Package Masonry: Generating Executables for Interpreted Languages

Frank Ruiz, SRE
@fruizperman
LESS WORK FOR THE SRE
What are build systems?

Tools?
High Level Overview

Generates Executables

- LIBRARY
  - Build
  - Test
  - Dependencies
  - Versioning

BUILD SYSTEM

Interpreter & Libraries

Packaged Executable
THE MEAT
Build Tools
Simplicity Is Bliss
$ ./myapp
First Up - Python

Python Executable (Zip File)

Python has had the ability to execute directories or ZIP-format archives as scripts since version 2.6

- PEP 411
- ./python myapp.zip

Add some tooling and you get:

- Essentially Virtual Environments
- ./myapp

Enables:

- Self contained archive
PEX and Pyzzer
Search PYPI
Next Up - Ruby
Let’s review some possibilities

High Level Approaches
• Portable Ruby Binaries
• AOT Compiler

Why you care:
• Self contained libs.
• Self contained interpreter.
Traveling Ruby, and Ruby Packer
Search Gems
Last Item - NodeJS

The options

The node way:
- AOT Compiler

Why you care:
- Self contained libs.
- Self contained interpreter.
pkg, nexe, and Node-Packer
Check npmjs
Build Systems - Pants, Bazel, and Buck
Isn’t this already solved?
Problem - Maintenance and Security

Container

Base Image Layers

Application Layer

Dependency Layer

OVERLAP

OVERLAP
High Level Overview

- Cloud VM
- Bare Metal
- Container

Packaged Executable
Validation points.
Not all tools are created equal.
Thank you!
fruiz@box.com

Were Hiring!
https://www.box.com/careers