



Co-funded by European Union
Civil Protection



PROMETHEUS PROJECT

A Data Management System For USAR Operations

INSTALLATION MANUAL

Procedures to install Prometheus software



Indice

1. Introduction	3
1.1 Document scope.....	3
2. General description	3
2.1 Architecture and schemes	3
2.2 Sector of interest	4
2.3 Application Domain.....	4
2.4 Service Level Agreement	4
2.5 Security	4
2.6 User access, authentication and privileged	4
2.7 Developer platform and frameworks	4
2.8 Service names	4
3 Technical requirements.....	5
3.1 Mobile and web client requirements	5
3.2 Server requirements	5
3.3 Network requirements	5
4 Installation and configuration	5
4.1 Mobile app	5
4.2 Server installation	5
Virtual Machine installation	5
Containerized installation	6
Docker.compose.yml	6
.env file.....	11
4.4 Post installation	12
Configure Grafana dashboards.....	12

1. Introduction

1.1 Document scope

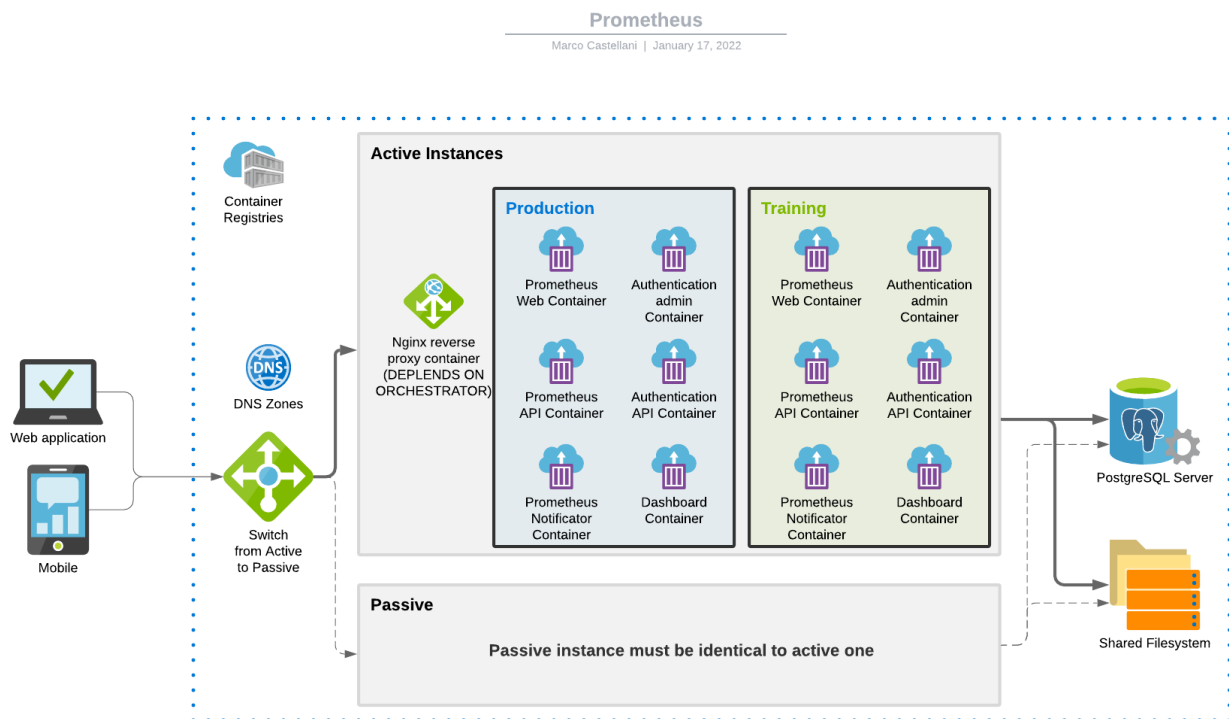
The purpose of this document is to provide a description of the IT infrastructure for this procedure.

2. General description

The application aims to allow the proper management of information relating to emergency events of seismic nature, focused on the search and rescue of people. The PROMETHEUS platform is divided into several modules, such as: WEB management interface for UCC operators, Mobile Android application, WEB interface for the management of users and their application permissions, the database to store data

2.1 Architecture and schemes

Below you can find an overall architectural drawing, where you can find how modules cooperate together.



2.2 Sector of interest

The area of interest is management.

2.3 Application Domain

The domain of the application is the rescue of people in the context of emergencies caused by earthquakes, both locally and nationally. The actors interested in using the platform belong mainly to 2 categories:

- USAR rescuers
- External observer

In the first case, these are people belonging to the Department of Fire, Public Rescue and Civil Defense; the latter category includes all those figures who need to be able to monitor the evolution of the emergency, both in terms of how many victims have been identified, and the geographical location and pressure on the territory caused by the emergency itself.

2.4 Service Level Agreement

Given the criticality of the application domain, the system is designed to be installed in an Active-Passive cluster to guarantee "High Availability" or equivalent containered infrastructure. The installer will have to define the appropriate "Disaster Recovery" and "Business Continuity" plans.

2.5 Security

The system requires the HTTPS standard with 2048bit certificate signed with "SHA-256 with RSA Encryption" for its web sites. The system is composed by some http/s endpoints for which it is necessary to provide the appropriate DNS configurations and SSL certificates. The database must not be reachable from outside. The authentication server uses the OpenId Connect 2/OAUTH standard and generates during installation a signature certificate that can be replaced with a system intervention.

2.6 User access, authentication and privileged

There is four kinds of user's role:

- Admin: certified user who can create other users and change their access permissions
- Managerial+Operational: Web user managing the UCC portal only
- User: user who is allowed to login on the mobile device (on field rescuer)
- Observer: external user, displays data in read-only mode and some portal information is obfuscated.

2.7 Developer platform and frameworks

The server component of the application is written with Microsoft .net core 3.1, and for this reason it can run natively on both Windows and Linux OSs. Docker container has been chosen as the main mode of releasing software, it is recommended to provide a container orchestrator such as Kubernetes and a registry for images to create a production environment.

The mobile app is written in Microsoft Xamarin Forms and made available through APK published both on the institutional website of Prometheus project and on the official Google store.

Data are stored in some databases of a PostgreSQL RDBMS. To ensure High Availability it is necessary to have an HA instance PostgreSQL and a shared network share where the images and audio files will be hosted.

2.8 Service names

The services' names are freely changeable and can be find in the container configuration, Below you can find a docker-compose.yml file in which database and reverse proxy have been add to the file together with the applications containers. The same docker compose is used to configure Prometheus inside the virtual machine.

3 Technical requirements

3.1 Mobile and web client requirements

The smartphone app is provided via apk and can be installed on Android 6.0 or higher devices. Android devices are often clipped or customized especially on the multimedia side; we recommend installing from Google Play Store the app "VLC for Android". Web applications supports Chrome, Edge, Firefox, Safari Osx browser in current version.

3.2 Server requirements

An example Prometheus installation has been realized installing and configuring containers inside a virtual machine based on the operating system Ubuntu 20.04.3 LTS, in Open Virtual Appliance (ova) format usable in any virtualization program, enabling the support for the extensions VT-X of Intel or AMD-V of AMD. This file is freely downloadable from Prometheus project website and, can be considered "ready for a production". Unfortunately, with only one VM will not be possible to guarantee high availability. Additional requirements are:

- DNS names for web access
- HTTPS connection through appropriate SSL certificate
- Clustered PostgreSQL v12 with PosGIS extensions enabled
- Network Share for images and files

To run the virtual machine, we suggest:

- 8 VCPU
- 16 GB RAM
- 10 GB of free space (the sizing of the space needed is highly dependent on the usage guidelines that will be adopted for field operations).

In the operational phase Prometheus will be unused for a long time and will suffer peaks of use when some serious event occurs; ensuring an "elastic" allocation of resources could be an important factor to optimize the available resources normally.

3.3 Network requirements

The software is accessed entirely via https on port 443. The relative IP must be exposed on the Internet and resolved by some DNS alias (see above).

4 Installation and configuration

4.1 Mobile app

The mobile app is an Android apk. This file must be provided to users using the common techniques. For example it can be downloaded via HTTP/S.

4.2 Server installation

The installation of services used by Prometheus app and web portal is available using Docker containers. In this way it is possible to implement high availability scenarios. Along with the containers we also provide a Virtual Machine preconfigured with Docker-Compose. This VM is also suitable for a quick evaluation of the software or as a reference for a more sophisticated configuration.

Prometheus needs a PostgreSQL v12 database with PosGIS extensions enabled; this database must guarantee its HA by itself.

Prometheus needs some volumes to store logs, images, and other data. These volumes, in distributed scenarios, must be guaranteed HA. For example, volumes can be mounted on shared file shares

Virtual Machine installation

The virtual machine for Prometheus services is available through an .ova file (see above) accessible with the credentials

login: Prometheus

password: Prometheus

Whilst default admin for web interface is

user: admin
password: 123Prometheus!

After logging in, the following steps should be taken:

- Define the DNS names of the application and make sure they are globally recognized.
If this is not possible, edit the /etc/hosts file with the domain names used by the services.
- Compare the env-file section for the list of services needed.

If you want to test installation before having DNS records, you can just copy /etc/hosts to the hosts file in the machine where you want to use the browser.

- Change directory to prometheus-server folder
- Modify env-example file and setup your variables according to /etc/hosts or your real domain
- rename env-example to .env.
- run command **docker-compose up -d**

NOTE: depending on your configuration you may need to give write permission to grafana folder (ex. command: sudo chmod a+w grafana/ -R)

NOTE: we strongly recommend using HTTPS configuring NGINX template inside templates folder

NOTE: first run of "docker-compose up" configures all the systems using the DNS names. Run it only when your DNS are set correctly or otherwise start over deleting folders and databases

Containered installation

Below you can find the docker-compose and .env files (also present in the attachment) as an example of a possible configuration. The examples are the same as those found within the evaluation VM.

Docker.compose.yml

```
version: "3.4"

services:
  vvf.prometheus.web:
    image: registry.prometheusproject.eu/vvf.prometheus.web:latest
    container_name: vvf.dev.prometheus.web
    environment:
      - ASPNETCORE_URLS=http://+:80
      # JS variables
      - ClientId=js_vvf
      - Scope=apires_vvf
      - ApiUri=${ApiUri}/api/v1/
      - EnvironmentCss=real
      - OtherEnvironmentUri=${OtherEnvironmentUri}
      - RedirectUri=${WebAppUri}/signIn.html
      - AuthorizationUri=${IdentityUri}/connect/authorize
      - LogoutUrl=${IdentityUri}/account/logout?autologout=true
      - BaseUri=${ApiUri}/
      - AdministrationUrl=${IdentityAdminUri}
    depends_on:
      - postgres-db
    expose:
      - "80"
    restart: always
    networks:
      prometheus:

  vvf.prometheus.api:
    image: registry.prometheusproject.eu/vvf.prometheus.api:latest
    container_name: vvf.dev.prometheus.api
    environment:
```

```

- ASPNETCORE_URLS=http://+:80
- AppSettings__AuthorityUrl=${IdentityUri}
- AppSettings__ApiClientId=vvf.api
- AppSettings__ApiName=apires_vvf
- AppSettings__ApiSecret=vvfsecret
- AppSettings__LogoutUrl=${IdentityUri}/account/logout?autologout=true
- AppSettings__ChangePasswordUrl=${IdentityUri}/account/changepassword?mail=
- AppSettings__AdministrationUrl=${IdentityAdminUri}
- AppSettings__DefaultState=IT
- AppSettings__CORSOrigins=${WebAppUri},${ApiUri}
- AppSettings__WebAlias=${WebAppUri}
- AppSettings__DocumentAreaBasePath=/document-area
- AppSettings__DocumentMaxFileSizeInMb=5
- AppSettings__ApiRequireHttpsMetadata=${BlockUnsecure}
- Logging__LogLevel__Default=Information
- ConnectionStrings__PrometheusContext=Host=postgres-db;Database=vvf_prometheus;Username=postgres;Password=docker;
- ConnectionStrings__ArchiveConnectionString=Host=postgres-db;Database=vvf_prometheus_archive;Username=postgres;Password=docker;
- AppSettings__IsArchive=false
# JS Variables
- ClientId=js_vvf
- Scope=apires_vvf
- RedirectUri=${ApiUri}/LocalizationAdmin/signIn.html
- AuthorizationUri=${IdentityUri}/connect/authorize
- LogoutUrl=${IdentityUri}/account/logout?autologout=true
depends_on:
- postgres-db
expose:
- "80"
volumes:
- application-api-log:/app/logs
- application-uploads:/app/uploads
- application-kml-uploads:/app/kml_uploads
- document-area:/document-area
restart: always
networks:
prometheus:

```

vvf.prometheus.notificator:

image: registry.prometheusproject.eu/vvf.prometheus.notificator:latest

container_name: vvf.dev.prometheus.notificator

environment:

```

- ConnectionStrings__PrometheusContext=Host=postgres-db;Database=vvf_prometheus;Username=postgres;Password=docker;
# SMTP
- SmtpSettings__Host=${SmtpSettings__Host}
- SmtpSettings__Port=${SmtpSettings__Port}
- SmtpSettings__Username=${SmtpSettings__Username}
- SmtpSettings__Password=${SmtpSettings__Password}
- SmtpSettings__From=${SmtpSettings__From}
- SmtpSettings__DisplayName=${SmtpSettings__DisplayName}
# App settings
- AppSettings__WebAlias=${WebAppUri}
depends_on:
- postgres-db
restart: always
networks:
prometheus:

```

vvf.prometheus.training.web:

image: registry.prometheusproject.eu/vvf.prometheus.web:latest

container_name: vvf.dev.prometheus.training.web

environment:

```

- ASPNETCORE_URLS=http://+:80
# JS variables
- ClientId=js_vvf
- Scope=apires_vvf
- ApiUri=${TrainingApiUri}/api/v1/
- EnvironmentCss=training
- OtherEnvironmentUri=${WebAppUri}
- RedirectUri=${OtherEnvironmentUri}/signIn.html
- AuthorizationUri=${IdentityUri}/connect/authorize
- LogoutUrl=${IdentityUri}/account/logout?autologout=true
- BaseUri=${TrainingApiUri}
- AdministrationUrl=${IdentityAdminUri}
depends_on:
- postgres-db
expose:

```

```

- "80"
restart: always
networks:
  prometheus:

vvf.prometheus.training.api:
image: registry.prometheusproject.eu/vvf.prometheus.api:latest
container_name: vvf.dev.prometheus.training.api
environment:
  - ASPNETCORE_URLS=http://+:80
  - AppSettings__AuthorityUrl=${IdentityUri}
  - AppSettings__ApiClientId=vvf.api
  - AppSettings__ApiName=apires_vvf
  - AppSettings__ApiSecret=vvfsecret
  - AppSettings__LogoutUrl=${IdentityUri}/account/logout?autologout=true
  - AppSettings__ChangePasswordUrl=${IdentityUri}/account/changepassword?mail=
  - AppSettings__AdministrationUrl=${TrainingApiUri}
  - AppSettings__DefaultState=IT
  - AppSettings__CORSOrigins=${OtherEnvironmentUri};${TrainingApiUri}
  - AppSettings__WebAlias=${OtherEnvironmentUri}
  - AppSettings__DocumentAreaBasePath=/document-area
  - AppSettings__DocumentMaxFileSizeInMb=5
  - Logging__LogLevel__Default=Information
  - ConnectionStrings__PrometheusContext=Host=postgres-db;Database=vvf_prometheus_training;Username=postgres;Password=docker;
  - ConnectionStrings__ArchiveConnectionString=Host=postgres-db;Database=vvf_prometheus_archive;Username=postgres;Password=docker;
  - AppSettings__IsArchive=true
  - AppSettings__ApiRequireHttpsMetadata=${BlockUnsecure}
# JS Variables
  - ClientId=js_vvf
  - Scope=apires_vvf
  - RedirectUri=${TrainingApiUri}/LocalizationAdmin/signIn.html
  - AuthorizationUri=${IdentityUri}/connect/authorize
  - LogoutUrl=${IdentityUri}/account/logout?autologout=true
depends_on:
  - postgres-db
expose:
  - "80"
volumes:
  - application-training-api-log:/app/logs
  - application-training-uploads:/app/uploads
  - application-training-kml-uploads:/app/kml_uploads
  - document-area-training:/document-area
restart: always
networks:
  prometheus:

vvf.prometheus.training.notificator:
image: registry.prometheusproject.eu/vvf.prometheus.notificator:latest
container_name: vvf.dev.prometheus.training.notificator
environment:
  - ConnectionStrings__PrometheusContext=Host=postgres-db;Database=vvf_prometheus_training;Username=postgres;Password=docker;
# SMTP
  - SmtpSettings__Host=${SmtpSettings__Host}
  - SmtpSettings__Port=${SmtpSettings__Port}
  - SmtpSettings__Username=${SmtpSettings__Username}
  - SmtpSettings__Password=${SmtpSettings__Password}
  - SmtpSettings__From=${SmtpSettings__From}
  - SmtpSettings__DisplayName=${SmtpSettings__DisplayName}
# App settings
  - AppSettings__WebAlias=${OtherEnvironmentUri}
depends_on:
  - postgres-db
restart: always
networks:
  prometheus:

vvf.prometheus.archive.web:
image: registry.prometheusproject.eu/vvf.prometheus.web:latest
container_name: vvf.dev.prometheus.archive.web
environment:
  - ASPNETCORE_URLS=http://+:80
# JS variables
  - ClientId=js_vvf
  - Scope=apires_vvf
  - ApiUri=${ArchiveApiUri}/api/v1/
  - EnvironmentCss=archive

```


- OtherEnvironmentUri=\${WebAppUri}
- RedirectUri=\${ArchiveWbApp}/signIn.html
- AuthorizationUri=\${IdentityUri}/connect/authorize
- LogoutUri=\${IdentityUri}/account/logout?autologout=true
- BaseUri=\${ArchiveApiUri}
- AdministrationUrl=\${IdentityAdminUri}

depends_on:

- postgres-db

expose:

- "80"

restart: always

networks:

prometheus:

vvf.prometheus.archive.api:

image: registry.prometheusproject.eu/vvf.prometheus.api:latest

container_name: vvf.dev.prometheus.archive.api

environment:

- ASPNETCORE_URLS=http://+:80
- AppSettings__AuthorityUrl=\${IdentityUri}
- AppSettings__ApiClientId=vvf.api
- AppSettings__ApiName=apires_vvf
- AppSettings__ApiSecret=vvfsecret
- AppSettings__LogoutUrl=\${IdentityUri}/account/logout?autologout=true
- AppSettings__ChangePasswordUrl=\${IdentityUri}/account/changepassword?mail=
- AppSettings__AdministrationUrl=\${IdentityAdminUri}
- AppSettings__DefaultState=IT
- AppSettings__CORSSOrigins=\${ArchiveWbApp};\${IdentityAdminUri}
- AppSettings__WebAlias=\${ArchiveWbApp}
- AppSettings__DocumentAreaBasePath=/document-area
- AppSettings__DocumentMaxFileSizeInMb=5
- Logging__LogLevel__Default=Information
- ConnectionStrings__PrometheusContext=Host=postgres-db;Database=vvf_prometheus_archive;Username=postgres;Password=docker;
- AppSettings__IsArchive=true
- AppSettings__ApiRequireHttpsMetadata=\${BlockUnsecure}

JS Variables

- ClientId=js_vvf
- Scope=apires_vvf
- RedirectUri=\${ArchiveApiUri}/LocalizationAdmin/signIn.html
- AuthorizationUri=\${IdentityUri}/connect/authorize
- LogoutUri=\${IdentityUri}/account/logout?autologout=true

depends_on:

- postgres-db

expose:

- "80"

volumes:

- application-archive-api-log:/app/logs
- application-uploads:/app/uploads
- application-kml-uploads:/app/kml_uploads
- document-area:/document-area

restart: always

networks:

prometheus:

giuneco.identity:

image: registry.prometheusproject.eu/giuneco.identity.server:latest

container_name: vvf.dev.identity.server

environment:

- ConnectionStrings__ClientConnectionString=Host=postgres-db;Database=vvf_identity;Username=postgres;Password=docker;
- ConnectionStrings__ClientConnectionStringDev=Host=postgres-db;Database=vvf_identity;Username=postgres;Password=docker;
- AppSettings__Authority=\${IdentityUri}
- AppSettings__Application=\${WebAppUri}
- AppSettings__ApplicationApi=\${ApiUri}
- AppSettings__ApplicationUriRedirect=\${WebAppUri},\${OtherEnvironmentUri},\${ArchiveWbApp}
- AppSettings__ApplicationApiUriRedirect=\${ApiUri},\${TrainingApiUri},\${ArchiveApiUri}
- AppSettings__ApiSecret=vvfsecret
- AppSettings__MobileApiSecret=vvfmobilesecret
- AdminConfiguration__IdentityAdminBaseUrl=\${IdentityAdminUri}
- AdminConfiguration__IdentityServerBaseUrl=\${IdentityUri}
- ASPNETCORE_URLS=http://+:80
- CLIENT=VVF
- DB_PROVIDER=postgresql

SMTP

- SmtptSenderSettings__FromEmail=prometheus@giuneco.com
- SmtptSenderSettings__FromName=Prometheus Notificator
- SmtptSenderSettings__Host=\${SmtptSettings__Host}

```

- SmtplibSenderSettings__Port=587
- SmtplibSenderSettings__username=prometheus@giuneco.com
- SmtplibSenderSettings__password=1d85f43346ee8770cd8e4d38d310aaac-2fbc671d-f400cab1
- SmtplibSenderSettings__SecureSocketOptions=StartTls
- ThirdPartySettings__GrafanaSettings__ClientId=grafana
- ThirdPartySettings__GrafanaSettings__ClientSecret=123Grafana!
- ThirdPartySettings__GrafanaSettings__RedirectUri=${GrafanaUri}/login/generic_oauth

```

depends_on:

```
- postgres-db
```

expose:

```
- "80"
```

restart: always

entrypoint: >

```

/bin/sh -c "
    /usr/sbin/update-ca-certificates
    cd /app
    dotnet Giuneco.Identity.dll
"
```

networks:

prometheus:

giuneco.identity.admin:

image: registry.prometheusproject.eu/giuneco.identity.admin:latest

container_name: vvf.dev.identity.admin

environment:

```

- ConnectionStrings__ClientConnectionString=Host=postgres-db;Database=vvf_identity;Username=postgres;Password=docker;
- AdminConfiguration__IdentityServerBaseUrl=${IdentityUri}
- AdminConfiguration__IdentityAdminBaseUrl=${IdentityAdminUri}
- AdminConfiguration__ClientSecret=ClientSecret
- ASPNETCORE_URLS=http://+:80
- DB_PROVIDER=postgresql

```

depends_on:

```

- postgres-db
- giuneco.identity

```

expose:

```
- "80"
```

restart: always

networks:

prometheus:

grafana:

image: grafana/grafana:8.1.0

container_name: vvf.dev.grafana

networks:

prometheus:

volumes:

```

- ./grafana/data:/var/lib/grafana
- ./grafana/log:/var/log/grafana

```

environment:

```

- GF_SERVER_DOMAIN=${GrafanaUri}
- GF_SERVER_ROOT_URL=${GrafanaUri}/
- GF_DATABASE_TYPE=postgres
- GF_DATABASE_HOST=postgres-db
- GF_DATABASE_NAME=vvf_grafana
- GF_DATABASE_USER=postgres
- GF_DATABASE_PASSWORD=docker
- GF_USERS_DEFAULT_THEME=light
- GF_AUTH_GENERIC_OAUTH_ENABLED=true
- GF_AUTH_GENERIC_OAUTH_NAME=Prometheus Auth
- GF_AUTH_GENERIC_OAUTH_ALLOW_SIGN_UP=true
- GF_AUTH_GENERIC_OAUTH_CLIENT_ID=grafana
- GF_AUTH_GENERIC_OAUTH_CLIENT_SECRET=123Grafana!
- GF_AUTH_GENERIC_OAUTH_SCOPES=openid profile roles email
- GF_AUTH_GENERIC_OAUTH_EMAIL_ATTRIBUTE_PATH=email
- GF_AUTH_GENERIC_OAUTH_AUTH_URL=${IdentityUri}/connect/authorize
- GF_AUTH_GENERIC_OAUTH_TOKEN_URL=${IdentityUri}/connect/token
- GF_AUTH_GENERIC_OAUTH_API_URL=${IdentityUri}/connect/userinfo
- GF_AUTH_GENERIC_OAUTH_ROLE_ATTRIBUTE_PATH=role
- GF_AUTH_GENERIC_OAUTH_TLS_SKIP_VERIFY_INSECURE=true
- GF_AUTH_OAUTH_AUTO_LOGIN=true
- GF_SECURITY_ALLOW_EMBEDDING=true
- GF_INSTALL_PLUGINS=grafana-worldmap-panel

```

depends_on:

```

- postgres-db
- giuneco.identity

```

restart: always

```

postgres-db:
  image: postgis/postgis:12-master
  container_name: vvf.dev.postgresql
  environment:
    - POSTGRES_USER=postgres
    - POSTGRES_PASSWORD=docker
    - POSTGRES_MULTIPLE_DATABASES=vvf_identity,vvf_prometheus,vvf_prometheus_training,vvf_prometheus_archive,vvf_grafana
  volumes:
    - pg-data:/var/lib/postgresql/data
    - ./pg-init-scripts:/docker-entrypoint-initdb.d
  restart: always
  networks:
    prometheus:

nginx:
  image: nginx:latest
  volumes:
    - ./templates:/etc/nginx/templates
  environment:
    - ApiUriName=${ApiUriName}
    - TrainingApiUriName=${TrainingApiUriName}
    - ArchiveApiUriName=${ArchiveApiUriName}
    - WebAppUriName=${WebAppUriName}
    - OtherEnvironmentUriName=${OtherEnvironmentUriName}
    - ArchiveWbAppName=${ArchiveWbAppName}
    - IdentityUriName=${IdentityUriName}
    - IdentityAdminUriName=${IdentityAdminUriName}
    - GrafanaUrlName=${GrafanaUrlName}
  ports:
    - "80:80"
  networks:
    prometheus:
  depends_on:
    - giuneco.identity
    - giuneco.identity.admin
    - grafana

volumes:
  pg-data:
    name: vvf-dev-database
  application-uploads:
    name: vvf-dev-uploads
  application-training-uploads:
    name: vvf-dev-training-uploads
  application-kml-uploads:
    name: vvf-dev-kml-uploads
  application-training-kml-uploads:
    name: vvf-dev-training-kml-uploads
  application-api-log:
    name: vvf-dev-api-log
  application-training-api-log:
    name: vvf-dev-training-api-log
  application-archive-api-log:
    name: vvf-dev-archive-api-log
  document-area:
    name: vvf-dev-document-area
  document-area-training:
    name: vvf-dev-document-area-training

networks:
  prometheus:

```

.env file

```

WebAppUriName=web.prometheus.local
ApiUriName=api.prometheus.local
OtherEnvironmentUriName=train.prometheus.local
TrainingApiUriName=train_api.prometheus.local
ArchiveWbAppName=archive.prometheus.local
ArchiveApiUriName=archive_api.prometheus.local
IdentityAdminUriName=identity_admin.prometheus.local
IdentityUriName=identity.prometheus.local

```

```
GrafanaUrlName=grafana.prometheus.local

WebAppUri=http://web.prometheus.local
ApiUri=http://api.prometheus.local
OtherEnvironmentUri=http://train.prometheus.local
TrainingApiUri=http://train_api.prometheus.local
ArchiveWebApp=http://archive.prometheus.local
ArchiveApiUri=http://archive_api.prometheus.local
IdentityAdminUri=http://identity_admin.prometheus.local
IdentityUri=http://identity.prometheus.local
GrafanaUrl=http://grafana.prometheus.local
```

```
SmtpSettings__Host=XXXXXX
SmtpSettings__Port=587
SmtpSettings__Username=prometheus@giuneco.com
SmtpSettings__Password=XXXXX
SmtpSettings__From=XXXX
SmtpSettings__DisplayName=Prometheus Notificator
BlockUnsecure=false
```

To configure the database it is necessary to put the files contained in pg-init-scripts.zip next to the docker-compose.yml to allow the correct seed of the databases. In case of installation on an external database, check the consistency of the variables within the scripts

4.4 Post installation

At the end of startup process you can login at the address:
<http://web.prometheus.local> or the one you choose in DNS settings

user: admin

password: 123Prometheus!

Configure Grafana dashboards

Open <http://grafana.prometheus.local> or you address for this container

- go to "Configuration"->"Data Sources" menu,
- add postgres db as default datasource (according to your database installation or reading it from docker-config.yml)
- Go to "Dashboard"->"Manage" and menu
Import dashboards present in DashBoard folder attached to this manual.

infoprometheus@vigilfuoco.it
www.prometheusproject.eu
<https://www.vigilfuoco.it/asp/asp/Page.aspx?IdPage=10314>



Italian National Fire and Rescue Service
Piazza del Viminale, 1 - Rome
www.prometheusproject.eu

Disclaimer:
The content of this Manual represents the views of the author only and is his/her sole responsibility.
The European Commission does not accept any responsibility for use that may be made of the information it contains.