



PostgreSQL und alles was dazu gehört - Elephant Shed

Alexander Sosna,
Adrian Vondendriesch

pgconf.de 2018

Alexander Sosna

- ▶ 4 Jahre bei credativ
- ▶ Projektleiter
- ▶ Technische Aufgaben

Adrian Vondendriesch

- ▶ 5 Jahre bei credativ
- ▶ Senior Berater
- ▶ Debian Maintainer

- ▶ IT-Dienstleister seit 1999
- ▶ Gründer u. Mitarbeiter aus Debian- u. PostgreSQL-Community
- ▶ Betrieb kritischer Infrastruktur mit OSS
- ▶ Schulungen
- ▶ Am Standort Mönchengladbach (NRW)
 - ▶ 40+ Mitarbeiter
 - ▶ 7 Kollegen im Datenbankteam

- ▶ Was muss beim Einsatz beachtet werden?

Unix-Philosophie

- ▶ Schreibe Computerprogramme so, dass sie nur eine Aufgabe erledigen und diese gut machen.
- ▶ Schreibe Programme so, dass sie zusammenarbeiten.
- ▶ Schreibe Programme so, dass sie Textströme verarbeiten, denn das ist eine universelle Schnittstelle.

<https://de.wikipedia.org/wiki/Unix-Philosophie>

PostgreSQL - Im luftleeren Raum?

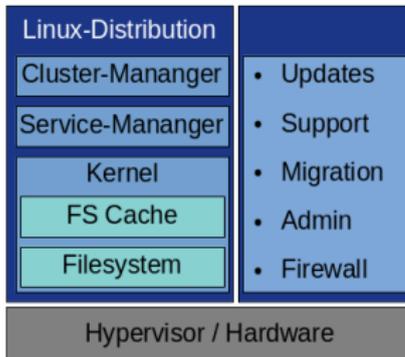


PostgreSQL

PostgreSQL - Im luftleeren Raum?



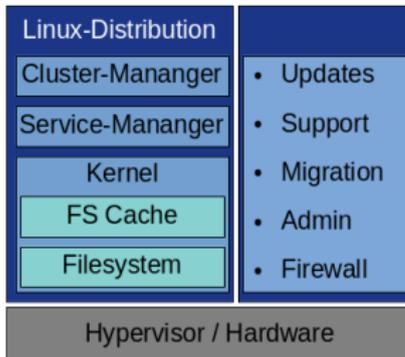
PostgreSQL



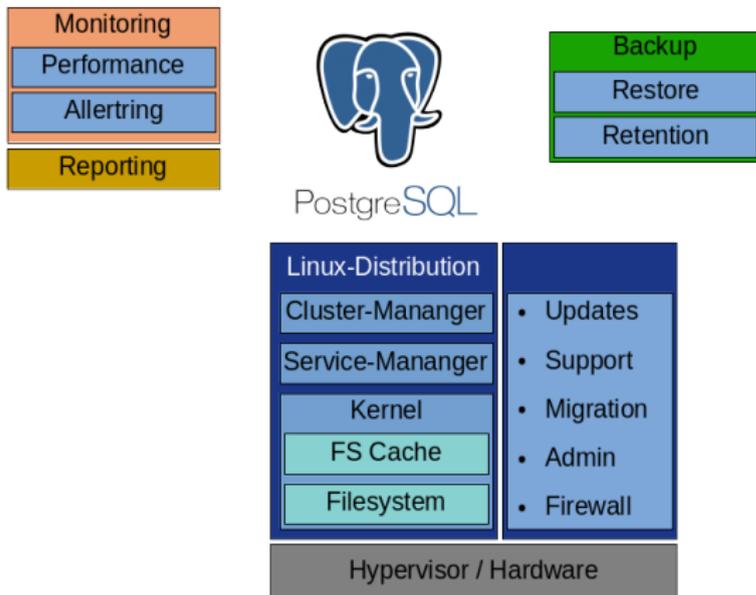
PostgreSQL - Im luftleeren Raum?



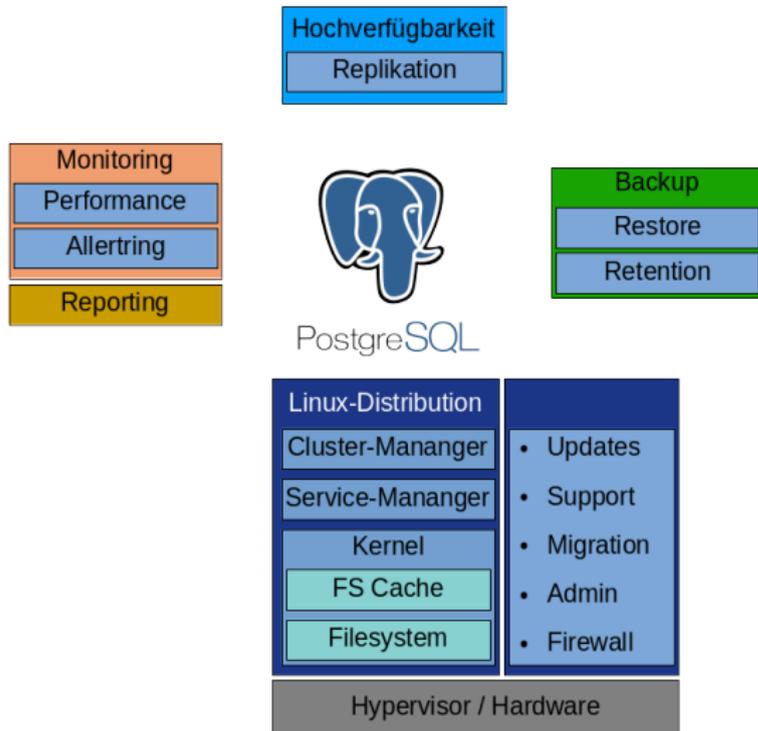
PostgreSQL



PostgreSQL - Ökosystem



PostgreSQL - Ökosystem



Kleiner Satz an Werkzeugen wird mitgeliefert, dafür existiert ein reichhaltiges und vielfältiges Ökosystem.

▶ Vorteile

- ▶ Schlanke Basis
- ▶ Gute Möglichkeit der Anpassung und Integration an die eigene Infrastruktur, Anforderungen und Prozesse

▶ Nachteile

- ▶ Keine All-in-One-Lösung
- ▶ Recherche und Architekturaufwand notwendig
- ▶ Tool-Auswahl: Oft gibt es viele konkurrierende Möglichkeiten
- ▶ Initialer Aufbau einer geeigneten Umgebung aufwendig
- ▶ Umfangreiche Tests des Gesamtsystems nötig

- ▶ Diverse Betriebssysteme
- ▶ Embedded Devices => Großrechner
- ▶ Architekturen: x86, PPC, ARM, ...
- ▶ Einzelinstallation => DB-Hoster (PostgreSQL as a Service)
- ▶ Konzeption, Aufbau, Betrieb und Review individueller Umgebungen

Wir sind federführend bei der Paketierung des PostgreSQL-Ökosystems für Debian/Ubuntu.

Klassische Einführung von PostgreSQL

- ▶ Langfristige Planung
- ▶ Hohe Anforderungen an Integration
 - ▶ Firmenprozesse
 - ▶ eigene Infrastruktur

Neue Anfragen

- ▶ **Zeitnahe** Migration von kommerziellem RDBMS zu PostgreSQL
- ▶ Lieber vorhandene Lösung als individuelles Konzept

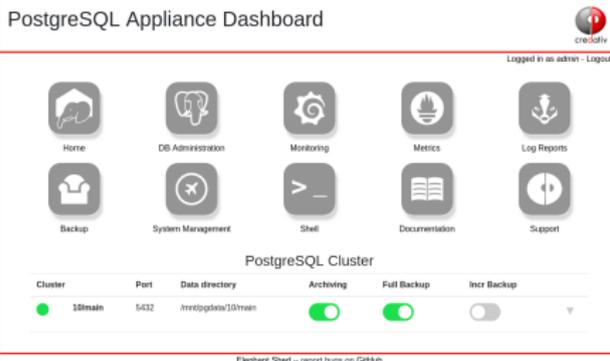
- ▶ Fertiger Blueprint
 - ▶ Ausprobieren
 - ▶ Testen
 - ▶ Betrieb
- ▶ PostgreSQL-Einstieg und -Betrieb so einfach wie möglich gestalten

- ▶ In sich geschlossenen
 - ▶ Alle Komponenten enthalten
 - ▶ Alle Komponenten aufeinander abgestimmt
- ▶ Für breiten Anwenderkreis (besonders Einsteiger)
 - ▶ Viele Anwendungsfälle abdecken
 - ▶ Jedoch kein hoher Aufwand für Nischen

- ▶ Modularer Aufbau
 - ▶ Erweiterbar
 - ▶ Nutzung der Komponenten für individuelle Setups
- ▶ 100% Open Source, keine proprietären Komponenten
- ▶ Langfristige Pflege und kommerzieller Support

- ▶ Monitoring
- ▶ Datenbankadministration
- ▶ Backups
- ▶ Logauswertung
- ▶ Systemverwaltung

PostgreSQL Appliance Dashboard



credativ
Logged in as admin - Logout

Home DB Administration Monitoring Metrics Log Reports
Backup System Management Shell Documentation Support

PostgreSQL Cluster

Cluster	Port	Data directory	Archiving	Full Backup	Incr Backup
10/main	5432	/var/lib/pgsql/10/main	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Elephant Shed - report bugs on GitHub

- ▶ Web basiertes Interface
- ▶ Zugriff via Apache Reverse Proxy (SSL)
- ▶ SSO via PAM Authentication

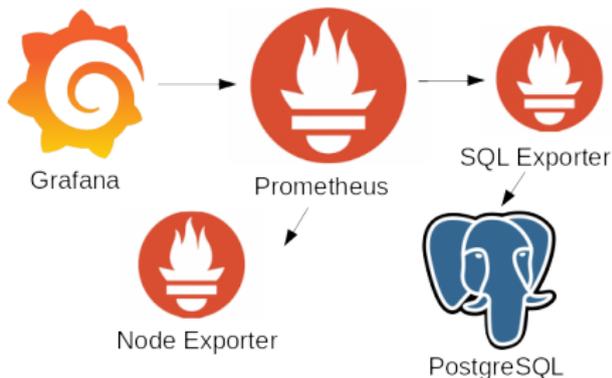


PostgreSQL

Anpassung der Standardkonfiguration:

- ▶ Listen Parameter
- ▶ Logging
- ▶ Backup
- ▶ Replication
- ▶ Extensions

- ▶ Grafana - Anzeige von Metriken
- ▶ Prometheus - Speicherung der Metriken
- ▶ SQL Exporter - Export von PostgreSQL Metriken
- ▶ Node Exporter - Export von Systemmetriken



Metriken:

- ▶ System (CPU, Memory, Disk-IO, Netzwerk-IO, Disk-Usage, etc.)
- ▶ Cluster (Verbindungen, Locks, Prepared Transactions, etc.)
- ▶ Datenbanken (Benutzer, Objektstatistiken, Queries, etc.)

Automatische Anpassung bei Veränderung:

- ▶ Neue Cluster
- ▶ Neue Datenbanken
- ▶ Neue Dateisysteme
- ▶ etc.



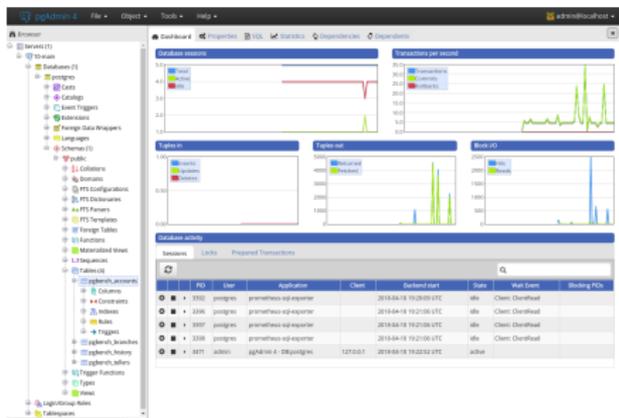
- ▶ Zusammenfassung aktueller Performance Werte
- ▶ Systemmetriken
- ▶ Clustermetriken
- ▶ Datenbankmetriken
- ▶ Replikation

Nicht vorkonfiguriert, aber:

- ▶ einfache Alerting Rules via Grafana,
- ▶ komplexere Alerting Rules via Prometheus möglich

Via pgAdmin4. Ermöglicht:

- ▶ Datenbankadministration
 - ▶ Schemaverwaltung
 - ▶ Userverwaltung
 - ▶ Dump / Restore
- ▶ Ad-hoc SQL Queries



- ▶ Physical Backups / Basebackups
- ▶ WAL Archiving / PITR Fähig

Features:

- ▶ Compression (On Disk / Network)
- ▶ Checksumming
- ▶ Encryption
- ▶ Full, Differential und Incremental Backups
- ▶ Backup to Disk und S3



Backups - Basiskonfiguration



- ▶ Vorbereitete postgresql.conf
- ▶ Vorkonfigurierte Retention Policy
- ▶ Vorbereitete Systemd-Timer
- ▶ Ad-hoc Backups möglich
- ▶ Aktivierung durch Webinterface

PostgreSQL Cluster

Cluster	Port	Data directory	Archiving	Full Backup	Incr Backup	
10/main	5432	/mnt/pgdata/10/main	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	▲

systemd Report Backup

Service Log Run Show Full Incremental Info

Generierung von HTML Reports via pgBadger.

- ▶ Aufarbeitung der Logdateien, z.B.:
 - ▶ Fehlermeldungen
 - ▶ Statement Laufzeit
(`log_min_duration_statement`)
 - ▶ Verbindungen über Zeitraum
- ▶ Automatische Generierung:
 - ▶ Täglich via Systemd-Timer
 - ▶ Ad-Hoc via Webinterface
 - ▶ Für jeden Cluster



Via Cockpit.

- ▶ Verwaltung von Services (start, stop, enable, disable)
- ▶ Review und Einspielen von Updates
- ▶ Journal Liveview
- ▶ User Administration



Aktuell nur Debian 9 (Stretch)

- ▶ x86_64
- ▶ PPC64el

Weitere geplant:

- ▶ Ubuntu 18.04
- ▶ CentOS 7

- ▶ Installation via Debian Pakete
- ▶ Pakete in eigenem Repository (`packages.credativ.com`)
- ▶ Abhängigkeiten auf `apt.postgresql.org`
- ▶ Beinhaltet 3rd Party Pakete

Keine Konfiguration nötig

Schritte

```
sudo apt-get install curl ca-certificates apt-transport-https

# Use official PostgreSQL repo, apt.postgresql.org
echo "deb http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main" | \
  sudo tee -a /etc/apt/sources.list.d/pgdg.list
curl https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Use credativ repo, packages.credativ.com
echo "deb https://packages.credativ.com/public/postgresql/ stretch-stable main" | \
  sudo tee -a /etc/apt/sources.list.d/elephant-shed.list
curl https://packages.credativ.com/public/postgresql/aptly.key | sudo apt-key add -

# Install elephant-shed
sudo apt-get update
sudo apt-get install elephant-shed

# Choose desired PostgreSQL versions to install
sudo apt-get install postgresql-10

# Every user in the group "elephant-shed" is allowed to login at the portal
# Add all needed users to this group
sudo adduser <USERNAME> elephant-shed
```

<https://elephant-shed.io/project/debian/>

Schritte

```
vagrant init credativ/elephant-shed  
vagrant up
```

```
https://elephant-shed.io/project/vagrant/
```

DEMO

- ▶ Plattformen
 - ▶ Ubuntu 18.04 aktuell in Entwicklung
 - ▶ CentOS 7 geplant für Q3
- ▶ Bereitstellung einer postgresql-common API
 - ▶ Konfigurationsänderungen via Webinterface
 - ▶ Cluster Management via Webinterface
- ▶ Kontinuierliche Aktualisierung der Komponenten

Fragen? Gerne auch später!

- ▶ Alexander Sosna <alexander.sosna@credativ.de>
- ▶ Adrian Vondendriesch <adrian.vondendriesch@credativ.de>

Weiterführende Informationen

- ▶ <https://elephant-shed.io>
- ▶ <https://credativ.de/postgresql-competence-center>
- ▶ <http://credativ.de/blog/postgresql-schulungen-bei-credativ>
- ▶ <https://thomas-krenn.com/de/loesungen/postgresql>