Packaging your D app

DConf’19 – Lionello Lunesu – Enuma Technologies
@LioLunesu
So you want to package your D app?
Cloud

- Google Cloud Platform
- Amazon Web Services
- Microsoft Azure
- IBM Bluemix
- DigitalOcean
- OpenShift by Red Hat
- Alibaba Cloud
After Principal Component Analysis:
Docker

- One container for local testing and cloud deployments
- Contains all dependencies
- Convenient for web services
  - and CLI tutorials
- Less convenient for local CLI tools that access state
- Containers can be pushed to Docker Hub registry
- Supported by most cloud providers
- Stateless containers are ready to be auto-scaled
FROM dlanguage/dmd
COPY hello.d .
RUN dmd hello.d
CMD ["./hello"]

$ docker build .
Successfully built 62525b23b4f5

$ docker run 62525b23b4f5
Hello world
Dependency hell

- Container is ~560MiB. Does it matter?
- Built container still has all build tools; not needed for running
- What else does the base image contain?
- What base image to use instead of dl{\textit{anguage/dmd}}?

- Best base image is \texttt{scratch}, 0-bytes
  - But must copy all dependencies too
  - Recursive \texttt{ldd}?
FROM dlanguage/dmd AS build-stage
RUN apt-get update && apt-get -y install \
    git \
    python-setuptools

# Install dockerize from GIT (or COPY from local submodule)
RUN git clone --depth=1 https://github.com/larsks/dockerize.git
WORKDIR dockerize
RUN python setup.py install

WORKDIR /
COPY hello.d .
RUN dmd hello.d
RUN dockerize -o /output -n /hello

FROM scratch
COPY --from=build-stage /output /
CMD ["./hello"]
FROM dlanguage/dmd AS build-stage
RUN apt-get update && apt-get -y install \
  git \
  python-setuptools
RUN git clone --depth=1 https://github.com/larsks/dockerize.git
WORKDIR dockerize
RUN python setup.py install

WORKDIR /
COPY dub.json .
RUN dub build ; true
COPY hello.d source/
RUN dub build
RUN dockerize -o /output -n /hello

FROM scratch
COPY --from=build-stage /output /
CMD ["./hello"]
> Nix

• The Purely Functional Package Manager
• Cross-platform: *nix, macOS, FreeBSD
  • Even works in WSL!*
• One package for local testing and cloud deployments
• Nix expressions describe how to build a package from source
• All dependencies in /nix/store, **no global install**
• Paths based on hash of their inputs, not name or version
• Packages can be merged into the nixpkgs repository
• Automatic runtime dependencies

$ curl https://nixos.org/nix/install | sh
with (import <nixpkgs> {});

stdenv.mkDerivation {
    name = "hellod";
    nativeBuildInputs = [ dmd ];
    src = ./.;
    buildPhase = ''
        dmd hello.d
    ''; 
    installPhase = ''
        mkdir -p $out/bin
        cp hello $out/bin
    ''; 
}

$ nix-env --if ./hellod.nix
Dub with Nix

- Nix builds are pure, deterministic
  - no arbitrary internet access
- Need derivations for all dependencies
- For each dependency ...
  - Checkout the required version from its GIT repository
  - `dub build` or `dub add-local`

- Use [dub2nix](https://noun.ly/boy) to generate `dub.selections.nix` from json
Dub-aware Nix derivation

```nix
with (import ./mkDub.nix {}); # from dub2nix repo

mkDubDerivation {
  version = "0.1.0";
  src = ./.;
  deps = [];
  # defaults to dub.selections.nix
}

$ nix-env -if https://github.com/lionello/dub2nix/archive/master.zip

$ dub2nix save
# This file was generated by [https://github.com/lionello/dub2nix](https://github.com/lionello/dub2nix) v0.1.0

```
[ {
  fetch = {
    type = "git";
    url = "https://github.com/vibe-d/vibe.d.git";
    rev = "v0.8.5";
    sha256 = "0s1caxqmq2497j5x8h06f44nr597h9zac8qxxml9531kaqkhbzgy";
  };
}

{ 
  fetch = {
    type = "git";
    url = "https://github.com/wilzbach/stdx-allocator.git";
    rev = "v2.77.5";
    sha256 = "03av8zp5p6vf6fg005xbmwnjfw96jyrr4dcj4m56c4a3vx7v72pk";
  };
} ]
```
Docker with Nix

```nix
with (import <nixpkgs> {{}});

let
  hellod = (import ./.hellod.nix);
in
  dockerTools.buildImage {
    name = hellod.name;
    config = {
      Cmd = ["${hellod}/bin/hello"];
    };
  }
```
> **Chocolatey**

- The package manager for Windows
- Uses Nuget packages
  - optional install/update/uninstall powershell scripts
- Packages can be uploaded to the Chocolatey Community Feed

```powershell
> Set-ExecutionPolicy Bypass -Scope Process -Force;
```
<?xml version="1.0" encoding="utf-8"?>
<package xmlns="http://schemas.microsoft.com/packaging/2015/06/nuspec.xsd">
  <metadata>
    <id>helloD</id>
    <version>0.1.0</version>
    <authors>Lionello Lunesu</authors>
    <description>Simple package for hello world in D</description>
  </metadata>
  <files>
    <file src="tools\hello.exe" target="tools" />
  </files>
</package>
>> Chocolatey overview

> choco new

> choco pack

> choco install hellod

> choco uninstall hellod
<table>
<thead>
<tr>
<th></th>
<th><strong>Docker</strong></th>
<th><strong>Nix</strong></th>
<th><strong>Chocolatey</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Build OS</strong></td>
<td>Linux, macOS</td>
<td>Linux, FreeBSD, macOS</td>
<td>Windows, Linux (Mono)</td>
</tr>
<tr>
<td><strong>Target OS</strong></td>
<td>Linux</td>
<td>Linux, FreeBSD, macOS</td>
<td>Windows</td>
</tr>
<tr>
<td><strong>Target</strong></td>
<td>Cloud service</td>
<td>Cloud, CLI, GUI</td>
<td>CLI, GUI</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>Dockerfile</td>
<td>Nix derivation</td>
<td>Nuspec</td>
</tr>
<tr>
<td><strong>Artifact</strong></td>
<td>Image (.tar.gz)</td>
<td>Nix store path (.nar)</td>
<td>.nupkg</td>
</tr>
<tr>
<td><strong>Runtime Dependencies</strong></td>
<td>Explicit</td>
<td>Automatic</td>
<td>Explicit</td>
</tr>
<tr>
<td><strong>Package registry</strong></td>
<td>Docker Hub</td>
<td>Nixpkgs repo</td>
<td>Community Feed</td>
</tr>
<tr>
<td><strong>Build command</strong></td>
<td>docker build</td>
<td>nix-build</td>
<td>choco pack</td>
</tr>
<tr>
<td><strong>Install/run command</strong></td>
<td>docker pull / docker run</td>
<td>nix-env / nix-shell</td>
<td>choco install</td>
</tr>
</tbody>
</table>
References

• https://docs.docker.com/develop/develop-images/multistage-build/
• https://github.com/larsks/dockerize
• https://nixos.org/nix/
• https://github.com/lionello/dub2nix
• https://chocolatey.org
> Thank you!

Lionello Lunesu – Enuma Technologies
lio@enuma.io
@LioLunesu