Enterprise Git - the hard bits

Matthew Barr

Enterprise Git - the hard bits

Matthew Barr, Architect

Overview

- Introduction
- Git hosting options
- o18n
- Safety & Best Practices

Admissions

Not

Lawyer

Not

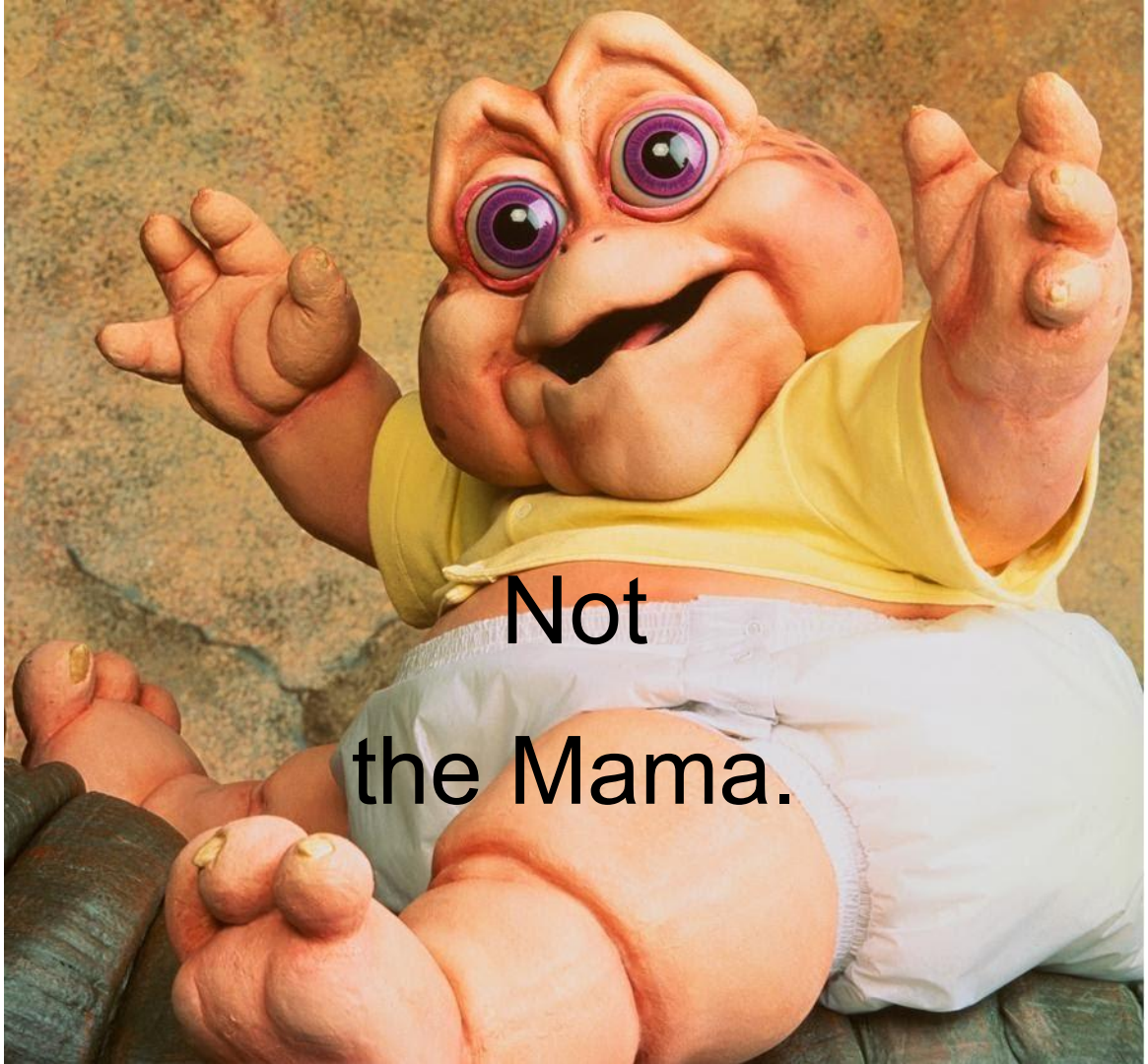
Compliance

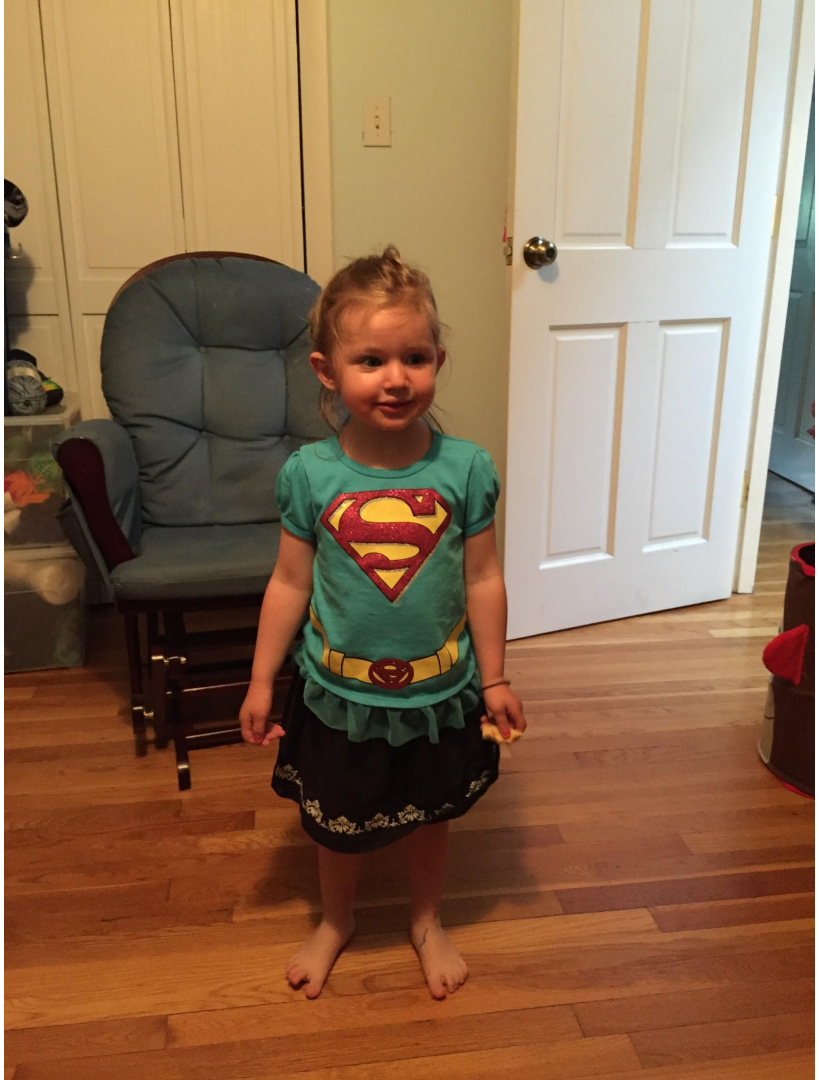
Not

Internal Audit

Not

PCI Assessor (QSA)





Me:

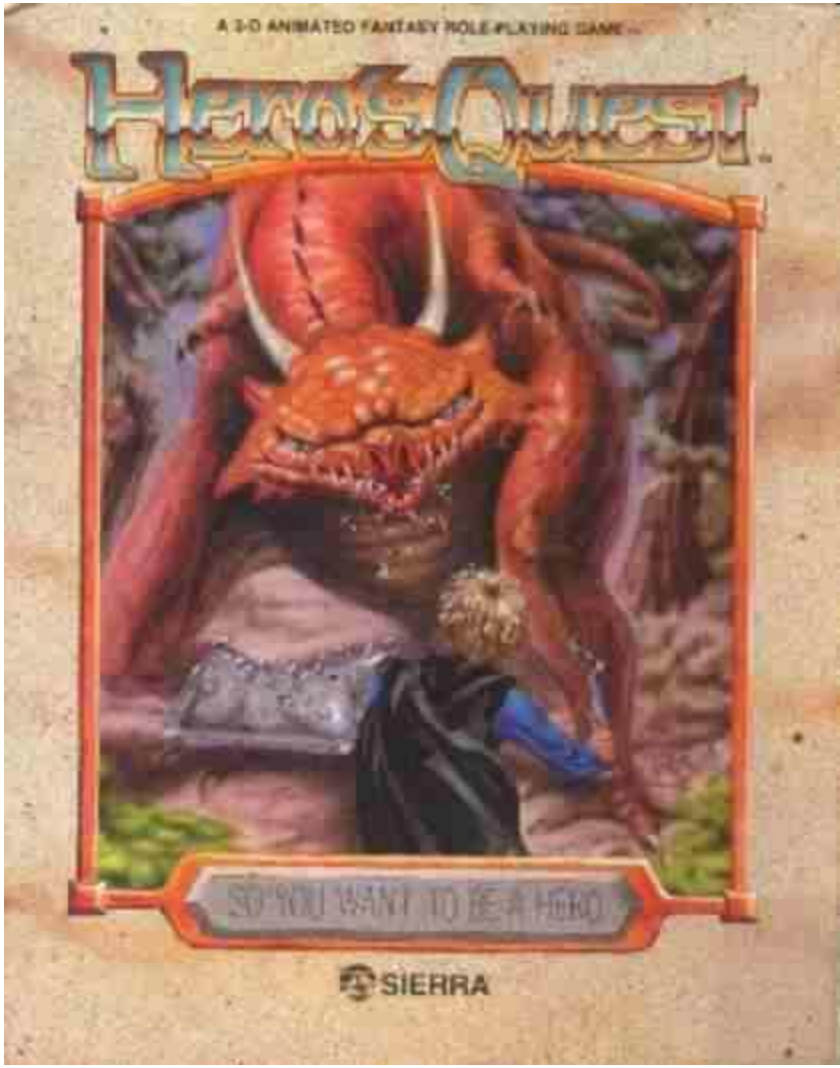
SysAdmin / DevOps Engineer for 20 years

- Lehman Bros, MarkitServ
- Community Connect, Snap Interactive
- Nokia

Focus @ Akamai: Developer Productivity

- Provide tools for our engineers
- SCM, Build, CI & Test systems
- Current project: Horizontally scalable build farm w/ Docker agents

So you want to be
~~a hero~~
store your code in Git



GitHub or Bitbucket

- Hosted
- Great features
- Low overhead

- Great for small teams
- Even medium size

Self hosted options

- GitLab
- Gitolite
- cgit

Enterprise

- Github Enterprise
- Bitbucket Server (Atlassian) (née Stash)
- Gitlab Enterprise
- Perforce GitSwarm

Git @ Akamai

- Currently: 6000+ repositories, 115+ Projects/Organizations
  - Not primary code repository (yet)

- Relaunched 1 year ago

- Stash Data Center Edition
- 2 sites
  - 2 App Servers
  - 2 DB nodes
  - Netapp filer & load balancer

o16n (Operationalization)*

* Gordon Marx

HA, DR, GeoDiversity & Backups

- Varies by product
- Github Enterprise
  - Clustering
  - Active / Passive Node
  - Point in time snapshots
- Bitbucket Server
  - Self Service Backups, DB replication, Snapshots
  - Improvement in Bitbucket Server (Stash)
    - Smart Mirrors
    - Zero Downtime Backups

Authentication for the enterprise

- Mandate: No passwords
- 3 types of access
  - WebUI
  - Git (SSH, HTTPS)
  - API

- SAML for WebUI
- SSH key sync script from LDAP
- X.509 Client auth for API

Safety & Best Practices

PCI, SOX, etc.

Boils down to:

- Prevent unauthorized changes
- Review change!

Code Review - Pull Requests

- Sign offs - +1, approvers
- Prevent merges without PR’s
- Merge commits
  - Audit points, in git log

Code Integrity

- Branching workflow
  - Combination Gitflow + Feature Branch (Github)
    - No Develop branch, but flexibility for QA
    - Can be CD
- Protected branches
  - Limited users can merge
- No force push / rewriting history
- Unapprove PR’s when modified
  - Really? Provided by optional plugin?

Q: Who wrote that code?

- Pusher != committer
- Committer

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

- GPG?
- Log all commits/pusher?

Access Control

- 1000’s of repos = 1000’s of ACLs
- Organizations / Projects
- LDAP groups?
- Access Controls
  - Who manages, approves access?
  - Audits access, quarterly?
- Separation of Concerns
  - Ops can’t modify code
  - Prove it!

Automation

- API’s!
- Configure
- External Front Ends
  - User Mgmt
  - Webhooks
  - Audit settings

References

- Github Enterprise Documentation
- Bitbucket Server Documentation

Matthew Barr

- https://www.akamai.com
- mbarr@akamai.com
- @matthewbarr - Twitter & Github:
- mbarr@mbarr.net


Akamai
FASTER FORWARD