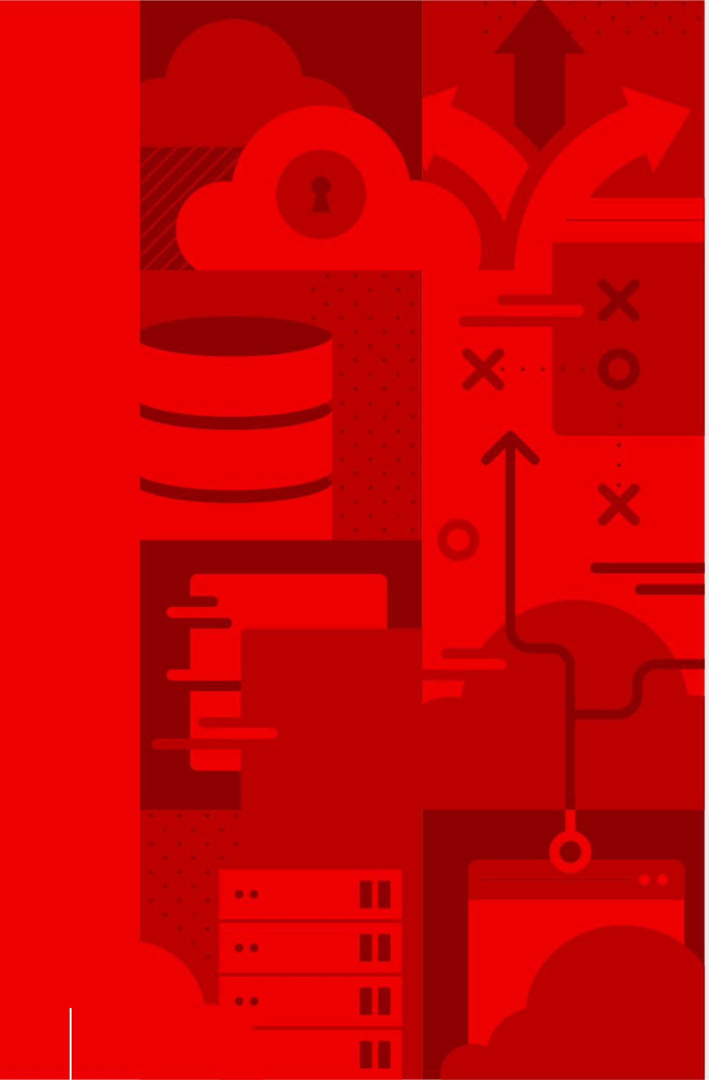


Settle your CI environment with the Software Factory

Szymon Datko & Adrian Fusco Arnejo

OpenInfra Summit Asia – September 3rd, 2024

About Us

Szymon Datko
Senior Software Engineer

- – Linux enthusiast and free/open source software lover.
- – Loves playing board and computer games.
- – Teacher at Wrocław University of Science and Technology.

Adrian Fusco Arnejo
Software Engineer

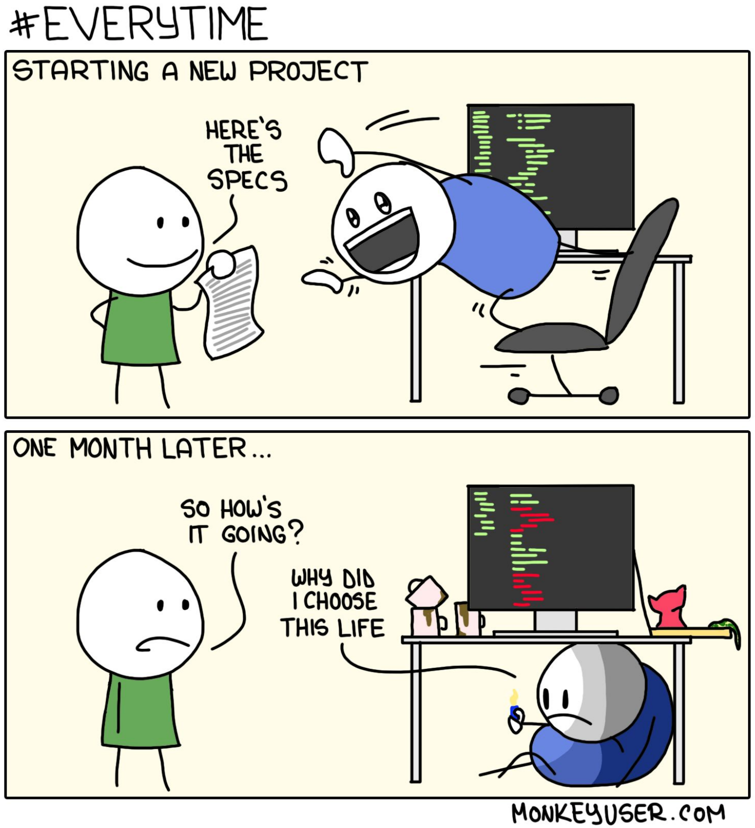
- – DevOps soul, hartened with Perl and Bash.
- – Passionate for traveling, food and immersions into different cultures.
- – Fluently speaks Español, English, Galego, Italiano, learning Turkçe.

Preparing for a new project?

For start, one may need:

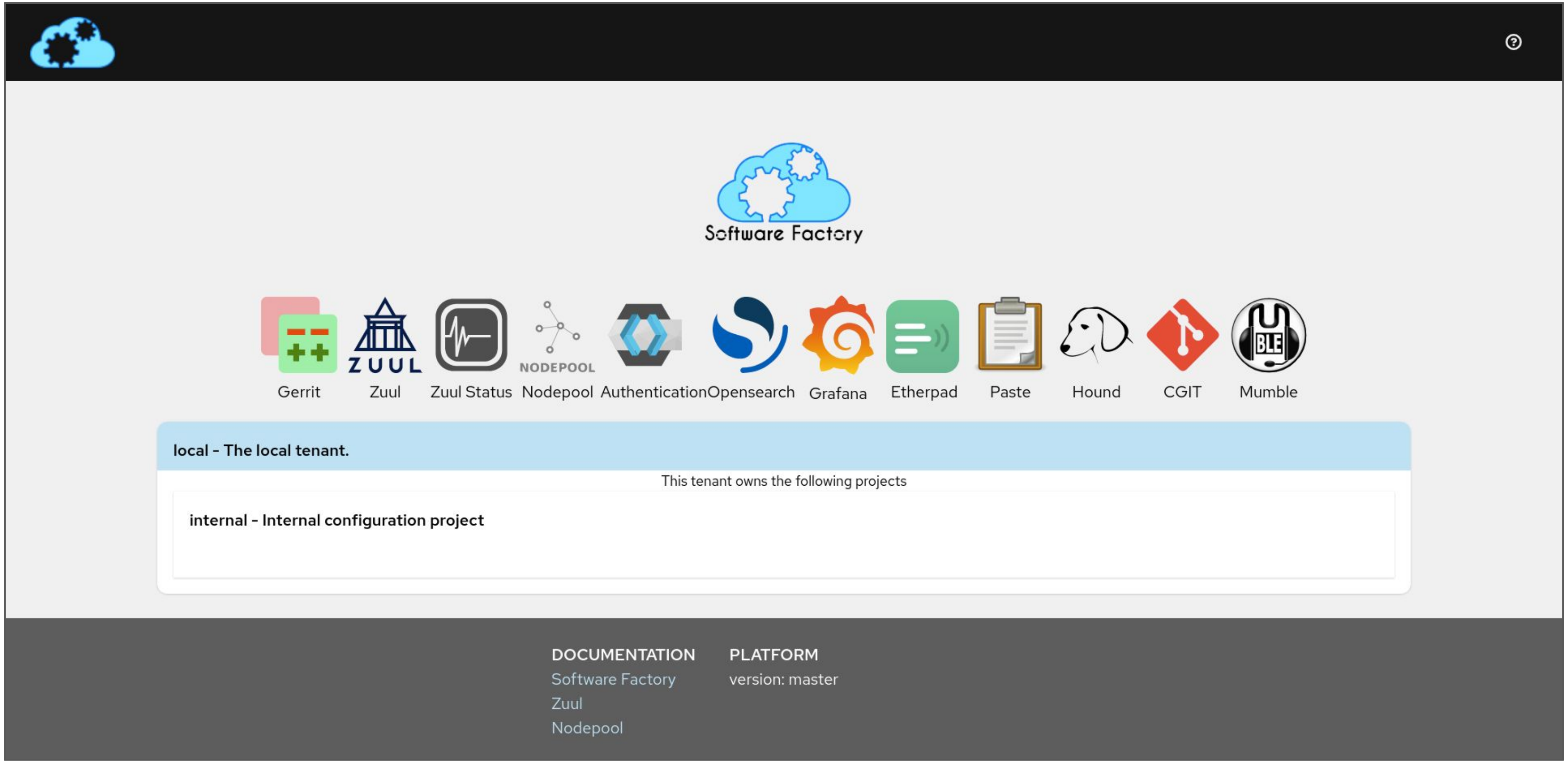
- A source code **repository** (and a review system).
- An **automation** suite for testing (CI/CD).
- System and service **metrics** (monitoring).
- Logs search engine and processing.
- Identity and access management.
- Additional collaboration tools (notes, communication).

Then, manage and keep everything up-to-date…



https://www.monkeyuser.com/2018/everytime/

Meet the Software Factory project!



Software Factory

local - The local tenant.

This tenant owns the following projects

internal - Internal configuration project

DOCUMENTATION

Software Factory

Zuul

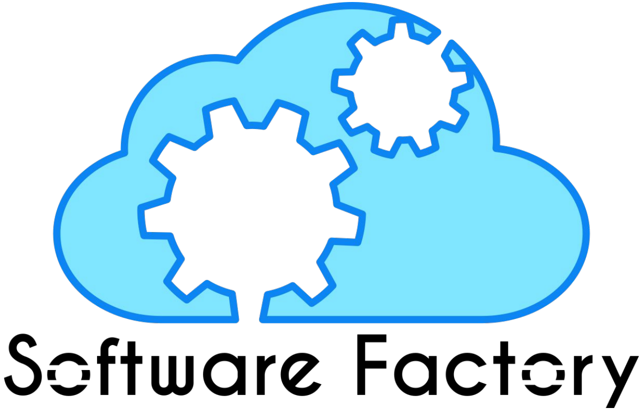
Nodepool

PLATFORM

version: master

A bundle for all your needs

- A **collection** of components that provides a powerful platform to build software.
- Seamlessly **integrates** services covering each step in the production chain.
- Completely **Open-Source** in its nature: `https://www.softwarefactory-project.io`
- It **simplifies** system operations:
  - a single tool to deploy all services,
  - backup and recovery process,
  - integrated instance monitoring,
  - software upgrades.



Components!

- Main services:
  - Code review system: **Gerrit**.
  - CI/CD solution: **Zuul** & **Nodepool**.
- Additional services:
  - Code search: **Hound**.
  - Identity and access management: **Keycloak**.
  - Jobs log search engine: **Logstash**, **OpenSearch**.
  - System and services metrics: **Telegraf**, **Influxdb**, **Grafana**.
  - Collaborative tools: **Etherpad**, **Lodgeit**, **Mumble**.

Components: code review system

- Utilizes `Gerrit` for storing and managing source code.
- Originally at developed at Google as a fork of Rietveld,

  a code review tool for Subversion version control system.
- Designed to provide a robust and efficient collaboration on changes.
- Web interface oriented on reviewing and commenting code changes.
- Heavily integrated with **git** version control system.
  - Even it’s database (NoteDb) is git-based!
- Supports fine-grained access control with multiple approval labels and levels of votes.

Components: CI/CD solution

- `Zuul` – a project gating system.
- `Nodepool` – a system for managing test node resources.
- Solution for Continuous Integration,

  as well as for Continuous Delivery.
- Relies on **Ansible** playbooks/roles.
- Supports cross-project dependencies.
- Drives one of the largest CI system

  in the open source world – OpenDev!

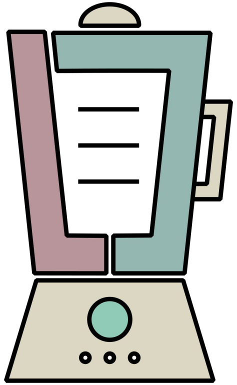
Stop Merging Broken Code

Components: system and services metrics

- Data Collection:
  - `Telegraf`: for gathering metrics and events.
  - Collect, transform & output the data.
- Data Storage:
  - `InfluxDB`: time series database that handles metrics, logs & traces.
- Monitoring & Alerting:
  - `Grafana`: data visualization and monitoring platform.
  - Also possible to be managed as code via `Jsonnet`.

Components: jobs log search engine

- Server-side data processing pipeline – `Logstash`.
  - Ingest data from sources, transform it and send to indexer.
- RESTful search and analytics engine – `OpenSearch`.
  - Stores indexed logs for further browsing and analysis.
  - Visualization of data with OpenSearch Dashboards.
- Also there is `logreduce` / `logjuicer` available.
  - Highlights distinctive parts to save time in finding root causes.

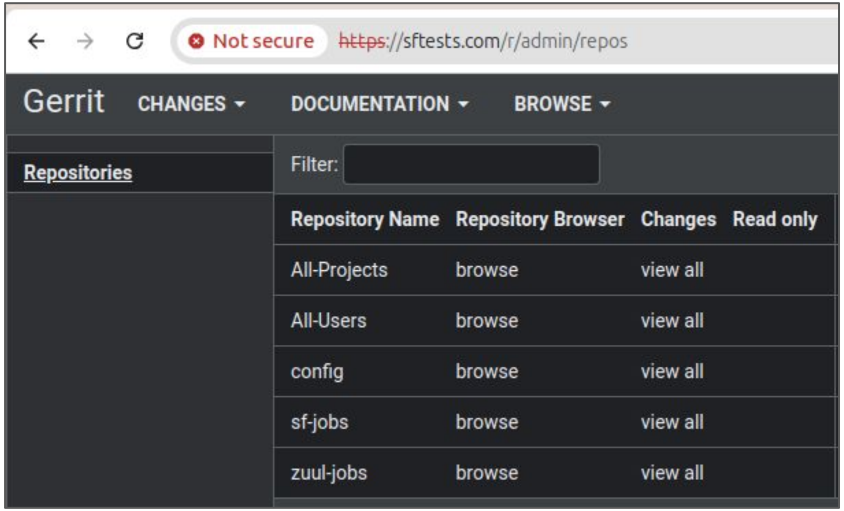


Components: identity and access management

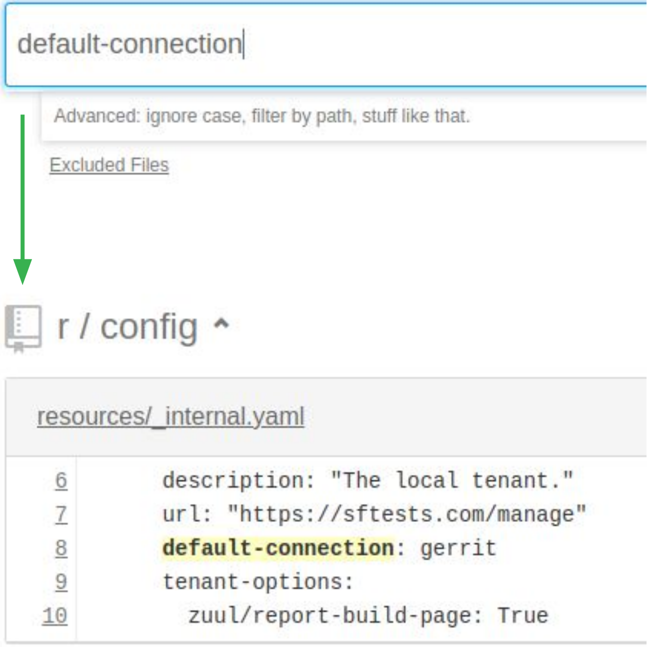
- IAM provided and managed by `Keycloak` service.
- Built-in support to connect/sync users from LDAP or Active Directory servers.
  - Also a relational database can be used.
- Enables SSO on different components that allows authentication.
  - E.g. Gerrit, Zuul, OpenSearch or Grafana.
- Offers 2FA support.
- Realm, Admin & Users Management Console.

Components: code search tool

- Extremely fast source code search engine – **Hound**.
- Simplicity: a static React frontend and Go-based index backend.
- Allows rapid search in all git repos hosted in Software Factory.



Repository Name	Repository Browser	Changes	Read only
All-Projects	browse	view all	
All-Users	browse	view all	
config	browse	view all	
sf-jobs	browse	view all	
zuul-jobs	browse	view all	



default-connection

Advanced: ignore case, filter by path, stuff like that.

Excluded Files

r / config

resources/_internal.yaml

```
      description: "The local tenant."
      url: "https://sftests.com/manage"
      default-connection: gerrit
      tenant-options:
        zuul/report-build-page: True
```

Components: collaborative tools

- Collaborative edition provided by `Etherpad`.
  - A modern really-real-time collaborative document editor with rich text editing capabilities.
  - Customizable via plugins (can be managed via Software Factory config).
  - Real-time chat feature & multi-language support.
- Pastebin-like tool that helps sharing code snippets – `LodgeIt`.
- Voice chat component – `Mumble`.
  - Open-source, low-latency, high quality voice chat software.
  - Encrypted communication & public/private-key authentication by default.

Why it is worth it?

- It provides a Continuous Integration and Deployment system that **works out of box**, with everything ready to be used after deployment.
- It can be deployed **anywhere**: baremetal systems and virtual machines.
- It keeps its **configuration as code**, that is: versioned, reviewed, tested and can be rolled back for any change.
- It is **fast** to deploy (about 15-30 minutes) ~ works exceptionally well as a **sandbox playground**.
    - Nevertheless, it is a fully **production-ready** soliton as well!
    - No hard ties to the list of components – most of them are optional.
- More: https://softwarefactory-project.io/docs/faqs/added_value_of_sf.html

Installation

- CentOS 7: Installation via RPM packages:
    - `yum install -y https://softwarefactory-project.io/repos/sf-release-3.8.rpm`
    - `yum update -y`
    - `yum install -y sf-config`
    - Note: CentOS Linux 7 reached end of life (EOL) on June 30th, 2024.
        - `sed -i 's|^mirrorlist|#mirrorlist|g' /etc/yum.repos.d/CentOS-*`
        - `sed -i 's|#baseurl=http://mirror.centos.org|baseurl=https://download.cf.centos.org|g' /etc/…`

- RHEL 9: Installation via ansible roles using the Software Factory Configuration Management.

Configuration

- The whole configuration is kept under **`/etc/software-factory/`** directory.
- Two main files of interest.
  - **`arch.yaml`** – lists the services to be deployed.
  - **`sfconfig.yaml`** – contains configuration for the components.
    - Notable settings are: **`admin_password`** and **`fqdn`**.
- Details and examples of valid architecture configurations can be found in:
  - documentation: `https://softwarefactory-project.io/docs/operator/architecture.html`;
  - repository: `https://softwarefactory-project.io/cgit/software-factory/sf-config/tree/refarch`;

Usage

- To run the deployment the following command has to be executed:

  `sfconfig`

  - It sets the services in podman containers on host.
  - It runs Ansible roles per each component.
    - Defined in **`/usr/share/sf-config/ansible/roles/`**
- It has to be re-executed also for every configuration change. (**!**)
- It takes about 15-30 minutes for the deployment/configuration update to finish.

Example – set your own playground

- Your own personal sandbox can be quickly deployed using our supplementary material:
  https://github.com/adrianfusco/openinfra2024-software-factory-playground

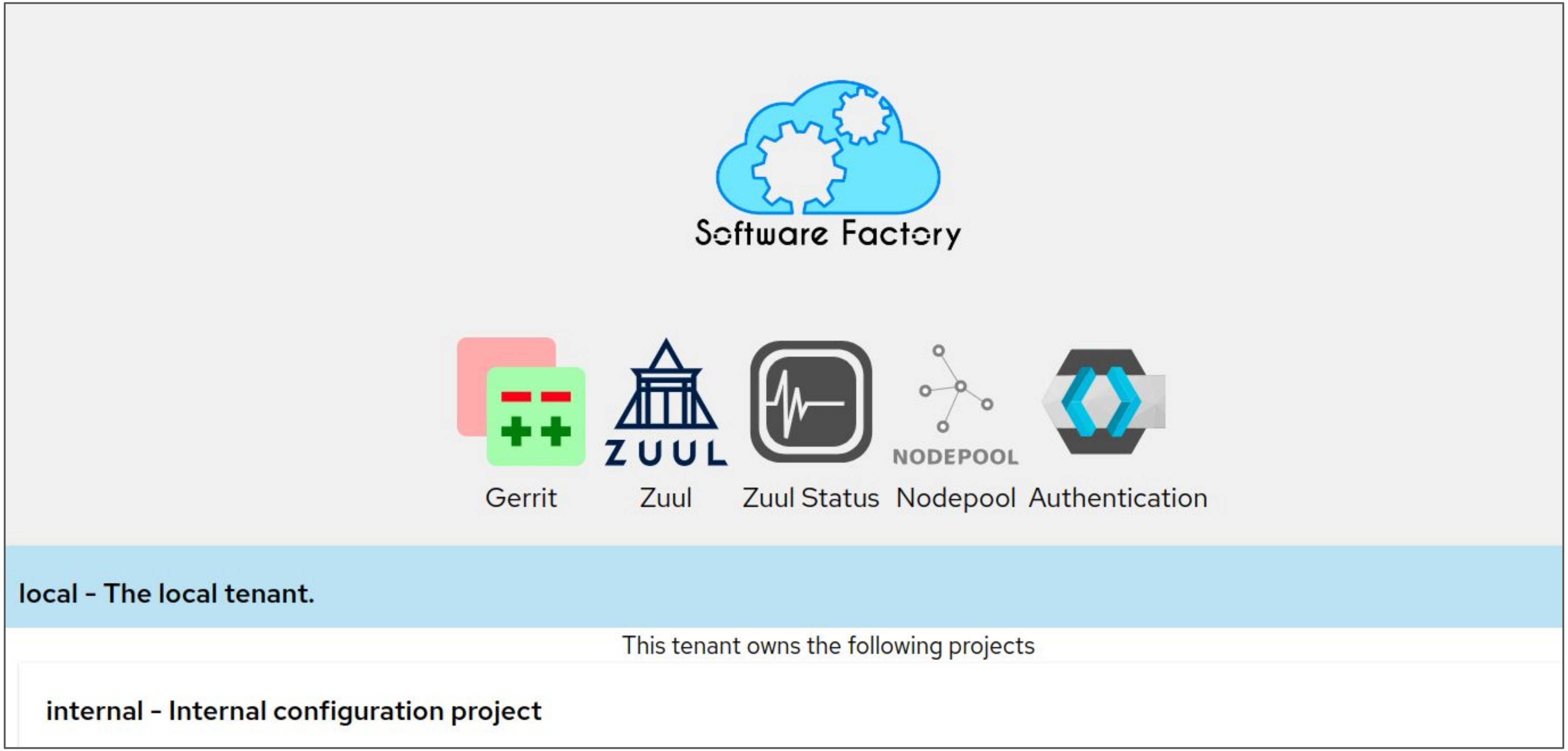
Example – deployment in virtual machine

```
# Prerequisite: install Vagrant and it’s libvirt plugin

$ git clone https://github.com/adrianfusco/openinfra2024-software-factory-playground.git

$ cd openinfra2024-software-factory-playground/Centos7-deployment

$ vagrant up --provider=libvirt

Bringing machine sftests.com' up with 'libvirt' provider...
==> sftests.com: Creating image (snapshot of base box volume).
==> sftests.com: Creating domain with the following settings...
==> sftests.com:  -- Name:              Centos7_test.sftests.com
==> sftests.com:  -- Description:       Source: /home/afuscoar/openinfra2024-software-factory…
==> sftests.com:  -- Domain type:       kvm
==> sftests.com:  -- Cpus:              6
==> sftests.com:  -- Feature:           acpi
==> sftests.com:  -- Feature:           pae
==> sftests.com:  -- Memory:            8192M
==> sftests.com:  -- Base box:          centos/7
```

Example – preparing access from web browser

```
$ vagrant global-status

id       name                provider  state    directory
----------------------------------------------------------------------------------------------
7dd2891  sftests.com         libvirt   running  /home/afuscoar/openinfra-2024-software-factory…


# We must add the IP in `/etc/hosts` file to resolve the VM address in hypervisor


$ vagrant ssh -c "hostname -I | awk '{print \$1}'"

192.168.121.21


$ echo "192.168.121.21 sftests.com" | sudo tee -a /etc/hosts


# Now the http://sftests.com address can be opened in the web browser
```

Example – SF available in web browser



local - The local tenant.

This tenant owns the following projects

internal - Internal configuration project

Example – services running as containers

```
# After the installation we can see the podman containers running.
# Each one corresponding with the components we enabled in the configuration.

$ podman ps

CONTAINER ID  IMAGE                                                     COMMAND               CREATED    STATUS  NAMES
3c1595ab2563  quay.io/software-factory/zuul-web-sf38:10.0.0-1           /usr/local/bin/zu...  1 day ago  Up      zuul-web
dbf7ec733bb2  quay.io/software-factory/zuul-executor-sf38:10.0.0-1      /usr/local/bin/zu...  1 day ago  Up      zuul-executor
ef8ac8885be9  quay.io/software-factory/zuul-scheduler-sf38:10.0.0-1     /usr/local/bin/zu...  1 day ago  Up      zuul-scheduler
bc27854b990b  quay.io/software-factory/nodepool-launcher-sf38:10.0.0-1  /usr/local/bin/no...  1 day ago  Up      nodepool-launcher
715dd76c2c96  quay.io/software-factory/logserver:4.9.3-2                httpd -DFOREGROUN...  1 day ago  Up      logserver
e10b920e0e60  quay.io/software-factory/purgelogs:0.2.3-1                bash -c /usr/loca...  1 day ago  Up      purgelogs
0d6f053c18a0  quay.io/software-factory/managesf-sf38:0.32.1-1           managesf.sh           1 day ago  Up      managesf
b5fd9b5f721f  quay.io/software-factory/gerrit-sf38:3.7.8-5              /bin/bash             1 day ago  Up      gerrit
b3f3901414c9  quay.io/software-factory/mosquitto:2.0.14-1               mosquitto -c /etc...  1 day ago  Up      mosquitto
da4c01b1d572  quay.io/software-factory/zookeeper:3.8.0-1                zkServer.sh start...  1 day ago  Up      zookeeper
9bf225789933  quay.io/software-factory/github-ssh-key-updater:0.0.4-1   ./github-ssh-key-...  1 day ago  Up      github-ssh-key-…
84dfca9cf748  quay.io/software-factory/keycloak:19.0.1-4                start --http-port...  1 day ago  Up      keycloak
c0b8bee2f943  quay.io/software-factory/mariadb-sf38:10.5.9-2            mysqld                1 day ago  Up      mysql
```

Show me what you got!

https://knowyourmeme.com/memes/show-me-what-you-got

A note about the future release

A move from
distro-centric approach
to
cloud-native deployment.

Coming soon!

- Th **SF-Operator**: a Zuul-based CI infrastructure for OpenShift.
- The current project status is: **Alpha** - NOT PRODUCTION READY.
- Details: https://softwarefactory-project.github.io/sf-operator/

Level 1 - Basic Install - **10/10**

- Zuul Scheduler: ✅
- Zuul Executor: ✅
- Zuul Web: ✅
- Zuul Merger: ✅
- Nodepool Launcher: ✅
- Nodepool Builder: ✅
- Zookeeper: ✅
- MariaDB: ✅
- Log Server: ✅
- Internal Config Repository, bootstrapped pipelines and default jobs: ✅

Level 2 - Seamless upgrades - **2/2**

- Operator: ✅
- Operands: ✅

Level 3 - Full Lifecycle - **3/5**

- SF 3.8.x migration ❌
- Backup: ✅
- Restore: ✅
- Rolling deployments: ❌
- Reconfiguration: ✅

Level 4 - Deep Insights - **1/3**

- Operator metrics: ❌
- Operand metrics: ✅
- Alerts: ❌ (WIP)

Level 5 - Auto pilot - **0/3**

- Auto-scaling : ❌
- Auto-healing: ❌
- Auto-tuning: ❌

Summary – Software Factory

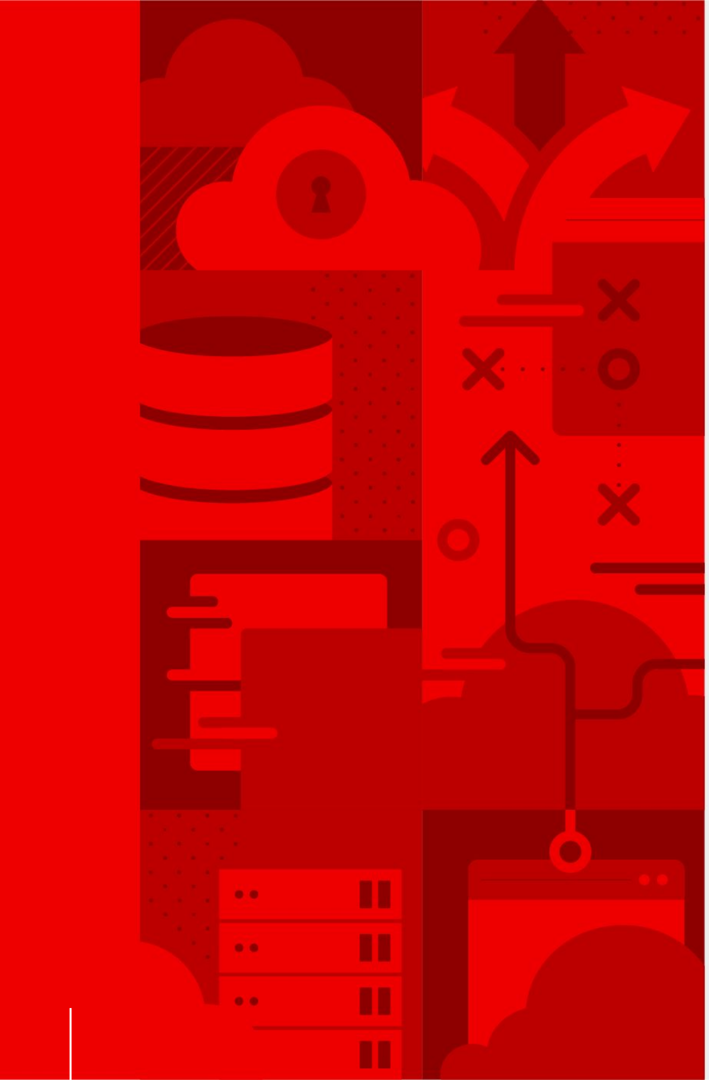
- https://www.softwarefactory-project.io
- A powerful platform to build software, working out of the box.
- Integrates services, covering each step in the production chain.
- Fast to deploy, easy to configure and manage.
  - Works exceptionally well as a sandbox playground for learning.
  - Yet, it is a fully production-ready solution as well!
- Visit our repository with examples – play and learn by yourself!

  `https://github.com/adrianfusco/openinfra2024-software-factory-playground`
- Next steps? → Try `https://www.softwarefactory-project.io/tag/zuul-hands-on-series.html`

Grab the slides!

SCAN ME

`https://datko.pl/oi-suwon.pdf`



Settle your CI environment with the Software Factory

Szymon Datko & **Adrian Fusco Arnejo**

OpenInfra Summit Asia – September 3rd, 2024

Red Hat