

# Camps

## Development Platform

“The Search For A Silver Bullet”

# Introduction

- Not a programmer
- Linux
- Started with Interchange 2001/2002
- Taurus...
- Moving along nicely

# Time to Get Serious!

- 2008 things get complicated
- Growing competition
  - Easy to have a basic site
  - PHP carts
- Little or no loyalty on web
  - Price

# Time to Get Serious!

- SEO is getting complex
  - Duplicate content
- Need new features
- Development strategy?

# Time to Get Serious!

## Development Options

- Live Edits
  - Bad Idea
- VM's
  - Resources
  - Time to configure
  - Managing code base

# Time to Get Serious!

- Has to be a better way
- We need a silver bullet!
- I'm calling Endpoint
  - Ron Phipps
  - Sorry Sam BUT..

# The Silver Bullet?

- Camps ([icdevgroup.org](http://icdevgroup.org))
  - Git version control
  - Separate web server, database, IC, etc
  - Lightweight/fast
  - Multiple projects/developers
  - Git version control
  - Perl based
  - Ethan Rowe and Jon Jensen

# How It Works

- Git project minus config files

- Set values in local-config

dns\_domain:westbranchangler.com

hostname: \_\_CAMP\_NUMBER\_\_.camp.\_\_CAMP\_DNS\_DOMAIN\_\_

- Tokenize config files for all services

ServerName \_\_CAMP\_HOSTNAME\_\_



# How It Works

- Local copy of production db dump
- Camps pulls source into users camp
- Writes configs, populates db and starts all services. YES!



# How It Looks

```
-- ~camp/  
  |-- angler/  
    |-- angler.git/  
      |-- local-config  
      |-- camp-config-files  
      |-- mysql/  
        |-- my.cnf  
        |-- tmp/  
        |-- angler.sql  
      |-- etc/  
        |-- interchange/  
          |-- interchange_local.cfg
```

# Simple Easy Perfect....Not Exactly

- Not simple to setup
- ALL environment details and paths

# Simple Easy Perfect....Not Exactly

- Must be very thorough.  
Example snip of bin/interchange

```
#!/bin/sh
....
icroot=__CAMP_PATH_/interchange
pidfile=$icroot/var/run/interchange.pid

export MINIVEND_ROOT=$icroot
export INTERCHANGE_OTHERLIBS=$icroot/
    custom/lib
exec $icroot/bin/interchange.bin \
    -pidfile $pidfile \
    PIDfile=$pifile \
    "$@"
```

# Good News

- Once setup easy to reuse camp “template”
- Development environment is available in minutes.

# Good News

- Taking risks isn't a problem  
#rmcamp --number=12  
#mkcamp --comment "This is easy"
- Not true with VM.

# Experiences

- Camps use single local Perl. /usr/local/bin/perl.
- Test a more modern Perl on a camp?
- Racke/NITESI the quest for Interchange 6
  - Dancer should have perl 5.14+
  - Nginx/Plack running local on Centos 5 perl 5.8
  - Interchange running on 5.10

# Experiences

- CentOS 6 5.10.1 **BAD**
- Endpoints CentOS 6 repo 5.14.1 **BETTER**
- Modern Perl needs modern tools
- Can't we all just get along?



# Assumptions

- Carton - Perl module dependency manager
  - <http://search.cpan.org/~miyagawa/ Carton/>
- Perlbrew: admin-free perl installation management tool.
  - <http://perlbrew.pl/>

# Perlbrew

- Perlbrew = Perl environment flexibility
- System Perl nice and clean.
- Multiple Perls easy

```
[camp@c1]$ perlbrew list
* perl-5.10.1
  perl-5.14.4
```

- Changing Perl is simple

```
[camp@c1]$ perlbrew switch perl-5.14.4  
perl-5.10.1  
* perl-5.14.4
```

# Perlbrew + Camps

- Install Perlbrew in camp home
- Define environment in .bashrc

```
export PERLBREW_ROOT=/home/camp/perl5  
source ${PERLBREW_ROOT}/etc/bashrc
```

# Perlbrew + Camps

- Make shebangs flexible `#!/usr/bin/env perl`
- This isn't magic set your `%ENV`

# Perlbrew + Camps

```
#!/bin/bash
```

```
## clear
```

```
unset PERL5LIB;
```

```
## set perlbrew env
```

```
export PERLBREW_ROOT='/home/camp/perl5';
```

```
export PERLBREW_HOME="$HOME/.perlbrew";
```

```
source ${PERLBREW_ROOT}/etc/bashrc;
```

```
## set camp perl here
```

```
perlbrew switch perl-5.14.3;
```

```
## set local libs
```

```
export PERL5LIB=/home/sam/camp50/interchange/custom/lib"${PERL5LIB:+:$PERL5LIB}";
```

```
export PERL5LIB=/home/sam/camp50/interchange/lib"${PERL5LIB:+:$PERL5LIB}";
```

```
export PERL5LIB=/home/sam/camp50/local/lib/perl5"${PERL5LIB:+:$PERL5LIB}";
```

# Carton + Perlbrew

- We have Perl now we need the modules
- Carton to the rescue
- In the camp home give Perlbrew users Carton  
#perlbrew install-cpanm  
#cpanm Carton

#z

# Carton + Perlbrew + Camps = 8-)

- Create module list camp12/cpanfile

```
requires 'HTTP::Parser::XS', '0.007';  
requires 'HTTP::Server::Simple';  
requires 'HTTP::Server::Simple::PSGI';  
requires 'Hash::Merge';  
requires 'Hash::MultiValue';  
requires 'Image::Magick';
```



# Carton + Perlbrew + Camps = 8-)

- Build routine install carton?
- Wow look at how long it takes my camp to build...

Successfully installed Scalar-List-Utills-1.32 (upgraded from 1.23)

Successfully installed namespace-clean-0.24

Successfully installed Type-Tiny-0.026

Successfully installed Exporter-Tiny-0.026

Successfully installed PadWalker-1.96

Successfully installed lexical-underscore-0.001

Successfully installed Scalar-Does-0.201

Successfully installed Params-Validate-1.08

# Carton + Perlbrew + Camps = 8-)

- Better options? **Yes**
- #carton bundle bundles cached tarballs
- locate vendor/cache in ~camp then symlink into camp?
- Add vendor/cache to git?
- install -cached

# Camp Domination

- Full independent services instead of sharing system apache.
- Compile services Nginx, Plack
- Control of all services, init.d scripts
- Test current stable versions

# Camp Domination

## Master.pm

```
sub run_camp_scripts {  
    my ($c_number) = @_;  
    warn "Reading camp install file though already initialized...\n" if $initialized;  
    my $file = File::Spec->catfile( type_path(), 'camp-install-scripts', );  
    open(my $INSTALL, '<', $file) or die "Type-specific config file missing: $file\n";  
    while( my $line = <$INSTALL> ) {  
        my @args = ( "bash", "-c", "/home/$camp_user/camp$c_number/$line" );  
        system(@args);  
    }  
    close $INSTALL or die "Error closing $file: $!\n";  
}
```

# Camp Domination

## Example install-services.sh

```
#!/bin/bash
```

```
## This script installs services during camp creation
```

```
cd __CAMP_PATH__
```

```
# let carton install all proper modules
```

```
carton install
```

```
# lets make sure we are using the right perl
```

```
source __CAMP_PATH__/etc/perlbrew_config.sh
```

```
CONFIGURE_PARAMS="--prefix=__CAMP_PATH__/etc/nginx --without-http_rewrite_module --blah blah blah
```

```
NGINX_SRC="__CAMP_PATH__/src/Nginx"
```

```
cd ${NGINX_SRC}
```

```
make clean
```

```
echo "Configuring Nginx..."
```

```
./configure ${CONFIGURE_PARAMS}
```

# What's Next?

- Camps as production deployment. Camp-hq?
- <https://github.com/sbatschelet/interchange-camps>
- Camps + Interchange and Josh Lavin's new strap template  
<https://github.com/interchange/interchange>
- <https://github.com/devcamps/camps>
- <https://github.com/perusion/strap>
- Could use some input/help 😊

# Conclusion

- Even basic carts should have a development environment
- Camps is very scalable and just plain awesome.
- Camps should be part of Interchange get developers started out of the box.
- Backcountry uses it, Jon Jensen and Ethan Rowe wrote it.



# Questions?

