Emergency Preparedness

Creating a Disaster Recovery Plan for your Drupal Site

Ronan Dowling – Gorton Studios/NodeSquirrel.com
DrupalCorn 2014

Who am I?

- Lead developer at Gorton Studios
- Maintainer of Backup and Migrate
- Founder/Creator of NodeSquirrel
- ronan@gortonstudios.com
- @ronan4000

Who are you?

- Drupal site owners and builders
- Drupal shops
- Small to medium sites

Who are you not?

- Sysops/Devops
  - You already know this stuff
- Enterprise or large sites
  - These tools may not scale up

What is a DRP?

"A disaster recovery plan (DRP) is a documented process or set of procedures to recover and protect a business IT infrastructure in the event of a disaster."

3 Basic Features:

1. preventive measures
2. detective measures
3. corrective measures

– http://en.wikipedia.org/wiki/Disaster_recovery_plan

Typical advice

- Write down every possible scenario
- Write down the solution to every problem
- Practice!

Less intimidating approach

- Identify all the things that can fail
- Figure out how to replace them
- Practice!

The parts

- Domain Registrar
- Authoritative Name Servers (DNS)
- Host Network
  - (load balancers, front end cache)
- Web Server(s)
- Drupal and Modules
- Database(s)
- Uploaded Files

Risks to your site

- User Errors
- Bad Services
- Hackers
  - Intrusion
  - DDOS
- Success
  - The Reddit Hug/Slashdot Effect
- Natural Disasters

What do you need to do?

1. Preventive measures
2. Detective measures
3. Corrective measures

Preventative Measures

"Controls aimed at preventing an event from occurring."

- http://en.wikipedia.org/wiki/Disaster_recovery

- Use Drupal security best practices
- Use good vendors
  - Host, registrars etc.
- Build in redundancy
- Train your users

Preventative Tools

CloudFlare

- http://cloudflare.com
- CDN/DNS/Front-end Cache
- Protects from hackers
- Prevent DDOS (intentional or unintentional)
- Free or $20+/mo
- See also: Incapsula –
  - http://www.incapsula.com/

Hosted DNS

- Amazon Route 53
- dnsbycomodo.com
- dyn.com
  - “Outsourcing DNS is part of a sound disaster prevention strategy.”

http://en.wikipedia.org/wiki/List_of_managed_DNS_providers

- Some protection from DDOS
- Better uptime (than cheap registrars)
- Actual redundancy

Detective Measures

"Controls aimed at detecting or discovering unwanted events."

- http://en.wikipedia.org/wiki/Disaster_recovery

- Don't wait until your users tell you your site is down.

Detective Tools

Pingdom

- http://pingdom.com
- Uptime monitor
- Visits your website periodically
- Emails you if the site is down
- Free for 1 site or $14+/mo for more
- See Also:
  - UptimeRobot
  - Mon.itor.us

Application Monitoring

- New Relic/Naigos/Appneta
- Checks the health of the server
  - Resource usage etc.
- Detect problems before they’re critical
- Installed on your server
- Talk to your host

Wormly

- http://wormly.com
- Application monitoring for the rest of us
- Install a PHP script on your server
- Reports and tracks usage (memory, cpu, etc.)
- $19+/mo

Drupal Monitor

- http://drupalmonitor.com
- Drupal-specific app monitoring
- Install a module
- Reports various site stats
- Track multiple sites
- Free (freemium coming)

Corrective Measures

"Controls aimed at correcting or restoring the system after a disaster or an event."

- http://en.wikipedia.org/wiki/Disaster_recovery

- The meat of the DRP

Corrective Tools

Backup!

Redundancy for data

4 Components of Drupal

- Server Configuration
- Code
- Database
- Uploaded Files

Server Configuration

- Changes almost never
- Not too hard to recover without backup
- Difficult to back up
- Ask your host
- Keep a record of custom configuration

Drupal Code

- Changes rarely
- Sometimes possible to recover without backup
- Most of it is on drupal.org/github etc.
- Should be in a VCS
  - git, svn
- Automate Deployment (dploy.io)

Database

- Changes frequently
- Impossible to recover without backup
- Easy to backup
- A few MB to a few GB
- Tools:
  - Backup and Migrate
  - phpMyAdmin
  - MySQLDump

Uploaded Files

- Change infrequently
- Difficult-ish to recover without backup
- Relatively difficult to back up
- Hundreds of MB+
- Restoring is slow
- Tools:
  - Backup and Migrate (3)
  - Rsync
  - Custom scripts

Levels of Backup

Server level vs Application level

Server-level backup

- Provided by hosts
- Backs up config/db/code/files
- Slow to recover
- Dependant on host/sysop
- Best for total system failure

Application-level Backup

- Backup Drupal DB and Files
- Controlled by site owner/admin
- Recover in seconds
- No support tickets needed
- Best for user error and partial failure

Content-level Backup

- Per-node versioning
- Recover specific nodes/entities
- Built in to Drupal core
- Best for: localized user error
- Not good for: Things that aren’t entities. Deletes.

Offsite vs Onsite Backup

Onsite Backup

- Quickest Backup
- Quickest Recovery
- Not good for system failure

Offsite Backup

- Slower to backup
- More effort to set up
- Available when your server is down

- Offsite backup options
  - NodeSquirrel
  - Amazon S3
  - FTP to another host
  - Email (DON’T DO THIS)

- Offsite backup from your host is NOT offsite

Restore

Restoring your site

- Depends on your backup solutions
- Depends on how ‘down’ your site is
- Practice
- Time your practice

Accessing Services

Know how to log-in in an emergency

Keep all logins together

- Web host, Registrar, DNS, CDN, etc.
- Store online and offline

Store tech support contacts

- Web host, Registrar, DNS, CDN, etc.
- Don’t rely on the company’s ticketing system
  - Also store email, phone, twitter

Email password reset

- Have all account password reset to same email
  - Don’t use a real user’s email
  - Don’t use your website’s domain/server
  - Forward to anybody who might need to recover
  - Consider 2-factor auth
- Test resetting passwords

Your written plan

- A list of 3rd party services with:
  - Login credentials
  - Account email
  - Support contacts
- A list of internal people responsible for recovery
- The location, type and frequency of every backup

Questions?

ronan@gortonstudios.com

@ronan4000