



GoodData

P2D2 2019

Život postgresového serveru bez ručních zásahů

Jakub Jedelský

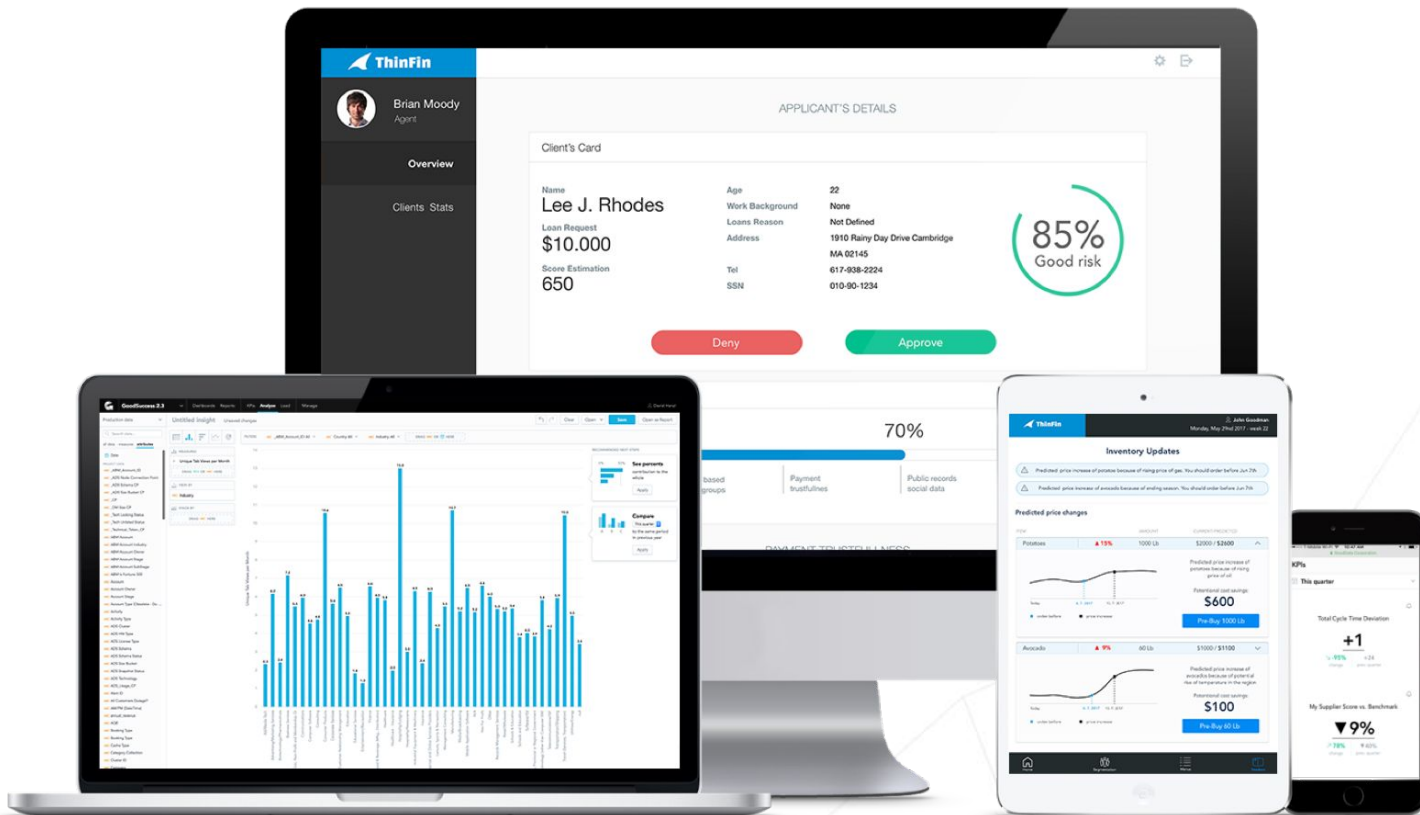


“GD je hodně silný v automatizaci”
– Pavel Stěhule



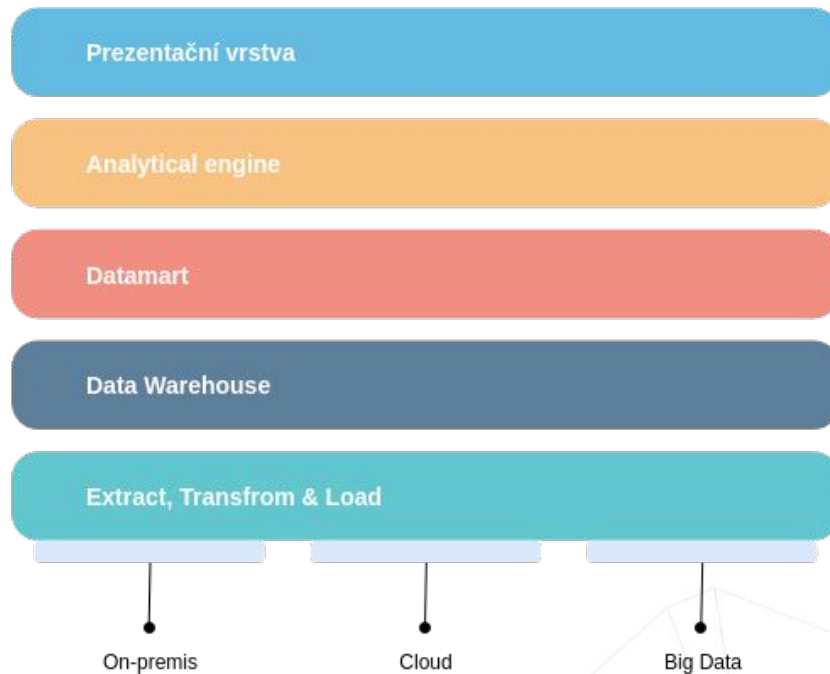
GoodData

Analytické aplikace velkého rozsahu



GoodData

Platforma



GoodData

Číslo: příklad zákazníka

- ▶ 29 000 projektů
- ▶ 880 000 koncových uživatelů
- ▶ 22,7 TB dat
- ▶ Největší: 750 GB s 13 500 uživateli

- ▶ 29 000 projektů
 - ▶ 880 000 koncových uživatelů
 - ▶ 22,7 TB dat
 - ▶ Největší: 750 GB s 13 500 uživateli
-
- ▶ 4 554 678 zobrazení / den
 - ▶ 15 000 / minuta ve špičce
 - ▶ 20 000 000 queries / den
 - ▶ 2 300 hodin počítání

Projekt

- ▶ Informace o uživateli
- ▶ Metadata
- ▶ Data
- ▶ Cache

Projekt

- Informace o uživateli

- Metadata



- Data



- Cache



Projekt

- Informace o uživateli

- Metadata



- Data



- Cache



29 000 × 3

Život serveru

- ▶ Deployment
- ▶ Údržba
- ▶ Ukončení



Život serveru

Deployment

- ▶ Deployment
 - ▶ Vytvoření virtuálního serveru
 - ▶ Konfigurace
 - ▶ Start služeb
 - ▶ Registrace do aplikace
- ▶ Údržba
- ▶ Ukončení



Život serveru

Údržba

- ▶ Deployment
 - ▶ Vytvoření virtuálního serveru
 - ▶ Konfigurace
 - ▶ Start služeb
 - ▶ Registrace do aplikace
- ▶ Údržba
 - ▶ Monitoring
 - ▶ Alerting
 - ▶ Logování
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Release / ad hoc operativa
- ▶ Ukončení



Život serveru

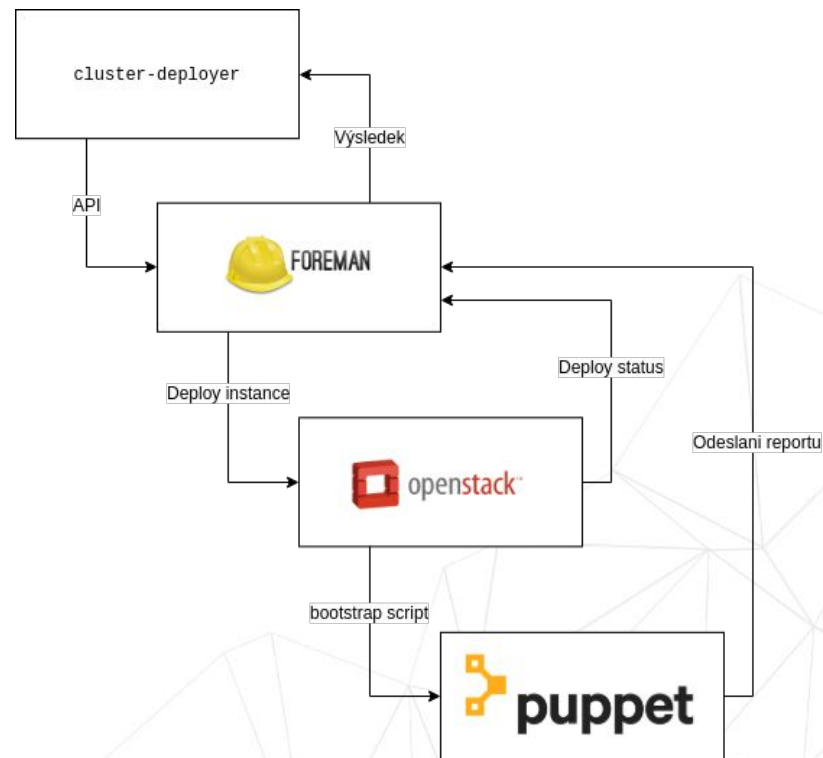
Ukončení

- ▶ Deployment
 - ▶ Vytvoření virtuálního serveru
 - ▶ Konfigurace
 - ▶ Start služeb
 - ▶ Registrace do aplikace
- ▶ Údržba
 - ▶ Monitoring
 - ▶ Alerting
 - ▶ Logování
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Release / ad hoc operativa
- ▶ Ukončení
 - ▶ Vyklopení serveru
 - ▶ Odpojení z aplikace
 - ▶ Terminace

Život serveru

Deployment

- Deployment
 - **Vytvoření virtuálního serveru**
 - Konfigurace
 - Start služeb
 - Registrace do aplikace



Deployment

Vytvoření virtuálního serveru

Cluster-deployer:

- ▶ Vlastní nástroj
- ▶ Foreman klient
- ▶ YAML konfigurace
- ▶ Verzování konfigurace



Deployment

Vytvoření virtuálního serveru

Cluster-deployer:

- ▶ Vlastní nástroj
- ▶ Foreman klient
- ▶ YAML konfigurace
- ▶ Verzování konfigurace

```
---
test-pg01:
  description: "Postgresql Datamart"
  flavor: c4r30e950
  parameters:
    icinga_services: !csv
      - pgsql
      - ...
  role: pdwh
  type: cl_pdwh
  require:
    - test-maria01
```


Deployment

Vytvoření virtuálního serveru

Cluster-deployer:

- ▶ Vlastní nástroj
- ▶ Foreman klient
- ▶ YAML konfigurace
- ▶ Verzování konfigurace

```
---
test-pg01:
  description: "Postgresql Datamart"
  flavor: c4r30e950
  parameters:
    icinga_services: !csv
      - pgsql
      - ...
  role: pdwh
  type: cl_pdwh
  require:
    - test-maria01
```

```
$ cluster-deployer --hosts test-pg01 test-maria01 -t 2 -- cluster/64.yaml
```

Deployment

Vytvoření virtuálního serveru

Cluster-deployer:

- ▶ Vlastní nástroj
- ▶ Foreman klient
- ▶ YAML konfigurace
- ▶ Verzování konfigurace

OS řešení:



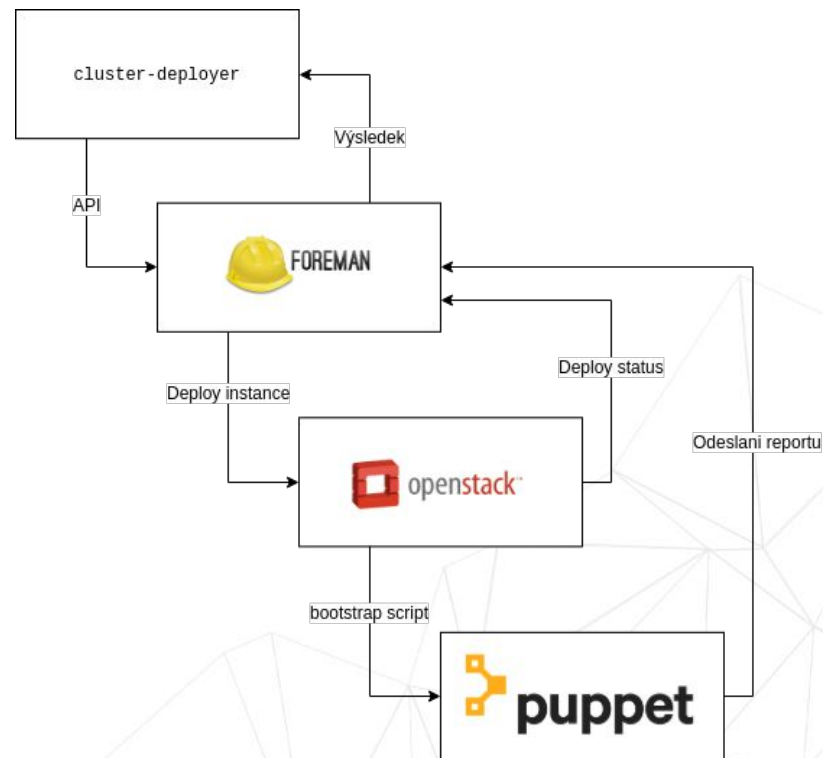
```
---
test-pg01:
  description: "Postgresql Datamart"
  flavor: c4r30e950
  parameters:
    icinga_services: !csv
      - pgsql
      - ...
  role: pdwh
  type: cl_pdwh
  require:
    - test-maria01
```

```
$ cluster-deployer --hosts test-pg01 test-maria01 -t 2 -- cluster/64.yaml
```

Život serveru

Deployment

- Deployment
 - Vytvoření virtuálního serveru
 - **Konfigurace**
 - **Start služeb**
 - **Registrace do aplikace**



Deployment

Konfigurace, start služeb, registrace do aplikace

Puppet:

- ▶ Software Configuration Management
- ▶ Vlastní deklarativní jazyk
- ▶ Popisuje stav serveru
- ▶ Release
 - ▶ Změny konfigurace
 - ▶ Instance nových a aktualizovaných balíčků
 - ▶ Start služeb



Deployment

Konfigurace, start služeb, registrace do aplikace

Puppet:

- ▶ Software Configuration Management
- ▶ Vlastní deklarativní jazyk
- ▶ Popisuje stav serveru
- ▶ Release
 - ▶ Změny konfigurace
 - ▶ Instance nových a aktualizovaných balíků
 - ▶ Start služeb

```
# postgresql.pp

class { 'postgresql::server' :
}
```

Deployment

Konfigurace, start služeb, registrace do aplikace

Puppet:

- ▶ Software Configuration Management
- ▶ Vlastní deklarativní jazyk
- ▶ Popisuje stav serveru
- ▶ Release
 - ▶ Změny konfigurace
 - ▶ Instance nových a aktualizovaných balíčků
 - ▶ Start služeb

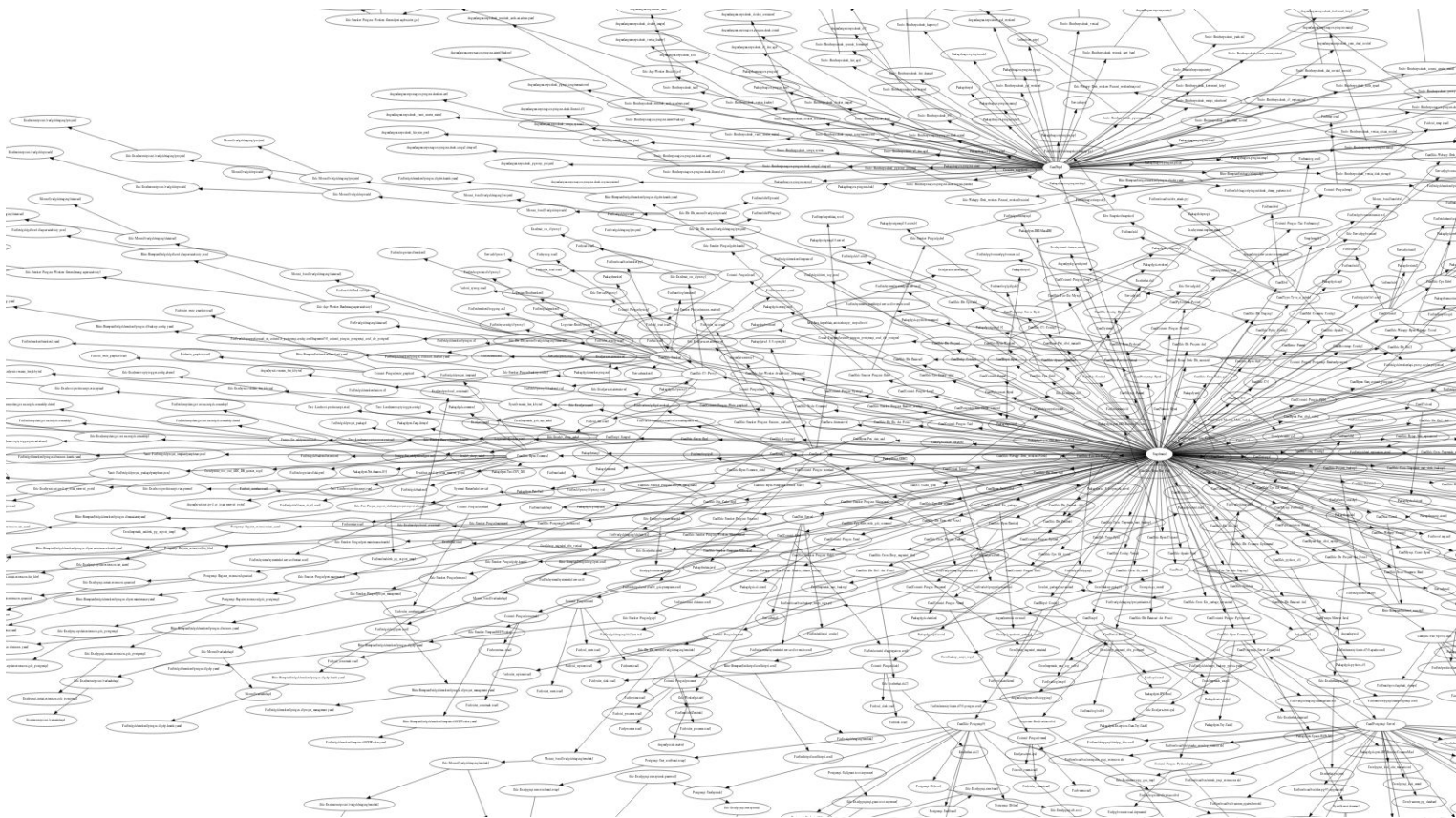
```
# postgresql.pp

class { 'postgresql::server' :
}

postgresql::server::db { 'mydb' :
  user => 'dbadmin',
  password => postgresql_password( 'dbadmin', 'Secret!' ),
}
```

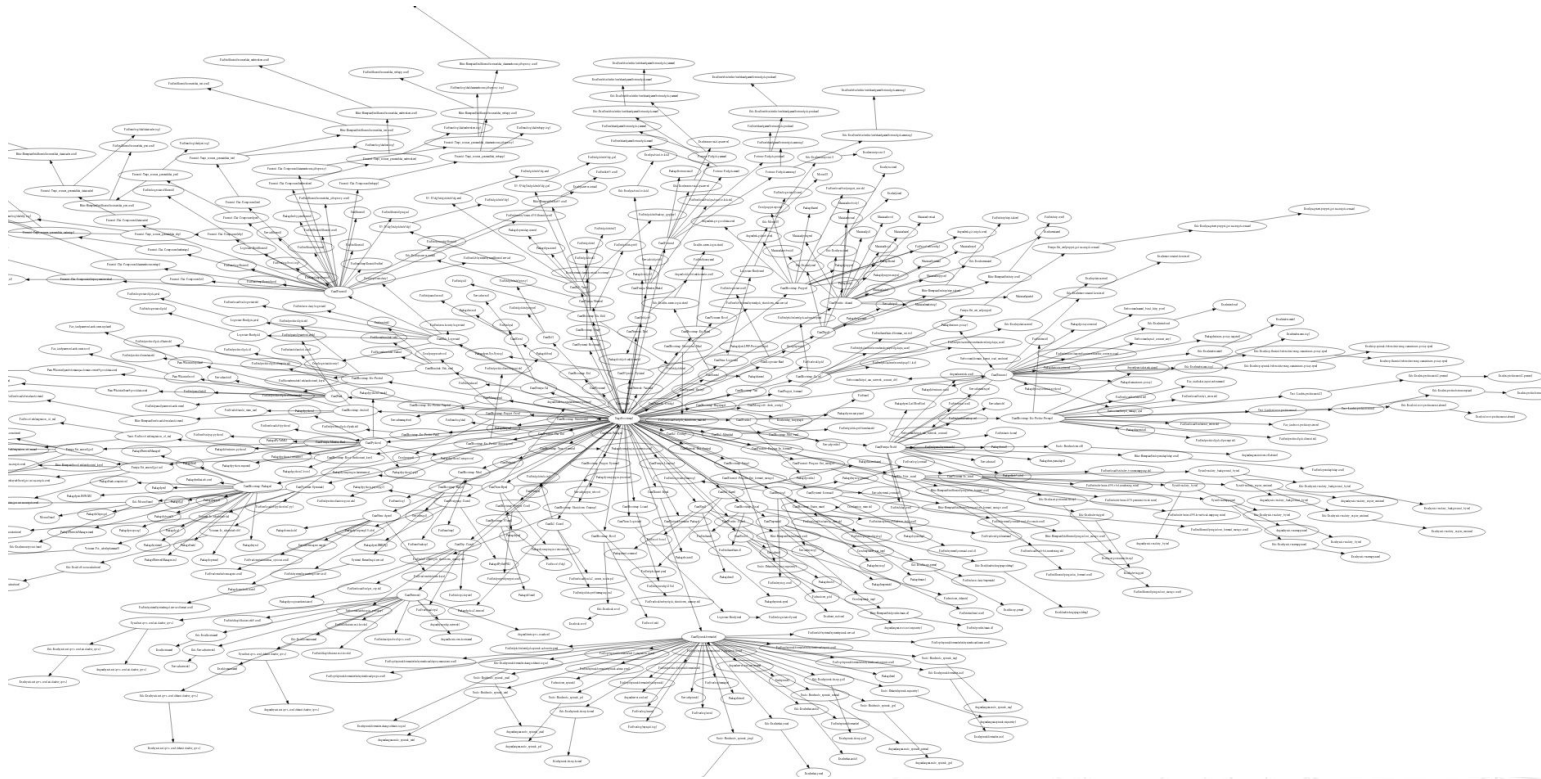
Deployment

Puppet pro Pg nodu v Gooddata



Deployment

Puppet pro Pg nodu v Gooddata



Život serveru

Údržba

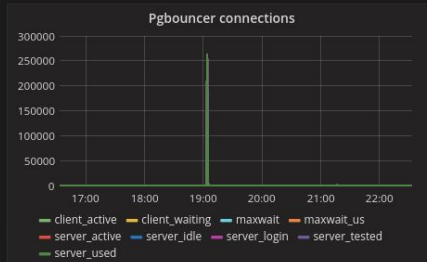
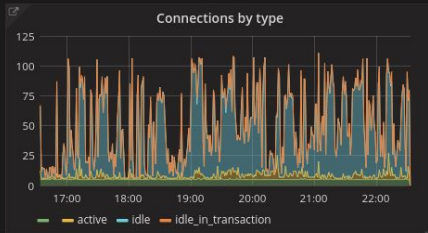
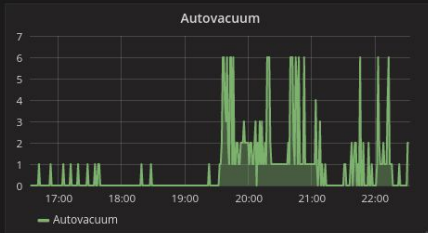
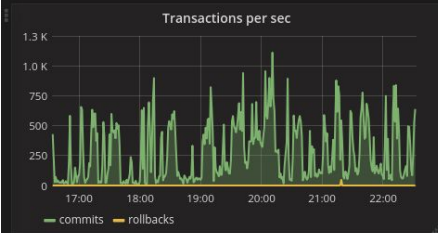
- ▶ Údržba
 - ▶ **Monitoring**
 - ▶ Alerting
 - ▶ Logování
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Release / ad hoc operativa



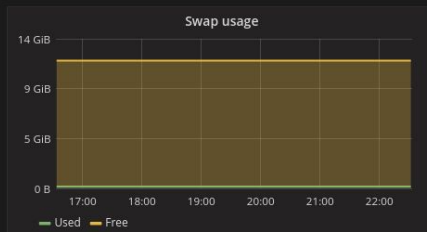
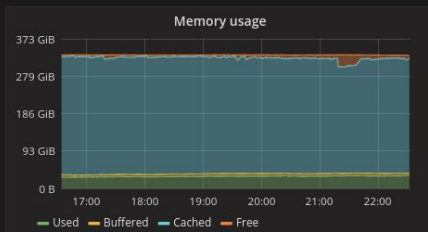
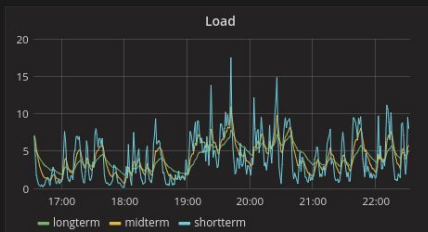
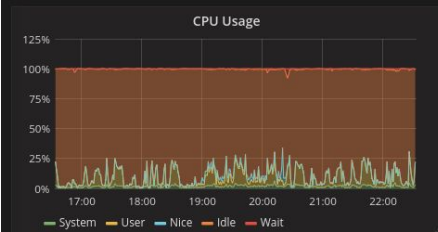
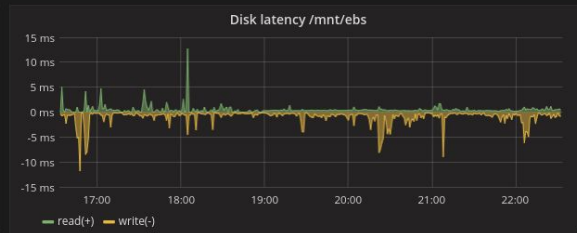
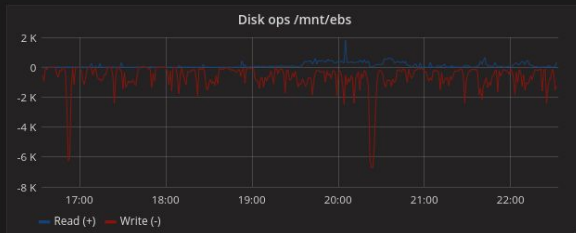
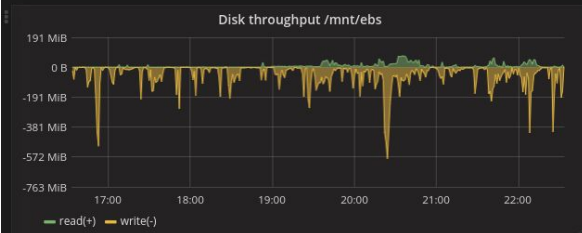
Monitoring

Graphite & Grafana

Postgresql details



System details



Monitoring

Lokální vs. vzdálený

Lokální:

- ▶ `collectd`
- ▶ doručeno lokálně na server puppetem



Monitoring

Lokální vs. vzdálený

Lokální:

- ▶ collectd
- ▶ doručeno lokálně na server puppetem

```
# monitoring.pp

collectd::plugin::postgresql::query { 'select-1' :
  statement => "SELECT 1 AS one",
  results   => [{
    type           => 'gauge',
    valuesfrom     => 'one',
    instanceprefix => 'select-1',
  ]}
}

collectd::plugin::postgresql::database { 'postgres' :
  (...)
  query => ['select-1']
}
```

Monitoring

Lokální vs. vzdálený

Lokální:

- ▶ collectd
- ▶ doručeno lokálně na server puppetem

- + server odesílá vlastní info
- + nepotřebuje vzdálený přístup
- pomalý release

```
# monitoring.pp

collectd::plugin::postgresql::query { 'select-1' :
  statement => "SELECT 1 AS one",
  results   => [{
    type           => 'gauge',
    valuesfrom     => 'one',
    instanceprefix => 'select-1',
  ]}
}

collectd::plugin::postgresql::database { 'postgres' :
  (...)
  query => ['select-1']
}
```

Monitoring

Lokální vs. vzdálený

Vzdálený:

- ▶ Vlastní služba (immutabilní infrastruktura)
- ▶ Doručováno po merge

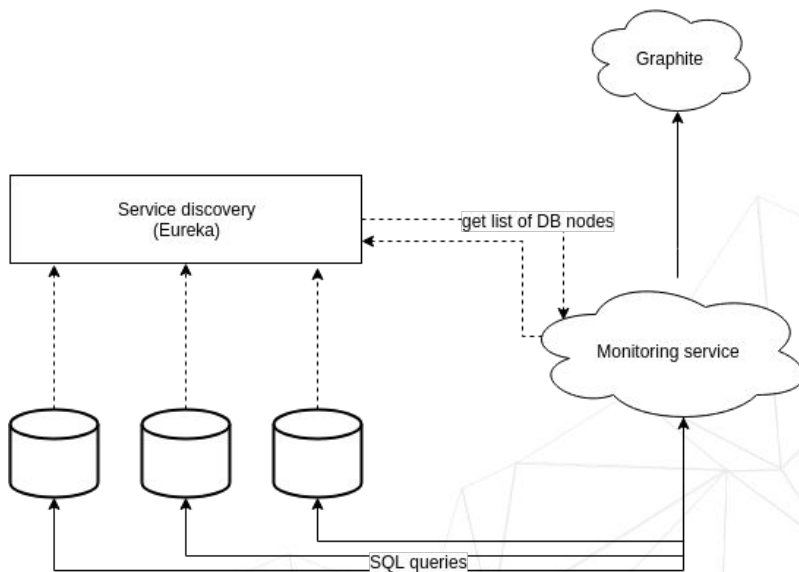


Monitoring

Lokální vs. vzdálený

Vzdálený:

- ▶ Vlastní služba (immutabilní infrastruktura)
- ▶ Doručováno po merge



Lokální vs. vzdálený

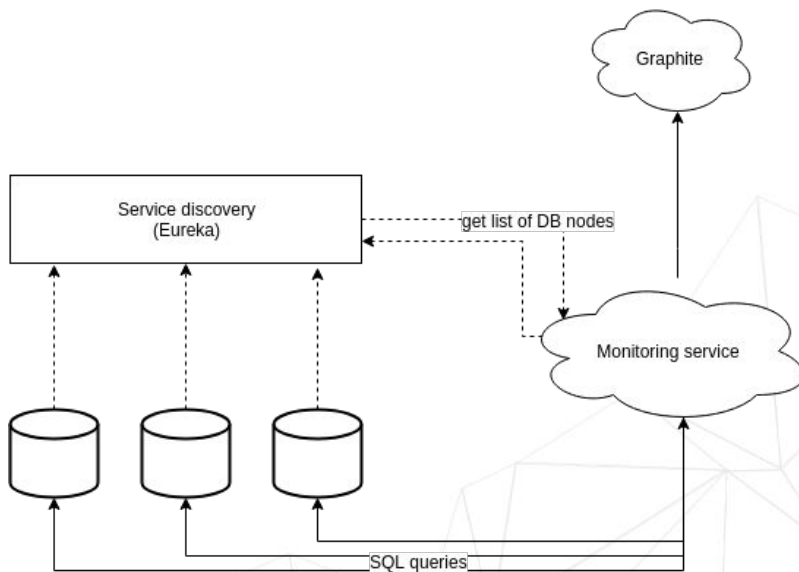
Lokální vs. vzdálený

Vzdálený:

- ▶ Vlastní služba (immutabilní infrastruktura)
- ▶ Doručováno po merge

+ rychlý release

- vyžaduje třetí službu (service discovery)



Monitoring

Grafana dashboard

Grafana dashboard builder:

- ▶ Popis dashbordu v YAML
- ▶ Verzovatelné
- ▶ Open source
- ▶ <https://github.com/jakubplichtha/grafana-dashboard-builder/>

```
- name: overview
  dashboard:
    title: '{dashboard-prefix} Overview'
    tags:
      - tag1
      - tag2
    <<: *default-dashboard
    rows:
      - row:
          title: '{dashboard-prefix}-row'
          panels:
            - graph:
                span: 3
                title: Frontend
                target:
                  'aliasByMetric({metric-prefix}.frontend.*)'
            - graph:
                span: 3
                title: Backend
                target:
                  'aliasByMetric({metric-prefix}.backend.*)'
```

Život serveru

Údržba

- ▶ Údržba
 - ▶ Monitoring
 - ▶ **Alerting**
 - ▶ Logování
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Release / ad hoc operativa



Alerting



- ▶ E-mail
- ▶ Pagerduty.com



Alerting

Puppet



- ▶ E-mail
- ▶ Pagerduty.com



```
# icinga template
apply Service "pgsql" {
    check_command = "nrpe"
    vars.nrpe_command = "check_pgsql"
    display_name= "PgSQL"
    assign where "pgsql" in host.vars.gdc_services
}
```

Alerting

Puppet + cluster-deployer



- ▶ E-mail
- ▶ Pagerduty.com



```
# icinga template
apply Service "pgsql" {
    check_command = "nrpe"
    vars.nrpe_command = "check_pgsql"
    display_name= "PgSQL"
    assign where "pgsql" in host.vars.gdc_services
}
```

```
# cluster-deployer
stg3-pdwh01:
  parameters:
    icinga_services: !csv
      - pgsql
```

Život serveru

Údržba

- ▶ Údržba
 - ▶ Monitoring
 - ▶ Alerting
 - ▶ **Logování**
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Release / ad hoc operativa



Logování

- ▶ Aplikační log
- ▶ Postgresql log
- ▶ Postgresql slow queries
- ▶ Postgresql auto_explain
- ▶ Vertica queries



- ▶ Aplikační log
- ▶ Postgresql log
- ▶ Postgresql slow queries
- ▶ Postgresql auto_explain
- ▶ Vertica queries



splunk>

VERTICA



- ▶ Aplikační log
- ▶ Postgresql log
- ▶ Postgresql slow queries
- ▶ Postgresql auto_explain
- ▶ Vertica queries

splunk[®]

?

VERTICA



- ▶ Aplikační log
- ▶ Postgresql log
- ▶ Postgresql slow queries
- ▶ Postgresql auto_explain
- ▶ Vertica queries

splunk®

?

VERTICA

```
index=os source=/mnt/log/postgresql.log "automatic vacuum" "pg_catalog"  
| rex field=_raw "automatic vacuum of table \"(?<db_name>\\w+)\\.\\.\\.\""  
| stats count by db_name,host  
| search count > 500
```

Logování

- ▶ Aplikační log
- ▶ Postgresql log
- ▶ Postgresql slow queries
- ▶ Postgresql auto_explain
- ▶ Vertica queries

splunk® >

?

VERTICA

OS řešení:



elasticsearch



kibana

```
index=os source=/mnt/log/postgresql.log "automatic vacuum" "pg_catalog"  
| rex field=_raw "automatic vacuum of table \"(?<db_name>\\w+)\\.\\.\\.\""  
| stats count by db_name,host  
| search count > 500
```

Život serveru

Údržba

- ▶ Údržba
 - ▶ Monitoring
 - ▶ Alerting
 - ▶ Logování
 - ▶ **Rozdělování zátěže (balancing)**
 - ▶ Release / ad hoc operativa



Balancing

Redistribuce DB skrz platformu

Priority:

1. Počítání
2. Load
3. Velikost



Balancing

Redistribuce DB skrz platformu

Priority:

1. Počítání
2. Load
3. Velikost

} Hodinová historická data



Balancing

Redistribuce DB skrz platformu

Priority:

1. Počítání
2. Load
3. Velikost

Hodinová historická data



Skóre projektu



Balancing

Redistribuce DB skrz platformu

Priority:

1. Počítání
2. Load
3. Velikost

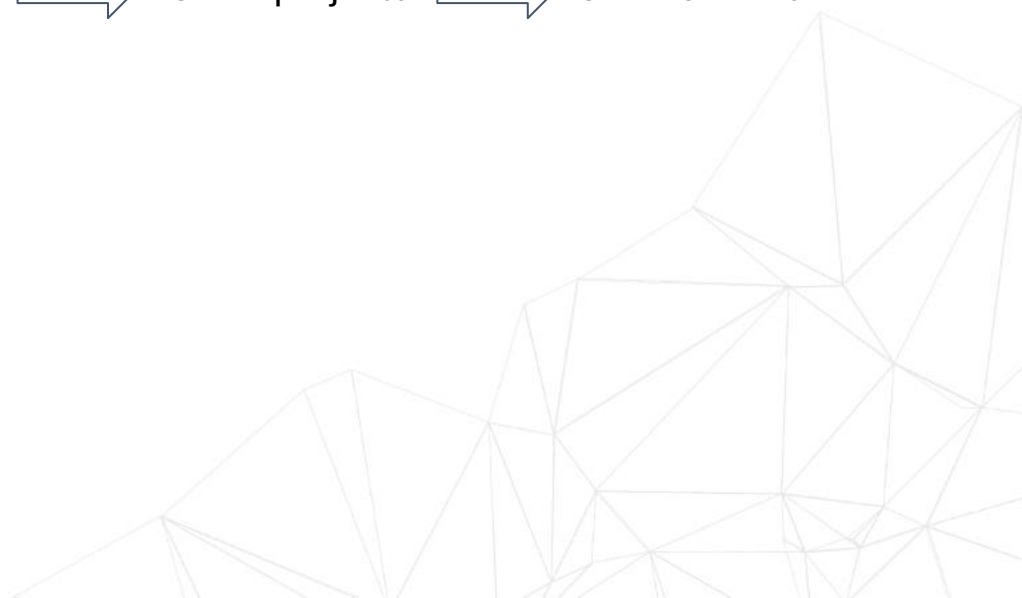
Hodinová historická data



Skóre projektu



Skóre serveru



Balancing

Redistribuce DB skrz platformu

Priority:

1. Počítání
2. Load
3. Velikost

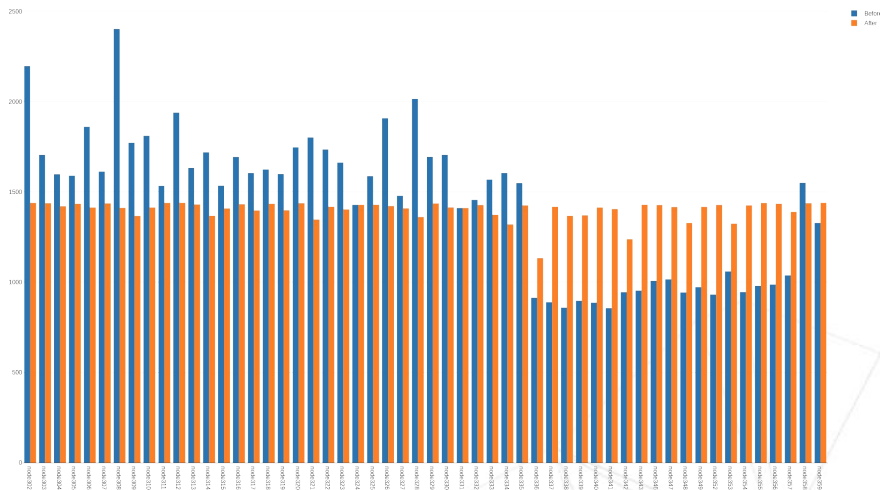
Hodinová historická data



Skóre projektu




Skóre serveru



Balancing

Redistribuce DB skrz platformu

Migrace:

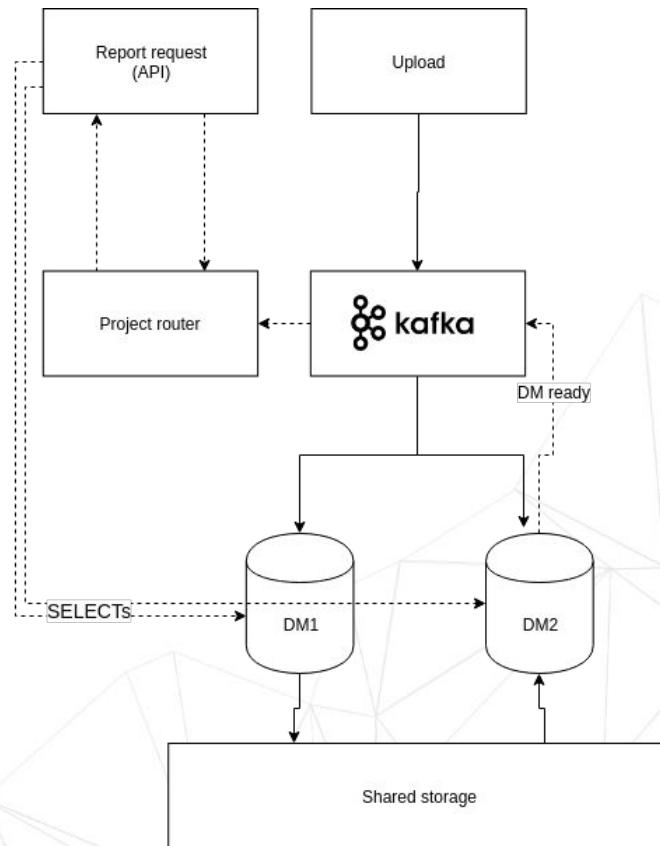
- ▶ PostgreSQL ↔ PostgreSQL
 - ▶ PostgreSQL ↔ Vertica
 - ▶ Vertica ↔ Vertica
 - ▶ Cross DC
- 

Balancing

Redistribuce DB skrz platformu

Migrace:

- ▶ PostgreSQL ↔ PostgreSQL
 - ▶ PostgreSQL ↔ Vertica
 - ▶ Vertica ↔ Vertica
 - ▶ Cross DC
1. Zapnutí commit logu
 2. Export + import dat
 3. Přepnutí zdroje v Project router
 4. Vyčištění



Život serveru

Údržba

- ▶ Údržba
 - ▶ Monitoring
 - ▶ Alerting
 - ▶ Rozdělování zátěže (balancing)
 - ▶ Logování
 - ▶ **Release / ad hoc operativa**



Release / ad hoc operativa

Historie: clusterExec.py

Historie: clusterExec.py

- ▶ Multi-ssh klient
- ▶ Vzdálené spouštění příkazů
- ▶ Upload souborů
- ▶ List nodů v (sub)clusteru



Release / ad hoc operativa

Historie: clusterExec.py

Historie: clusterExec.py

- ▶ Multi-ssh klient
- ▶ Vzdálené spouštění příkazů
- ▶ Upload souborů
- ▶ List nodů v (sub)clusteru

+ rychlost



Release / ad hoc operativa

Historie: clusterExec.py

Historie: clusterExec.py

- ▶ Multi-ssh klient
- ▶ Vzdálené spouštění příkazů
- ▶ Upload souborů
- ▶ List nodů v (sub)clusteru

+ rychlost

- bez historie
- netestovatelné
- pouze pro uživatele s přístupem na produkci



Release / ad hoc operativa

Historie: clusterExec.py

Historie: clusterExec.py

- ▶ Multi-ssh klient
- ▶ Vzdálené spouštění příkazů
- ▶ Upload souborů
- ▶ List nodů v (sub)clusteru

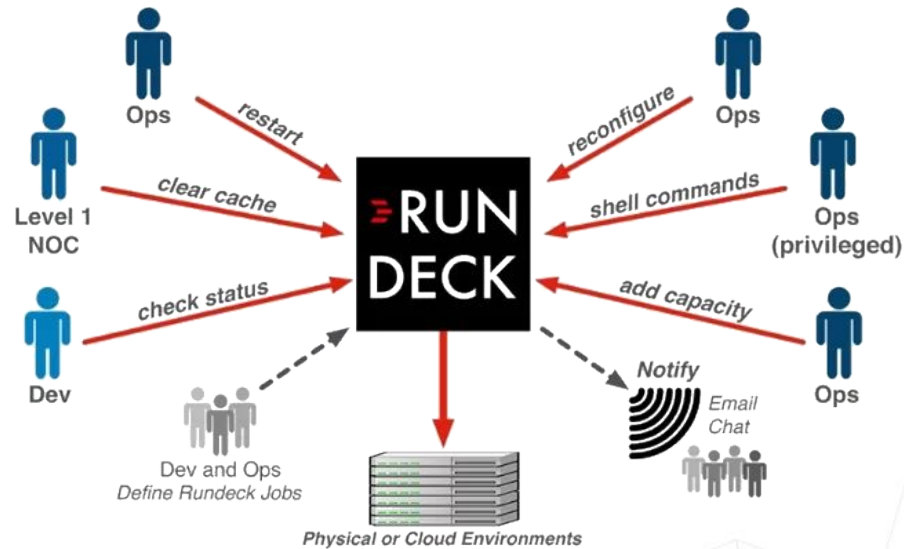
+ rychlost

```
clusterExec.py -c prod -- 'sudo rm -rf / tmp/directory/'
```

- bez historie
- netestovatelné
- pouze pro uživatele s přístupem na produkci

Release / ad hoc operativa

Rundeck



Release / ad hoc operativa

Rundeck

Rundeck:

- ▶ YAML konfigurace
 - ▶ Code review
 - ▶ Verzovatelné
- ▶ Skládání úloh jako “Lego® kostičky”

```
- description: 'Stop PostgreSQL Database'
  name: {{name}}_stop_postgres_db
  options:
  - description: 'Wait time in minutes'
    name: stop-postgres-db.wait
    value: '15'
    required: true
  sequence:
    commands:
    - description: 'Stop postgres database'
      script: |-
        set -x

        function wait_for_postgres() {
          local SERVICE_NAME=$1
          for ((i=1;i<="@option.stop-postgres-db.wait@";++i)); do
            echo "Waiting till postgres is stopped"
            sudo service $SERVICE_NAME status
            if [[ $? -gt 0 ]]; then
              echo "Postgres reported it shut down successfully"
              return 0
            fi
            sleep 60
          done

          echo "postgres database did not stop in @option.stop-postgres-db.wait@ minutes"
          return 1
        }

        systemctl stop postgresql
        systemctl stop pgbouncer

        wait_for_postgres $POSTGRES_SERVICE
        exit $?
```

Život serveru

Ukončení

- ▶ Ukončení
 - ▶ Vyklizení serveru
 - ▶ Odpojení z aplikace
 - ▶ Terminace



- ▶ Ukončení
 - ▶ Vyklizení serveru
 - ▶ Odpojení z aplikace
 - ▶ Terminace



- ▶ Ukončení
 - ▶ **Vyklizení serveru**
 - ▶ Odpojení z aplikace
 - ▶ Terminace



1. Bezvýpadkový rebalancing

- ▶ Ukončení
 - ▶ Vyklizení serveru
 - ▶ **Odpojení z aplikace**
 - ▶ Terminace



1. Bezvýpadkový rebalancing
2. Odpojení (shell cmd)

Život serveru

Ukončení

- ▶ Ukončení
 - ▶ Vyklizení serveru
 - ▶ Odpojení z aplikace
 - ▶ **Terminace**



1. Bezvýpadkový rebalancing
2. Odpojení (shell cmd)

3. `$ cluster-deployer --hosts test-pg01 test-maria01 -t 2 -- cluster/64.yaml`

Život serveru

Ukončení

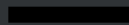
- ▶ Ukončení
 - ▶ Vyklizení serveru
 - ▶ Odpojení z aplikace
 - ▶ **Terminace**



1. Bezvýpadkový rebalancing
2. Odpojení (shell cmd)

3. `$ cluster-deployer --terminate --hosts test-pg01 test-maria01 -t 2 -- cluster/64.yaml`

Díky.



`jakub.jedelsky@gmail.com`

`https://medium.com/gooddata-developers`

Q?