

```
version: "3.5"

networks:
  net:
    driver: overlay
    attachable: true

volumes:
  prometheus: {}
  grafana: {}
  alertmanager: {}
  postgres: {}

configs:
  caddy_config:
    file: ./caddy/Caddyfile
  dockerd_config:
    file: ./dockerd-exporter/Caddyfile
  node_rules:
    file: ./prometheus/rules/swarm_node.rules.yml
  task_rules:
    file: ./prometheus/rules/swarm_task.rules.yml

services:
  pg_prometheus:
    image: timescale/pg_prometheus:master
    networks:
      - net
    ports:
      - "5432:5432"
    command: ["postgres", "-csynchronous_commit=off"]
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.role == manager

  prometheus_postgresql_adapter:
    image: timescale/prometheus-postgresql-adapter:master
    networks:
      - net
    ports:
      - "9201:9201"
    command: ["-pg-host=pg_prometheus", "-pg-prometheus-log-samples"]
    deploy:
      mode: replicated
      replicas: 1
      placement:
        constraints:
          - node.role == manager

  dockerd-exporter:
    image: stefanprodan/caddy
    networks:
      - net
    environment:
      - DOCKER_GWBRIDGE_IP=172.18.0.1
    configs:
      - source: dockerd_config
        target: /etc/caddy/Caddyfile
    deploy:
```

```
    mode: global
    resources:
      limits:
        memory: 128M
      reservations:
        memory: 64M

cadvisor:
  image: google/cadvisor
  networks:
    - net
  command: -logtostderr -docker_only
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock:ro
    - /:/rootfs:ro
    - /var/run:/var/run
    - /sys:/sys:ro
    - /var/lib/docker:/var/lib/docker:ro
  deploy:
    mode: global
    resources:
      limits:
        memory: 128M
      reservations:
        memory: 64M

grafana:
  image: stefanprodan/swarmprom-grafana:4.6.3
  networks:
    - net
  environment:
    - GF_SECURITY_ADMIN_USER=${ADMIN_USER:-admin}
    - GF_SECURITY_ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
    - GF_USERS_ALLOW_SIGN_UP=false
    #- GF_SERVER_ROOT_URL=${GF_SERVER_ROOT_URL:-localhost}
    #- GF_SMTP_ENABLED=${GF_SMTP_ENABLED:-false}
    #- GF_SMTP_FROM_ADDRESS=${GF_SMTP_FROM_ADDRESS:-grafana@test.com}
    #- GF_SMTP_FROM_NAME=${GF_SMTP_FROM_NAME:-Grafana}
    #- GF_SMTP_HOST=${GF_SMTP_HOST:-smtp:25}
    #- GF_SMTP_USER=${GF_SMTP_USER}
    #- GF_SMTP_PASSWORD=${GF_SMTP_PASSWORD}
  volumes:
    - grafana:/var/lib/grafana
  deploy:
    mode: replicated
    replicas: 1
    placement:
      constraints:
        - node.role == manager
    resources:
      limits:
        memory: 128M
      reservations:
        memory: 64M

alertmanager:
  image: stefanprodan/swarmprom-alertmanager:v0.14.0
  networks:
    - net
  environment:
    - SLACK_URL=${SLACK_URL:-https://hooks.slack.com/services/TOKEN}
    - SLACK_CHANNEL=${SLACK_CHANNEL:-general}
```

```
- SLACK_USER=${SLACK_USER:-alertmanager}
command:
- '--config.file=/etc/alertmanager/alertmanager.yml'
- '--storage.path=/alertmanager'
volumes:
- alertmanager:/alertmanager
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.role == manager
resources:
limits:
memory: 128M
reservations:
memory: 64M

unsee:
image: cloudflare/unsee:v0.8.0
networks:
- net
environment:
- "ALERTMANAGER_URI=default:http://alertmanager:9093"
deploy:
mode: replicated
replicas: 1

node-exporter:
image: stefanprodan/swarmprom-node-exporter:v0.15.2
networks:
- net
environment:
- NODE_ID={{.Node.ID}}
volumes:
- /proc:/host/proc:ro
- /sys:/host/sys:ro
- /:/rootfs:ro
- /etc/hostname:/etc/nodename
command:
- '--path.sysfs=/host/sys'
- '--path.procfs=/host/proc'
- '--collector.textfile.directory=/etc/node-exporter/'
- '--collector.filesystem.ignored-mount-points=^/(sys|proc|dev|host|etc)($$|/)'
# no collectors are explicitly enabled here, because the defaults are just
fine,
# see https://github.com/prometheus/node_exporter
# disable ipvs collector because it barfs the node-exporter logs full with
errors on my centos 7 vm's
- '--no-collector.ipvs'
deploy:
mode: global
resources:
limits:
memory: 128M
reservations:
memory: 64M

prometheus:
#image: stefanprodan/swarmprom-prometheus:v2.2.0-rc.0
image: shambakey1/prometheus:latest
networks:
```

```
- net
command:
- '--config.file=/etc/prometheus/prometheus.yml'
- '--web.console.libraries=/etc/prometheus/console_libraries'
- '--web.console.templates=/etc/prometheus/consoles'
# - '--storage.tsdb.path=/prometheus'
# - '--storage.tsdb.retention=24h'
volumes:
- prometheus:/prometheus
configs:
- source: node_rules
  target: /etc/prometheus/swarm_node.rules.yml
- source: task_rules
  target: /etc/prometheus/swarm_task.rules.yml
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.role == manager
resources:
limits:
memory: 2048M
reservations:
memory: 1024M

caddy:
image: stefanprodan/caddy
ports:
- "3000:3000"
- "9090:9090"
- "9093:9093"
- "9094:9094"
networks:
- net
environment:
- ADMIN_USER=${ADMIN_USER:-admin}
- ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
configs:
- source: caddy_config
  target: /etc/caddy/Caddyfile
deploy:
mode: replicated
replicas: 1
placement:
constraints:
- node.role == manager
resources:
limits:
memory: 128M
reservations:
memory: 64M
healthcheck:
test: ["CMD", "curl", "-f", "http://localhost:3000"]
interval: 5s
timeout: 1s
retries: 5
```