EESSI meeting

July 1st 2021

https://github.com/EESSI/meetings/wiki

Agenda

1. Quick introduction by new people
2. EESSI-related meetings in last month [Bob, Kenneth]
3. Progress update per EESSI layer [Bob, Peter, Kenneth, Alan, Caspar]
4. 2021.06 version of pilot repository [Bob, Kenneth]
5. AWS/Azure update [Kenneth, Bob, Terje, Henk-Jan, Laura]
6. Infrastructure in AWS [Terje]
7. Progress towards automating deployment of software installations [Kenneth, Bob, Terje]
8. Q&A

Quick introduction by new people

New people on the call: feel free to introduce yourself!

- Who are you, where do you work, on what?
- Why are you interested in the EESSI project?
- Are you planning to actively contribute,
  and if so, to which aspect(s) of the project?

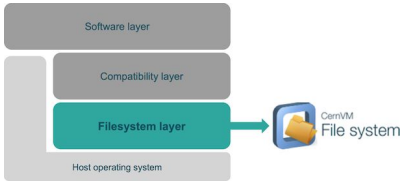
EESSI-related meetings

- June 8th: Monthly CernVM-FS coordination meeting (Kenneth)
  - Extra manpower coming to CernVM-FS dev team
  - “Some serious sponsoring from industry”

- June 22 + June 29: EESSI automated software deployment sprint (Bob, Kenneth, Terje)
  - See dedicated slide

[Bob, Kenneth]

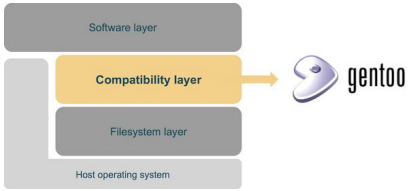
Progress update: filesystem layer



- New client and build containers moved from Docker Hub to GitHub Container Registry:
  - https://github.com/orgs/EESSI/packages + https://github.com/EESSI/filesystem-layer/tree/main/containers
  - `client-pilot:centos7` -> standard client image (also includes fuse-overlayfs)
  - `build-node:debian10` -> software build image
  - Multi-arch image (one name for all architectures), automatically built via Github Action workflow
  - All documentation and README files should have been updated
  - New PR for adding some minimal testing of the images (PR #88)
- Bob finally fetched the yubikey from the office…
  - Still need to figure out how this works
  - Then we need to plan a date to regenerate the keys, and provide new public keys

[Bob, Kenneth]

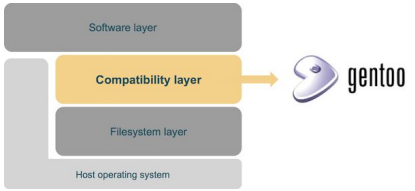
Progress update: compatibility layer



- GitHub Action to automatically build multi-arch `bootstrap-prefix` container images
  - Similar to the ones from filesystem layer
- Lots of PRs merged for a new version of the compatibility layer:
  - Updated bootstrap script and Gentoo snapshots
  - New package set for 2021.06 with updated packages (now also including ReFrame)
  - Use git instead of web-rsync for syncing the Gentoo overlay (should be much faster)
  - Run ReFrame test suite at the end of the playbook (see test/compat_layer.py)
  - Fix timezone issue by making a symlink to the host
  - Minor improvements to playbook
  - Improved mechanism for injecting host libraries (e.g. NVIDIA drivers, or anything in compat layer)
- Small issue with 2021.06 compat layers w.r.t. default GCC (no LTO support)
  - Exposed a bug in EasyBuild when building GCC, easily fixed (see PR #2498)

[Bob]

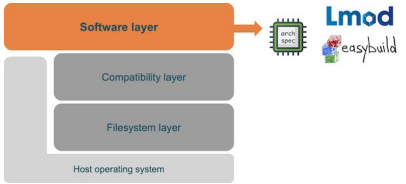
Progress update: compatibility layer



- Issue with failing checks for the Github Action that tests our installation playbook
  - Depends on a Prefix container built by 3rd party, but hasn't been updated in a while...
  - Bootstrapping Gentoo Prefix from scratch in an Github Action takes (way) too long…
  - We need to provide something ourselves to make the tests more reliable
    - ➢ Gentoo Prefix container image
    - ➢ Gentoo Prefix in a test repository on CVMFS (`ci.eessi-hpc.org`?)

[Bob]

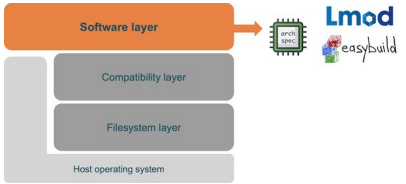
Progress update: software layer



- Using new container images, pulled from GitHub container registry (see docs)
- Build script updated for 2021.06 pilot + EasyBuild v4.4.0 (see PR #119)
  - Incl. update to ReFrame 3.6.2
  - EasyBuild v4.4.1 will be released really soon...
- Enhancements to EasyBuild hooks used for EESSI by Alan (see PR#116)
  - Inject path into RPATH section that can be used to override MPI libraries
  - Via `/cvmfs/pilot.eessi-hpc.org/host_injections` + CernVM-FS variant symlink
  - `/opt/eessi -> /cvmfs/pilot.eessi-hpc.org/host_injections` (see issue #89)
- Reworked GROMACS test by Caspar (see PR #115)
- Scripts for automating ingestion of software installations from build node to Stratum-0

[Kenneth]

Progress update: software layer

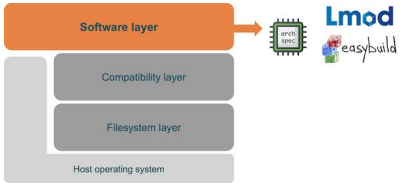


- Injection of alternative MPI:
  - Machinery in place (override RPATH when MPI is in toolchain of software)
    - Variable symlink to host location to control what is picked up
    - Need to also override `bin`, but can be done with a system MPI module + PATH priority
    - Need script to create required directory structure (and do sanity checks)
  - Requires ABI compatibility with MPI library used in software layer
  - Currently being tested by Alan...

- CUDA support:
  - Seems possible to upgrade CUDA drivers in user space, EESSI CUDA packages would then only require minimum GPU driver version (450.80.02 for CUDA11)
  - Can be baked into our script that will make CUDA available to EESSI
  - No reply yet from NVIDIA w.r.t. meeting to discuss including CUDA in EESSI

[Alan]

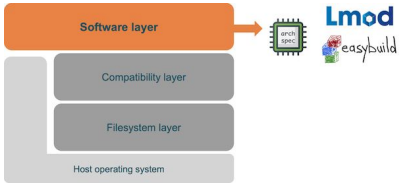
Progress update: testing



- Discussion with ReFrame devs on library-of-tests. Concept:
  - Library test: contains (only) test specific things (e.g. run command, sanity check, etc). Defines what should be set by test implementation
  - Test implementation: contains system specific things. Derives from library test, sets e.g. modules to load, num_tasks, etc
- Useful structure for EESSI, but EESSI needs even test implementation to be ‘generic’
  - Pro: library tests can be reused (ReFrame community benefits ánd can help maintain)
  - Pro: sites (e.g. Amazon) could create own test implementations instead of generic EESSI ones

[Caspar]

Progress update: testing



- PR with example (thanks @Javier Otero for excellent feedback): https://github.com/EESSI/software-layer/pull/115/files
- Requires:
  - Users to specify (#) GPU devices and #cores in ReFrame config
  - EasyBuild (flat) module naming scheme
  - Other than that, should run out of the box!

[Caspar]

EESSI pilot repository

https://eessi.github.io/docs/pilot

NOT FOR PRODUCTION USE!

2021.06 version of pilot software stack

Current status:

- Compatibility layer in place for `x86_64` + `aarch64` + **`ppc64le`** 🎉
- May still need some very minor tweaks (like fixing issue with default GCC in Gentoo Prefix env)

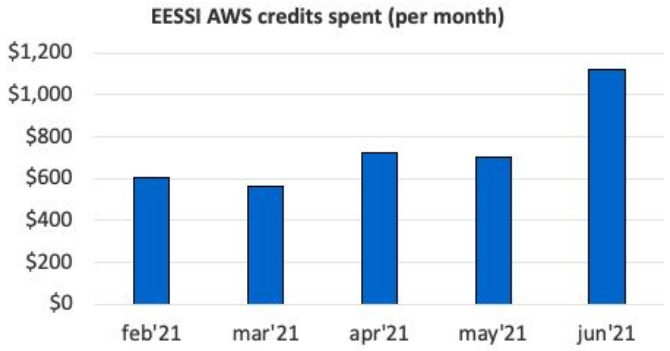
To do:

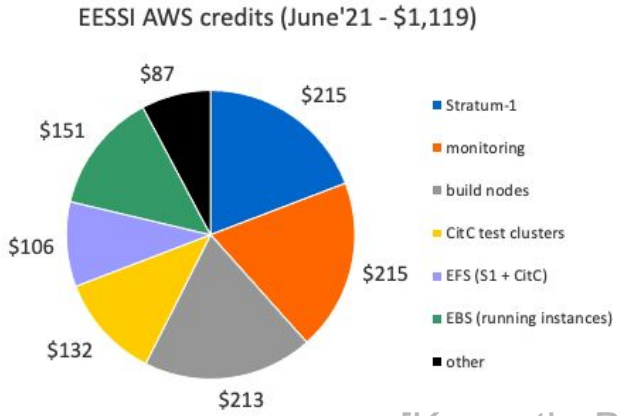
- Install software (work-in-progress)
  - Target CPUs:
    - `{aarch64,x86_64,ppc64le}/generic`
    - `intel/{haswell,skylake_avx512}`, `amd/zen2`, `aarch64/graviton2`, `ppc64le/power9le`
  - Bioconductor (R), GROMACS, OpenFOAM, TensorFlow, Spark, IPython, Horovod, QuantumESPRESSO, …
- Provide init scripts
- GPU installations: on hold (cfr. discussion with NVIDIA on CUDA)

[Bob, Kenneth]

Usage of sponsored AWS credits

- Sponsored credits ($25,000) are being put to good use!
- **Ask in #aws-resources Slack channel to get access!**
- In June ‘21: ~$1,119 worth of credits spent
- on Stratum-1, monitoring node, build nodes, EasyBuild testing, ...
- ~$3,700 worth of credits spent in total





[Kenneth, Bob, Terje]

Azure sponsorship

- Progress to date (a personal perspective)
  - Background - strategic meeting Oct 2020
  - EESSI + Azure
    - Budget - $40,000 Azure credits (via SURF)
    - Timelines...and best shot!
    - Governance, project management etc.
- Next steps
  - The work itself
  - Key Performance Indicators (KPIs)
  - Long term goals, further opportunities & collaboration

[Laura]

AWS infrastructure: status

- Updated images with gh, ansible, awscli, and the tarball upload script...
- Each generated node gets IAM-bound to allow for uploading to the S3 staging bucket
- Increased capacity in key regions by a factor of 30 — “GROMACS ready”
- Fixed a bug where we sometimes got RHEL Beta versions for image sources in AWS
  - The image name is the same but with “BETA” *added* after HVM
- Redid alerts for AWS cost management
- Started looking at migrating lambdas to Terraform

[Terje]

AWS infrastructure: Terraform challenges

- Multi-user terraform and SSH keys is… tricky…
  - Terraform requires all files in a config to be readable to validate state. Good.
  - Terraform requires the private ssh key to be mode 0400. Also good.
  - Combining those two requirements in multi-user environment? Uhm… Not good?
- Solved via throw-away keys and a bit of shrugging
- Also set up a dedicated `terraformer` user for infrastructure management

[Terje]

Progress on automating software deployment

Long-term goals:

- **Secure automated** deployment of software installations into EESSI
- Limit human intervention as much as possible (yet require human review/approval)
- Minimal requirements for build nodes (no admin privileges required, only Singularity)
- Allow contributions / requests to add software installations (via GitHub PRs?)
- Validate by running tests in CI before deploying to EESSI repository on Stratum-0
- Evaluate impact of changes being made (like security updates in compat layer)

[Kenneth, Bob, Terje]

Progress on automating software deployment

Deployment of previous EESSI pilot repository (2021.03)

- Compat layer is built using Ansible playbook and container image to bootstrap Prefix
  - Tarball is created **manually**, ingestion is done **manually** on Stratum-0 by Bob
- Human babysits software build script on a build node
  - see https://eessi.github.io/docs/software_layer/build_nodes/
  - Controlled build environment via “build node” container image
  - Could be in own infrastructure, or cloud instance (AWS, Azure, …)
- Human **manually** creates tarball of software installations to ingest into EESSI repo
- Tarball is shipped (via scp) to Stratum-0 for ingestion
- Bob runs ingestion procedure **manually**

[Kenneth, Bob, Terje]

Progress on automating software deployment

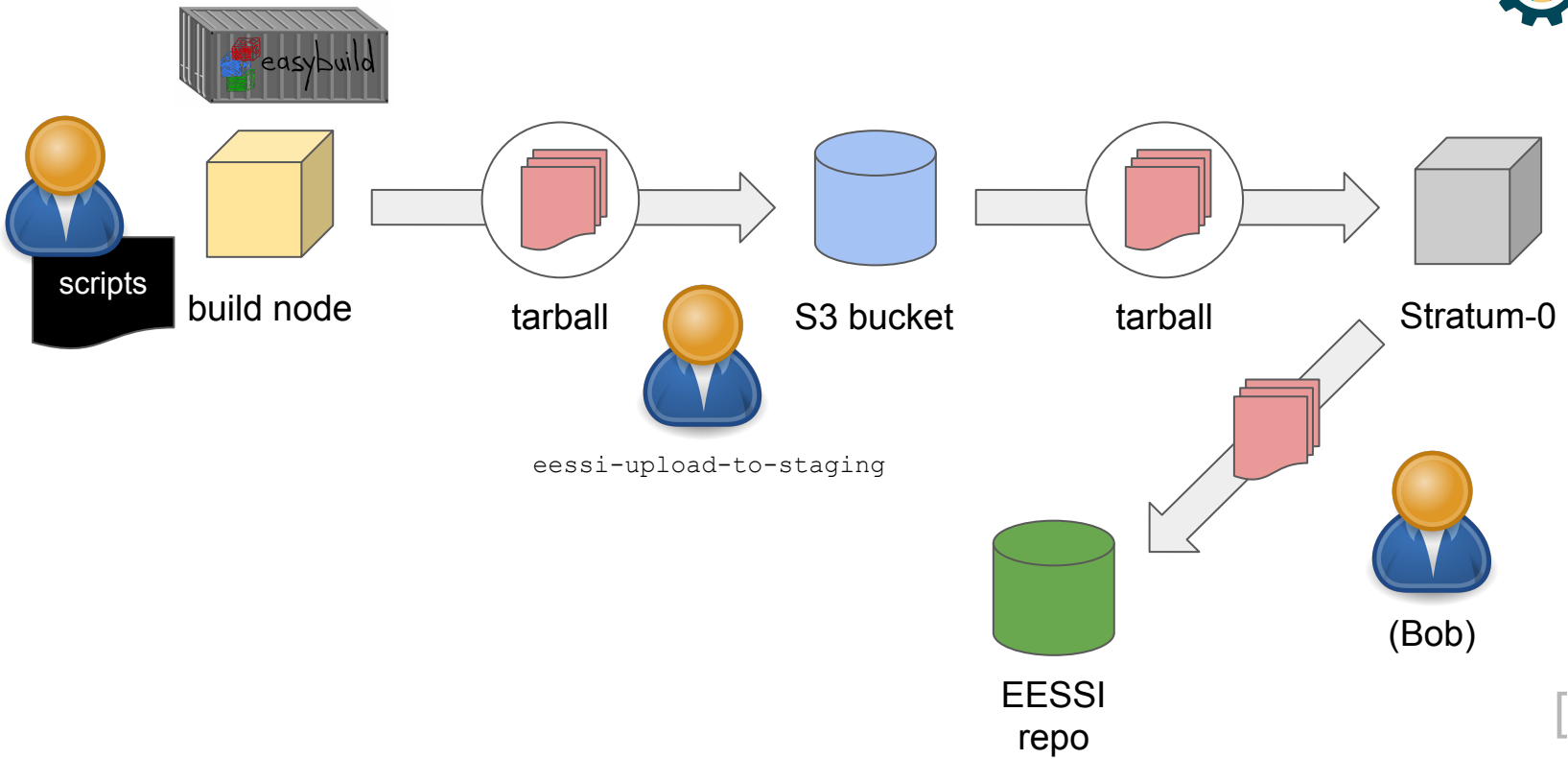
Deployment of 2021.06 EESSI pilot repository (**work-in-progress**)

- Compat layer is built using Ansible playbook and container image to bootstrap Prefix
  - Tarball is created manually, ingestion is done **semi-automatically** on Stratum-0 by Bob
- Human starts **build-and-deploy script** on a build node (see PR #120)
  - Build script takes care of software installations (see PR #119)
  - **Tarball** with only new/changed files is **created via script** (see PR #111)
  - Tarball is uploaded to **S3 bucket** in AWS (+ Slack bot reports on it in `#staging`)
- Bob ingests tarball **semi-automatically** into EESSI repo on Stratum-0
  - Soon via PRs to `EESSI/staging` GitHub repository?

[Kenneth, Bob, Terje]

Progress on automating software deployment

Current status (roughly)



[Kenneth]

Progress on automating software deployment

Goal: automated, with human oversight