

Scaling Zabbix with containers

Robert Silva

Tech Leader at JLCP

Zabbix Trainer and Zabbix Expert



Zabbix project requirement

- Monitor more than 3k NVPS;
- Fault tolerant;
- Resilient;
- Scale horizontally.



Usual installation

- Virtual machines/Physical servers;
- Operating system;
 - Install
 - Update
 - Tuning
- Install Zabbix
- Backup files



Usual problems

- Services unavailable:
 - Frontend down;
 - Zabbix server down;



Usual solution

- Human intervention to restart;
 - Create a new instance;
 - Restore backup.
-
- This is not enough, we need more.



Standard solutions

- Corosync;
- Pacemaker;

That is a “standard solution”.



Nowadays, there are **better solutions for HA** than corosync/pacemaker

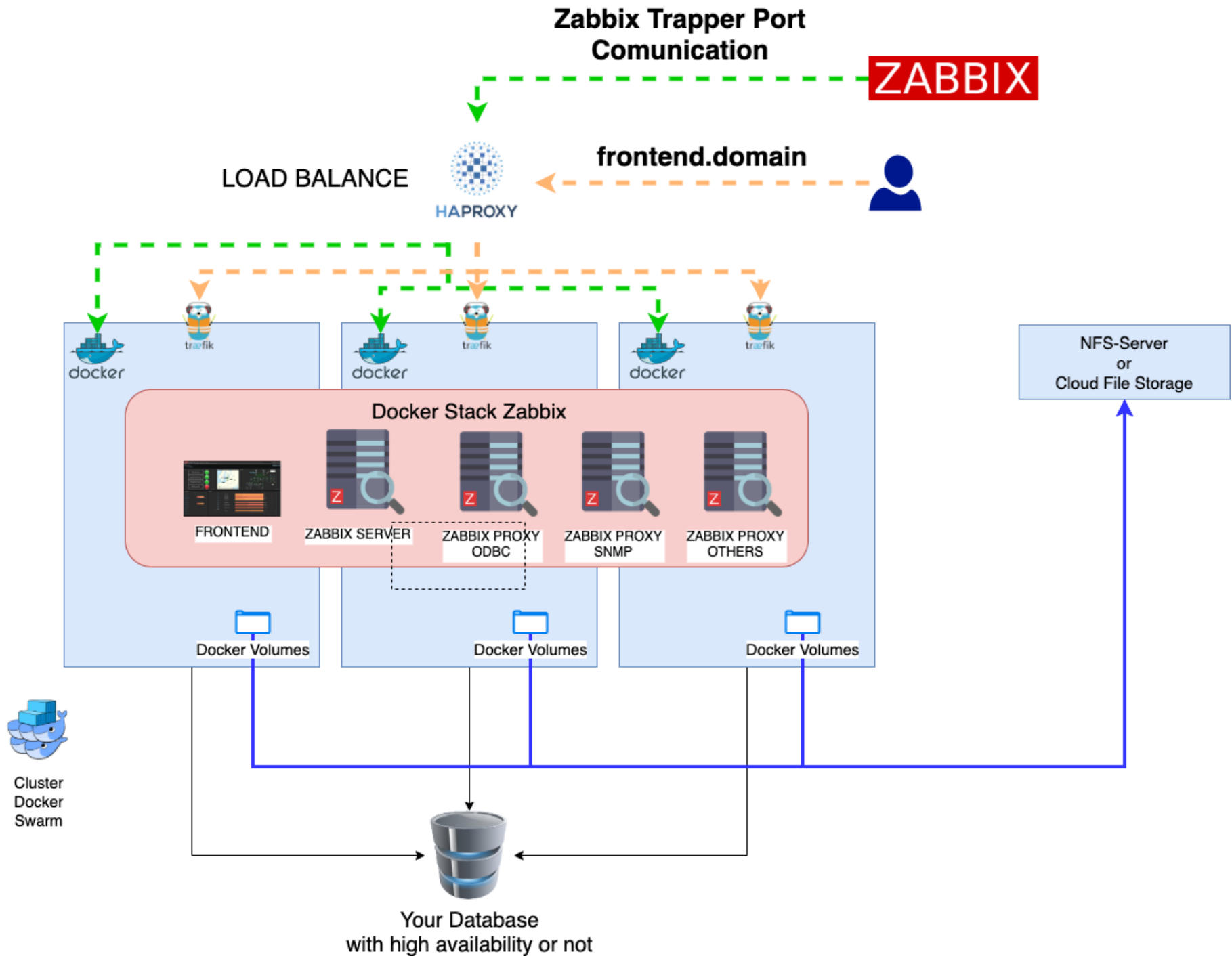
**I won't talk about
database!**

New approach

- Docker;
- Orchestration with Docker Swarm;
- Reverse Proxy;
- GIT
- Plus: CI/CD



Initial architecture

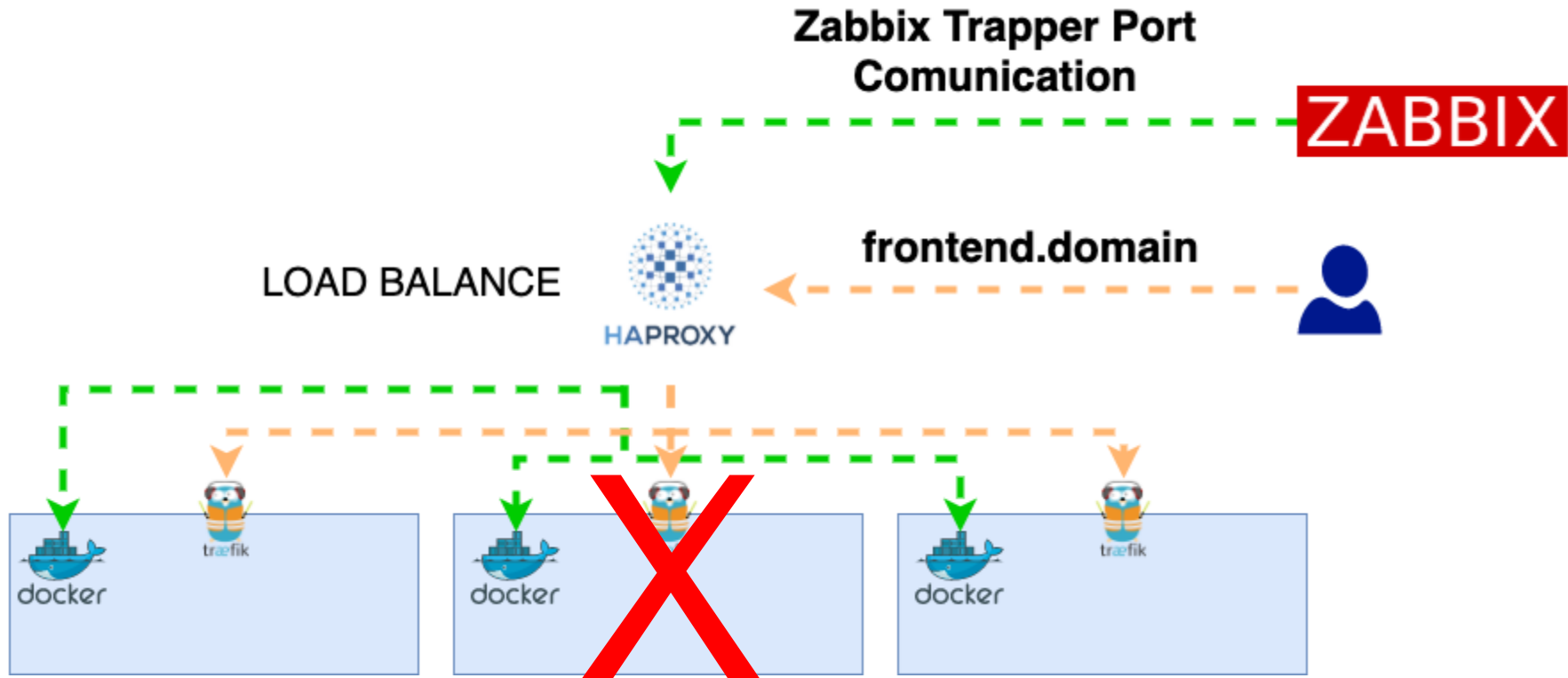


HA Proxy

- Responsible for receiving incoming connections and directing them to the nodes of the docker swarm cluster



HA Proxy



HA Proxy (haproxy.cfg)

```
frontend traefik
mode http
bind 0.0.0.0:80
option forwardfor
monitor-uri /health
default_backend backend_traefik
```

```
backend backend_traefik
mode http
cookie Zabbix prefix
server DOCKERHOST1 10.250.6.52:8080 cookie DOCKERHOST1 check
server DOCKERHOST2 10.250.6.53:8080 cookie DOCKERHOST2 check
server DOCKERHOST3 10.250.6.54:8080 cookie DOCKERHOST3 check
stats admin if TRUE
option tcp-check
```



HA Proxy (haproxy.cfg)

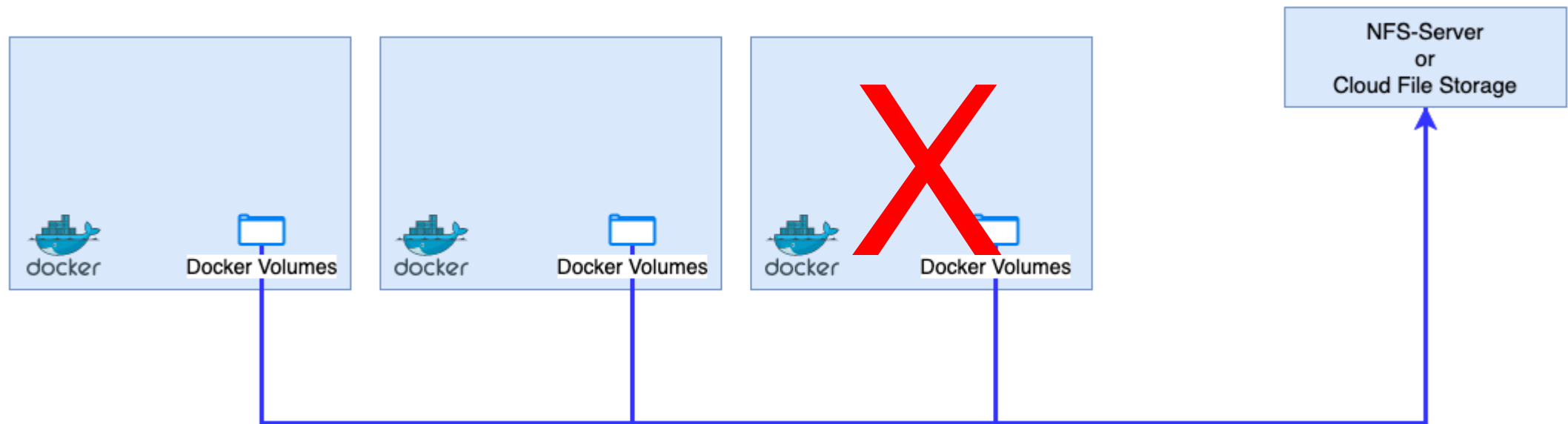
```
frontend zabbix_server
mode tcp
bind 0.0.0.0:10051
default_backend backend_zabbix_server
```

```
backend backend_zabbix_server
mode tcp
server DOCKERHOST1 10.250.6.52:10051 check
server DOCKERHOST2 10.250.6.53:10051 check
server DOCKERHOST3 10.250.6.54:10051 check
stats admin if TRUE
option tcp-check
```



NFS Server

- Responsible for storing the mapped files in the containers.



NFS Server

NFS Server

```
mkdir /data/data-docker  
vim /etc/exports  
/data/data-docker/ *(rw,sync,no_root_squash,no_subtree_check)
```

NFS Client

```
vim /etc/fstab <NFS_IP>:/data/data-docker /mnt/data-docker nfs defaults 0 0
```



Hosts Docker and Docker Swarm

- Responsible for perform and orchestrating the containers;
- Swarm consists of one or more nodes.
- There are two types:
 - Managers
 - Workers

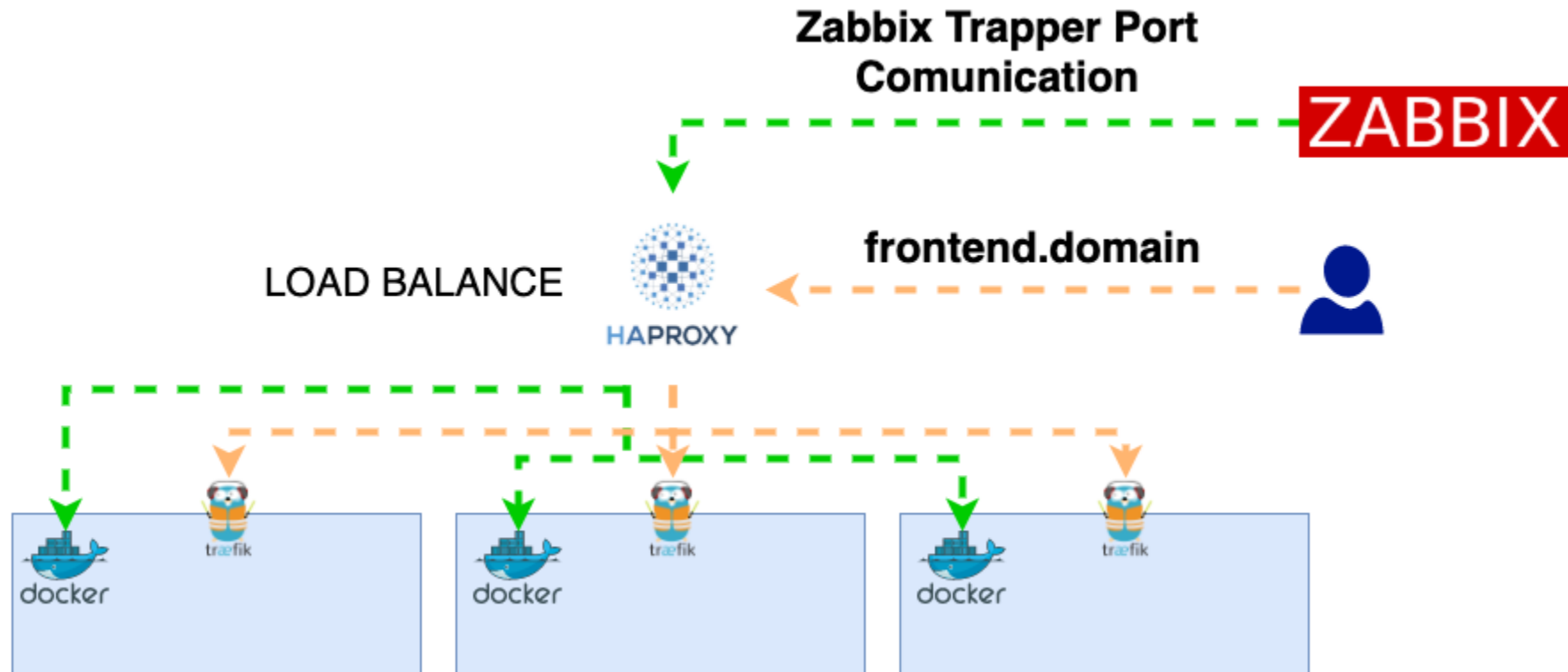


Reverse Proxy

- Responsible for receiving a connection and redirecting to the containers;
- Usually used nginx and traefik



Reverse Proxy



Examples of the Compose file and Deploy

What is the compose file?

The Compose file is a YAML file defining services, networks and volumes.

The default path for a Compose file is `./docker-compose.yml`



Reverse Proxy

traefik:

image: traefik:v2.2.8

deploy:

placement:

constraints:

- node.role == manager

replicas: 1

restart_policy:

condition: on-failure

labels:

Dashboard traefik

- "traefik.enable=true"

- "traefik.http.services.justAdummyService.loadbalancer.server.port=1337"

- "traefik.http.routers.traefik.tls=true"

- "traefik.http.routers.traefik.rule=Host(`zabbix-traefik.mydomain`)"

- "traefik.http.routers.traefik.service=api@internal"



Reverse Proxy

Auth Dashboard

- "traefik.http.routers.traefik.middlewares=traefik-auth"
- "traefik.http.middlewares.traefik-auth.basicauth.users=admin:<password>"

Redirect all HTTP to HTTPS permanently

- "traefik.http.routers.http_catchall.rule=HostRegexp(`{any:.+}`)"
- "traefik.http.routers.http_catchall.entrypoints=web"
- "traefik.http.routers.http_catchall.middlewares=https_redirect"
- "traefik.http.middlewares.https_redirect.redirectscheme.scheme=https"
- "traefik.http.middlewares.https_redirect.redirectscheme.permanent=true"



Reverse Proxy

command:

- "--api=true"
- "--log.level=INFO"
- "--providers.docker.endpoint=unix:///var/run/docker.sock"
- "--providers.docker.swarmMode=true"
- "--providers.docker.exposedbydefault=false"
- "--providers.file.directory=/etc/traefik/dynamic"
- "--entrypoints.web.address=:80"
- "--entrypoints.websecure.address=:443"



Zabbix Server

zabbix-server:

image: zabbix/zabbix-server-mysql:centos-5.0-latest

env_file:

- ./envs/zabbix-server/common.env

networks:

- "monitoring-network"

volumes:

- /mnt/data-docker/zabbix-server/externalscripts:/usr/lib/zabbix/externalscripts:ro

- /mnt/data-docker/zabbix-server/alertscripts:/usr/lib/zabbix/alertscripts:ro

ports:

- "10051:10051"

deploy:

<<: *template-deploy

labels:

- "traefik.enable=false"



Frontend

zabbix-frontend:

image: zabbix/zabbix-web-nginx-mysql:alpine-5.0.1

env_file:

- ./envs/zabbix-frontend/common.env

networks:

- "monitoring-network"

deploy:

<<: *template-deploy

replicas: 5

labels:

- "traefik.enable=true"
- "traefik.http.routers.zabbix-frontend.tls=true"
- "traefik.http.routers.zabbix-frontend.rule=Host(`frontend.domain`)"
- "traefik.http.routers.zabbix-frontend.entrypoints=web"
- "traefik.http.routers.zabbix-frontend.entrypoints=websecure"
- "traefik.http.services.zabbix-frontend.loadbalancer.server.port=8080"



Deploy

- Connect to a host docker
- Enter the NFS directory
- Deploy the services manually

```
# docker stack deploy -c docker-compose.yaml zabbix
```



NOTES

Important points

- Docker Image: For zabbix server and zabbix proxy is not enough;
 - You can need additional installations
- Networks: Be careful, network conflicts will happen
 - Change the default networks:
 - Overlay
 - Docker bridge



Custom image

```
ARG ZABBIX_BASE=centos
ARG ZABBIX_VERSION=5.0.3
FROM zabbix/zabbix-proxy-sqlite3:${ZABBIX_BASE}-${ZABBIX_VERSION}

ENV ORACLE_HOME=/usr/lib/oracle/12.2/client64
ENV LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib/oracle/12.2/client64/lib
ENV PATH=$PATH:/usr/lib/oracle/12.2/client64/lib

# Install ODBC
COPY ./drivers-oracle-12.2.0.1.0 /root/
COPY odbc.sh /root
RUN chmod +x /root/odbc.sh && \
/root/odbc.sh
```



Custom image

```
# Install Python3
```

```
COPY requirements.txt /requirements.txt
```

```
WORKDIR /
```

```
RUN yum install -y epel-release && \
```

```
    yum search python3 && \
```

```
    yum install -y python36 python36-pip && \
```

```
    python3 -m pip install -r requirements.txt
```

```
# Install SNMP
```

```
RUN yum install -y net-snmp-utils net-snmp wget vim telnet traceroute
```



Custom image

- Build manually;
- Send to registry.



Manual actions

- Deploy manually;
- Build manually.

We can improve this!



Gitlab CI/CD

- Create a repository for each component
 - Zabbix Server
 - Frontend
 - Zabbix Proxy
- Enable Pipelines
- Create a file called **.gitlab-ci.yml**



.gitlab-ci.yml

```
1 variables:
2   CI_REGISTRY: docker.           .com
3   ZABBIX_VERSION_PROD: 5.0.3
4
5 build_dev:
6   variables:
7     BUILD_VER: $ZABBIX_VERSION_PROD
8   image: docker-hub.arti         /docker:19
9   stage: build
10  services:
11   - docker-hub.art              /docker:dind
12  before_script:
13   - apk add --update curl && rm -rf /var/cache/apk/*
14   - docker login --username "$CI_REGISTRY_USER" -p "$CI_REGISTRY_PASSWORD" $CI_REGISTRY
15  script:
16   - >
17     docker build --build-arg ZABBIX_VERSION=$BUILD_VER
18     --build-arg BUILD_DATE=`date -u +"%Y-%m-%dT%H:%M:%SZ"`
19     --build-arg BUILD_TAG=$CI_COMMIT_REF_SLUG
20     --pull
21     --tag "<your_registry>/zabbix/$CI_PROJECT_NAME:$CI_COMMIT_REF_SLUG" .
22   - docker push "<your_registry>/zabbix/$CI_PROJECT_NAME:$CI_COMMIT_REF_SLUG"
23  only:
24   refs: [ prod ]
25   changes:
26   - Dockerfile
27   - externalscripts/**/*
28   - .gitlab-ci.yml
```



.gitlab-ci.yml

```
30 deploy_zabbixproxy:
31   stage: deploy
32   image: docker:19
33   before_script:
34     - export ZBX_ENVIRONMENT=prod
35     - docker login -u "$CI_REGISTRY_USER" -p "$CI_REGISTRY_PASSWORD" $CI_REGISTRY
36   script:
37     - docker stack deploy --with-registry-auth --compose-file=docker-compose.yml zabbix_${CI_COMMIT_REF_SLUG}
38   tags:
39     - runner-dc-1
40   only:
41     - prod
42   when:
43     - manual
```



Benefits

- If any Zabbix component stops, swarm will automatically start a new service / container;
- We don't need to connect to the terminal to start the environment.
- Simple deployment
- Simple administration.



THANK YOU!

Robert Silva - robert.silva@jlcp.com.br

Tech Leader at JLCP

Zabbix Trainer and Zabbix Expert

