

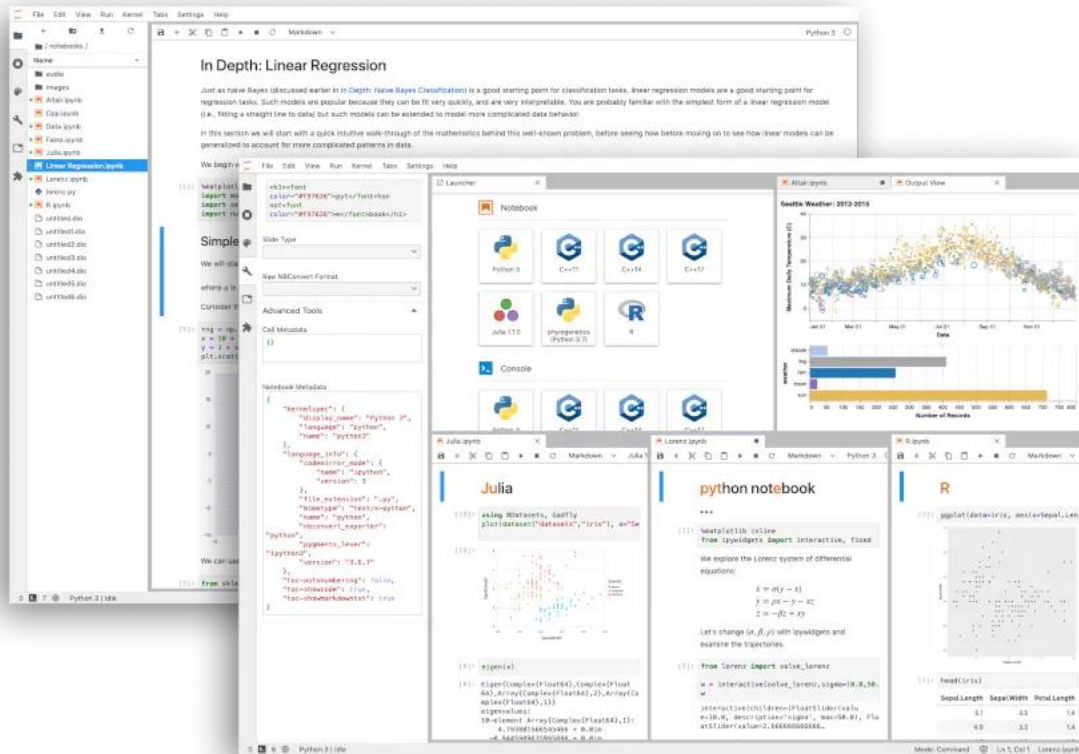
Integrácia a nasadenie softvérových aplikácií 3

Peter Bednár

Príklad vývojového prostredia UUI DataLab

JupyterLab (1)

- Webová aplikácia pre interaktívne výpočty a dátovú analytiku



JupyterLab (2)

- Kombinuje kód, textovú dokumentáciu (Markdown) a dátové vizualizácie v jednom dokumente
- Integrované vývojové prostredie
 - Rozširujúce pluginy
- Podporuje viacero programovacích jazykov (Python, R, Julia)
 - Kód je spúšťaný v samostatných procesoch - *kernel*



JupyterHub

- Umožňuje prístup viacerých používateľov k JupyterLab aplikáciám bežiacim v serverovom prostredí
- Hlavné vlastnosti:
 - Multipoužívateľské prostredie – samostatná aplikácia pre každého používateľa
 - Manažovanie zdrojov (CPU, pamäť, zdieľanie dát)
 - Podpora autentifikácie (OAuth, LDAP, GitHub, lokálne kontá)
 - Škálovateľnosť



Docker Swarm

- Umožňuje spojiť viacero Docker serverov do jedného **virtuálneho klastra**
 - Manažuje kontajnery pre viacero aplikácií, ktoré bežia na viacerých serveroch
- Aplikácie sú spúšťané ako služby, ktoré možno replikovať aj na viacero serverov
 - Škálovateľnosť
- Manažérske a pracovné uzly
- Zabezpečená sieťová komunikácia medzi uzlami (a kontajnermi)



Docker Compose

- Manažovanie prostredí zložených z viacerých služieb/kontajnerov
- Konfigurácia celého prostredia v jednom konfiguračnom súbore
 - `docker-compose.yml` – definuje služby, sieťové prepojenia a dátové úložiská pripojené ku kontajnerom

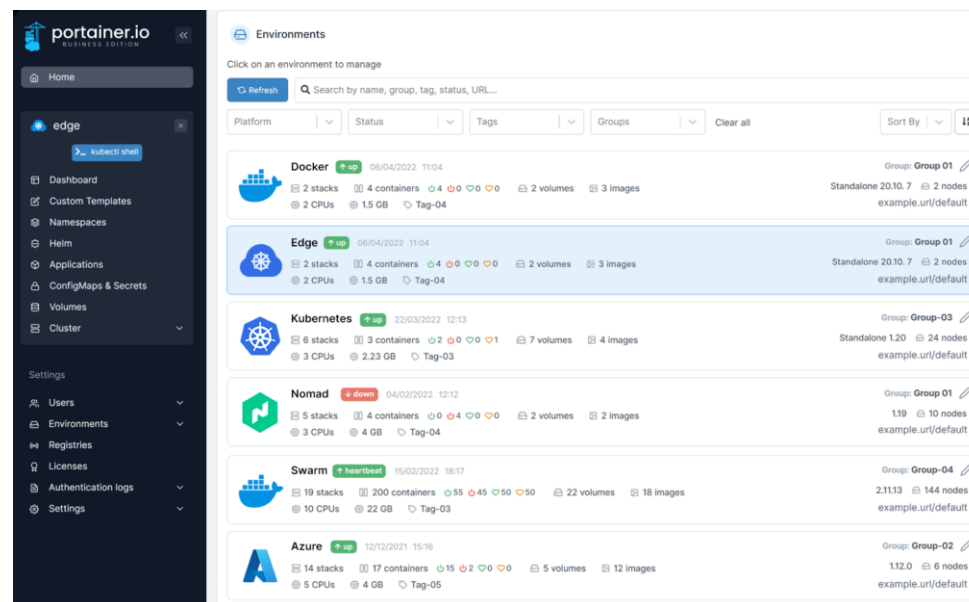
`docker stack deploy datalab`

`docker stack rm datalab`

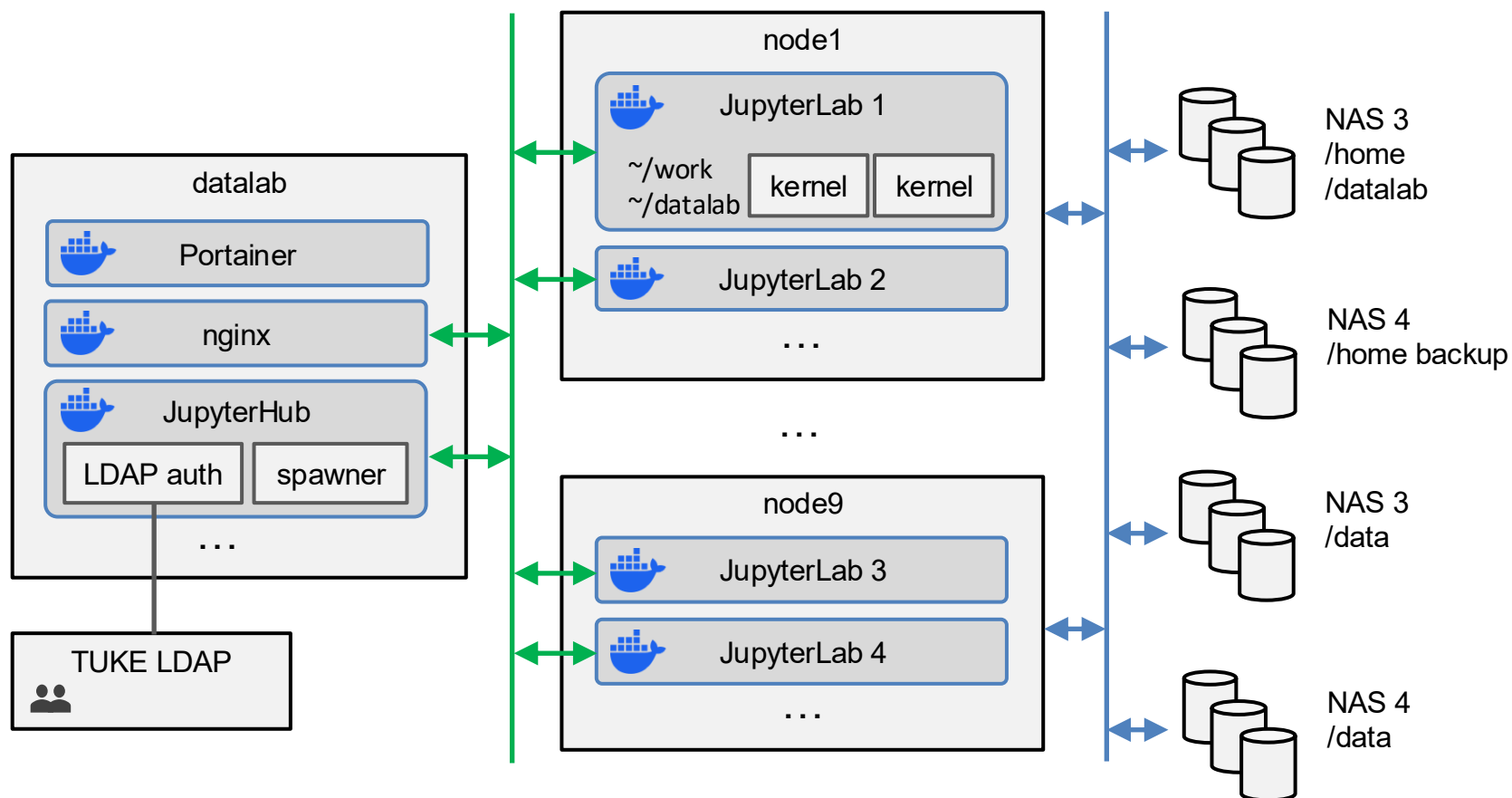


Portainer

- Poskytuje používateľské rozhranie pre manažovanie kontajnerových prostredí
- Docker, Docker Swarm, Kubernetes
- Zjednodušuje spúšťanie, monitorovanie a manažovanie kontajnerov a služieb



UUI DataLab



Reprodukovateľné procesy analýzy dát

Reprodukovateľné dáta + kód + modely

Reprodukovateľné procesy analýzy dát (1)

- Výber a integrácia dát
 - Dátové snímky
 - Monotónne dátové úložiská
- Predspracovanie
 - Formátovanie
 - Čistenie (normalizácia, vyhladzovanie, vysporiadanie sa s chýbajúcimi hodnotami, redukcia)
 - Konštrukcia nových atribútov

Reprodukovateľné procesy analýzy dát (2)

- Modelovanie
 - Rozdelenie dátových množín
 - Optimalizácia nastavení algoritmov a operátorov predspracovania
- Predikcia
 - Aktuálna verzia modelu
 - Predspracovanie s rovnakou postupnosťou operácií

Výber a integrácia dát (1)

- Vytvorenie **dátovej snímky** (*snapshot*)
 - Export do perzistentných dátových súborov
 - Čo najpresnejšie zachovanie pôvodných dát pri konverzii do dátového formátu
 - Sémantické verziovanie dátových množín
 - Umožňuje prenos dát medzi prostrediami
 - Obmedzené pre veľké dáta

Výber a integrácia dát (2)

- Monotónny prístup k dátovým zdrojom
 - Priamy prístup k dátam v databáze - ak sa existujúce záznamy nemenia a iba sa pridávajú nové záznamy
 - Deterministický rozsah a poradie dát

```
SELECT * FROM invoice;
```

```
SELECT * FROM invoice
```

```
WHERE created_at BETWEEN '2024-01-01' AND '2025-12-31'
```

```
ORDER BY id;
```

Predspracovanie - Pipeline

- Postupnosť operácií pre predspracovanie dát + model
- Zabezpečuje rovnaké predspracovanie dát počas učenia a pri predikcii
 - Reprodukovateľnosť
 - Zjednodušenie nasadenia
 - Modulárnosť a udržiateľnosť kódu

Pipeline v Scikit-learn

- Dokumentácia

<https://scikit-learn.org/stable/modules/compose.html#combining-estimators>

- Príklad

https://github.com/peterbednar/insa/blob/main/prednaska3/pipeline_example.py

- Hlavné typy

- `sklearn.pipeline.Pipeline`



Programovanie operátorov v Scikit-learn

- Objektovo-orientovaný návrh
 - Transformačná funkcia + nastavenia
- Základné typy
 - `sklearn.base.BaseEstimator`
 - `sklearn.base.TransformerMixin`
- Príklad
- <https://github.com/peterbednar/insa/blob/main/prednaska3/processors.py>



Kontrolovanie náhodnosti (1)

- Náhodné rozdelenie dát
 - Trénovacia/validačná/testovacia množina, krížová validácia
- Náhodná inicializovanie parametrov modelu
 - Napr. váh pri neurónových sieťach, výber atribútov pri náhodných stromoch a pod.



Kontrolovanie náhodnosti (2)

- Generátory pseudo-náhodných čísel
 - Funkcia, ktorá pre zadanú inicializačnú hodnotu (*seed*) vygeneruje postupnosť čísel, ktorá sa javí náhodne
- Nastavenie generátora pre reprodukovateľné operácie - `random_state`

```
rnd = numpy.random.RandomState(1234)  
x = rnd.randint(1000, size=5)
```

```
# [815 723 294 53 204]
```



Výber príznakov

- Pri experimentovaní je súčasťou pipeline spolu s modelom
- Pre produkčný kód je lepšie špecifikovať iba vybrané atribúty

https://github.com/peterbednar/insa/blob/main/prednaska3/feature_selection.py

