
Setup

Guida installazione

smartfactory40-setup.pdf



*Sistema di fabbrica per la pianificazione ed il controllo della
produzione conforme “Piano Transizione 4.0”*

rel. 0.09

Storia delle versioni

Data	Commento	Autore	Ver
16/02/23	prima stesura	sm	0.09

Sommario

Deploy SmartFactory40	2
Installazione DBMS	2
Installazione DBMS - Debian 11	2
Installare VMS	3
Installare SmartFactory40	3
Prerequisiti	6
Deploy SmartFactory40	6
Standard Release	8
Packed Release	10
Deploy VMS	11
Deploy Soft-EP	11
Predisposizione Ambienti	12
RaspberryPI	12
CentOS (RedHat Enterprise)	12
Configurazione vhost	14

Deploy SmartFactory40

Installazione DBMS

Il software SmartFactory40 utilizza il dbms mysql per memorizzare le informazioni relative ai dispositivi interconnessi e le informazioni del gestionale.

Installazione DBMS - Debian 11

L'installazione del server mysql viene eseguita con i seguenti comandi:

Per aggiungere i repository APT relativi a MySQL è necessario scaricare il seguente pacchetto:

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb
```

per poi installarlo:

```
sudo dpkg -i mysql-apt-config_0.8.22-1_all.deb
```

e aggiornare i pacchetti con i nuovi repository e procedere con l'installazione:

```
sudo apt update  
sudo apt install mysql-server
```

Dopo aver installato il DBMS è necessario creare un nuovo database e un utente che possiede i permessi per poter accedere.

Il servizio MySQL s avvia automaticamente, è possibile verificare lo stato con il comando:

```
sudo service mysql status
```

```
mysql> create schema smartfactory40;  
mysql> CREATE USER 'daas';  
mysql> GRANT ALL ON smartfactory40.* TO 'daas'@'localhost';  
mysql> FLUSH PRIVILEGES;  
mysql> exit;
```

Creare utente

```
groupadd -r daas
adduser -G daas daas
passwd daas ( entra. )
su -l daas
```

Installare VMS

Parte integrante dell'infrastruttura Daas è il gateway perimetrale denominato VMS che si occupa di interagire con i dispositivi "End-Point".

Scaricare il pacchetto per la piattaforma host:

????

Installare SmartFactory40

Scaricare i pacchetti base app:

```
curl https://www.smartfactory40.it/releases/smart-factory40-src-0.9.12.zip
```

Creare la cartella dove verranno poi decompressi i file:

```
mkdir /var/www/apps/smartfactory40
```

Dopo aver decompresso i sorgenti è possibile procedere con la compilazione e la configurazione della web-application.

YARN

```
yarn install
yarn package:build
yarn build
```

NPM

```
npm install
( on win10 powershell: npm install rimraf -g , npm i -g @nestjs/cli )

npm install -g npm@7.24.2 ( use old npm version )
npm run package:build
```

```
npm run build
```

Configurare la app

creare file .env.local (copiando il file .env) e aggiornare i parametri di configurazione

Lanciare la app

YARN

```
yarn start:prod
```

NPM

```
yarn start:prod
```

In caso di problemi di memoria:

YARN

```
NODE_OPTIONS="--max-old-space-size=4096" yarn start
```

NPM

```
NODE_OPTIONS="--max-old-space-size=4096" npm run start
```

configurare l'istanza VMS

Application:

```
vi .env.local
```

```
# contenuto tipo del file
DATABASE_TYPE=mariadb
DATABASE_HOST=localhost
DATABASE_PORT=3306
DATABASE_USER=vamp
DATABASE_PASSWORD=password
DATABASE_NAME=vamp_analyzer
```

Creazione servizio in systemd

Creiamo il file del servizio in:

```
nano /lib/systemd/system/smartfactory40.service
```

inserire e salvare la seguente configurazione

```
[Unit]
Description=SmartFactory4.0 App
After=network.target

[Service]
Environment=APP_SECRET=asdaksas3oqkfolwlf3pop
Environment=DATABASE_HOST=localhost
Environment=DATABASE_PORT=3306
Environment=DATABASE_USER=vamp
Environment=DATABASE_PASSWORD=password
Environment=DATABASE_NAME=camus
Environment=WEB_SERVER_PORT=3000
Type=simple
User=root (cambiare)
# se installazione da pacchetto
ExecStart=/path_alla_cartella_con_il_file_scaricato/smartfactory40-1.0 > prod.log
# se installazione via codice sorgente
ExecStart=/usr/bin/node path_alla_cartella_con_il_codice/dist/main > prod.log
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Ricaricare il daemon di systemd

```
systemctl daemon-reload
```

Avviare

```
systemctl start smartfactory40
```

Aggiungere la app tra quelle da avviare al boot

```
systemctl enable smartfactory40
```

Prerequisiti

Le installazioni sono portabili e richiedono ambienti configurati con i seguenti servizi:

`Node.js`
`Npm / Yarn`
`MariaDB`
`Nginx / Apache2`

Pianificare l'installazione

requisiti hardware e software
valutazione del traffico

Deploy SmartFactory40

Creare utente

```
groupadd -r daas
adduser -G daas daas
passwd daas ( entra. )
su -l daas
```

Creare database

```
sudo mysql
create schema smartfactory40
exit
sudo mysql smartfactory40
```

```
CREATE USER 'daas'
GRANT ALL ON *.* TO 'daas'@'localhost' IDENTIFIED BY 'password' WITH GRANT
OPTION;
FLUSH PRIVILEGES;
exit
```

Installare VMS

Scaricare il pacchetto per la piattaforma host:

Installare Smartfactory40

Standard Release

Scaricare pacchetti

i pacchetti base app:

```
curl https://www.smartfactory40.it/releases/smart-factory40-src-0.9.1.0.zip
```

(url: <https://demo.smartfactory40.it/assets/smart-factory-40-0.376.0.zip>)

Decomprimere i files nella cartella /var/www/apps/smartfactory40

unzip vms,

unzip app_

configurare l'app di front-end:

YARN

```
yarn install
yarn package:build
yarn build
```

NPM

```
npm install
( on win10 powershell: npm install rimraf -g , npm i -g @nestjs/cli )
```

```
npm install -g npm@7.24.2      ( use old npm version )
npm run package:build
npm run build
```

Configurare la app

creare file .env.local (copiando il file .env) e aggiornare i parametri di configurazione

Lanciare la app

YARN

```
yarn start:prod
```

NPM

```
yarn start:prod
```

In caso di problemi di memoria:

YARN

```
NODE_OPTIONS="--max-old-space-size=4096" yarn start
```

NPM

```
NODE_OPTIONS="--max-old-space-size=4096" npm run start
```

configurare l'istanza VMS

(se ci sono più vms sulla stessa macchina che succede ?)

Setup database

Database:

```
sudo mysql
```

```
CREATE SCHEMA vamp_analyzer;
```

```
GRANT ALL ON *.* TO 'vamp'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;
```

```
FLUSH PRIVILEGES;  
exit
```

Application:

```
vi .env.local
```

```
# contenuto tipo del file  
DATABASE_TYPE=mariadb  
DATABASE_HOST=localhost  
DATABASE_PORT=3306  
DATABASE_USER=vamp  
DATABASE_PASSWORD=password  
DATABASE_NAME=vamp_analyzer
```

```
sudo vi /etc/nginx/conf.d/vamp-analyzer.conf
```

```
# contenuto tipo del file  
server {  
    listen      8001 default_server;  
    listen      [::]:8001 default_server;  
    server_name _;  
    #root        /usr/share/nginx/html;  
    location / {  
        proxy_pass http://localhost:3000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
    error_page 404 /404.html;  
    location = /404.html {  
    }  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
    }  
}
```

```
sudo vi /etc/nginx/nginx.conf
```

```
# contenuto tipo del file  
user nginx;  
worker_processes auto;  
error_log /var/log/nginx/error.log;  
pid /run/nginx.pid;  
include /usr/share/nginx/modules/*.conf;  
events {  
    worker_connections 1024;  
}  
http {  
    log_format main '$remote_addr - $remote_user [$time_local] "$request" ' '  
        '$status $body_bytes_sent "$http_referer" ' '  
        '"$http_user_agent" "$http_x_forwarded_for"';  
    access_log /var/log/nginx/access.log main;  
    sendfile on;  
    tcp_nopush on;  
    tcp_nodelay on;  
    keepalive_timeout 65;  
    types_hash_max_size 2048;  
    include /etc/nginx/mime.types;  
    default_type application/octet-stream;  
    include /etc/nginx/conf.d/*.conf;
```

```
}
```

```
yarn start:web  
sudo systemctl start nginx
```

Packed Release

Per creare il pacchetto

```
pkg . -t x64 -o app --debug
```

Scaricare pacchetti

i pacchetti base app:

```
curl https://www.smartfactory40.it/releases/app\_smartfactory40-XXX.zip
```

<https://demo.smartfactory40.it/assets/smart-factory40-1.0> (da spostare)

```
chmod +x smart-factory40-1.0
```

Creazione servizio in systemd

Creiamo il file del servizio in:

```
nano /lib/systemd/system/smartfactory40.service
```

inserire e salvare la seguente configurazione

```
[Unit]  
Description=SmartFactory4.0 App  
After=network.target  
  
[Service]  
Environment=APP_SECRET=asdaksas3oqkfolwlf3pop  
Environment=DATABASE_HOST=localhost  
Environment=DATABASE_PORT=3306  
Environment=DATABASE_USER=vamp  
Environment=DATABASE_PASSWORD=password  
Environment=DATABASE_NAME=camus  
Environment=WEB_SERVER_PORT=3000  
Type=simple  
User=root (cambiare)  
# se installazione da pacchetto  
ExecStart=/path_alla_cartella_con_il_file_scaricato/smartfactory40-1.0 > prod.log  
# se installazione via codice sorgente  
ExecStart=/usr/bin/node path_alla_cartella_con_il_codice/dist/main > prod.log  
Restart=on-failure
```

```
[Install]  
WantedBy=multi-user.target
```

Ricaricare il daemon di systemd
`systemctl daemon-reload`

Avviare
`systemctl start smartfactory40`

Aggiungere la app tra quelle da avviare al boot
`systemctl enable smartfactory40`

Deploy VMS

```
curl https://www.smartfactory40.it/releases/vms\_xxx-xxxxxx.zip
```

Configurare il vms:

Deploy Soft-EP

```
curl https://www.smartfactory40.it/releases/ep\_Z32KOMPACTevo-v0.0.1\_CentOS.zip
```

Predisposizione Ambienti

RaspberryPI

CentOS (RedHat Enterprise)

User

```
useradd admin
passwd 1234
usermod -aG wheel nes
su - nes
```

Node.js

```
curl -sL https://rpm.nodesource.com/setup_10.x | sudo bash -
sudo yum install -y nodejs
```

Yarn

```
curl --silent --location https://dl.yarnpkg.com/rpm/yarn.repo |
sudo tee /etc/yum.repos.d/yarn.repo
sudo rpm --import https://dl.yarnpkg.com/rpm/pubkey.gpg
sudo yum install yarn
yarn --version
```

```
sudo yum install libsecret
sudo yum install git
```

nginx

```
sudo yum install epel-release
sudo yum install nginx
```

Firewalld

```
sudo cp /usr/lib/firewalld/services/http.xml
/usr/lib/firewalld/services/http01.xml
sudo vi /usr/lib/firewalld/services/http01.xml
sudo firewall-cmd --permanent --zone=public --add-service=http01
sudo firewall-cmd --reload
```

MariaDB

```
sudo yum install mariadb-server
```

Installare nodejs

<https://nodejs.org/en/download/package-manager/>

Nginx (web server / reverse proxy)

```
sudo apt update
sudo apt install nginx
```

Configurazione vhost

```
sudo rm /etc/nginx/sites-enabled/default
sudo nano /etc/nginx/sites-enabled/smart-factory
```

contenuto del file

```
server {
    listen 80;
    listen [::]:80;

    server_name www.IL_DOMINIO.com;

    location / {
        proxy_pass http://localhost:3001;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}

sudo service nginx restart
```