Automate/Script all the things

(or push button)

From the DBA's point of view

How much stuff you got there?

- When we first started out
  - < 10 Databases, 4 Test environments, and < 20 Ddevelopers
- Now
  - >2700 Databases, 50+ Test Environments, >100 Developers

About Me

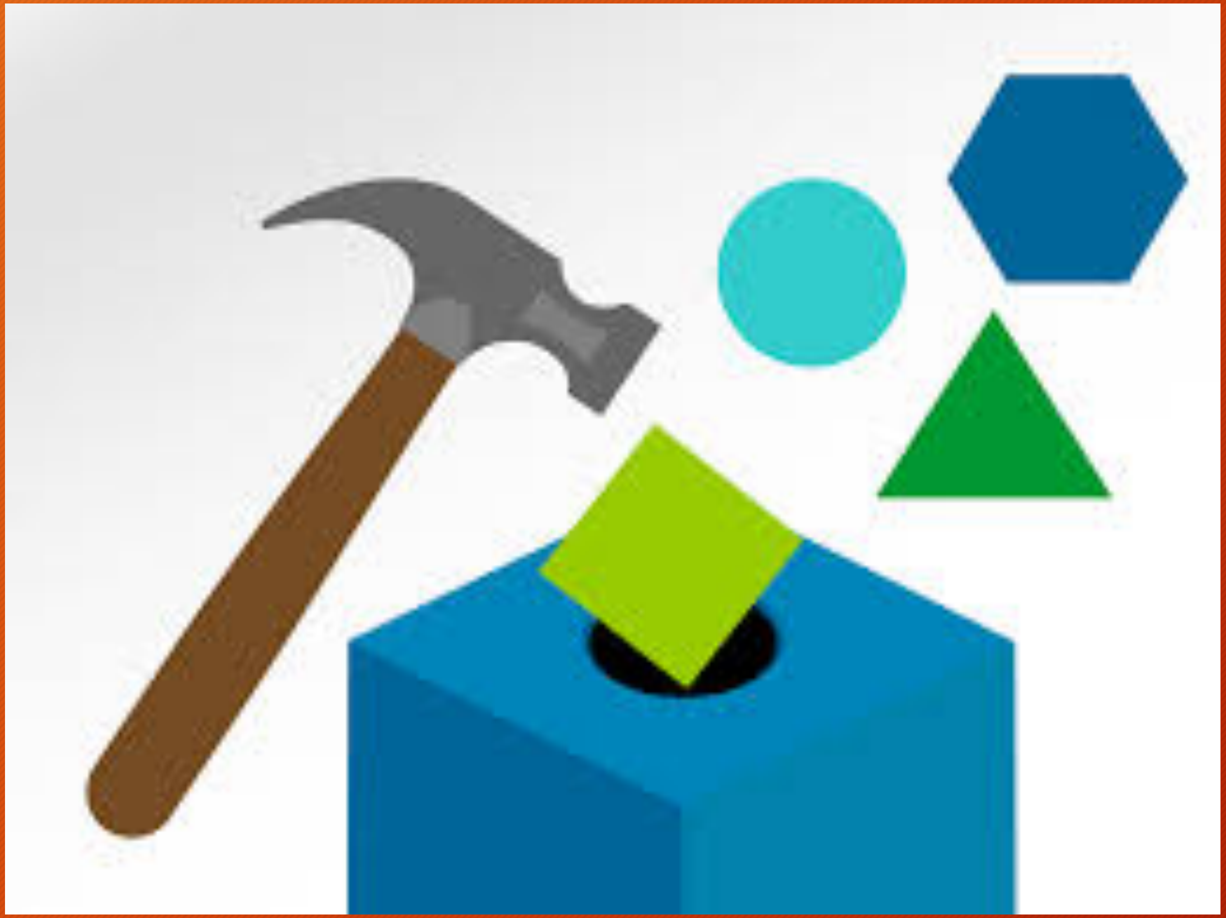
- JP O’Donnell
- LinkedIn profile jp-o-donnell-12996018
- Tech world for a while
  - Network Engineer
  - Tier 1-3 support
  - FNG
  - Developer
  - Hadoop
  - DBA

The DBA Responsibilities

- Keep environments running
- Configuration of not databases, datbase servers, and the hosts they run on
- Monitoring
- Creation and maintenance of our environments
- Working with developers

Snow flakes?

One size fits all?



Define the template

- Where do you draw the line
- What works today, probably won’t work tomorrow
  - But make it easy to change, ”scalability"

Templates

- Systems
  - Installed services
  - IP Configs
  - Domain
- Servers
  - Memory
  - Disk
  - Cores
- Database configs
  - Database version, installation location, etc…
  - Config files
  - Connection options

Jenkins

MKVM

- Automation of creating servers
- Templates for each size
  - With over writeable options
- Takes care of the network configuration
  - DNS Entries
  - IP Address
- Registering of server

But wait, there’s more in Jenkins

- Testing
  - Application
  - Environment
  - Lights on
- Remove VM
- Quick status

We wanted more

- Certain systems needed slightly different options
  - Monitoring
  - Alarming
  - Connections
  - Cache settings

puppet

Modules

- Creation of users
- Configuration files
- Server versions
- Deployment of changes
- Script it out vs Puppet
    - Speed of change/new things

Share what we do

- https://github.com/covermymeds
    - Puppet
    - Postgres
    - Dbdeployer
    - Etc..

Monitoring

- Nagios
- Shinken
- Module to maintain the checks, and allow for easy overwritten of the checks
- Easy way to silence things while doing maintenance

Analytics

- OLTP vs OLAP
- CAAPI
  - As we grew it grew
  - Resource availablility
- Pager alerts
  - One script to check them all

Password

- Changing usernames
  - Automatic rotation
    - Possibly in the future
  - Template
    - Easy 1 step change
- Generation of passwords
  - Easily scriptable

ETL

- Why ETL
  - Reference data
- Drawbacks
  - Fragile
- Analytics
  - SQL server vs postgres

ETL

ETL Design

- Production to masters
  - Pull as little as possible from Production
- Masters to children
  - Staggard pulls
  - Keep track of schedules
  - One source of truth
- Why all the things?

Permissions

- Multi databases
- Read vs Write
- Multible schemas
- Easy to manipulate

I talk to the damn customers

Developers

- Didn’t have to submit
  - Review optional
  - Not speedy
- Version control?
  - Merging nightmare
  - Different version of databases everywhere

Human sacrifice! Dogs and cats
living together. Mass hysteria!

- Implement a process
  - Pull Request review
  - Github

Manage all the database

- DDL changes
- DML changes
- Stored Procedures
- Functions
- Initial designs

But Why?

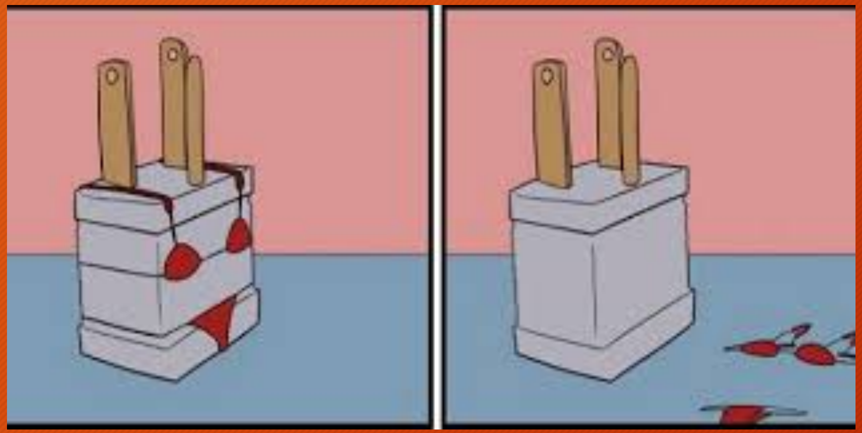
- Better to take the time upfront
- Avoid technical debt
- ~~Gives us a sense of power~~
- Standards

Evolved

- 2 DBAs
  - 100+ devs
  - Process in place to keep up
- Future
  - Github approval tag
  - Auto deploy

Dbdeployer

- https://github.com/covermymeds/dbdeployer



Anyone can use

- No manually typing
  - Pulls directly from version control
    - Currently github
  - Simple options
- Once merged puppet takes over

What’s next

- Solutions need to be for now and the future
- Narrow solutions don’t work
- No Snowflakes

Change Data Capture

- ETL of data, the problem
  - How do you know what changed
  - Created_at, Updated_at

- Database keep track instead of ETL Tool

- Secondary tables
  - Different Schema
  - Primary keys
    - Can’t hard delete

Now what?

- Backups
  - Auto restore
- Environments
  - Auto build/destroy

Thanks

- Questions?
- https://github.com/covermymeds/dbdeployer
- https://github.com/covermymeds