memory mapped files
- No PID limits
- Containers are “semi-privileged”
Roadmap

- GrootFS/OCI (Docker) image push
- Concourse worker
- Nano server (multi-rootfs/see GrootFS)
Under Consideration

Hyper-V Isolation

- Shared kernel by default
- Hyper-V isolation is an option
  - Not implemented in winc/CF
  - Intended for “hostile multi-tenant workloads”
  - Requires nested virtualization support
  - Heavy-weight in comparison with shared kernel isolation
Call to Action

- We’re hiring!
- We love pull requests
- Start using winc/Guardian on Windows
- cf-deployment