Industrial PostgreSQL

Christoph Mönch-Tegeder

2ndQuadrant
http://www.2ndquadrant.com/

2018-10-25

2ndQuadrant®
PostgreSQL

Introduction

- Consultant with 2ndQuadrant
- Based out of Karlsruhe, Germany
- Travelling and visiting customers

2ndQuadrant®
PostgreSQL

This Customer...

- *No Names, please!*
- well-established company
- world-wide presence

This Customer: History

- in business: longer than any of us
  - before IT was invented
- uses IT longer than PostgreSQL exists

This Customer: Numbers

- employees: about 300000
- revenue: some 100 Billion Euros
  - enough for a small country

What's their Business?

Every business, regardless of what business they think they're in, is an IT company. **Christopher Little (link not found)**

- some actively ignore reality

Their Business

What's Running?

- Lots of Linux
- Uses several commercial (closed-source) databases
- And Now: PostgreSQL officially supported

Tons of Documentation

- Need to have operating concepts
- Special packaging process
- OS setup, file system layout, Linux users, ...
- PostgreSQL configuration, extensions, ...
- Backup and recovery strategies

Tons of Documentation

- Need to have operating concepts
- Special packaging process
- OS setup, file system layout, Linux users, ...
- PostgreSQL configuration, extensions, ...
- Backup and recovery strategies
- A good stack of documents

Security Framework

- *Hardening Technical Controls*
- Security checklist from Corporate IT
  - Relevant configuration parameters
  - SSL ciphers and certificates
  - Schema hardening (`REVOKE ALL PRIVILEGES FROM PUBLIC`)

Security Framework

- *Hardening Technical Controls*
- Security checklist from Corporate IT
  - Relevant configuration parameters
  - SSL ciphers and certificates
  - Schema hardening (`REVOKE ALL PRIVILEGES FROM PUBLIC`)
- Mostly Common Sense

Role Concept

- `SUPERUSER` not for end-users
- Group Roles with Default Privileges
  - Schema Owners, for DDL
  - Application Users, data r/w
  - Read Only access
- Login Roles are assigned to the groups

Landscape

- One Company, many data centers
- High degree of autonomy
- Central IT defines some standards
  - Approves technology for different kinds of applications

Corporate Guidelines

- *Virtual Competence Team*
- Committee formed by stakeholders
- Defines corporate standards
- Long-term planning

Building a PostgreSQL Service

- Largest and most complex data center
- Traditional, *Iron Age* organisation
- Their Service Managers, Project Managers
- Contractors handling the products
- Operations outsourced & offshored
- Tickets for Everything

Infrastructure

- Everything running on vmware
- The infrastructure rocks

```
# dd if=/dev/zero of=/dev/sdb1 bs=1M count=270000
270000+0 records in
270000+0 records out
283115520000 bytes (283 GB, 264 GiB) copied,
    465.113 s, 609 MB/s
```

- Backups on IBM Spectrum Protect (TSM)

Backup

- TSM is a **must** here
  - base backup and archive, then let TSM backup that
  - use `barman`, then backup `barman`
  - special TSM agent: Repostor Data Protector

Backup

- TSM is a **must** here
  - base backup and archive, then let TSM backup that
  - use `barman`, then backup `barman`
  - special TSM agent: Repostor Data Protector
- Repostor Licence already bought by Backup Team
- so we have to use it (but it works!)

How To PostgreSQL?

- Detailed Work Instructions
  - CPU, RAM, storage specified by customer
  - users, package installation, `initdb`, configuration
  - backup configuration (TSM!)
  - everthing as a ticket

How To PostgreSQL?

- Detailed Work Instructions
  - CPU, RAM, storage specified by customer
  - users, package installation, `initdb`, configuration
  - backup configuration (TSM!)
  - everthing as a ticket
- Thanks, but No Thanks.

Solution 1: Standardize Everything

- Standards for
  - PostgreSQL configuration
  - Linux configuration
  - VMs (CPU, RAM) and Storage
  - Authentication (use Active Directory)
- 4 Standard Sizes: S, M, L, XL (*T-Shirt Sizes*)

Can We Script it?

- This is Linux, so we have shell?

  Your simple script is someone else's bad day

  Rachel Kroll, http://rachelbythebay.com/w/2013/08/01/script/

Solution 2: Configuration Management

- Declarative System (just like SQL)
- Describe the desired System state
- Configuration Management executes any required changes
- *Idempotency*: subsequent invocations result in the same state

Saltstack

- OS Team already uses Saltstack for initial configuration
- No central server (at first)
- Salt-States packaged and distributed as RPM
- Some scripting still required
  - Test VM for compliance
  - Write Salt grains

Some Hacking Ensued...

- Initial setup and configuration
- SSL certificates (self-signed)
- Configuration changes
- Role management (add/remove)
- Software updates

Some Pain Remains

- register database with TSM
  - TSM is another team
- *real* centrally signed SSL certificates
  - infrastructure not there (yet)

Communications

- So many different roles
  - Technology Partner, Business Partner, Technical Project Lead, Service Team, Supplier, Operations, VCT
  - Each and Every service
- Highly regulated communications
- Long cycle times, Chinese Whispers

Infrastructure

- Building Infrastructure is about *removing* options
- Provide working and manageable solutions
- Variance across systems makes mass-management hard
- **But:** too many restrictions make the service unattractive

Staying Ahead

- Find users and talk to them!
  - Organisations sometimes make that harder than it should be
- Bring user feedback to Technology Management
- Coordinate with other Services

What We Learned

- Make extra efforts to create communication channels
- Large organizations move slowly
- Patience: many people involved, much communication
- Work with the Operations Team (not against)!
- First few servers require extra care

What They Learned

- Automation: much more than scripts
- Iterative Approaches are a Good Thing
- Building new services is not running a well-known service

Questions?

Thank You