



Exceptional service in the national interest

ONE SCRIPT TO RULE THEM ALL

Lessons Learned and Iterations

David Collins

Empire DevOps

NLIT S³C 2024, Seattle, Washington



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

EMPIRE



- [A Performance Portable Plasma Simulation Code](#)
- Developed under DOE's Advanced Simulation and Computing (ASC) and Advanced Technology Development and Mitigation (ATDM) programs.
- Built on other codes developed at Sandia
 - [Trilinos](#)
 - [Kokkos](#) (snapshot into Trilinos)
 - [DARMA](#) (snapshot into Empire)



[Original whitepaper](#) in 2020 Collegeville Workshop on Scientific Software

Original Problem

- Fragmented and decentralized build & test processes across platforms and developers
 - Developers all had local environment, build & test scripts
 - Different workflows per platform
- Organic growth
- Varied Trilinos configurations (dozens of CMake -D flags)

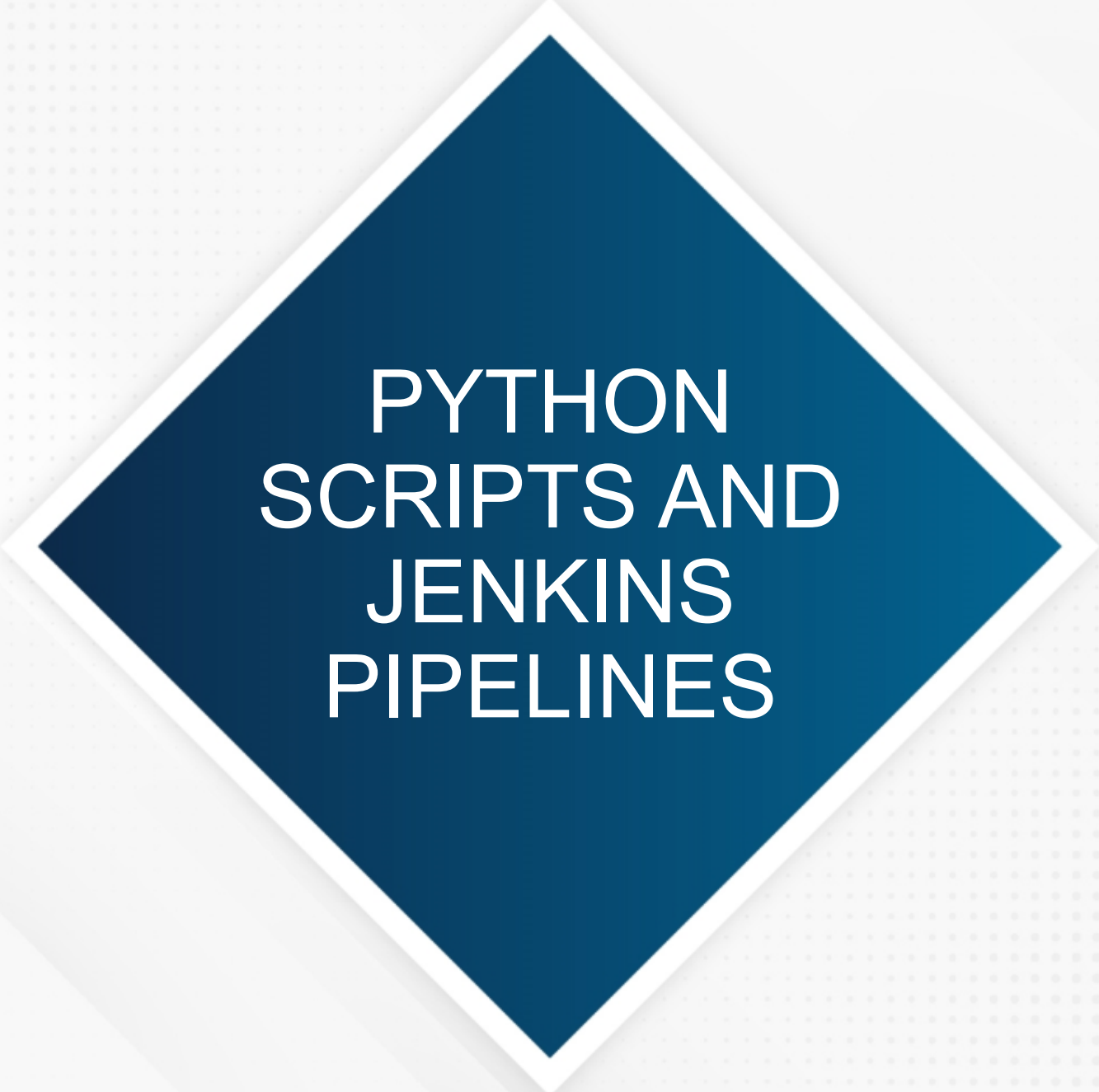
Solution

- BuildScripts repository
 - Three main components: build_empire.py, build_trilinos.py, Jenkins groovy files
 - Unify workflows for all developers on all platforms
 - Unify per-platform environments for all developers

NOTABLE FEATURES OF BUILD_EMPIRE.PY AND BUILD_TRILINOS.PY



- Clone, configure, build, test Trilinos & Empire
- Unified script for Jenkins on all platforms
 - Consistent builds across platforms
- .ini-based environments & CMake configurations
 - Trilinos uses same .ini format
- Detailed logging
 - stdout/stderr, environment, pwd, etc.
 - Invocation
 - Git SHA table
- Dry run mode
- Simple abstraction for developers



PYTHON SCRIPTS AND JENKINS PIPELINES

BUILDSCRIPTS EXAMPLES



```
build_trilinos.py \  
  --compiler gnu \  
  --build-type debug \  
  --node-type openmp \  
  --lib-type shared \  
  --stage \  
    clone \  
    configure \  
    build \  
    ctest \  
    install
```

```
JOB_NAME=Trilinos-gnu-debug-openmp-shared \  
build_trilinos.py \  
  --stage \  
    clone \  
    configure \  
    build \  
    ctest \  
    install
```

- Both are equivalent
- Both run git clone, cmake, ninja, ctest, install with appropriate options.

BUILDSCRIPTS AND JENKINS IN CONTEXT



build_empire.py

build_trilinos.py

jenkins

jobs

build_empire.groovy

build_trilinos.groovy

deploy.groovy

test_deploy.groovy

pipelines

d2m.groovy

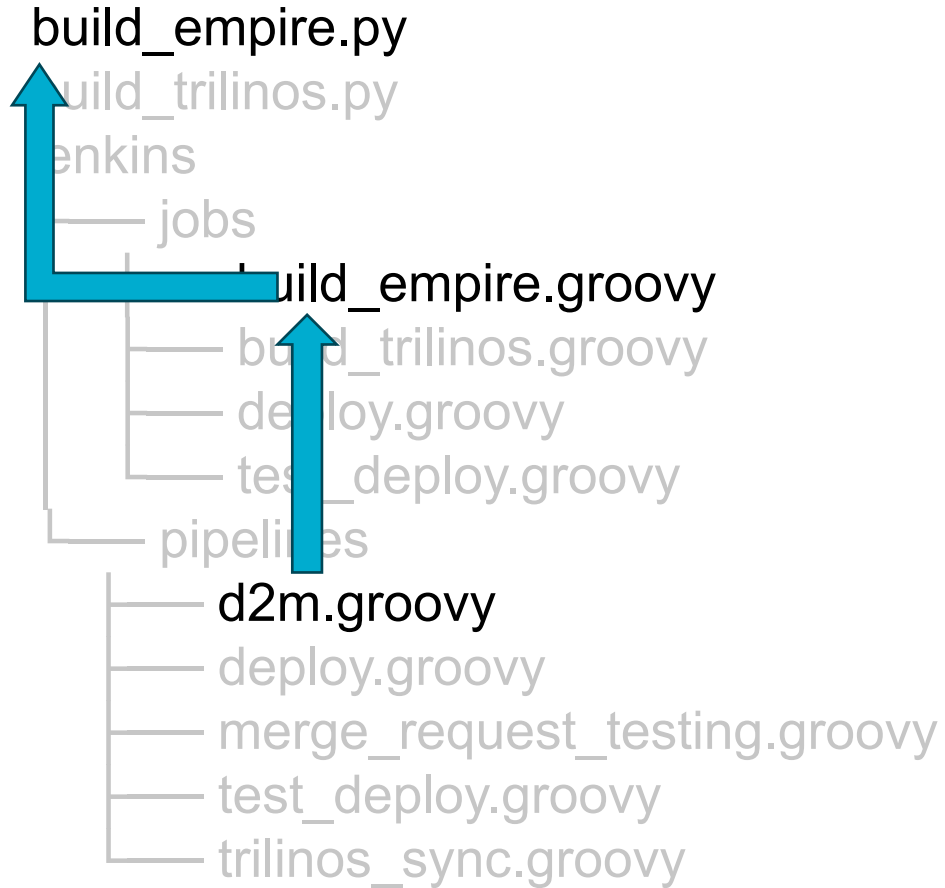
deploy.groovy

merge_request_testing.groovy

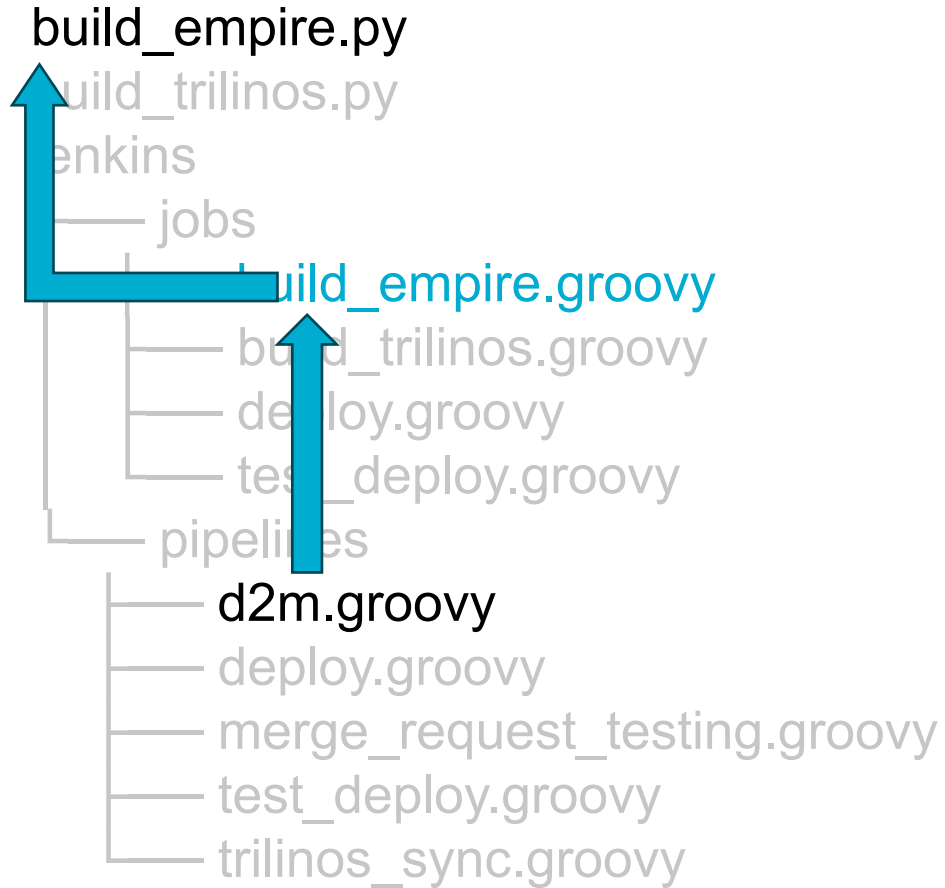
test_deploy.groovy

trilinos_sync.groovy

BUILDSCRIPTS AND JENKINS IN CONTEXT

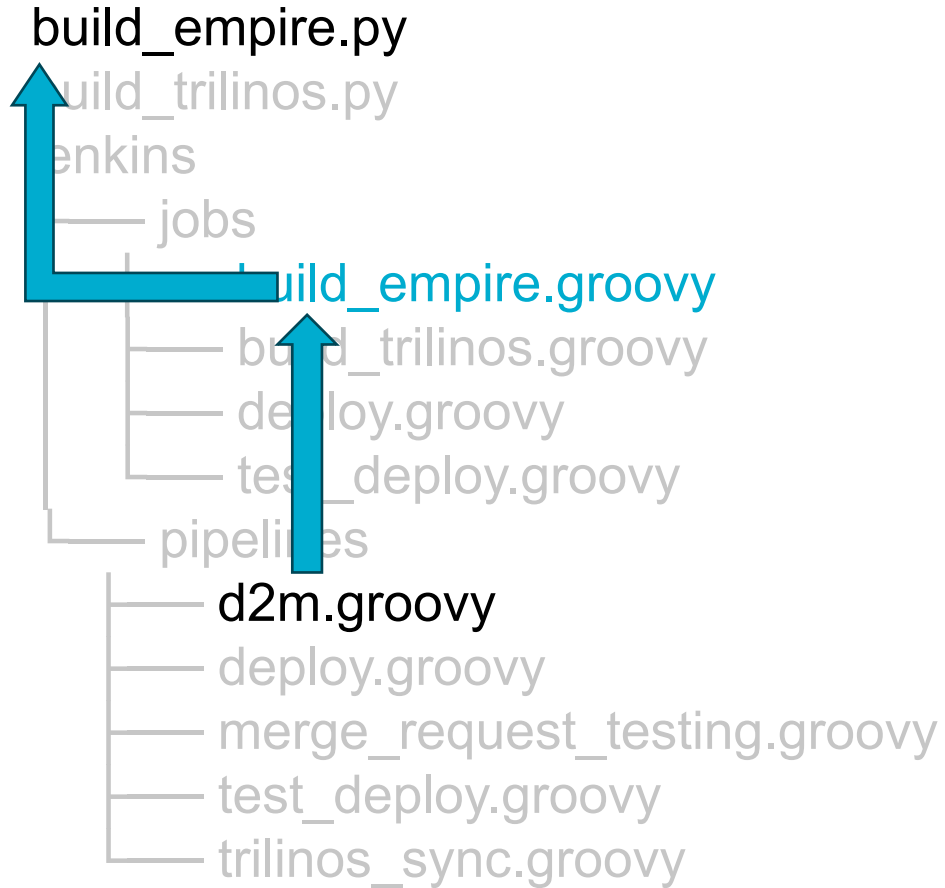


BUILDSRIPTS AND JENKINS IN CONTEXT



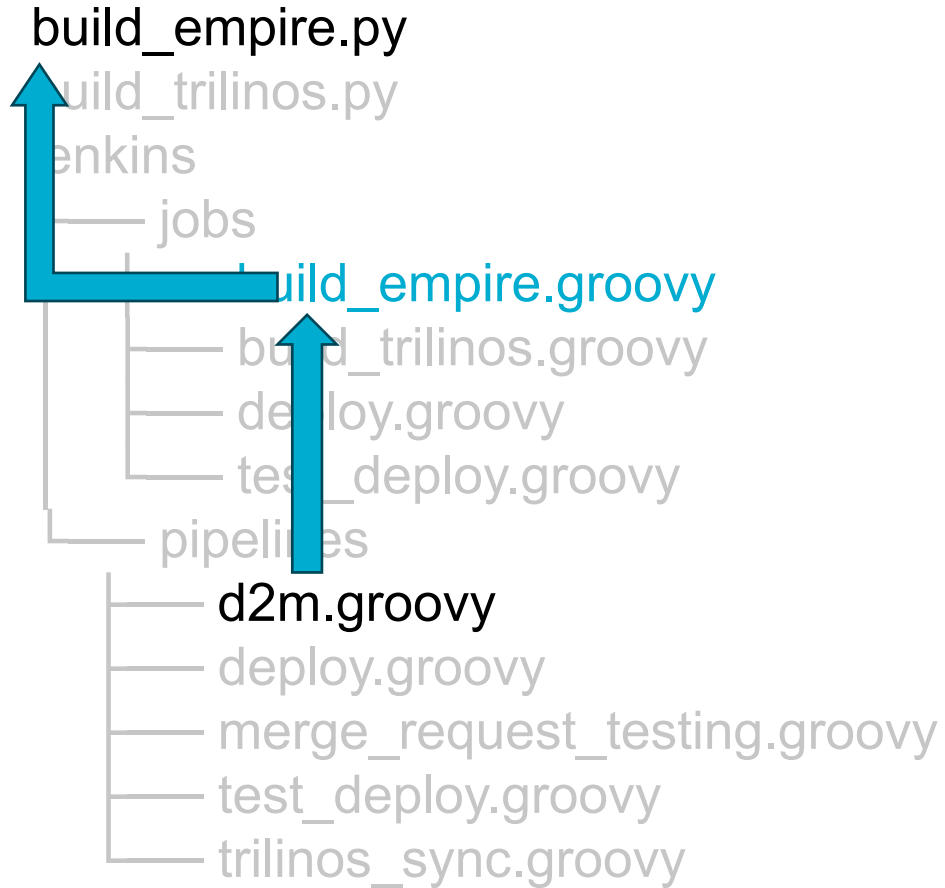
1. JOB_NAME=Clone-EMPIRE-gnu-opt-openmp-static
2. JOB_NAME=Configure-EMPIRE-gnu-opt-openmp-static
3. JOB_NAME=Build-EMPIRE-gnu-opt-openmp-static
4. JOB_NAME=ctest-EMPIRE-gnu-opt-openmp-static
5. JOB_NAME=Install-EMPIRE-gnu-opt-openmp-static

BUILDSRIPTS AND JENKINS IN CONTEXT



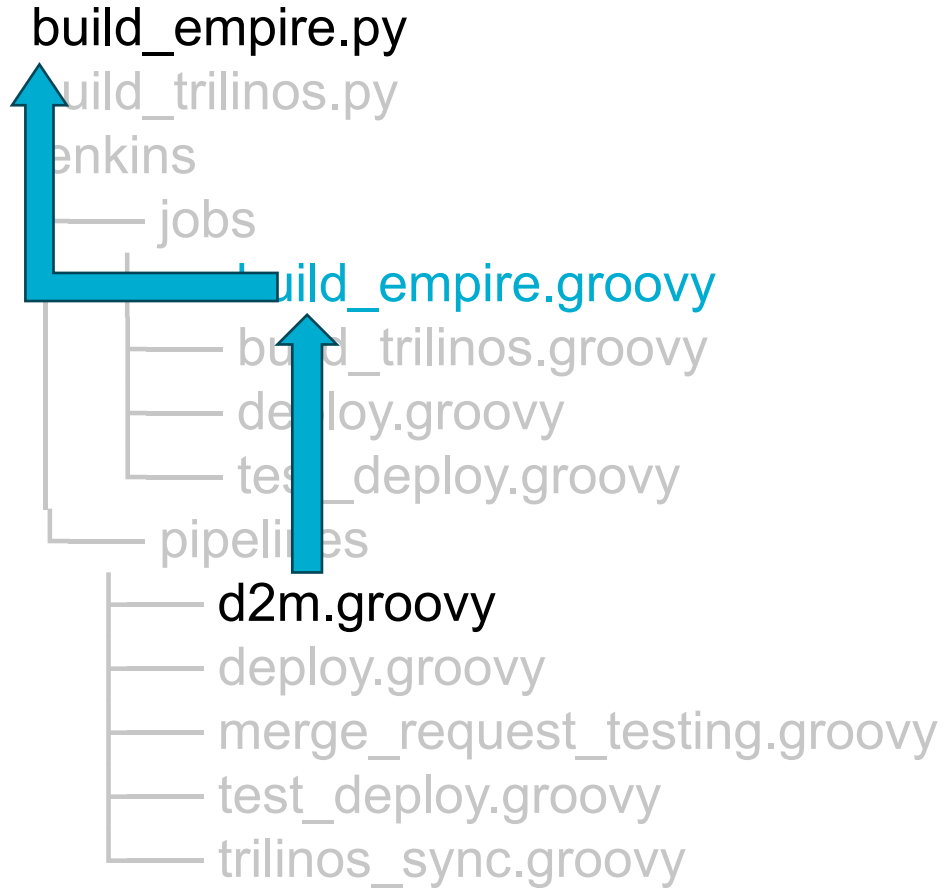
1. JOB_NAME=Clone-EMPIRE-gnu-opt-openmp-static
2. JOB_NAME=Configure-EMPIRE-gnu-opt-openmp-static
3. JOB_NAME=Build-EMPIRE-gnu-opt-openmp-static
4. JOB_NAME=ctest-EMPIRE-gnu-opt-openmp-static
5. JOB_NAME=Install-EMPIRE-gnu-opt-openmp-static

BUILDSCRIPTS AND JENKINS IN CONTEXT



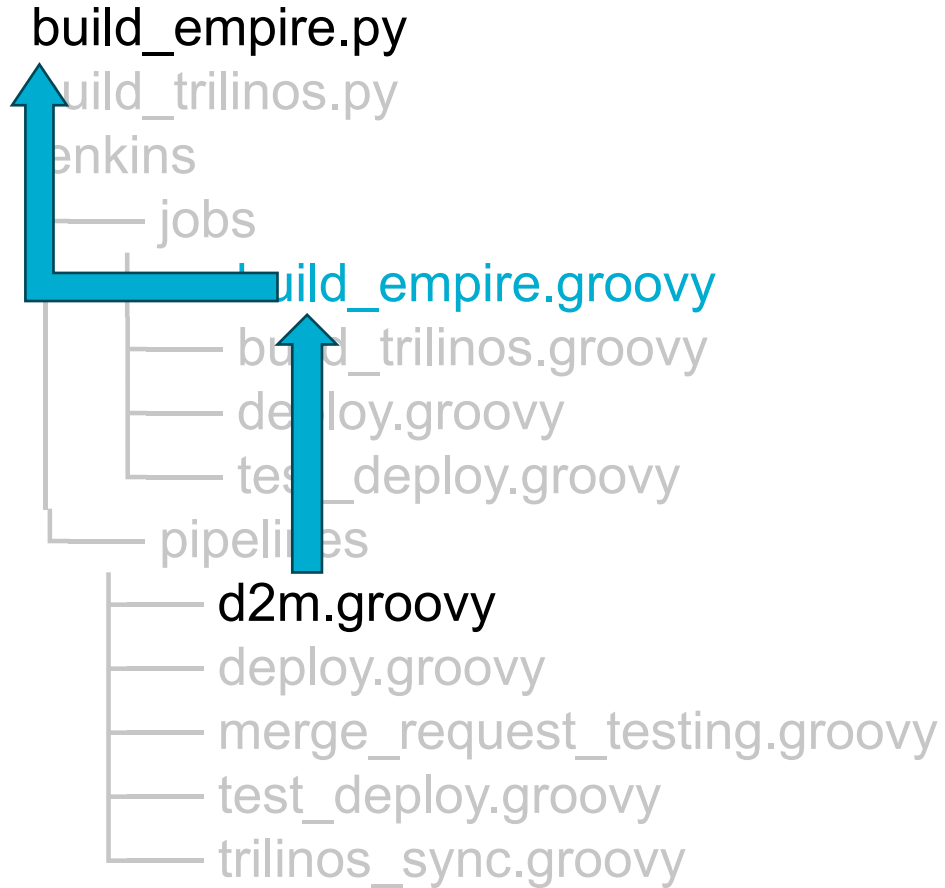
1. JOB_NAME=Clone-EMPIRE-gnu-opt-openmp-static
2. JOB_NAME=Configure-EMPIRE-gnu-opt-openmp-static
3. JOB_NAME=Build-EMPIRE-gnu-opt-openmp-static
4. JOB_NAME=ctest-EMPIRE-gnu-opt-openmp-static
5. JOB_NAME=Install-EMPIRE-gnu-opt-openmp-static

BUILDSRIPTS AND JENKINS IN CONTEXT



1. JOB_NAME=Clone-EMPIRE-gnu-opt-openmp-static
2. JOB_NAME=Configure-EMPIRE-gnu-opt-openmp-static
3. JOB_NAME=Build-EMPIRE-gnu-opt-openmp-static
4. JOB_NAME=ctest-EMPIRE-gnu-opt-openmp-static
5. JOB_NAME=Install-EMPIRE-gnu-opt-openmp-static

BUILDSCRIPTS AND JENKINS IN CONTEXT



1. JOB_NAME=Clone-EMPIRE-gnu-opt-openmp-static
2. JOB_NAME=Configure-EMPIRE-gnu-opt-openmp-static
3. JOB_NAME=Build-EMPIRE-gnu-opt-openmp-static
4. JOB_NAME=ctest-EMPIRE-gnu-opt-openmp-static
5. JOB_NAME=Install-EMPIRE-gnu-opt-openmp-static

ALL THE RESPONSIBILITIES



- Same script in does many different things, depending on **JOB_NAME**
 - Compiler
 - Build type
 - Library type
 - OpenMP/CUDA/Serial
 - Stages
 - Clone
 - Configure (CMake)
 - Build (Ninja)
 - Test
 - Install



.INI CONFIGURATION

.INI CONFIGURATIONS



- Per-hostname (regex)
 - Advantage: uniform HPC builds
 - Advantage: can give to Trilinos for troubleshooting
 - Disadvantage: fixed set of hostnames
- Assumed fixed third party library installs
 - Must provide paths ahead of time
- Include directives
 - Purpose: deduplication (less copy/paste)
 - Problem: following include after include (after include ...)
- Replaced standard environments/configurations in Trilinos repository

.INI ILLUSTRATION



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

.INI ILLUSTRATION (2)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

Full build Name

```
[rhel7\_sems-gnu-8.3.0-openmpi-1.10.1-openmp\_release-debug\_static\_no-kokkos-arch\_no-asan\_no-complex\_no-fpic\_mpi\_no-pt\_no-rdc\_no-uvm\_deprecated-on\_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

.INI ILLUSTRATION (3)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-
enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

Let's drill down a bit on this include (use) directive

.INI ILLUSTRATION (4)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO] Included section includes other sections

```
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

.INI ILLUSTRATION (5)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

Let's drill down a bit on this include (use) directive

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

.INI ILLUSTRATION (6)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

Drill down all the way to [COMMON]

.INI ILLUSTRATION (6)



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

Drill down all the way to [COMMON]

.INI ILLUSTRATION REMARK



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

Not shown: all this is scattered throughout thousands of lines of config

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-kokkos-arch_no-asan_no-complex_no-fpic_mpi_no-pt_no-rdc_no-uvm_deprecated-on_no-package-enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```

.INI ILLUSTRATION REMARK



```
[COMMON]
opt-set-cmake-var CMAKE_GENERATOR STRING : Ninja
opt-set-cmake-var Trilinos_PARALLEL_LINK_JOBS_LIMIT STRING : 8
opt-set-cmake-var Trilinos_SHOW_GIT_COMMIT_PARENTS BOOL : ON
```

...

Not shown: all this is scattered throughout thousands of lines of config

```
[RHEL7]
use COMMON
opt-set-cmake-var SEACASaprepro_lib_aprepro_lib_array_test_DISABLE BOOL : ON
```

...

```
[RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO]
use RHEL7
use USE-ASAN|NO
use USE-FPIC|NO
use USE-MPI|YES
use USE-PT|NO
```

...

```
[rhel7_sems-gnu-8.3.0-openmpi-1.10.1-openmp_release-debug_static_no-k
enables]
use RHEL7_SEMS_COMPILER|GNU
use NODE-TYPE|OPENMP
use BUILD-TYPE|RELEASE-DEBUG
use RHEL7_SEMS_LIB-TYPE|STATIC
use KOKKOS-ARCH|NO-KOKKOS-ARCH
use RHEL7_SEMS_USE-ASAN|NO_USE-FPIC|NO_USE-MPI|YES_USE-PT|NO
use USE-COMPLEX|NO
use USE-RDC|NO
use USE-UVM|NO
use USE-DEPRECATED|YES
use PACKAGE-ENABLES|NO-PACKAGE-ENABLES
use COMMON
```



DISCUSSION

ISSUES



- Rigid
 - Breaks single responsibility and open-closed principles of SOLID
 - Difficult to change .ini
- Hidden logic
 - Despite .ini, lots of environment manipulation in python scripts
- Organic Growth
 - Still here; in a different form
 - .ini files (include directives)
 - Jenkins infrastructure grew unmaintainable
 - JOB_NAME: same script, many roles
- Abstractions hide workflow complexity

ORIGINAL SOLUTION IMPACT



- Positive
 - Single source of truth for build/test workflows
 - Unified workflow across platforms
 - Largely consistent builds
 - Lots of logging
 - .ini configurations also used by Trilinos
 - `build_trilinos.py` and `build_empire.py` designed for users and Jenkins alike
- Negative
 - Highly rigid
 - Makes highly non-portable binaries
 - Nobody wants to touch BuildScripts
 - Adapting for other codes is difficult
 - Developer dependency on BuildScripts
 - Reproducing BuildScripts workflows by hand difficult

LESSONS LEARNED



- The real advantage: one workflow
 - One script to “rule them all” does not scale.
 - All of the aforementioned benefits from one workflow
 - Consistent builds, workflows, etc.
 - Single source of truth for the
- What else we can do
 - Single responsibility
 - Open-closed design
 - Flat configuration (.json)
 - Allow dynamic location/building of libraries
 - Functions should have the same general behavior, regardless of environment

SUMMARY

- One workflow > one script
- Follow SOLID principles
- Flat configuration (no include directives)
- Try to avoid per-hostname behavior
- Don't hide complexities that developers can fix
- Don't drastically change function behavior with things like environment variables



QUESTIONS?

