Tutorial: Reproducible distributed environments with NixOS Compose

Quentin Guilloteau, Jonathan Bleuzen, Millian Poquet, Olivier Richard

Tutorial: Reproducible distributed environments with NixOS Compose

Quentin Guilloteau[1], Jonathan Bleuzen[1], Millian Poquet[2], and Olivier Richard[1]

[1]Univ. Grenoble Alpes, Inria, CNRS, LIG, F-38000 Grenoble, France
[2]Université de Toulouse, IRIT, CNRS, Toulouse INP, UT3, Toulouse, France

1 Abstract

Developing software environments for experiments is an iterative and time consuming process. Users usually build multiple times the image, adding every time a forgotten dependency or fixing a previously added one. As the building time of such images takes around ten minutes for full system tarballs, it does not encourage the experimenters to follow good reproducible practices when setting them up. As a result, those images cannot be rebuilt nor modified by someone else.

In this tutorial, we introduce the users to NixOS Compose[GBPR22], a tool based on Nix[DdJV04] and NixOS[DL08] to generate and deploy reproducible environments on distributed platforms. We will first present Nix and the notions required to use NixOS Compose. As NixOS Compose can target several platforms, the users will set up their environment with lightweight containers (docker) on their local machines, allowing them to iterate quickly on their environment description. Once the environment ready with containers, users will be able to quickly test it on the Grid'5000 testbed[BCAC+13] using `kexec`, before generating a full system tarball.

2 Topic & Relevance

Although the scientific community is traversing a reproducibility crisis, and the computer science field does not make an exception, there is a global trend to improve the reproducibility practices of scientists and professionals. HPC software environments are complex, and applications often require complicated software stacks. One missing dependency in this stack could lead to difficult or even unreproducible results in the future. However, tools like the functional package manager Nix[DdJV04] address this issue by tracing every application's dependency, which this tutorial focuses on. Moreover, we also present two other solutions, the NixOS Linux Distribution[DL08] and our tool: NixOS Compose[GBPR22].

The NixOS Linux distribution, which extends the Nix paradigm to the whole operating system, improves the Quality-of-Life of experimenters who also need to control the entire system (kernel, firmware, services, etc.). Also, building deployment images for any kind of system and environment is an iterative and time-consuming process. That is why we introduce NixOS Compose, a tool based on NixOS that allows the users first to test their reproducible images locally using lightweight tools (containers and Virtual Machines) before deploying them on physical hardware (e.g., Grid'5000[BCAC+13]). This tool aims to reduce the friction of developing clean, reproducible, and distributed environments while enabling users to share their configurations with anyone to achieve the same settings.

We believe this tutorial will benefit CARLA attendees, assisting them in improving the reproducibility of their experiments and executions in single nodes or distributed setups.

3 Resources, Duration & Audience

The tutorial will last **two slots of 4 hours**, and will be in **English**. The first slot will introduce Nix and its concepts through examples and common usage. In the second slot, the attendees will use NixOS Compose to produce a distributed environment for an experiment. Attendees will be granted a temporary access to the Grid'5000 testbed to deploy their environment on physical machines.

Audience: Attendees will need a Linux or MacOS laptop with an Internet connection, as well as root privileges on their machine (required to install Nix). No knowledge of NixOS is needed, but basic knowledge of the Linux environments and tools as well as basic notions of functional programming would be appreciated.

Content: In this tutorial, attendees will *(i)* learn about Functional package managers and Nix, *(ii)* create reproducible packages with Nix, *(iii)* create reproducible and sharable software environments with Nix, *(iv)* create reproducible docker image from Nix packages, *(v)* create reproducible NixOS image, and *(vi)* use NixOS Compose to create reproducible and distributed NixOS environments.

4 Previous Editions

History: There are have been several previous editions for the Nix part: `https://nix-tutorial.gitlabpages.inria.fr/nix-tutorial/index.html`.

Novelty: We will present NixOS Compose and take an example of distributed experiment to build a reproducible environments.

5 Organizers

Quentin Guilloteau holds a master degree in Computer Science from ENSIMAG engineering school in France (2020). He is currently a PhD candidate in Computer Science at the Grenoble Alpes University (France) on the topic of an Autonomic approach to runtime management of HPC cluster resources. He is also interested in reproducible research.

Jonathan Bleuzen has a bachelor's degree in Computer Science from The University of Western Brittany (UBO), a Msc in Computer Science from Grenoble Alpes University (2019). He is currently a research engineer in the Grenoble Computer Science Laboratory, in the DATAMOVE team.

Millian Poquet is an Associate professor at University Toulouse III. His research interests include experimental practice on distributed systems, simulation of distributed systems and resource management, and energy optimization problems in high-performance computing. He obtained his Ph.D. in computer science from University of Grenoble, and his M.S. in computer science from the University of Orléans.

Olivier Richard is an Associate professor at Grenoble Alpes University. His research interests are focused on system architecture for high performance computing and large distributed system. He also works on tools and methods to enhance experiments' reproducibility in these domains. He co-designed the Grid'5000 national testbed (head of Grenoble's site since 2003).

References

[BCAC+13] Daniel Balouek, Alexandra Carpen Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Pérez, Flavien Quesnel, Cyril Rohr, and Luc Sarzyniec. Adding virtualization capabilities to the Grid'5000 testbed. In Ivan I. Ivanov, Marten van Sinderen, Frank Leymann, and Tony Shan, editors, *Cloud Computing and Services Science*, volume 367 of *Communications in Computer and Information Science*, pages 3–20. Springer International Publishing, 2013.

[DdJV04] Eelco Dolstra, Merijn de Jonge, and Eelco Visser. Nix: A Safe and Policy-Free System for Software Deployment. page 14, 2004.

[DL08] Eelco Dolstra and Andres Löh. Nixos: A purely functional linux distribution. *SIGPLAN Not.*, 43(9):367–378, sep 2008.

[GBPR22] Quentin Guilloteau, Jonathan Bleuzen, Millian Poquet, and Olivier Richard. Painless transposition of reproducible distributed environments with nixos compose. In *2022 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2022.