Lazytainer

With Lazytainer, you can have containers start up when you need them and then shut down after a specified amount of time after you're done using them!

- Docker Compose Example 1 (Lazy)
- Docker Compose Example 2 (Lazy2)
- Docker Compose Example 3 (Lazy3)

Docker Compose Example 1 (Lazy)

```
version: "3"
services:
  lazytainer:
    container_name: lazytainer
    image: ghcr.io/vmorganp/lazytainer:master
    environment:
      - PORT=8081      # comma separated list of ports...or just the one
      - LABEL=lazytainer     # value of lazytainer.marker for other containers that lazytainer checks
      - TIMEOUT=15           # OPTIONAL number of seconds to let container idle
      - MINPACKETTHRESH=10   # OPTIONAL number of packets that must be recieved to keepalive/start container
      - POLLRATE=1           # OPTIONAL number of seconds to sleep between polls
      - VERBOSE=true         # probably set this to false unless you're debugging or doing the initial demo
    ports:
      - 8081:8080
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro


  youtube:
    container_name: youtube
    image: modenaf360/youtube-dl-nas
    environment:
      - MY_ID=dbtech
      - MY_PW=password
    volumes:
      - /home/docker/YouTube:/downfolder
    network_mode: service:lazytainer
    depends_on:
      - lazytainer # wait for lazytainer to start before starting
    labels:
```

```
      - "lazytainer.marker=lazytainer" # required label to make it work
```

Docker Compose Example 2 (Lazy2)

```
version: "3"
services:
  lazytainer2:
    container_name: lazytainer2
    image: ghcr.io/vmorganp/lazytainer:master
    environment:
      - PORT=3000            # comma separated list of ports...or just the one
      - LABEL=lazytainer2    # value of lazytainer.marker for other containers that lazytainer
checks
      - TIMEOUT=15           # OPTIONAL number of seconds to let container idle
      - MINPACKETTHRESH=10   # OPTIONAL number of packets that must be recieved to
keepalive/start container
      - POLLRATE=1           # OPTIONAL number of seconds to sleep between polls
      - VERBOSE=true         # probably set this to false unless you're debugging or doing the
initial demo
    ports:
      - 3000:3000
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
  taisun:
    image: linuxserver/taisun
    container_name: taisun2
    network_mode: service:lazytainer2
    depends_on:
      - lazytainer2
    labels:
      - "lazytainer.marker=lazytainer2"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
```

Docker Compose Example 3 (Lazy3)

```
version: "3"
services:
  lazytainer3:
    container_name: lazytainer3
    image: ghcr.io/vmorganp/lazytainer:master
    environment:
      - PORT=8080, 3000          # comma separated list of ports...or just the one
      - LABEL=lazytainer3     # value of lazytainer.marker for other containers that
lazytainer checks
      - TIMEOUT=15           # OPTIONAL number of seconds to let container idle
      - MINPACKETTHRESH=10   # OPTIONAL number of packets that must be recieved to
keepalive/start container
      - POLLRATE=1           # OPTIONAL number of seconds to sleep between polls
      - VERBOSE=true         # probably set this to false unless you're debugging or doing the
initial demo
    ports:
      - 8080:8080
      - 3000:3000
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro


  youtube:
    container_name: youtube3
    image: modenaf360/youtube-dl-nas
    environment:
      - MY_ID=dbtech
      - MY_PW=password
    volumes:
      - /home/docker/YouTube:/downfolder
    network_mode: service:lazytainer3
    depends_on:
      - lazytainer3 # wait for lazytainer to start before starting
```

```
    labels:
      - "lazytainer.marker=lazytainer3" # required label to make it work


  taisun:
    image: linuxserver/taisun
    container_name: taisun3
    restart: unless-stopped
    network_mode: service:lazytainer3
    depends_on:
      - lazytainer3
    labels:
      - "lazytainer.marker=lazytainer3"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
```