Platys – Provisioning of modern data platforms simplified!

Guido Schmutz, Ulises Fasoli

Agenda

trivadis

- Motivation for Platys
- Installing Platys
- Platys in Action

trivadis

Motivation for Platys?

gschmutz

Requirements for a Platform for Lab, PoC and PoV projects

trivadis

For Labs, PoC and PoV projects as well as trainings we wanted a platform

1. which is easy to deploy
2. supports modular, extensible and configurable set of services
3. where new services can be added quickly and tested as part of an existing platform (set of services)
4. where no infrastructure specialist is needed for provisioning
5. lightweight, low footprint (resources needed)
6. simple, lightweight transportation
7. runs on-premises and in the cloud

Platform for Lab, PoC and PoV projects

trivadis

		Bare Metal	VM	Cloud PaaS	Docker & Docker Compose
1	Easy Deployment	+	++	+++	+++
2	Modular, extensible	++	++	++	+++
3	Add new services quickly	+	+	++ (1)	+++
4	No infrastructre specialist needed	+	++	+++	+++
5	Lightweight footprint	+	++	+	+++
6	Simple, lightweight transportation	-	+	++ (2)	+++
7	On prem & cloud	+	+++	++	+++ (3)

1) If exists as PaaS
2) Within same cloud provider
3) Runs anywhere Docker & Docker Compose runs

What is Docker, what is Docker Compose?

trivadis

Docker – running a single service

```
docker run -d -p 18630:18630 -e SDC_CONF_MONITOR_MEMORY:true -v sdc-data:/data
--name streamsets-dc streamsets/datacollector:3.15.0
```

Docker Compose – running a set of services belonging together

```
docker-compose up -d
```

```
streamsets-1:
    image: confluentinc/cp-enterprise-kafka:5.5.0
    ports:
      - 18630:18630
    environment:
      SDC_CONF_MONITOR_MEMORY: 'true'
      SDC_CONF_PIPELINE_MAX_RUNNERS_COUNT: 50
    volumes:
      - ./sdc-data:/data
 kafka-1:
    image:
    depends_on:
      - zookeeper-1
    ports:
      - 9092:9092
    ...
```

`platys` – Trivadis Platform in a Box

trivadis

- open-source toolset, developed by the Trivadis Platform Factory
- help provisioning Modern Data Platforms based on Docker and Docker Compose
- its main use is for small-scale Data Lab projects, Proof-of-Concepts (PoC) or Proof-of-value (PoV) projects as well as trainings

- See here for how to install `platys`
- https://github.com/TrivadisPF/platys

Trivadis `platys` – Platform in a Box

trivadis

Platform Stack

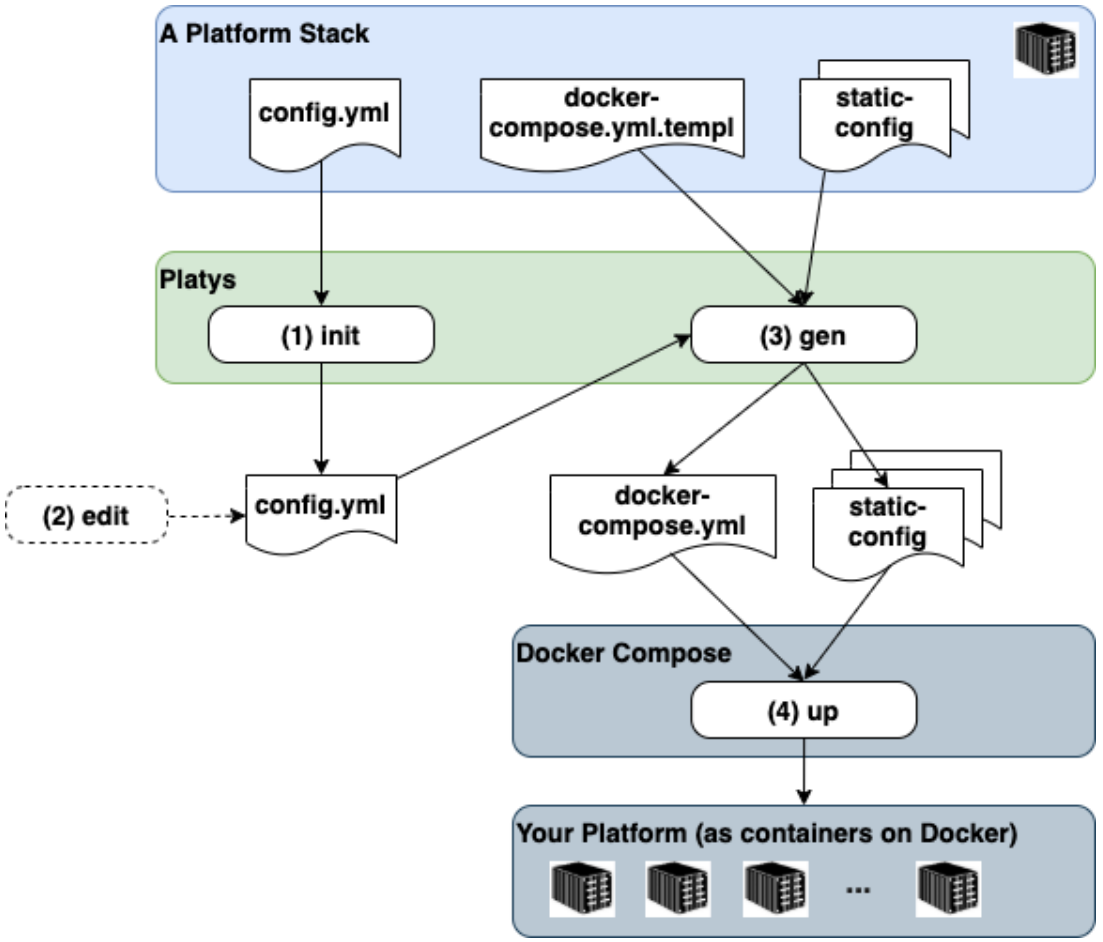
- A given set of services to choose from
- A generator for generating a set of these services as a Platform

Platform

- An instance of a set of services generated based on a Platform Stack
- Fully runnable platform based on Docker Compose and Docker

Platys

- The CLI tool to control the Platform Stack



gschmutz

trivadis

Installing Platys

gschmutz

Trivadis `platys` – How to install? (I)

trivadis

Platys runs on **Liunx**, **Mac OS-X** and **Windows**

Installation is comparable to installing docker-compose, on Ubuntu perform the following steps

1. Download the binary

```
sudo curl -L
"https://github.com/TrivadisPF/platys/releases/download/2.2.0/p
latys.tar.gz" -o /tmp/platys.tar.gz
```

2. Untar and move it to /usr/local/bin

```
tar zvxf /tmp/platys.tar.gz
sudo mv platys.dist/ /usr/local/lib/
sudo chown -R root:root /usr/local/lib/platys.dist/
sudo rm /tmp/platys.tar.gz
```

Trivadis `platys` – How to install? (II)

trivadis

3. Download the binary

```
sudo ln -s /usr/local/lib/platys.dist/platys /usr/local/bin/platys
```

4. Use the `--version` option to check that the generator has been installed successfully

```
$ platys --version
Platys - Trivadis Platform in a Box - v2.2.0
```

trivadis

Platys in Action

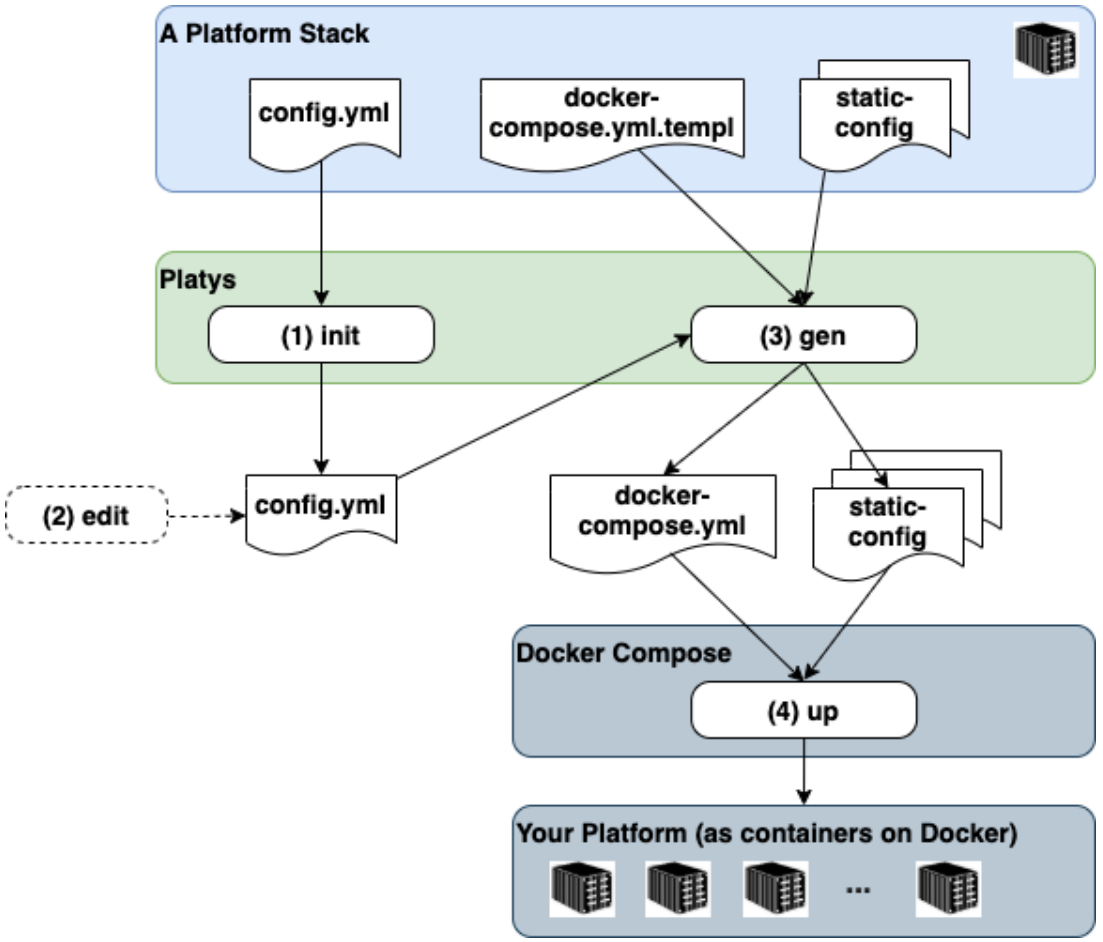
gschmutz

Trivadis `platys` – Platform in a Box

trivadis

Steps for working with `platys`

1. Init a platform based on a given platform stack
2. Edit the config.yml files to enable services needed
3. Generate the docker-compose platform
4. Start the platform

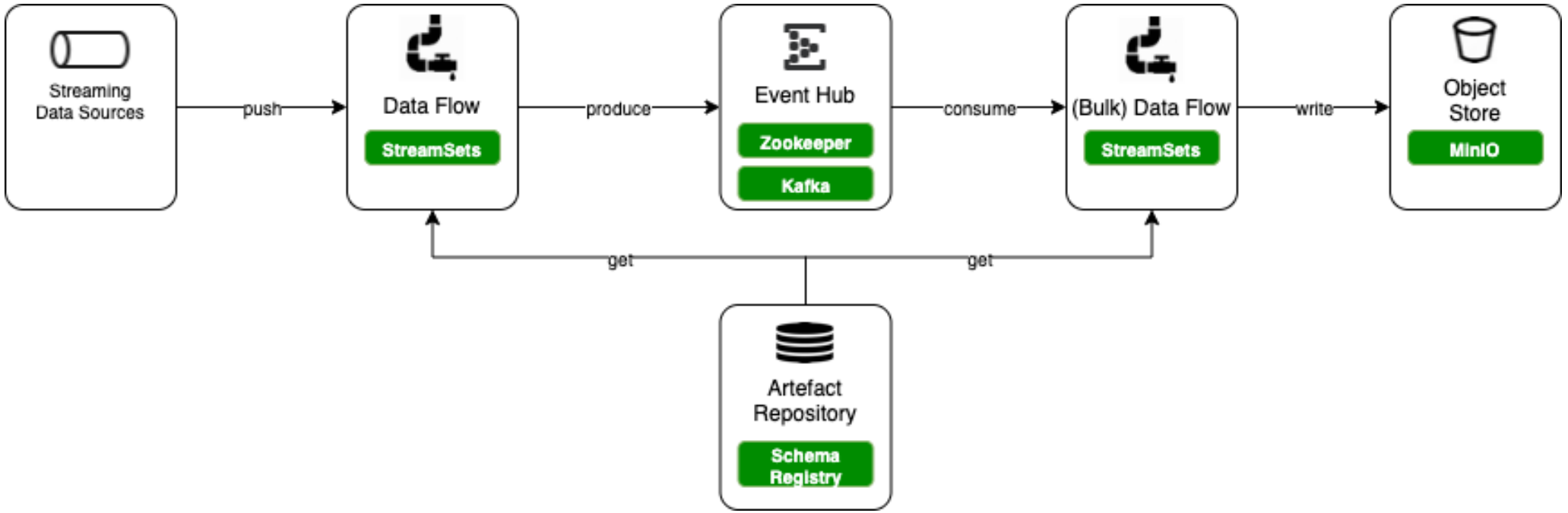


gschmutz

Trivadis platys – How does it work? (I)

trivadis

Let’s see Platys in Action by generating a simple Platform with the following service



Trivadis `platys` – How does it work? (II)

trivadis

1. Create a folder as your platform working directory

```
mkdir platys-demo
dd platys-demo
```

2. Initialize the platform using the `platys init` command

```
platys init -n platys-demo-platform
     -sn trivadis/platys-modern-data-platform -sv 1.5.0
```

3. A `config.yml` file is created locally, holding all the possible options of the modern-data-platform 1.5.0 stack. See here for a documentation of the various options.

Trivadis `platys` – How does it work? (III)

trivadis

4. Use an editor to set the needed services to `true`.

```
# Default values for the generator
# this file can be used as a template for a custom configuration
# or to know about the different variables available for the generator
    platys:
       platform-name: ‘platys-demo'
       stack-image-name: 'trivadis/platys-modern-data-platform'
       stack-image-version: ‘1.5.0’
       structure: ‘flat’

    # ===== Apache Zookeeper ========
    KAFKA_enable: true
    # one of enterprise, community
    KAFKA_edition: 'community'
    KAFKA_volume_map_data: false
    KAFKA_broker_nodes: 3
    KAFKA_delete_topic_enable: false
    KAFKA_auto_create_topics_enable: false
```

Trivadis `platys` – How does it work? (IV)

trivadis

5. Run `platys gen` to generate your platform

```
platys gen
```

6. All the necessary artefacts for the Docker Compose platform have been generated and are ready to be started
7. Start the platform using `docker-compose up`

```
docker-compose up -d
```

Structure: Flat vs. Subfolder

trivadis

Flat

```
platys-demo
├── config.yml
├── docker-compose.yml
├── conf
├── container-volume
├── data-transfer
├── init
├── plugins
└── scripts
```

Subfolder

```
platys-demo
├── config.yml
└── my-platform
    ├── conf
    ├── container-volume
    ├── data-transfer
    ├── init
    ├── plugins
    ├── scripts
    └── docker-compose.yml
```

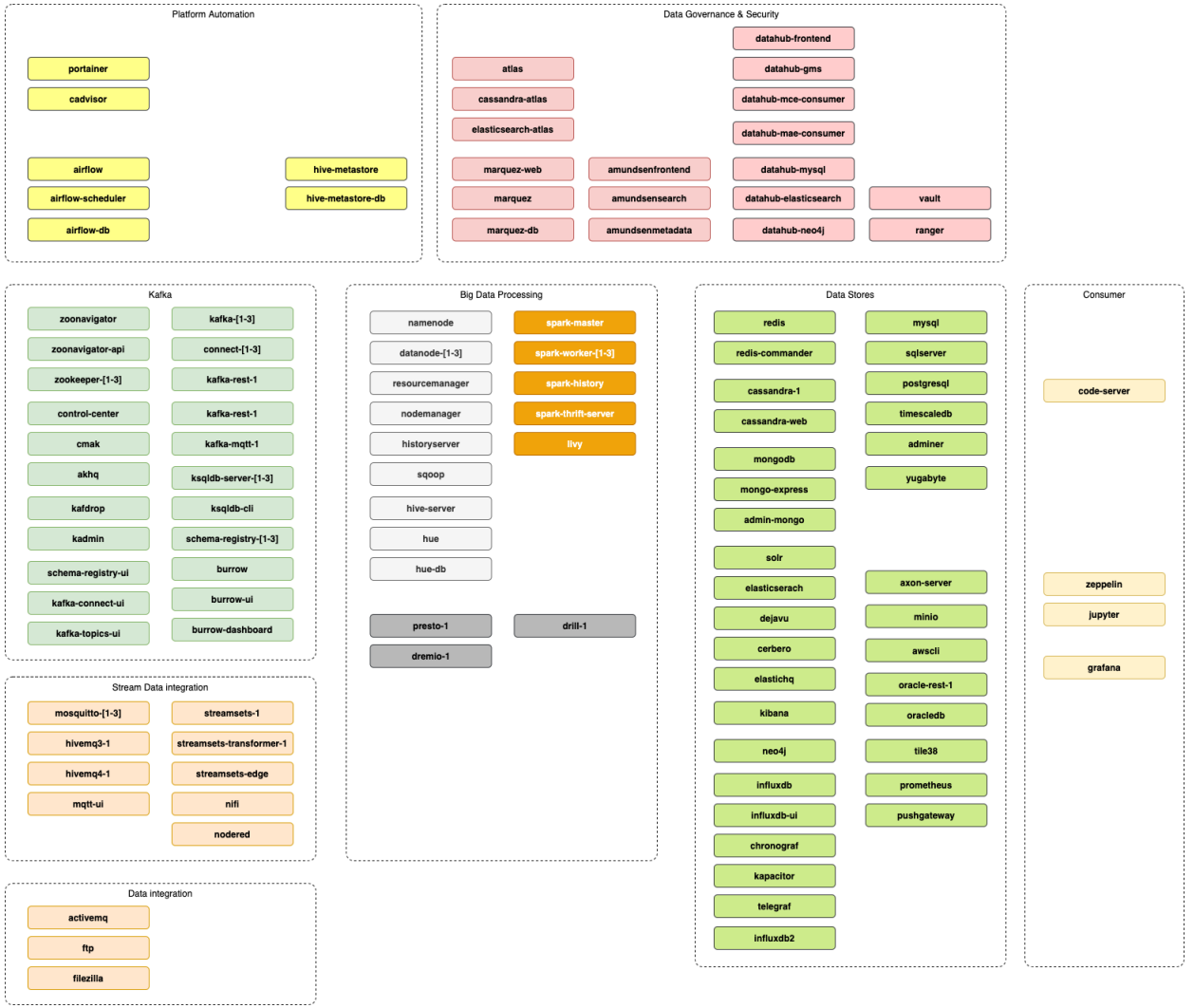
gschmutz

modern-data-platform V1.5

trivadis

These are the services the curent platform supports

Check the Configuration documentation for how to use



What else?

trivadis

- `docker-compose.override.yml` for adding / re-configuring things not (yet) covered by platys
- `-s` flag for directly creating a valid config.yml
- Cookbooks will be made available showing various aspects
- Multiple platforms in one box ;-) => ARM and x86-64

- Slack channel will come
- We are hiring :-)

Trivadis `platys` – Tips and Tricks

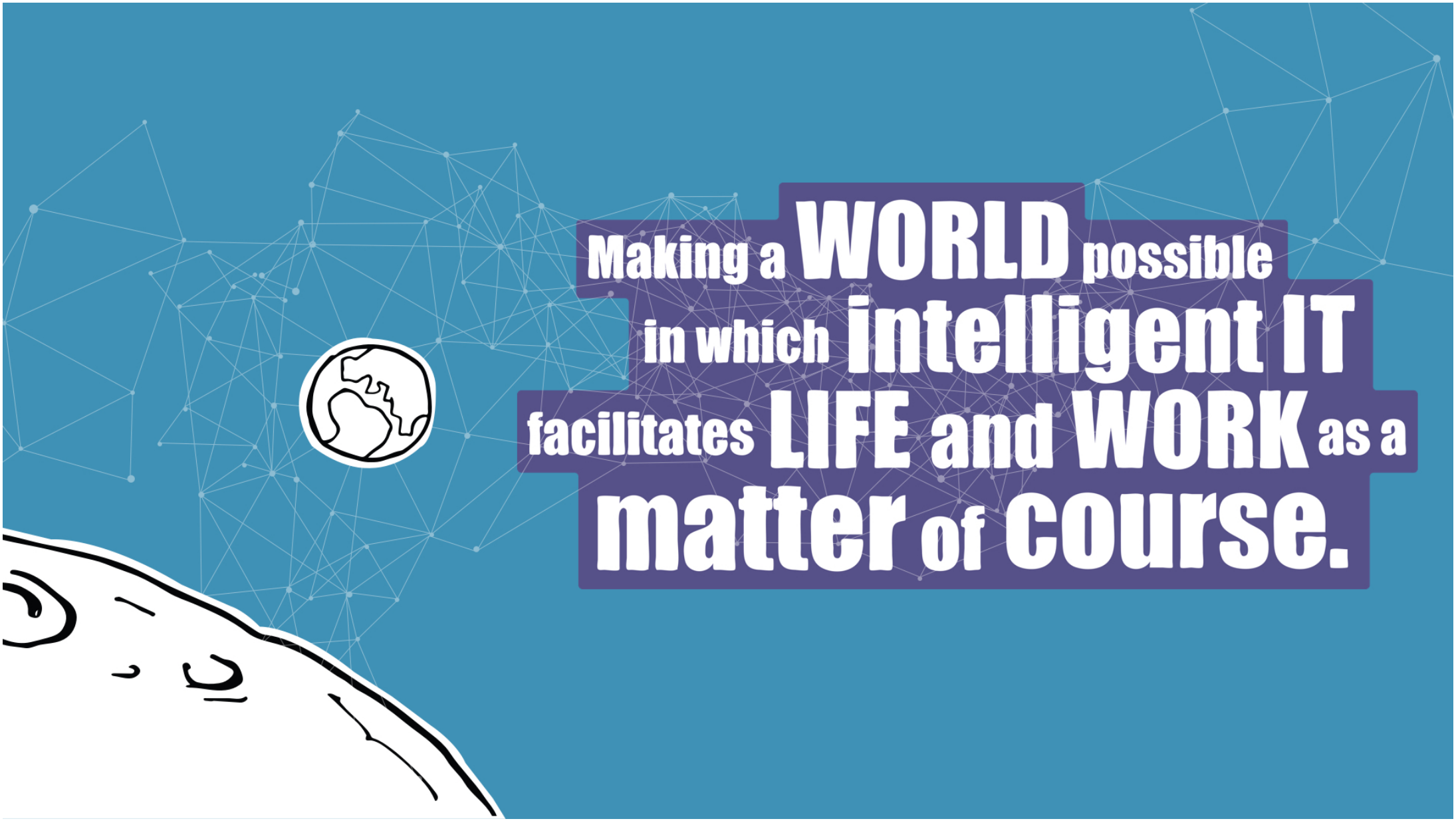
trivadis

- In the init you can also pass a list of services to directly set to true to omit having to edit the `config.yml`

```
platys init -n platys-demo-platform
     -sn trivadis/platys-modern-data-platform -sv 1.5.0
     -f -s KAFKA,SPARK,JUPYTER
```

- You can list the available service of a given platform stack by using the `list_services` command

```
platys list_services -sn trivadis/platys-modern-data-platform
                     -sv 1.5.0
```


Making a WORLD possible
in which intelligent IT
facilitates LIFE and WORK as a
matter of course.