Welcome to the Oficial Synchro Foundation documentation

Here you will find documentation that will help you install and maintain Synchro Foundation

Foundation

This document refers to the version 24.04.10

Version compatibility

This foundation version is only compatible with `Synchro4Me` application family. If you need to run `REINF` please download the version `21.09.23.1` .

Developers: about foundation-multitenancy-config

This version has "breaking changes" All projects and modules using this version that has `foundation-multitenant-config` dependency should use at least the version `22.06.27` or newer. See changelog "breaking changes" for more details.

For latest recommended version check this link.

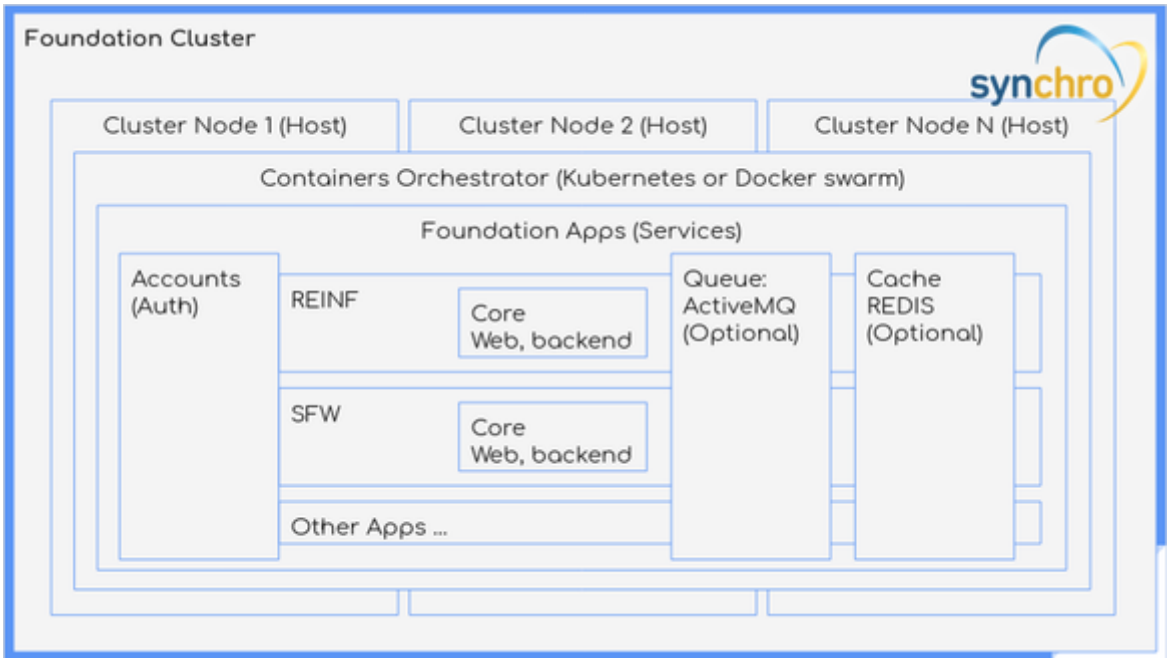
Introduction

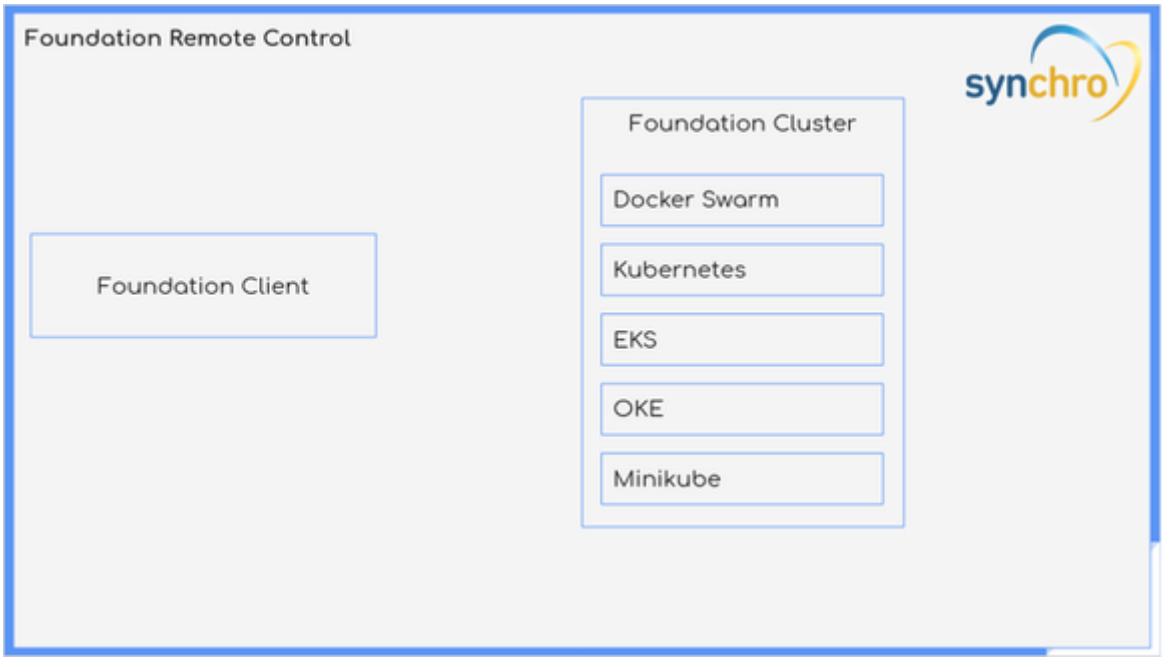
Installing and maintaining systems is complex and requires a lot of effort from the IT staff. In most cases, the scenario is composed of several systems, each with its specific requirements and needs for uptime, configuration, monitoring, fault tolerance, computational resources, etc. Due to this diversity of requirements, the IT team is obliged to train and manage each system individually, with low reuse of this knowledge and computational resources.

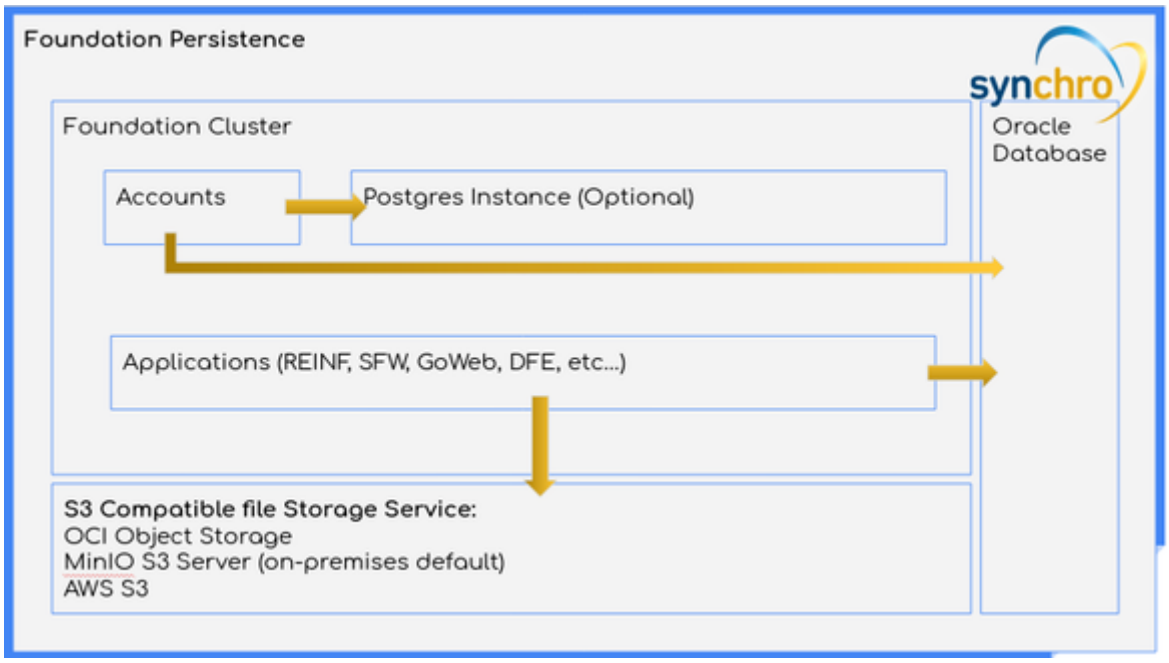
Based on this vision and always striving for excellence, Synchro presents the Foundation. As a result of the most advanced research applied in software development technology, Synchro Foundation is a middle-ware that unifies the management of all systems, facilitating installation, configuration, monitoring and updating.

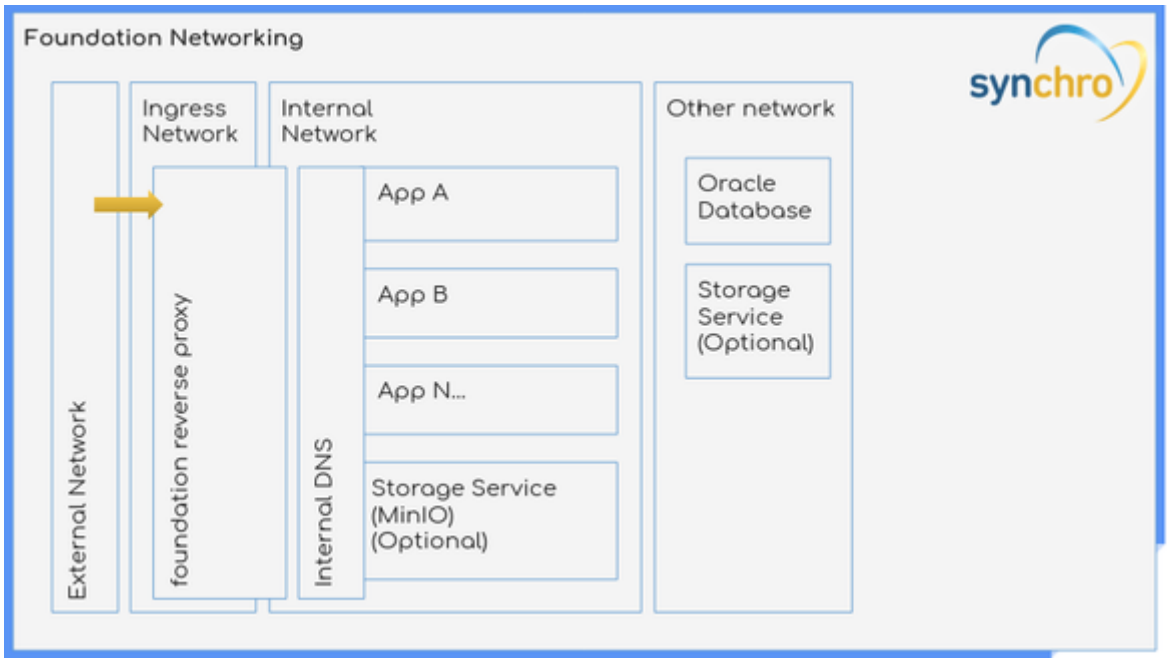
In response to the increasingly demanding scenario regarding the complexity of solutions, such as the need for updating due to legal changes, integration, management and visibility of large volumes of data, low response time in the execution of processes. Foundation is the Synchro middleware platform that meets all these needs and brings benefits to the On-Premises environment that are currently only possible in the cloud.

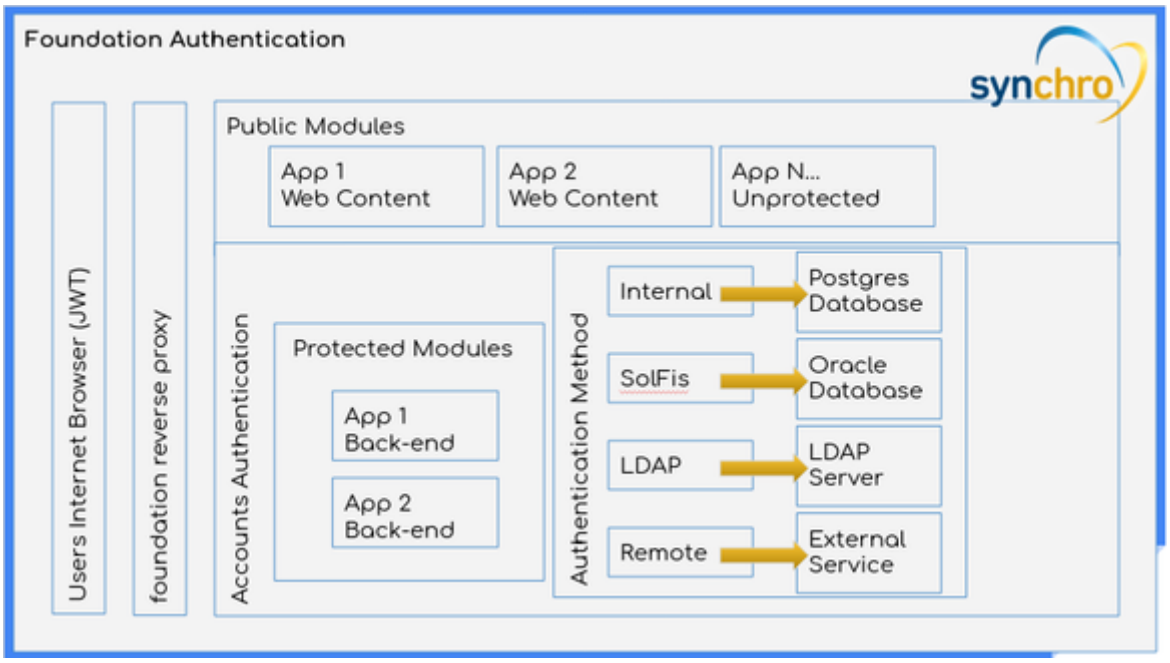
How it works?

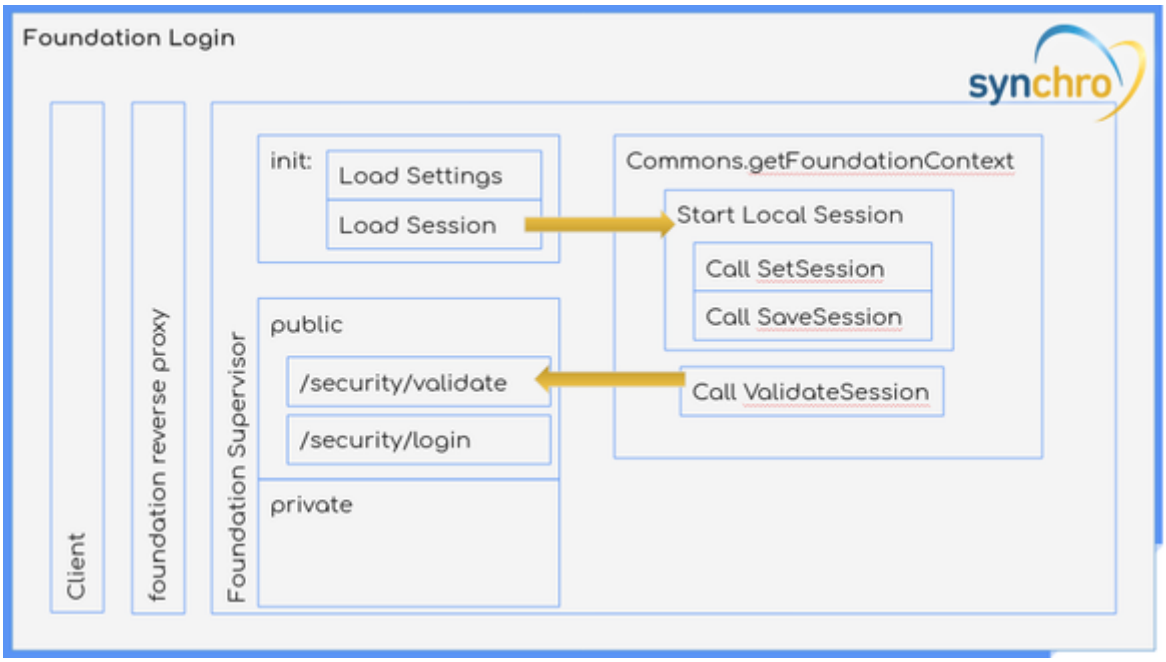

Foundation Cluster
synchro
Cluster Node 1 (Host)
Cluster Node 2 (Host)
Cluster Node N (Host)
Containers Orchestrator (Kubernetes or Docker swarm)
Foundation Apps (Services)
Accounts (Auth)
REINF
Core Web, backend
SFW
Core Web, backend
Other Apps ...
Queue: ActiveMQ (Optional)
Cache REDIS (Optional)

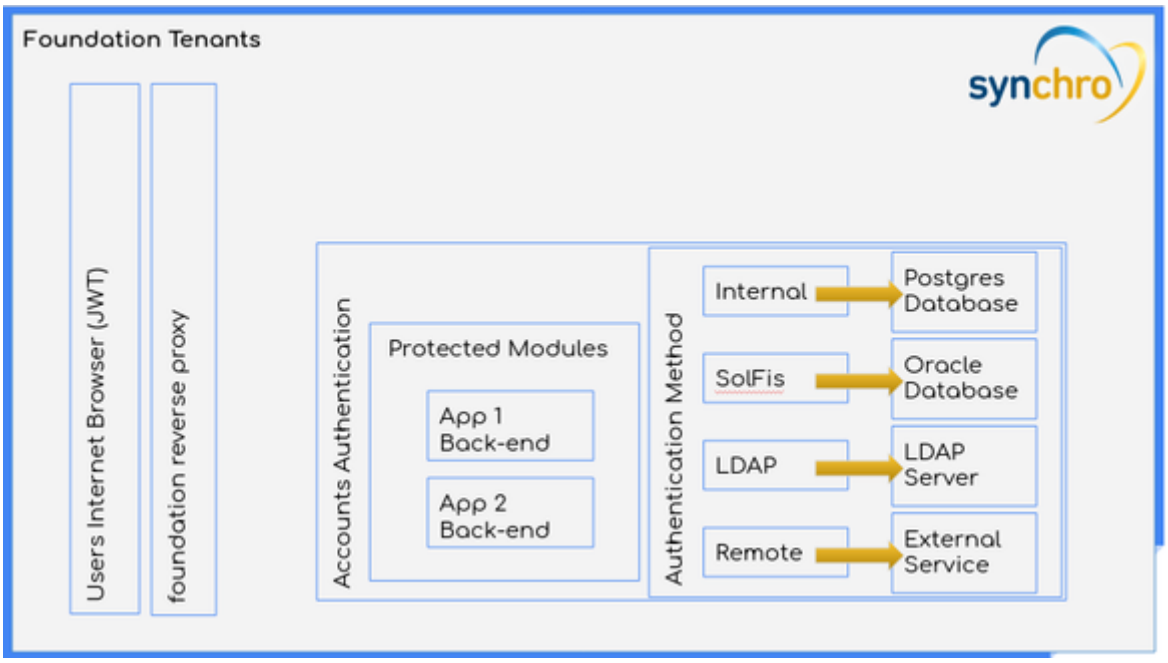

Foundation Remote Control
synchro
Foundation Client
Foundation Cluster
Docker Swarm
Kubernetes
EKS
OKE
Minikube


Foundation Persistence
synchro
Foundation Cluster
Oracle Database
Accounts
Postgres Instance (Optional)
Applications (REINF, SFW, GoWeb, DFE, etc...)
S3 Compatible file Storage Service:
OCI Object Storage
MinIO S3 Server (on-premises default)
AWS S3


Foundation Networking
synchro
External Network
Ingress Network
foundation reverse proxy
Internal Network
Internal DNS
App A
App B
App N...
Storage Service (MinIO) (Optional)
Other network
Oracle Database
Storage Service (Optional)


Foundation Authentication
synchro
Users Internet Browser (JWT)
foundation reverse proxy
Public Modules
App 1
Web Content
App 2
Web Content
App N...
Unprotected
Accounts Authentication
Protected Modules
App 1
Back-end
App 2
Back-end
Authentication Method
Internal
Postgres
Database
SolFis
Oracle
Database
LDAP
LDAP
Server
Remote
External
Service

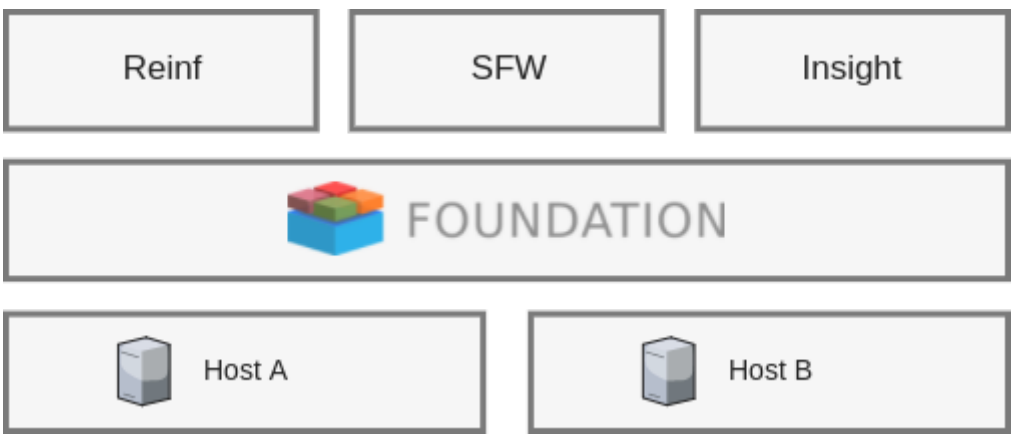

Foundation Login
synchro
Client
foundation reverse proxy
Foundation Supervisor
init:
Load Settings
Load Session
Commons.getFoundationContext
Start Local Session
Call SetSession
Call SaveSession
public
/security/validate
/security/login
Call ValidateSession
private



Foundation creates a cluster of one or more servers and exposes a unified interface to the product modules. That way, you can run more than one product on the same infrastructure. The middleware manages the distribution of the computational load among the nodes of the cluster. From this abstraction, instead of managing each product on each host individually, management becomes the middleware level, which is much simpler, as it is done through an intuitive and easy to use web interface.

The Foundation is installed on the servers forming a cluster and, from there, the product modules are installed at the Foundation, which leads to the unification of the management process of Synchro systems. It is important to highlight that this unification brings a lot of time and effort to the IT management, as it reduces the diversity of characteristics to be met and simplifies all the work of keeping the product modules in production.

In practice this means that, for example, REINF can be installed and managed on the same infrastructure as the Web Fiscal Solution (EFD PIS / COFINS and ECF), reusing computing resources and making configurations, updates and maintenance following the same procedures.



Another practical benefit is the notification of new versions of the product modules directly on the Foundation's interface and automated management of the upgrade process allowing it to be done in one click, promoting the approved version for automated production, drastically reducing the risks generated by the work manual.

The installation of Synchro Foundation is very simple, as it is done using the package managers present in all Linux distributions. After all, this is what they were designed for and work very well.

From there, all interaction of the IT professional with the Synchro Foundation occurs through a web interface. Through this interface it is possible to install modules, monitor resource consumption, obtain logs and perform updates. There is also a text mode tool that has the same functionality as the web interface.

Requirements

Hardware Requirements

Synchro Foundation requires:

- 2,6GB of RAM exclusive for foundation platform (not included applications modules requirement).
- 10GB of disk storage at Root Volume (`/`). Currently Synchro Foundation takes 4GB when installed.
- 2G of disk storage at the foundation `$USER` folder (usually `/root/.foundation`).
- Disk storage on a volume of your choice: Check the Synchro4me requirements.

Sudo access

For Foundation installation and managment is mandatory sudo/root access.

Disk Storage

1. If you are making the lite installation 8GB is enough for the Foundation itself.
2. Disk Storage requirement may vary based on your use and how many Synchro Apps you have running. Please check application minimun requirements.
3. If you don't have 2G in `/root/.foundation` folder, you can create a symbolic link to any place where you have this space. It is used to decompress `tar.gz` `*.module` files via commandline. It's necessary in installation. When using foundation-view(web) you don't need this.

Warning

It's recommended a dedicated server to host Foundation (can be a virtual server).

Access requirements

Grant access to links below during the installation or upgrading of Synchro Foundation:

- Foundation registry: `https://foundationregistry.synchro.com.br:443`
- Rancher: `rpm.rancher.io`

- K3s: `get.k3s.io`
- K3s - Update: `update.k3s.io`
- K3s - AWS: `k3s-ci-builds.s3.amazonaws.com`
- GitHub: `github.com`

Software Requirements

Linux Distribution

- CentOS 7.4 64-bit (or newer)
- Oracle Linux 8.6 64-bit (or newer)
- RHEL 7.9 64-bit (or newer)
- SUSE (NOT Supported)
- Ubuntu 22.04.1 LTS 64-bit (or newer)

Any

About the Linux disto, the important thing is the container support and a kernel 3.10.1+. Because of a bug fixed in 3.10.0 some distros may have problems with synchro foundation.

How to know which Linux distribution you are using?

There is no universal way to discover you distro. Here some tips: - `cat /etc/issue` - `lsb_release -a` - Most of distros create a file in `/etc/*-release` with information you can check. - `uname -a` - `cat /proc/version` - `dmsg | head -1`

File System

Foundation is able to choose the best use of the file system according to your distro and kernel version. So it's not a requisite. You can use whatever you want, but if you use `xfs` you need the `d_type` support enabled. Foundation will create a `overlayfs` upper your file system.

Warning

The only corner case is a `xfs` file system with `d_type` disabled. Enable `d_type` on your file system to make it work.

What is `d_type`

`d_type` is the term used in Linux kernel that stands for “directory entry type”. Directory entry is a data structure that Linux kernel used to describe the information of a directory on the file system. `d_type` is a field in that data structure which represents the type of an “file” the directory entry points to. It could be a directory, a regular file, some special file such as a pipe, a char device, a socket file etc. `d_type` information was added to Linux kernel version 2.6.4. Since then Linux Filesystem started to implement it over time. However still some file system don’t implement yet, some implement it in a optional way, i.e. it could be enabled or disabled depends on how the user creates the file system.

How to know if `XFS` support d_type

To detect if your `XFS` partition supports `d_type` use the command `xfs_info` and look for `ftype=1`

OS Specific Requirements

K3s service will not run when nm-cloud-setup.service is enabled and running!

By default k3s.service has an pre requirement that checks if nm-cloud-setup.service is enabled and running in your server. nm-cloud-setup is a network manager, commonly used in cloud servers like AWS, and must to be stopped and disabled to execute the k3s.service successfully. For more details how to disable nm-cloud-setup.service see the troubleshotting section: Amazon AWS instances disable nm-cloud-setup.service to run k3s.service.

CentOS

The CentOS-extras repository must be enabled. This repository is enabled by default, but if you have disabled it, you need to re-enable it.

Oracle Linux

You need the 64-bit version of Oracle Linux 8.6 (or newer), running the Red Hat Compatible kernel (RHCK) 3.10.0-514 or newer. Older versions of Oracle Linux are not supported.

Foundation will not install on Oracle Linux with SELinux enabled!

If you have SELinux enabled and you attempt to install Synchro Foundation, you will get an error that the container-selinux package cannot be found.

Linux Kernel

Kernel 3.10.0-514 is the absolute minimum kernel version required by Foundation. Newer versions are preferred though.

Required linux kernel options:

Foundation needs kernel with: [2]

```
CONFIG_CGROUP_*=m
CONFIG_CGROUP_DEVICE=m
CONFIG_BRIDGE=m
CONFIG_AUFS_FS=m (patch from aufs3)
CONFIG_NETFILTER_XT_MATCH_ADDRTYPE=m
CONFIG_VETH=m
bridge-utils ( for CONFIG_BRIDGE kernel options )
```

> **Warning**
>
> DO NOT use built-in(e.g.: y), use modules(m) in kernel.

If you are using kernel option `CONFIG_IKCONFIG=y`, you can see the current kernel options using:

```
zcat /proc/config.gz
```

Enabling `cgroups`

Linux `cgroup` is required to control resources. It's recommended to enable cgroup v2, since this allows no-root users to use it efectivelly.

To enable `cgroup` v2, add `systemd.unified_cgroup_hierarchy=1` to the `GRUB_CMDLINE_LINUX` line in `/etc/default/grub` and run sudo update-grub.

If `grubby` command is available on your system, this step can be also accomplished with `sudo grubby --update-kernel=ALL --args="systemd.unified_cgroup_hierarchy=1"`.

Linux Packages

The Linux packages listed bellow are required by Foundation and validated by the Linux Package Manager during installation.

- container-selinux >= 2.9
- device-mapper-libs >= 1.02.90-1
- device-mapper-persistent-data

- lvm2
- /bin/sh
- iptables
- libcgroup(v2)
- systemd-units
- tar
- xz

Network

Foundation requires:

- port 80 to be opened for inbound traffic on the host.
- ip_forward active: `/proc/sys/net/ipv4/ip_forward` with value `1`

Warning

Each Synchro App has its own network requirements in terms of ports and hosts it needs to access.

Xorg server and graphical environment

Foundation server does not need any X server, but some users like to activate Xorg, with gnome or some other graphical environment. The component gnome-shell has memory leak problems that may affect foundation required resources.[3]

Additional non Synchro software

Keep in mind that you need to provide additional requirements for any other software installed in foundation server.

1. The Synchro foundation is a modular platform. Therefore, the memory and disk usage depends on the modules you are using.
2. By default the supported distros already have this kernel config flags.
3. https://gitlab.gnome.org/GNOME/gnome-shell/issues/64.

Foundation Downloads

Version compatibility

This foundation version is only compatible with `Synchro4Me` application family. If you need to run `REINF` please download the version `21.09.23.1`.

Check the downloaded file

Please, check the `md5sum` after the image download. This way you can avoid corrupted files.
`asciinema(../assets/md5sum.asciinema)`

What to download

- If you are installing a new environment or upgrading from a previous version after v1.3.4, the full `rpm` for your distro file is enough.
- If you are a developer, or a advanced user, you can optionally download the foundation client binary and the module files of your choice.

Command line client

Foundation client 24.04.10

```
Application:    foundation client
Version:        24.04.10
Size:           45810865 bytes
Modified:       2024-04-11 01:48:21.957837760 +0000
md5sum:         7fbb9f6e6432ac9dba19ef8bd4395d6c
```

RPM files

Foundation 24.04.10 for Oracle Linux 9 (Compatible: Red Hat 9) - Lite version (for online installation)

```
Size:      8510452 bytes
Modified:  2024-04-11 01:37:02.811348576 +0000
md5sum:    13ec4ebd4438f40e1183a627956fb903
```

Foundation 24.04.10 for SUSE Linux Enterprise 12 - Lite RPM (for online installation)

```
Size:      8510584 bytes
Modified:  2024-04-11 01:26:51.455748414 +0000
md5sum:    b4d6eb89f2d4b0d14fd2081febd6faa6
```

Foundation 24.04.10 for CentOS 7 (Compatible: Red Hat/Fedora/Oracle Linux) - Lite version (for online installation)

```
Size:      8510464 bytes
Modified:  2024-04-11 01:17:59.406030882 +0000
md5sum:    70c9c10d493fd8b28daf88a6401bea16
```

Foundation 24.04.10 for SUSE Linux Enterprise 12 - Full RPM (for offline installation)

```
Size:      1600622532 bytes
Modified:  2024-04-11 01:36:51.487346265 +0000
md5sum:    1c145bfddb1596ebcab8aa549f99abce
```

Foundation 24.04.10 for Oracle Linux 9 (Compatible: Red Hat 9) - Full RPM (for offline installation)

```
Size:      1600622520 bytes
Modified:  2024-04-11 01:47:09.790806753 +0000
md5sum:    b11975e64f201110cbe6d1518f07bc69
```

Foundation 24.04.10 for CentOS 7 (Compatible: Red Hat/Fedora/Oracle Linux) - Full RPM (for offline installation)

```
Size:      1600622532 bytes
Modified:  2024-04-11 01:26:41.631767789 +0000
md5sum:    29b770c65e5233fb740f912da965af2f
```

Synchro Foundation Modules

Foundation Application Modules are distributed in two ways.

a) A simple small file with instructions for foundation to download the app via internet;

b) A full application file, for closed onpremises environments with no internet access;

You can download the latests versions of foundation modules here:

foundation/authlayer:24.04.10

foundation-authlayer-24.04.10-full.module

```
Application:    foundation
Module:         authlayer
Version:        24.04.10
Content:        Foundation Module for Closed OnPremises Environment(with no internet access).
Size:           167443014 bytes
Modified:       2024-04-11 01:13:54.824494164 +0000
md5sum:         6b95934f019aeaa055ce19e60927c2b1
```

foundation-authlayer-24.04.10.module

```
Application:    foundation
Module:         authlayer
Version:        24.04.10
Content:        Foundation Module for environments with internet access.
Size:           873 bytes
Modified:       2024-04-11 01:13:54.824494164 +0000
md5sum:         3ddfb845bab13738aa7f2d256279b2e6
```

foundation/certificates:24.04.10

foundation-certificates-24.04.10-full.module

```
Application:    foundation
Module:         certificates
Version:        24.04.10
Content:        Foundation Module for Closed OnPremises Environment(with no internet access).
Size:           158322098 bytes
Modified:       2024-04-11 01:14:09.956301072 +0000
md5sum:         475797842f8c7db1f014b67c0e1193a6
```

foundation-certificates-24.04.10.module

```
Application:    foundation
Module:         certificates
Version:        24.04.10
Content:        Foundation Module for environments with internet access.
Size:           561 bytes
Modified:       2024-04-11 01:14:09.956301072 +0000
md5sum:         21328b14df5ec74dab9dcb34918392c3
```

foundation/engine:24.04.10

foundation-engine-24.04.10-full.module

```
Application:   foundation
Module:        engine
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          218205551 bytes
Modified:      2024-04-11 01:14:52.007796136 +0000
md5sum:        daa65007389e974d86c638bf38d086d9
```

foundation-engine-24.04.10.module

```
Application:   foundation
Module:        engine
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          612 bytes
Modified:      2024-04-11 01:14:52.007796136 +0000
md5sum:        bcfd68206971bcd7a36d52c8484aeadb
```

foundation/keycloak:24.04.10

foundation-keycloak-24.04.10-full.module

```
Application:   foundation
Module:        keycloak
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          466932587 bytes
Modified:      2024-04-11 01:15:12.607564940 +0000
md5sum:        64809dc0b477027729d34f19db7befd1
```

foundation-keycloak-24.04.10.module

```
Application:   foundation
Module:        keycloak
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          290 bytes
Modified:      2024-04-11 01:15:12.611564897 +0000
md5sum:        58e17bb9e6aaa498c895002252ada813
```

foundation/licenses:24.04.10

foundation-licenses-24.04.10-full.module

```
Application:   foundation
Module:        licenses
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          157657561 bytes
Modified:      2024-04-11 01:15:27.739401543 +0000
md5sum:        e74c5446713c400705865e041daad49d
```

foundation-licenses-24.04.10.module

```
Application:   foundation
Module:        licenses
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          371 bytes
Modified:      2024-04-11 01:15:27.739401543 +0000
md5sum:        0b8815eea098e3df9aa1383b99c7a26e
```

foundation/logs:24.04.10

foundation-logs-24.04.10-full.module

```
Application:   foundation
Module:        logs
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          104803859 bytes
Modified:      2024-04-11 01:15:41.231260304 +0000
md5sum:        54593e4201cc361d435cfa19ac6e40e9
```

foundation-logs-24.04.10.module

```
Application:   foundation
Module:        logs
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          247 bytes
Modified:      2024-04-11 01:15:41.235260263 +0000
md5sum:        1671fd490dce5d8ddec829d623d140b0
```

foundation/monitor:24.04.10

foundation-monitor-24.04.10-full.module

```
Application:   foundation
Module:        monitor
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          257192970 bytes
Modified:      2024-04-11 01:16:08.214989963 +0000
md5sum:        f8e83c3341e69426eea14996a0383482
```

foundation-monitor-24.04.10.module

```
Application:   foundation
Module:        monitor
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          631 bytes
Modified:      2024-04-11 01:16:08.214989963 +0000
md5sum:        8deb21274ab8ea0f681da47e7e130726
```

foundation/postgres:24.04.10

foundation-postgres-24.04.10-full.module

```
Application:   foundation
Module:        postgres
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          31151894 bytes
Modified:      2024-04-11 01:16:12.830945292 +0000
md5sum:        0c02015c4bb19e892e390d03e4987ab3
```

foundation-postgres-24.04.10.module

```
Application:   foundation
Module:        postgres
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          186 bytes
Modified:      2024-04-11 01:16:12.830945292 +0000
md5sum:        9314f0c2ed1c1c04ee398f8a8a232c55
```

foundation/view:24.04.10

foundation-view-24.04.10-full.module

```
Application:   foundation
Module:        view
Version:       24.04.10
Content:       Foundation Module for Closed OnPremises Environment(with no internet access).
Size:          24826537 bytes
Modified:      2024-04-11 01:17:34.906222331 +0000
md5sum:        9b4cc96cdd7eac4a0127f00e6b81a074
```

foundation-view-24.04.10.module

```
Application:   foundation
Module:        view
Version:       24.04.10
Content:       Foundation Module for environments with internet access.
Size:          220 bytes
Modified:      2024-04-11 01:17:34.906222331 +0000
md5sum:        b2ca4127ef848d2ae4aa3d6e477c2431
```

Synchro Foundation Applications

REINF, ISS, SOLFIS and other Synchro Softwares are available as foundation modules at Synchro's products page for download.

Install Synchro Foundation

Supported Platforms

For hardware and platform details, please refer to Requirements.

Updating from previous versions

If you already installed previous versions of Synchro Foundation and you are **updating to newer versions**, there are a few steps you must execute before proceed:

1. Execute the command to stop Foundation: `foundation down`
2. Remove previous installation: `yum remove synchro-foundation`

or update mannualy.

Video

Installing Foundation

Checking your distribution

To check and make sure what is yout linux distro, run:

```
cat /etc/os-release
```

Red Hat Enterprise Linux

Red Hat requires subscription and internet access to install/upgrade packages. CentoOS is a community distro Red Hat compatible.

If the host does not have internet access or has expired subscription, you may have to download the package manually at CentOS repo.

Considerations in adopting RHEL 9.

1. Change user to sudo

```
sudo su -
```

2. Install required packages

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

3. Enable RHEL extras or EPEL for RHEL 8 and 9.
   For RHEL 7 or later:

```
yum-config-manager --enable rhel-7-server-extras-rpms
```

   Depending on cloud provider, you may also need to enable another repository. For AWS:

```
yum-config-manager --enable rhui-REGION-rhel-server-extras
```

   For RHEL 8:

```
subscription-manager repos --enable codeready-builder-for-rhel-8-$(arch)-rpms
```

```
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   For RHEL9:

```
subscription-manager repos --enable codeready-builder-for-rhel-9-$(arch)-rpms
```

```
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

4. Update the yum package index.
   For RHEL 7 or later:

```
yum makecache fast
```

   For REHL 8 or newer:

```
yum update
```

5. Download Synchro Foundation RPM.
6. Install Synchro Foundation

```
yum install <FOUNDATION-FILE>.rpm
```

Problem: libcgroup RHEL 9

```
[...]
Problem: conflicting requests
- nothing provides libcgroup needed by synchro-foundation-23.08.11-7dd20394.x86_64
```

Check update

CentOS

1. Change user to sudo

```
sudo su -
```

2. Install required packages

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

3. Update the yum package index.

```
yum makecache fast
```

4. Download Synchro Foundation RPM.
5. Install Synchro Foundation

```
yum install <FOUNDATION-FILE>.rpm
```

Ubuntu

1. Change user to sudo

```
sudo su -
```

2. Install required packages

```
apt-get install libltdl7
```

You need to install it before proceeding:

```
apt-get install alien
```

Distro: "Ubuntu 14.04 LTS"?

```
apt-get install libsystemd-journal0
```

3. Download Synchro Foundation RPM.
4. Run following command to change the format of the package.

```
alien <FOUNDATION-FILE>.rpm
```

5. Install `.deb` Synchro Foundation

```
dpkg -i <FOUNDATION-FILE>.deb
```

Version

Notice that Foundation works with two different versions of Ubuntu: Artful and Trusty. Make sure to download the right version for your system.

Ubuntu 16.04 LTS

This version of Ubuntu will very likely raise the following error:

```
Error: Failed to setup foundation config (code 4006) Synchronizing state of docker.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable docker
```

A workaround is to run the following command: `mv /etc/init.d/ /etc/systemd/system`

Oracle Linux

1. Change user to sudo

```
sudo su -
```

2. Install required packages

```
yum install libtool-ltdl
```

3. Download Synchro Foundation RPM.

4. Install Synchro Foundation

```
yum install <FOUNDATION-FILE>.rpm
```

Observation

In case you has any issues with the installation, please take a look at troubleshooting page.

Installing Foundation Mannualy

In case you has any problems with your distro, you can install foundation manually, to see how follows the link.

Foundation Config

After installing Foundation by following the instructions above, let's config foundation in the next section.

Post-installation Configuration

Download and Install

- Download
- Install

Configuration

Foundation is conceived with the idea of convention over configuration. Only a very minimal setup is required before getting started.

Foundation encrypted config file.

Right after installation, Foundation creates an encrypted config file under `/etc/foundation/${PROFILE}.settings`, where ${PROFILE} is the profile name you provide in the configuration. This file specifies where Foundation keeps its data as well as some security. If removed, you will need to run the `foundation config` again.

Review this config to make sure it meets all requirements.

See requirements

Tip

Maybe is necessary to stop and disable your firewall to run foundation at on-premise installation, to do that run the command below.

```
systemctl stop firewalld && systemctl disable firewalld
```

If you prefer, create a rule at your firewall instead of disable it: Check k3s docs.

After stop firewall or create rule exception, restart k3s service it was installed:

```
systemctl restart k3s
```

Foundation has a command that must run right after it's installed. This command is responsible for setting up OS-specific things like adding Foundation as a service, you don't need to reboot the server, but if you do a server reboot, the service is re-established.

Video

Setup

Sudo access needed.

To proceed with foundation configuration, make sure you have sudo/root access by running:

```
sudo su -
```

To start the configuration type:

```
foundation config
```

Tip

It may seem confusing, but don't worry; the default configuration may be enough to get the Foundation read to go. So if you don't want to customize anything, you can keep pressing `Enter` until the last question.

Foundation update.

You must rerun the setup command to update Foundation to a new version.

Let's see all options...

Foundation Profiles

```
INFO[0000] Reading profiles from /etc/foundation/
default

QUESTION: Select your profile file (current: default):
```

The first step is to select or create a foundation profile. The Foundation profile is an encrypted file where all configs are stored. You can have multiple profiles, but on the current machine, only the `default` profile can be used to start a foundation. Others profiles can be helpful to connect and manage remote foundation installations. So in the typical scenario, when configuring a local setup in the current server, you need to hit the `Enter` key to choose the `default` profile.

Foundation Orchestrator

```
INFO[0070] Supported orchestrators:
kubernetes(k8s)

QUESTION: Orchestrator[]: kubernetes
```

Foundation has support for the container orchestrator `kubernetes`, and is already set as the default orchestrator.

Foundation Provider

```
INFO[0002] Supported platform provider:
eks - Amazon Elastic Kubernetes Service
oke - Oracle Kubernetes Engine
k3s - Lightweight Kubernetes

QUESTION: Kubernetes platform provider[k3s]:
```

An on-premise setup requires a `k3s` provider.

Foundation Namespace

Kubernetes Namespaces.

Avoid using Kubernetes Namespaces like:

NAME:

- default
- kube-system
- kube-public
- kube-node-lease

```
INFO[0025] k3s - Lightweight Kubernetes
INFO[0101] [INFO]  Using v1.25.6+k3s1 as release
...
INFO[0101] Starting basic requirements check...
WARN[0101] Sorry, Foundation is not ready to do remote check yet.
INFO[0074] Listing current Kubernetes Namespaces
```

```
NAME              STATUS   AGE
default           Active   42d
kube-system       Active   42d
kube-public       Active   42d
kube-node-lease   Active   42d

QUESTION: Foundation Namespace [foundation]:
```

The `Namespace` is a way to separate environments, so you can set up and use different configurations and applications for quality assurance and production.

Create a new namespace, for example, `foundation`, and press `Enter`.

```
QUESTION: Foundation Namespace [foundation]: foundation
INFO[0082] The namespaces [foundation] do not exist.

QUESTION: Should I create the namespace [foundation] ? (y/N)
```

Press `y` and `Enter`.

Foundation Registry

The Foundation Registry is a server that stores and release images modules for Foundation.

```
QUESTION: Foundation registry [https://foundationregistry.synchro.com.br:443]:
```

Registry test fails.

```
WARN[0027] Registry test fail: dial tcp 172.19.7.229:443: i/o timeout
```

Foundation pulls all docker images from `https://foundationregistry.synchro.com.br:443`

It's strongly recommended that you do release access to the address: `https://foundationregistry.synchro.com.br:443`

Foundation volume location

```
QUESTION: Foundation volume location [/foundation]:
```

The `Foundation volume location` is the file system path where the `Foundation` will store all k3s images, applications and data. `Foundation` will create four folders, `kubelet`, `etc-rancher`, `rancher` and `system`. The `rancher` and `etc-rancher` folder is where k3s points instead the default `/var/lib/rancher` and `/etc/rancher`, So you don't need to backup it to the `<foundation>/rancher` or `<foundation>/etc-rancher`, but if they are removed, you will need to download or load all the k3s/application images manually. The `system` folder is where `Foundation` stores all application data.

Volume Backup.

All the Foundation's data and its apps are stored at the location you provided on the "Foundation volume location" configuration. You **must** back up the `foundation/system` folder to avoid data loss.

Storage Configuration

All data in Foundation are stored in an S3-compatible service. This module is called `foundation-storage`. By default, the Foundation has a built-in storage module. But if your infra has an AWS S3 or a private S3-like MinIO server, you can customize it to use.

If you want to use the default built-in S3-compatible storage server:

```
QUESTION: Change Storage Configuration? (y/n): y

QUESTION: (Storage) Type (Local/Remote) [Local]:
```

Or, if you want to put your S3 server configuration:

```
QUESTION: Change Storage Configuration? (y/n): y
(Storage) Type (Local/Remote) [Local]: Remote
(Storage) URL [foundation-storage:9000]:
(Storage) Access Key [krBliBVTeLkXJ9z2FA1pEjdUJ1EpW82T]:
(Storage) Secret Key [ADM4Oa13UUrn5QFHGU6f4I4w6a3zjVDW]:
(Storage) Bucket [Foundation]:
```

Gateway(Reverse Proxy)

```
QUESTION: Change Gateway(Reverse Proxy) Configuration? (Current: 80 , y/N): y

QUESTION: Define new proxy port [80]:
```

Foundation provides a single entry point for applications. You can define here which port to use. The default port for web applications is 80. So it's recommended as a default.

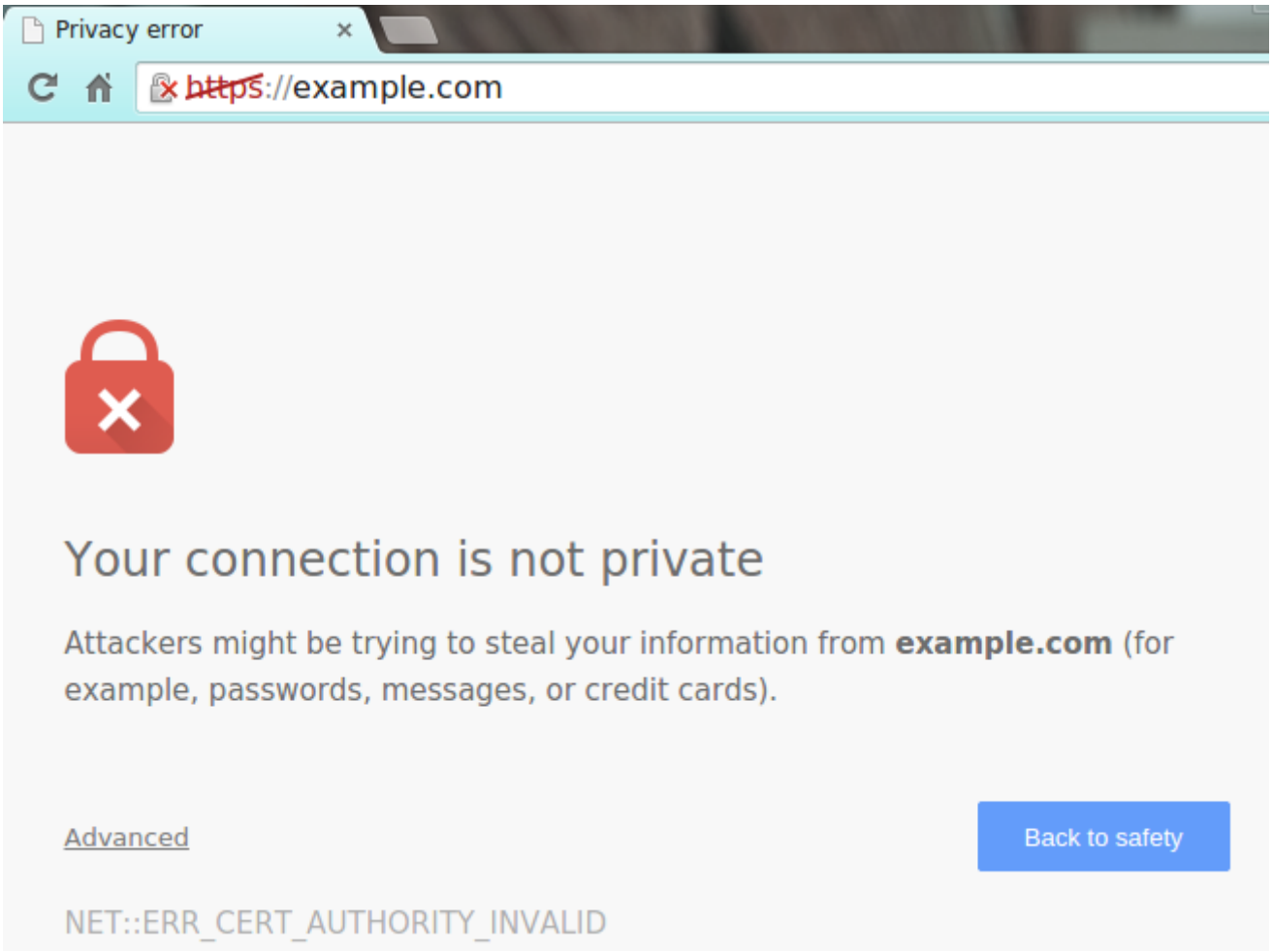
Here you can set the SSL configurations to enable Secure HTTP. The valid TLS versions are VersionSSL30, VersionTLS10, VersionTLS11, VersionTLS12, and VersionTLS13.

HTTPS (SSL/TLS)

To improve security, you can add an SSL/TLS certificate for your Foundation server to enable HTTPS.

For these steps, you need a certificate and key file at hand. If you don't have one, please see Generating TLS Self Signed Certificate and Key to generate a self-signed certificate and key for

test purposes only (non-production) assuming it will not be a secure certificate created by a certifying unit. Since this kind of certificate is not recognized as valid for most browsers, your users will be presented with an error like this:



In this case, your users will have to add an exception to the browser security configuration (available in Advanced options).

SSL certificates are responsible for the encryption between a browser and a web server. The certificate is a way to assure that the site is who it claims to be. The entity responsible for generating and signing a certificate is known as Certificate Authorities (CA).

The validation process of a certificate depends on which type of certificate a given domain has or wants to acquire. There are three types of certificate levels:

- DV: Domain Validation (an elementary certificate that validates only the server domain)
- OV: Organization Validation (validates the domain and displays some company business details)
- EV: Extended Validation (perform a full business authentication and activates green address bar)

No matter what kind of certificate you are using, self-signed or created by certifying unit, you will must need the certificate and the key in the formats below:

- A Certificate public key (.crt file) used to encrypt data on the browser side with RSA PKCS1 cryptography.
- A private key (.key file) is used to decrypt the data on the server side.

These two files will be used on the Foundation HTTPS configuration.

Certificate and Key file location

We recommend to store your `.crt` and `.key` file at you `foundation` volume path, the default is `/foundation`. In the case of creating a new folder structure inside foundation volume path, the recommendation is that this new structure belongs to `root user and group`.

Enable TLS Option

```
QUESTION: Enable TLS (y/n) [n]: y
```

Enter the TLS/SSL version

The default version is `VersionTLS12`.

```
QUESTION: [TLS] Min version [VersionTLS12]:
```

Foundation supports the following versions:

- VersionSSL30 (SSL 3.0)
- VersionTLS10 (TLS 1.0)
- VersionTLS11 (TLS 1.1)
- VersionTLS12 (TLS 1.2)

Certificate Chiper Suite.

```
QUESTION: [TLS] Cipher suite (separated by commas, without spaces) []:
```

A cipher suite is a set of algorithms that usually contains: a key exchange algorithm, a bulk encryption algorithm, and a Message Authentication Code (MAC) algorithm.

Foundation supports the following algorithms:

- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305

Certificate file

Insert the Certificate file absolute path

```
QUESTION: [TLS] Certificate file []:
```

Certificate Key file

Insert the Certificate Key file absolute path

```
QUESTION: [TLS] Key file []:
```

Domain name

```
QUESTION: Using domain name in a multitenant solution []:
```

If you have different tenants: In many multitenant, a domain name is used to identify a tenant.

Example for Domain name:

URL: `synchro.com.br`

Synchro4me DNS Requirements

Some Synchro4me applications requires an DNS configured to your server to identify your tenant. Like `foundation-prd.synchro.com.br` in this case the tenant is `FOUNDATIONPRD`. Please consult the Synchro4Me manual to see DNS requirements.

Keycloak server information

Open Source Identity and Access Management For Modern Applications and Services, check Keycloak.

Connection Type

```
QUESTION: (Keycloak) Connection Type (Local/Remote) [Local]:
```

Connection Type: Local

The first time and the Connection Type is **Local**; the properties will be filled out automatically; press `Enter` until the `Postgres` step:

Property	Value
Realm	synchro
Client ID	foundation-authentication
Client Secret	secret

URL server

```
Available network interfaces on this machine:
Interface: enp3s0       Status: up
       Address: 172.27.10.125  SubMask: 172.27.0.0/16
       Address: fe80::499b:451a:ba2f:ee81      SubMask: fe80::/64
Interface: wlp5s0       Status: up
```

```
        Address: 192.168.0.160  SubMask: 192.168.0.0/24
        Address: fe80::9a8f:4fbf:7744:6ba6      SubMask: fe80::/64

QUESTION: (Keycloak) URL [http://192.168.0.160/keycloak]:
```

- **Type: Local**

The application automatically populates the URL server field with interface IP.

Pattern: `http://subdomain.domain/keycloak`

> **Proxy TLS Enabled**
>
> If the Proxy TLS/HTTPS configuration is enabled, change the Keycloak Url Server to HTTPS instead HTTP.
>
> Pattern: `https://subdomain.domain/keycloak`

- **Type: Remote**

Get the `Frontend URL` from Realm; see Keycloak Realm

Realm

```
QUESTION: (Keycloak) Realm [synchro]:
```

Get `Realm name` from Realm; see Keycloak Realm

Client ID

```
QUESTION: (Keycloak) Client ID [foundation-authentication]:
```

Get `Client ID` from Client authentication; see Keycloak Clients authentication

Client Secret

```
QUESTION: (Keycloak) Client Secret [************]:
```

Get `Secret credentials` from client authentication; see Keycloak Clients authentication

Postgres configuration

Foundation has a Postgres module to simplify multi-tenancy management. The Authlayer module depends on it. To install and start the `foundation-postgres` module before `foundation-authlayer`.

Server

```
QUESTION: (Postgres) remote or local(embedded) server [local]?
```

- **Type: Local**

The application automatically populates the fields.

- **Type: Remote**

You'll need to fill out all the text fields.

Hostname

```
QUESTION: (Postgres) Hostname or IP []:
```

Port

```
QUESTION: (Postgres) Port []:
```

Database

```
QUESTION: (Postgres) Database []:
```

User

```
QUESTION: (Postgres) User []:
```

Password

```
QUESTION: (Postgres) Password []:
```

Load base images

```
INFO[0006] Saving settings to /etc/foundation/default.settings
INFO[0006] Applying settings for profile default
INFO[0060] Creating cronjobs.
INFO[0512] kubectl client set to development
INFO[0014] Done
```

The last step is to autoload some foundation core images. Then, you can download it from the internet. We will read all files in `/etc/foundation`, looking for docker images to load. Those resources are installed via the `rpm` file. If you installed Foundation another way, you could not have the files in the `/etc/foundation` directory.

Foundation start

After configuring the Foundation by following the instructions above, let's start Foundation in the next section.

Running Foundation

If you follow the documentation step by step, at this point you have foundation installed, but it's not running

Sudo access needed.

To proceed with foundation configuration, make sure you have sudo/root access by running:

```
sudo su -
```

Video

Start

```
foundation start
```

Verbosity level

We have decrease the verbosity level of the Foundation Start.

If you need a highest verbosity level, run the command:

```
foundation start --verbose
```

```
INFO[0000] Starting foundation 23.08.11
INFO[0050] [Foundation Core] Starting services
INFO[0055] [Foundation Module] Proxy service created
INFO[0055] [Foundation Module] Storage service created
INFO[0056] [Foundation Module] Supervisor service created
INFO[0056] [Foundation Core] Waiting until foundation core be ready...
INFO[0216] [Foundation Core] Started
INFO[0216] [All modules] Starting
INFO[0178] Starting foundation/engine:23.08.11...
INFO[0184] Started foundation/engine:23.08.11
INFO[0185] Starting foundation/postgres:23.08.11...
INFO[0187] Started foundation/postgres:23.08.11
```

```
...
INFO[0319] [All modules] Done
INFO[0319] Foundation started successfully
```

Fail to start storage

If you don't have the correct permissions, you will receive the error:

==========

INFO[0000] [Network] Creating overlay network INFO[0000] [Network] Done INFO[0000] [Foundation Core] Starting services INFO[0000] Creating Proxy service with size Pico (120 Mbytes) INFO[0003] [Foundation Module] Proxy started INFO[0003] Creating Supervisor service with size Pico (120 Mbytes) INFO[0006] [Foundation Module] Supervisor started INFO[0006] Creating Storage service with size Pico (120 Mbytes)

Service 'foundation-storage' is slow at starting and is not responding yet. Do you want to wait? (y/n): n

Foundation start process aborted

==========

In this case, try again as root (sudo) or fix the permissions.

Access Web App

1. Access Keycloak configuration.
2. Access View.

Monitor startup

You can monitor foundation startup using commands like:

```
k3s kubectl get pods
```

```
NAME                                  READY   STATUS    RESTARTS   AGE
foundation-storage-6b985b7c76-wrs4d          1/1     Running   0          5d
foundation-supervisor-847cd77d57-qxdq8       1/1     Running   0          5d
foundation-engine-7b8c75665b-s7p5g           1/1     Running   0          5d
foundation-postgres-78598ffbcd-hzclt         1/1     Running   0          5d
foundation-proxy-d5c7874b-244lk              1/1     Running   1          5d
foundation-keycloak-6b57cbfd57-8dgkp         1/1     Running   0          5d
foundation-view-8cfcd954f-m8s74              1/1     Running   0          5d
foundation-certificates-7b6f4c6df4-f9mct     1/1     Running   0          5d
```

```
foundation-licenses-756cf8dbff-4dwxq          1/1     Running   0          5d
foundation-logs-65b945c6d6-nzjnp              1/1     Running   0          5d
```

Additional modules

The foundation core modules are: `storage`, `proxy` and `supervisor`. When they are running, you are ready to add and start other modules.

But you will need some additional modules to support applications.

- Engine
- Postgress
- Authlayer
- View
- Certificates
- Logs
- Licenses
- Monitor
- Keycloak

These files are automatically installed in the server `/etc/foundation` folder by the `*-full.rpm` file, but if you installed using other method than the full rpm file, you need to assure the additional modules install manually.

Each module has two files, one small and other bigger. e.g.:

```
-rw-rw-r-- 1 ggs ggs 139176319 jul 19 17:51 foundation-engine-21.07.30-full.module
-rw-rw-r-- 1 ggs ggs       435 jul 19 17:51 foundation-engine-21.07.30.module
```

To facilitate understanding of foundation module's name:

Name: foundation-engine-21.07.30.module

Description: appName-moduleName-moduleVersion.module

In this case, if your server has access to `foundationregistry.synchro.com.br`, you don't need the big file. So you can get the file without `-full` keyword. Otherwise you will need the big one.

- The big file has a full docker image and does not need to access `foundationregistry.synchro.com.br` registry to download it.

- The small file is easy to store and deploy, but at the first module start docker will download the full image from `foundationregistry.synchro.com.br`.

Foundation Engine module

There is a additional module very important to foundation. The `engine` was separated just to allow updates to be applied without affecting the foundation core.

The `engine` module needs to be installed via command-line and will not be successful in web interface (view module).

Foundation Postgres

Foundation has a postgres module to simplify multitenancy management. Authlayer module depends on it. So install and start `foundation-postgres` module before `foundation-authlayer`.

Foundation Authlayer

Foundation has a built-in authlayer management module responsible to authenticate user with Keycloak before redirect to applications. This way applications don't need to worry about authentication.

For more information, see foundation-authlayer.

Foundation View

Foundation view is a web module, after install you will be able to manage foundation using any HTTP browser.

For more information, see foundation-view.

Foundation Certificates

Foundation certificates is a certificate module, after install you will be able to manage KeyStore and TrustStore using any HTTP browser.

For more information, see foundation-certificates.

Foundation Logs

Foundation logs is a logs module, after install you will be able to view log module using any HTTP browser.

For more information, see foundation-logs.

Foundation Licenses

Licenses is a Foundation's module responsible for providing the information for Synchro customers to license the contracted products into Foundation.

For more information, see foundation-licenses.

Foundation Monitor

Unify metrics with log and trace analytics on foundation-monitor managed service. This module integrated, real-time alerting, security Dashboard and more.

- Under Construction

Foundation Keycloak

Keycloak is a open source identity and access management for modern applications and services.

For more information, see foundation-keycloak.

Other foundation modules

Each application can be additional module requirements. Please check application documentation.

Foundation Commands

Foundation provides a few command line features, so you can manage, verify and troubleshoot Foundation and apps.

The features are provided through the binary `foundation` installed on your on Linux server.

For more information, see command line.

Manually installing foundation module:

Adding a foundation module

Starting a foundation module

Who needs this section

IMPORTANT: User that currently have foundation v1.3.4 installed and want to upgrade.

If you do not have foundation v1.3.4 installed, please skip to next section.

`asciinema(../assets/migracao-1.3.4.asciinema)`

Foundation migration v1.3.4 to latest

To facilitate the version upgrade, a `bash` script was created for a single execution, in which the `data` folder is backed up to preserve the data of the applications that use `local storage` in the `Foundation` directory and a secure reinstallation is performed from the old version to the new one.

The foundation 1.3.4 is very different internally, and the configuration need to be remade. Then it's a good idea to output and save the current connection data. For this, with foundation 1.3.4 running:

```
docker ps --filter name=foundation_accounts-pg --format 'docker exec {{ .ID }} psql accounts accounts -c
"SELECT * FROM ACCOUNTS_PROVIDER;"' | sh
```

Expected output is like:

```
 provider_type |                provider_config
---------------+-------------------------------------------------
 SOLFIS        | {"url":"jdbcdb","user":"user","password":"pw"}
(1 row)
```

Migration script

Before executing the script, it is necessary to highlight a few points:

1. Have the system root user password at hand
2. If you don't already download the migration script, download the Synchro-Foundation new version(RPM file) for your environment. You can get it at: Download Foundation
3. The script performs a backup of the `local storage (minio)` of Foundation 1.3.4 to the `home` of the system, a folder will be created with the name `foundation-data-bkp-'data of the backup in format (year, month , day)'` . Exp: `foundation-data-bkp-20190605`

4. For the containment of possible problems, v1.3.4 is also completely backed up, together with images and data from the local storage if any, all content will be moved to a folder named with the prefix `-v134-bkp` in the same directory where the Foundation was installed.

Performing the migration

> **Tip**
>
> Save your LDAP and Database setup before continue, this script target to convert file structure, but tenant configuration will need to be reconfigured after install.

At the beginning of the process you will be asked for the password `sudo` so that all commands can be entered correctly, and so that there is no problem related to permission during the entire execution.

execute the steps described in install

The following instruction will appear, and you will be asked to manually enter the path and file for the update:

```
$ ##### Start new package install #####

$ Enter the file path to foundation installation file: <TYPE_THE_FOUNDATION_RPM_PATH_HERE>
```

When the installation is complete, the first configuration of the `foundation` will start, which is very similar to the old version, and which can be found in detail step by step in this link.

The following options will be requested and need to be filled:

```
$ INFO[0000] Configuring Foundation
$ Foundation Namespace []:
$ Foundation volume location []:
$ Foundation volume driver []:
$ Change Network Configuration? (eno1 - y/n): y

$ Available network interfaces on this machine:
$ eno1 172.27.11.190 172.27.0.0/16
$ eno1 fe80::2d84:982c:1a61:aa32 fe80::/64

$ (Network) Define which interface to use []:
$ (Network) Foundation services custom subnet []:
$ (Network) Foundation ingress subnet []:
$ (Network) Foundation ingress gateway []:

$ Change Proxy Configuration? (y/n): y
$ Define new proxy port []:
$ Enable TLS (y/n) []:
```

```
$ Change Storage Configuration? (y/n): y
$ (Storage) Type (Local/Remote) []:

$ INFO[0049] Applying settings
$ INFO[0118] Done
```

After finishing the configuration, permissions of some directories will be changed for the `non-root` user of the system so that any future execution / maintenance does not require this type of access.

At the end, the Foundation will start and be ready for use.

Issues downloading older reports

If you have ant issue download older reports, maybe some issue in data copy step. You can fix it with:

```
cp -r <path-to-old-foundation-1.3.4>/system/objectstore/data/reinf/* <path-foundation>/system/<namespace>/foundation/storage/
```

If the error persist contact Foundation support team

Who needs this section

IMPORTANT: User that currently have foundation version 2.0 or higher installed.

Download

1. Download latest version of Synchro Foundation Full RPM.

Simple upgrade

Important

This step only applies if the distributions that used the RPM file format.

If not, please go to next step.

Sudo access needed.

To proceed with foundation configuration, make sure you have sudo/root access by running:

```
sudo su -
```

1. Stop Foundation

```
foundation stop
```

!!! danger "Foundation version 22.11.07" `foundation stop` not works

```
run:

a. `kubectl get namespace`

b. `kubectl delete namespace <NAMESPACE>`
```

1. Install Synchro Foundation

```
yum install <FOUNDATION-FILE>.rpm
```

2. Set up Foundation

```
foundation config
```

3. The documentation for this step contains more detailed information.
4. Start Foundation

```
foundation start
```

Updating Foundation Manually

1. Create a folder to extract the content

```
mkdir foundation_rpm_files && cd foundation_rpm_files
```

2. Extract the rpm

```
rpm2cpio <PATH-TO-FILE>/<FOUNDATION-FILE>.rpm | cpio -idmv
```

3. Stop Foundation

```
foundation stop
```

4. Replace oldest Foundation command line with the new one

```
mv ./usr/bin/foundation /usr/bin/
```

   Obs: To make sure where the command line is installed, run:

```
whereis foundation
```

   Check if Foundation is works with the new version

```
foundation --version
```

5. Set up Foundation

```
foundation config
```

6. Start Foundation

```
foundation start
```

foundation-authlayer

Overview

Authlayer is a Foundation's module responsible users authentication with keycloak.

Setup

For more information, see configuration.

Developer Questions

ClientId

Don't worry about ClientId for roles.

The clientId is formed by TenantId from cookie and follow the convention: client-environment

Authorization Bearer

If you don't have a FoundationID, it's possible to pass Authorization Bearer

```
curl -X GET "http://127.0.0.1/authlayer/api/keycloak/users/{email}" \
-H "Authorization: Bearer
4b7TW-6rKvxw3_OVtpx_WiQXFXRdT7uvUufQh-1gD08LKCXOXd8SHxH9IlpzqXz6j8ADaQ" \
--compressed
```

Endpoints available for Authlayer Module

Your app can make requests to the following REST endpoints:

Actions

Clients

- External Login

Users

- Get user by Email

- Get user role mappings
- Create a new user
- Update the user
- Update password
- Redirect to keycloak my account link
- Get keycloak JWT from user

Roles

- Get all roles for the client
- Get a role by name
- Create a new role for client
- Add client-level roles to the user role mapping
- Delete client-level roles from user role mapping

External Login

Shell

```
curl -X POST "http://127.0.0.1/authlayer/external/login" \
--data-binary '{"clientId": "teste-dev", "clientSecret": "93tI3I61XYUJQQFx1I1J"}' \
--compressed
```

Success Responses


```
{"success":true,"result":"","details":null,"content":
{"access_token":"eyJhbGciOiJSUzI1NiIsInR5cCIgOi","token_type":"Bearer","refresh_token":"","expiry":""},"versior
```


Get user by Email

Code samples:

Shell

```
curl -X GET "http://127.0.0.1/authlayer/api/keycloak/users/{email}" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses


```
{
  "success":true,
  "result":null,
```

```
  "details":null,
  "content":{
    [User representation]
  },
  "version":""
}
```


Get user role mappings

Code samples:

Shell

```
curl -X GET "http://127.0.0.1/authlayer/api/keycloak/users/{email}/role-mappings" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses


```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Mappings representation]
  },
  "version":""
}
```


Mappings representation

Create a new user

Code samples:

Object Payload (JSON)

User representation

Shell

```
curl -X POST "http://127.0.0.1/authlayer/api/keycloak/users" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--data-binary '{"lastName": "new test", "email": "new.test@synchro.com.br",
"temporarypassword":"synchro#@12345"    }' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [User representation]
  },
  "version":""
}
```


Update the user

Code samples:

Object Payload (JSON)

User representation

Shell

```
curl -X PUT "http://127.0.0.1/authlayer/api/keycloak/users/{email}" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--data-binary '{"lastName": "fulano", "email": "new.test@synchro.com.br"}' \
--compressed
```

Success Responses


```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [User representation]
  },
  "version":""
}
```


Update password

Code samples:

Shell

```
curl -X PUT "http://127.0.0.1/authlayer/api/keycloak/users/{email}/credentials-reset" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":null,
  "version":""
}
```


Redirect to keycloak my account link

Code samples:

Shell

```
curl -X GET "http://127.0.0.1/authlayer/api/keycloak/users/redirect/myaccount" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses

redirect

Get keycloak JWT from user

Code samples:

Shell

```
curl -X GET "http://127.0.0.1/authlayer/api/keycloak/users/{stateId}/jwt" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses


```
{
  "success":true,
  "result":"",
  "details":null,
  "content":{
    "access_token":"eyJhbGciOiJIUzI1NiJ9...",
    "token_type":"Bearer",
    "refresh_token":"eyJhbGciOiJIUzI1NiJ9...",
    "expiry":"2022-05-17T18:05:15.953589759Z"
  },
  "version":""
}
```

Get all roles for the client

Code samples:

Shell

```
curl "http://127.0.0.1/authlayer/api/keycloak/app/{appName}/roles/list" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Role representation]
  },
  "version":""
}
```

Get a role by name

Code samples:

Shell

```
curl "http://127.0.0.1/authlayer/api/keycloak/app/{appName}/roles/{name}" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Role representation]
  },
  "version":""
}
```

Role representation

Create a new role for client

Code samples:

Object Payload (JSON)

Role representation

Shell

```
curl -X POST "http://127.0.0.1/authlayer/api/keycloak/app/{appName}/roles" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--data-binary '{"name": "foundation-dev-teste-api", "description": "foundation dev test api"}' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Role representation]
  },
  "version":""
}
```

Add client-level roles to the user role mapping

Code samples:

Object Payload (JSON)

Role representation

Shell

```
curl -X POST "http://127.0.0.1/authlayer/api/keycloak/app/{appName}/users/{email}/role-mappings" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--data-binary '[{"id": "75610180-76d9-412f-b3cd-fda3194df381", "name": "foundation-dev-teste-api"}]' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Role representation]
  },
  "version":""
}
```

Delete client-level roles from user role mapping

Code samples:

Object Payload (JSON)

Role representation

Shell

```
curl -X DELETE "http://127.0.0.1/authlayer/api/keycloak/app/{appName}/users/{email}/role-mappings" \
-H 'Cookie:FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
--data-binary '[{"id": "75610180-76d9-412f-b3cd-fda3194df381", "name": "foundation-dev-teste-api"}]' \
--compressed
```

Success Responses

```
{
  "success":true,
  "result":null,
  "details":null,
  "content":{
    [Role representation]
  },
  "version":""
}
```

Schemas

Users

Check the official Keycloak user representation

Roles

Check the official Keycloak role representation

Foundation Keycloak configuration

Keycloak is a open source identity and access management for modern applications and services.

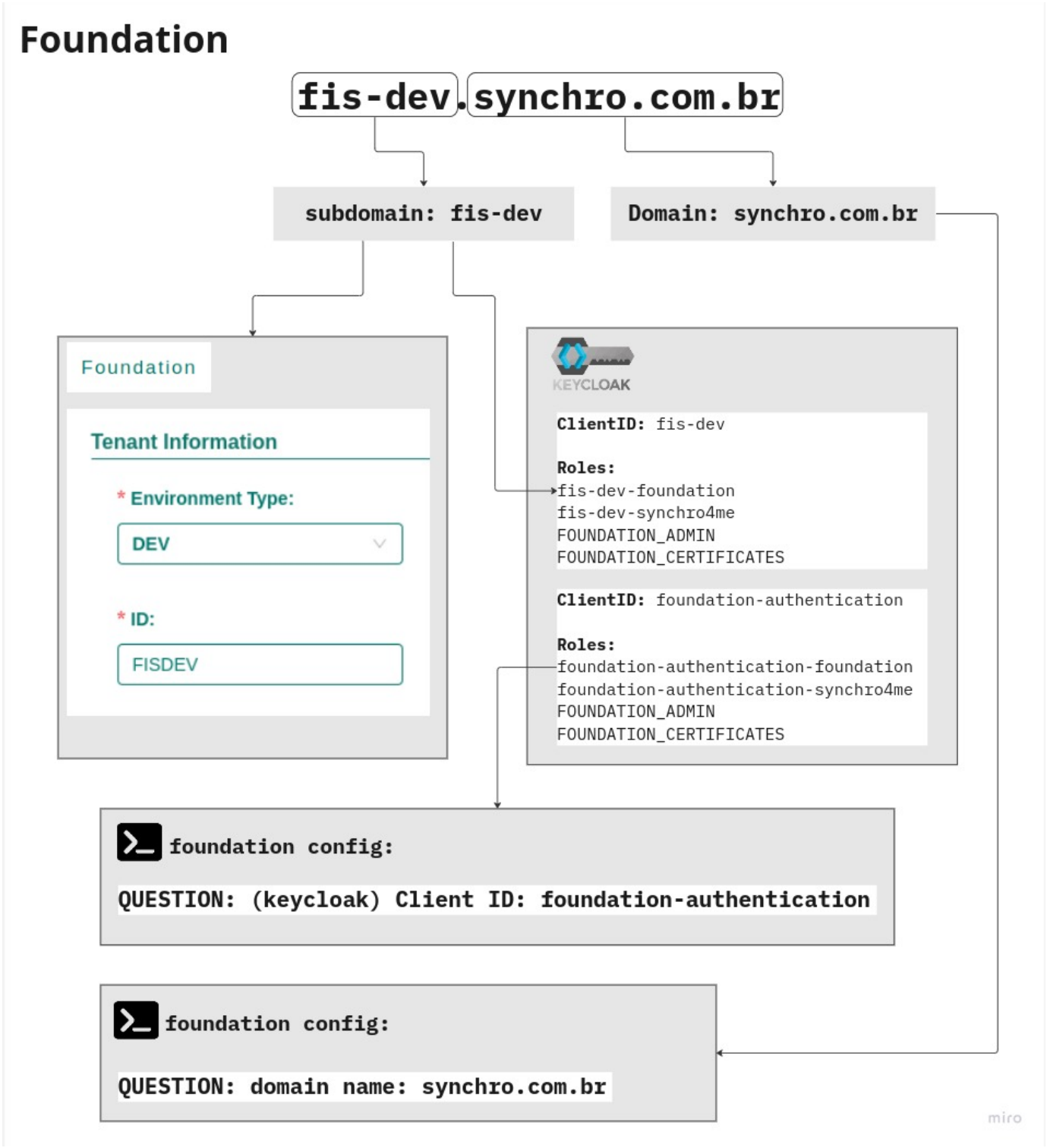
The official site of the Keycloak.

For more information about Keycloak features and concepts, see keycloak.org/documentation.

How Tenants works with Keycloak?

Example 1: Hyphens in subdomains

Parts	value
URL	fis-dev.synchro.com.br
Subdomain	fis-dev
Domain	synchro.com.br
ClientID	fis-dev
Environment Type	DEV
TenantID	FISDEV



Example 2: No hyphens in subdomains

Parts	value
URL	serverb180.john.com.br

Parts	value
Subdomain	serverb180
Domain	john.com.br
ClientID	serverb180
Environment Type	B180
TenantID	SERVERB180

On-cloud

Nothing to do.

If you have any doubts or want information you may always contact cloud team by email: `operacaocloud@synchro.com.br` .

On-premises

The Initial Setup is started automatically, check default values here

Procedure

1. Access Foundation View.
   1.1 Do Foundation Login.
   1.2 Create or check a Tenant environment.
   1.3 Create a Tenant.
2. Do Keycloak Login.
   2.1 Do Adding roles to the keycloak user.

foundation Keycloak Login

At this point, you need an administrator account that can act as a super admin with full permisions to manage all parts of Keycloak. With this account, you can log into the Keycloak Admin Console where you create realms and users and register applications that are secured by Keycloak.

For more information about Keycloak administrator, see administration guide.

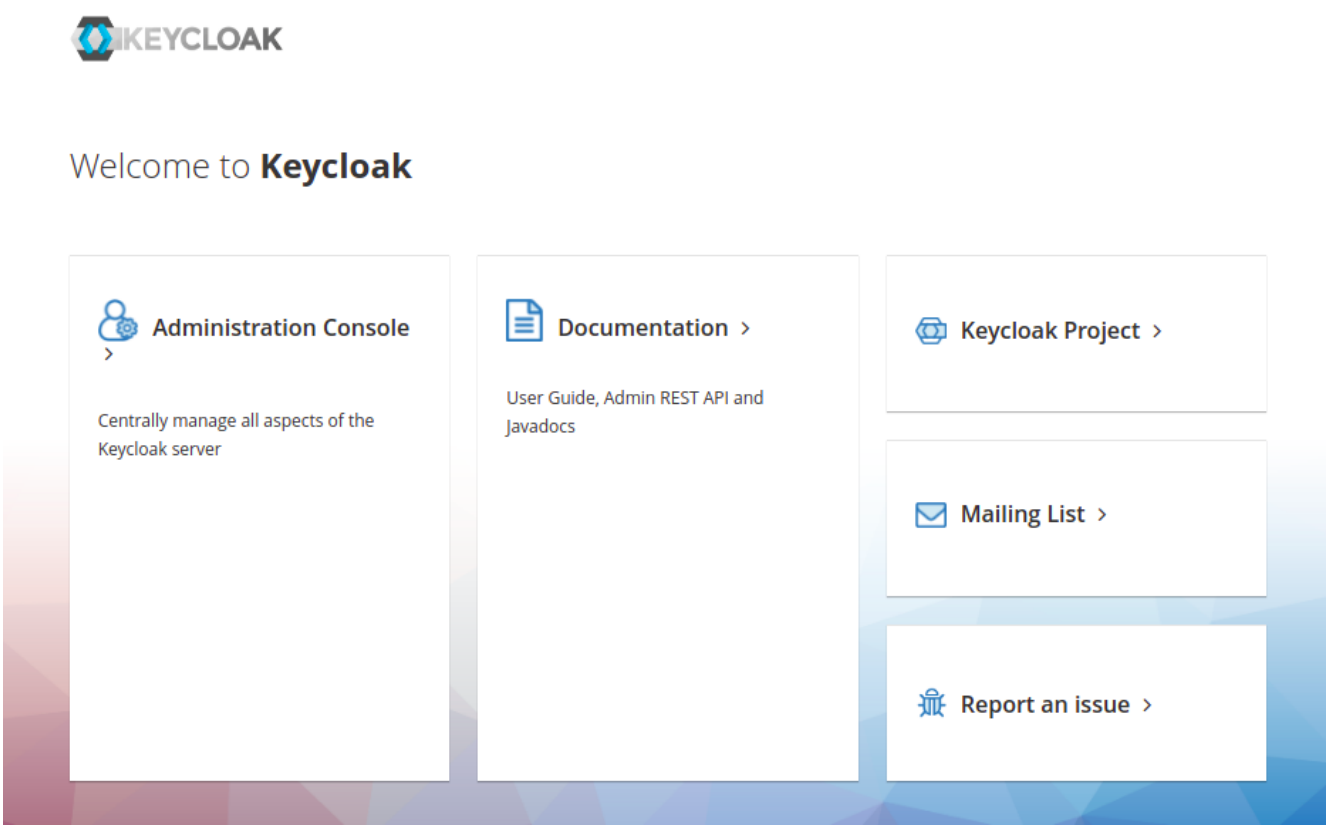
Login

Procedure

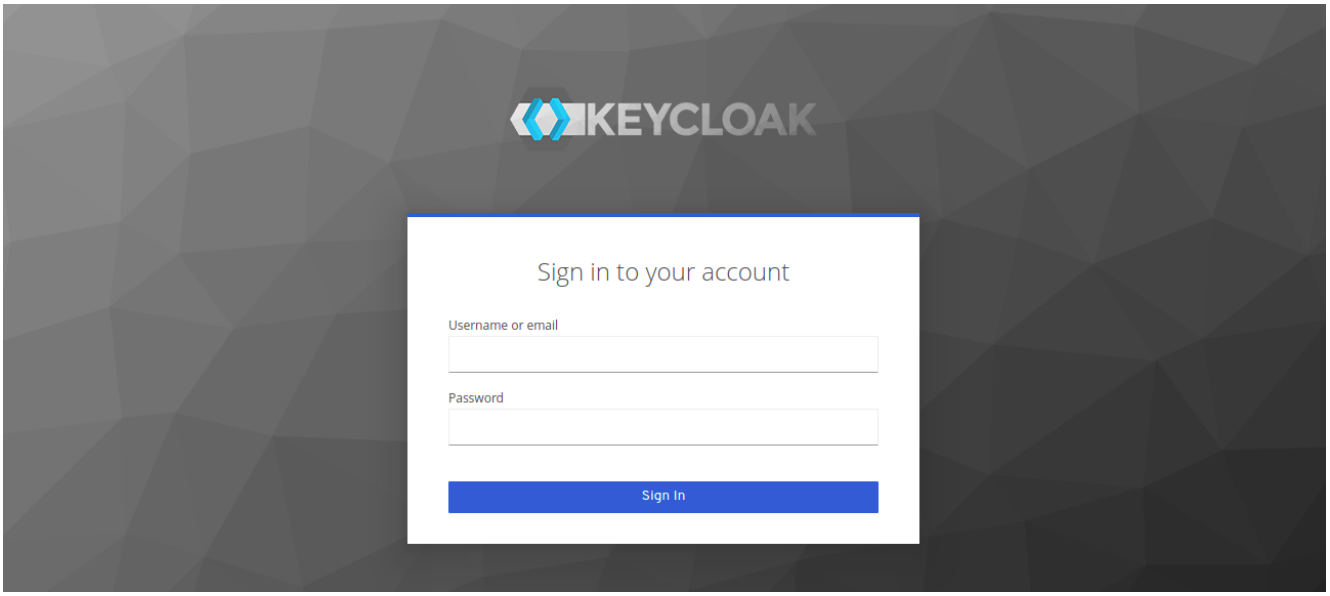
1. Access: `http://<server-ip>:<foundation-port>/keycloak`

Get link to acess keycloak view:

```
sudo foundation config --get-keycloak-link
```



2. Click the **Administration Console**



3. Do login with default keycloak user.

> **Manager user**
>
> This user there is a specific role, for more information, see keycloak user administration.

user: `adminKeycloak`

pass: `adminKeycloak`

IMPORTANT: We recommend that you change the password after login, for higher security.

Check change user password.

Creating keycloak user administration

Admin Console

Through the admin console administrators can centrally manage all aspects of the Keycloak server.

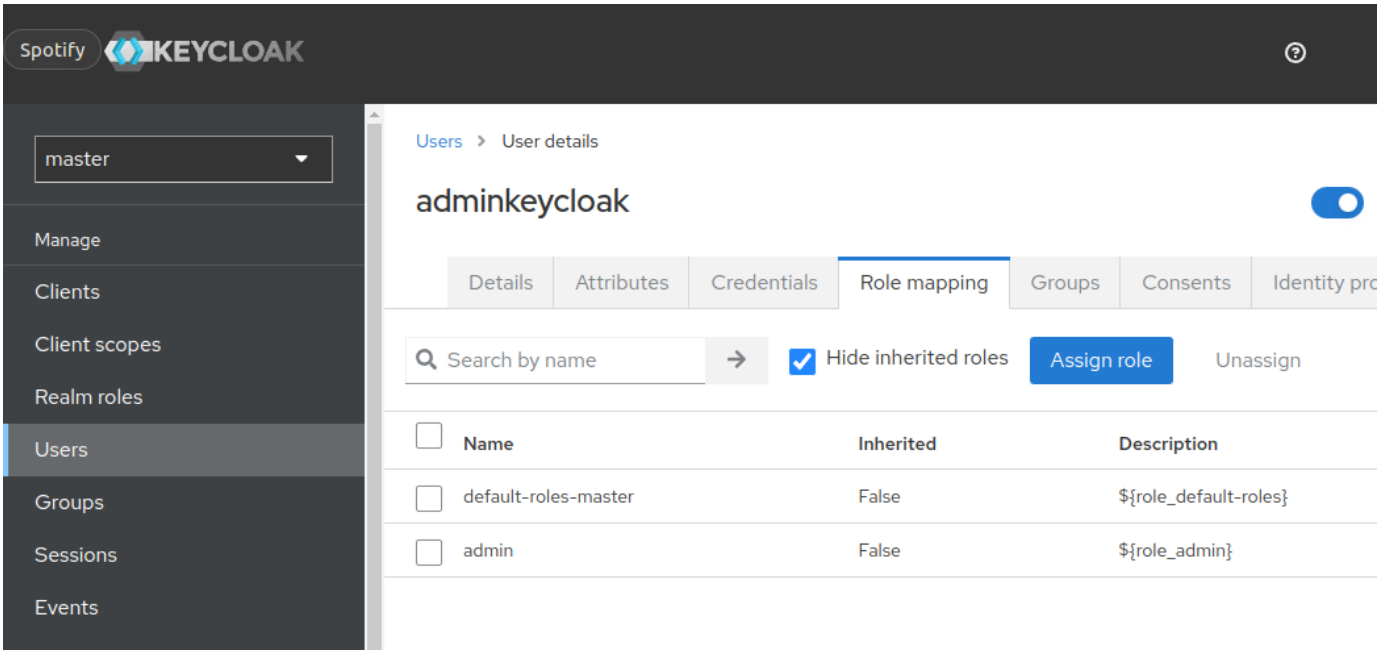
They can enable and disable various features. They can configure identity brokering and user federation.

They can create and manage applications and services, and define fine-grained authorization policies.

They can also manage users, including permissions and sessions.

Procedure

1. Select `master` realm.
2. Create a keycloak user.
3. Click the "Credentials tab" and "Set Password".
4. Click the "Role mapping tab" and assign `admin` role.



Foundation Keycloak user

Creating users

From the Admin Console, you have a wide range of actions you can perform to manage users.

Procedure

If you don't have an keycloak user account in Realm configured, you can create one now by follow steps:

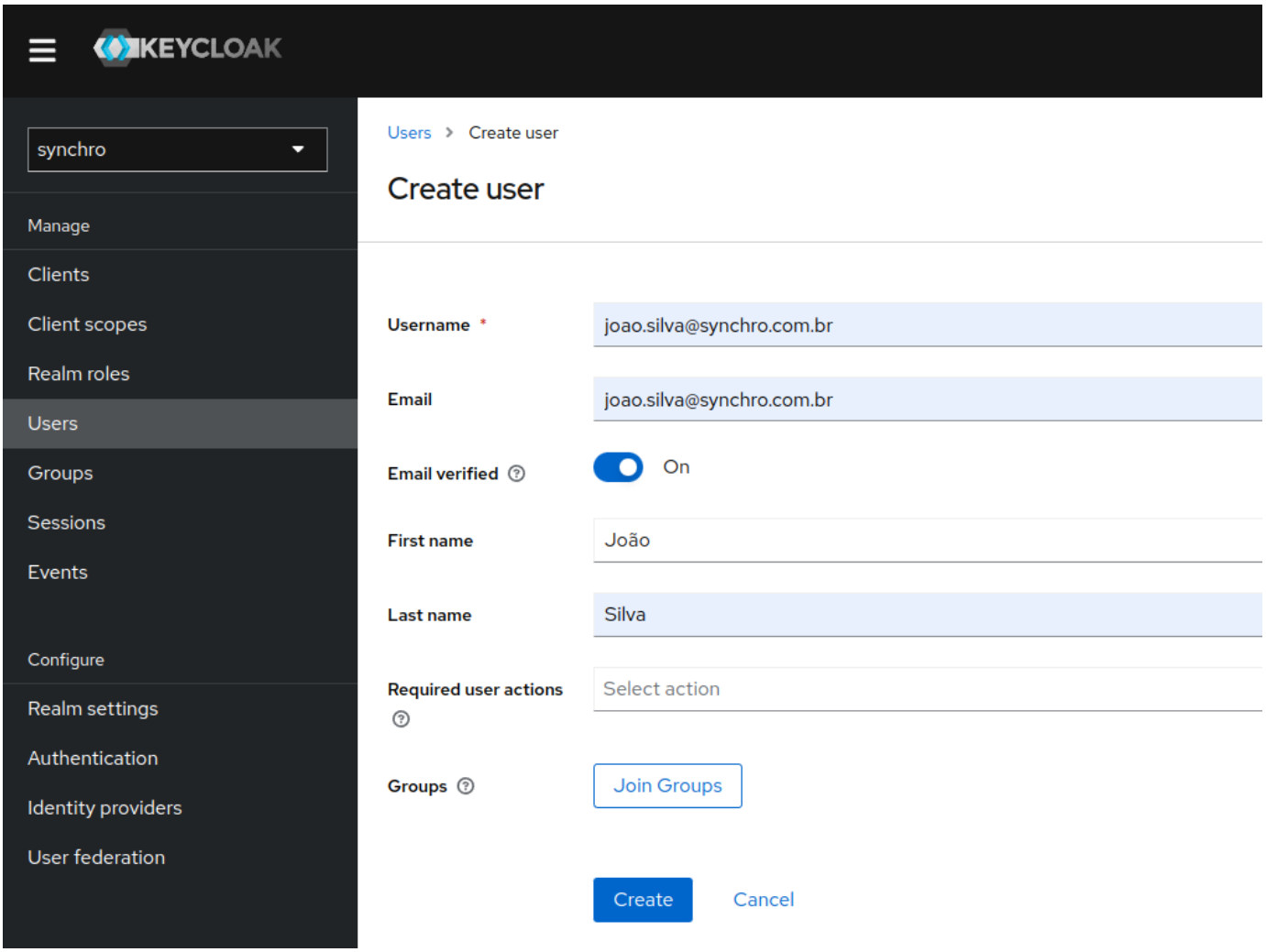
User feature details

If you need "User feature details", see keycloak user.

Username

Need Username equal email address

1. Create a keycloak user.
2. Click the "Credentials tab" and "Set Password".
3. Roles feature, check Adding roles to the user role mapping.



Change user password

Procedure

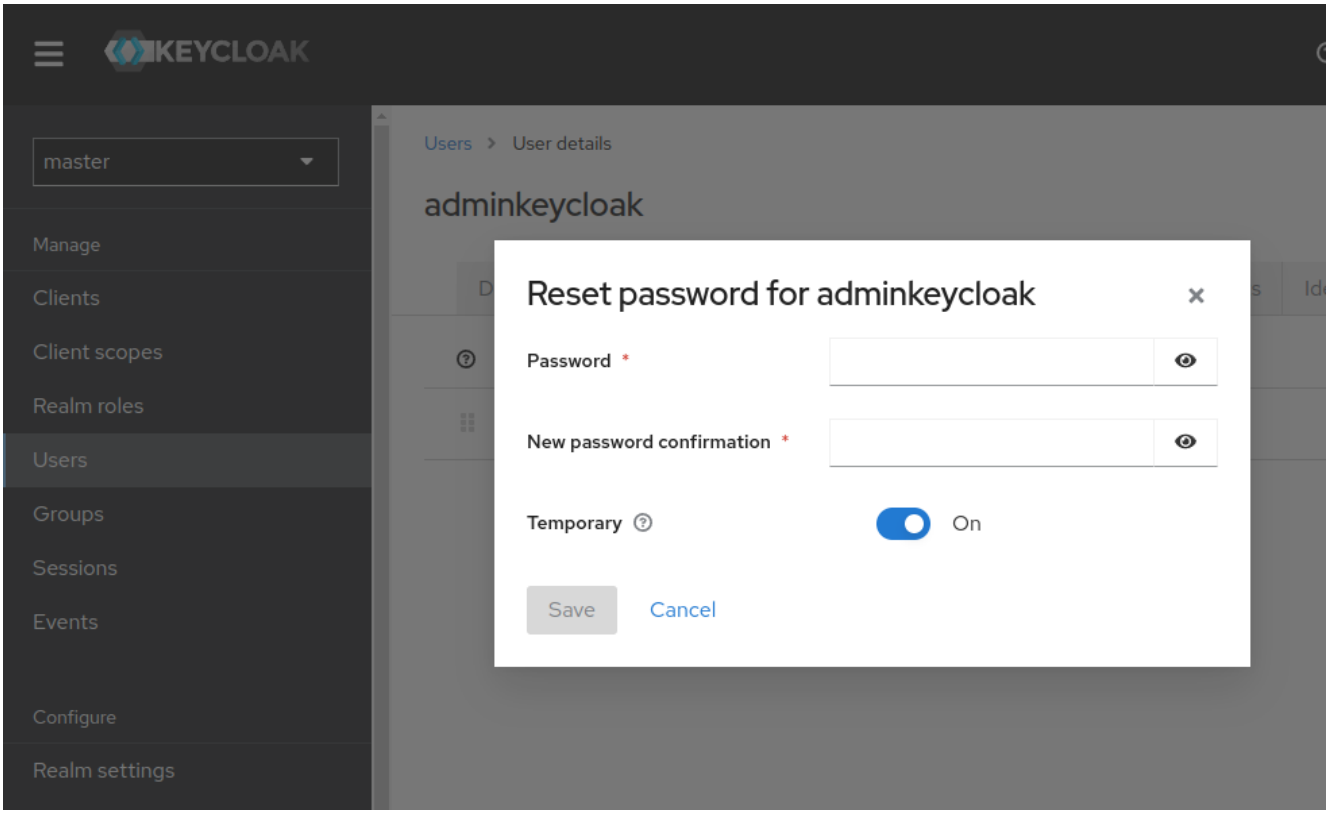
1. Select the realm.

> **Important**
>
> **AdminKeycloak User?**
>
> Select "Master" Realm
>
> Please insert a valid Email address in step 3.

2. Click "Users" in the menu. The Users page is displayed.
3. Select a user.

4. Click the "Credentials" tab.
5. Click the "Reset password" button.
6. Type a new password in the Reset Password section.

> **Temporary button**
>
> If Temporary is ON, the user must change the password at the first login. To allow users to keep the password supplied, set Temporary to OFF. The user must click Set Password to change the password.

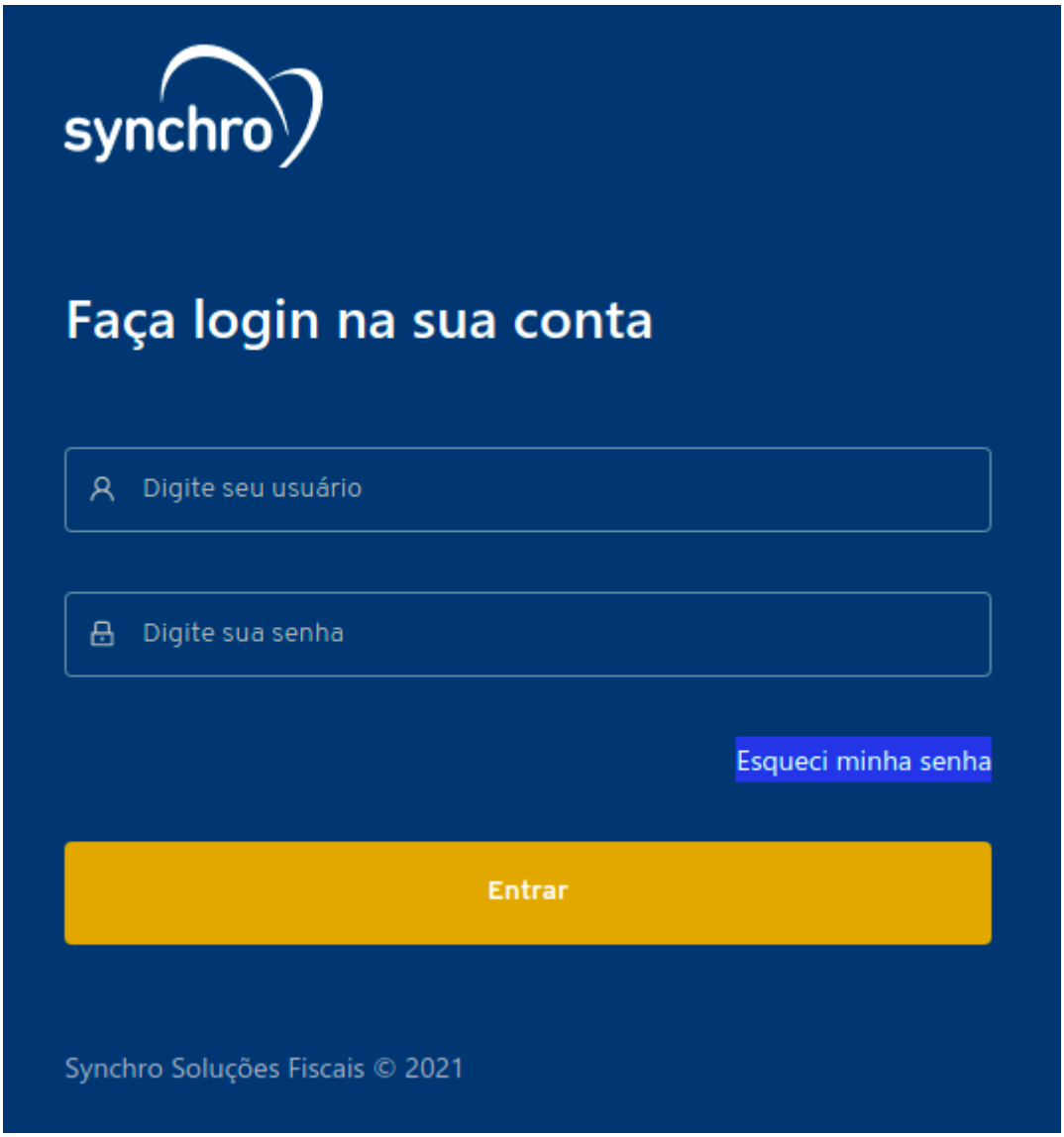


Recover password

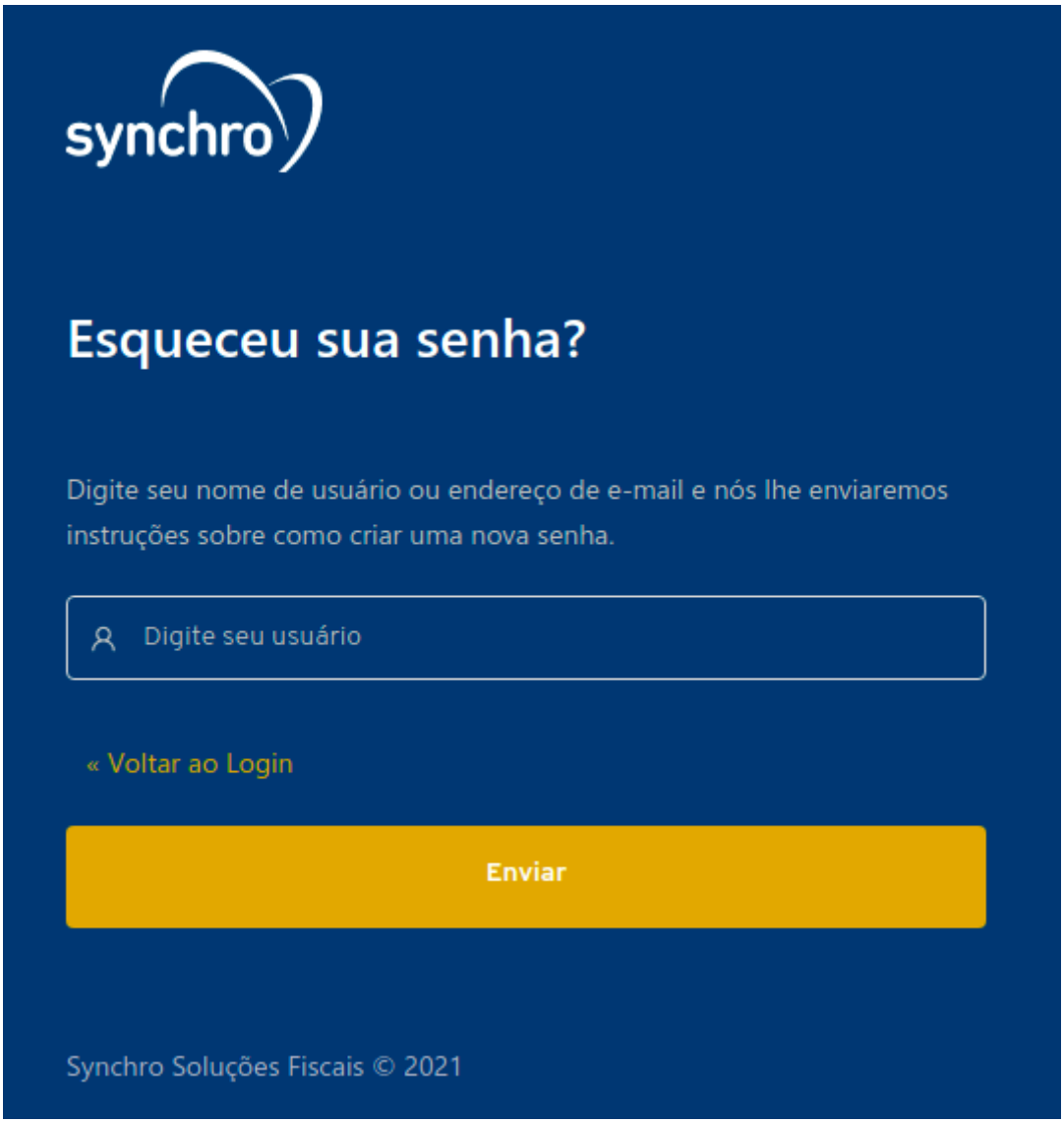
> **If it's not working**
>
> Check Keycloak Realm email settings.

Procedure

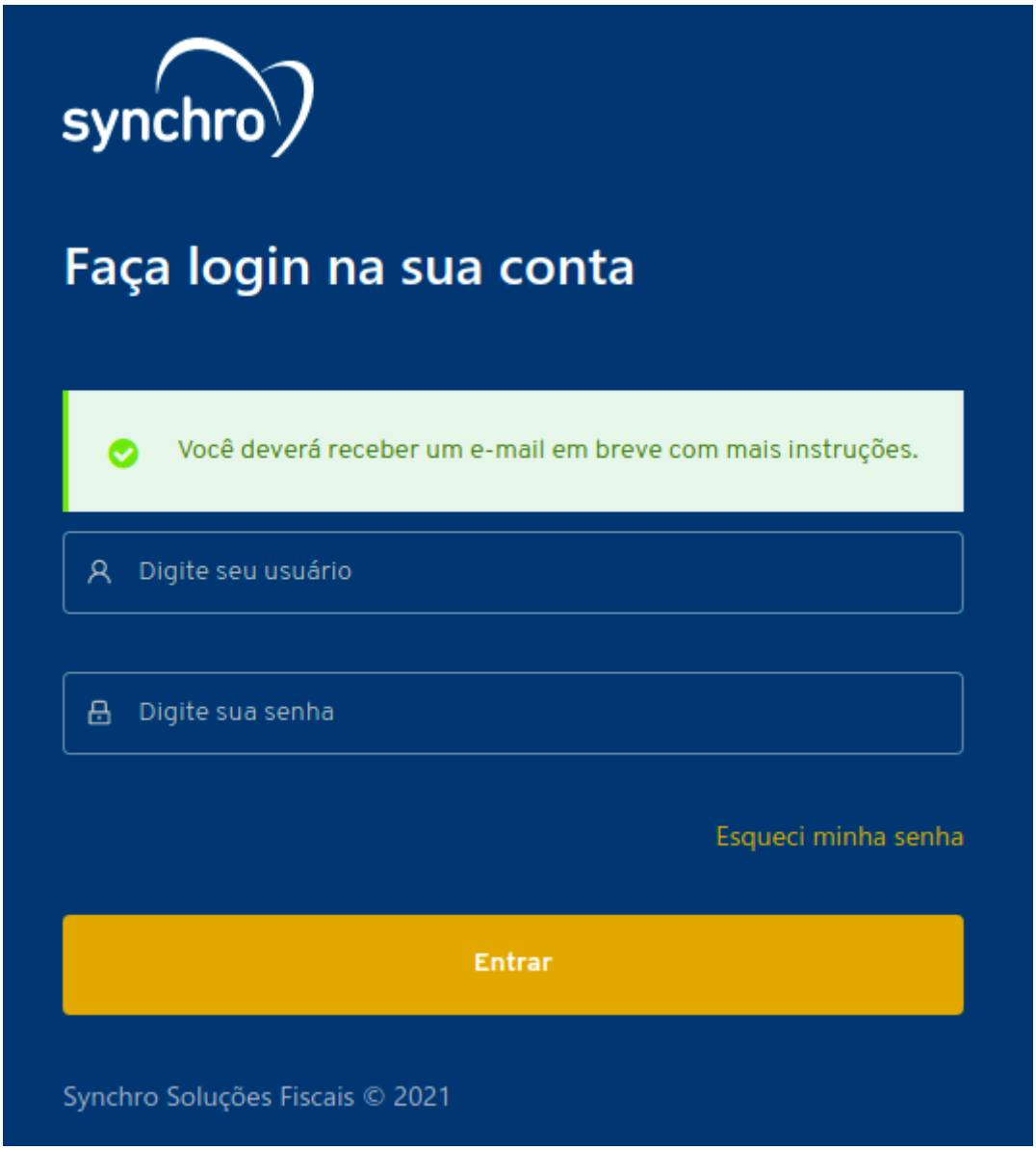
1. Click Forgot password.



2. Please enter your username or your email in order to recover your password.



3. You will receive an email with detailed instructions.


synchro
Faça login na sua conta
Você deverá receber um e-mail em breve com mais instruções.
Digite seu usuário
Digite sua senha
Esqueci minha senha
Entrar
Synchro Soluções Fiscais © 2021

Foundation Keycloak Standard

We are providing initial configuration, check below all informations:

Synchro realm with default values

Property	Value	Advanced
Realm	synchro	Realm
Clients	foundation-authentication	Clients
Clients roles	FOUNDATION_ADMIN FOUNDATION_CERTIFICATES foundation-authentication-foundation foundation-authentication-synchro4me	Clients roles

IMPORTANT

We recommend that you change the foundation-authentication client credentials, for higher security.

Change foundation-authentication credentials.

Foundation Keycloak Advanced

Setup

The steps below describe how to configure keycloak to foundation:

- Realm
  - Email
  - Themes
- Clients
  - Client to authentication
  - Client for tenant
  - Clients roles to access URI
    - Foundation certificates
    - Foundation admin
- Adding roles to the user role mapping
- LDAP configuration

Realm

Once you have an administrative account for the Admin Console, you can configure realms. A realm is a space where you manage objects, including users, applications, roles, and groups. A user belongs to and logs into a realm. One Keycloak deployment can define, store, and manage as many realms as there is space for in the database.

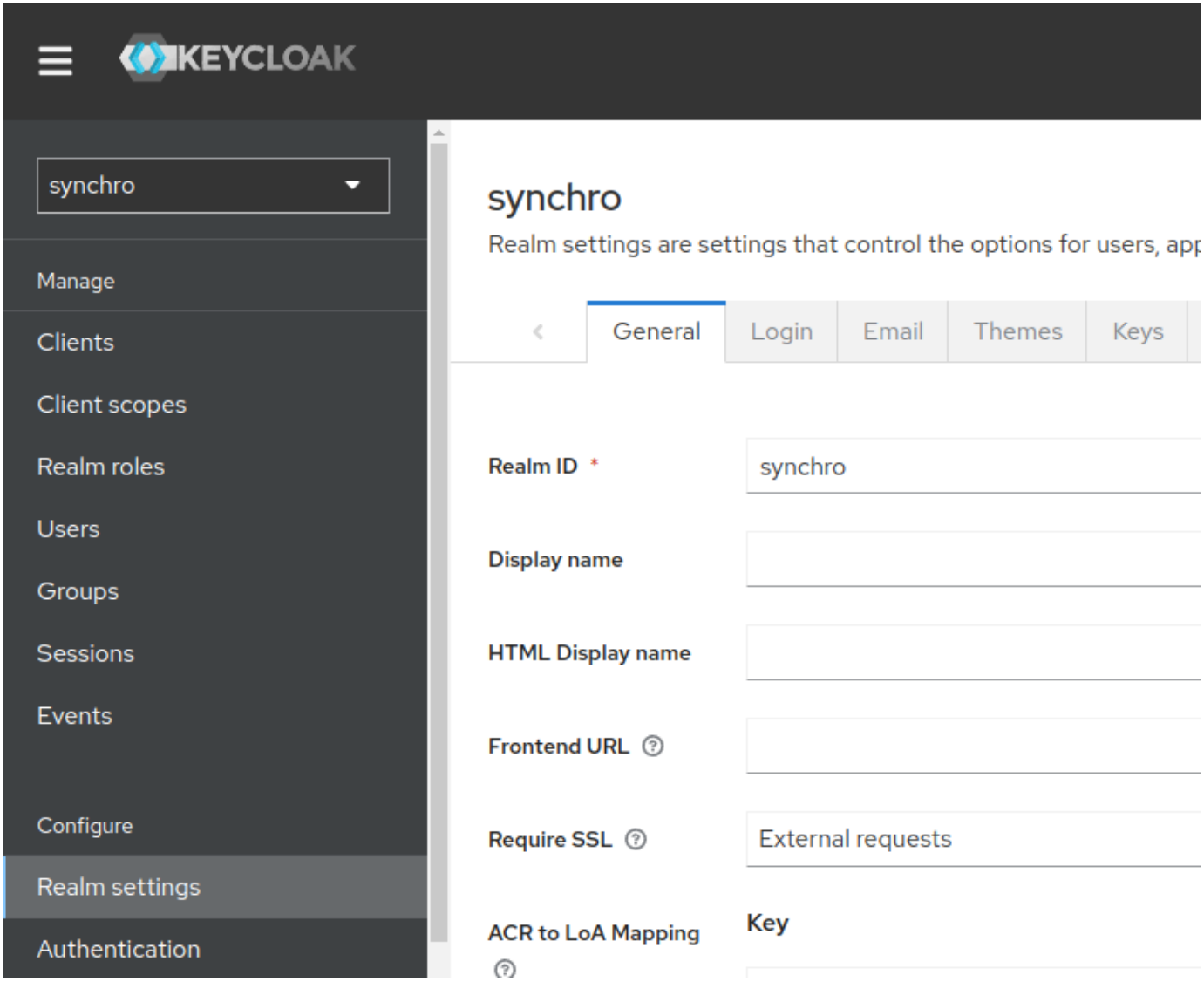
About Login

Check Login information.

Important

It's strongly recommended that you do **not use** the master realm to manage the users and applications in your organization. Keep the master realm as a place for super admins to create and manage the realms in your system. This keeps things clean and organized.

Realm feature, see keycloak realm.



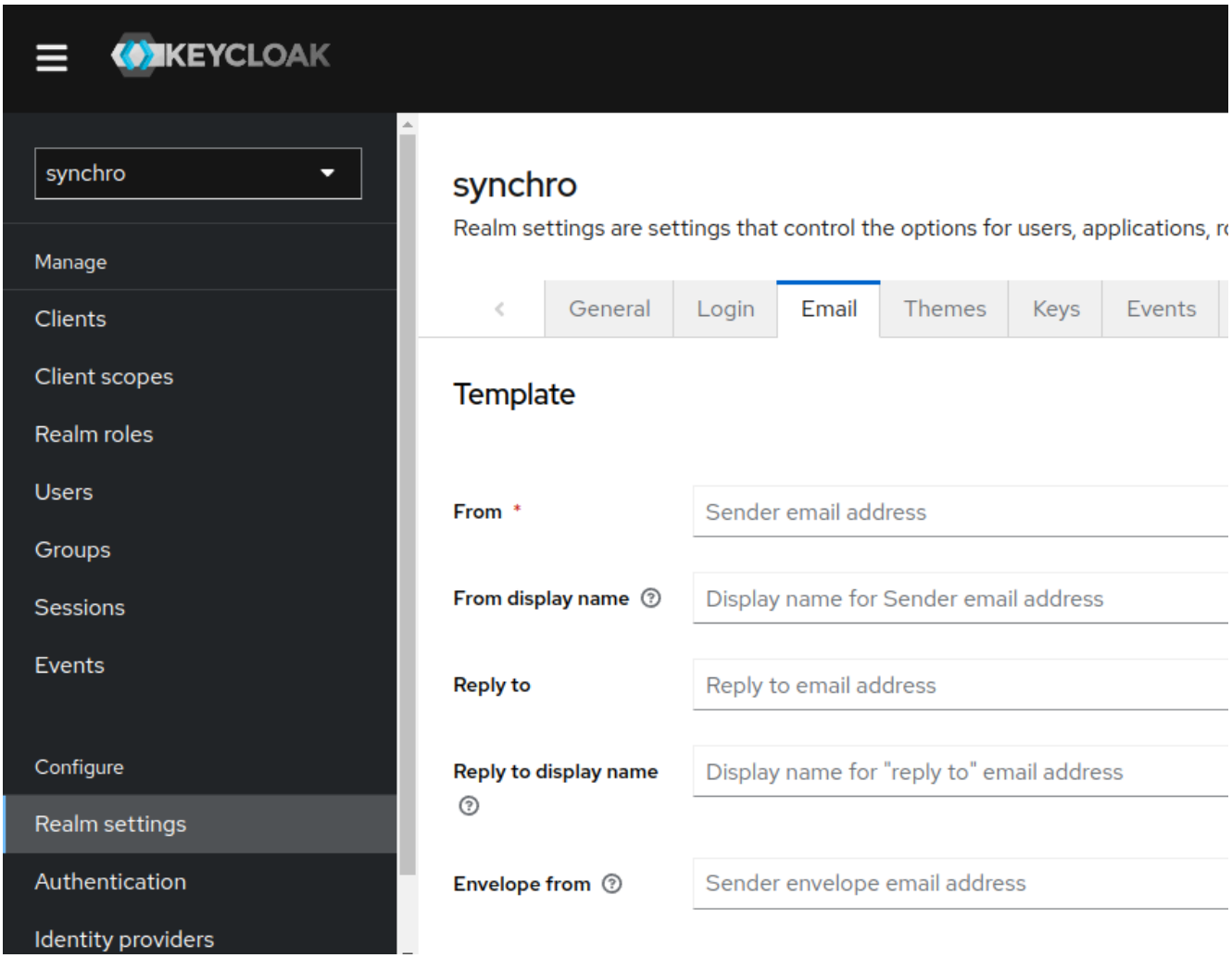
Email

Keycloak sends emails to users to verify their email addresses, when they forget their passwords, or when an administrator needs to receive notifications about a server event. To enable Keycloak to send emails, you provide Keycloak with your SMTP server settings.

For more information, see Keycloak Email.

Procedure

1. Click "Realm settings" in the menu.
2. Click the "Email" tab.



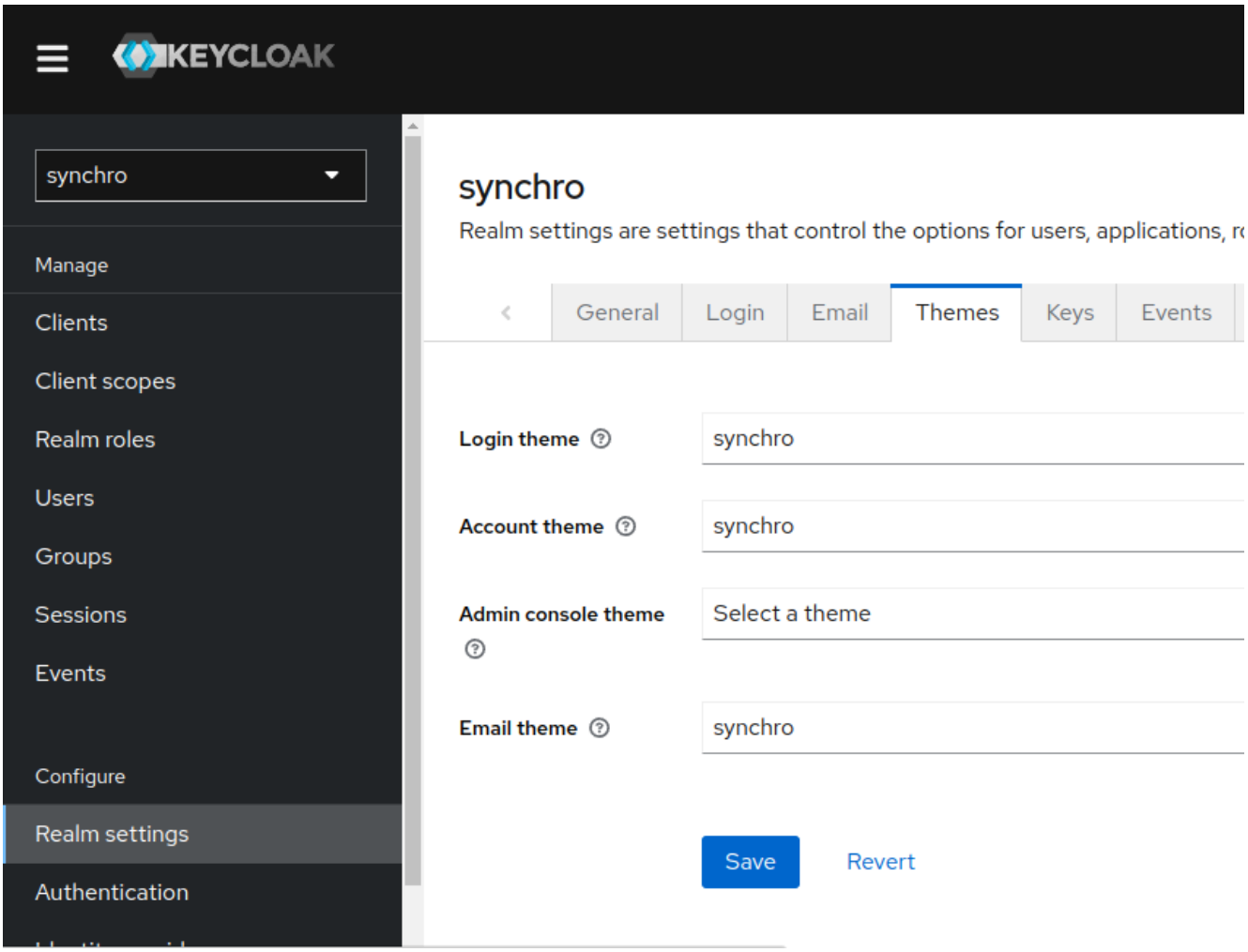
3. Fill in the fields and toggle the switches as needed.

Themes

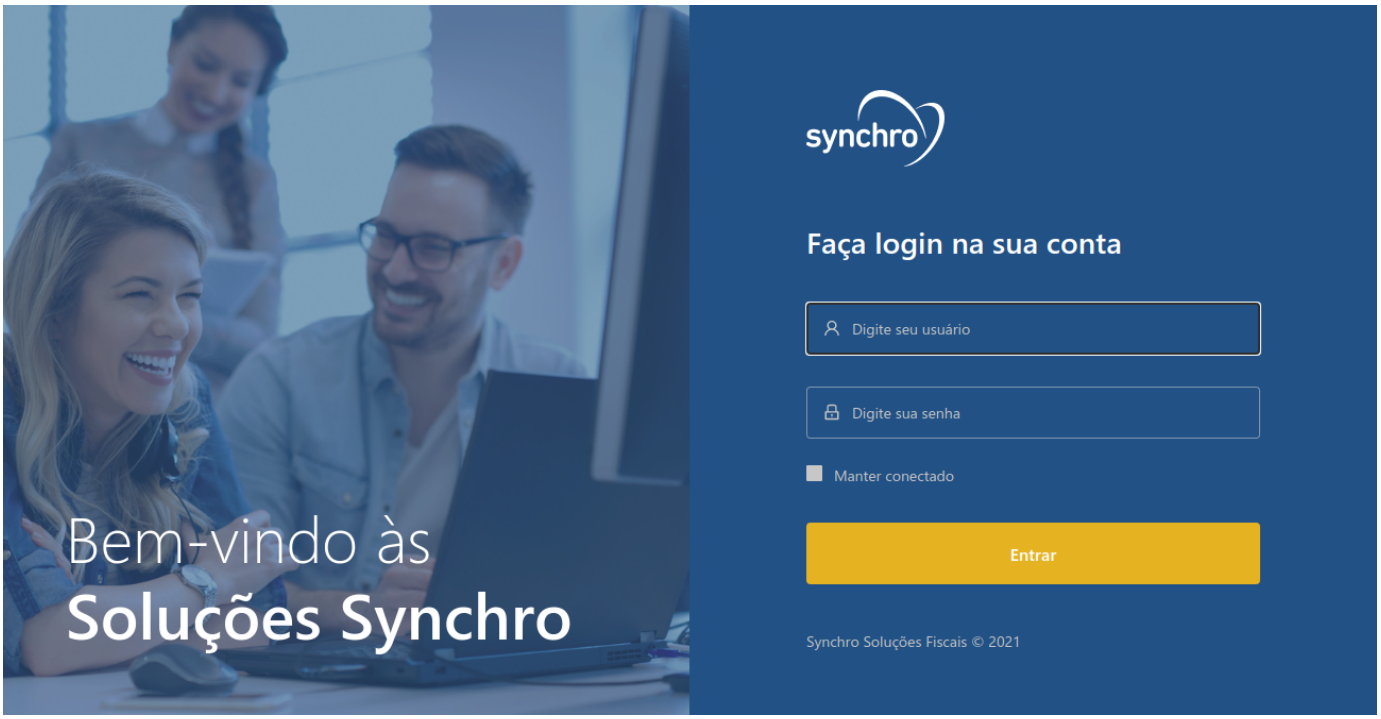
Keycloak provides theme support for web pages and emails.

Procedure

1. Select Realm.
2. Click "Realm Settings" in the menu.
3. Click the "Themes" tab.
4. Select `synchro` theme available themes box.



Login page with synchro theme



Clients

Clients are entities that can request Keycloak to authenticate a user or get roles information.

Procedure

1. Click "Clients" in the menu.
2. Click "Create".
3. Create a Client ID following the Pattern:

   Example for Client ID:

   Pattern: `<tenant>-<environment>`

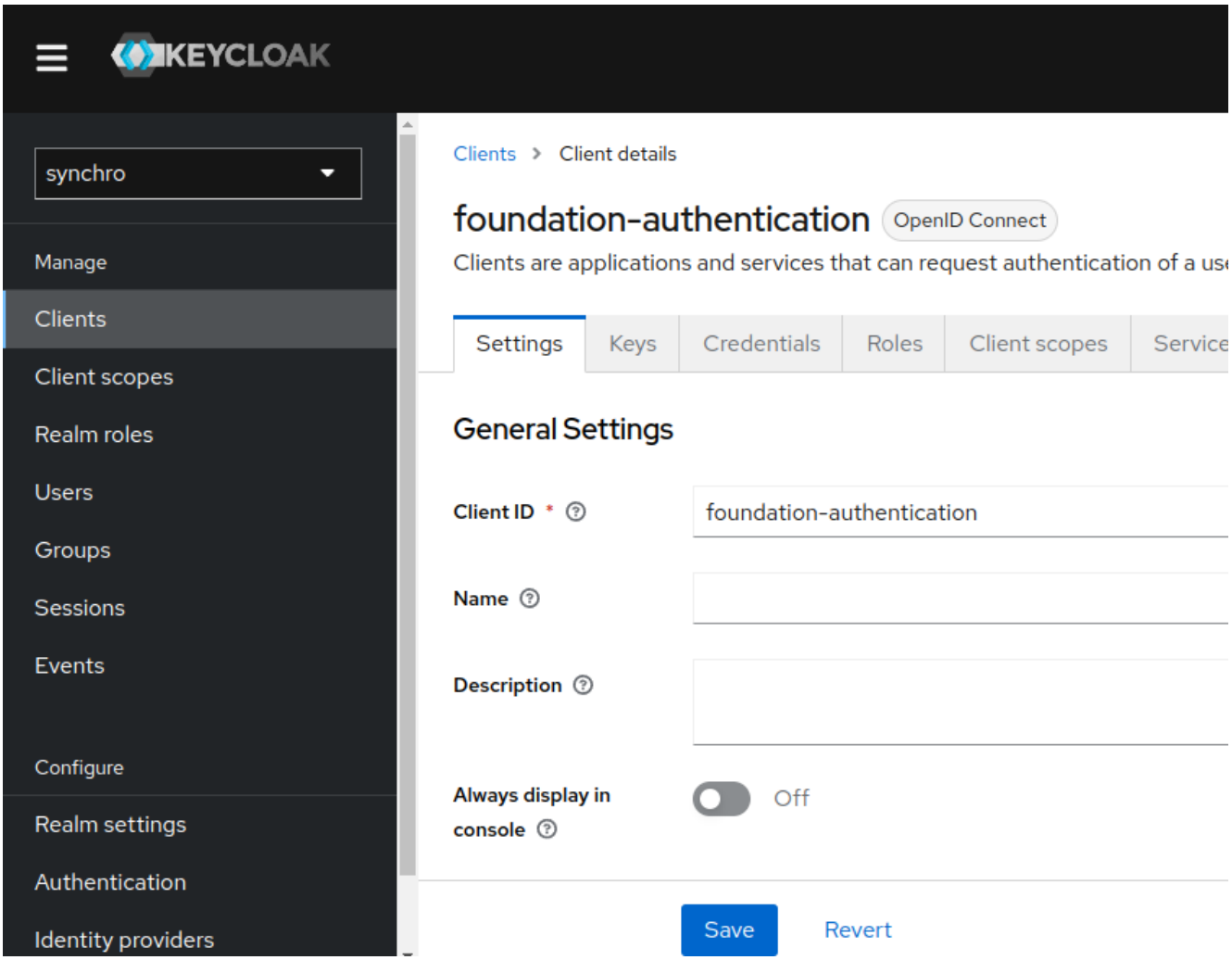
   TenantID: `SYNCHRODESENVOLVIMENTO`

   Environment: `desenvolvimento`

   Client ID: `synchro-desenvolvimento`

   Check available environment default list.

   Client feature, see keycloak client.
4. Save.



Client to authentication

Procedure

1. At Client Settings Tab in General Settings fill client id field with the client ID name, we suggest `foundation-authentication`
2. Insert a "Valid Redirect URIs", In `Access settings` group the `default` value for Valid redirect URLs is `http://*`, `https://*`:

   Required field. Enter a URL pattern and click + to add and - to remove existing URLs and click Save. You can use wildcards at the end of the URL pattern.

Security advise

Using the default values `http://*` and `https://*` makes your keycloak client accepts authentications redirects to all url protocols and adresses. It's a full wild card settings. To make your enviroment more secure we recommend edit this values to accept only recirects came from specifics `foundation` servers and protocols.

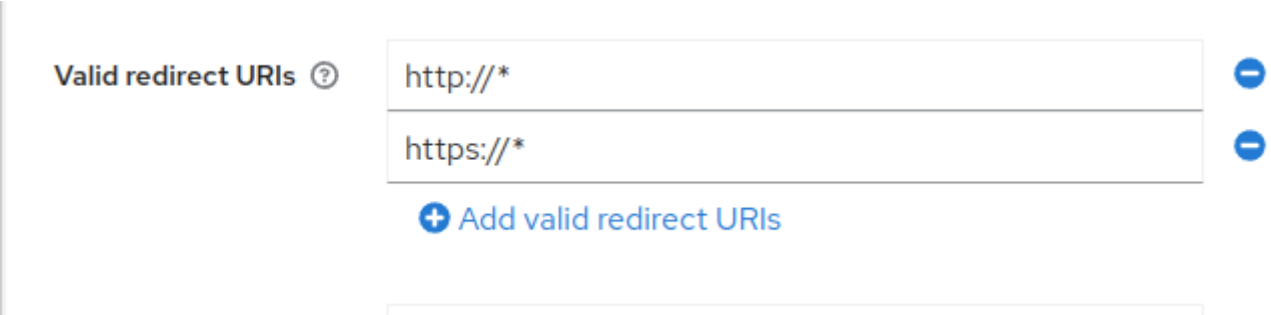
See example bellow:

Generic/Default: `http://*` and/or `https://*`

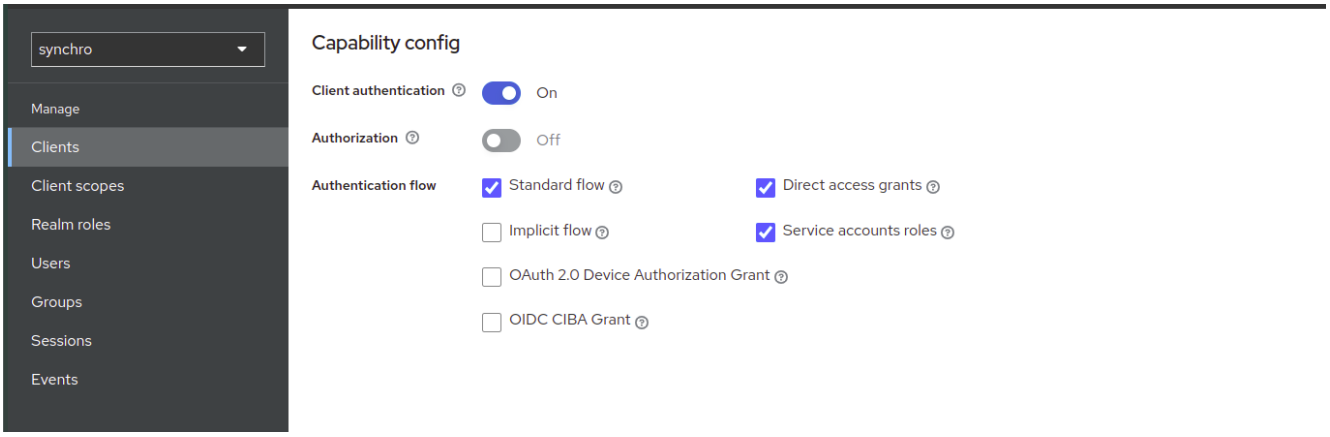
or

Specific: `http://172.25.0.0/*` and/or `http://synchro-dev/*`

Basic settings, see Keycloak Basic configuration.

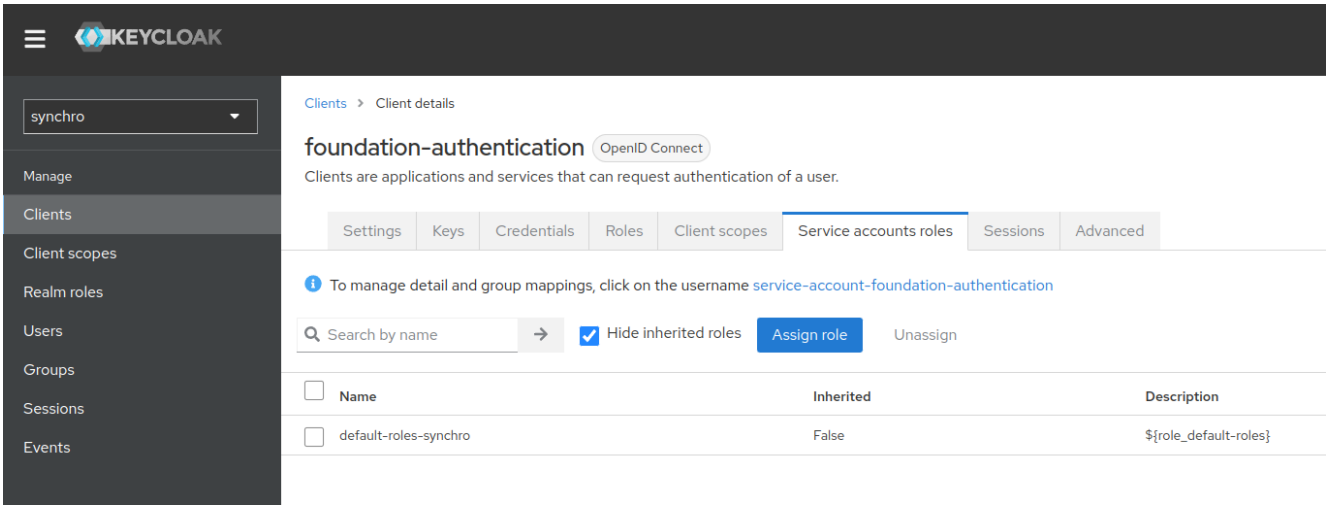


3. Enable "Client authentication", "Service Accounts roles" and "Direct access grants" properties at `Capibility Config` group.

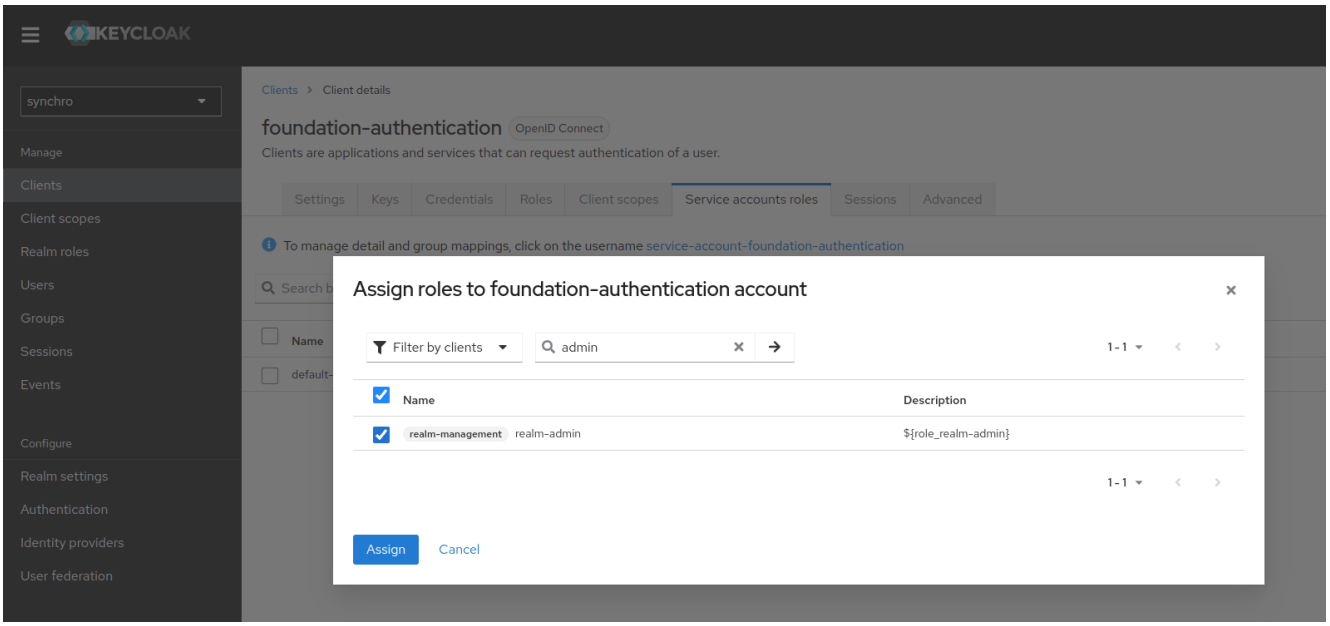


4. Assign realm-admin role to "Service Accounts roles"

4.1. Click in "Assign Role"

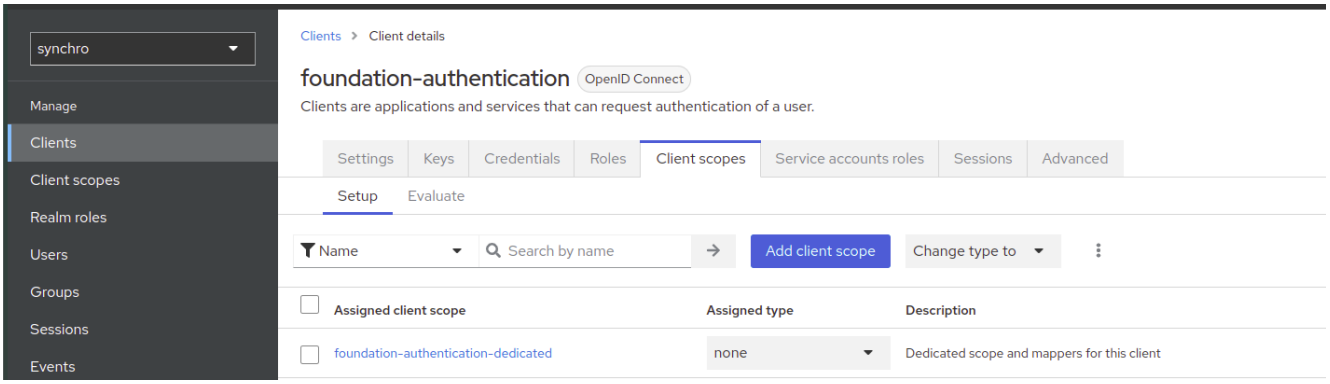


4.2. Select "realm-admin" roles and click in "Assign"

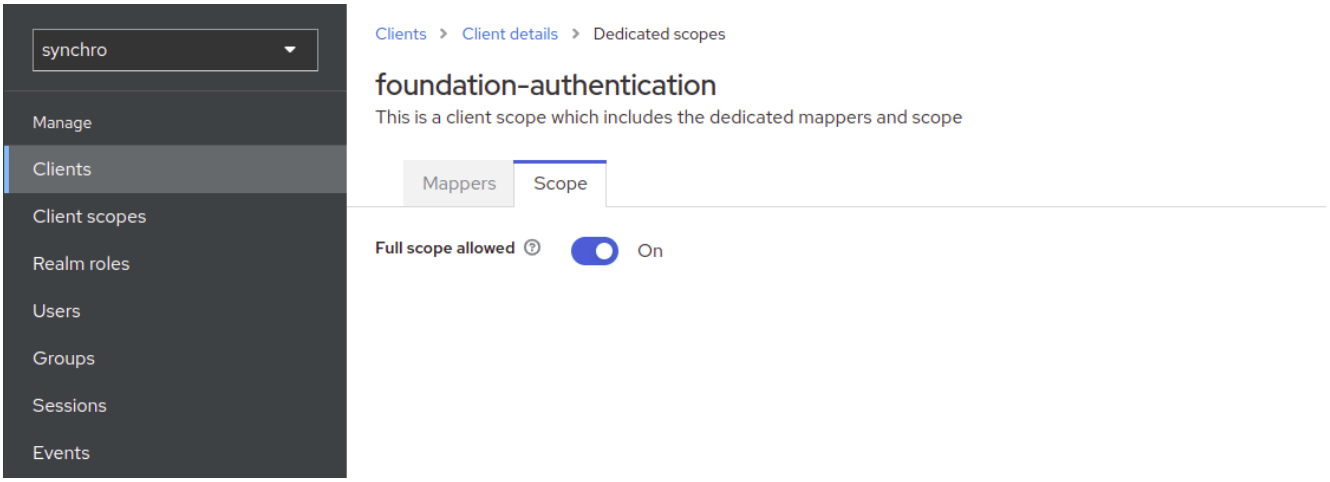


5. Active foundation-authentication-dedicated full scope.

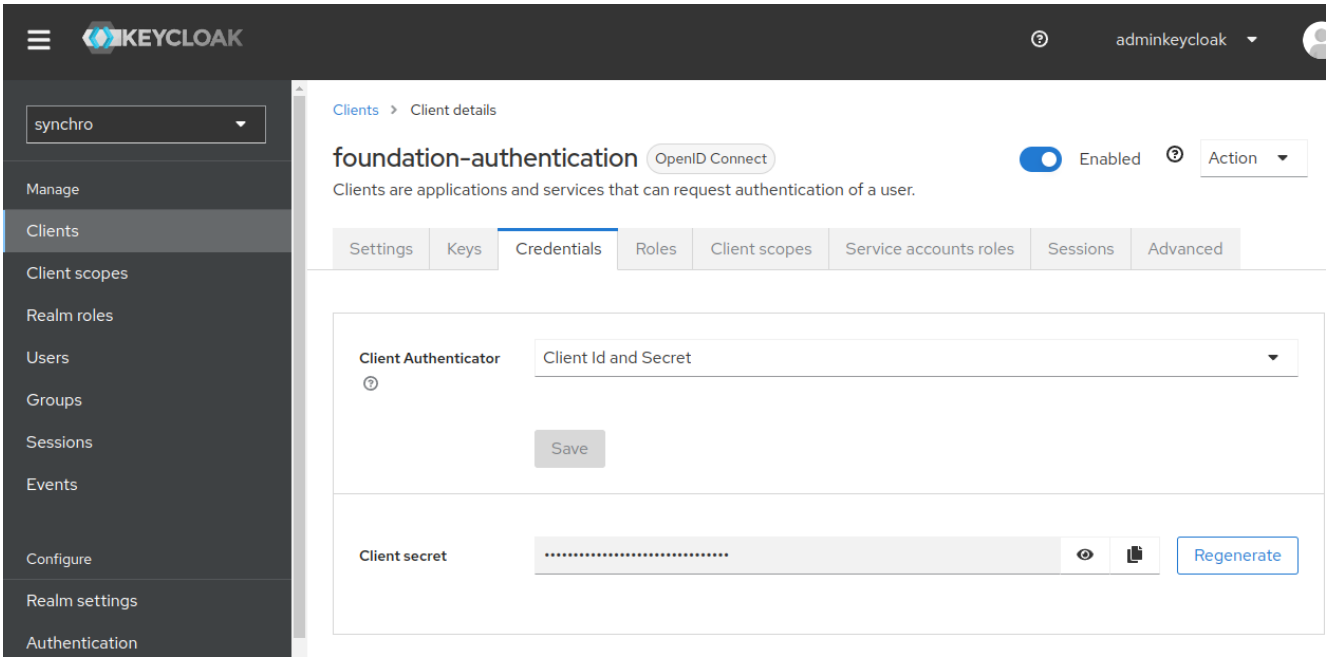
5.1. Click in "foundation-authentication-dedicated"



5.2. Go to Scope tab, then active Full scope allowed toggle



6. Save.
7. Click the "Credentials" tab, now there is a secret.



8. To configure client authentication in foundation:

   8.1. Change keycloak information

Client for tenant

If your environment is Multitenancy, create a client for each Tenant.

Procedure

1. See creating a client.
2. See creating a client roles applications

Clients roles application

Most often, clients are applications and services that want to use Keycloak to secure themselves and provide a single sign-on solution. Clients can also be entities that just want to request identity information or an access token so that they can securely invoke other services on the network that are secured by Keycloak.

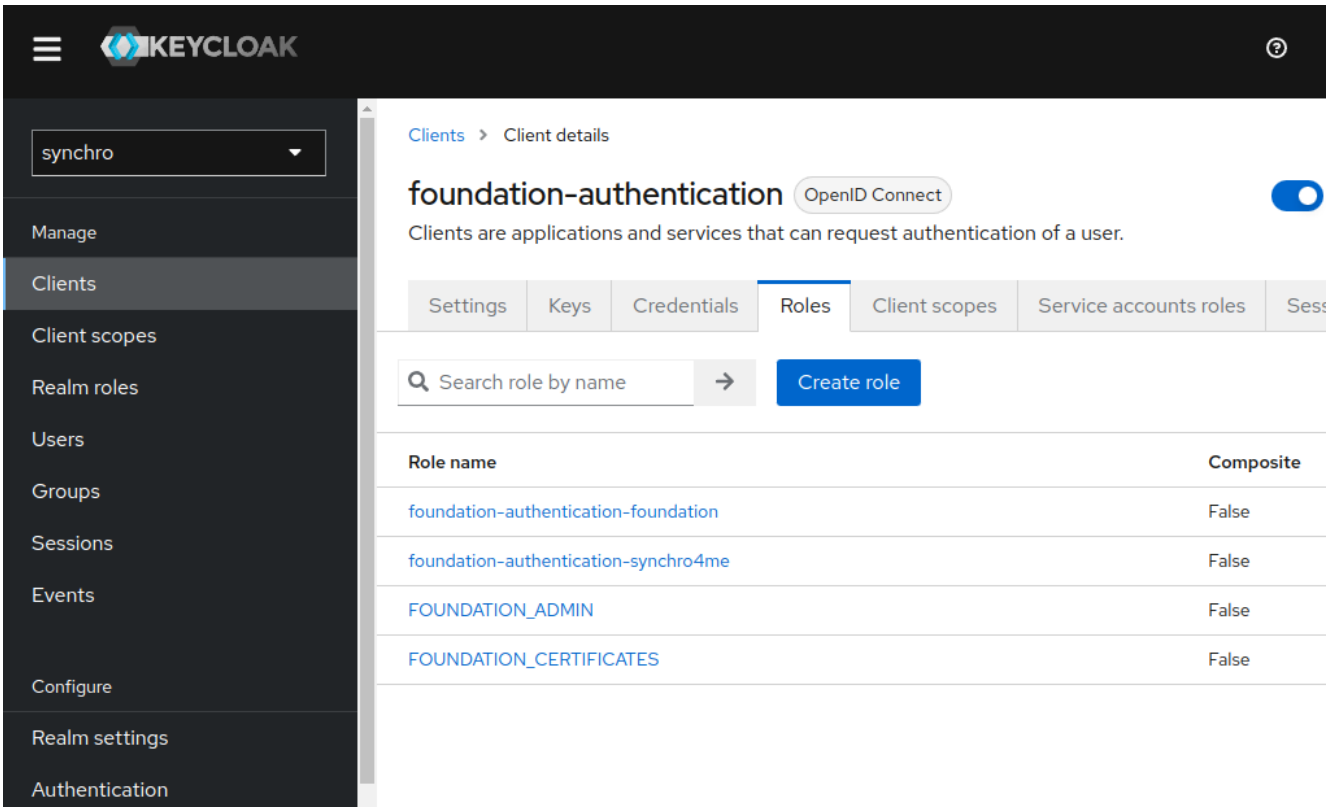
All Clients needs this role to access application URI

Procedure

1. Click "Clients" in the menu.
2. Select your client.
3. Click the "Roles" tab, and "Create role" button.
4. Add a role following the pattern.

> **See example bellow:**
>
> Pattern: `<clientID>-<application>`
>
> Client ID: `synchro-desenvolvimento`
>
> application: `foundation`
>
> Result: `synchro-desenvolvimento-foundation`

Role mapping feature, see Restrict user role mapping.



5. If this client needs open foundation administration console, see foundation admin.

Foundation certificates

Foundation need a clients role `FOUNDATION_CERTIFICATES` to update Keystore administration console.

> **Important**
>
> If you need to update the Keystore, add this role in your specific Client Authenticaton or Client Tenant.

Foundation admin

Foundation need a clients role `FOUNDATION_ADMIN` to open administration console.

> **Important**
>
> If you need access administration console, add this role in your specific Client Authenticaton or Client Tenant.

Adding roles to the user

You can assign role mappings to a user through the Role Mappings tab for that user.

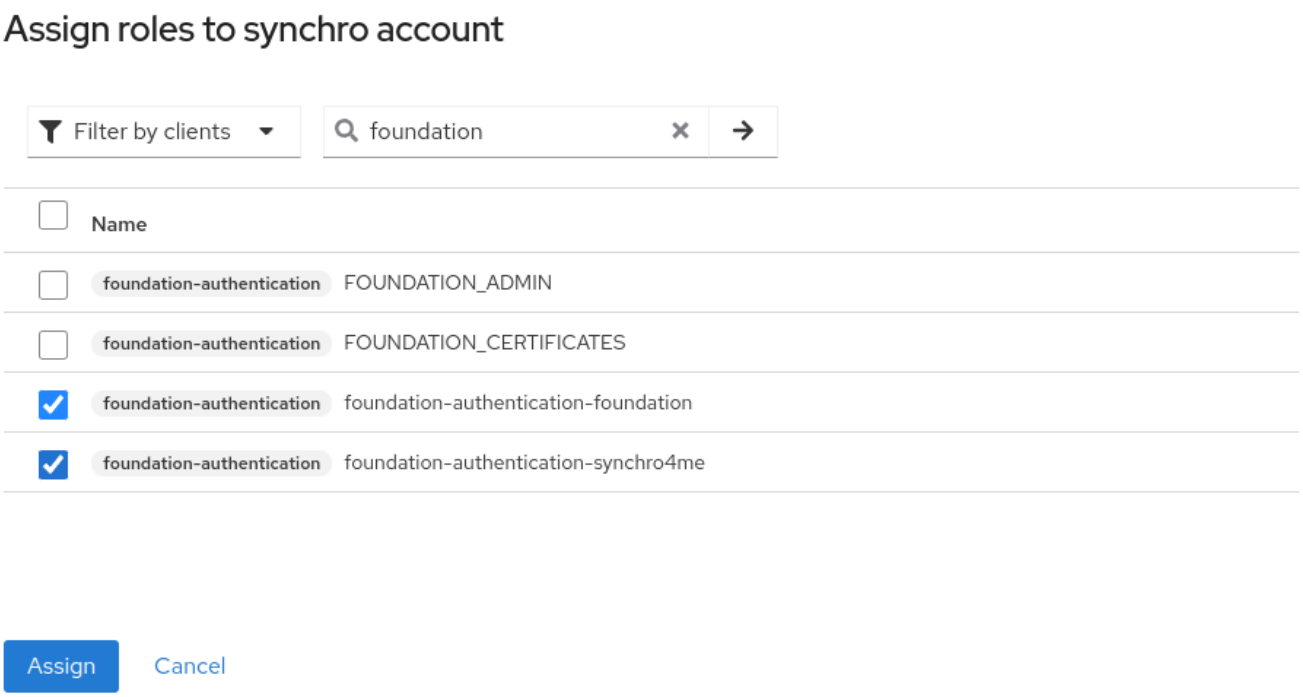
Foundation roles explanation

Foundation administration console: FOUNDATION_ADMIN.

Update Keystore: FOUNDATION_CERTIFICATES.

Procedure

1. Click "Users" in the menu.
2. Click the user that you want to assigning a role. If the user is not displayed, click View all users or search the user by mail at the search field.
3. Click the "Role Mapping" tab.
4. Click the "Assign role" button.
5. Select "Filter by clients" and search by role name.



6. Selected roles that you want and click "Assign" button.
7. Do user Logout/Login in application to get new roles.

See more, in Keycloak assigning role mappings.

Regenerate Client secret

Procedure

1. Click "Clients" in the menu.
2. Select Client.
3. Click the "Credentials" tab.
4. Click the "Regenerate" button.
5. To configure client authentication in foundation:
   5.1. Change keycloak information.
6. Access View.

Change keycloak information

```
sudo foundation config --on-premises-keycloak
```

```
INFO[0000] Reading profiles from /etc/foundation/
default

QUESTION: Select your profile file (current: default):
```

```
QUESTION: This command changes your keycloak settings to local. Use only if you are an on premise
installation. CONTINUE? (y/N): y
```

Domain name

```
QUESTION: Using domain name in a multitenant solution []:
```

If you have different tenants: In many multitenant, a domain name is used to identify a tenant.

> **Example for Domain name:**
>
> URL: `synchro.com.br`

Synchro4me DNS Requirements

Some Synchro4me applications requires an DNS configured to your server to identify your tenant. Like `foundation-prd.synchro.com.br` in this case the tenant is `FOUNDATIONPRD`. Please consult the Synchro4Me manual to see DNS requirements.

URL server

```
QUESTION: (Keycloak) URL [http://192.168.0.160/keycloak]:
```

Realm

```
QUESTION: (Keycloak) Realm [synchro]:
```

Get `Realm name` from Realm, see Keycloak Realm

Client ID

```
QUESTION: (Keycloak) Client ID [foundation-authentication]:
```

Get `Client ID` from Client authentication, see Keycloak Clients authentication

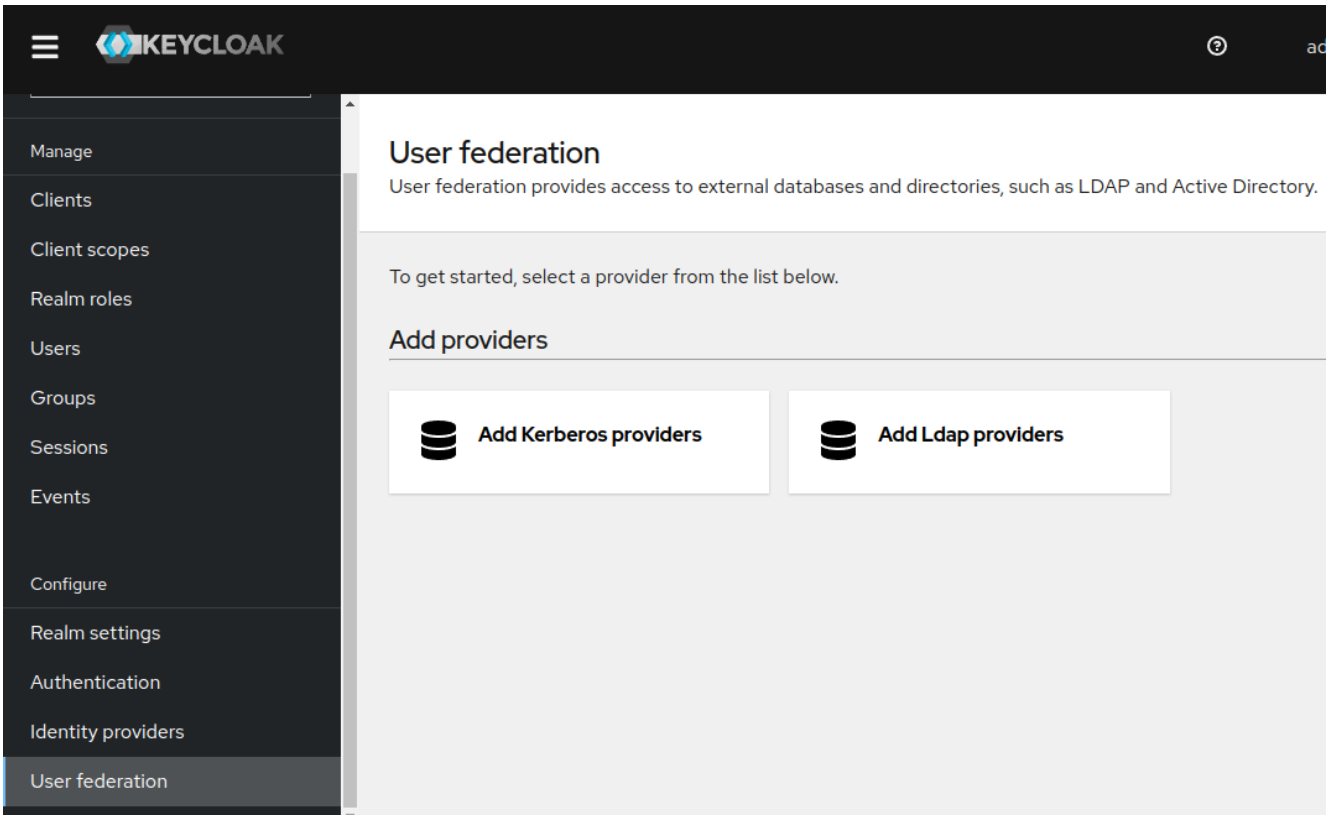
Client Secret

```
QUESTION: (Keycloak) Client Secret [secret]: B24KXaFbwPkwokBngVyjSp
```

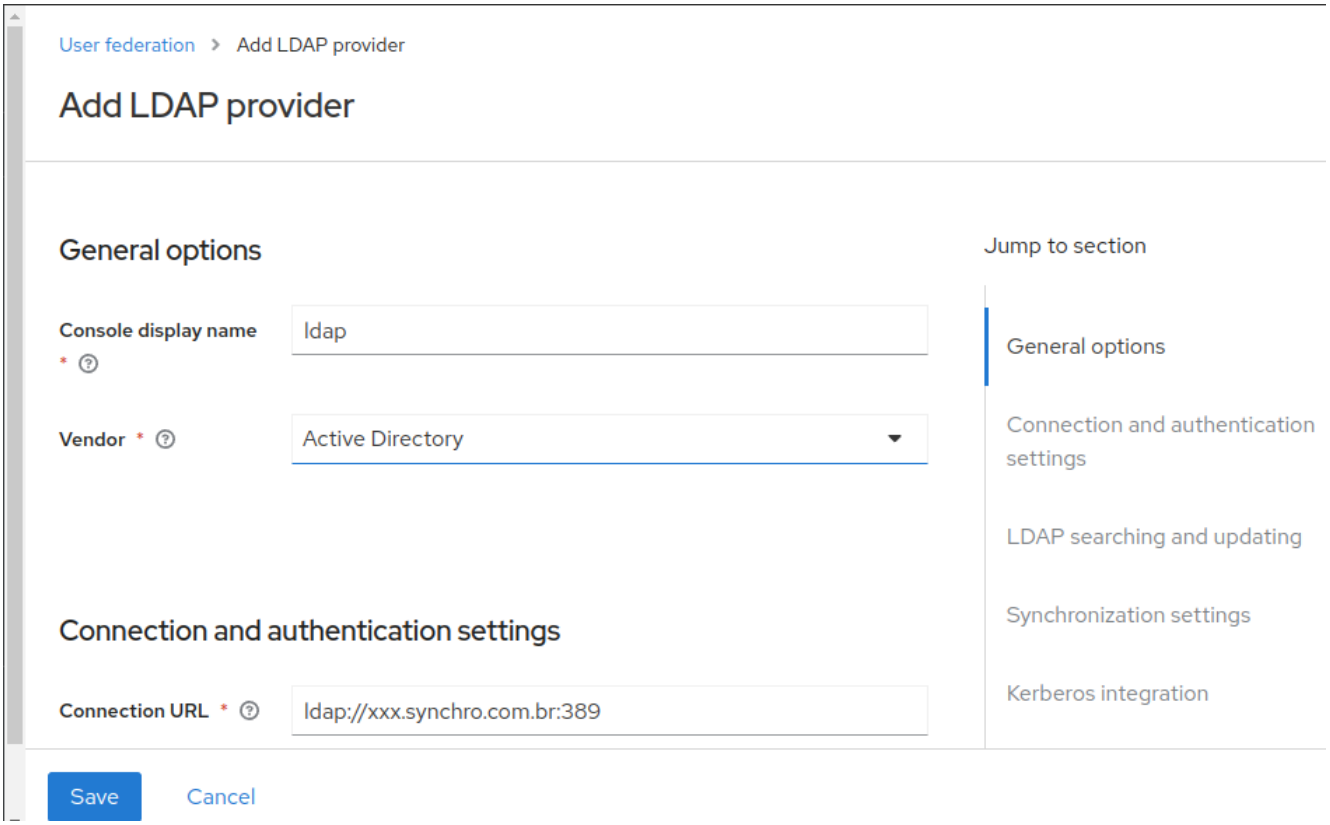
Get `Secret credentials` from client authentication, see Keycloak Clients authentication

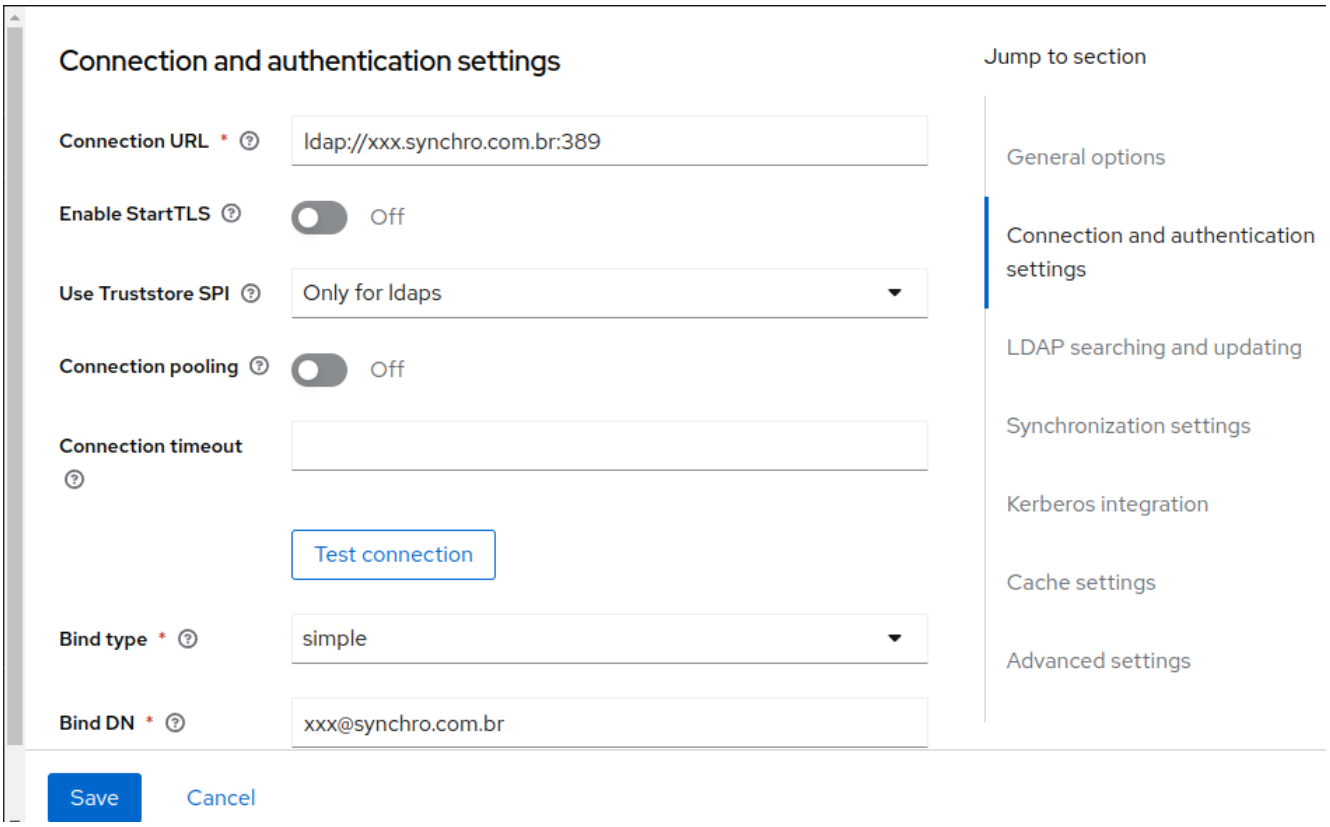
LDAP configuration

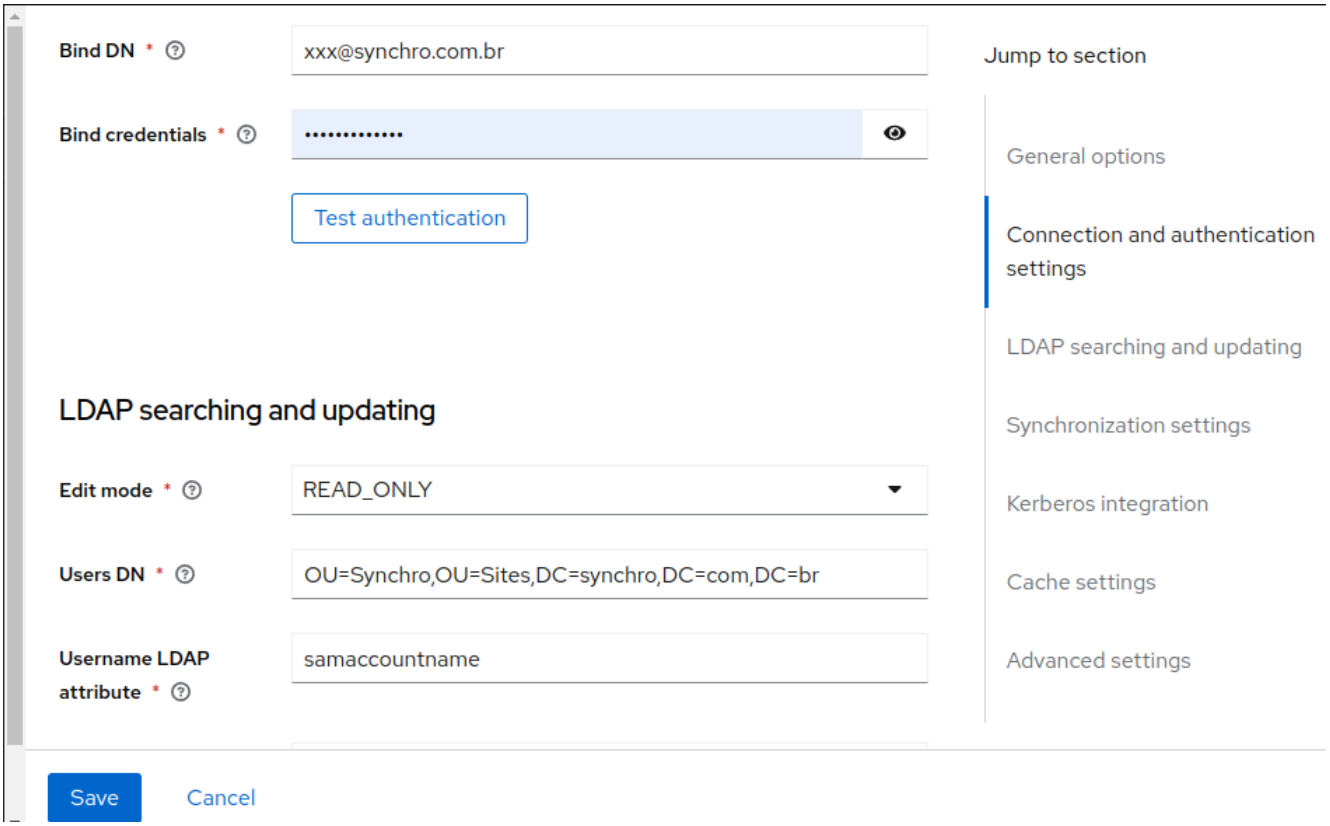
1. Click "User Federation" in the menu and "Add Ldap providers".

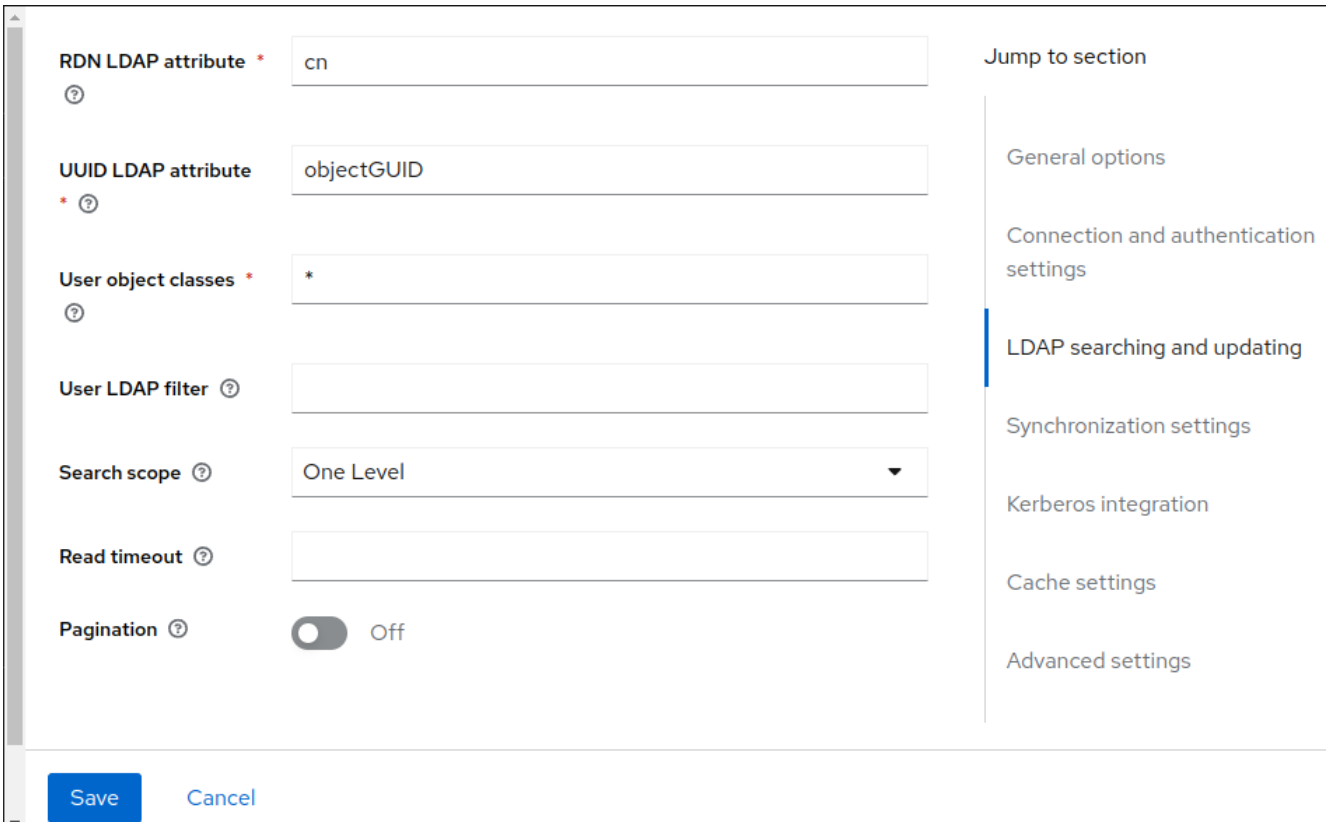


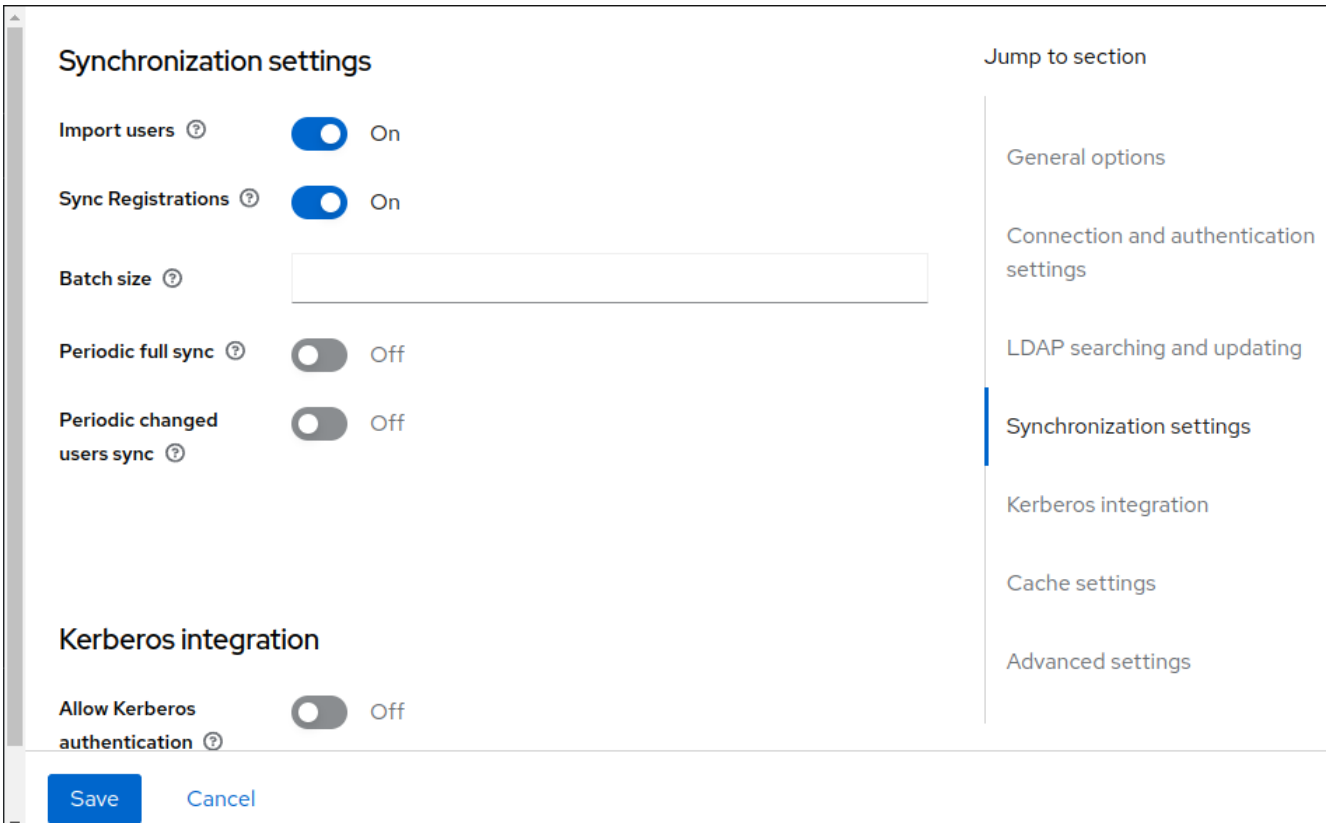
2. Fill all fields like the example below




Connection and authentication settings
Connection URL *
ldap://xxx.synchro.com.br:389
Enable StartTLS
Off
Use Truststore SPI
Only for ldaps
Connection pooling
Off
Connection timeout
Test connection
Bind type *
simple
Bind DN *
xxx@synchro.com.br
Save
Cancel
Jump to section
General options
Connection and authentication settings
LDAP searching and updating
Synchronization settings
Kerberos integration
Cache settings
Advanced settings

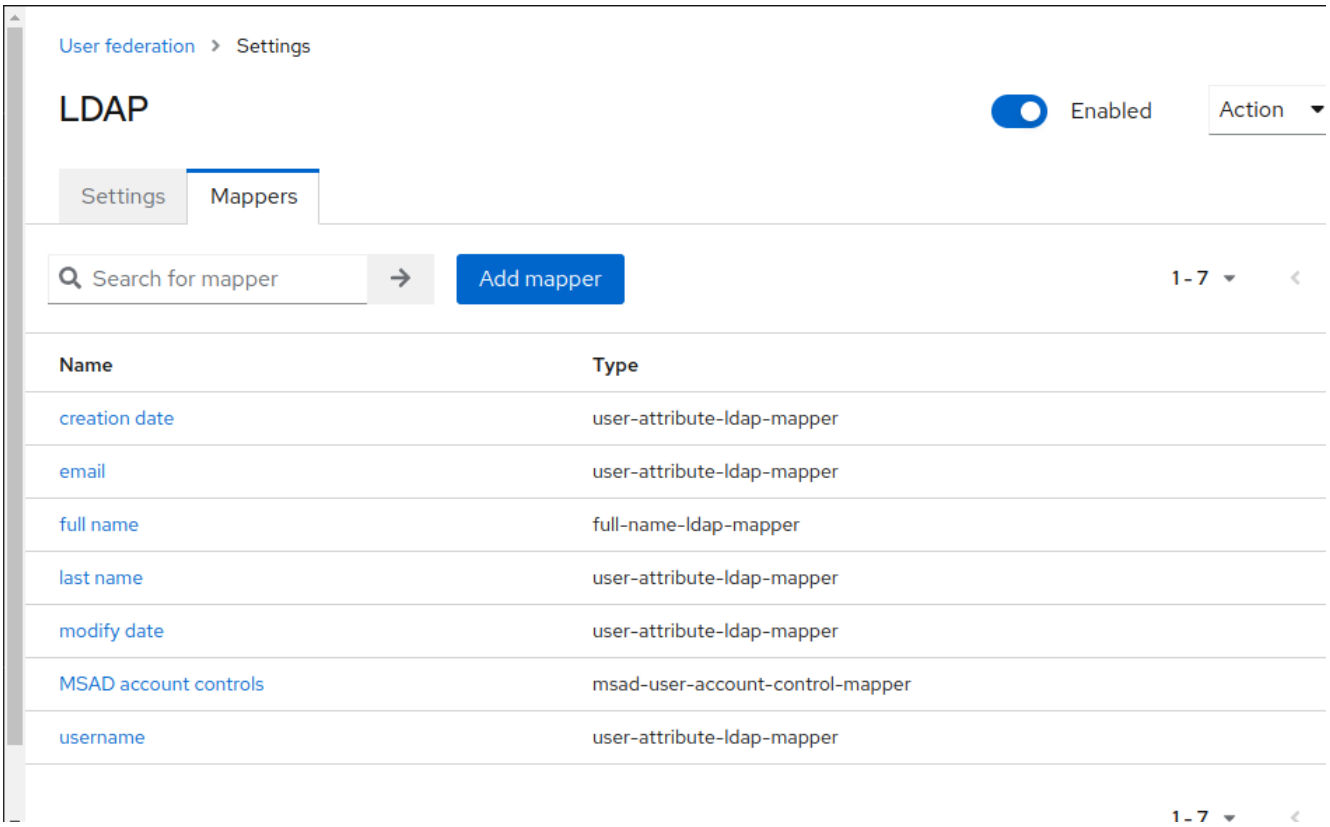

Bind DN *
xxx@synchro.com.br
Bind credentials *
Test authentication
LDAP searching and updating
Edit mode *
READ_ONLY
Users DN *
OU=Synchro,OU=Sites,DC=synchro,DC=com,DC=br
Username LDAP attribute *
samaccountname
Save
Cancel
Jump to section
General options
Connection and authentication settings
LDAP searching and updating
Synchronization settings
Kerberos integration
Cache settings
Advanced settings



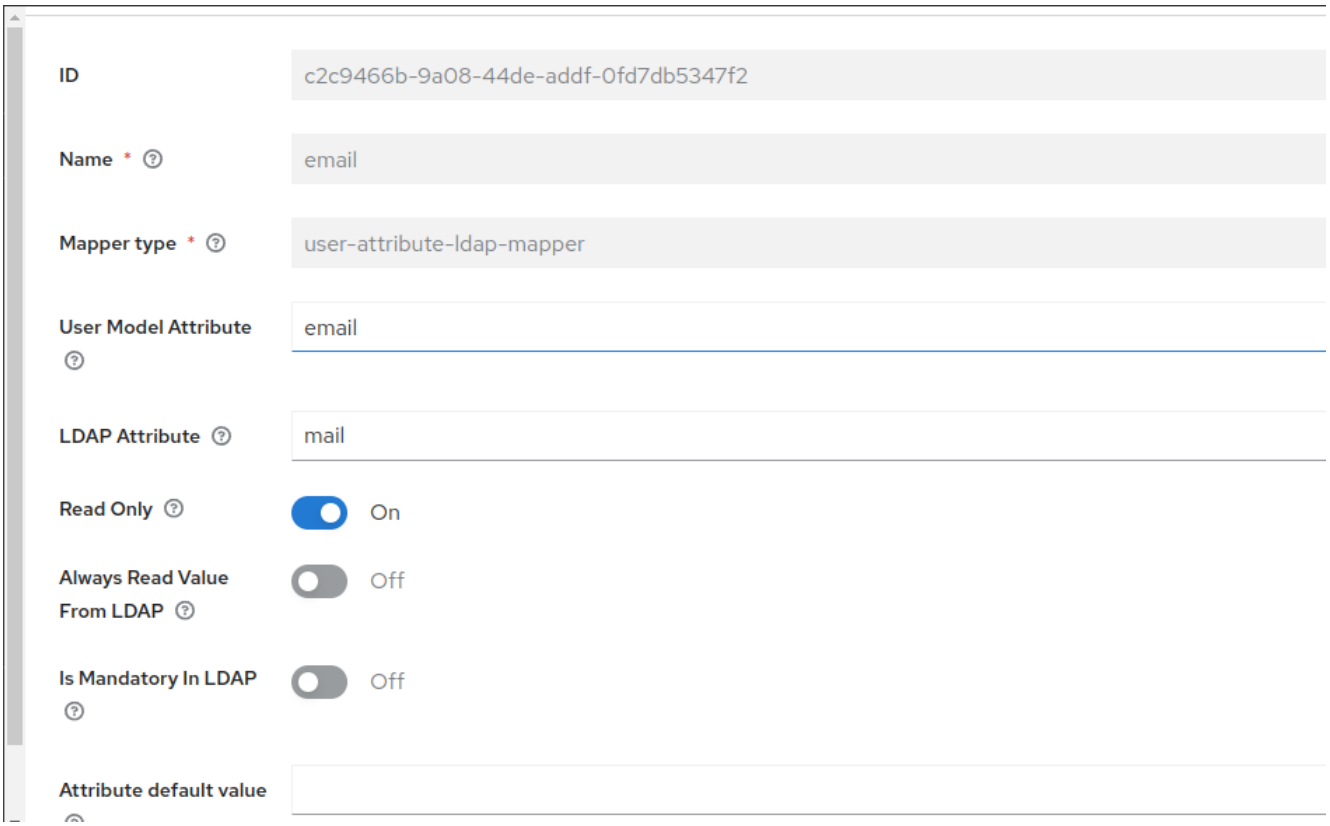


3. Save and click "Mappers" tab, to create ldap fields relations, create all relations what you need:

   Mapper list:



Mail relation example:

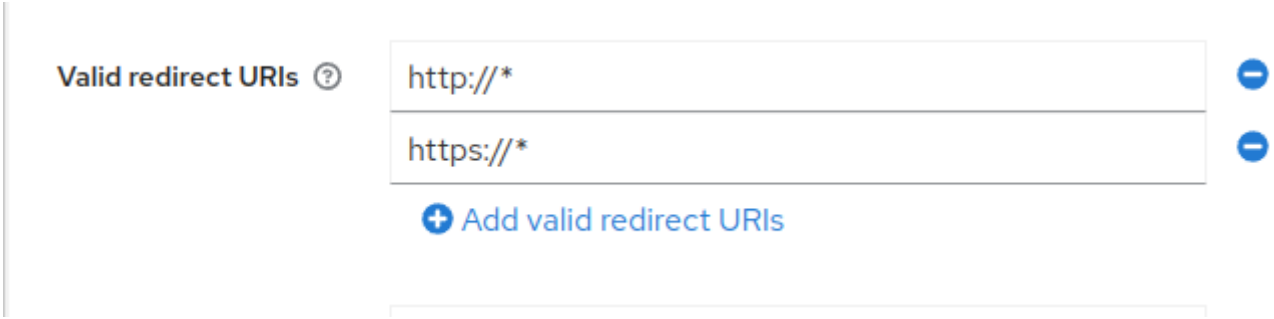


See more details, in Official Keycloak LDAP configuration site.

Valid Redirect URIs

The fields for "Valid Redirect URIs", In `Access settings` at your Keycloak Client to authentication configuration need you attention for more security.

The `default` values for "Valid redirect URIs" is `http://*`, `https://*`:



Using the default values `http://*` and `https://*` makes your keycloak client accepts authentications redirects to all uri,protocols and adresses. It's a full wild card settings.

To make your enviroment more secure we recommend edit this values to accept only recirects came from specifics `foundation` servers and protocols.

Enter a URL pattern and click + to add and - to remove existing URIs and click Save. You can use wildcards at the end of the URI pattern.

See example bellow:

Generic/Default: `http://*` and/or `https://*`
or
Specific: `http://172.25.0.0/*` and/or `http://synchro-dev/*`

For basic settings, see Keycloak Basic configuration.

Keycloak Videos

Sobre o keycloak

Synchro realm

Foundation Client authentication

Client tenant

User and roles mappings

Ldap

foundation-certificates

Overview

Foundation certificates is a certificate module, after install you will be able to manage KeyStore and TrustStore using any HTTP browser.

KeyStore

A keystore stores private key entries, certificates with public keys or just secret keys that we may use for various cryptographic purposes. It stores each by an alias for ease of lookup.

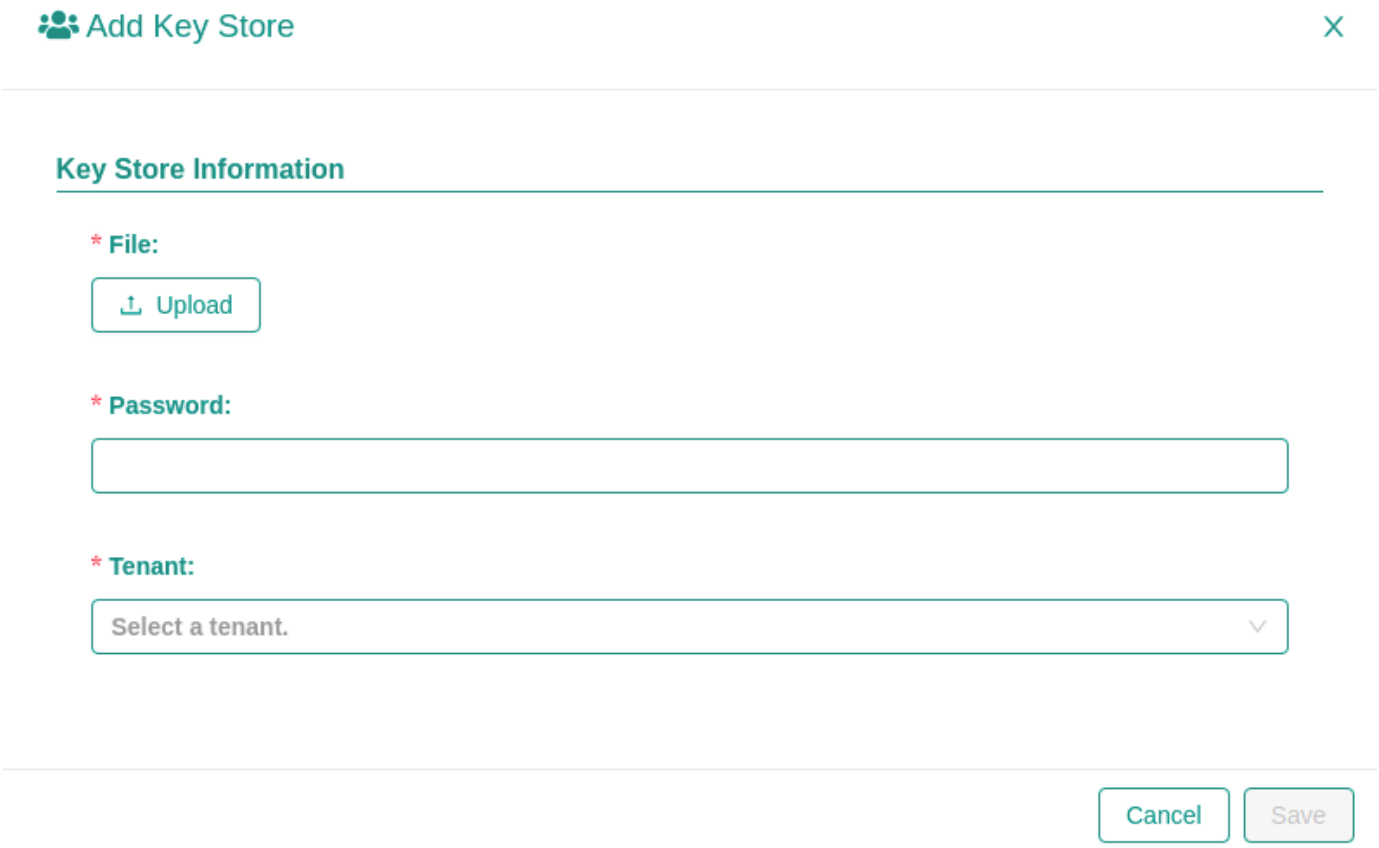
Essentially, a keystore used as a truststore will contain a number of (CA) certificates.

KeyStore configuration

How do I configure my keystore?

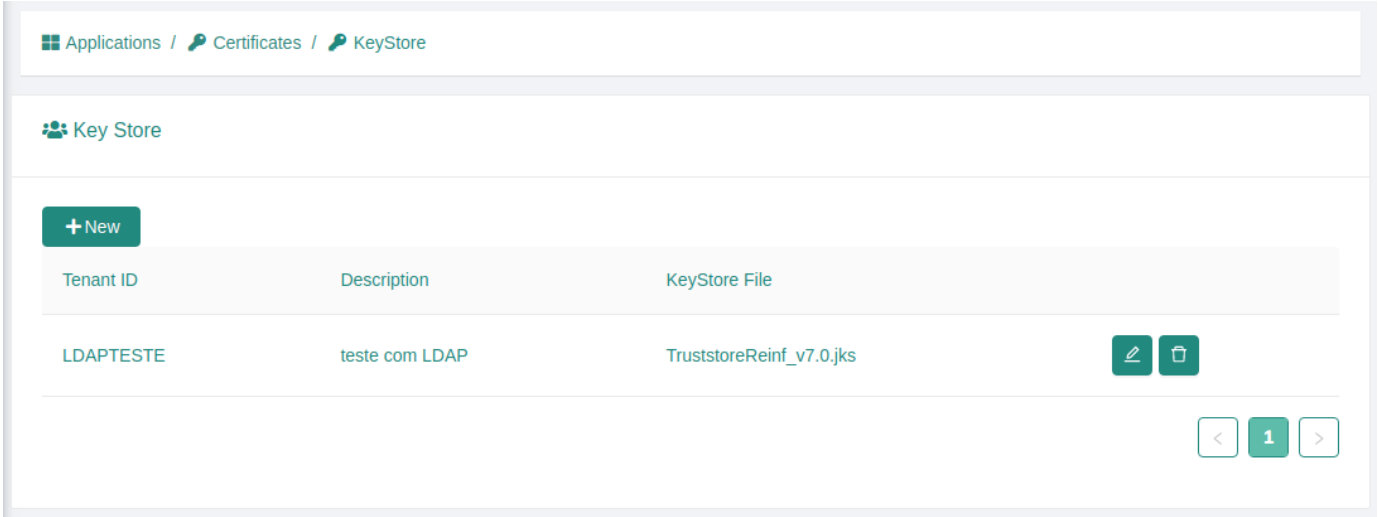
Important: login required

Add KeyStore



1. Upload file
2. Input keystore password
3. Select tenant to access keystore

Delete KeyStore



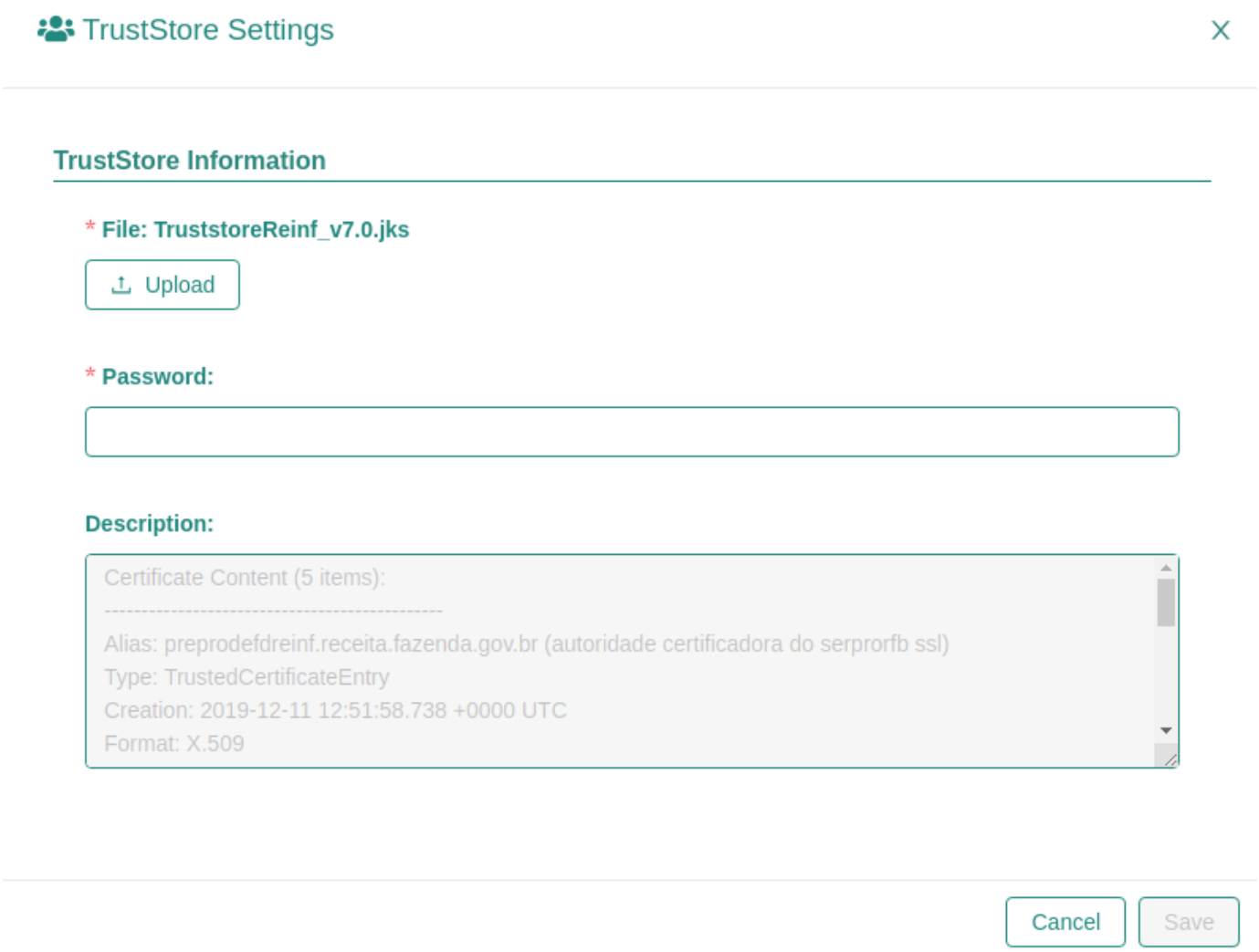
TrustStore

A truststore is the opposite – while a keystore typically holds onto certificates that identify us, a truststore holds onto certificates that identify others.

TrustStore file

Truststore is available in Synchro products website

Add or update TrustStore



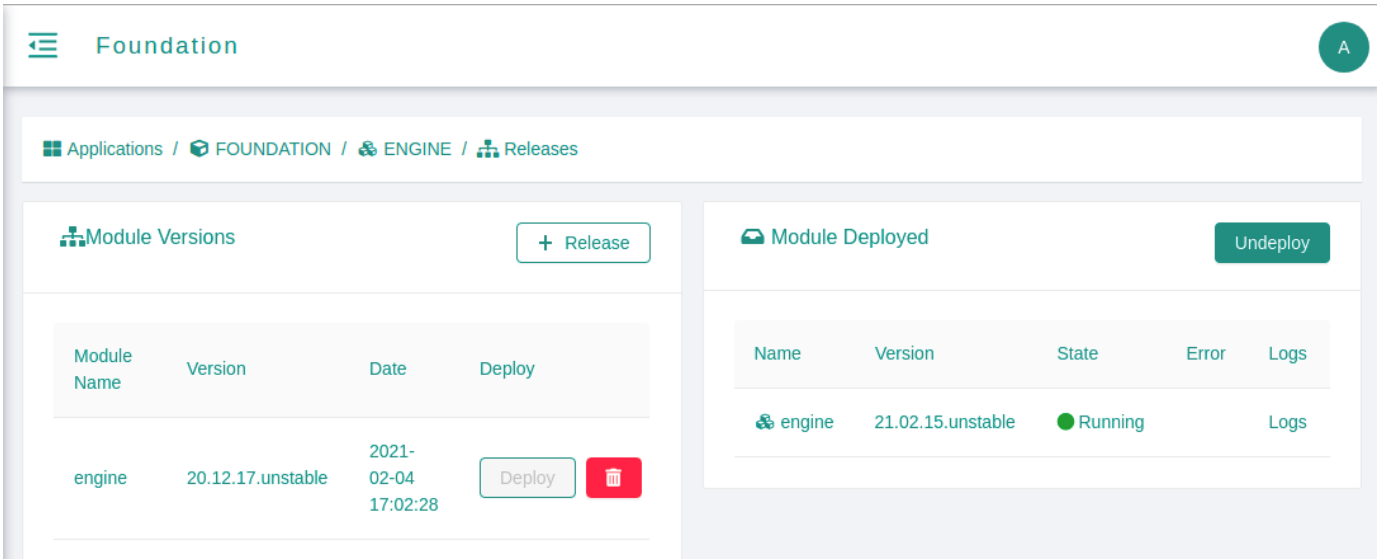
1. Upload file
2. Input truststore password

foundation-logs

The Foundation `logs` is a optional module that you can use to see the logs.

> **Tip**
>
> About `logs` module is recommended but not mandatory.

Once up and running, it is possible to access logs in module deployed card.



It's possible extract the module via `http://host:port/logs/` , anyway, if you don't have this module installed, you can access all the logs for yourself using the command-line and `kubectl logs -f service/<SERVICENAME>` command.

You can check the available registered services with `kubectl get services` .

foundation-licenses

- **UNDER CONSTRUCTION**
- **The contents of this page are subject to change**

Overview

Licenses is a Foundation's module responsible for providing the information for Synchro customers to license the contracted products into Foundation.

This module will not work on its own, internet connection is necessary, not for all, the access must be set for a specific address `(This specific address will be informed in application requirements)`.

Setup

Starting the licenses module, it's necessary to inform `Synchro licensing URL`

Synchro licensing URL

Please fill in the information requested with: `host:port`

Important

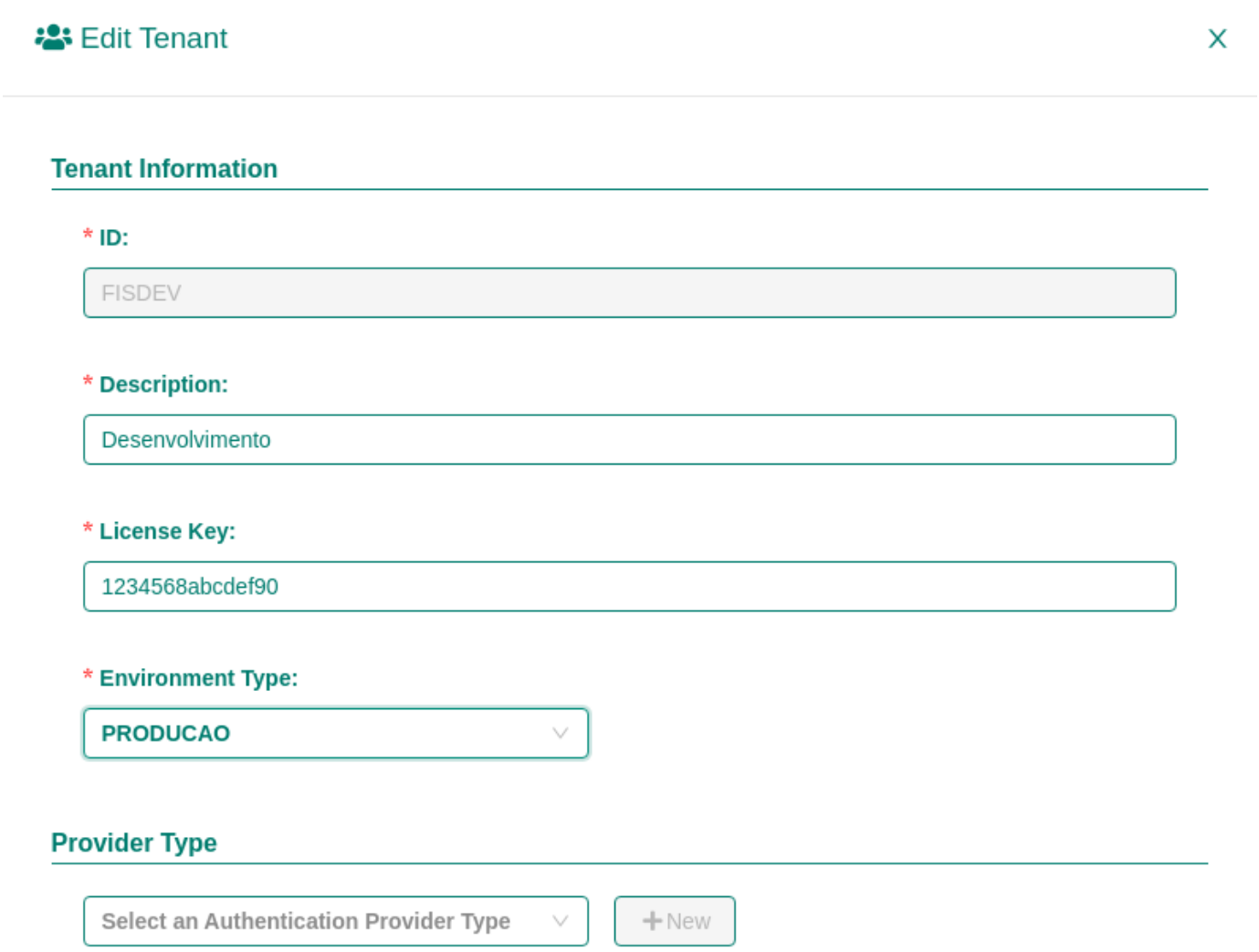
It is also important to make sure that specific address is not being blocked by your firewall.

License Requirements

In order to ensure a correct performance of licenses module, must be informed:

- License key
- Environment Type

Fill in the information requested in Tenant information.



About the License key

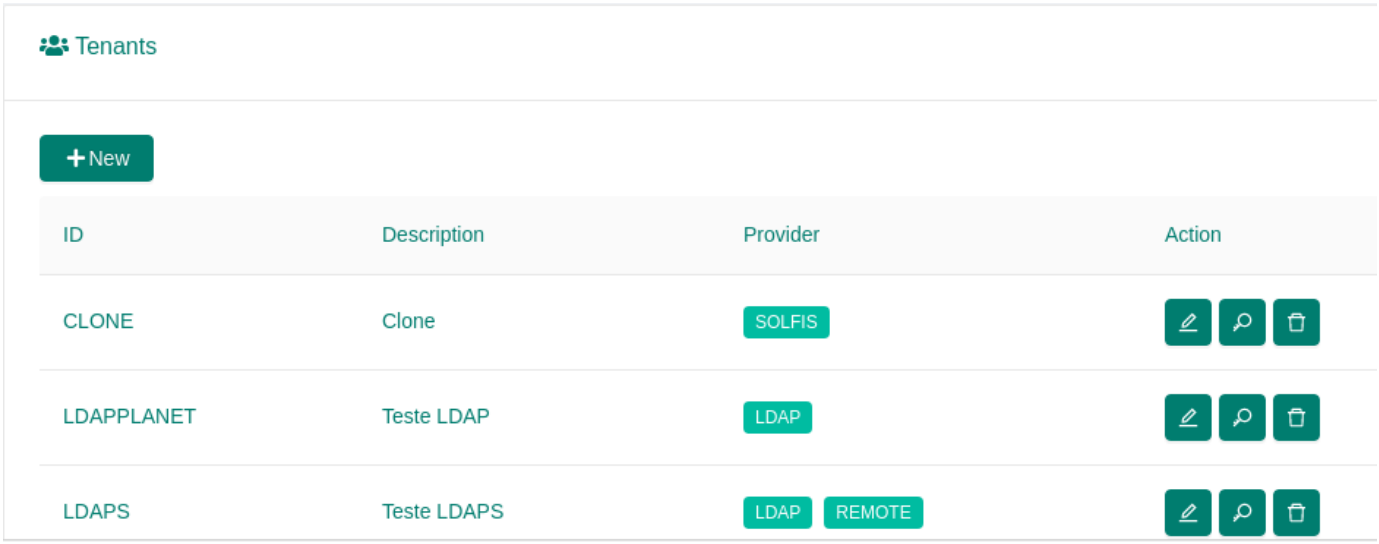
The customer will receive the `license key` by a specific department.

Not received yet? For now, fill in with any value.

License Activation

Activation is the process of activating a license that allows you to use the application until the license expires.

Use the License Activation button in Tenant list to start the licensing process.



Failed License Activation

If license activation fails, you'll see Unlicensed Product in the title bar of your Foundation apps, and access to apps are disabled. To restore all features of apps, you'll need to fix the problem that's causing activation to fail.

Developer Questions

Endpoints available for Licenses Module

Your app can make requests to the following REST endpoints:

Actions

- GET Validation (/validation)
- POST Collect (/collect)

Token

Don't worry about Token, it is generated by Foundation with Tenant ID.

Validation

Service of periodical validation of software license to ensure that the products are up-to-date and operating correctly.

Code samples:

Object Payload (JSON)

```
{
  "tenant":"TESTE"
}
```

Shell

```
curl -X GET \
  -H "Content-type: application/json" \
  -H "Accept: application/json" \
  -d '{"tenant":"TESTE"}' \
  -H 'Cookie: JSESSIONID=node0uyah95og25441xcb4r052xorx27.node0;
FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
  "http://127.0.0.1/licenses/validation"
```

Success Responses


```
{
  "success":true,
  "result":"Success Validation",
  "details":null,
  "content":{
    "sistemas":[
      {
        "sistema":"SFISC",
        "produtos":[
          {
            "codigocliente":"XXX",
            "sistema":"SFISC",
            "tituloproduto":"Governança",
            "siglaproduto":"OBR_CLOUD",
            "statusproduto":"ATIVO",
            "quantidadesites":1,
            "quantidadeestabelecimentos":7,
            "quantidadeusuarios":2,
            "quantidadetransacoes":0,
            "quantidadeempregados":0,
            "datainiciovigencia":"Apr 1, 2017 12:00:00 AM",
            "dataterminovigencia":"Mar 31, 2018 12:00:00 AM"
          },
          {
            "codigocliente":"XXX",
            "sistema":"WEB",
            "tituloproduto":"Apuração de Contribuições",
            "siglaproduto":"APURA",
            "statusproduto":"ATIVO",
            "quantidadesites":4,
            "quantidadeestabelecimentos":3,
            "quantidadeusuarios":4,
            "quantidadetransacoes":0,
            "quantidadeempregados":0,
            "datainiciovigencia":"Jun 1, 2016 12:00:00 AM",
            "dataterminovigencia":"May 31, 2017 12:00:00 AM"
```

```
          }
        ]
      }
    ]
  },
  "version":""
}
```

Error Response

```
{
  "success":false,
  "result":"An error occurred while trying to read tenant details",
  "details":null,
  "content":{
    "tenant":"",
    "environment":"",
    "error":"Tenant not found",
    "response":""
  },
  "version":""
}
```

Collect

Service of periodical collect of software information, version and environment.

Code samples:

Object Payload (JSON)

```
{
  "tenant":"TESTE",
  "siglaProduto":"XsX",
  "versao":"1.4v",
  "parametros":[
    {
      "chave":"PARAM0001",
      "valor":"1"
    }
  ]
}
```

Shell

```
  curl 'http://127.0.0.1/licenses/collect' \
  -H 'Cookie: JSESSIONID=node0uyah95og25441xcb4r052xorx27.node0;
FOUNDATIONID=eyJhbGciOiJIUzI1NiJ9' \
  --data-binary '{"tenant":"TESTE","siglaProduto":"XsX","versao":"1.4v","parametros":
[{"chave":"PARAM0001","valor":"1"}]}' \
  --compressed
```

Success Responses


```
{
  "success":true,
  "result":"Success Collect",
  "details":null,
  "content":{
    "msg":"Coleta salva com sucesso!"
  },
  "version":""
}
```


Error response


```
{
  "success":false,
  "result":"An error occurred while trying to collect information",
  "details":null,
  "content":{
    "errorCode":150,
    "message":"Parâmetros obrigatórios: token, siglaProduto, versao, ambiente e possuir pelo menos 1
parâmetro"
  },
  "version":""
}
```

View is a Foundation's module responsible for display a web page that is viewed in an Internet browser.

Access

You can access foundation by accessing `http://<server-ip>:<foundation-port>/foundation`

Get link to acess Foundation view:

```
sudo foundation config --get-foundation-link
```

Video

Main features:

- Login page
- Application
- Certificates
- Environments
- Tenants

Login page

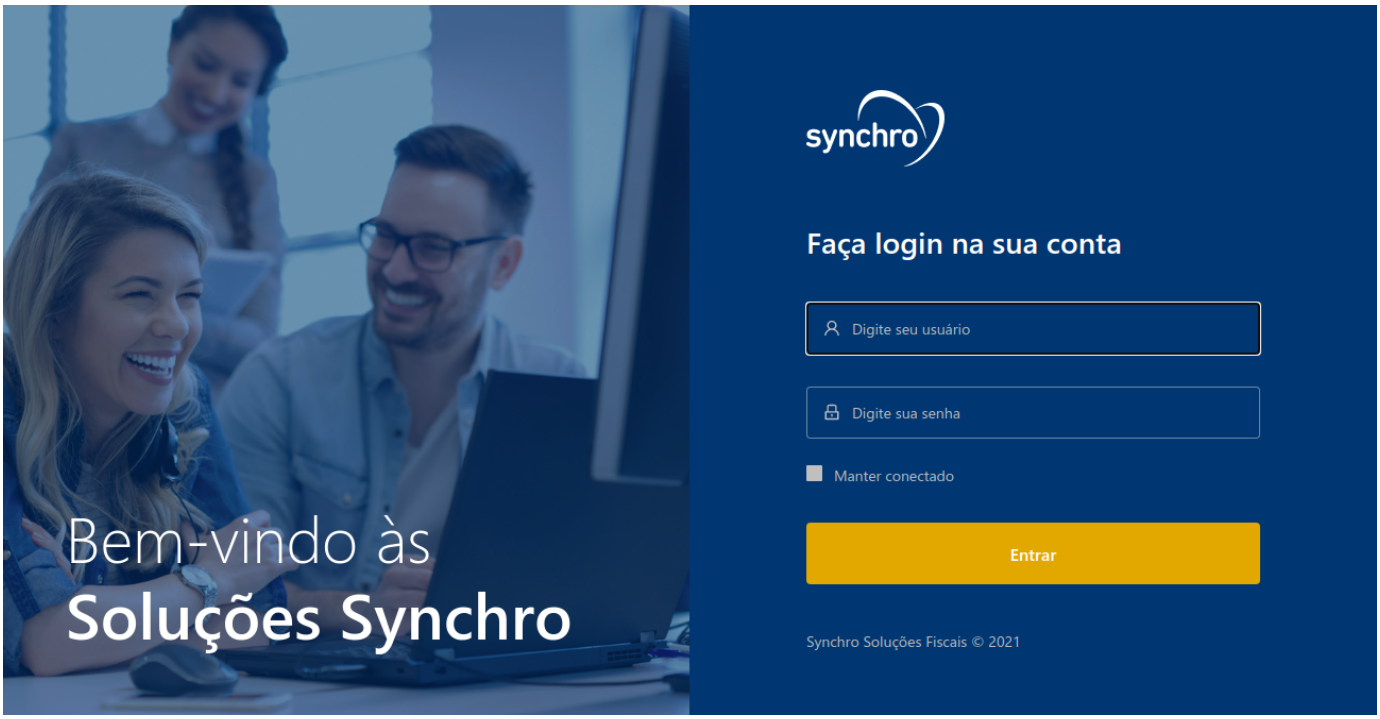
When you open the foundation or any protected application in our platform in a web-browser, you are redirected to foundation authlayer login page.

Do login with your keycloak user or `Synchro default user` .

Synchro default user

user: `synchro`

pass: `Synchro@123`



Application

An application, also referred to as an application program or application software, is a computer software package that performs a specific function directly for an end user or, in some cases, for another application. An application can be self-contained or a group of programs.

Important

Users with `FOUNDATION_ADMIN` role is required.

Certificates

Foundation certificates is a certificate module, after install you will be able to manage KeyStore and TrustStore using any HTTP browser.

For more information, see certificates.

Important

Users with `FOUNDATION_ADMIN` or `FOUNDATION_CERTIFICATES` role is required.

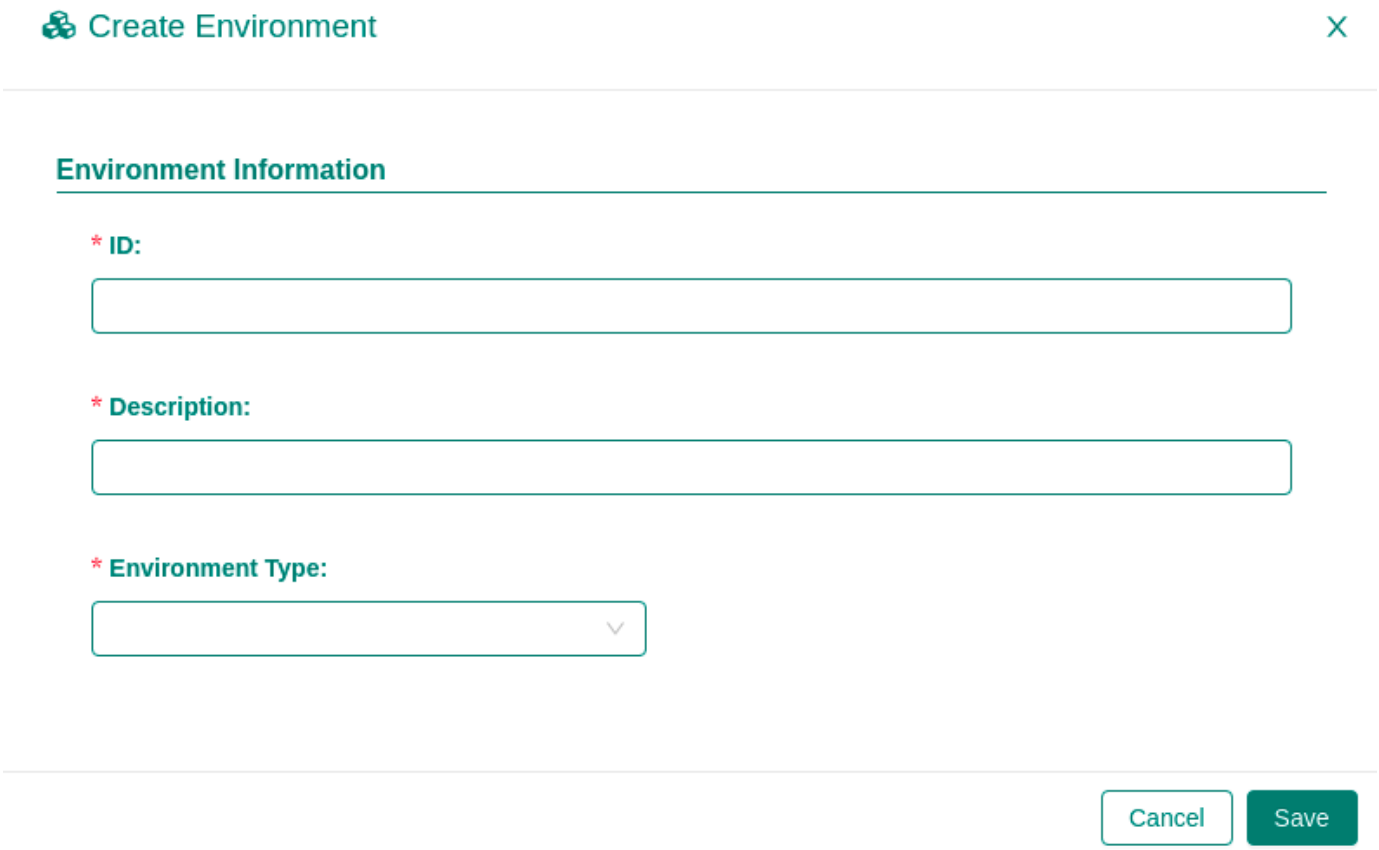
Environments

Definition of server environment. This information is used by Tenants.

> **Important**
>
> Users with `FOUNDATION_ADMIN` role is required.

Available environment default list

Property	Description	Type
DESENVOLVIMENTO	Ambiente de desenvolvimento	DEV
ACEITE	Ambiente de Aceite	UAT
HOMOLOGACAO	Ambiente de Homologação	QA
PRODUCAO	Ambiente de Produção	PROD
QA	Ambiente de QA	QA
UAT	Ambiente de UAT	UAT
DEV	Ambiente de DEV	DEV



ID

A unique identifier. Only uppercase letters and numbers, numbers not allowed at first character.

Description

Environment description.

Type

There are four different work environment types.

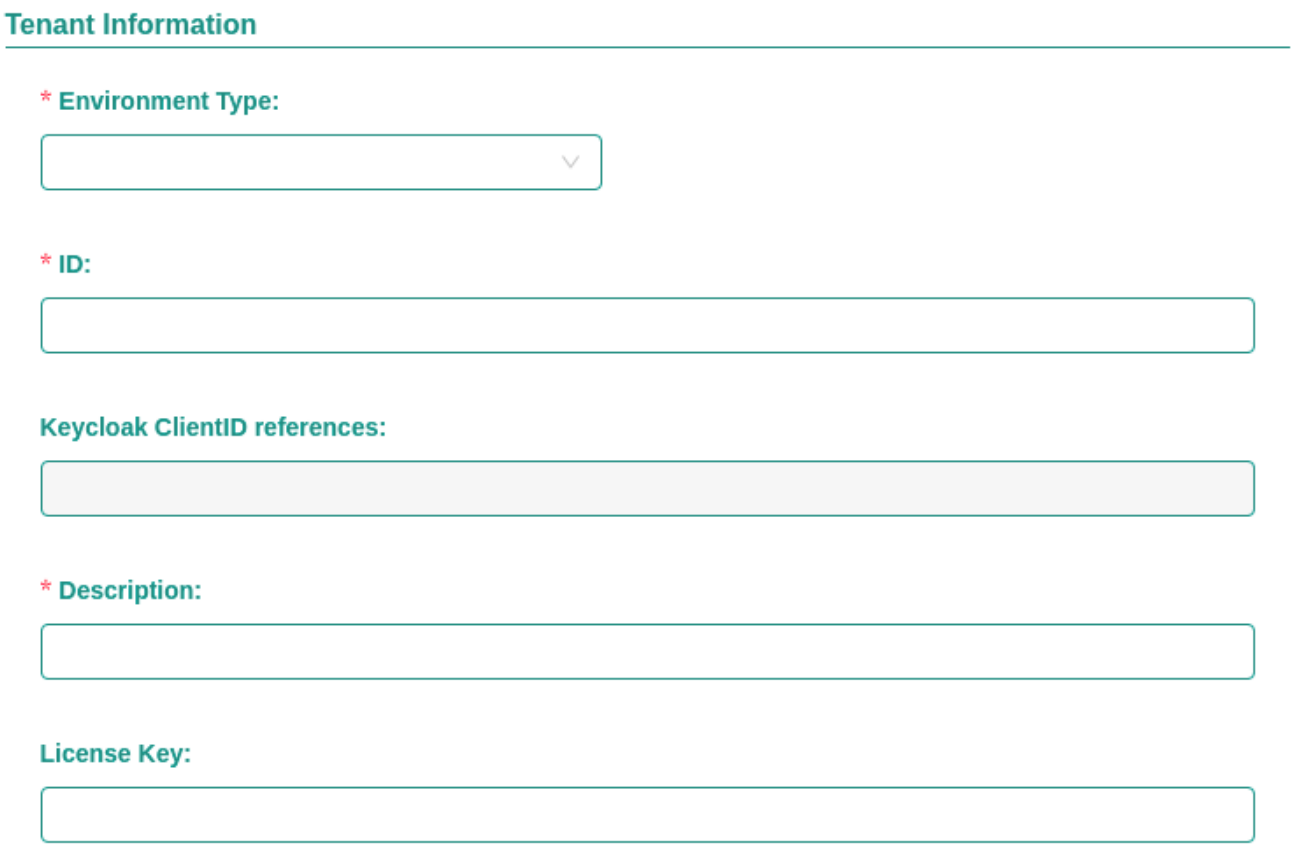
Property	Description
DEV	Development environment
PROD	Production environment
QA	Quality Assurance environment

Property	Description
UAT	User acceptance testing environment

Tenants

Tenant is a group of information about existing database, enviroment, license, etc. This information is used by applications.

> **Important**
>
> Users with `FOUNDATION_ADMIN` role is required.



Environment

For more information, see Environments.

ID

A unique identifier. Only uppercase letters and numbers, numbers not allowed at first character.

Keycloak ClientID references

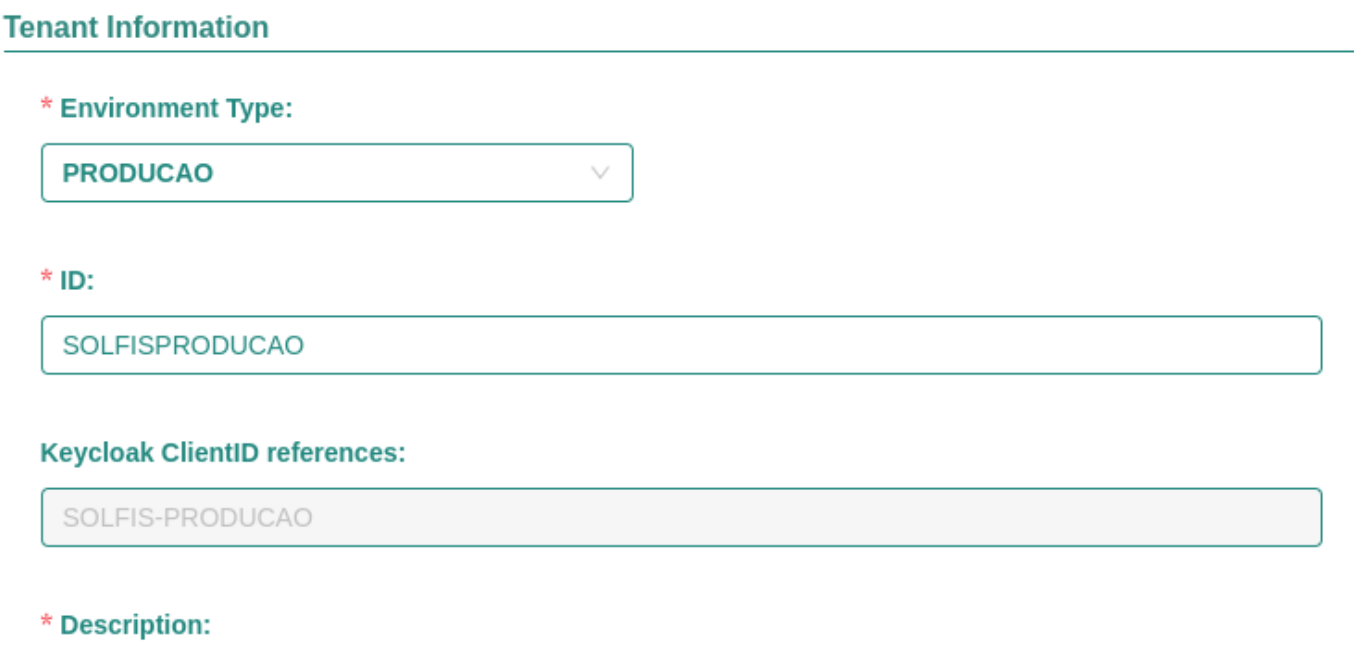
The Foundation are creating a Keycloak Clients references with TenantId and Environments selected.

Keycloak Clients example

Environment Type: PRODUCAO

TenantID: SOLFISPRODUCAO

ClientID: SOLFIS-PRODUCAO



Description

Tenant description.

License Key

Licenses is a Foundation's module responsible for providing the information for Synchro customers to license the contracted products into Foundation.

About the License key

The customer will receive the `license key` by a specific department.

Not received yet? For now, fill in with any value.

For more information, see licenses.

Providers available

Currently 6 providers types are supported:

1. Solfis (Solução Fiscal)
2. DFe Manager (Documentos Fiscais Eletrônicos)
3. Agr (Automação de Guias de Pagamento)
4. Gestaocreditos (Gestão de créditos)
5. Sfw (Solução Fiscal Web)
6. Variables

How To Use

JDBC string	Service Name	SID

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=server_name)(PORT=port))
(CONNECT_DATA=(SERVICE_NAME=serviceName)))

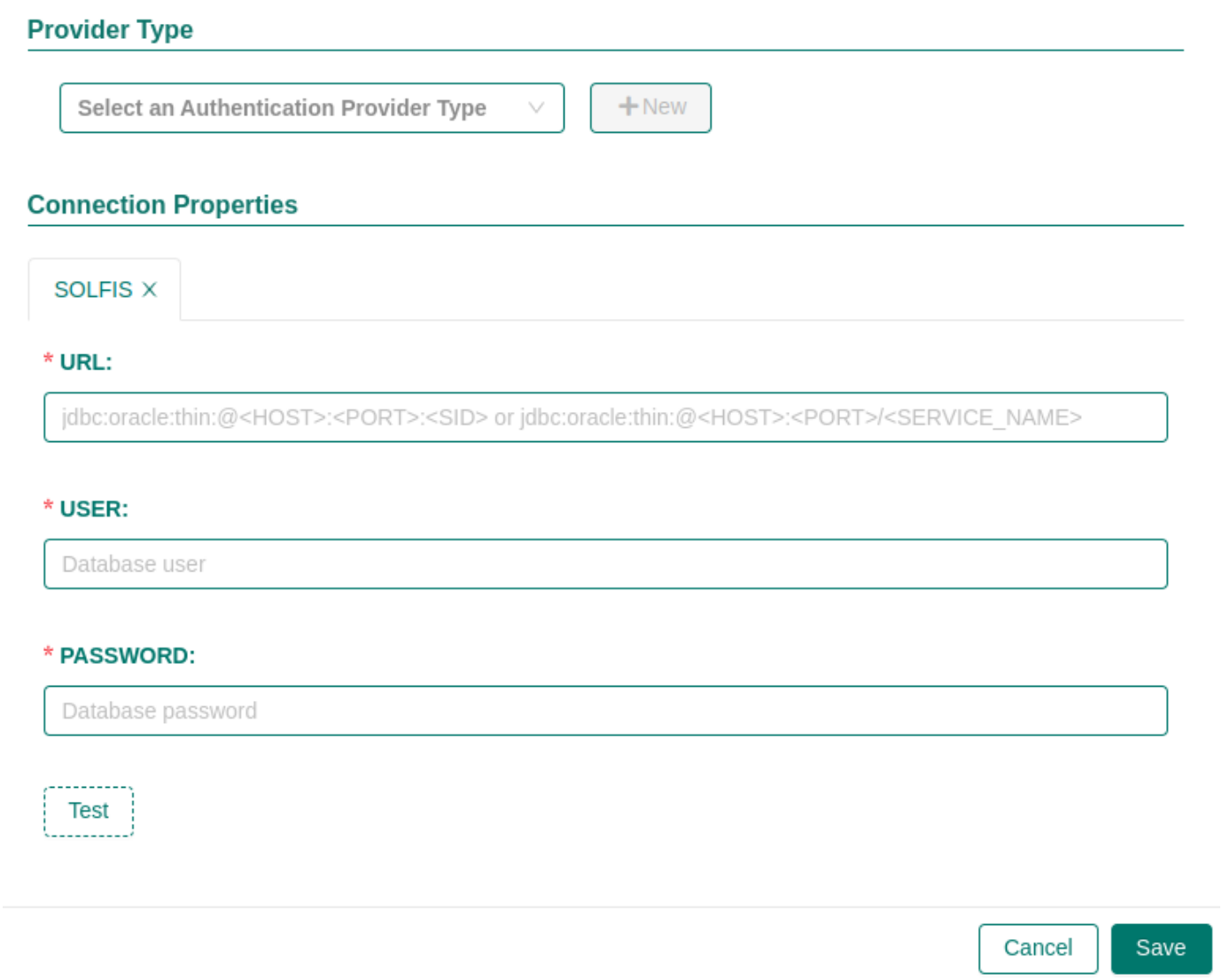
jdbc:oracle:thin:@<HOST>:<PORT>/<SERVICE_NAME>

jdbc:oracle:thin:@<HOST>:<PORT>:<SID>
```

SolFis

Solução Fiscal

You only need to inform JDBC Oracle Connection String, Database User Name and Database User Password.

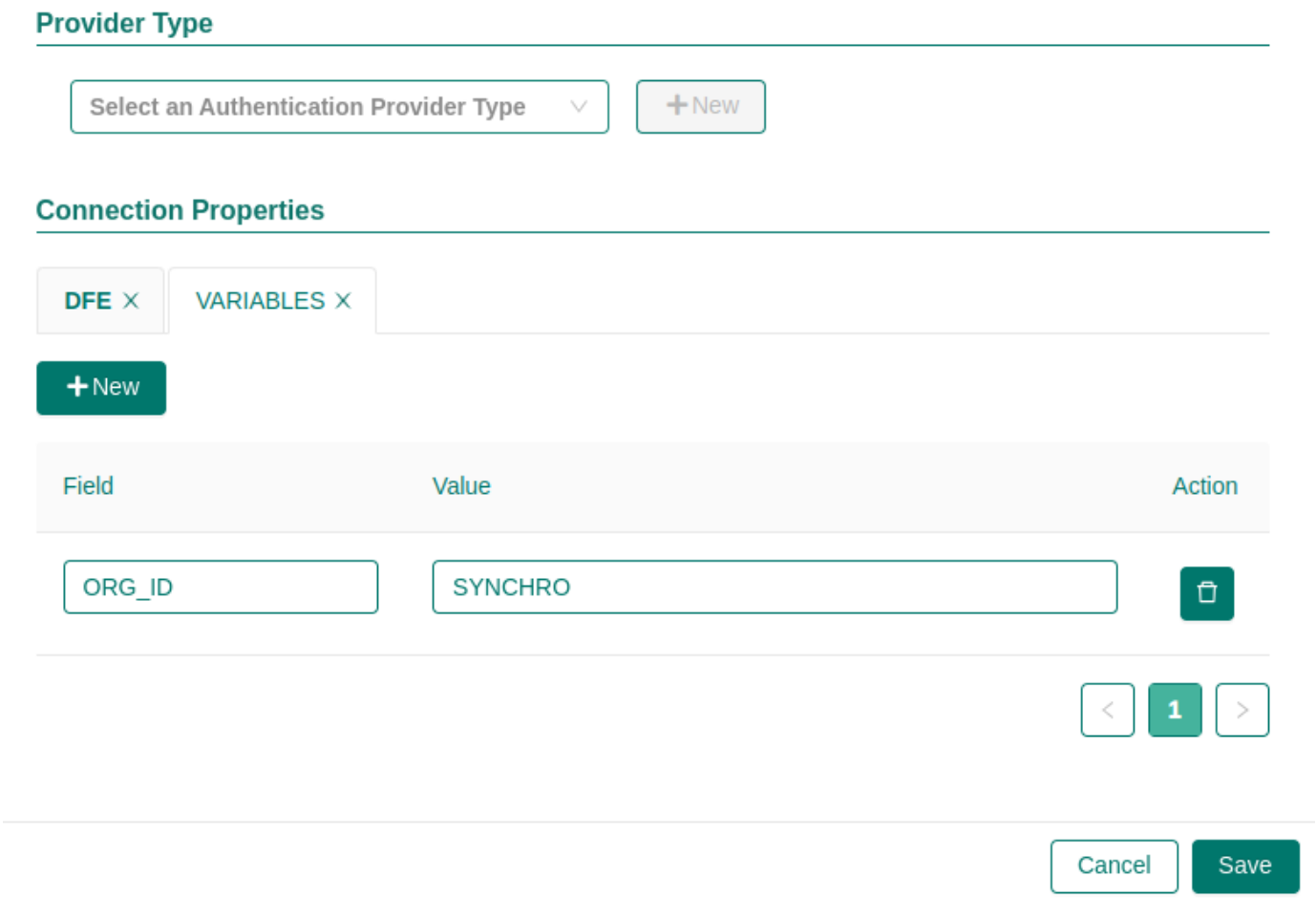


DFe Manager

Documentos Fiscais Eletrônicos

DFe provider requires:

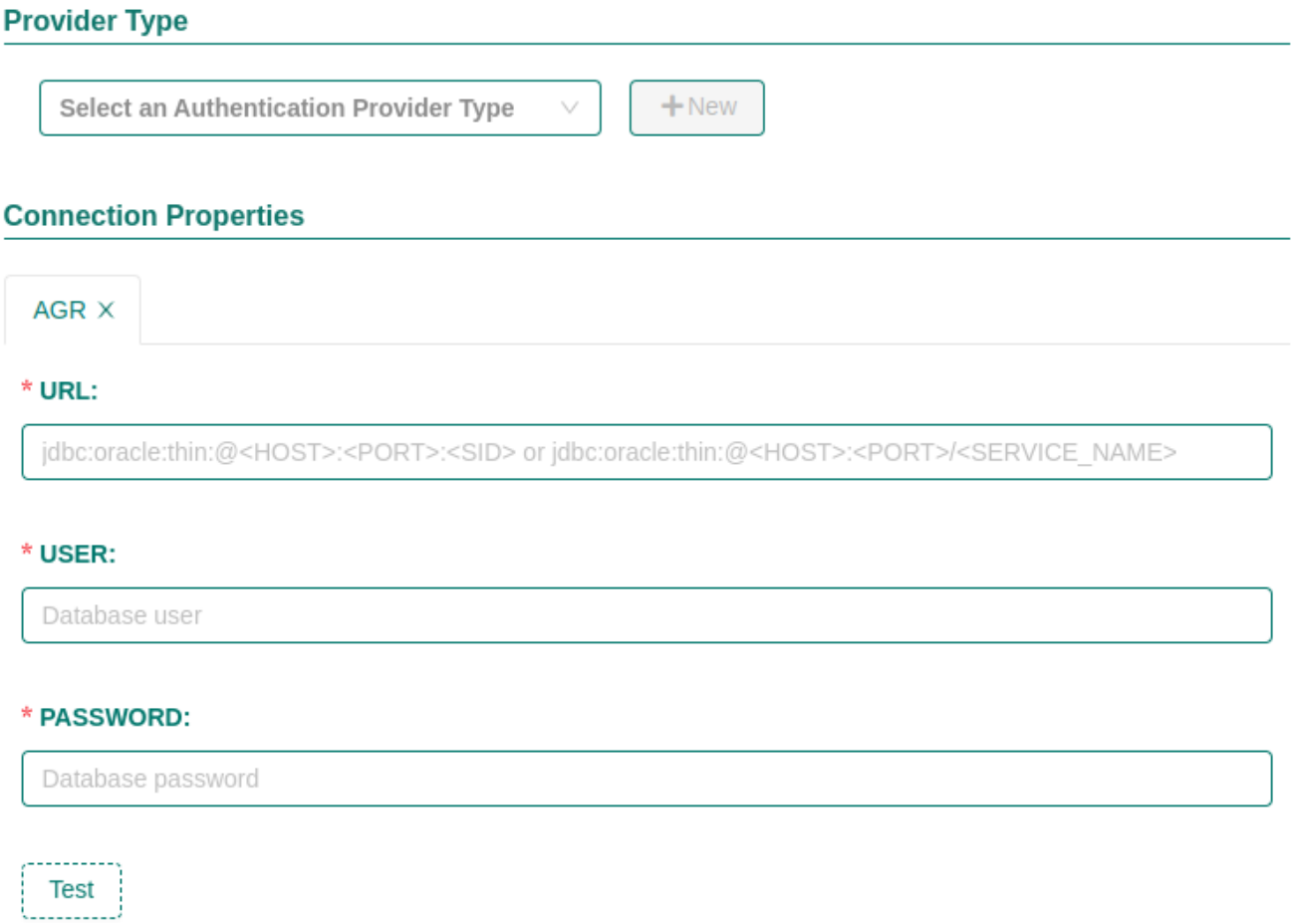
- Inform JDBC Oracle Connection String, Database User Name and Database User Password.
- As a complement it is required inform field ORG_ID at tenant variables provider.



Agr

Automação de Guias de Pagamento

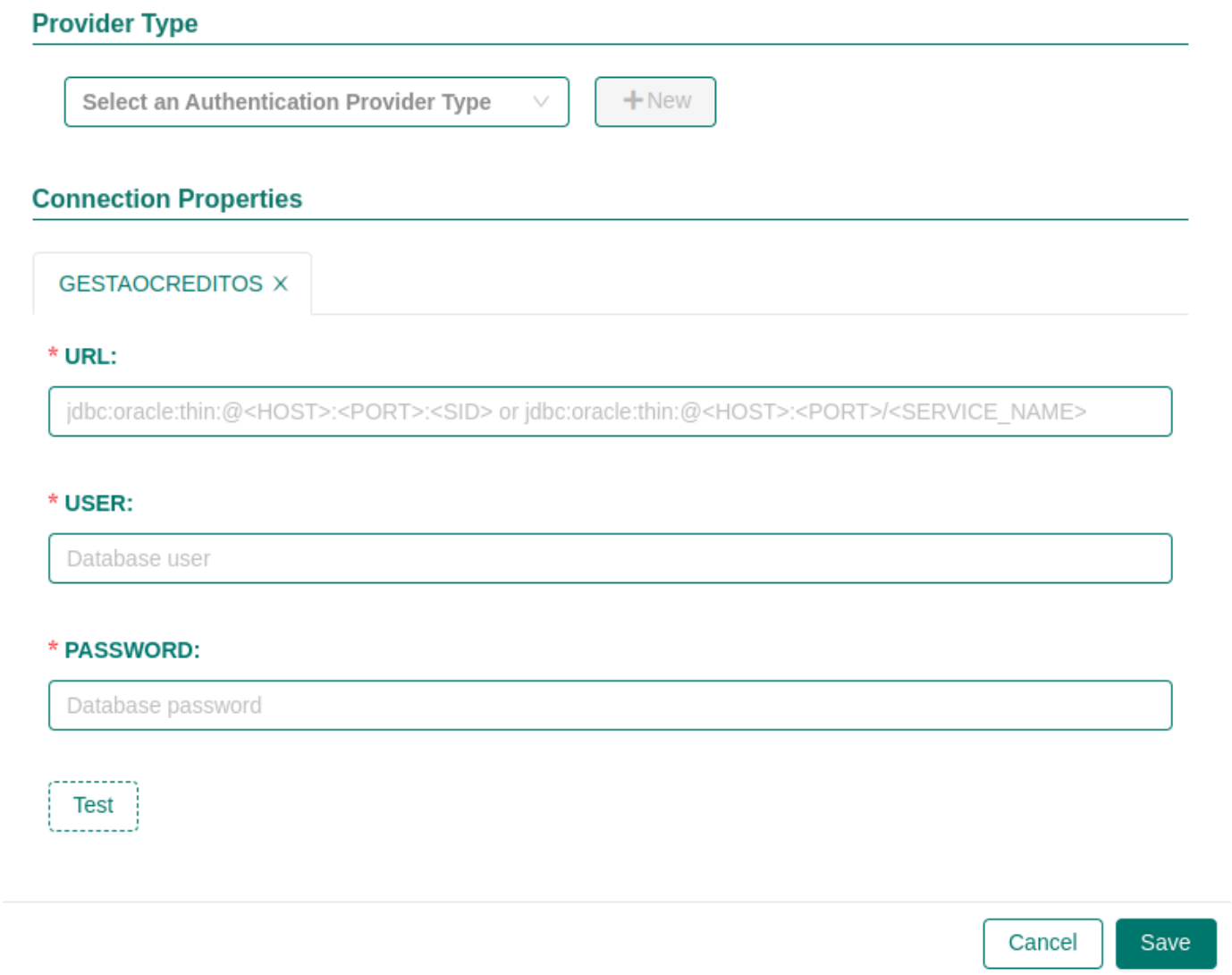
You only need to inform JDBC Oracle Connection String, Database User Name and Database User Password.



Gestaocreditos

Gestão de créditos

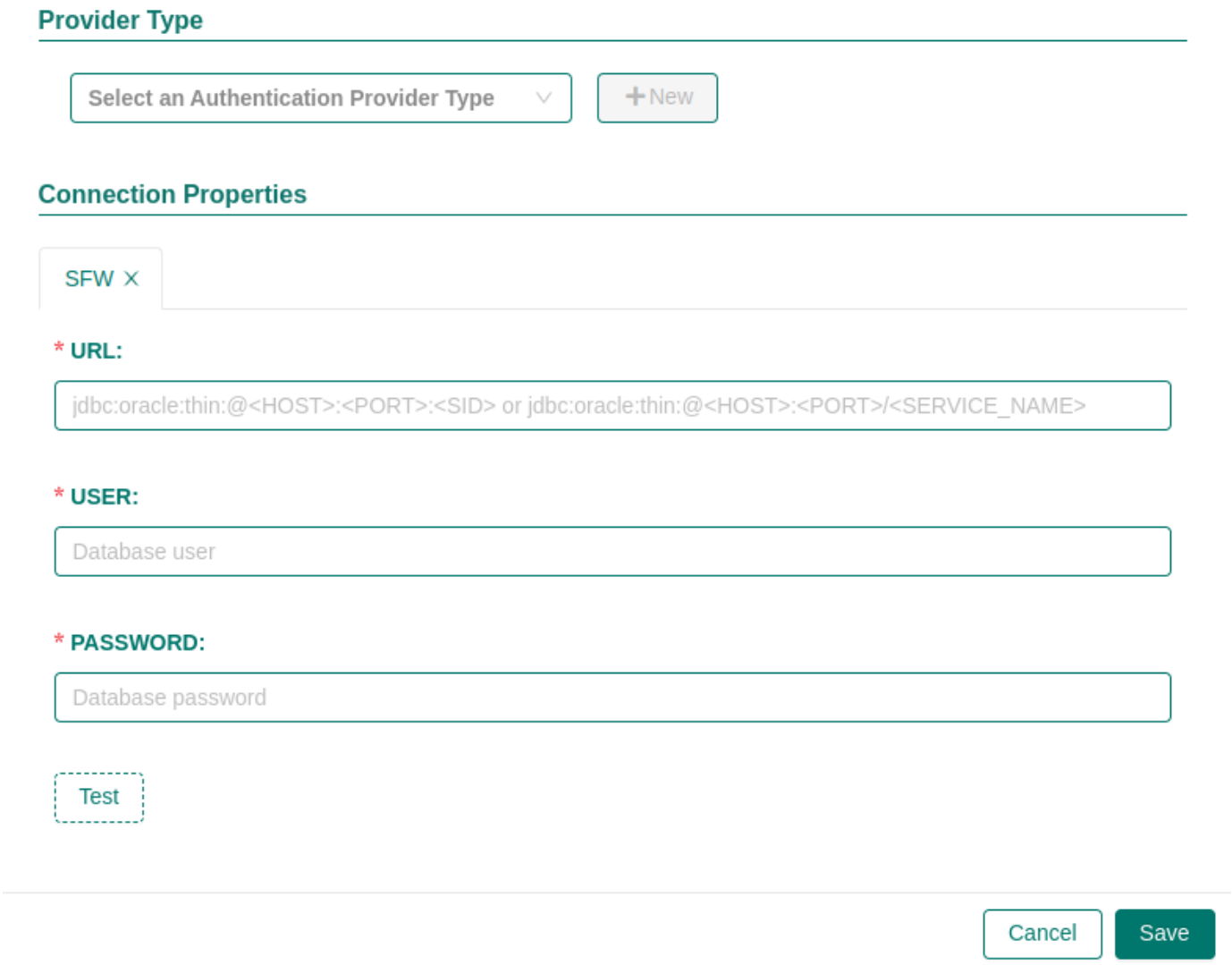
You only need to inform JDBC Oracle Connection String, Database User Name and Database User Password.



Sfw

Solução Fiscal Web

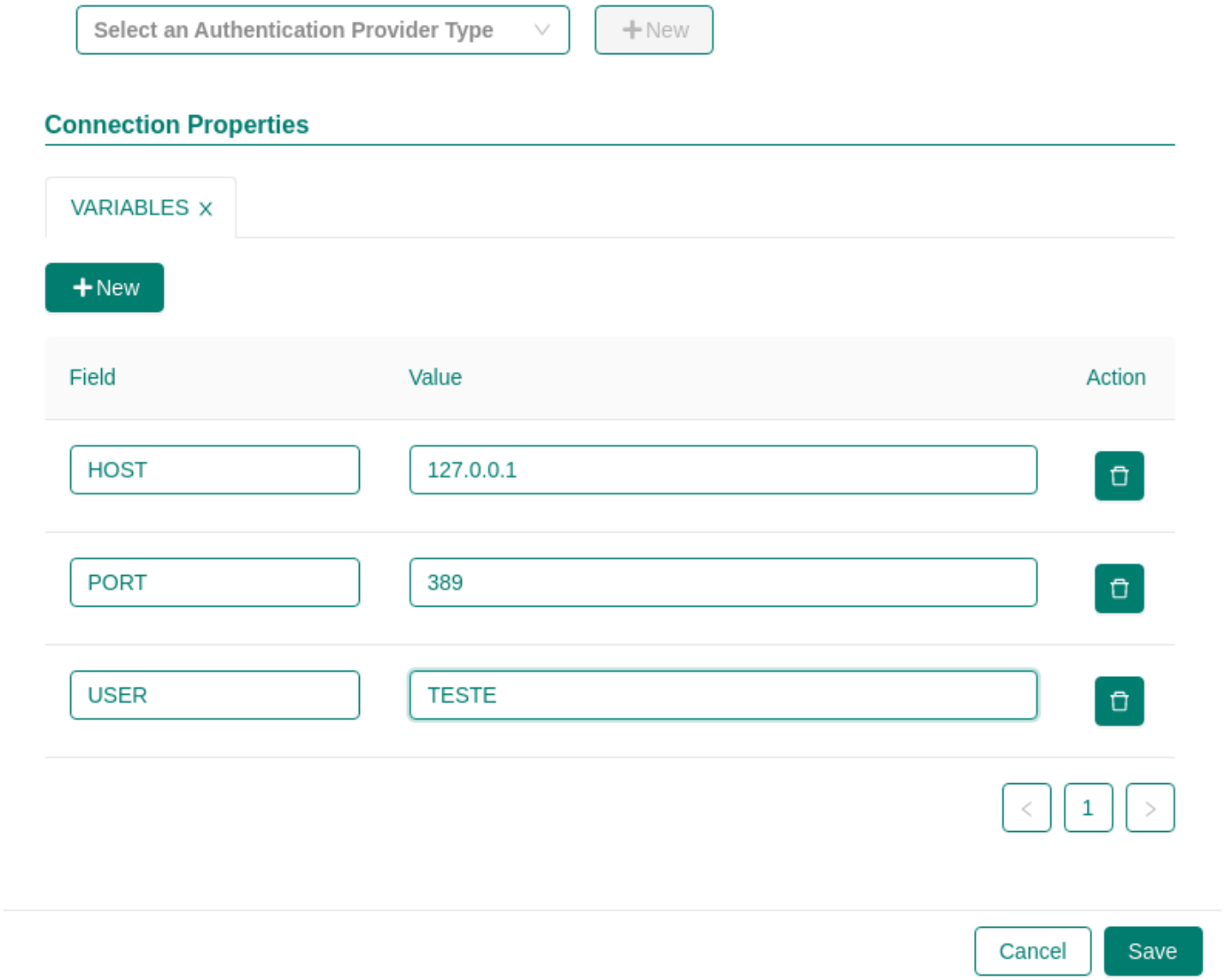
You only need to inform JDBC Oracle Connection String, Database User Name and Database User Password.



Variables

With Variables provider, it is possible to create a provider configuration by tenant (key, value).

> **Warning**
>
> The variables provider is not Environment Variables



Developer Questions

For more information, see internal tenant.

Directory Structure

Overview

This page describes Foundation directory struture.

Foundation directory

The Foundation directory contains your applications, data, images, certificates, etc.

With this version, the `deployed` folder has been moved from `applications` to a new folder `deployments`.

Before structure (version 21.09.23 and earlier):

```
root:/foundation/system/default/storage/foundation/default#
├─ applications/
|  ├─ deployed/
|  ├─ foundation/
|  └─ reinf/
├─ keystore/
└─ truststore/
```

Current structure:

```
root:/foundation/system/default/storage/foundation/default#
├─ applications/
|  ├─ foundation/
|  └─ reinf/
├─ deployments/
|  └─ deployed/
├─ keystore/
└─ truststore/
```

Applications

About applications struture:

```
root:/foundation/system/default/storage/foundation/default/applications#
├─ foundation/
|  ├─ authlayer/
|  ├─ certificates/
|  ├─ view/
```

```
| └ .../
└ reinf/
| ├ amqp/
| ├ cache/
| └ core/
...
```

Deployments

About deployments struture:

```
root:/foundation/system/default/storage/foundation/default/deployments#
├ deployed/
| ├ foundation-authlayer/
| ├ foundation-certificates/
| ├ foundation-view/
| └ .../
└ history/
| └ applications/
|   ├ foundation/
|   | ├ authlayer/
|   | ├ certificates/
|   | ├ view/
|   | └ .../
|   └ reinf/
```

Deployed

All modules deployed there is a file in deployed folder.

History

Save information about module version deployed.

1. Foundation reload a latest config deployed to new module version without suggestion.
2. The registry deploy settings and undeploy module version is saved.

> **Tip**
>
> To clean all history deployments: `sudo foundation clean --history`

Foundation

This is the main doc for Foundation's command line, known as `foundation`. It's a tool delivered with Foundation that allows you to monitor and manage your apps.

Core Concepts

Before you get into the command line, you need to get familiar with some core concepts.

Foundation is built around 3 main concepts:

- **Image**: contains all the instructions needed to run the app, that is all the compiled code in a freeze state.
- **Instance**: is a running app created based on an image.
- **Service**: is responsible for managing the life cycle of an instance.

Basic Commands

Help

```
foundation --help
```

```
NAME:
   foundation - Synchro Foundation Client

USAGE:
   foundation [global options] command [command options] [arguments...]

VERSION:
   21.09.23

DESCRIPTION:
   Foundation is a platform for Synchro applications. You can learn more at: https://foundation.synchro.com.br

COMMANDS:
   application, app  Manage apps
   clean             Clear unused (old) data from foundation directories
   completion        Return scripts for shell autocompletion configuration
   config, setup     Setup Foundation config (requires root)
   info, check       Display system-wide information
   login             Starts a new Foundation session
   logout            Closes current Foundation session
```

```
   module          Manage Foundation's modules (login required)
   package         Create a app-module-version.module file for use in Foundation
   start, up       Starts Foundation
   status          Health check report
   stop, down      Stops Foundation
   tenant          Manage Foundation's tenants (login required)
   user            Manages user access to foundation services (requires root)
   version         Print client version
   web, www        (Experimental) Starts a web server on provided port (default 8082) with a interactive web
interface, to configure foundation
   help, h         Shows a list of commands or help for one command

GLOBAL OPTIONS:
   --help, -h     show help (default: false)
   --version, -v  print the version (default: false)
```

start

Boots up Foundation with services.

```
foundation start
```

```
INFO[0000] Starting foundation 23.08.03
INFO[0000] [Foundation Core] Starting services
INFO[0010] [Foundation Module] Proxy service created
INFO[0011] [Foundation Module] Storage service created
INFO[0013] [Foundation Module] Supervisor service created
INFO[0013] [Foundation Core] Foundation Core Components starting. Please wait... (1)...
INFO[0045] [Foundation Core] Foundation Core Components starting. Please wait... (2)...
INFO[0109] [Foundation Core] Foundation Core Components starting. Please wait... (3)...
INFO[0109] [Foundation Core] Started
INFO[0109] [All modules] Starting
INFO[0110] Stop command sent locally for foundation/engine.
INFO[0111] Starting foundation/engine:23.08.03...
...
```

config

Setup foundation config (requires root).

```
foundation config
```

For more information, see configuration.

print

Print Setup foundation config.

```
foundation config --print --profile-file /etc/foundation/default.settings
```

login

Create a session for commands that requires authentication. The login is only valid to current user. If you do a login with `sudo` it's not valid for non-root users and vice-versa.

Do login with your keycloak user.

```
foundation login
```

```
INFO[0000] Foundation URL: http://0.0.0.0:80

QUESTION: Login: username

QUESTION: Password:
INFO[0004] Login succeeded
```

Info

```
foundation check
```

```
INFO[0000] Starting basic requirements check...
WARN[0000] Linux distro: [ubuntu 20.04] not tested
INFO[0000] [5.4.0-8] Kernel:    OK
INFO[0000] Available Memory: [6950M]    OK
INFO[0000] ip_forward   OK
```

Status

List all Foundation's services statuses.

```
foundation status
```

```
INFO[0002] Current Session: http://0.0.0.0:80
INFO[0002] proxy        OK      0.0.0.0:80
INFO[0002] supervisor   OK
INFO[0002] storage      OK
INFO[0002] engine       OK
INFO[0002] postgres     OK
INFO[0002] keycloak     OK
INFO[0002] authlayer    OK
INFO[0002] view         OK
INFO[0002] certificates OK
INFO[0002] logs         OK
```

```
INFO[0002] licenses     OK
INFO[0003] monitor      OK
```

autofix

Stop previous modules and start the modules from core version.

```
foundation status --autofix
```

stop

Turns off Foundation system gracefully.

```
foundation stop
```

```
INFO[0004] Stopping all services at http://0.0.0.0:80
INFO[0004] Services stopped successfully
```

Modules commands

module add

Add a foundation module.

```
foundation module add path/to/file.module
```

Adding other foundation modules

Adding other modules consists of doing the same steps mentioned above. Just doing now for the chosen new module.

```
foundation module add /path/to/otherApp-moduleName-version.module
```

```
foundation module start --app otherApp --name moduleName --version <version>
```

module start

Start a module.

```
foundation module start --app <app-name> --name <module-name> --version <module-version>
```

module status

List foundation modules statuses.

```
foundation module status --app <app-name> --name <module-name>
```

module stop

Start a module.

```
foundation module stop --app <app-name> --name <module-name>
```

module remove

Remove a module.

```
foundation module remove --app <app-name> --name <module-name> --version <module-version>
```

Uninstall Synchro Foundation

This section describes how to uninstall Synchro Foundation on RPM and Deb based distros.

`asciinema(../assets/uninstall-foundation.asciinema)`

Gracefully Shutdown

Before uninstalling Foundation, make sure to run `$ foundation stop`.

Backup

Foundation has a configuration file in `/etc/foundation/*.settings`. After uninstalling or reconfigure Foundation, this file is removed or overwritten according to the operation ran.

Data

Foundation setup data are stored in `/etc/foundation` folder, and foundation applications data are in `<foundatipn-path>/system/`. Uninstall does not remove this folder.

RPM-like distros (CentOS, Fedora, Oracle Linux and RedHat)

1. Run the following command to remove Foundation's binary:

```
sudo yum remove synchro-foundation
```

Ubuntu

1. Run the following command to remove Foundation's binary:

```
sudo apt-get remove synchro-foundation
```

Suse Linux

1. Run the following command to remove Foundation's binary:

```
sudo apt-get remove synchro-foundation
```

Release Notes

The foundation's team strongly recommends to keeping your environment up to date with the latest version

24.04.10

- foundation-authlayer:
  - Encrypt database passwords;
- foundation-certificates:
  - Encrypt keystore password on upload;

23.12.15

- foundation-cmd:
  - Fix set-context for k3s provider.
- foundation-authlayer:
  - Add User ClientId and Environment to FOUNDATIONID JWT
  - Add HTTPOnly = true to FOUNDATIONID JWT
  - Add Secure = true to FOUNDATIONID JWT
- foundation-view:
  - Fix keycloak client id integration at tenant store
- foundation-keycloak:
  - Fix keycloak login page for non synchro.com.br providers
- Documentation:
  - New Oracle Linux 9 and Red Hat 9 .rpm installation files
  - Keycloak Advanced:
    - Add more details to client authentication steps.
    - Add VALID REDIRECT URI recomendation and instructions.
  - Configuration:
    - Add Synchro4me DNS Requitements information.

        - Add more details about HTTPS/TLS .cert and .key files cryptography, format, folder pemissions and configuration.
        - Add Epel repo and yum update instruction for RHEL 8 and 9 .
    - Troubleshooting:
        - Add NO_PROXY configuration steps to solve pod logs issues.
        - Add reference to Keycloak Advanced Valid Redirect URIs
        - Add how to solve AWS EC2 nm-cloud-setup.service issue

23.10.03

- foundation-cmd:
    - Fix context to default when provider is k3s at Config and Start
    - Verify if Keycloak URL Settings is blank at Supervisor /status
    - Supervisor GetKeycloakSettings new blank validations
- foundation-authlayer
    - New tenant database test
    - LoadKeycloakSettings status code validations
- foundation-view
    - Add xss protection
- foundation-keycloak
    - Fix /tmp permissions inside pod
- Documentation
    - New tenant database test examples

23.08.31

- foundation-cmd:
    - Insert foundation namespace value like option
    - Create `foundation` namespace suggestion
    - Create `foundation` bucket
    - Fix securityKey regenerate
- foundation-view:
    - EnvironmentID accepting numbers at the beginning

- Documentation:
  - Install
  - Configuration
  - Start
  - Update
  - Troubleshooting
  - Keycloak Advanced Configuration

23.08.11

- foundation-cmd:
  - Bugfix: Ask for server address loop

23.08.09

- foundation-cmd:
  - Insert question to skip k3s installation
  - Fix k3s get loadbalancer ingress IP
- Documentation:
  - Update troubleshooting items
  - Update Keycloak URL configuration
  - Update command line items

23.06.07

- General:
  - Implements Healthcheck timeout to spec.yaml
  - Increase Engine default memory to 200mb
- foundation-cmd:
  - Fix .rootcheck file creation
- Documentation:
  - Update memory requirements

23.06.06

- foundation-cmd:
  - Fix default foundation registry
  - Renew local session before start others foundation modules

23.06.05

- foundation-cmd:
  - ImagePullPolicy relative with registry access
  - Fix oci/oke commands relative path
- foundation-view:
  - Show module deployed when status is NotFound;
- foundation-authlayer:
  - Fix user logout where session is expired;
  - Update mount clientID based on subdomain;

23.05.29 (breaking-change)

- General:
  - New sequence of `foundation config` information
  - Enable TLS/HTTPS to kubernetes;
- foundation-logs:
  - Fix permission to access modules logs;
- foundation-authlayer:
  - Create API to support external login;
- foundation-view:
  - New deploy tab to resize memory apps;
  - Fix image import from full module;
  - New button to stop all modules from application;
- Documentation:
  - Update docs with new instructions and videos;

23.03.20 (breaking-change)

- General:
  - Fix, set k3s version to use stable v1.25.6+k3s1
  - Increase healthcheck Interval and StartDelay value
- foundation-keycloak:
  - Fix keycloak initial import scripts;

23.03.15 (breaking-change)

- General:
  - Remove verbosity from `foundation start` ;
  - Fix `foundation stop` ;
  - Update modules spec to load local images, if internet access is blocked;
  - Add cronjob and mirrored pause image to foundation full rpm;
- foundation-keycloak:
  - Update keycloak version from 16.1.0 to 20.0.5;
- foundation-logs:
  - Fix permission to access modules logs;
- foundation-licenses:
  - Update APIs to access authlayer module contracts.

22.11.07 (breaking-change)

- General:
  - ♯631, Creation of the new module foundation-authlayer with Keycloak integration:
  - ♯635, Creation of the new module foundation-keycloak;
  - Added kubernetes(k3s) for on-premises;
  - Create deployment history;
  - Removal of foundation-accounts module from foundation package;
  - Removal docker platform;

- foundation-supervisor:
  - BugFix:
    - ♯563, Fix intermittent error "404 page not found" when foundation started;
- foundation-view:
  - Added Environment type list;
  - Update user profile;
  - ♯652, Removal of DEFAULT_TENANT;
  - Removal of LDAP from internal tenants;
- foundation-client:
  - ♯668, Add postgres config to foundation settings;
- Documentation:
  - Update docs with new instructions;

21.09.23

- foundation-view:
  - Features:
    - Added tenant filter on tenant list and deployments release;
    - Added new button on tenant list to display modules when have an specific tenant;
    - Application module list has been changed to accept new feature like tenant list by module when the module is running;
  - BugFix:
    - Fix multitenancy condition at deploy modal
  - Improvements:
    - CSS for pagination has been changed to be clear when the page is selected;
    - Delete modules name from module version;
    - ♯621 Not show tenant list when multitenant=false at deploy modal;
    - ♯621 Updated module and submodules status at deployed card;
  - BugFixes:
    - Deploy modules services (without tenant);
    - Module was stopped and display different status like NotFound;
    - Cannot set property 'children' of undefined;

- foundation-supervisor:
  - BugFixes:
    - Fixed when delete all module version into modules file list;
    - ♯589 Upload application.module by view not works;
- foundation-client:
  - Improvement:
    - foundation status --autofix updating module release file;
  - BugFixes:
    - Fix short module stop command when appName has hyphen at the name
    - Fix short module stop command when has slash to separate app and module
- foundation-engine:
  - BugFixes:
    - Fix engine cluster role permissions to create services and crd resources
    - Fix system property FOUNDATION_ORCHESTRATOR that was hardcoded
    - Fix service removal on k8s module stop;
- foundation-accounts:
  - Improvement: ♯617 Added new URLs into remote provider to serve DFE, SFW and GESTAOCREDITOS applications;
  - BugFixes:
    - ♯628 Added LoginException when tenant does not exist;
    - ♯615 When admin password has been changed, accounts not works with new password;
- Documentation:
  - Improvements:
    - Updated foundation start page with more information;
    - Updated foundation-logs page with more information;
    - Updated troubleshooting page;
    - Developers: Added internal tenant page;

21.07.16

- General:
  - Modules foundation-license, foundation-logs and foundation-monitor has been added to initialize on foundation start
- foundation-accounts:
  - Feature: ♯575 updated DFE provider to do authentication(ORG_ID at variables provider is required);
  - Feature: ♯607 added new providers(SFW, AGR, GESTAOCREDITOS);
  - BugFix on authentication with user admin when tenant default not found;
- foundation-client:
  - Feature: ♯584 Added tenant control from commandline CRUD (Create, Recover, Update and Delete);
  - Feature: ♯590 BugFix on foundation status;
  - BugFix on session file creation when unix user name has symbols;
- foundation-supervisor:
  - added new enviroment variables to set log level;
  - Endpoint to list cards per tenant;
  - apps.yaml new format;
- foundation-storage:
  - memory limit increased to 100Mib;
- foundation-engine:
  - fixed complex submodules names in kuberentes;
  - added basic k3s support;
  - added title and description to foundation modules;
- foundation-view:
  - Show only cards allowed for the current logged tenant;
  - ♯605 Fix view to show tenant selection when groups porperties not exists
- foundation-license:
  - Parameters json struct fix on RequestPayloadCollect and RequestBodyCollect;
- Documentation
  - Removed non-root start instructions;
  - Fix alternate description for modules full and lite
  - Updated information about module foundation-accounts

21.06.14

- General:
  - foundation-license integrated to the rpm
- foundation-client:
  - BugFix local session file can be problematic when username has symbols;
  - BugFix module remove break when installed module is from a incompatible version;
  - BugFix module stop removes PID file from `deployed folder of missing modules;
- foundation-accounts
  - LDAP authentication fix
- foundation-monitor:
  - activating telemetry with jaeger
  - Prometheus for logs
  - grafana for dashboards
- foundation-view:
  - fix hidden cards;
  - fix card redirect path;
- foundation-supervisor:
  - redirect path fixed for login;
- foundation-accounts:
  - telemetry activated to jaeger;
  - upgrade to latest spring-boot 2.5.0;

21.05.23

- foundation-supervisor:
  - default login redirect fix
- foundation-view:
  - hidden card fix;
  - card redirect link fix;
- foundation-module:
  - grafana dashboard interface
  - prometheus logs management

- jaeger telemetry interface

21.05.12

- foundation-accounts:
  - BugFixes
    - ♯579 accounts don't set tenantid on userinfo when default tenant was logged;
- foundation-view:
  - Improvement: ♯580 Add user friendly title and description to cards;
  - Improvement: Only show modules with Hide=false property in spec.yaml;
  - Feature: ♯334 Add module redirect path for non admin users;
  - BugFixes
    - ♯516 on tenant screen save bottom does not enable whan using paste;
- documentation:
  - Improvement: ♯578 Insert foundation version in first documentation page;
- general
  - Improvement ♯576: Spec health check improved to allow define startDelay and Interval
  - engine and supervisor now listen port 80
  - Bugfix ♯334 Multiple exposed ports on same module fixed
  - Docker version upgrade to 20.10.6
- command line client
  - BugFixes
    - auto-detect server url when using ssl;
    - ♯548 wait all modules start before consider foundation start done;
    - ♯406 changing profile fixed;
    - profile saving fix on non-default ones;
    - ♯517 swarm init fix when server has multiple IPv4 addresses on selected interface
    - ♯508 Fix module remove message when module is running;
    - ♯548 REINF core don't start automatically on foundation start
  - features:
    - autodetect https storage from remote url;
    - default oci remote objectstorage detection;

        - ♯551 automatic legacy storage detection and copy/link files from 1.3.4 and 20.06.11;
        - ♯354 Kubernetes Support
            - upgraded to v1.18.10;
            - setup kubeconfig (context and namespace set);
            - oci - remote config/start;
            - store foundation config in a secret;
            - autodetect load-balancer address;
            - auto-configure traefik custom resource definitions;
            - kubectl foundation plugin;

- foundation-proxy:
    - traefik k8s config fix;
    - traefik upgrade to version 2.4.0 (latest);
- foundation-engine:
    - Added custom dns support;
    - bugfix:
        - X-Foundation-Proxied header fixed for unprotected routes;
        - ♯573 docker swarm keepPrefix not working, removed middleware reference on router to fix issue;
- foundation-accounts:
    - ♯571 multi-providers per tenant api (BREAKING CHANGE);
    - ♯575 add DFE provider;
- foundation-view
    - bugfix: #348 fix drag'n'drop unexpected behaviour when dropping file outside upload card;
    - bugfix: card link for non admin users on multiple modules apps;
    - improvement:
        - ♯568 LDAP config do not require credentials when server allows anon search;
        - ♯349 add placeholders to tip user on jdbc,ldap and others TENANTS configs;
    - features
        - ♯560 select/unselect all tenants on deploy;
        - logs link in running app card;
        - ♯569 multi-providers per tenant (BREAKING CHANGE);

    - ♯453 foundation do not provide tenant database credentials on environment variable to containers anymore. apps should use accounts internal api to recover it;
    - ♯293 sensible data interceptor;

- foundation-licenses
    - ♯492 module api created for license control
    - ♯574 add license at tenant setup
- docs
    - asciinema support
    - upgrading
    - migration tips
    - modules
        - logs: details
        - logs: Redirection logs to external services AKA AWS CloudWatch
    - developers
        - ♯404 Synchro recommendations and development patterns linked
        - Timezone issues
        - Dump timezone
        - using WSL2 on windows
    - troubleshooting
        - foundation-logs: unable to connect logs
        - Tenant screen error when trying to open tenant edit screen
        - failed to upload the application: undefined
        - rebooting the server
        - Unexpected kernel message
        - TrustStore error when uploading - Unrecoverable private key

20.11.22

- foundation core
    - Improvement #495 added docker proxy http interface for docker swarm provider
    - docker socket proxy added for security reasons
    - docker upgrade to 19.03.9

- commandline:
  - BugFixes
    - ♯501 Foundation trustore dont delete the old jks when new one is uploaded
    - ♯547 Foundation don't start multitenancy in command line
    - ♯496 Intermitent foundation start fail 404 on pulling images
    - ♯509 not loading config file
    - ♯506 foundation stop not working
    - bug when start is too slow and expires session token
    - ♯487 removed load spinner cursor buggy for putty terminals;
    - ♯503 foundation login bug as non-root user
    - Issue ♯313 autofix some issues on start and tries to auto-update foundation modules to current version
  - only decrypts and print config when provided a file with read access
  - foundation config test docker registry connectivity
  - Code Review and DRY
  - detect/link old storage path (20.06.12)
  - foundation check in config time
  - logs using json format
  - better log tracing control
  - experimental web control interface
  - Kubernetes remote control improooved
  - Foundation config allows swarm init using interfaces with multiple addresses
  - Require Valid Session now prompt for login
  - implements #384 fakeroot detection
  - local server detection improooved
  - cmd...session.go: ask user to change admin password on login;
  - cmd/../session.go: Cookie renamed to jwt;
  - cmd/../proxy/spec.go: no more resource config in docker socket. its by default;
  - cmd/../supervisor/spec.go: no more mounting foundation.settings ask supervisor when need;
  - `foundation config`
    - each user can have their own profile;
    - use `FOUNDATION_PROFILE` environment to detect/select current profile

    - cmd/config: removed jq dependecy;
    - cmd/config: refactor to support multiple cloud providers;
    - cmd/config: improved oci support, create cluster, policies, add user to policies, select region, compartment and more;
    - implements #456 custom docker registry - now allows to customize remote docker registry server;
    - registry check
    - Implement #424 - Auto load images without ask user;
    - ocir support
        - implemented #470 - implement oci loadbalacer selection
        - OCI types
        - create cluster oci
        - oci kubernetes version update
        - oci checkuser
        - checkOci refact
        - askForOciCompartment
        - Policies completes
        - getCurrentUser
        - fix a bunch of code to keep closer to automate oci configuration.
        - and various more oci config support
    - node pool creating
- `foundation start`
    - traefik uses now a tcp proxy for docker.sock
    - storage and supervisor now uses builtin engine code to start
    - now start modules in parallel;
    - reuse code from engine to start proxy;
    - moved getmissingpropertiess to config
    - reuse code from engine to start proxy
- `foundation module start`
    - cmd..create.go: module.Replicas fix when replicas is zero
- health check from users perspective (endpoints exposed in reverse proxy);
- added health check for storage;
- added health check for supervisor;

    - command help update;
    - `foundation check` Implement #428 - added `/proc/sys/net/ipv4/ip_forward` to foundation check;
    - Better messages to user;
    - Remove recover from main, allowing stacktrace in nil pointer cases
    - reuse code from commons removing service/types.go
    - create.go: reuse code from commons
- commons
    - resource: added replicas and health check support
    - session:
        - better logs and constants for errors
        - cookie to jwt refactor, log fix, **connection leak fixed**
    - utils: get/set current profile for foundation
    - refactor to support multiple cloud providers.
    - log.go: trace in panic fixed.
    - input: no more logging secret fields
- Proxy:
    - improvement #368 traefik release upgrade to latest(2.3.1);
    - kubernetes and swarm builds merged with arguments;
    - for security reasons, added http proxy to `/var/lib/docker.sock`. No need to bind it anymore;
    - Not a privileged container anymore
    - container do not need to be privileged anymore;
    - spec.go: redefined to foundation module pattern
- Supervisor:
    - BugFix: #343 fix wrong memory use for read chunck on uploading modules
    - Bugfix #401 related - jwt token fix;
    - security.go: added validation for forward auth
    - supervisor/spec.go: reuse commons code
    - create secure endpoint to get remote settings;
    - detection of local server improved;
    - added health check for supervisor;
    - Implements #393 now getting `FOUNDATION_SECURITY_KEY` in a secure way;
    - log verbosity increased;

    - modules now can have a tar file with static resources;
    - new endpoint for upload files in static http server using S3 protocol(experimental);
    - new log api endpoint;
    - new generic Logger that supports multiple output including websocket streamimg;
    - Foundation context object to allow functional programming concepts;
    - GetModuleStatus now optionally look for version;
    - LatestRelese detection improved;
    - storage client with https support;
- View:
    - check roles on groups too;
    - Direct link to application logs;
    - fix module deployed card status;
    - fix permission check in null roles;
- Storage:
    - MinIO source release update to latest before default cryptography activation(`RELEASE.2019-10-12T01-39-57Z`);
    - storage/spec.go reuse commons code;
    - Container memory limit changed to 30MB;
    - not a privileged container anymore;
    - added health check for storage;
    - Activated MinIO Dashboard interface to allow direct management of buckets and files.
- Engine:
    - Remove sensitive information from logs.
    - not a privileged container anymore
    - disabled custom route ports to avoid module port conflicts
    - engine/swarm: if engine and supervisor, custom mounts
    - create.go: added replica support
    - update #368 Traefik refactor for new traefik release
    - Error message more detailed;
    - Added support for bind mode mount.PropagationRPrivate
    - using foundationcontext to log and detect settings like orchestrator
    - trust in module size
    - reuse code spec from commons

- accounts
  - BugFix: #488 Http 500 on invalid tenant
  - BugFix: #493 Http 500 on incorrect password or user
  - BugFix: #512 Fallback to LDAP error
  - BugFix: #483 Tenant ID should be uppercase
  - Add LDAP config detail
- Certificates:
  - Golang upgrade to 1.13
  - Roles Handlers improved to support `FOUNDATION_ADMIN` and `FOUNDATION_CERTIFICATE` roles
- Documentation:
  - index: added architectural overview images;
  - Requirements:
    - added inbount traffic;
    - added `ip_forward`;
  - migration:
    - added tip to user save tenant data from 1.3.4 before run migration scripts
  - Uninstall page
    - Backup warning added;
  - New page with issue reporting instructions;
  - For developers:
    - CI instructions;
    - Environment instructions;
    - Multi-tenancy instructions;
    - Tips page added;
    - LDAP test resources;
    - Timezone tips;
  - troubleshooting:
    - added "testing docker installation";
    - added "Docker swarm does not detect node";
    - added "Can't remove `/foundation/images` folder";
    - added "kernel panic";
    - added "Some foundation modules does not work";

        - added "LDAP Problems";
        - added "User autenticate but application gives a error";
        - added "All seems ok, but database connection fail";
        - added "filesystem space usage at 100%";
        - added "foundation-accounts not ready";
        - added "Foundation commandline timeout at all commands"
        - added "cgroup: cannot allocate memory"
        - added TLS instructions
        - added DNS checking
        - added service replicas unstable
        - added couldwatch howto
        - added certificate issue using xfs without dtype
        - added path rellocation instructions
- Module spec:
    - Add Healthcheck config
    - Add Replicas config
    - removed duplicated struct
    - Optional strip prefix
- Internal:
    - build ci registry login with gitlab vars;
    - Remove deprecated code;
    - CI Pipeline improvements;
    - Docker Registry is now generic;
    - Code Review, refactor, DRY;
    - Enabled new golang module versioning system;
    - oci,kubectl:
        - files for interact with oci and kubectl
        - update k8s scripts adding rabbit and jitsi
        - merge kubeconfig file
    - tests using qemu
    - scripts to detect tenant config and reinf database kit version on k8s

20.06.12

- BugFix: Certificates group permission fix
- BugFix: View - pagination color changed
- BugFix: View - Fix breadcrumb tenants
- BugFix: Unprotected URL Forward
- Feature: Cloud migration scripts
- Improvement: tip on oracle linux not starting containers
- Improvement: better logs on proxy and unprotected url forward fix
- Documentation update - RedHat subscription expired
- Documentation update - troubleshooting, foundation start fail
- Documentation update - Suse tips
- k8s scripts update
- CI improvements

20.05.04

- Improvement: OCI Object Storage should be false by default (s3 https off)
- Improvement: Add support user to foundation internal users
- Improvement: #461 User `support` created. Modules foundation-accounts, foundation-certificates and foundation-view affected.

20.04.01

- BugFix: #453 apps using LDAP does not receive database configuration data (accounts)
- Improvement: OCI migration script
- Supervisor: Find sulfix DB to tenant when using LDAP

20.03.20

- BugFix: #453 apps using LDAP does not receive database configuration data
- Documentation: Troubleshooting improved
- foundation-accounts jvm memory config tuning

20.03.12

- Improvement #439: Tenants now are case insensitive
- BugFix: #451 foundation-accounts creating multiple database connection pool for same tenancy
- BugFix: #449 Fix edit tenant are saving obfuscate password
- BugFix: #450 Ldap tenancy login dont set tenantid
- Documentation update: add `/tmp` space requirements and troubleshooting tips

20.03.10

- BugFix: #304 When connection config changes foundation-accounts do not auto-update
- BugFix: #433 when there is a module without releases foundation start partially
- BugFix: #438 Auto detect tenance by subdomain hostname in synchro.com.br
- BugFix: #442 Foundation certificates does not send tenantid when wrong password was passed and tenant select does not has search field
- BugFix: #443 Remove symbols from subdomain at accounts login page
- BugFix: #446 LDAP auth connection error
- BugFix: #448 Documentation - fix migration script download link
- Improvement: #444: Order application card modules alphabetically
- Feature: #447 Add support for https on minio(necessary for OCI Object Storage Compatibility tool)
- Documentation update: install and release-notes

20.01.31

- BugFix: #363 Deploy via commandline don't ask for tenants
- BugFix: #432 Foundation don't remove module.
- BugFix: #434 Fix async module start creating channels
- Better debug logs

20.01.27

- Docker upgrade to 19.03.5
- Better debug logs

- BugFix: #426 Foundation don't create deployed files at startup
- BugFix: #427 Foundation dont start third part modules
- BugFix: #431 SolFis authentication provider should support encript and raw passwords

20.01.17

- BugFix: #415 - In the first config, if we don't have ~/.foundation-session file, config fails to validate current ServerAddress
- BugFix: #416 - On a clean install from full-rpm, the first start don't start non-core critical apps (engine, postgres, accounts, view)
- BugFix: #418 - foundation check generate nil pointer when not authenticated

20.01.15

```
- BugFix:
    - Fix #385 Default rofile settings name = None
    - Fix #387 Foundation does not load other foundation module at start
    - Fix #403 Command line dont request manual parameters
    - Fix #408 Foundation config does not work for hosts without port 22 opened
    - Fix #410 On first config foundation does not crea the '/etc/foundation' directory
    - Fix #411 Autocomplete verbose
    - Fix #401 Supervisor endpoint `security/admin/enabled` return false for admin user
```

20.01.01

- GetModules trust local session and does not ask for password on foundation start
- Add init sequence control to important foundation modules engine, postgres, accounts
- Core init sequence changed from proxy,supervisor,storage to proxy,storage,supervisor.
- Added orchestrator to config validators on start.
- Remove unnecessary startLocalSession Wrapper on supervisor
- Added a friendly message when foundation already stopped.
- Closes #381
- Fix #380
- Docs and version upgrade

19.12.2

- Better warning when application has a wrong type on Size property
- Deploy using web interface load values from previous deploy (last by date), only for empty values in current deploy.
- Continuous integrations improvements
- Log improve in foundation package instruction
- Permission problem fixed on foundation-view:19.12.1 and foundation-storage:19.12.1 image
- Documentation improvement for developers
  - Added manual.pdf file for download/print documentation.
  - Added newrelic page

19.06.1(2.0) - 2019-08-14 (BREAKING CHANGE)

- Change version control system to year.month.build
- Memory adjustments
- Login interface refactor
- Single tenant deprecated
- Documentation upgrade
- Jarvis refactor to new view module
- Accounts refactor
- Allow remove older versions in view module
- Do not load passwords in view module
- Modularize message systems to allow any MQ (ActiveMQ, RabbitMQ, ZeroMQ, etc)
- Service init refactor
- foundation client refactor
- foundation export release
- foundation refactor disk free feature
- foundation-logs interface to access logs in browser
- Certificates module
- Multitenancy support refactor
- Timezone update for uptime report
- Documentation update

1.3.4 - 2018-05-23

- Add Monitoring Support in Foundation Daemon (#197)

1.3.3 - 2018-06-19

- Add support to customize JVM arguments for java based modules (#211)

1.3.2 - 2018-03-09

- Add Cipher Suite and minimum TLS Configuration (#185)
- Change services to allow running as a non root (#138)
- Create the following new commands: service inspect, node inspect, container inspect, network inspect and image inspect (#170)
- Fix LDAP config to support a more restricted filter before auth (#186)
- Improve apps configuration form (#161)

1.3.1 - 2018-02-09

- Add HTTPS support (#156)
- Create command to extract environment data (#158)
- Create error logs for commands (#157)

1.3.0 - 2018-01-08

- Add multitenancy support for apps (#41)
- Add support for multiple network interfaces (#48)
- Add support for different range of IPS (#38)
- Add support for adding users (#34)
- Add validation for compatibility during apps release (#32)
- Create connection test via UI (#33)
- Create deb packaging (Ubuntu Trusty and Artful) (#40)
- Create user management via UI (#40)
- Change main proxy engine (#35)
- Fix deployed services inconsistency
- Fix network deletion (#39)
- Fix lots of bugs (#44, #45 and #47)
- Improve command line interface (foundation)

1.2.4 - 2017-11-07

- Disable integration with upstream server and image registry

1.2.3 - 2017-11-03

- Add validation of requirements (#98 and #112)
- Add command to list server info (#111)
- Add port support on foundation-conf (#113)

1.2.2 - 2017-10-30

- Add support for backend-only apps (#105)
- Add security params to apps's modules (#107)

1.2.1 - 2017-10-26

- Add commands to start/stop container engine (foundation system start/stop)
- Fix storage-driver: device -> mapper-overlay

1.2.0 - 2017-10-17

- Add configuration management (#40)
- Add support for updating apps (#24)
- Add support for Foundation's directory (#60, #94)
- Add requirements validation (#63)
- Create new UI (#31 and #96)
- Fix linkerd's routes management (#73)
- Fix lots of bugs in accounts (#79, #80, #82, and #95)
- Fix lots of bugs in hallofjustice (#56, #87, #99 and #104)
- Update project docs (#97)

1.1.0 - 2017-09-19

- Add foundation to server initialization
- Add logs visualization through Foundation

- Add apps's log visualization through Foundation
- Create auth service to support Solução Fiscal
- Create offline deploy through Foundation

1.0.2 - 2017-09-04

- Create new features for managing logs, services, images and instances

1.0.1-RC1 - 2017-09-01

- Add support to nano instance (#32)
- Add confirmation dialog before removing deploy
- Fix foundation being created through a stack deploy
- Re-validates the feature of CPU reservation (#57)

Troubleshooting

Fail to migrate path

```
ERRO[0000] ERROR: Fail to migrate to new application deployed path: GET1 http://<ip>:80/supervisor/api/apps/deployed Code: 500 {"success":false,"result":"Could not migrate to new application deployed path","details":{"Error":"The specified bucket does not exist"},"content":null,"version":""}
GET1 http://<ip>:80/supervisor/api/apps/deployed Code: 500 {"success":false,"result":"Could not migrate to new application deployed path","details":{"Error":"The specified bucket does not exist"},"content":null,"version":""}
```

Procedure:

> **Warning**
>
> Replace `${foundation_path}` to Volume location path, check:
>
> ```
> foundation config --print --profile-file /etc/foundation/default.settings
> ```

```
mkdir -p /${foundation_path}/system/default/foundation/storage/foundation
```

Install k3s Offline

If the servers there is no Internet connection.

Procedure:

1. Download files:

   ```
   wget https://foundation.synchro.com.br/Install_Foundation_off-line.zip
   ```

2. Unzip files:

   ```
   unzip Install_Foundation_off-line.zip
   ```

3. enter a directory:

   ```
   cd Install_Foundation_off-line
   ```

4. Open `readme.txt` file and follow the steps:

```
cat README.txt
```

Error to load image

```
ERRO[0300] Fail load base image: supervisor.tar.gz
ERRO[0300] Fail load base image: proxy.tar.gz
ERRO[0300] Fail load base image: storage.tar.gz
...
```

Check your filesystem type with `df -Th`.

If you are using `xfs` use `xfs_info` to see if you have dtype enable `ftype=1`.

Fail to get kubernetes namespaces

```
QUESTION: Kubernetes platform provider[k3s]:
INFO[0002] k3s - Lightweight Kubernetes
INFO[0304]
ERRO[0300] [exit status 127]
...
ERRO[0300] [fail to get kubernetes namespaces exit status 127]
...
```

```
kubernetes(k8s)
INFO[0002] Supported platform provider:
eks - Amazon Elastic Kubernetes Service
oke - Oracle Kubernetes Engine
k3s - Lightweight Kubernetes

QUESTION: Kubernetes platform provider[k3s]:
INFO[0002] k3s - Lightweight Kubernetes
INFO[0304]
ERRO[0304] [exit status 127]
INFO[0304] Starting basic requirements check...
WARN[0304] Sorry, Foundation is not ready to do remote check yet.
INFO[0304] Listing current Kubernetes Namespaces
ERRO[0304] [Fail to query kubernetes namespaces exit status 127]
WARN[0304] Fail to list kubernetes namespaces
ERRO[0304] [Fail to get kubernetes namespaces exit status 127]
ERRO[0304] [Fail to update config exit status 127]
exit status 127
```

Check if k3s has been instaled:

```
k3s --version
```

Check access requirementes, here.

Foundation modules don't start

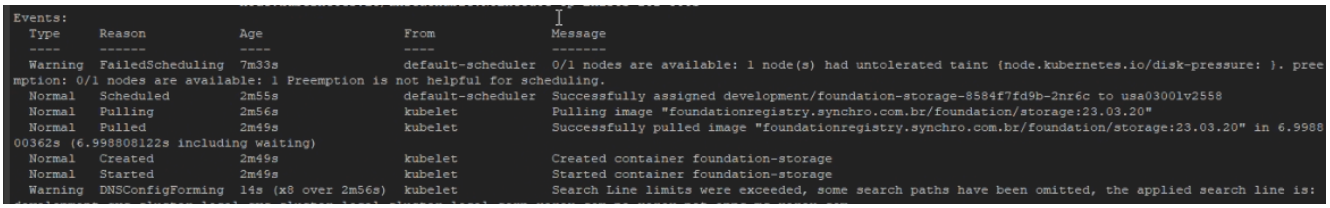
1. Check pods status:

```
kubectl get pods
```

```
NAME                                    READY   STATUS              RESTARTS      AGE
foundation-storage-56f46c6d9d-ljz8n      0/1     ContainerCreating   0             4m45s
foundation-proxy-56958957f8-vg5z9        0/1     ContainerCreating   0             4m45s
foundation-supervisor-56as8957f8-54wsc   0/1     ContainerCreating   0             4m45s
```

2. Describe pod:

```
kubectl describe pod foundation-supervisor-56as8957f8-54wsc
...
... Message
    -------
    0/1 nodes are available: 1 node(s) had untoleraited taint ...
```



3. Check if `/var` have size enough(>4GB):

```
df -Th
```

Procedure:

1. kill all process:

```
k3s-killall.sh
```

2. Stop k3s:

```
systemctl stop k3s
```

3. Check the "Volume Location", by default is `/foundation`. If the value is correct:

```
mv -v /var/lib/kubelet /foundation/kubelet
```

```
ln -sv /foundation/kubelet /var/lib/kubelet
```

```
mv -v /var/lib/rancher /foundation/rancher
```

```
ln -sv /foundation/rancher /var/lib/rancher
```

4. Daemon-reload:

```
systemctl daemon-reload
```

5. Start k3s:

```
systemctl start k3s
```

6. Get your namespace:

```
kubectl get namespace
```

```
NAME              STATUS   AGE
default           Active   34d
kube-system       Active   34d
kube-public       Active   34d
kube-node-lease   Active   34d
development       Active   11d
```

7. Check config context:

```
kubectl config get-contexts
```

```
CURRENT   NAME      CLUSTER   AUTHINFO   NAMESPACE
*         default   default   default
```

If the namespace is empty, execute:

```
sudo kubectl config set-context --namespace development --current
```

```
Context "default" modified.
```

Check if it's ok:

```
kubectl config get-contexts
```

```
CURRENT   NAME      CLUSTER   AUTHINFO   NAMESPACE
*         default   default   default    development
```

8. Delete namespace:

```
kubectl delete namespace development
```

9. Foundation start:

```
foundation start
```

Waiting until foundation core be ready(9)...

1. Open new terminal
2. Check pods:

```
kubectl get pods
```

```
NAME                              READY   STATUS    RESTARTS     AGE
foundation-proxy-864548cbc9-jvx5w       0/1     Running   0          4m13s
[...]
```

3. Check if firewalld is inactive:

```
systemctl status firewalld
```

If is Active, disable:

```
systemctl stop firewalld && systemctl disable firewalld
```

If you prefer, create a rule at your firewall instead of disable it: Check k3s docs.

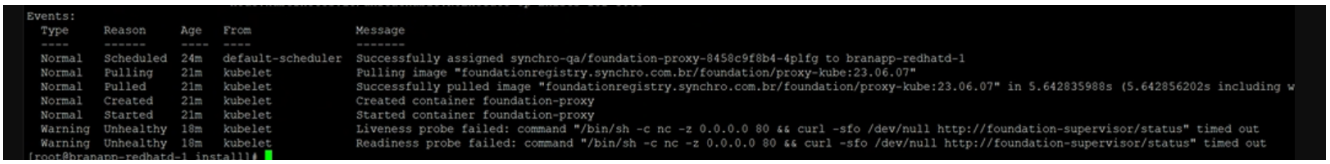
After stop firewall or create rule exception, restart k3s service it was installed:

```
systemctl restart k3s
```

4. Describe proxy pod:

```
kubectl describe pod -l module=proxy
```

Liveness and readiness probe failed:



4. Disable iptables:

```
iptables -F
```

```
iptables -t nat -L
```

Cannot allocate memory

> **Warning**
>
> This commands was running on linux kernel: 3.10.0-1160...

1. Check pods status:

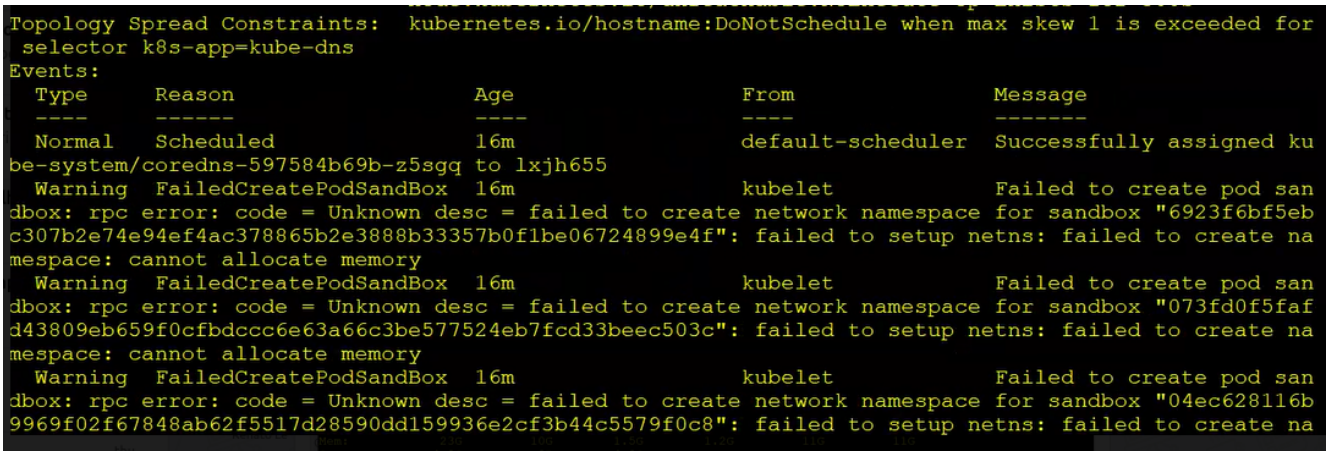
```
kubectl get pods
```

```
NAME                                    READY   STATUS              RESTARTS      AGE
foundation-storage-56f46c6d9d-ljz8n      0/1     ContainerCreating   0             4m45s
foundation-proxy-56958957f8-vg5z9        0/1     ContainerCreating   0             4m45s
foundation-supervisor-56as8957f8-54wsc   0/1     ContainerCreating   0             4m45s
```

2. Describe pod:

```
kubectl describe pod foundation-supervisor-56as8957f8-54wsc
```

```
...
... Message
    -------
    cannot allocate memory ...
```



3. Check requirements.
4. Stop k3s:

```
k3s-killall.sh
```

5. Check free memory:

```
free -h
```

6. Release Linux Memory Cache:

   - To free pagecache:

   ```
   echo 1 > /proc/sys/vm/drop_caches
   ```

   - To free dentries and inodes: ```bash echo 2 > /proc/sys/vm/drop_caches

   ```
   - To free pagecache, dentries and inodes:
    ```bash
   echo 3 > /proc/sys/vm/drop_caches
   ```

7. Check free memory again:

```
free -h
```

8. Check cgtop:

```
systemd-cgtop
```

IMPORTANT: If there is locked memory from cgroups, you must reboot the server.

Cannot restart a linux server?

8.1. List slices with memory:
```

```
systemd-cgtop | docker
```

8.2. List slices only:

```
systemd-cgls | grep docker
```

8.3. To remove slices:

```
umount /sys/fs/cgroup/<subsystem>/<slice_name>
```

```
rmdir /sys/fs/cgroup/<subsystem>/<slice_name>
```

Kubectl config set-context

Check context:

```
kubectl config get-contexts
```

```
CURRENT   NAME      CLUSTER   AUTHINFO   NAMESPACE
*         default   default   default
```

If the namespace is empty, execute:

1. Get your namespace:

```
kubectl get namespace
```

```
NAME              STATUS   AGE
default           Active   34d
kube-system       Active   34d
kube-public       Active   34d
kube-node-lease   Active   34d
producao          Active   11d
```

2. Input your namespace in config context:

```
sudo kubectl config set-context --namespace producao --current
```

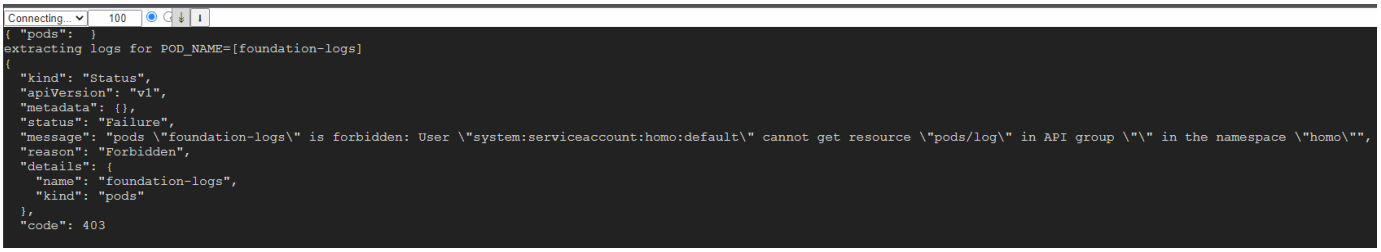
```
Context "default" modified.
```

3. Check if it's ok:

```
kubectl config get-contexts
```

```
CURRENT   NAME      CLUSTER   AUTHINFO   NAMESPACE
*         default   default   default    producao
```

Logs is Forbidden



Enable permission to foundation-logs:

1. Check serviceAccount:

```
kubectl get deploy foundation-logs -o jsonpath="{.spec.template.spec.serviceAccount}"
```

```
default%
```

2. Alter serviceAccount to foundation-engine:

```
kubectl set serviceaccount deployment foundation-logs foundation-engine
```

```
deployment.apps/foundation-logs serviceaccount updated
```

3. Check your kubectl namespace:

```
kubectl get namespace
```

```
NAME              STATUS   AGE
default           Active   34d
kube-system       Active   34d
kube-public       Active   34d
kube-node-lease   Active   34d
producao          Active   11d
```

4. Check current permission from foundation-engine userAccount:

> **${namespace} field**
>
> Replace the field ${namespace} with your kubectl namespace

```
kubectl get clusterrole ${namespace}-foundation-engine -o jsonpath="{.rules[0].resources}"
```

```
["services","endpoints","secrets","serviceaccounts"]%
```

5. Update required roles, adding pods permission in "pods/log":

> **${namespace} field**
>
> Replace the field ${namespace} with your kubectl namespace

```
kubectl patch clusterrole ${namespace}-foundation-engine --type='json' --patch='[{"op": "add", "path": "/rules/0/resources/-", "value": "pods"}, {"op": "add", "path": "/rules/0/resources/-", "value": "pods/log"}]'
```

```
clusterrole.rbac.authorization.k8s.io/tixa-foundation-engine patched
```

Cannot access services on SLES 12.1

On SLES 12.1 Foundation cannot be accessed on port 80. This is due to the absence of IPVS module, which is responsible for load balancing access to services.

In order to fix this, we need to load `ip_vs` kernel module.

Fail to update config ip_forward

Enable IP Forwarding

check ip_forward

```
cat /proc/sys/net/ipv4/ip_forward
```

should be 1, to change it:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Interactive way:

```
yast
```

Go to System -> Network Settings -> Routing Check enable IP Forwarding checkbox.

```
reboot
```

A persistent way is by using `sysctl`

```
# As root...
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward=0

# If it is disabled, re-enable it in the running configuration first:

# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1


# Reload the sysctl.conf file and check the value again. If it is disabled again, edit the /etc/sysctl.conf file and update the value to 1 in the file.

# sysctl -p /etc/sysctl.conf
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
```

Check Requirements

You could use the commands bellow to check if an environment meets the requirements to run Foundation.

RAM

```
$ free -m
              total        used        free      shared  buff/cache   available
Mem:           7915        2534        2556         475        2825        4491
Swap:          7935           0        7935
```

You must check that 'Total Mem' is ~8000. Its nice to check 'Free Mem' as well.

Disk Space

```
$ df -h

Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/fedora-root   49G   44G  2.8G  95% /
tmpfs                   3.9G  172K  3.9G   1% /tmp
```

```
/dev/sda1              477M  190M  258M  43% /boot
/dev/mapper/fedora-home  163G   98G   57G  64% /home
```

You must check 'Available Disk Space' on the partition Foundation installed into. For example, if `volumes.images` and `volumes.system` of `/etc/foundation/foundation-conf.yaml` are point to `/foundation`, you must have 30GB of available disk space in the root partition.

> **Attention**
>
> Some installations are not based on the root partition. So be aware of the configured volume paths of `/etc/foundation/foundation-conf.yaml` while checking available disk space.

The command `lsblk` also may help on troubleshooting disk space issues. It lists all partitions alongside its size and mount point.

```
$ lsblk

NAME          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda             8:0    0 223.6G  0 disk
├─sda1          8:1    0   500M  0 part /boot
└─sda2          8:2    0 223.1G  0 part
  ├─fedora-root 253:0    0    50G  0 lvm  /
  ├─fedora-swap 253:1    0   7.8G  0 lvm  [SWAP]
  └─fedora-home 253:2    0 165.3G  0 lvm  /home
sdb             8:16   1   3.7G  0 disk
├─sdb1          8:17   1   1.5G  0 part /run/media/pvf/Fedora-WS-Live-26-1-5
├─sdb2          8:18   1   6.4M  0 part
└─sdb3          8:19   1  13.7M  0 part
sr0            11:0    1  1024M  0 rom
```

Device Mapper

Can't set cookie `dm_task_set_cookie`

```
devmapper: Error activating devmapper device for
'6bf91878789809febd403ed5e87f715f4c9a2d3b7f257e90fbe3d34dd0f8e816-init': devicemapper: Can't set
cookie dm_task_set_cookie failed
```

Action:

```
sudo dmsetup udevcomplete_all
```

Mount point problems

Check your filesystem type with `df -T` if you are using `xfs` use `xfs_info` to see if you have dtype enable `ftype=1`. If you are using `ext4` ensure you have `shared` flag on `/etc/fstab`. You can test `mount --make-shared /foundation` where `/foundation` is the path where your foundation disk are mounted.

Oracle Linux not starting containers

```
$ semanage permissive -a container_runtime_t
```

this issue can occour in other distro where there is no `semanage` command installed, in this case you may receive the output:

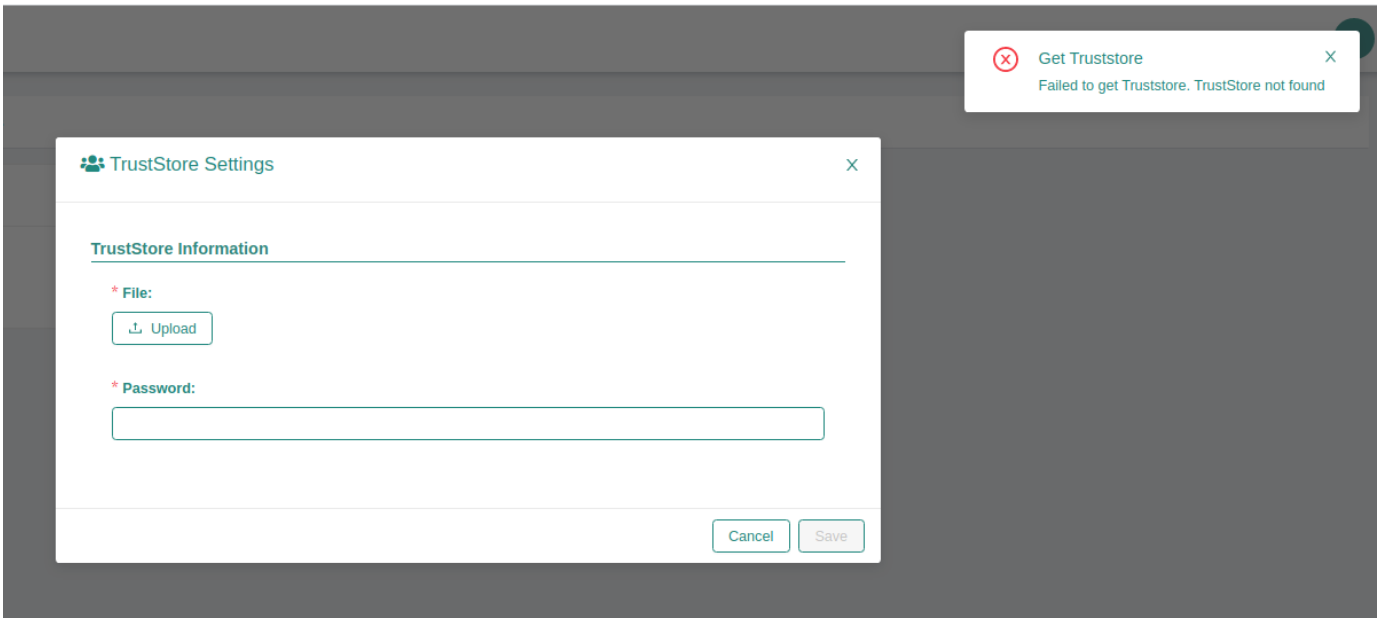
```
Command 'semanage' not found, but can be installed with:
```

Then you should install the `policycoreutils-python-utils` package. The package name may change for each distro, but the install will be something like:

```
$ sudo apt install policycoreutils-python-utils
```

Foundation certificates upload file successfull but don't detail it

If you are trying to upload an certificate and the error below happens after a successfull certificate uploaded:



1. Verify if the file are uploaded sucessfully:

```
ls /{FOUNDATION_PATH}/system/default/storage/foundation/default/truststore

or

ls /{FOUNDATION_PATH}/system/default/storage/foundation/default/keystore
```

1. Verify foundation certificates logs and check if the output is like the below.

```
$ kubectl logs -f service/foundation-certificates

foundation-certificates...| 2020/08/25 15:14:29 --- Listing objects ---
foundation-certificates...| 2020/08/25 15:14:29 Prefix: truststore
foundation-certificates...| 2020/08/25 15:14:29 --- Uploading object ---
foundation-certificates...| 2020/08/25 15:14:29 Bucket name: foundation
foundation-certificates...| 2020/08/25 15:14:29 Object name: default/truststore/file-file-synchro-
Br@zil2010#.jks
foundation-certificates...| 2020/08/25 15:14:29 Bucket (foundation) already exists, skipping it.
foundation-certificates...| 2020/08/25 15:14:29 --- Uploading object ---
foundation-certificates...| 2020/08/25 15:14:29 Bucket name: foundation
foundation-certificates...| 2020/08/25 15:14:29 Object name: default/truststore/password
foundation-certificates...| 2020/08/25 15:14:29 Bucket (foundation) already exists, skipping it.
foundation-certificates...| 2020/08/25 15:05:59 --- Listing objects ---
foundation-certificates...| 2020/08/25 15:05:59 Prefix: default/truststore/file-
foundation-certificates...| Error: TrustStore not found
```

1. Check Foundation filesystem type with `df -T` if you are using `xfs` use `xfs_info` to see if you have dtype enable `ftype=1`. If you are using `ext4` ensure you have shared flag on `/etc/fstab`. You can test `mount --make-shared /foundation` where `/foundation` is the path where your foundation disk are mounted.

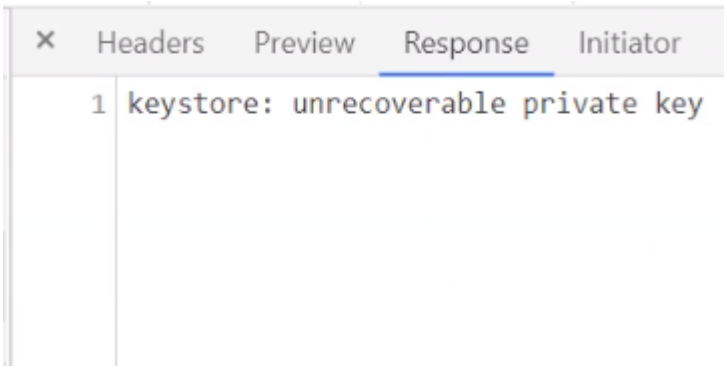
High CPU usage by gvfs-udisks2-vo

```
systemctl stop --user gvfs-udisks2-volume-monitor
```

https://github.com/ubuntu/microk8s/issues/500

TrustStore error when uploading - Unrecoverable private key

You can check de devtools console in the browser, if you see this error:



The file is probably in another unsupported format like PKCS12.

The only supported format for keystore is JKS. Sometimes users just rename a PKCS12 keystore to `.jks` file and this is not enough for a proper conversion.

To check the current format of the file you can use:

```
keytool -list -keystore nome.jks
```

If jks file is different of JSK (PKCS12 is a common error here) we need to convert to the right one, using the command below:

```
keytool -importkeystore -srckeystore ./current-file.jks -srcstoretype pkcs12 -destkeystore ./new-file.jks -deststoretype jks
```

failed to upload the application: undefined

When trying to upload any `file.module` by Web interface and the error message is:

`failed to upload the application: undefined`

It is necessary to add a new module by command line:

```
$ foundation module add path/to/file.module
```

Unexpected kernel message

```
Message from syslogd@<hostname> at <date time> ...
 kernel:unregister_netdevice: waiting for lo to become free. Usage count = 1
```

This is a already fixed Linux Kernel bug reported until kernel 4.19.30. check this thread: https://github.com/moby/moby/issues/5618

Invalid Signature Issue

When using kubernetes as container orchestrator, if you have multiple terminals accessing your foundation cluster, it's a common issue to have a wrong `securityKey` on your settings file.

You can see a problem like this when trying to login via foundation command line:

```
$ foundation login your-environment
INFO[0001] Foundation URL: http://your-environment
QUESTION: Login: user.name
QUESTION: Password:
ERRO[0007] [signature is invalid]
```

To fix it you can use the command to recover the current valid `securityKey`:

```
kubectl get secrets foundation -o json | jq .data.config -r | base64 -d | jq .securityKey -r
```

output

```
vSsdfewerrSKX8H7xsdfasdfWr86qtp
```

Then use the recovered key to update your local settings file with:

```
foundation config --SetSecurityKey vSsdfewerrSKX8H7xsdfasdfWr86qtp
```

output

```
INFO[0001] Saving settings to /home/ggs/.foundation/etc/foundation/your-environment.settings
```

Clean foundation deployed history for old apps releases

> **Attention**
>
> The following steps erases the history for all deployed apps. Please backup the following folder before continue: "/{foundation_installation_path}/system/default/foundation/storage/foundation/default/deployments/history"

When using foundation at latests versions maybe you have some troubles with old apps environment variables, to fix this, reproduce the steps below with the problematic application:

Remove and add the same or an newer version for the problematic app:

```
sudo foundation module stop {appName}-{moduleName}

sudo foundation module rm {appName}-{moduleName}

sudo foundation add --path {absolute path for your .module file}
```

Perform foundation clean and start the module again:

```
sudo foundation clean --history

sudo foundation module start {appName}-{moduleName}:{moduleVersion}
```

An error ocurred when trying set state

Sometimes foundation-authlayer maybe start before foundation-postgres, this action cause some issues for executing DB migration scripts.

Check foundation-authlayer logs to see something like that:

```
 [Migrate][ERROR] Could not exec sql migration up: failed to connect to `host=foundation-postgres user=accounts database=accounts`: hostname resolving error (lookup foundation-postgres on 127.0.0.11:53: server misbehaving)
```

If the log above was presented, and the foundation-postgres service is running execute this command:

```
kubectl rollout restart deploy foundation-authlayer
```

502 Bad Gateway at Supervisor status api on k3s

Make sure if you has sudo/root privileges:

```
sudo su -
```

Sometimes foundation-proxy fail the request to validate if foudation-supervisor is ready. To check if supervisor/status api healthcheck is the problem reproduce this steps:

```
kubectl logs pod/$(sudo kubectl get pods | grep proxy | cut -c 1-38 | head -n 1)
```

If the logs shows 502 Bad Gateway error when making a request to 127.0.0.1/supervisor/status maybe you has a firewall problem, to validate it, please run the command below:

```
systemctl stop firewalld
```

If this command solve the problem you need to disable the firewall permanentily, or create a new rule:

```
systemctl disable firewalld
```

After restart k3s service

```
systemctl restart k3s
```

Generating TLS Self Signed Certificate and Key

1. Create the certificate and key:

   ```
   openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out MyCert.crt -keyout MyKey.key
   ```

   You will be prompted to add identifying information about your website or organization to the certificate. Since a self-signed certificate won't be used publicly, this information isn't necessary. If this certificate will be passed on to a certificate authority for signing, the information needs to be as accurate as possible.

   The following is a breakdown of the OpenSSL options used in this command. There are many other options available, but these will create a basic certificate which will be good for a year. For more information, see man openssl in your terminal.

   - newkey rsa:4096: Create a 4096 bit RSA key for use with the certificate. RSA 2048 is the default on more recent versions of OpenSSL but to be sure of the key size, you should specify it during creation.
   - x509: Create a self-signed certificate.
   - sha256: Generate the certificate request using 265-bit SHA (Secure Hash Algorithm).
   - days: Determines the length of time in days that the certificate is being issued for. For a self-signed certificate, this value can be increased as necessary.
   - nodes: Create a certificate that does not require a passphrase. If this option is excluded, you will be required to enter the passphrase in the console each time the application using it is restarted.

2. Restrict the key's permissions with needes, so that only root can access it:

   ```
   chmod 400 MyKey.key
   ```

3. Chipher used for this key is:

```
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
```

Configuring K3S no_proxy to solve pod logs issues.

If you are having errors to get logs from k3s pods, like the error below:

```
"proxyconnect tcp: proxy error from 127.0.0.1:6443 while dialing example.proxy.com.br:80, code 503: 503 Servide Unavailable"
```

You need to set it up the k3s no_proxy configuration.

To do that, is necessary to edit the `k3s.service.env` file, located at `/etc/systemd/system/k3s.service.env`

Obs: The K3s installation script will automatically take the `HTTP_PROXY`, `HTTPS_PROXY` and `NO_PROXY` variables from the current shell, if they are present, and write them to the environment file (`k3s.service.env`) of your systemd service.

To fix that issue you need to add or update your NO_PROXY line at `k3s.service.env` file and your shell variables too if exists, like the example below:

```
NO_PROXY="127.0.0.0/8,10.0.0.0/8,localhost,<YOUR_HOST_IP_ADDRESS>,<YOUR_MACHINE_HOSTNAME>"
```

After that update/reload your systemd configuration and restart k3s service:

```
systemctl daemon-reload
```

```
systemctl restart k3s
```

For more detailed information, please access the k3s official documentation

Keycloak Valid Redirect URIs security configuration

To see detailed information about how to setup `Valid Redirect URIs` at `keycloak` for more security when authenticating in your `foundation` server, go to Keycloak Advanced - Valid Redirect URI section.

Amazon AWS instances disable nm-cloud-setup.service to run k3s.service.

In some cases, AWS Instances/EC2 maybe has nm-cloud-setup.service enabled and running, by default the k3s service only work with nm-cloud-setup.service disabled and stopped.

For this case, we has two options:

1. Disable nm-cloud-setup.service and stop then:

   ```
   systemctl stop nm-cloud-setup.service nm-cloud-setup.timer
   systemctl disable nm-cloud-setup.service nm-cloud-setup.timer
   ```

   Refesh systemctl service files:

   ```
   systemctl daemon-reload
   ```

   Then, restart k3s service:

   ```
   systemctl restart k3s.service
   ```

2. Another option, is remove the `ExecStartPre` check from k3s.service file, located at `/etc/systemd/system/k3s.service`.

   > **Attention**
   >
   > Before perform the next steps, make sure with your infrastructure team if exists any network rules maybe blocks k3s.service communication.

   Remove or comment that line:

   ```
   ExecStartPre=/bin/sh -xc '! /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service'
   ```

   Refesh systemctl service files:

   ```
   systemctl daemon-reload
   ```

   Then, restart k3s service:

   ```
   systemctl restart k3s.service
   ```

Overview

Foundation is a PaaS that lets you run your app without struggling with infrastructure and others non-functional requirements. That means we try to do all non-functional code for you. So, you only have to worry about your code. Foundation will compile it and run it in a "cloud-based" approach, even in an on-prem environment. Not only that, but foundation takes care of requirements and installation details. When using foundation, to deploy, you only need to do a drag and drop.

Currently, foundation give you some choices, suporting docker-swarm or kubernetes(on-premises, eks and oke) under the hood.

Synchro recommendations and development patterns

https://git.synchro.com.br/wat/identidade-tecnologica

Why not use docker directly?

Infrastrcuture is a complex puzzle. You can mount this puzzle in many ways and few are manageable. Docker is a important part of this, but there is so much more. Docker seems simple, but when you try to put a system working on it you have some problems. You need a container orchestator, like docker-swarm or kubernetes. But they are not so easy to configure and maintain. Your orchestrator, and your docker need to run in an infrastructure, on-premises or cloud, and there are so many clouds (aws, ec2, eks, oci, oke, gce, gke ...)

Foundation has a predefined stack to manage it. There are others initiatives like foundation (openstack, rancher, cloudstack and others) but they have too generalist with much resources that turn they more complex than we need.

Foundation tries to be simple and easy.

To make that happen you need to declare the building blocks of your app in a file named `spec.yaml`. Each block needs a docker container image, and the `spec.yaml` contains all the information Foundation needs to know about your app such as its name, modules, services it depends on, environment variables, and so on. Based on this information, Foundation will build a Release and Deploy it across the platform.

How to adapt my application to run in foundation

In your source root, You need 3 files:

- `spec.yaml`
- `Dockerfile`
- `Makefile`

Nomenclature App

Why is it important?

The App Name and Module Name are concatenated to create a service name, and this service entry in internal DNS. With this all services can communicate easily with the others services. In this case, the hyphen(-) is used like a splitter.

It is recommended that no hyphen(-) in the App and Module Name, because that could cause a conflict on the algorithm.

About de dot(.), is prohibited. The dot(.) is a subdomain splitter of DNS entries, and this service breakdown when the Swarm Or Kubernetes Orchestrator create the service. For the same reason to use the simbol or rules that is not permitted in hostnames.

For good working App deploy with Foundation, before you create App properties must follow some rules to separated from it by:

App and module name

- Use: _ (underline)
- Do not use: . (dot)
- Do not use: - (hyphen)

Version

- Use: _ (underline)
- Use: . (dot)
- Do not use: - (hyphen)

spec.yaml

When creating your spec.yaml, please take a time to read the page.

Basically `spec.yaml` need to have this structure.

When done, your spec.yaml will be something like this one.

Dockerfile

There is nothing special about `Dockerfile`. This is a docker script to construct your container image. Please, try to not use root user inside your container, it's not mandatory, but is a good practice.

Makefile

Makefile is a script makes easier to build applications. It's very simple.

For foundation your Makefile needs to have the follow targets: - build: Here you can build your application in your way. We like to build inside a container using docker. This way we don't need to think on developer environment. If the dev has docker it will build. - image: docker save > build/export/ - export:

Here a sample Makefile:

```
#Default env vars
#autodetect version
VERSION:=$(shell bash -c "cat spec.yaml | grep version | cut -d ' ' -f2")
APP:=$(shell bash -c "cat spec.yaml | grep application | cut -d ' ' -f2")
MODULE:=$(shell bash -c "cat spec.yaml | grep name | cut -d ' ' -f2")

.PHONY: help
help: ## show make targets
    @awk 'BEGIN {FS = ":.*?## "} /^[a-zA-Z_-]+:.*?## / {sub("\\\\n",sprintf("\n%22c"," "), $$2);printf "
\033[36m%-20s\033[0m  %s\n", $$1, $$2}' $(MAKEFILE_LIST)

clean: ## remove build files
    rm -Rf build

.PHONY: update-version
update-version: ## update the version of project
    @(cat spec.yaml | awk '{ x=0; }; /^version: .*/ { split($$2, a, "."); bn=a[3]; v="'$$(date +%y.%m.)'" (++bn);
print $$1,v; x=1; }; { if ( x==0 ) print; }' > spec.yaml)

.PHONY: build
build: ## Build this module
    mkdir -p build
    docker run --rm  -v ${CURDIR}:/tmp/app\
                -w /tmp/app\
                foundationregistry.synchro.com.br/dev/jdk11 \
                sh -c "gradle --stacktrace --project-cache-dir /tmp/app/.foundation/gradle/project -g /tmp/
app/.foundation/gradle/global clean bootRepack"

.PHONY: image
image: ## Build the docker image for this module
    docker build -t foundationregistry.synchro.com.br/$(APP)/$(MODULE):$(VERSION) .
```

```
.PHONY: export
export: ## Exports docker images from this module to directory $(OUTPUT)/images
	mkdir -p build/export/images
	docker save foundationregistry.synchro.com.br/$(APP)/$(MODULE):$(VERSION) -o ./build/export/images/$
(APP)-$(MODULE).tar.gz

.PHONY: push
push: ## publish docker image do synchro foundationregistry
	docker push foundationregistry.synchro.com.br/$(APP)/$(MODULE):$(VERSION)

.PHONY: module
module: ## create the $(APP)-$(MODULE)-$(VERSION){,-FULL}.module files in build/export/ folder
	foundation package
```

Creating module file

To deploy applications in foundation, you need to generate a `app-module-version.module` file. This file contains the spec.yaml and optionally the docker image.

To generate just do:

```
foundation package
```

> **Important**
>
> You need the foundation client installed.

Then foundation will do:

- Validate the spec.yaml file
- make build
- make image
- make export
- Generate .module files

What is the .module file?

The `app-module-version.module` file is a `tar.gz` file renamed to `.module`. In fact you can create it by hand with `tar czvf app-module-version.module spec.yaml`.

Foundation create 2 .module files:

- `app-module-version.module`: A small file only with the spec.yaml file;

- `app-module-version-full.module` : A big file, with the spec.yaml and the docker image tar;

On environments with access to foundation registry, you can use the small, then foundation will download the image automatically. On closed environments you may need the big one. Or you can load the image by hand with docker load, but by using the full.module file you don't need it.

Multitenancy

Multitenancy

The term "software multitenancy" refers to a software architecture in which a single instance of software runs on a server and serves multiple tenants. Systems designed in such manner are often called shared (in contrast to dedicated or isolated). More at wikipedia

In practice it's a shared hardware and software resources that multiple clients can use in an isolated way.

So, you have a software deployed serving multiple clients, but they can not see each other data.

Foundation provides a platform to you create your software this way.

In fact, all authenticated software running in foundation is considered multitenant. But despite foundation efforts it's not so transparent. You need to understand the tenant concept and take care of some details inside your code. The foundation team are here ready to help you understand this. Please ask us.

Authentication

Your application do not need to worry about authentication. `foundation-proxy` and `foundation-accounts` will take care about it.

`foundation-proxy` is a core component in foundation. It receives all network trafic from outside of foundation network and routes it to the correct component.

`foundation-proxy` check if the target is protected (`secure: true` in spec.yaml) and forward to `foundation-accounts` if the user session is invalid.

`foundation-accounts` is a foundation module that allow users to login in and create a valid foundation session token.

`foundation-accounts` have providers. Currently it supports Internal, LDAP and SOLFIS. Next releases are planned to support SFW, OAUTH and SAML.

When you install foundation you have no tenants, and need to login using the default in memory user `admin` . Then default password is `admin` and you need to change it with `foundation passwd` .

When you create a tenant, you set the provider type, and `foundation-accounts` stores it in a postgres database, using `foundation-postgres` module.

Datasource management

To help developers we have created a java jar to manage spring datasources, and provide multitenancy support, git it a try: https://git.synchro.com.br/foundation/libs/foundation-multitenant-config

Dev Environment

For a foundation development environment you just need docker and the foundation command line client installed. But if you start foundation only with the client, you need to start all modules by yourself.

Keycloak Setup for Dev Environment

Foundation configurations

- foundation config

https://foundation.synchro.com.br/release/22.02.22.rc/configs/configuration/index.html#keycloak-server-information

- foundation start

configuraçoes externas no keycloak

1. Access permition to REALM: It is given to the master for his user with scsa-dev realm.
2. Enter in keycloak and give access permission to users;

[manager-users, manager-clients, etc...]

1. Access the foundation console to add FOUNDATION_ADMIN, FOUNDATION_CERTIFICATES since they must be registered in realm and/or must be already registered.

configuration on clients

1. Enable service accounts to clientID set in foundation config;
2. Put realm-admin in the scope tab and in service accounts roles;
3. If you have a custom URI that differs 127.0.0.1, add a valid redirect URI at Keycloak, create a role and grant user access to it, as mentioned below:

127.0.0.1 -> clientIdName is configured in foundation config: - if it is foundation-dev, tenantid will be foundationdev - if it is foundation-hmg, tenantid will be foundationhmg

For example, for you to access the foundation-dev url yo need foundation-dev role. aqui tem que ter a role cloud-dev-foundation ou o cloud-hmg-foundation

1. If you have any realm-role that needs to release in client, it is needed to add it in scope and release in realm-role.

Separates on-premises from cloud.

- In cloud, probably hmg and prd will be ready, but in dev environment is necessary the creation by the devs and/or utilize the register made by your team.
- no cloud, provavelmente estará pronto em hmg e prd, no ambiente de dev, é necessário a criação pelos devs e ou utilizar o cadastro realizado pela sua equipe.

spec.yaml

Overview

The `spec.yaml` describes your app to Foundation.

Main Structure

- Module

Others structures

- Config
- Service
- ContainerSize
- Dependency
- HealthCheck
- Mount
- ModuleStatus

Example

- Specification YAML configuration file

Module

Property	Description	Type	Default	Omitempty	V
foundationVersion	FoundationVersion is a data poiting for which foundation version this module was created	string	-	yes	
application	Application, Name and Version merged create a	string	-	no	

Property	Description	Type	Default	Omitempty	V
	unique identifier for this module				
name	Application, Name and Version merged create a unique identifier for this module	string	-	no	
version	Application, Name and Version merged create a unique identifier for this module	string	-	no	
title	Title to give a user friendly application name in interfaces like foundation-view	string	-	yes	
description	Description text to help users understand what this module is for in interfaces like foundation-view	string	-	yes	
hidden	Hidden=true means do not show card for this module in interface like foundation-view	boolean	false	no	2
config	Config groups all environment properties for this module	Config	-	yes	
services	Services configure paths on	Service list	-	yes	

Property	Description	Type	Default	Omitempty	V
	foundation proxy to expose services				
size	Size limit RAM memory of this module (not submodules)	ContainerSize	-	no	
dependencies	Dependencies tells foundation what other modules to look before start this one	Dependency list	-	yes	
replicas	How many instances of this module should be running?	uint16	0	no	
healthcheck	Healthcheck is a shell command that allows foundation to know if the module are stable	HealthCheck	-	no	
privileged	Privileged run this service in privileged mode, this can cause serious security concerns	boolean	-	yes	
roles	Roles allow this software gain access to cluster roles like access secrets or list services	string list	-	yes	

Property	Description	Type	Default	Omitempty	V
mounts	Mounts allows use relative host filesystem paths inside containers Absolute paths are used in privileged mode This can cause scalability issues in clusters. Use with caution	Mount list	-	yes	
status	Status are auto updated by foundation. You can ready it here at runtime	ModuleStatus	-	no	
details	Details reports foundation friendly status message	string	-	yes	
bundle	Bundle=true says this is a pack with submodules	boolean	false	no	
submodules	Submodules used in this pack	Module list	-	yes	
cardRedirectPath	CardRedirectPath is used to redirect a module page	string	-	no	2

Config

Config groups all environment properties

Property	Type	Default	Omitempty	Version
groups	Group list	-	no	
tenants	string list	-	no	21.05.07
dataProviders	string list	-	no	21.05.07
multitenancy	boolean	false	no	

Service

Configure paths on foundation proxy to expose services.

Developers: about service ports

It's extremely important to inform the ports that will be exposed for the correct configuration of the service.

Like example below. Spec.yaml example

Property	Type	Omitempty
port	int	no
route	Route	no
externalPort	int	yes
protocol	string	yes

ContainerSize

Size limit RAM memory

Value	Memory allocation
pico	120 Mbytes

Value	Memory allocation
nano	250 Mbytes
micro	500 Mbytes
small	1 Gbytes
medium	2 Gbytes
large	4 Gbytes
xlarge	8 Gbytes
2xlarge	16 Gbytes
4xlarge	32 Gbytes
8xlarge	64 Gbytes
30	Custom value in Mbytes - 30 Mbytes

Dependency

Defines a dependency

Property	Type	Omitempty
application	string	no
name	string	no
version	string	no

HealthCheck

Defines how to check if the container is working as expected and should receive requests

Property	Type	Omitempty
type	string	no
command	string	no
startDelay	string	no
interval	string	no

Mount

Defines a volume mount property for a service

Property	Type	Omitempty
host	string	no
container	string	no
mode	string	no

ModuleStatus

Defines a module status

Status	Description
Failed	Failed is used when module was unable to start
NotFound	NotFound is used when foundation could not see the service that was supposed to be there
Pending	Pending is used when the module is trying to start but not ready
Running	Running is used when all seems to be fine
Stopped	Stopped is used when the module is not running and this is expected
Unknown	Unknown is used when foundation was unable to detect the service

Example

The listing bellow is a full example of everything is possible to declare in `spec.yaml`.

```yaml
application: reinf
name: core
version: 3.2.1
title: Solfis | REINF
description: Apuração de tributos retidos para a EFD-REINF
hidden: false
config:
  groups:
  - name: System Settings
    properties:
    - name: FOUNDATION_STORAGE_ACCESS_KEY
      annotations: system
    - name: FOUNDATION_STORAGE_SECRET_KEY
      annotations: system
    - name: FOUNDATION_STORAGE_BUCKET
      annotations: system
    - name: FOUNDATION_STORAGE_URL
      annotations: system
  - name: Configuração de cache(REDIS) do Reinf
    properties:
    - name: CACHE_HOST
      description: Host (endereço do cache)
      value: reinf-cache
    - name: CACHE_PORT
      description: Porta de acesso type(int)
      value: "6379"
  - name: Configurações de fila(ACTIVEMQ)
    properties:
    - name: QUEUE_BROKER
      description: URL do Broker (endereço da fila)
      value: tcp://reinf-amqp:61616
    - name: QUEUE_USER
      description: Usuário
      value: admin
  - name: Webservice da RFB
    properties:
    - name: WS_URLTRANSMISSAO
      description: URL do webservice de transmissao da RFB
    - name: WS_SOAPACTIONTRANSMISSAO
      description: SOAP Action de transmissao
    - name: WS_URLCONSULTA
      description: URL do webservice de consulta da RFB
    - name: WS_SOAPACTIONCONSULTA
      description: SOAP Action de consulta
  - name: Proxy
    properties:
    - name: PROXY_HOST
      description: Hostname
      annotations: optional
    - name: PROXY_PORT
      description: Porta
      annotations: optional
```

```
    - name: PROXY_AUTH_USER
      description: Usuário
      annotations: optional
    - name: PROXY_AUTH_PASSWORD
      description: Senha
      annotations: type(password) optional
  - name: Dados de conexão com servidor SMTP
    properties:
    - name: SMTP_HOST
      description: Hostname
      annotations: optional
    - name: SMTP_PORT
      description: Porta
      annotations: optional
    - name: SMTP_TIPOAUTENTICACAO
      description: Tipo de autenticação (SSL/TLS)
      annotations: optional
    - name: SMTP_EMAILREMETENTE
      description: Email utilizado como remetente
      value: admin
      annotations: optional
    - name: SMTP_NOMEREMETENTE
      description: Nome utilizado como remetente
      value: Notificação Reinf
      annotations: optional
    - name: SMTP_AUTH_USER
      description: Usuário
      annotations: optional
    - name: SMTP_AUTH_PASSWORD
      description: Senha
      annotations: type(password) optional
  - name: Tenants
    properties:
    - name: TENANTS
      description: Tenant list
      value: TESTE
  tenants:
  - TESTE
  dataProviders:
  - SOLFIS
  multitenancy: true
size: ""
replicas: 0
healthcheck:
  type: ""
  command: ""
  startDelay: ""
  interval: ""
status: ""
bundle: true
submodules:
- application: ""
  name: web
  title: web module
  version: ""
  hidden: true
  services:
```

```
  - port: 8080
    route:
      path: /reinf
      secure: true
  size: "20"
  replicas: 0
  healthcheck:
    type: ""
    command: ""
    startDelay: ""
    interval: ""
  status: ""
  bundle: false
  cardRedirectPath: ""
- application: ""
  name: backend
  title: backend module
  version: ""
  hidden: true
  services:
  - port: 8080
    route:
      path: /reinf-backend
      secure: true
  size: "4096"
  replicas: 0
  healthcheck:
    type: shell
    command: curl 'http://localhost:8080/api/health-check/foundation'
    startDelay: 2m
    interval: 5s
  status: ""
  bundle: false
  cardRedirectPath: ""
- application: ""
  name: mensageria
  version: ""
  hidden: true
  services:
  - port: 8081
    route:
      path: /reinf-msg
      secure: false
  size: small
  replicas: 0
  healthcheck:
    type: ""
    command: ""
    startDelay: ""
    interval: ""
  status: ""
  bundle: false
  cardRedirectPath: ""
cardRedirectPath: ""
```

Foundation CLI

foundation is a command line interface for interacting with Foundation. It provides commands for release and deploy apps.

Download

Click here to download latest version of Foundation.

Install

Once downloaded copy it to /usr/bin and change its permissions to be executable:

1. Copy it to bin directory:

```
sudo cp foundation /usr/local/bin/foundation
```

2. Make it executable:

```
sudo chmod +x /usr/local/bin/foundation
```

3. Check it:

```
foundation --version
```

Package

Package command will compile your code locally, accordingly to the module definition, and build a release against a Foundation installation. Create your spec.yaml on the application root folder and run foundation package there.

```
foundation package
```

The command above releases an App against production environment. These environments are defined at your `spec.yaml`. See spec.yaml to learn more.

Once you app is packaged your .modules files will be on "build/export" path. After release you can access Foundation's web interface and deploy it.

Internal Tenant

Overview

This page describes internal tenant information for your app.

To get tenant information it is necessary use this URI:

```
http://foundation-authlayer/api/tenant/{tenantId}
```

Main Structure

- Tenant

Others structures

- Provider type
- Oracle
- Variables

Example

- Specification JSON information file

Tenant

Property	Description	Type	Version
id	A unique identifier. Only uppercase letters and numbers, numbers not allowed at first character	string	
description	Tenant description	string	
license_key	The customer will receive the license key by a specific department	string	21.05.12
environment	Definition of server environment	Environment	21.05.12
config			

Property	Description	Type	Version
	Config provider type properties for this tenant	Provider type list	
provider_type	Provider type	string	disused
apps	Application list	string	disused
valid	Valid	Boolean	

Environment

Environment default values

Property	Description	Type
DESENVOLVIMENTO	Ambiente de desenvolvimento	DEV
ACEITE	Ambiente de Aceite	UAT
HOMOLOGACAO	Ambiente de Homologação	QA
PRODUCAO	Ambiente de Produção	PROD

For more information, see enviroment type list.

Provider Type

Config provider type properties

Property	Description	Type	Version
SOLFIS	Solução Fiscal	Oracle	
DFE	Documentos Fiscais Eletrônicos	Oracle	
AGR	Automação de Guias de Pagamento	Oracle	

Property	Description	Type	Version
GESTAOCREDITOS	Gestão de créditos	Oracle	
SFW	Solução Fiscal Web	Oracle	
VARIABLES	Variables	Variables	

Oracle

Oracle configuration properties

Property	Placeholder
URL	jdbc:oracle:thin:@<host>:<port>:<sid>
USER	Database user
PASSWORD	Database password

Variables

Variables configuration properties

Property	Placeholder
Field	Only uppercase key
Value	Any value

Example

The listing bellow is a full example of everything is possible to receive in internal tenant information.

```
{
 "id":"TENANTID",
 "description":"Desenvolvimento",
 "config":{
    "DFE":{
     "URL":"jdbc:oracle:thin:@<host>:<port>:<sid>",
```

```
      "USER":"user_dfe",
      "PASSWORD":"**********"
     },
     "AGR":{
      "URL":"jdbc:oracle:thin:@<host>:<port>:<sid>",
      "USER":"user_agr",
      "PASSWORD":"**********"
     },
     "GESTAOCREDITOS":{
      "URL":"jdbc:oracle:thin:@<host>:<port>:<sid>",
      "USER":"user_gestao",
      "PASSWORD":"**********"
     },
     "SFW":{
      "URL":"jdbc:oracle:thin:@<host>:<port>:<sid>",
      "USER":"user_sfw",
      "PASSWORD":"**********"
     },
     "SOLFIS":{
      "URL":"jdbc:oracle:thin:@<host>:<port>:<sid>",
      "USER":"user_solfis",
      "PASSWORD":"**********"
     },
     "VARIABLES":{
      "ORG_ID":"82",
      "DESCRIPTION":"organization id"
     }
  },
  "apps":[],
  "provider_type":"",
  "license_key":"nolicense",
  "environment":"ACEITE",
  "valid":true
}
```

NewRelic

NewRelic is a tool to monitor, profile and throubleshooting apps. You can use it to discover things, like where my app is wasting time or resources. Also, you can see what endpoint are busy and how much time they use, track it down to what query is wasting time in database, a simple explanation of this query, and much more.

NewRelic supported languages[https://docs.newrelic.com/docs/agents/manage-apm-agents/installation/install-agent]:

```
C
Go
Java
.NET
Node.js
PHP
Python
Ruby
```

To activate newrelic in your app you need to install the agent: https://docs.newrelic.com/docs/agents/java-agent/installation/install-java-agent

And you need to have a config file and a license_key. Ask a foundation dev team member for it. We will support you in this task.

Feature Request and Issues

Reporting Issues

The Foundation is an audacious project that will be success only if all devs collaborate with the foundation dev team. Our devs are open to attend chats, but be advised that we have to focus on bugfixes and developing new features and stay atending chats will slow down our development.

The steps we suggest to do:

Bug Cases:

1. Be sure you are in the latest version of foundation[http://foundation.synchro.com.br]
2. Test and try to reproduce your problem
3. Search the documentation[http://foundation.synchro.com.br]
4. Search the issues in gitlab[https://git.synchro.com.br/foundation/foundation/issues?scope=all&utf8=%E2%9C%93&state=opened]
5. Create a new issue with all details needed to reproduce the problem.

Missing Feature Case:

1. Be sure you are in the latest version of foundation[http://foundation.synchro.com.br]
2. Search the backlog for feature requests in gitlab[https://git.synchro.com.br/foundation/foundation/issues?scope=all&utf8=%E2%9C%93&state=opened]

   a) If you found a feature request, vote up to give us a real importance of this feature and add all details you think important.

   b) If do not found, you can create a new feature request. Please provide all details you can and describe why the requested feature is important.

Force docker to run a specific image.

Docker builds can be tricky. You build a image, remove the running container that is using de previous image, then a new container is created with the new image you created few seconds ago, right ? Well, not exactly. By default, docker will query the image registry repository to define which image to use. If docker has access to the registry, then it will download the "official" image instead of using the one you created. To avoid this behaviour you need to define the image name as imagename:version@imageid. To discover the image id you can use:

```
$ docker image inspect foundationregistry.synchro.com.br/foundation/supervisor:19.12.1 | jq -r .[0].Id
sha256:0928b47c132d68e560fb57b7787ea36dd4d4594fbe18268f560bf52fc95f28c0
```

then you can set as:

```
$ docker service update --image foundationregistry.synchro.com.br/foundation/supervisor:
19.12.2@sha256:0928b47c132d68e560fb57b7787ea36dd4d4594fbe18268f560bf52fc95f28c0 foundation-
supervisor
foundation-supervisor
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
```

This way you can be sure it is running your image.

Link your commit with gitlab issues

Samples: Awesome commit message

Fix #20, Fixes #21 and Closes group/otherproject#22. This commit is also related to #17 and fixes #18, #19.

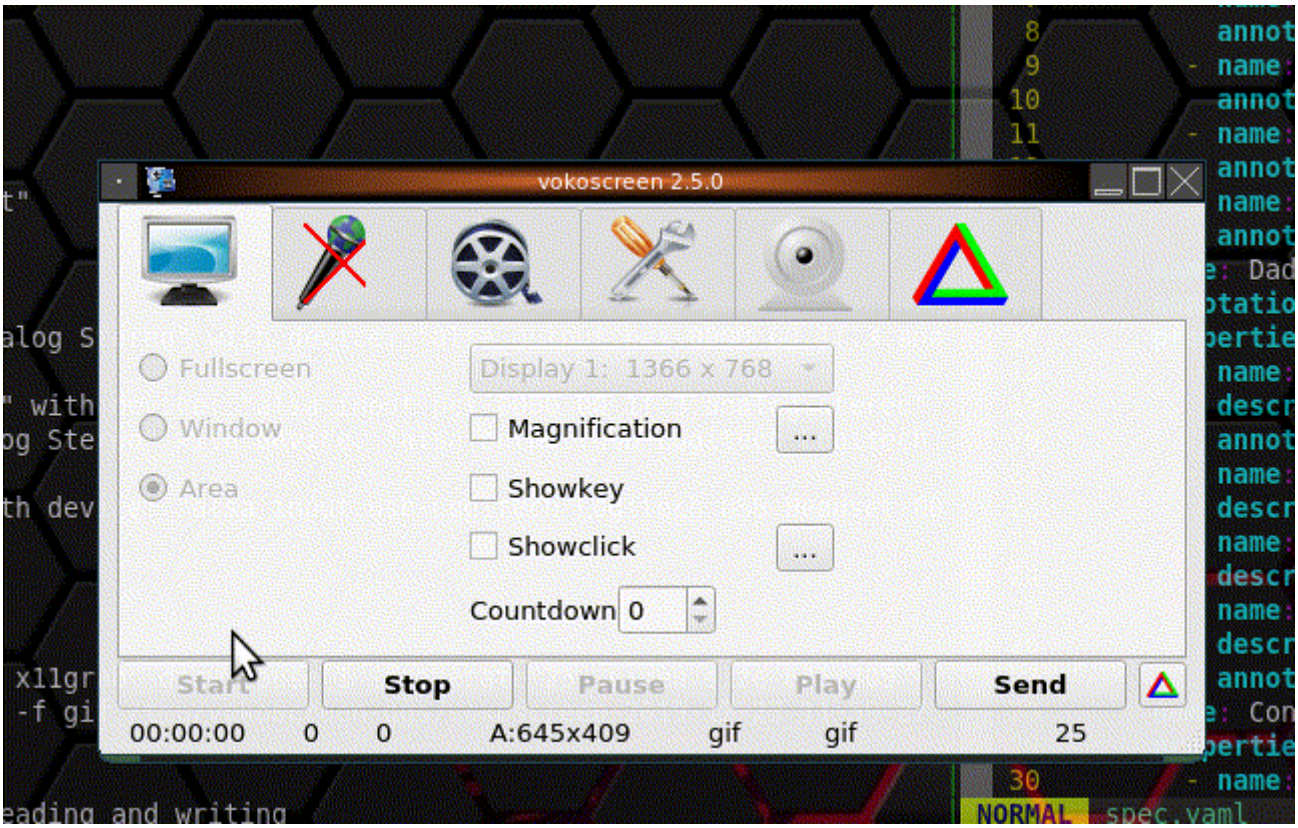
To reference an issue: #123 To reference a MR: !123 To reference a snippet $123

Similarly, even a closed issue can be referenced like Related to #78 and #93

Save a animated screenshot

Sometimes is productive to create an animated gif to demonstrate a feature, bug or an issue.

Vokoscreen(http://linuxecke.volkoh.de/vokoscreen/vokoscreen.html) is an easy way to record your screen and save it to animated gif.



Show your keys for screencast

key-mon(https://code.google.com/archive/p/key-mon/) screenkey(https://www.thregr.org/~wavexx/software/screenkey/)

Subnet reference

Managing class B [RFC1878][https://www.ietf.org/rfc/rfc1878.txt) Subnet Sheet Cheat

Connect database by commandline

You can use sqlline to connect databases using jdbc url

```
/usr/lib/jvm/java-13-openjdk-amd64/bin/java -classpath /home/ggs/src/foundation/apps/accounts/api/libs/ojdbc6-11.2.0.4.jar:/usr/share/java/jline.jar:/usr/share/java/sqlline.jar:/usr/share/java/hsqldb.jar sqlline.SqlLine --silent=true -u jdbc:oracle:thin:@<hostname>:1521:<instance> -n <schema> -p <password> --outputformat=csv
```

```
$ cat /usr/bin/sqlline
#!/bin/sh
```

```
# Include the wrappers utility script
. /usr/lib/java-wrappers/java-wrappers.sh

# We need a java runtime (any 1.4 work)
find_java_runtime all

# Define our classpath
find_jars jline sqlline postgresql-jdbc4 mariadb-java-client hsqldb jtds

JAVA_CLASSPATH="/home/ggs/src/foundation/apps/accounts/api/libs/ojdbc6-11.2.0.4.jar:$JAVA_CLASSPATH"
# Run SQLLine
run_java sqlline.SqlLine $extra_args "$@"
```

```
sqlline --silent=true -u jdbc:oracle:thin:@<HOST>:1521:<INSTANCE> -n <SCHEMA> -p <password> --
outputformat=csv < <(cat test.sql) 2>/dev/null > out.txt
```

Change log of a application

You can add this ti your `spec.yaml` in config section

```
config:
  groups
    - name: _JAVA_OPTIONS
      description: nivel de log
      value: -Dlogging.level.org.springframework=TRACE
```

Local LDAP Test Server

You can use this github repo to start a local ldap server you can use to test LDAP features.

```
docker run -p 389:389 -p 636:636 --name ldap jgnagy/testldap:latest
```

He uses `sn` instead the default `sAMAccount`. Check the github repo for details.

Timezone issues

In Linux, timezone information refer the files `/etc/localtime` and `/etc/timezone` and `TZ` environment variable. Keep in mind that some stacks replaces or ignores the default Linux timezone configuration. If you are running java application, for example, it uses `${JAVA_HOME}/lib/tzdb.dat` file, in older versions was `${JAVA_HOME}/lib/zi` folder.

Dump timezone

- The file `/etc/localtime` probably is a symbolic link. You can detect the real file with `realpath /etc/localtime` command.
- The file linked in `/etc/localtime` can be dumped with `zdump /etc/localtime`. If you whant to check more details can use the flag `-V` as argument
- `tzupdater.jar` in `tzdata-java` package can be used to check `tzdb.dat`
- In java you can force specific timezone with `-Duser.timezone=GMT-3` in the start command

WSL2

WSL2 is a resource Windows users can use to run Linux resources.

Synchro does not recommend or support WSL2 use for Synchro Foundation.

That said, it can be useful for development purposes, so here are some tips to achieve a sucessful instalation of Synchro Foundation on WSL2.

> **Warning**
>
> Do not use WSL2 docker integrated with Windows. `dockerd` should be runned inside linux to be able to create the `iptables` rules. Otherwise Foundation should start but you will not be able to receive requests from outside WSL2 instance.

Installing WSL2

https://docs.microsoft.com/pt-br/windows/wsl/install-win10 https://docs.docker.com/docker-for-windows/wsl/

Setting up Linux distro

Distro: Ubuntu-20.04

> **Important**
>
> Run the distro as Administration otherwise `iptables` will fail

Creating WSL configuration

${HOME}/.wslconfig

```
[wsl2]
memory=4GB # Limits VM memory in WSL 2 to 4 GB
processors=2 # Makes the WSL 2 VM use two virtual processors
```

Installing Synchro Foundation

- Sample version 20.11.22, update to current one.

```
wget https://foundation.synchro.com.br/download/synchro-foundation-20.11.22-e6b251b9~centos-7-full.x86_64.rpm
sudo apt update
sudo apt install alien
sudo alien synchro-foundation-20.11.22-e6b251b9~centos-7-full.x86_64.rpm
sudo dpkg -i synchro-foundation_20.11.22-1_amd64.deb
```

- Edit `/etc/sysctl.conf` and set `net.ipv4.ip_forward=1`
- Run:

```
sudo sysctl -p
sudo mount --make-shared /
sudo mkdir /sys/fs/cgroup/systemd
sudo mount -t cgroup -o none,name=systemd cgroup /sys/fs/cgroup/systemd
```

CGroups Related problem

- https://github.com/microsoft/WSL/issues/4189

Starting docker inside WSL2

```
sudo dockerd
```

Running foundation

```
sudo foundation config
sudo foundation up
```

You can ignore `systemctl` related erros.

Discovering IP to open in browser

```
hostname -I
```

Get the IP, open in your browser and cross fingers. Good luck.