Tips, Tricks and Tools for managing one or more medium to large CiviCRM installs

CiviCon 2012
Maintenance best practices
Drush & “drush make”

drupal.org/project/drush
drupal.org/project/drush_make
Configuring environments

Sysadmin tasks

- Apache / nginx
- php5-gd, php5-apc, ..
- Mysql
- Nagios checks
- Munin graphs
- Backups

Webdev tasks

- Deploy code (Drupal core, modules, CiviCRM) using drush makefiles
- Create sites, upgrade, clone
- Moves sites from dev / stage / production
## Site tasks

- **Verify**

### Site tasks table

**Site** | **Profile** | **Language** | **Created** | **Platform**
--- | --- | --- | --- | ---
- test9.aegir.localhost | Standard | English | 22 weeks ago | CiviCRM 4.1-TRUNK
- site8.aegir.localhost | Standard | English | 22 weeks ago | CiviCRM 4.1-TRUNK
- site6.aegir.localhost | Standard | English | 22 weeks ago | CiviCRM 4.1-TRUNK
- site5.aegir.localhost | Standard | English | 22 weeks ago | CiviCRM 4.1-TRUNK
- site2.aegir.localhost | Standard | English | 22 weeks ago | CiviCRM 4.1-TRUNK
- site1.aegir.localhost | Standard | English | 23 weeks ago | CiviCRM 4.0.6 - 2
- aqaba.net4.bidon.ca | Hostmaster | English | 23 weeks ago | hostmaster-6.x-1.7

## Queues

### Queues table

- **Task**
  - Delete civi test
  - Verify aqaba.net4.bidon.ca
  - Verify hostmaster-6.x-1.7
  - Verify hostmaster-6.x-1.6
  - Verify civi test

### Actions

- **View**
### Platforms

<table>
<thead>
<tr>
<th>Platform</th>
<th>Release</th>
<th>Server</th>
<th>Verified</th>
<th>Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>CiviCRM 4.0 E-1</td>
<td>drupal 7.8</td>
<td>aqaba</td>
<td>23 weeks ago</td>
<td>1</td>
</tr>
<tr>
<td>CiviCRM 4.1-TRUNK</td>
<td>drupal 7.6</td>
<td>aqaba</td>
<td>22 weeks ago</td>
<td>5</td>
</tr>
<tr>
<td>hostmaster</td>
<td>drupal 5.22</td>
<td>aqaba</td>
<td>21 weeks ago</td>
<td>0</td>
</tr>
<tr>
<td>hostmaster 6.x-1.0</td>
<td>drupal 5.22</td>
<td>aqaba</td>
<td>14 weeks ago</td>
<td>0</td>
</tr>
<tr>
<td>hostmaster 6.x-1.7</td>
<td>drupal 5.25</td>
<td>aqaba</td>
<td>18 hours ago</td>
<td>1</td>
</tr>
</tbody>
</table>

**Total sites hosted:** 7

### Queues

<table>
<thead>
<tr>
<th>Task</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete errors</td>
<td></td>
</tr>
<tr>
<td>Verify aqaba.net4.bidon.ca</td>
<td></td>
</tr>
<tr>
<td>Verify hostmaster 6.x-1.7</td>
<td></td>
</tr>
<tr>
<td>Verify hostmaster 6.x-1.6</td>
<td></td>
</tr>
<tr>
<td>Verify civ test</td>
<td></td>
</tr>
</tbody>
</table>

More tasks
### Servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Database</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>aqaba</td>
<td>✗ No</td>
<td>✓ apache</td>
</tr>
<tr>
<td>localhost</td>
<td>✓ mysql</td>
<td>✗ No</td>
</tr>
</tbody>
</table>

### Queues

<table>
<thead>
<tr>
<th>Task</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Delete civi test</td>
<td>View</td>
</tr>
<tr>
<td>✓ Verify aqaba.net4.bidon.ca</td>
<td>View</td>
</tr>
<tr>
<td>✓ Verify hostmaster-6.x-1.7</td>
<td>View</td>
</tr>
<tr>
<td>✓ Verify hostmaster-6.x-1.6</td>
<td>View</td>
</tr>
<tr>
<td>✗ Verify civi test</td>
<td>View</td>
</tr>
</tbody>
</table>

More tasks
Create Platform

Name: *

Choose a unique descriptive name for your platform. You very likely want this to be something like "Drupal 6.20".

Publish path: *

The absolute path on the filesystem where the sites will be hosted. This needs to be created manually and initialized before your platform works properly. It also needs to be a unique path not already in use by a platform on any server. For example, run the following shell commands:

```
cd /var/aegir/platforms
drush dl drupal
```

Your publish path is the absolute path to the directory that gets created. Alternatively, you can specify a makefile below, and the platform will be created automatically if the path specified here does not exist.

You are still required to enter the absolute path above, as it will be treated as the target directory by the makefile.

Makefile:

The absolute path on the filesystem or public URL of a makefile that will be used to create the platform in the directory specified above. If the directory already exists, this file will be ignored.

Platform access control:

Grant access to this platform for relevant clients where required. Leave all checkboxes unchecked to grant all clients access to this platform.

Save
## Create Site

**Domain name:** test10.aegir.localhost

**Client:** admin

The client who this site belongs to.

**Install profile:**
- [ ] Drupal
- [x] Standard
- [ ] Testing
- [ ] Minimal

The type of site to install. The profile selected here determines the list of supported platforms below.

**Platform:**
- [ ] CiviCRM 4.0.6 - 2
- [x] CiviCRM 4.1-TRUNK

The platform you want the site to be hosted on. Not seeing a certain platform? Platforms shown are those that support the profile above. If a different profile is selected, this list may change automatically.

**Language:** English

The language of site being installed.

**Database server:**
- [x] localhost

The database server the site will use to host its content.
Aegir installation

- Add Aegir's Debian repository to APT
- “apt-get install aegir”

Aegir can import existing platforms/sites.

More information:
community.aegirproject.org
drupal.org/project/provision_civicrm
More fun stuff

- Aegir-up
  www.drupal.org/project/aegir-up

- Hosting CiviCRM cron
  drupal.org/project/hosting_civicrm_cron
Puppet
Puppet – node configuration

node "xyz.vps.example.net" {
    $nagios_parents = "vps-server-a.example.net"
    $service_level = "production"
    $apache2_ssl = 'enabled'

    include koumbit::vserver
    include koumbit::munin
    include koumbit::service::apache2
    include koumbit::service::mysql

    postfix::mailalias { "aegir": recipient => "mlutfy"; }

    nagios::service {
        "www.example.org_ssl":
            check_command => "check_virtual_https!www.example.org!/"
        "www.example.org_ssl_cert":
            check_command => "https_port_cert!443!21";
    }
}
backupninja::rdiff { "main":
    installkey => false,
    host => "backup.koumbit.net",
    keep => "5D",
    type => "remote",
    home => "/backup/$fqdn",
    user => "backup-$hostname",
    include => [ "/var/spool/cron/crontabs",
                 "/var/backups","/etc", "/root", "/usr/local/*bin", "/var/lib/dpkg/status*",
                 "/var/lib", "/var/aegir " ],
    key => "AAAAB3[...]",
    backupkeytype => "dsa",
}
class koumbit::sudo {
    package { 'sudo': ensure => installed, } 
}
class koumbit::sudo {
  package { 'sudo': ensure => installed, }

  file { "/etc/sudoers":
    mode => 440, owner => root, group => 0,
  }
  if $operatingsystem == 'FreeBSD' {
    File["/etc/sudoers"] {
      path => "/usr/local/etc/sudoers",
      source => [ "puppet://$servername/secrets/sudoers.$fqdn", "puppet://$servername/secrets/sudoers.FreeBSD" ],
    }
  } else {
    File["/etc/sudoers"] {
      source => [ "puppet://$servername/secrets/sudoers.$fqdn",
                  "puppet://$servername/secrets/sudoers"
                 ],
    }
  }
}
class kouumbit::service::mysql {
  package { [ "mysql-server", "logtail" ]:
    ensure => "installed",
  }

  backupninja::config { conf:
    reportsuccess => false,
    reportwarning => false,
    when => "everyday at 23:40"
  }
}

backupninja::mysql { "main":
  sqldump => true,
  compress => true
}
Nagios & Munin
Questions?

http://ourpowerbase.net
http://www.koumbit.org