Running postgresql.org

DevOpsSaar
Saarbrücken, Germany

Magnus Hagander
magnus@hagander.net

Magnus Hagander

- Redpill Linpro
  - Principal database consultant
- PostgreSQL
  - Core Team member
  - Committer
  - PostgreSQL Europe
  - Infrastructure team

postgresql.org

A brief history

1996

- Single CVS server
- Single majordomo + sendmail server
- Hosted on FreeBSD jails
- Single datacenter

1997-2008

- Grew organically
- Individual services
- Few servers, mixed daemons
- Manually managed
- Unsustainable

2009-2010

- Analyze needs
- Rebuild for maintainability
- Rebuild for new needs
- Hosting options?

Challenges

- Availability
  - Global organization, global users
- Sustainability
  - Vendors that disappear
  - Or loose quality
- Maintainability
  - Limited resources
  - Time and money

Team

- 5 persons, 0 dedicated time
- Sweden, Austria, UK, US East, Chile

Core services

- Mailing lists
  - Most important resource
  - Including archives!
- Website & downloads
- GIT repository/access
- wiki
- Release build nodes
- Security

Other services

- Regional systems
- Event management
- IRC bots :)
- Buildfarm
- etc etc

Hosting

Hosting

- Self-hosted?
- Full service cloud
    - Why not just use <site du jour>
- Partial-service cloud
    - IaaS, PaaS etc

Self-hosting

- Maximize control
- Maximize flexibility
- Maximize sustainability
- More dedicated work
  - Or?

Full service cloud

- Zero control
- Lack of services
- Uncertain sustainability
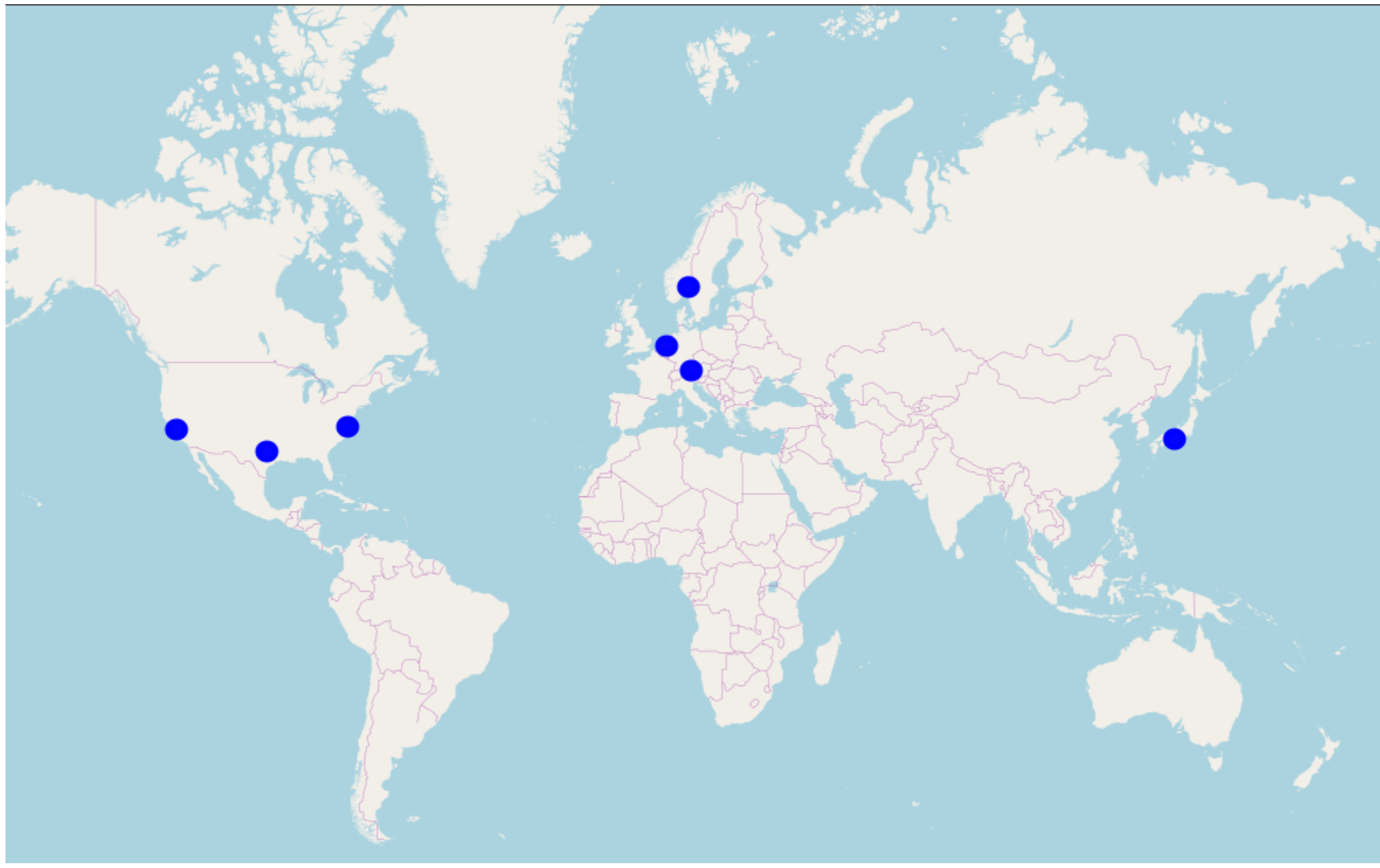 - Remember sourceforge?
- Minimal work!

IaaS/PaaS

- Similar to self-hosting
- Much more expensive
  - But we don't pay…
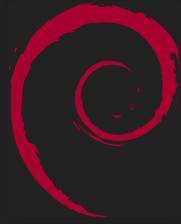- Limited vendors
  - We don't pay *today*

Conclusion

- Self-hosting
- Except not

Where we are

How does it work?

How does it work?

Limiting platforms

- *Drastically* simplifies things
- Much for free
- Config mgmt very straight forward
- Very little overhead

Everything is a package

- Everything is a DEB
- Two repositories
  - Controlled mirror of debian repos
  - Custom pginfra repo

Small web service

- Trivial django app
- Almost all edits through /admin/
- Config file templating
- Metadata exchange
- Custom CA
  - Everything is https + cert

Fundamental blocks

- VMs
- User management
- Package/config management
- Monitoring

VMs

- Simple KVM VMs
- Automatically provisioned
  - Using virt-install
  - *Not* imaged
- Bridged networks
- LVM disks

User management

- Few shell users
- Enforced ssh keys
- Enforced jumphosts
  - Host
  - Network

Packages & config

- Tools just installed
  - apt-get install
- Any service requires *wrapper*
  - e.g. pginfra-wrap-rsyncd
- Any multi-node service has native package
  - pginfra-pgweb

pginfra-wrap-rsyncd

```
├── DEBIAN
│   └── control
├── etc
│   └── rsyncd.conf
└── pginfra
    └── packageconfig.d
        └── rsyncd.py
```

pginfra-wrap-rsyncd

```
Package: pginfra-wrap-rsyncd
Version: 1
Section: base
Priority: optional
Architecture: all
Depends: rsync
Maintainer: webmaster@postgresql.org
Description: PostgreSQL Infrastructure wrapper package for rsyncd.
```

pginfra-wrap-rsyncd

```python
def allowed_services():
    return ['rsync', ]


def allowed_listeners():
    return [873, ]


def autobackup():
    yield '/etc/rsyncd.conf'


def backup_excludes():
    yield '/var/log/rsyncd'
```

Config "just python"

```python
def autobackup():
    if os.path.isfile('/etc/postgresql-common/pg_service.conf'):
        yield '/etc/postgresql-common/pg_service.conf'
    for v in os.listdir('/etc/postgresql/'):
        if re.match('^[\d\.]+$', v):
            # Looks like a version
            for root, dirs, files in os.walk(os.path.join('/etc/postg
                # Autobackup automatically takes care of directories,
                # just care about the files
                for f in files:
                    yield os.path.join(root, f)
```

Full packages

```
├── DEBIAN
│       ├── control
│       └── postinst
├── etc
│       └── systemd
│           └── system
│               └── varnishqueue.service
└── pginfra
        └── packageconfig.d
            └── pgweb.py
```

Full packages

```
Package: pginfra-pgweb
Version: 8
Section: base
Priority: optional
Architecture: all
Depends: pginfra-wrap-django, pginfra-wrap-nrpe-server, python3-crypt
Maintainer: webmaster@postgresql.org
Description: PostgreSQL Infrastructure wrapper package for the pgweb
```

Full packages

```python
def process_config():
    replace_file_from_template(
        '/pginfra/resources/pginfra-pgweb/settings_pginfra.py',
        '/usr/local/pgweb/pgweb/settings_pginfra.py',
    )

def nrpe_commands():
    return [
        ('check_varnishqueue', 'pgservice',
         'varnish purge queue',
         '/usr/local/pgweb/tools/varnishqueue/nagios_check.py "dbname
    ]
```

Deploying

- *apt-get install...*

Monitoring

Monitoring

- Nagios
  - (yes, still, and it's old)
  - Explicitly from external system
  - Includes security monitoring
- Prometheus + Grafana
- Log scanning

Security patching

- Mainly relies on Debian
- Constantly monitored
  - Including restart-needed!
- Distributed patching
  - It's just *apt-get upgrade*

Software

Software

- website
- git services
- wiki
- mailinglists
- buildfarm

website

- Custom django code
- PostgreSQL (d'uh)
- Varnish
  - Geo-distributed

Community authentication

- Single "postgresql.org identity"
- Specific for internal use
- Easier than oauth
- But less capable
  - Should be extended

git services

- Separated "hidden master"
- SSH auth for committers
- Public mirror
  - Wrapper authentication
- Simple commit msgs and policy enforcing

wiki

- Mediawiki
  - On PostgreSQL
  - Custom skin
  - Custom auth
- Ugghh....

mailinglists

- Used to run majordomo + mhonarc
- Both unmaintained

pglister

- Custom list manager
  - Just over 2kloc
- Web driven
  - Another 3kloc
  - Mostly boilerplate
- PostgreSQL centric

Buildfarm

- Distributed regression testing
- Statistics and collections
- Many platforms!

Summary

Stats o the day

- Datacenters: 7
  - (plus some cloud)
- Dedicated servers: 12
- VMs: 74
  - ~60 prod
- People: 0
  - (3)

Summary

- Automate *everything*
- Rely on existing platforms
- But don't be afraid to self-build

Thank you!

Magnus Hagander
magnus@hagander.net
@magnushagander
https://www.hagander.net/talks/