

CERN SFT Nightlies Publication via CVMFS Gateway Publication

Tim Ehmann – CERN EP-SFT

16/09/24

Outline

- Who we are
- Problem
- Implementation
- Results



Who we are



Who we are

- SPI Project of the CERN EP-SFT Department
- The SPI team provides comprehensive software stacks for LHC experiments and CERN users
- Stacks encompass Monte Carlo generators, Machine Learning tools, and Python modules.
- These stacks support various compilers, operating systems, and hardware architectures.
- Stacks are created and uploaded via a nightly Jenkins pipeline



Problem

Problem

Previous nightly pipeline

- we currently support 56 combinations (Stack x Arch x OS x Compiler x Buildtype) over 10 different stacks
- CVMFSInstall Stage takes ~17minutes and the queue usually exceeds 5 hours
- finishes at the end of the day depending on the amount of currently supported combinations
- taking too long to install all packages and create the views for all combinations

“

Unfortunately, today's nightly CVMFS installation directory at `/cvmfs/sft-nightlies.cern.ch/lcg/nightlies/dev3python311/Fri/` is empty - is this an accident ?

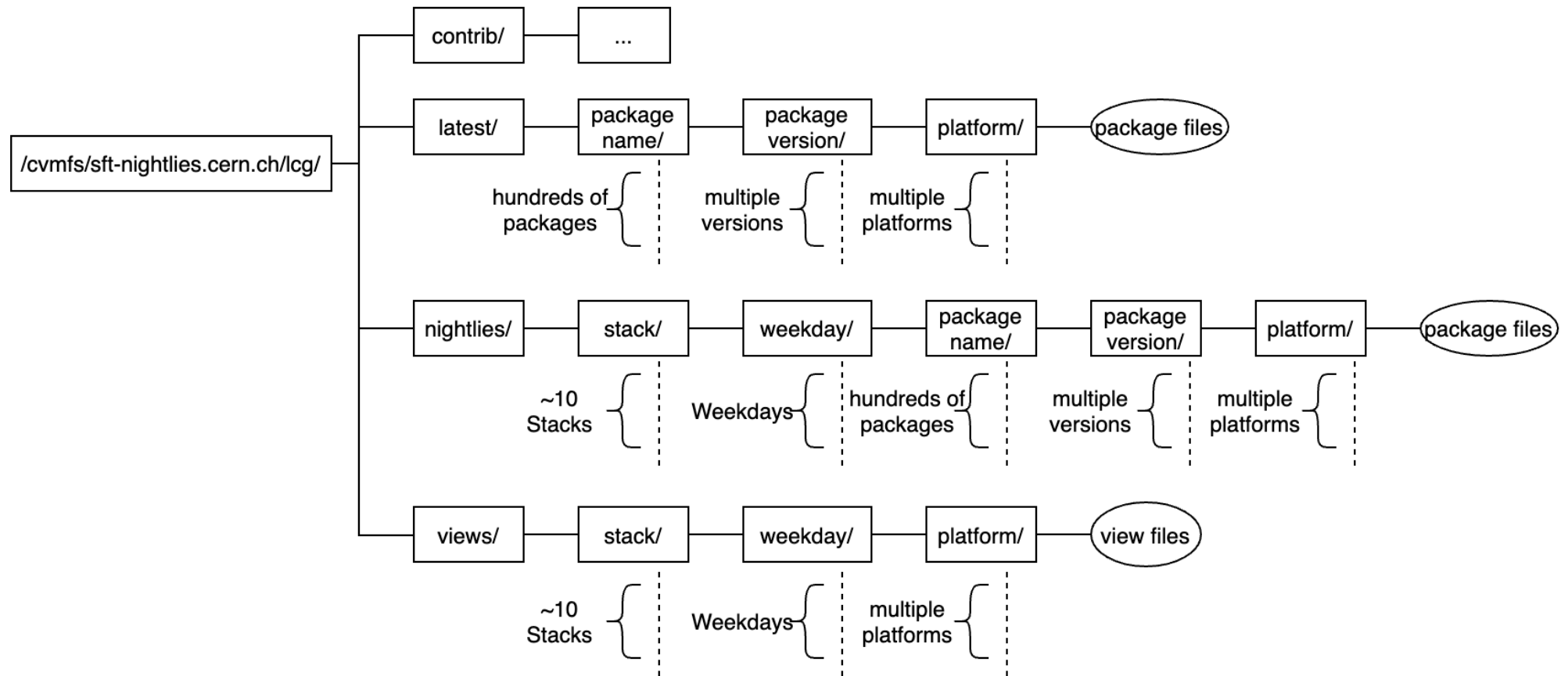
- Johannes Elmsheuser (Atlas)

Problem

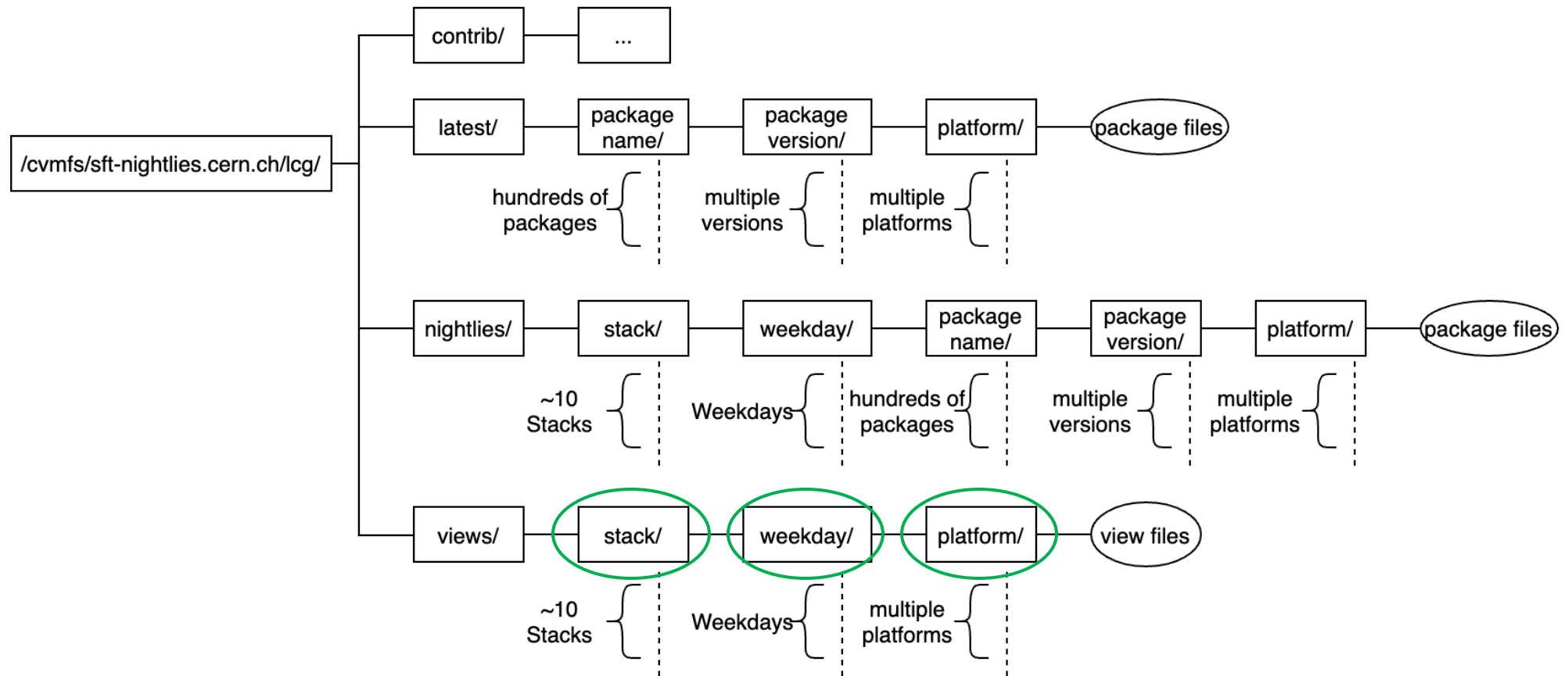
Solution: Parallelization

- allows faster installation times by processing the installation stage of different pipelines in parallel
- enables horizontal scaling by adding more publisher machines

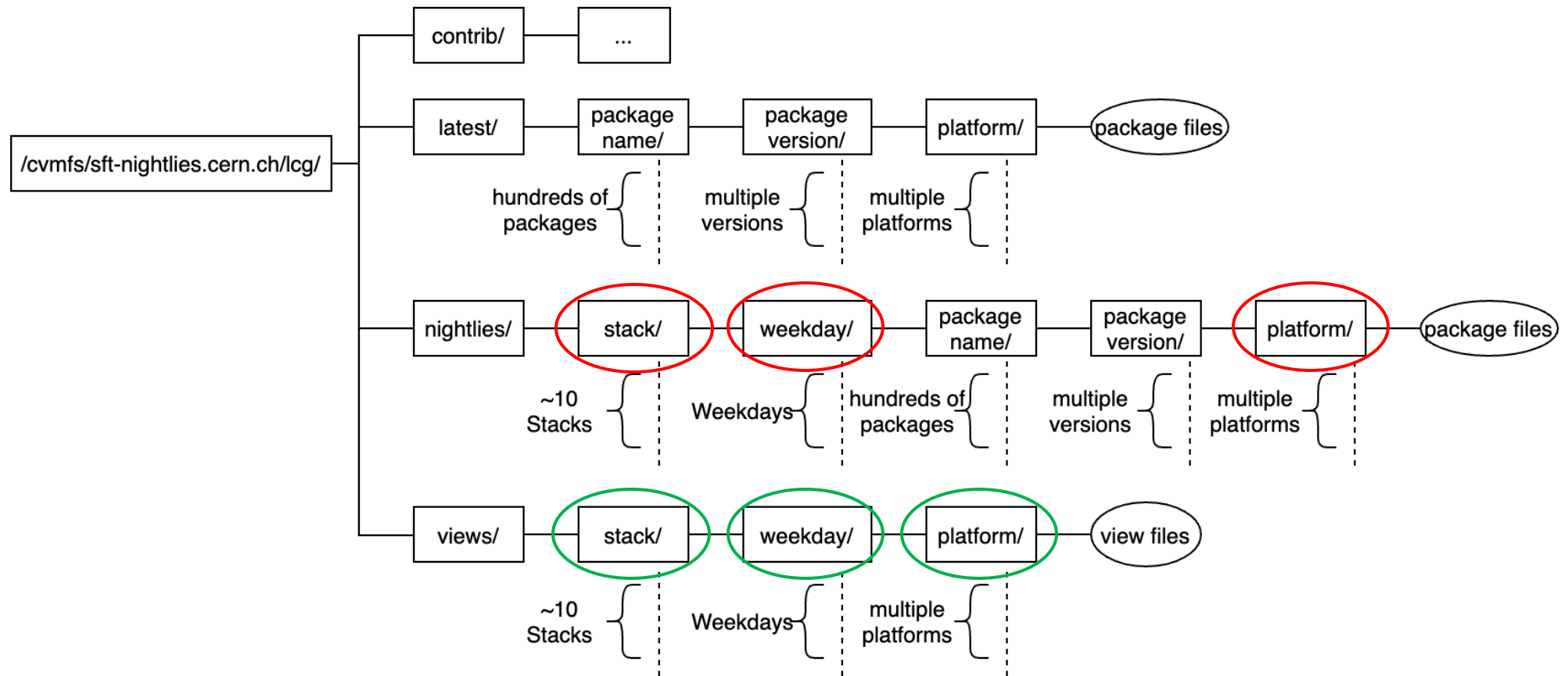
Current Directory Structure



Current Directory Structure



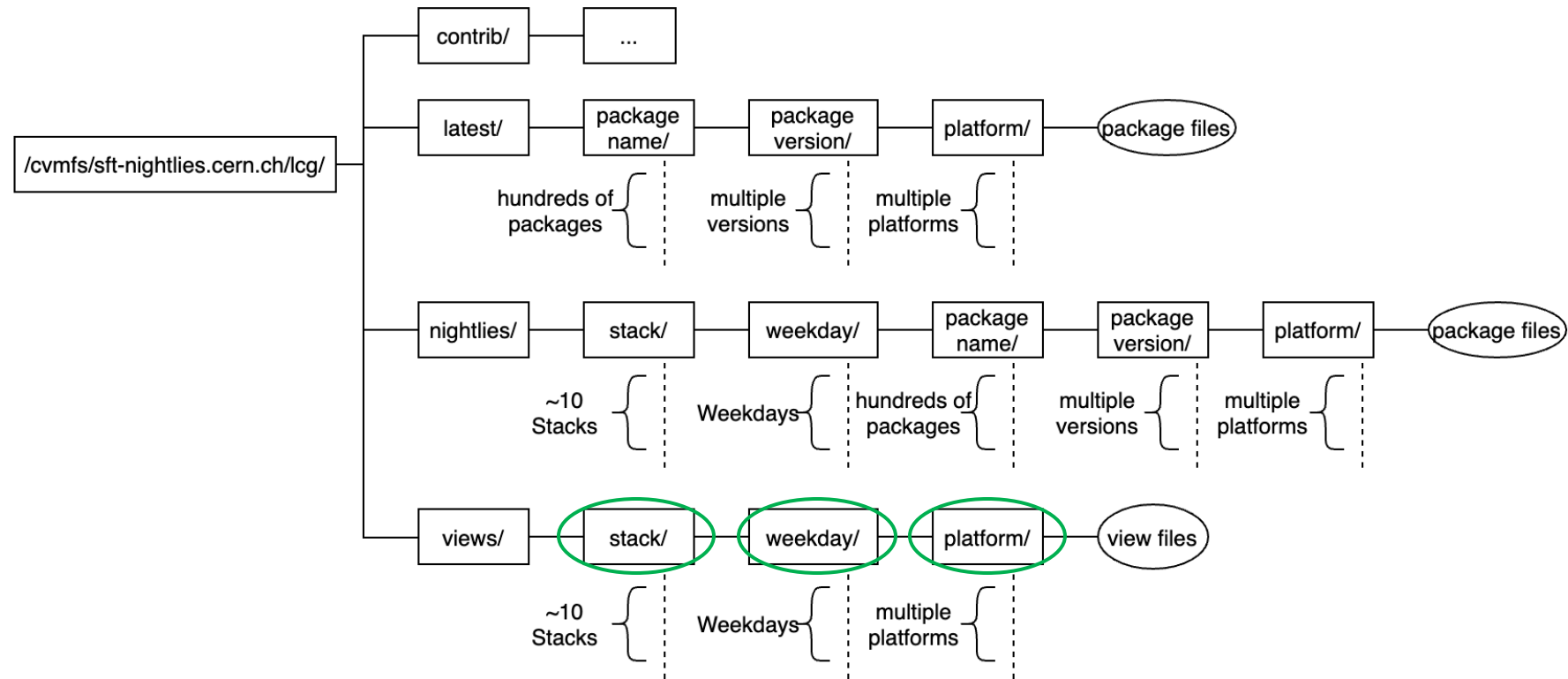
Current Directory Structure





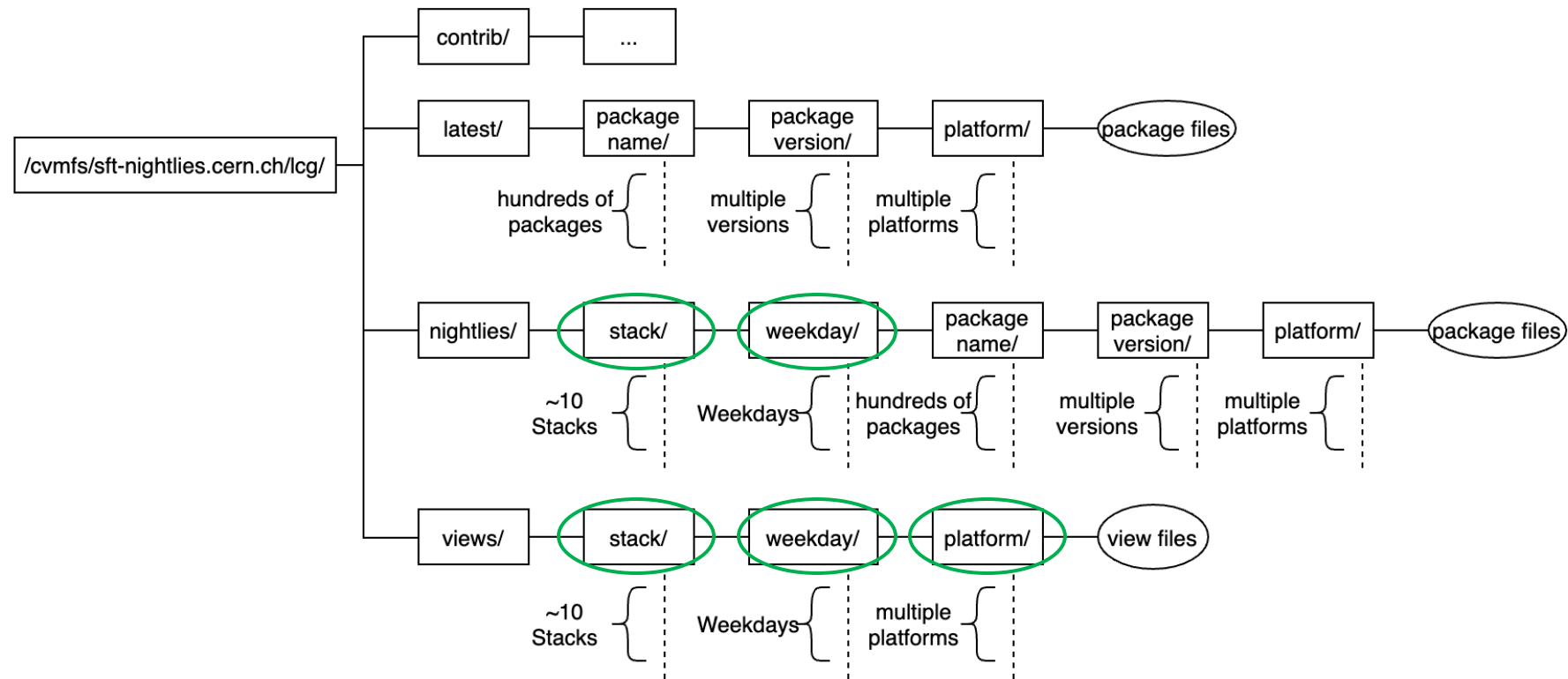
Implementation

Implementation



- Parallelisation for view creation

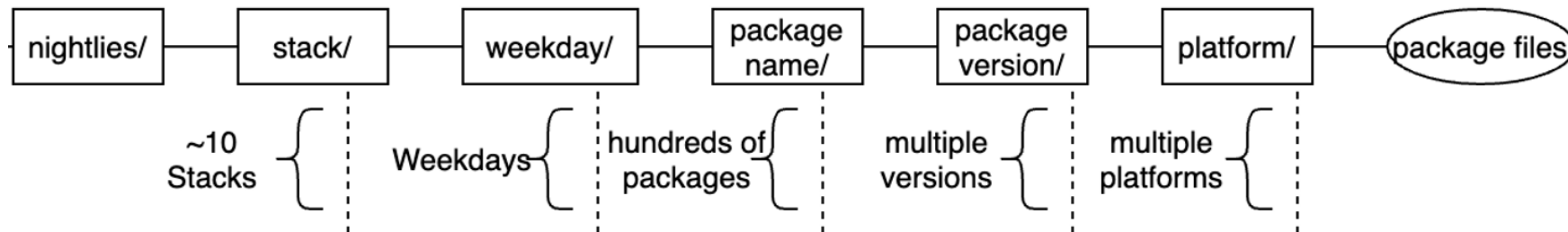
Implementation



- Parallelisation for view creation
- Parallelisation for installation in nightlies between different stacks (dev3, dev4, dev3cuda, dev4cuda, ...)

Implementation

- CVMFS Transaction Parallelization is handled by two main tools:
 - Jenkins Lockable Resources Plugin
 - CVMFS Leases
- Jenkins Resources prevent the allocation of machines to non-parallelizable combinations and manage the sequence of transactions



- `dev3lhcb-aarch64-alma9-gcc13-opt` → running on machine A
- `dev4-x86_64-alma9-gcc11-opt` → running on machine B
- `dev4-aarch64-alma9-gcc11-opt` → doesn't start running on machine C
- `dev4lhcb-x86_64-alma9-gcc13-dbg` → starts running on machine C

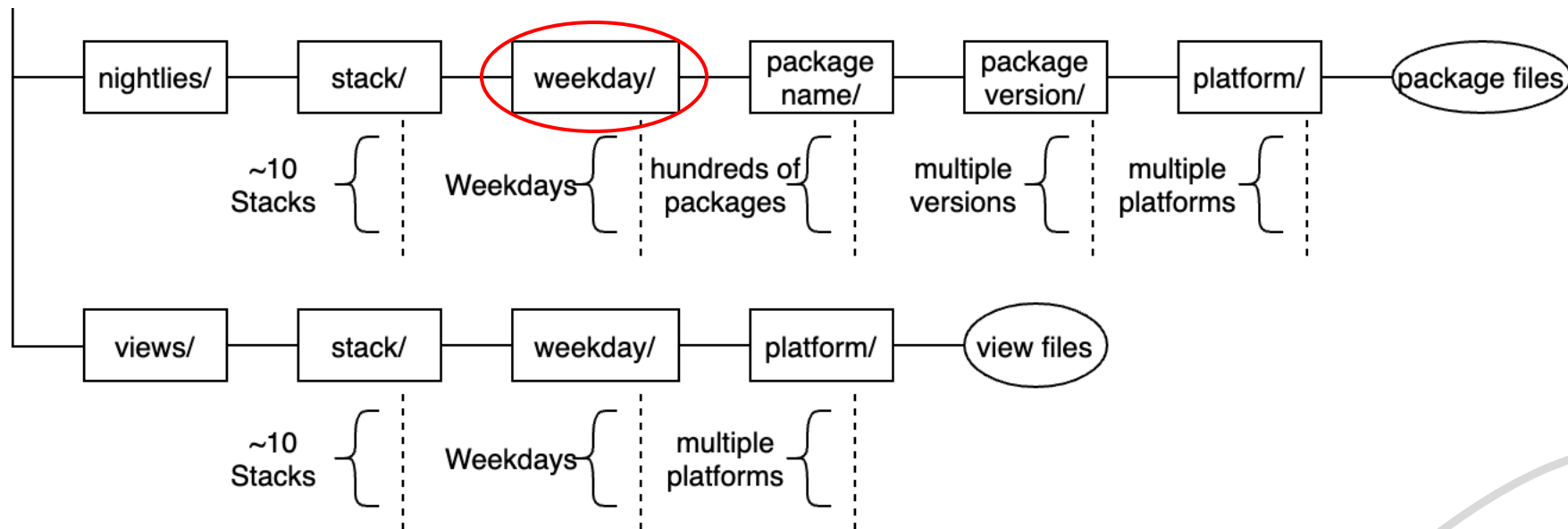
Implementation

Labels	Assigned resources	Free	Free in %
cvmfs-nightlies-publisher	4	4	100

- Created a resource with the Lockable Resources Jenkins Plugin
 - Contains 4 resources for the 4 used machines
- For parallelizable transactions, 1 resource is "locked"
- For non-parallelizable transactions, all 4 resources are "locked"
- For parallelization on the stack level, a lock is created that only exists during runtime

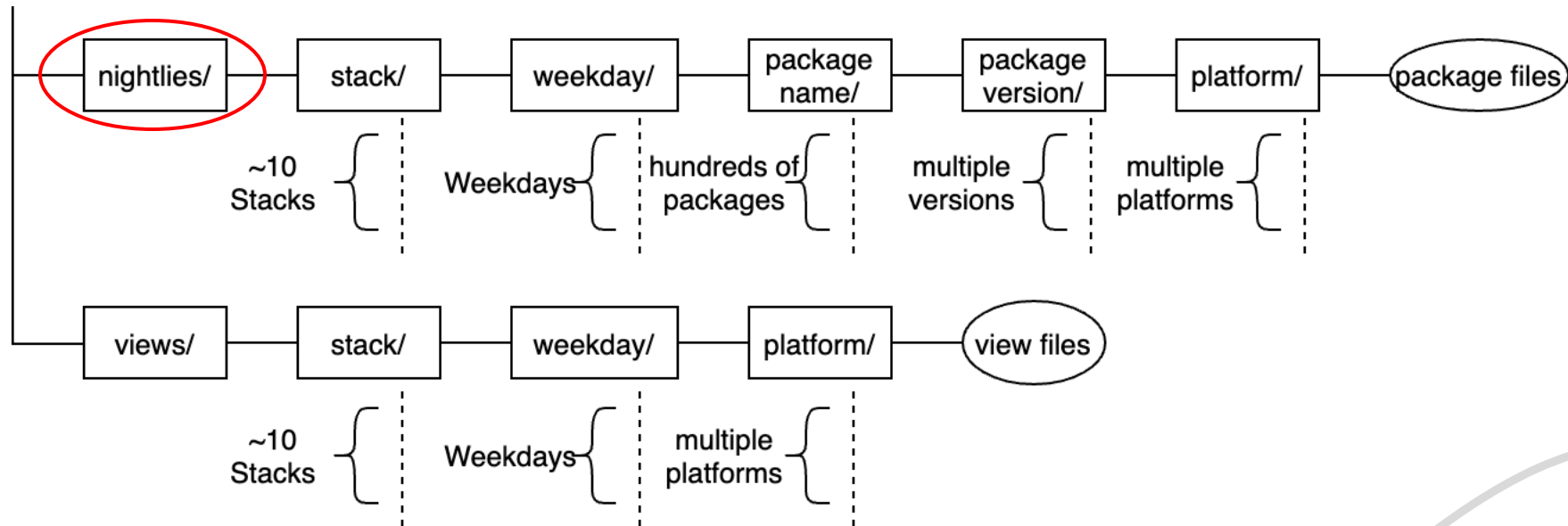
Implementation

- CVMFS Transactions are handled via leases
- The transaction always leases the deepest path possible that's required for the transaction



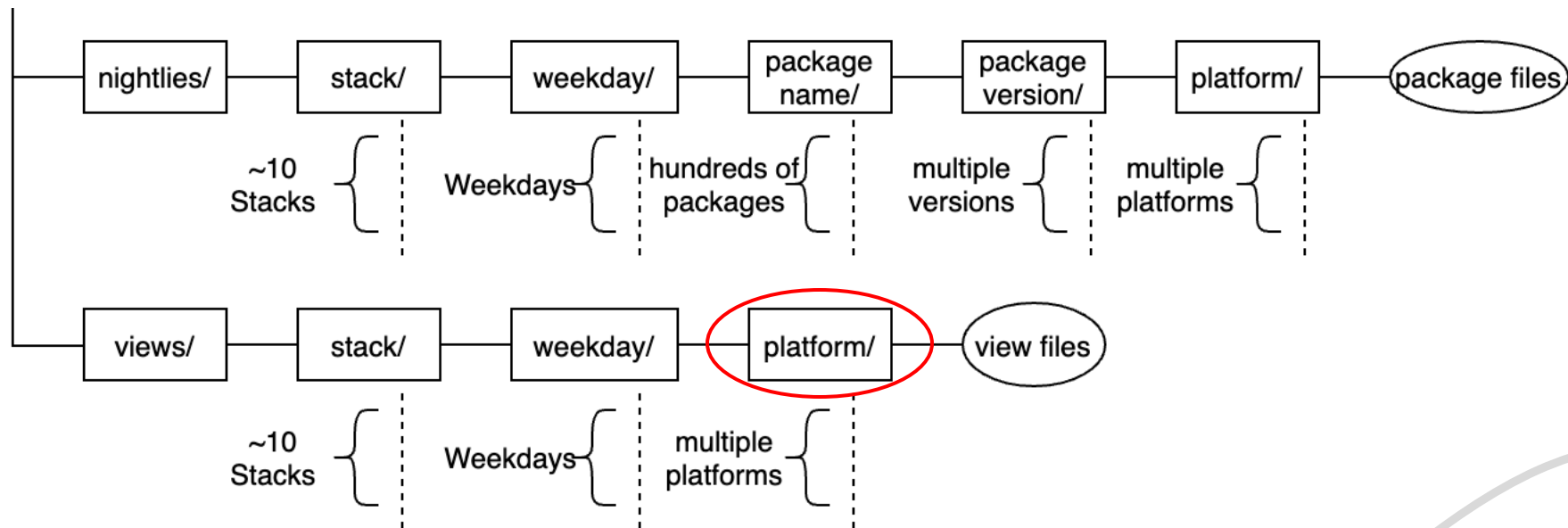
Implementation

- CVMFS Transactions are handled via leases
- The transaction always leases the deepest path possible that's required for the transaction



Implementation

- CVMFS Transactions are handled via leases
- The transaction always leases the deepest path possible that's required for the transaction



Implementation - Summary

- Split the CVMFS Installation into package files installation and view creation/installation
 - Package files installation only allows parallelization over different stacks, not different platforms in one stack
 - View creation allows full parallelization
- Parallelization management with CVMFS Leases and Jenkins Lockable Resources Plugin



Results

Results

- Full Nightly Pipeline runtime for all combinations is down to on average 6h17min over the past two weeks down from ~15 hours

Runtime of full nightly pipeline over the past 2 weeks

