Latest Software Stable Environment

A Simpler Way

Presented by
Troy Dawson
tdawson@redhat.com

Presentation Overview

1. The Problem
2. Past Solutions
3. Modularity – A Simpler Way
4. Demo

The Problem

The Problem

- Users want the latest version of Program A
- Users want an older version of Program B
- Users want Programs A & B on the same machine
- How do you keep a stable, enterprise grade, system, with both bleeding edge and old packages at the same time.

Past Solutions

Tar and Zip files

- Pros
  - Simple
  - Users can do it themselves
- Cons
  - Security
  - Support
  - Duplicating

SCL’s

Software Collection Libraries

- Pros
  - Multiple version of package on same machine
  - RPM based
    - Easy for security
    - Supported by Red Hat
    - Easy to duplicate

SCL’s

Software Collection Libraries

- Cons
  - Collection is in non-standard place
  - Have to setup environment before use
    - Often breaks scripts and automation
    - Often confuses developers
  - Often, not portable

Containers

- Pros
  - Easy to administer
  - Easy to duplicate
- Cons
  - Doesn’t really solve the problem
    - You still have to choose between rpm’s and tarballs
    - Only as secure as the software in the container

A Simpler Way

Modularity

- What is it?
  - Similar to yum/dnf groups but with versions
  - rpm’s are standard rpms, nothing custom
  - Only one version of a package at a time
    - Unless the packages are already setup to have multiple versions at a time
      - java-1.7.0-openjdk, java-1.8.0-openjdk
    - If that is the case, then you don’t really need modularity

Modularity

- What is it?
  - Able to automatically upgrade or downgrade between versions
    - dnf module install packages:2.5
    - dnf module install packages:1.8
      - Automatically downgrades all packages in the “packages:2.5” module to the versions in in the “packages:1.8” module
      - If there are any extra packages in “packages:1.8”, it installs them
      - If there are any extra packages in “packages:2.5”, it removes them

Modularity

- Three Parts
  - Module Server Side
    - Build the packages
    - Setup the repos
  - Module Developers
    - Configure the modules
    - Build modules (very similar to building rpms)
  - Module Users
    - Install and user modules
    - Should be as easy (or easier) than using groups

Modularity

- History
  - Modularity v1
    - Modules all the way down
    - F27 based
    - Canceled soon after it’s first release
  - Modularity v2
    - Hybrid OS

Modularity

- Modularity v1 (Modules all the way down)
  - Everything is a module
    - Bootup is a module (host)
    - BaseOS is a module (platform)
    - All applications are modules (AppStream)
  - Problems
    - When anything gets changed on platform, **all** modules have to be rebuilt. Very resource intensive.
    - Not all Fedora rpm packages would want to put their packages into modules. Would need a modularity team for everything. Very resource intensive

Modularity

- Modularity v2 (Hybrid OS)
  - Start with standard OS
  - Only use modules for those AppStreams that are enabled.
  - If there is a package in both the standard OS and a modules
    - If the module isn’t enabled, use the standard OS package
    - If the module is enabled, use the module package(s)
  - Was released with F28 GA release.

Demo

Demo

- #Start with standard F28
- dnf module list
- dnf module install nodejs:10
- node --version
- dnf module install nodejs:8
- node --version
- dnf module list

Demo

- man dnf
- dnf install @nodejs:6
- dnf module enable nodejs:9
- dnf module lock nodejs:9
- dnf module info nodejs:8
- dnf module streams

Demo

- dnf install fedpkg
- fedpkg module-*
- fedpkg module-overview

Demo

- man dnf
- dnf install @nodejs:6
- dnf module enable nodejs:9
- dnf module lock nodejs:9
- dnf module info nodejs:8
- dnf module streams

Questions?

Contact:
tdawson@redhat.com

References

- https://fedoramagazine.org/modularity-fedora-28-server-edition/
- https://docs.fedoraproject.org/fedora-project/subprojects/fesco/en-US/Using_Modules.html
- https://docs.pagure.org/modularity/
- https://docs.fedoraproject.org/fedora-project/subprojects/fesco/en-US/index.html
- man dnf (Module Command section)