Simula Grand Knowledge Festival 2026

# Automated System Installation of Virtual Machines and Computers

---

## Part 1

Thomas Dreibholz ( 托马斯博士 )
Simula Metropolitan Centre for Digital Engineering
dreibh@simula.no

June 4, 2026

# Contents

- Motivation
- The «Virtual Machine Image Builder and System Installation Scripts»
- The «System-Tools»
- How to create customised VMs/installations?

# Motivation

- A recurring task: a fresh Ubuntu/Debian/Fedora/FreeBSD installation
  - For project XYZ
  - For some testing of software
  - …
- Naïve solution:
  - Create «golden images» manually
  - Regularly update them, thoroughly document all manual steps

This does **not** work! No, really …

# Researchers have very specific requirements!

- Standard office user: web browser, e-mail, text/slides/spreadsheet
- Researcher:
  - Development tools (C/C++, Python, R, R packages, LaTeX, ...)
  - Very specific configuration: file system tuning, enhanced partitioning, advanced kernel features
  - Enhanced browser configuration: privacy, security, ad-blocker, ...
  - Special tools pre-installed and configured

**Manual install is insufficient, or time-consuming and error-prone!**

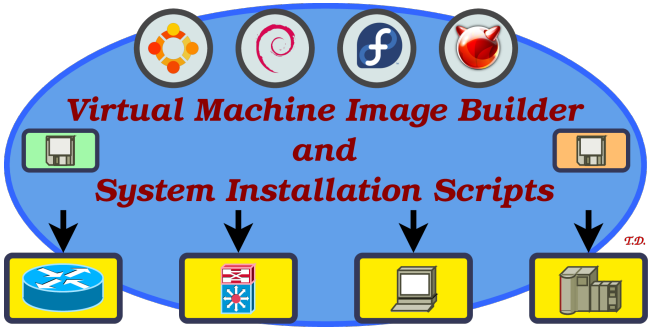
# The Proper Way: Automate the Installation

- Create scripts for each and every step
  - No manual tweaking of things!
  - Reuse already-existing steps
- Manage everything in a Git repository
  - Documentation
  - You (or somebody else) can read, debug, and adjust the steps
- Reproduce the steps easily → regularly create fresh images

A very useful idea! Has it been done already?

# Virtual Machine Image Builder and System Installation Scripts

- 14 years of Simula experience
- Used in various Simula projects, e.g.:
  - NorNet
  - NEAT
  - HENCSAT
  - 5gVINNI/RAKSHA
  - …




Homepage: https://www.nntb.no/~dreibh/vmimage-builder-scripts/

# Features

- Support of different hypervisors (for VMs/containers)
  - VirtualBox, Proxmox, QEMU (e.g. OpenStack), Docker
- Can run on a (pre-)installed system as well
- Support of multiple systems
  - Linux (Ubuntu, Debian, Fedora), FreeBSD
- Extensible

Git: https://github.com/simula/nornet-vmimage-builder-scripts

# How does it work?

- Direct installation (on current system):
  - `./make-direct <PROJECT>`
  - Executes the installation scripts
- VM/container installation:
  - `./make-<project>`
  - Using Packer (VM/container installation tool) to run a minimal installation with SSH server for login
  - Executes the installation scripts

# Projects

- Project:
  - A collection of installation scripts and corresponding files
  - Can inherit other projects
- Existing projects in “Virtual Machine Image Builder and System Installation Scripts”
  - Minimal, Basic (inherits Minimal)
  - Development (inherits Basic)
  - KDE (inherits Basic)
  - KDE+Development (inherits KDE and Development)

Starting points for your own projects!

# The «Basic» Project

- A project for command-line usage (e.g. SSH), no GUI
- Security-enhanced SSH-server configuration
- Various networking tools pre-installed
  - Ping, Traceroute, SubNetCalc, TSCTP, T-Shark, ...
  - Measurement tools: NetPerfMeter, HiPerConTracer
- Native language support (NLS): use the VM in your language
- System-Tools (details later)

https://www.nntb.no/~dreibh/vmimage-builder-scripts/#basic

# The «Development» Project

- Obviously, contains development tools
- Compilers, etc.: GCC and Clang C/C++, CMake, Valgrind, ...
- Python
- R and important R packages
- Docker
- ...

https://www.nntb.no/~dreibh/vmimage-builder-scripts/#development

# The «KDE» Project

- A fully-configured KDE desktop
- Preconfigured Firefox
  with privancy-enhanced settings, ad-blocker, etc.
- Graphics and LaTeX software (e.g. Inkscape, GIMP, Dia, Kile, ...)
- Fonts: Noto (all languages) and Open Sans font (used by Simula)
- Customised desktop, login-manager, and screen locker backgrounds

https://www.nntb.no/~dreibh/vmimage-builder-scripts/#kde

# System-Tools

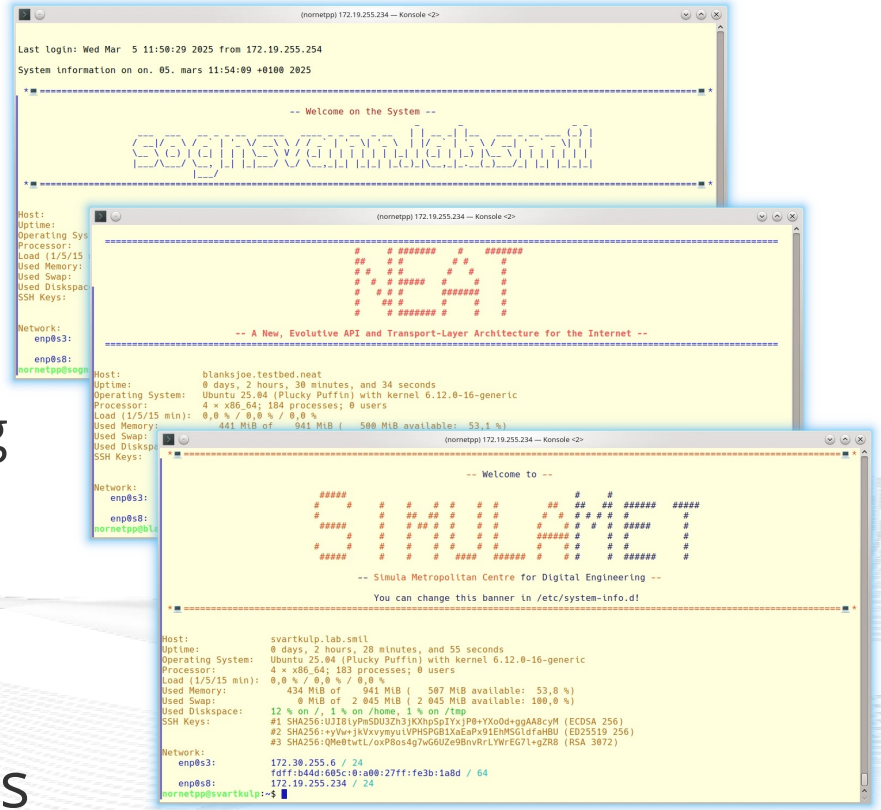
- A collection of tools for system management, e.g.:
  - Login information (System-Info)
  - Run updates (System-Maintenance)
  - Reset machine identity (Reset-Machine-ID)
- Branding
  - Login banners
  - Scripts for helping to create background images

System-Tools: https://www.nntb.no/~dreibh/system-tools/

# Branding with System-Info: System Status and Login Banners

- Not just some eye-candy …
  - *«Am I on the correct machine?»*

    (important, if you need to maintain hundreds of systems!)
  - E.g. using Figlet/Toilet-based scripting to prominently display the hostname
  - Show some project relation or details (e.g. NEAT, NorNet, RAKSHA, …)
- Also shows important system details



# Further Branding Possibilities

- Custom Boot Splash
- Custom GUI backgrounds for
  - Login manager
  - Desktop
  - Screen locker
- System-Tools provides scripting for
  - Text rendering
  - Effects (e.g. mosaic, oil painting, etc.)
  - Resizing and cropping

# Repository Directory Structure

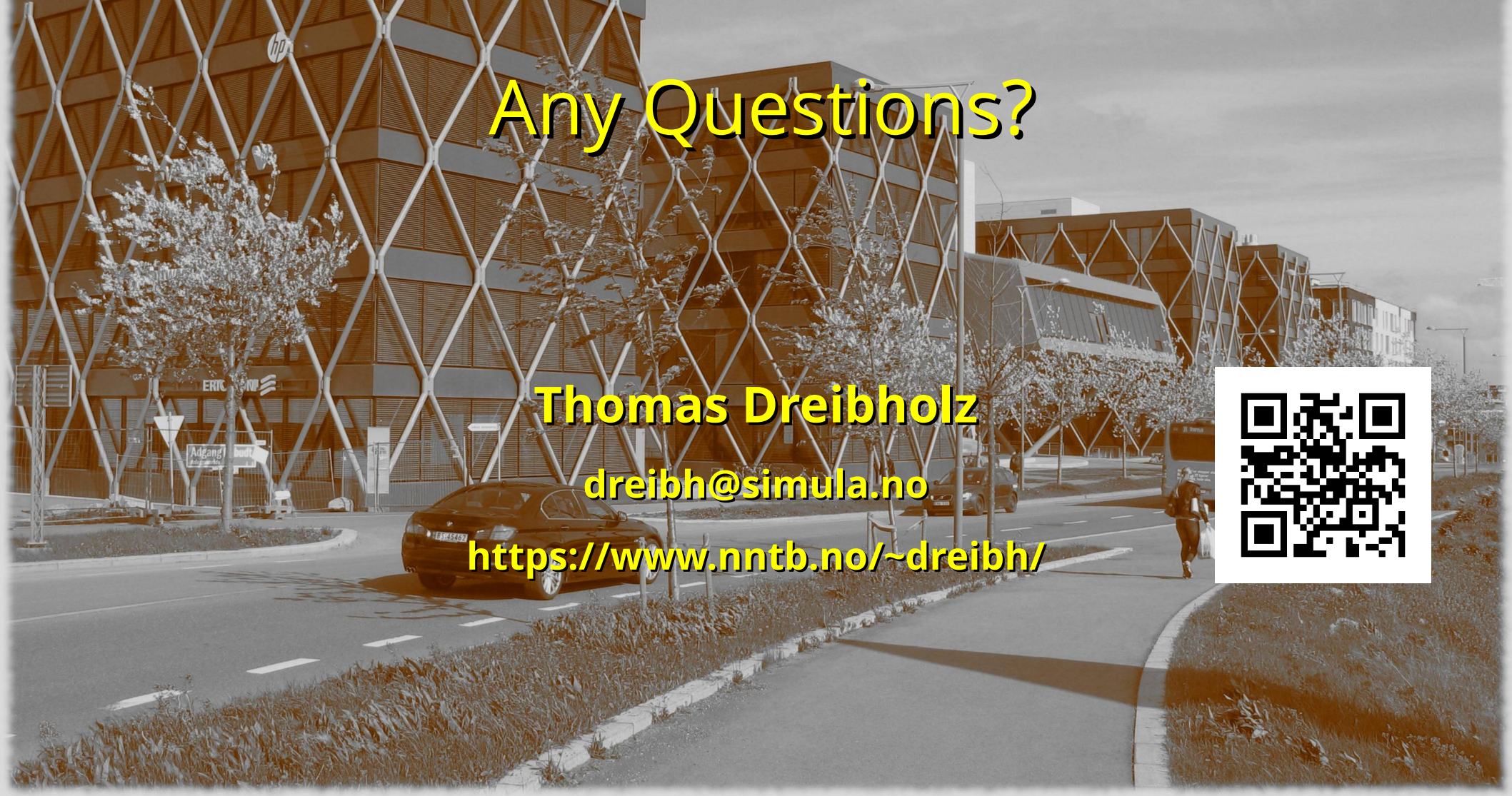
- `vmsetup`: Packer hypervisor configurations (e.g. VirtualBox)
- `installer`: Packer OS installation configurations
  (e.g. Ubuntu, FreeBSD, etc.)
- `projects`: The projects (Minimal, Basic, Development, KDE, ...)
- `scripts`: enumerated installation scripts used by the projects
- `files`: enumerated file lists (including background images!)

Git: https://github.com/simula/nornet-vmimage-builder-scripts

# After the break, in Part 2 …

- A practical example:
  - Demo VMs «Simula 25th Anniversary Edition»
  - With Simula-themed branding …
  - … and some custom Python-based AI software

Stay in this Room for Part 2!


Any Questions?
Thomas Dreibholz
dreibh@simula.no
https://www.nntb.no/~dreibh/

Simula Grand Knowledge Festival 2026

# Automated System Installation of Virtual Machines and Computers

---

## Part 2

Thomas Dreibholz ( 托马斯博士 )
Simula Metropolitan Centre for Digital Engineering
dreibh@simula.no

June 4, 2026

# Contents

- Goal
- What is needed?
- Adapting the scripts
- Running the installations
- Conclusions

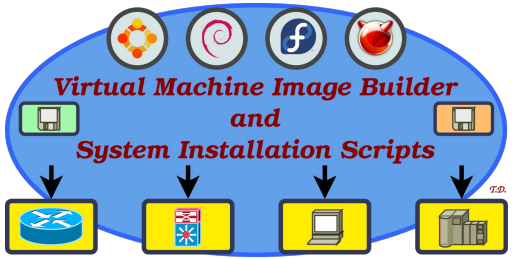
# Goal

- VirtualBox Demo VMs «Simula 25th Anniversary Edition»
  - A nice, Simula-themed KDE desktop
  - Typical researcher software (LaTeX, Python, R, ...)
  - Ready-to-use web browser (ad-blocker, privacy, etc.)
  - Some Simula/SimulaMet software preinstalled
  - Research software example, running in a Python Environment (like a lot of AI software ...)
- For Linux (current Ubuntu, Debian, Fedora) as well as FreeBSD!

# What is needed?

- Some experience with Unix system configuration and shell scripting
- Basic experience with Git

# Preparing a Git repository




1) Create a fork of https://github.com/simula/nornet-vmimage-builder-scripts

2) Clone your fork
```
git clone <your repository>
```

3) Add the upstream repository as well (for merging changes later!)
```
git remote add upstream \
   https://github.com/simula/nornet-vmimage-builder-scripts
```

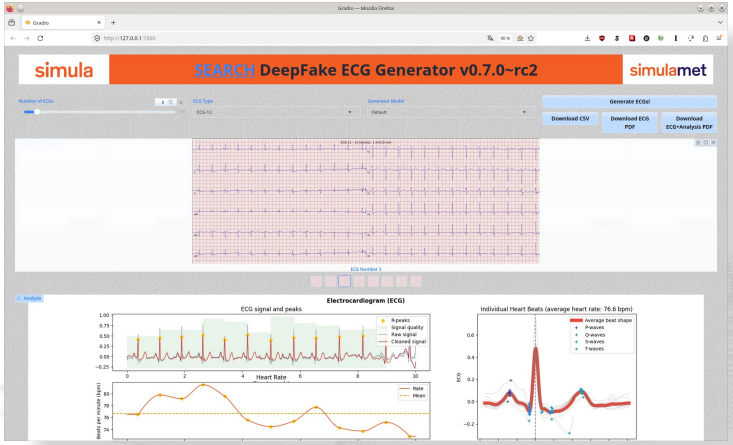
4) Add a branch, e.g. dreibh/simula25:
```
git branch dreibh/simula25
git checkout dreibh/simula25
```

# Making the project «Simula25»

- Reusing what is already there:
  - KDE, or better KDE+Development
  - **The really complicated part is already done!**
- Additional scripts for installing your software
  - `scripts/70-simula` and `files/70-simula*` (Simula banner)
  - `scripts/71-deepfakeecg` (DeepFakeECG Generator Plus application)
- First step: Create projects/Simula25.project:
  - `. ./KDE+Development.config`
  - `PROVISIONERS="${PROVISIONERS} 70-simula 71-deepfakeecg»`

# Custom Software Installation: DeepFakeECG

- SEARCH project's DeepFakeECG
- Basically:
  - Clone DeepFakeECG repository
  - Prepare Python Environment
  - Install the Python dependencies via PIP
  - Make a desktop icon + launcher script
- See: `scripts/71-deepfakeecg`

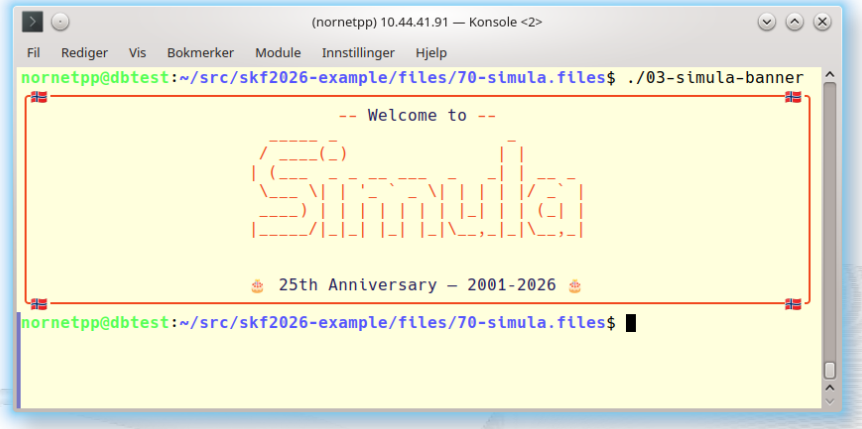



DeepFakeECG: https://huggingface.co/spaces/SEARCH-IHI/deepfake-ecg-generator-plus

# A Custom Login Banner

Basic artwork idea:

- Figlet for banners
- ANSI color sequences
  - Simula's **orange color**
    RGB: 241, 71, 29 (hex: F1471D)
  - SimulaMets's **blue color**
    RGB: 33, 27, 87 (hex: 211B57)
- System-Tools Print-UTF8 tool




- See `scripts/70-simula` and `files/70-simula*`

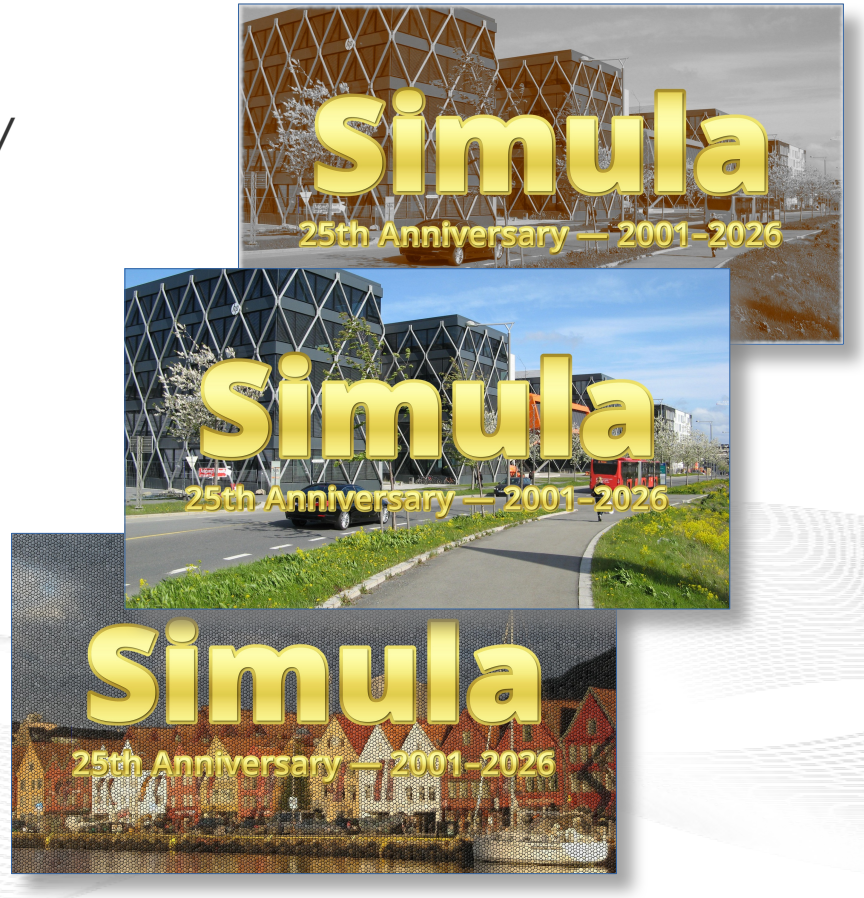
# Background Images with Some Branding

- The System Installation Scripts, and System-Tools, provide CMake-based generator scripts to generate backgrounds images in `files/backgrounds`
- Place images in `Input/`
- Check settings in `Settings.input`
- Adjust text rendering (`*.text`), and text overlay build (`build-overlay`)
- Test (`cmake . && make` → results in `Output/`)
- `files/generate-backgrounds` will finally build everything

Sounds a bit complicated? Just try, most steps are self-explaining ...

# Build the Background Images

- Put `Fornebu.webp` and `Bergen.webp` into `Input/`
- Enable «Old Photo» effect in `Settings.input`
- Adjust `Title.text` and `Subtitle.text`
  - Check `build-overlay` for the overlay details
  - Run `./test-overlay` to test overlay build
- Build everything
  - `cmake . && make`
  - Check the resulting artwork in `Output/`



# Configure the Backgrounds

1) Get the images into the new system
   - Bootsplash: `scripts/12-configure-grub` and `files/12-configure-grub`
     → extend `files/12-configure-grub`
   - Backgrounds: `scripts/20-kde and files/20-kde`
     → extend `files/20-kde`
2) Configure project «Simula25.config to use the images:

- `BOOTSPLASH="/usr/share/bootsplash/Bergen-3840x2160.jpeg"`
- `BACKGROUND_SDDM="/usr/share/desktop/Fornebu-3840x2160-oldphoto.jpeg"`
- `BACKGROUND_LOCKSCREEN="/usr/share/desktop/Fornebu-3840x2160-mosaic.jpeg"`
- `BACKGROUND_DESKTOP="/usr/share/desktop/Fornebu-3840x2160-plain.jpeg"`

# Build and Test the VMs

- The complete example repository:
  - https://github.com/dreibh/skf2026-example
  - Branch: dreibh/simula25
- Build the VM builds
  - Note: install Packer and VirtualBox!
  - Check and build script `make-simula25-vms`
- Login: simula / simula4me!
  (configurable in `make-simula25-vms`)


Ready-to-Run VirtualBox OVAs: https://packages.nntb.no/simula25

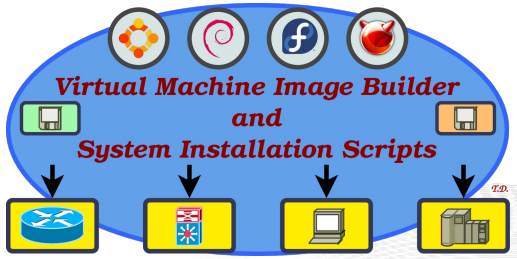
# **ERROR** – Something went wrong!

- Builds depend on online resources, which may change/be unavailable, etc.
- If something goes wrong (and this will happen sometimes):
  - Log into the incomplete VM
  - Clone your repository
  - Manually test and fix the script, commit, and push
- Try again!

You have a fix or improvement? Submit a pull request!

# Merge Upstream Changes

- The Virtual Machine Image Builder and System Installation Scripts are under steady development
  - Changes by the OS distributions
  - Improvements and fixes
- Track and merge upstream changes:
  - If not already done:
    ```
    git remote add upstream \
       https://github.com/simula/nornet-vmimage-builder-scripts
    ```
  - `git fetch --all`
  - `git merge upstream/master`



# Enjoy the freshly built VMs!

- Final step:
  - `sudo Reset-Machine-ID -H <HOSTNAME>`
  - Generates new machine ID `/etc/machine-id` and SSH keys!
  - Also sets the hostname
- Not used a VM/machine for a long time?
  - `sudo System-Maintenance`
  - Installs package updates, firmware updates (if possible), etc.

System-Tools: https://www.nntb.no/~dreibh/system-tools/


Any Questions?
Thomas Dreibholz
dreibh@simula.no
https://www.simula.no/people/dreibh