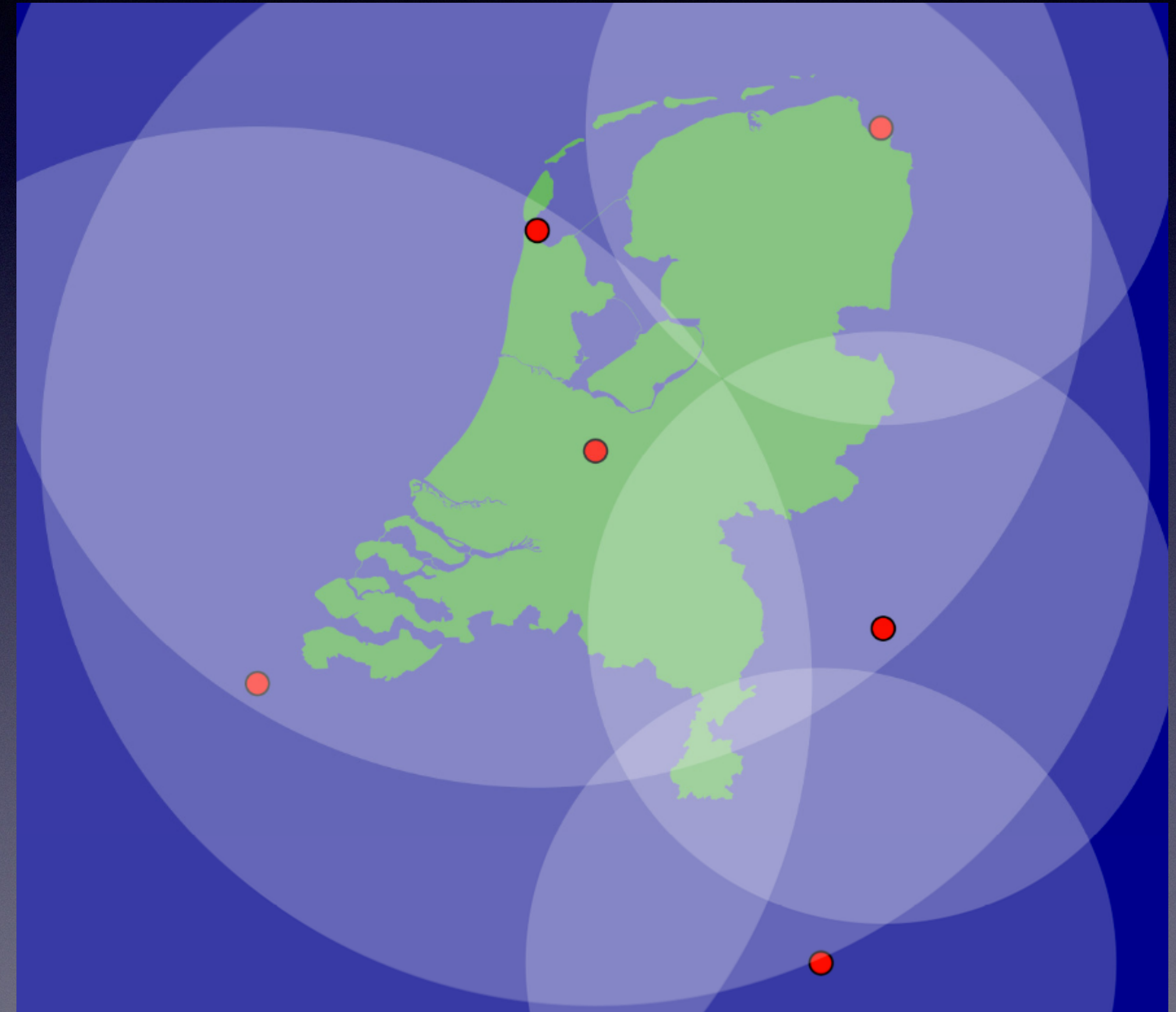


Simple Prefect setup: handy task runner for data projects

Reinout van Rees
Nelen & Schuurmans

Data projects

- Nelen&Schuurmans (Utrecht), *data driven water management*
- 40x water experts, 20x python&co
- Fancy: loads of data, numpy, gdal, postgresql, pandas
- Not so fancy: loads of scripts, lots of downloading, many long-running calculations



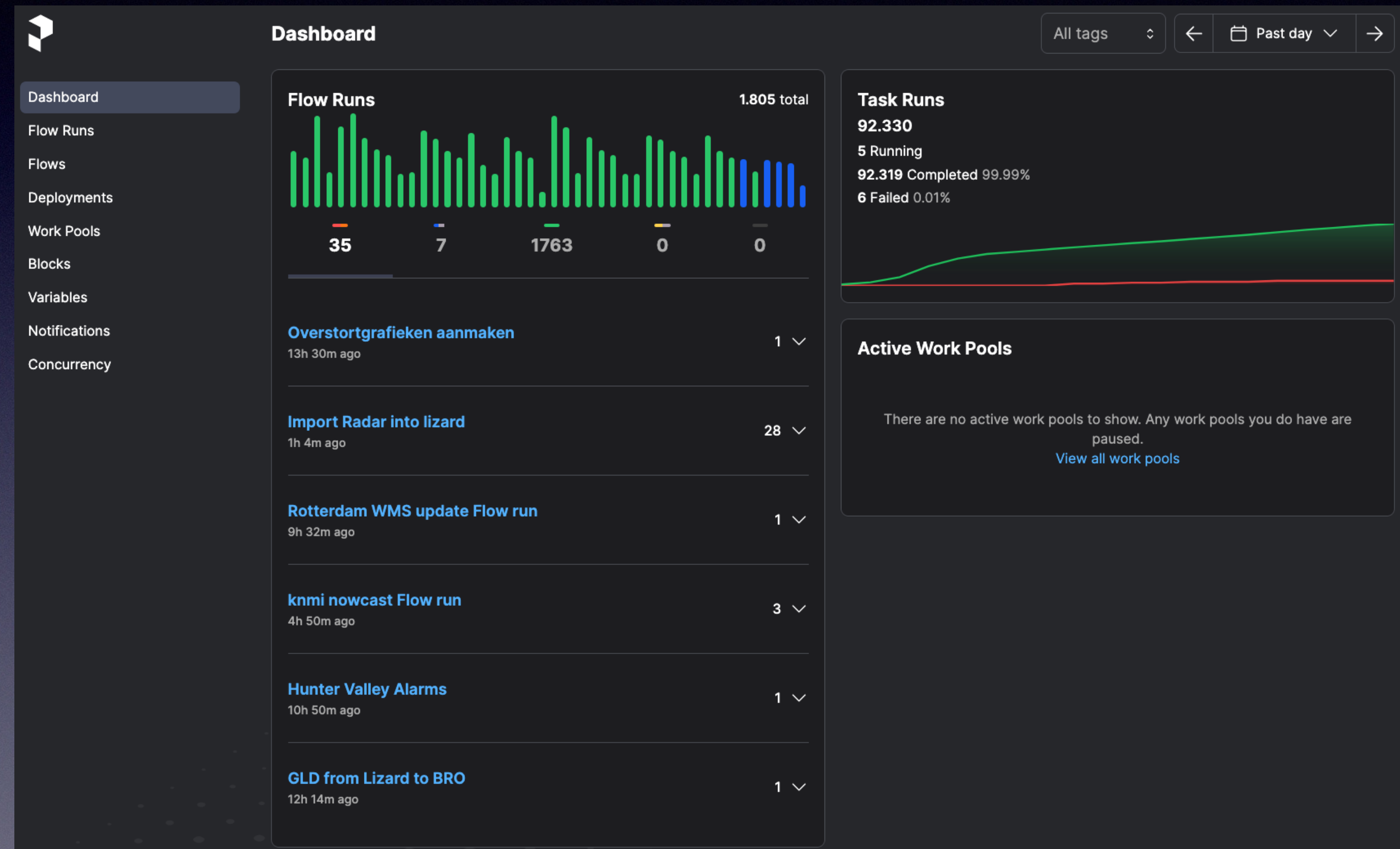
The old way

- Linux server
- First global python env, bit of virtualenv
- Later docker-compose
- Cronjob
- Problem: failures and logfiles



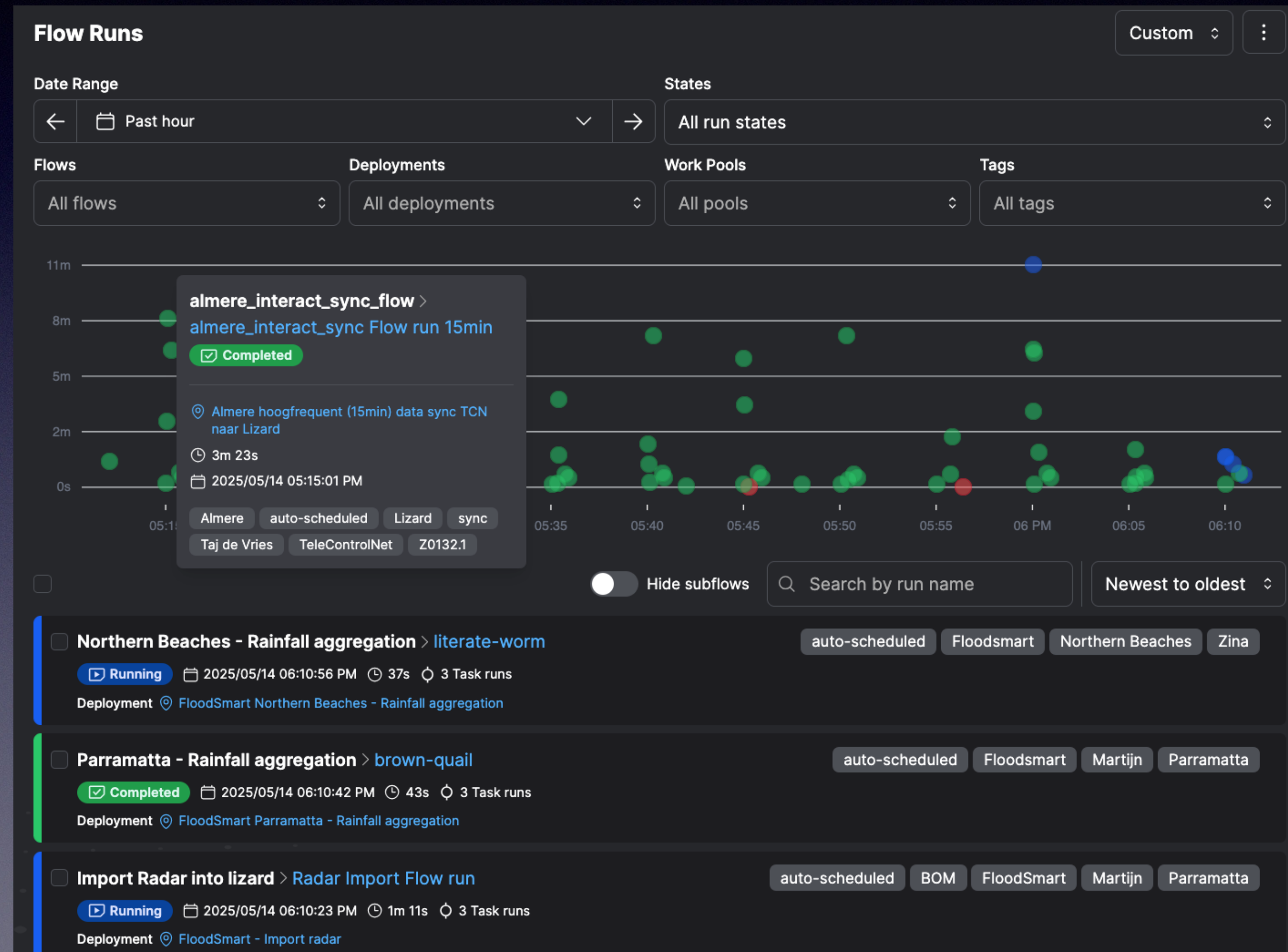
A solution: prefect

- <https://www.prefect.io/>
- Task runner
- Pure python
- Central server: dashboard + api
- Workflows



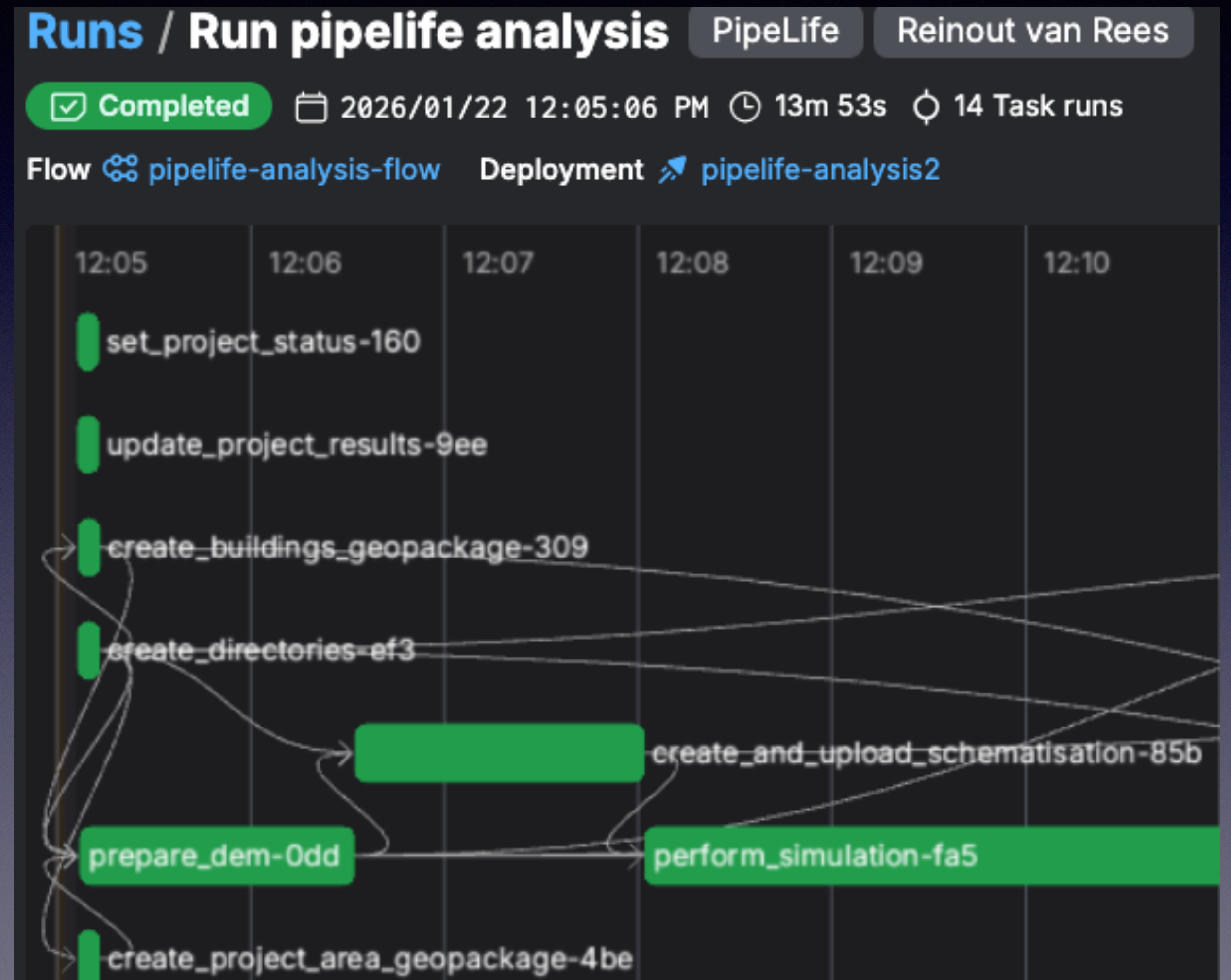
Prefect: the server

- Dashboard
- API
- Overview of workflows
- Start workflow runs
- Receive workflow logs/status
 - → live



Workflows: tasks+flows

- Task: function decorated with @task, one unit of whatever you want to track
- Parameters like “retries”
- Flow: function decorated with @flow, calls tasks (and can do other things)
- A flow is one complete sets of tasks
- `logger = get_run_logger()`

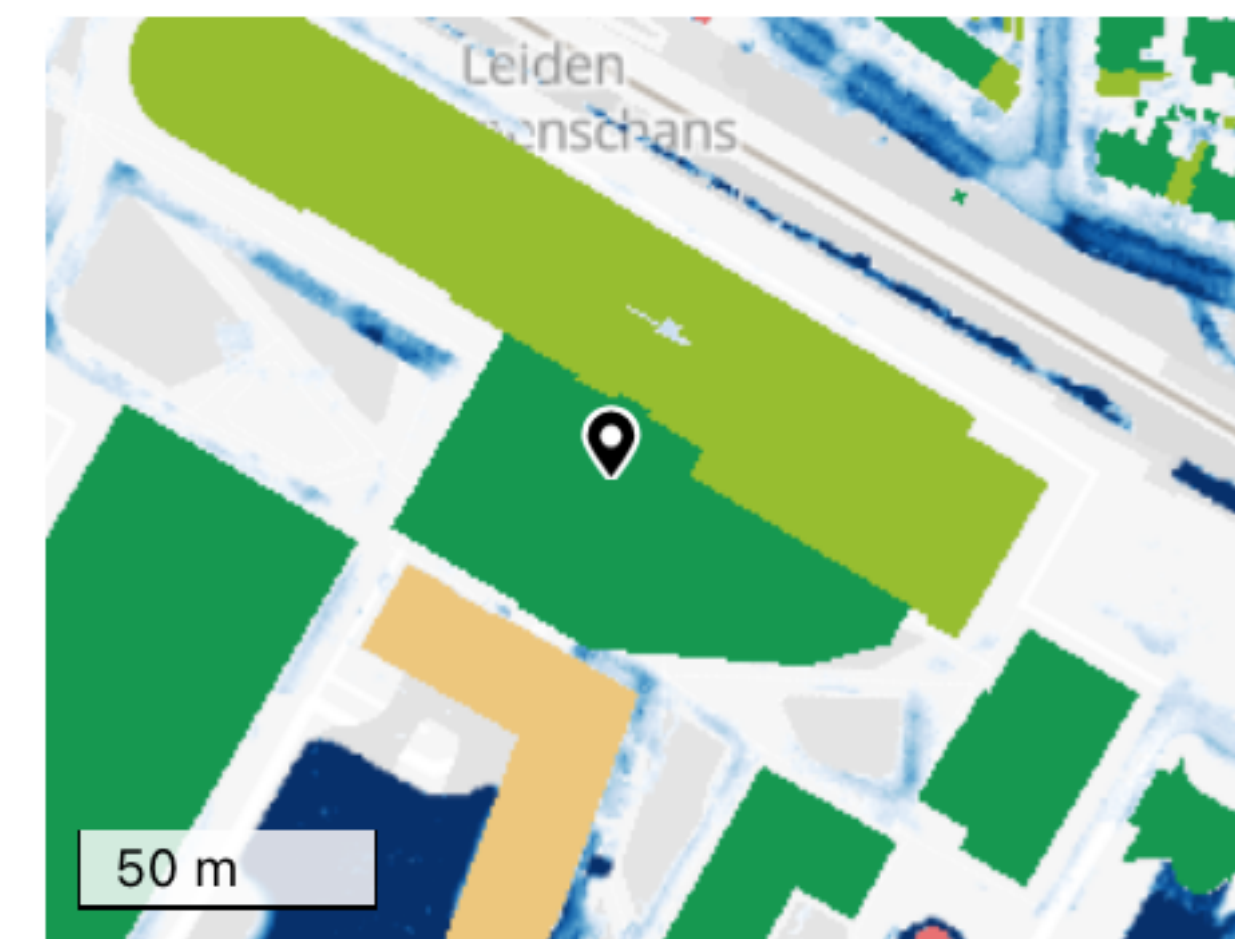


PROJECT

Project	Python Leiden
Datum	22 jan. 2026, 12:12
Naam	Reinout
Email	reinout@vanrees.org
Bui	70 (mm/uur)

LOCATIE

Locatie	52.1458, 4.4933
Postcode	2321 KS
Adres	Bètaplein 30
Plaatsnaam	Fruitbuurt
Gebouwfunctie	Bijeenkomst



BlueLabel Waterdiepte

A	0 cm
B	< 10 cm
C	10 - 15 cm
D	15 - 20 cm
E	> 20 cm

Wateroverlast



📍 Locatie pand



RESULTAAT

De kaarten geven een indicatie van de verwachte waterhoogtes bij een regenbui van 70 mm in 1 uur. Op de kaarten zijn ook de panden weergegeven die risico lopen op wateroverlast. Het pand dat is aangeduid op de kaart krijgt BlueLabel A.

Perfect server deploy

- “prefect server start”
- Docker image available
- Problem: authentication
- Simple solution: nginx with basic auth...
- Note: prefect plans to add authentication

```
server:
  image: prefecthq/prefect:2.20-python3.12
  restart: unless-stopped
  entrypoint: ["/opt/prefect/entrypoint.sh", "prefect", "server", "start"]
  ports:
    - 4200
  logging:
    options:
      max-size: "10m"
      max-file: "3"
  environment:
    PREFECT_SERVER_API_HOST: "0.0.0.0"
    PREFECT_API_URL: "https://prefect2.lizard.net/api"
    PREFECT_UI_URL: "https://prefect2.lizard.net/api"
    PREFECT_API_DATABASE_CONNECTION_URL: "postgresql+asyncpg://prefect:kik
```

```
server {
    listen 80;
    set $upstream1 server;
    server_name prefect2.lizard.net;
    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://$upstream1:4200;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        auth_basic "Vraag Taj of Reinout voor het wachtwoord";
        auth_basic_user_file /etc/nginx/htpasswd;
    }
}
```

Workflow: deployment

- Deployment: tells server how/when to run it
- cron= tells server when to run it automatically
- Without a cron, it “just” awaits a manual or programmatic call
- serve(): runs the deployment(s) and registers itself with the server at PREFECT_API_URL, awaiting commands

```
from flows import radar_gif_flow
from prefect import serve
import config

if __name__ == "__main__":
    config_deployment = config.Config()

    parrammata_gif_deployment = radar_gif_flow.to_deployment(
        name="HVFMS&Parramatta - GIF update",
        cron="*/5 * * * *",
        parameters={"config": config_deployment.model_dump()},
        description="Code found at: https://github.com/nens/prefect",
        tags=["Boran", "Eshter", "Parrammata", "Townsville", "Se"]
    )

    serve(parrammata_gif_deployment) # type: ignore # It is pos
```

Auto-update docker setup

- Github repo per workflow (made from a "cookiecutter" template)
- Github action to build docker image
- A docker-compose.yml with those images
- watchtower
- No need for server access for most

```
services:

# Watchtower checks for new containers every five minutes and updates them
# when available.
watchtower:
  image: containrrr/watchtower
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - $HOME/.docker/config.json:/config.json
  restart: unless-stopped
  command: --interval 300 --include-restarting --include-stopped --revive-stopped

prefect-saf-kpi-calculation:
  image: ghcr.io/nens/prefect-saf-kpi-calculation:main
  env_file:
    - prefect2.env
  volumes:
    - /mnt/lizard6-media-projecten:/code/output/lizard6-media-projecten
  restart: unless-stopped

prefect-nasa-gpm-processor:
  image: ghcr.io/nens/prefect-nasa-gpm-processor:main
  env_file:
    - prefect2.env
  restart: unless-stopped
```

Closing comments

- <https://github.com/nens/cookiecutter-prefect-template>
- README has server setup explanation
- Database cleanup (runs+logs)...
- `from prefect.deployments import run_deployment`

