Creating a Self-Installing "Package"

Brian Gerard
November, 2017

What Are We Talking About?

Something Simple To Use...

```
# Install in a test directory

bash$ ./install-yourself

...

# Install in the 'live' directory

bash$ ./install-yourself --live

...
```

But... Why?!?

Outrunning The Bureaucracy

Because I Grow Tired of Adding Cover Sheets to My TPS Reports

- We need something relatively foolproof

- We need it **before** we finish running through the approvals for other solutions

- It should still allow us to move quickly, safely forward

- It should not require any changes to the application code

How's This Going To Work?

Something Like This...

Self-Installer Sections

Preamble; set extraction paths and filenames; begin decoding of...

A big blob of text in a heredoc, which is really the base64-encoded script that actually installs the "real" package

Check that the extraction succeeded; if so, begin decoding of...

A second (bigger) blob of text in another heredoc, which is actually the base64-encoded gzipped tarball that contains the code to install

Check that the extraction succeeded; if so, run the decoded installation script

How Is It Built?

A Lot Like This...

Build Steps

Create a temp directory whose subdirectories mirror the final directory structure , perform any build steps and place the installable scripts, etc. where they belong under it.

Create a gzipped tarball of the temp directory in the packaging directory.

Copy the installation script into the packaging directory, making any necessary build time modifications.

Create the installer out of the pieces of the extraction script (head, middle, tail), with the base64-encoded installation script and tarball sandwiched between the sections.

The Grand Experiment

(A.K.A. "The Demo")

Questions?

- Github link (includes these slides):

  - https://github.com/briangerard/self-installing-package-poc

- Contact: bgerard@gmail.com

- Thank you!