POLYTECH GRENOBLE

Projet S10

Soutenance finale

Test d’infrastructures avec NixOS

Corentin Humbert
Corentin Sueur
Titouan Minier Mancini

Olivier Richard
Quentin Guilloteau
Jonathan Bleuzen

Rappel du sujet

- Expérimenter avec le système d'exploitation NixOS et l'outil NixOS-Compose pour réaliser des tests
- Effectuer une série de tests de déploiements localement et sur le réseau Grid'5000 pour différentes piles logicielles (K8S, ELK, Hadoop)
- Documenter nos expériences pour avoir des retours utilisateurs concrets afin d'améliorer l'utilisabilité de l'outil
- **Bénéfice personnel :** Découvrir le système NixOS et avoir un aperçu de l' étendu de ce qu'il est possible de faire avec

Les technologies

Nix - *Le package manager*

NixOS - *La distribution Linux*

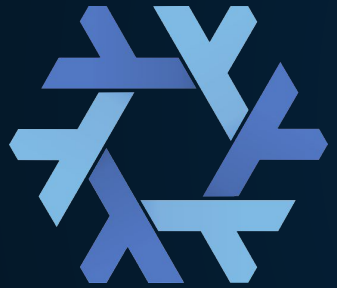
NixOS-Test - *La librairie de test*

NixOS-Compose :

Extension de l'utilisation de NixOS-Test vers d'autres plateformes que la VM : Docker, Grid'5000...

Piles logicielles à tester:

L'écosystème Nix



Nix fonctionne avec des dérivations qui définissent la construction de paquets stockés dans le store.

NixOS-Compose reprend les bases de NixOS-Test et propose d'écrire un fichier de composition, décrivant le système à construire à partir de dérivations, et les tests à effectuer sur le système une fois créé.

C'est cette construction du système qui peut être réalisée sur différents supports, comme une ou plusieurs machine(s) virtuelle(s), conteneur(s), ou noeud(s) sur Grid'5000.

Test d'infrastructures avec NixOS

Réalisations techniques

Your whole Operating System in a configuration file...

NixOS

...as a perfect support for all your software experiments.

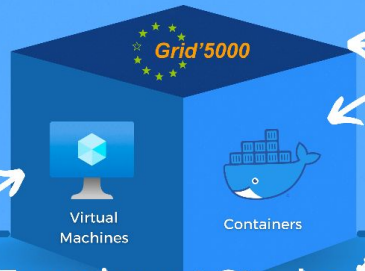
*The Operating System built on top of the Nix package manager, for a **functional**, **reproducible**, **declarative** and **reliable** approach to package management and system configuration.*

NixOS-Test

NixOS-Test is a library containing thousands of written tests for existing software and technologies. The tests themselves consist of Nix files split in two parts.

The first is the **configuration part** containing the various system configuration parameter used to deploy the software or technology on the system.

The second is the **test part** consisting of a Python script executing various commands to make sure the software is properly working on NixOS in **Virtual Machines**.



NixOS-Compose

NixOS-Compose is a tool developed by the Inria Datamove team and act as an extension of NixOS-Test.

The tool itself relies on experimental features of NixOS and its purpose is to **extend the usage of NixOS-Test** to deployment platforms other than the host operating system.

These other deployment platforms include containers through the **Docker Engine**, and the large-scale French testbed for experiment-driven research **Grid5000** which is a distributed infrastructure.

Project team:
Corentin Humbert
Corentin Sueur
Titouan Minier Mancini

Experiment Stacks

Datamove team:
Olivier Richard
Jonathan Bleuzen
Quentin Guilloteau

Kubernetes

*Kubernetes is an open-source system for automating **deployment**, **scaling**, and **management** of **containerized applications**. It runs a **Kubelet** agent on each machine and manages the cluster with components linked together through the **API server** and the network.*

The experiment

We will deploy a Kubernetes cluster providing **high availability** with 2 master nodes and 4 worker nodes. Inside this cluster, we will deploy an example of a **micro-service application**. To further reproduce **production grade** environment for our micro-service application, we will add the **service mesh Istio** to the cluster.

*Istio is a service mesh which helps in the management and monitoring of micro-service applications. In the context of Kubernetes, it will add a **sidecar container** managing the network, to all the pods, so the Business Logic is kept **separated** and **independent** while providing improved **telemetry**.*

We focus on deploying the example application from Istio repository: Bookinfo
Plus, we will add monitoring tools like Prometheus, Grafana, Jaeger and Kiali.

Through this experiment we will deploy on a **3-level abstraction** environment: NixOS, Kubernetes + Istio and the application itself. We will explore file structure to **separate** the Kubernetes and Istio configuration from the Kubernetes resources (deployments, services...) creation and the application setup.

The goal is to **simplify** the setup of a micro-service application itself and **abstract** the Kubernetes, Istio and monitoring layer.

ELK

"**ELK**" is an acronym for three open source projects: **Elasticsearch**, **Logstash**, and **Kibana**. Each of them plays an essential role in the ELK stack. Let us introduce each of them:

Elasticsearch is a distributed, search and analytics engine. The purpose of Elasticsearch is to centrally store data for lightning fast search, fine-tuned relevancy, and powerful analytics that scale with ease.

Logstash is a free and open server-side data processing pipeline that ingests data from a multitude of sources, transforms it, and delivers a usable output that can be sent to any application.

Kibana is a free and open user interface that can help visualize data stored in Elasticsearch. It comes with multiple functionalities to allow an efficient tracking of the different requests as well as a quick visualization of the software infrastructure performances.

The experiment

Our experiment will consist in writing proper configuration files in order to deploy the ELK stack using the **NixOS-Compose** project. The objective will be to document our experiences with it to provide feedback on how the stack can be deployed using NixOS-compose.

Hadoop

Hadoop is a set of tools helping to deal with huge amount of unstructured data in the context of the **Big Data**. It's a framework composed of four main modules plus software used for big data processing:

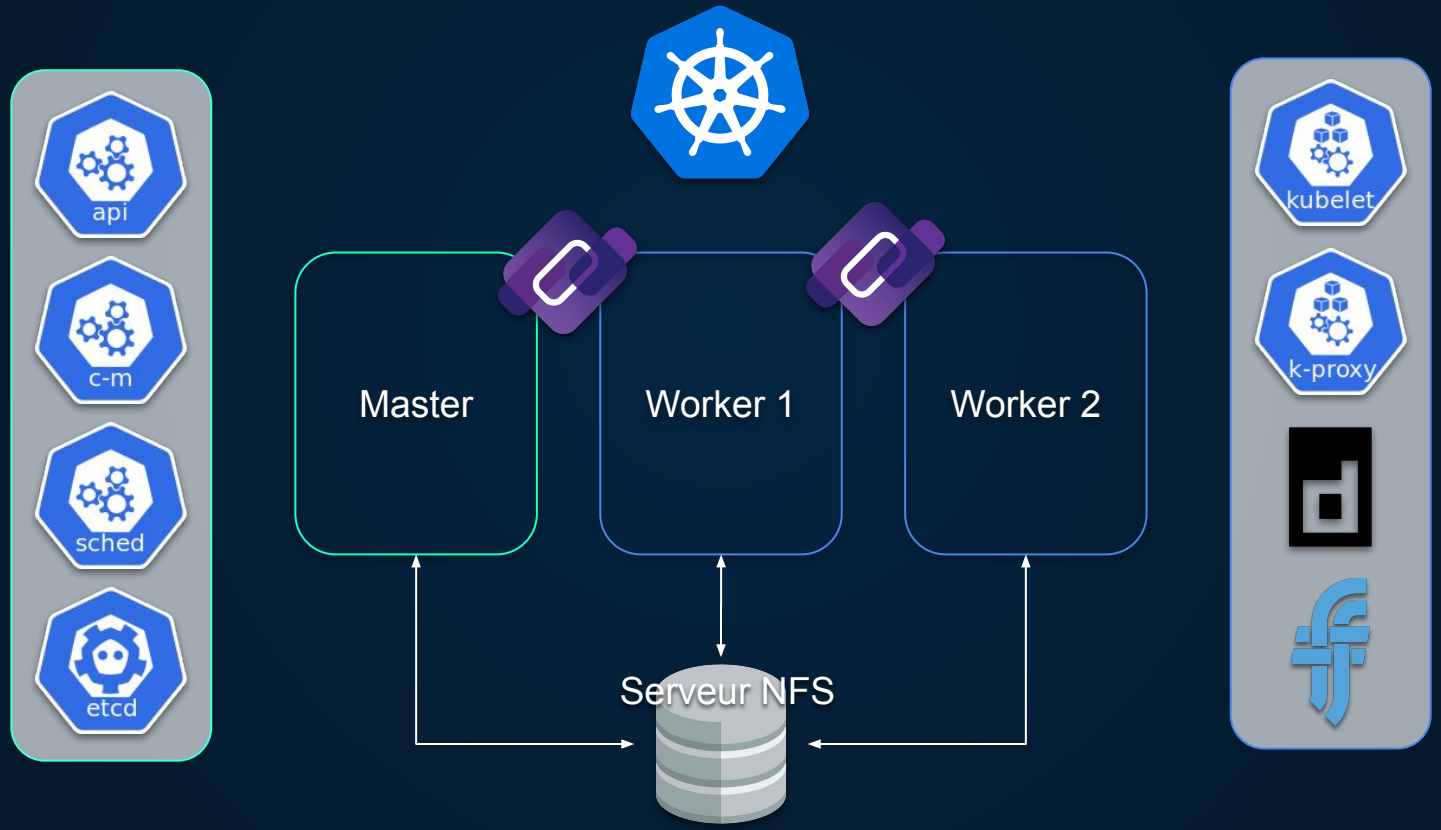
- The **Hadoop Distributed File System** (HDFS), a distributed file system specializing in data high availability and high-throughput access.
- **Yet Another Resource Negotiator** (YARN), as its name suggests, manages resources allocation and job scheduling inside a cluster.
- **Hadoop MapReduce**, a YARN-based system for parallel processing of large data sets.
- **Hadoop common**, utilities that support other modules.
- The different additional nodes nodes using Apache software, or other, that can access the resources through httpFS and execute commands on the main nodes via RPC.

The experiment

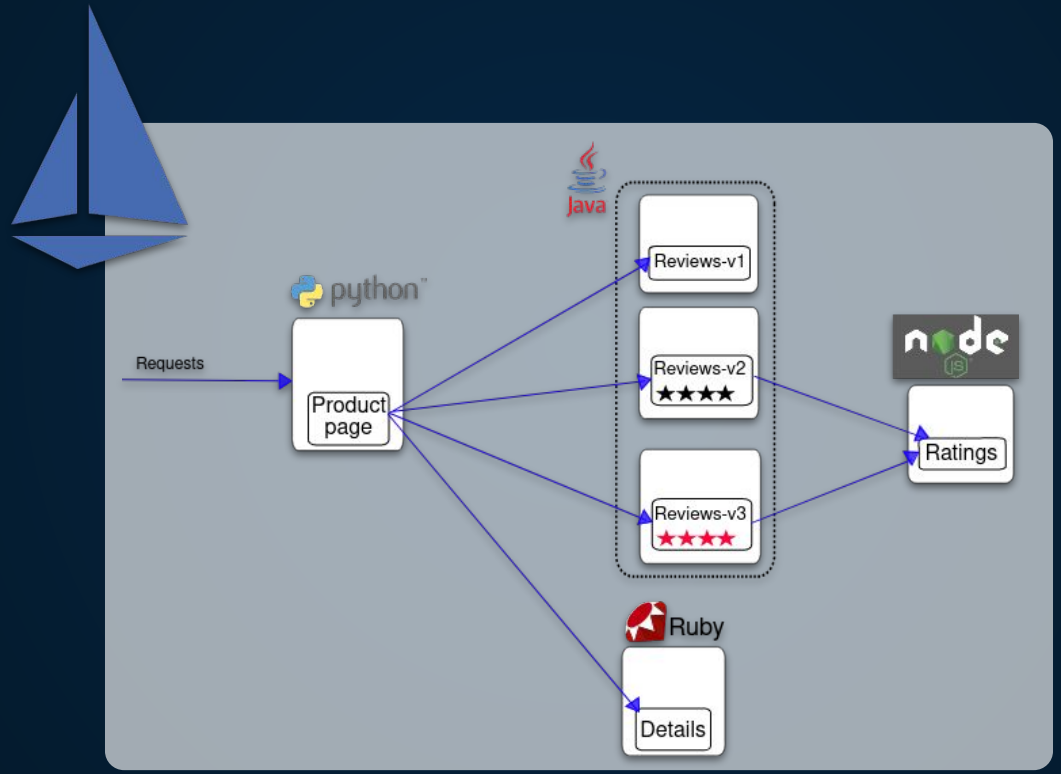
This experiment will consist in writing proper configuration files in order to deploy the Hadoop frameworks and software using the **NixOS-Compose** project. The configuration will consist in hosting the frameworks and multiple nodes.

We will deploy multiple nodes with the main frameworks along with the different services available on this sofware like Apache ZooKeeper, Apache Hive, Apache pig and others. We won't developp how these software work but rather how to configure them and make the use the base frameworks by making test cases and jobs.

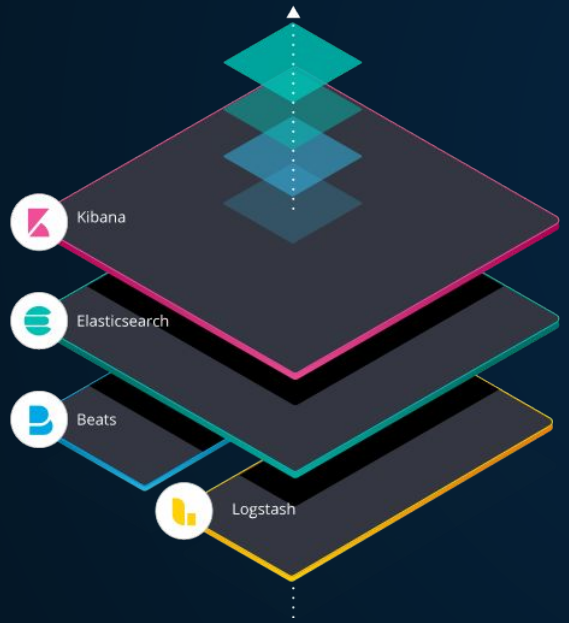
Réalisations techniques



Réalisations techniques

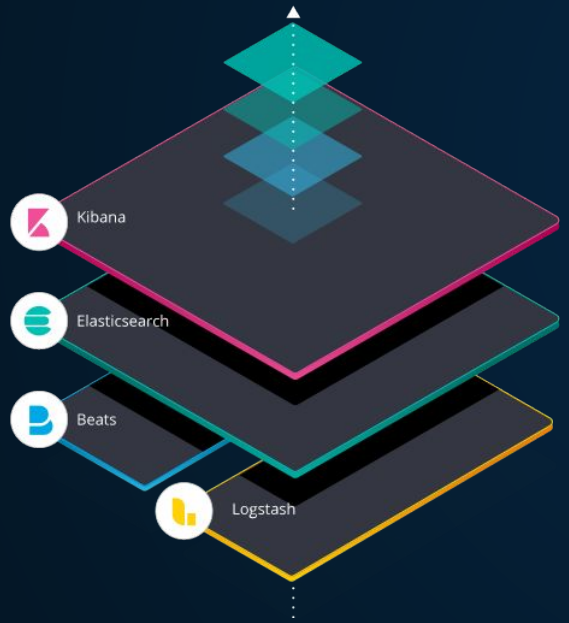


Réalisations techniques



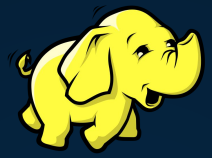
- Logstash : Pipe de traitement de données
- Elasticsearch : Base de données distribuée spécialisée dans l'indexation et la recherche de documents
- Kibana : Interface de visualisation de données
- Beats : Rassemble une multitude de data shippers

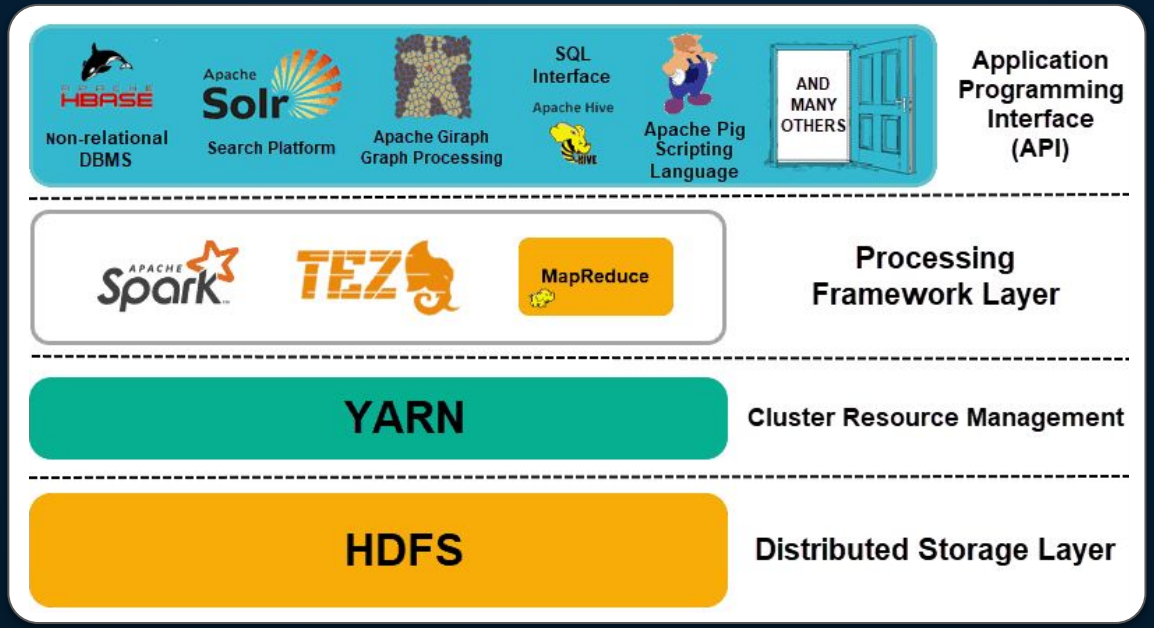
Réalisations techniques



- Réalisation d'une composition pour NixOS-Compose (ELK + Beats)
- Tests de déploiement de la solution sur Docker, nixos-test et Grid'5000
- Documentation du travail effectué et réalisation d'un tutoriel

Réalisations techniques





Gestion de projet

- Tenue d'un journal de bord par personne :
  - Documenter le travail réalisé au cours de la journée
  - Faire part des éventuels problèmes rencontrés
- Rédaction d'une roadmap ensuite mise à jour au fur et à mesure
- Organisation de réunions :
  - Une réunion quotidienne en équipe partielle pour un retour rapide sur l'avancée et éventuellement discuter des blocages
  - Une réunion hebdomadaire en équipe complète pour faire le point sur la semaine et éventuellement redéfinir les objectifs

Outils de gestion

Communication instantanée

Discord

Telegram

Réunions

BigBlueButton

Zoom

Gestion du code et de la documentation

GitLab

Conclusion

Découverte de l'environnement Nix

Découverte du projet NixOS-Compose

Apprentissage sur trois piles logicielles

Pratique des systèmes distribués

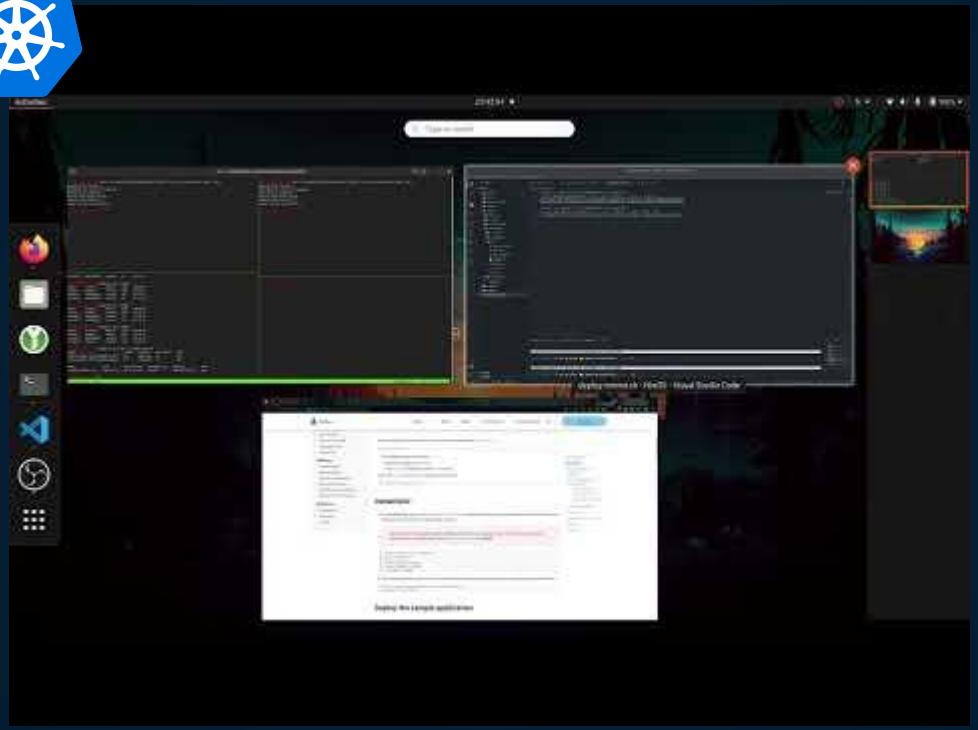
Production de compositions fonctionnelles

Production de tutoriels et retours utilisateurs

Collaboration avec l'équipe Datamove

Test d'infrastructures avec NixOS

Démonstration



https://youtu.be/uOh8BJPj7MU