

Velociraptor

<https://docs.velociraptor.app/docs/deployment/quickstart/>

Basic workflow

```
docker image load -i ./ir-velociraptor.tar
vim config/server.config.yaml
# add server ip
docker compose up

docker exec -it ir-velociraptor /opt/velociraptor/linux/velociraptor --
config ./config/server.config.yaml user add admin --role administrator

#velociraptor config client --org="root" --config
/velociraptor/config/server.config.yaml
docker exec -it ir-velociraptor /opt/velociraptor/linux/velociraptor config
client --org="root" --config /velociraptor/config/server.config.yaml >
client.config.yaml

#test
./velociraptor --config client.config.yaml client -v
velociraptor.exe --config client.config.yaml client -v

# run plaso
docker run -it --rm --entrypoint=/bin/bash -v ./:/data log2timeline/plaso

#
docker exec -it ir-velociraptor /opt/velociraptor/linux/velociraptor config
repack --msi /velociraptor/msi/#.msi client.config.yaml
/velociraptor/msi/velociraptor-repack.msi

docker image load -i ./splunk.tar
docker compose up
```

```
sed -i -E 's|(- https://)[0-9.]+(:[0-9]+/)|\1<ip>\2|' data/config.yaml
```

How it works:

-E: Enables extended regular expressions (so we can use groups like ()).

(- https://): Group 1 (\1), matches the prefix.

`[0-9.]+`: Matches any combination of numbers and periods (the IP address).

`(:[0-9]+)/`: Group 2 (`\2`), matches the port and trailing slash (e.g., `:20000/`).

`\1192.168.1.50\2`: Rebuilds the line using the original prefix, your new IP, and the original port.

`docker-compose.yml`

```
services:
  velociraptor:
    image: ir-velociraptor
    container_name: ir-velociraptor
    restart: always
    ports:
      - "20000:20000" # Port für die Agents (Clients)
      - "20001:20001" # Port für die Web-Oberfläche (GUI)
    volumes:
      - ./config:/velociraptor/config
      - ./logs:/velociraptor/logs
      - ./store:/velociraptor/store
      - ./clients:/velociraptor/clients
```