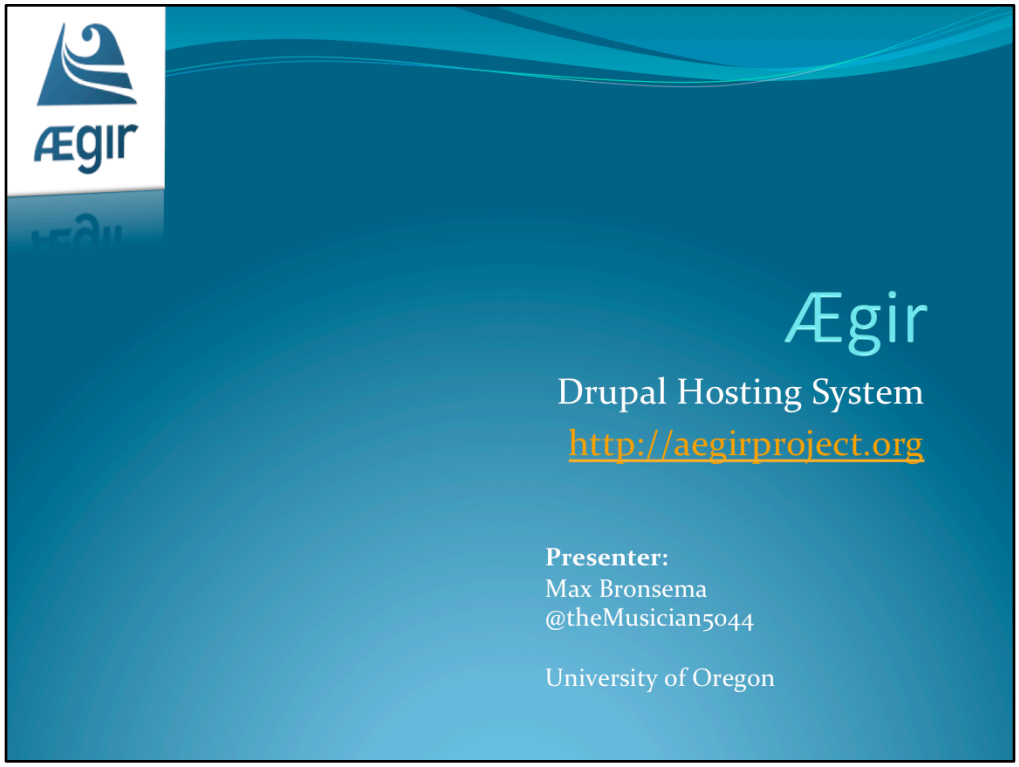
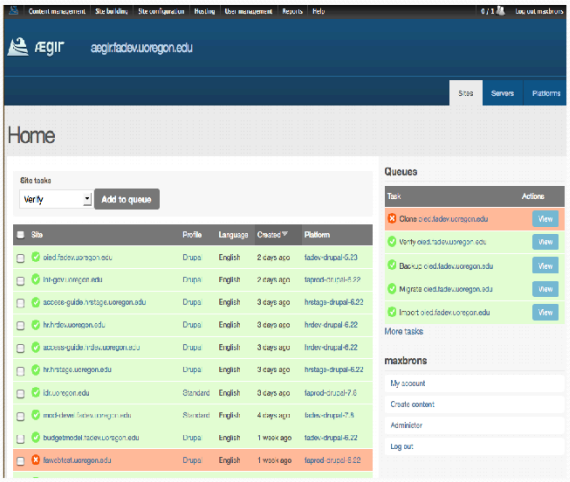


Introduction: Aegir is a graphical frontend for administering Drupal installations. The name is derived from Norse mythology where Aegir is the god of the sea, and if Drupal is a drop of water, then Aegir can rule them all.

Overview

1. Aegir
2. Install & Key Concepts
3. Benefits
4. Challenges
5. Live Demo
6. ?'s & Comments



Project Overview

- Posted to D.O. in 2007
- Many iterations, now at Version 1.4
- Supported by Koumbit.org, Computer Minds, mig5, and the community

Coolidge, Olivia, E. Legends of the North. 1951. The Riverside Press. Cambridge. MA. Illustrated by Edouard Sandoz.

Project Goals: ?

Ease of Management

Create backups, disable/delete sites, tracking of all modules installed within a multi-site installation. Manageable upgrades. Perform bulk tasks.

Rapid Installations and Deployments

Fill out a form and spin up a brand new install of either a d5, d6, or d7 site with the click of a button. Copy your completed test site to a staging site with three clicks, or to the live site.

Manage Multiple Servers

Aegir allows us to move sites around servers with ease. This means rapid prototyping, quick development, and a much easier time of staying current with patches.

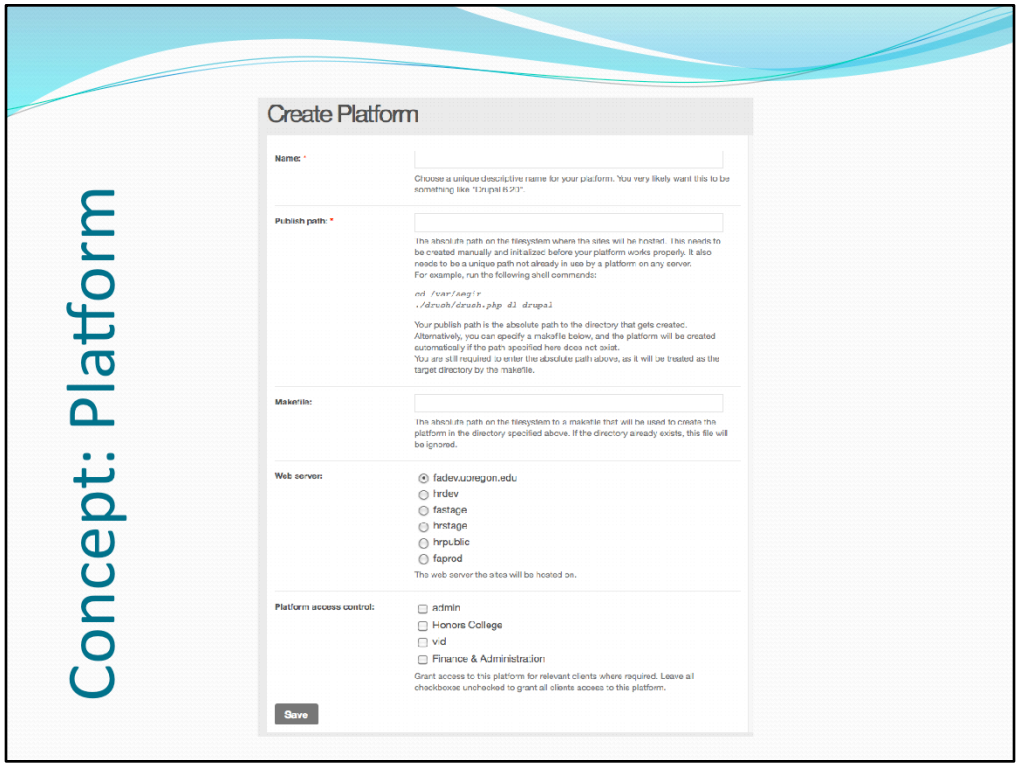
Aegir Components

- Backend: Provision
- Frontend: Hostmaster
- Magic Sauce: **Drush**
- Installation is fairly straightforward

To make this all work your hosting environment has to meet several requirements:

- LAMP/LEMP stack (Windows need not apply)
- Unix based OS
- Full control of the box ~ a VPS is great
- PHP 5.2 or greater, we use 5.33 with no problems
- Sudo, rsync, git and unzip

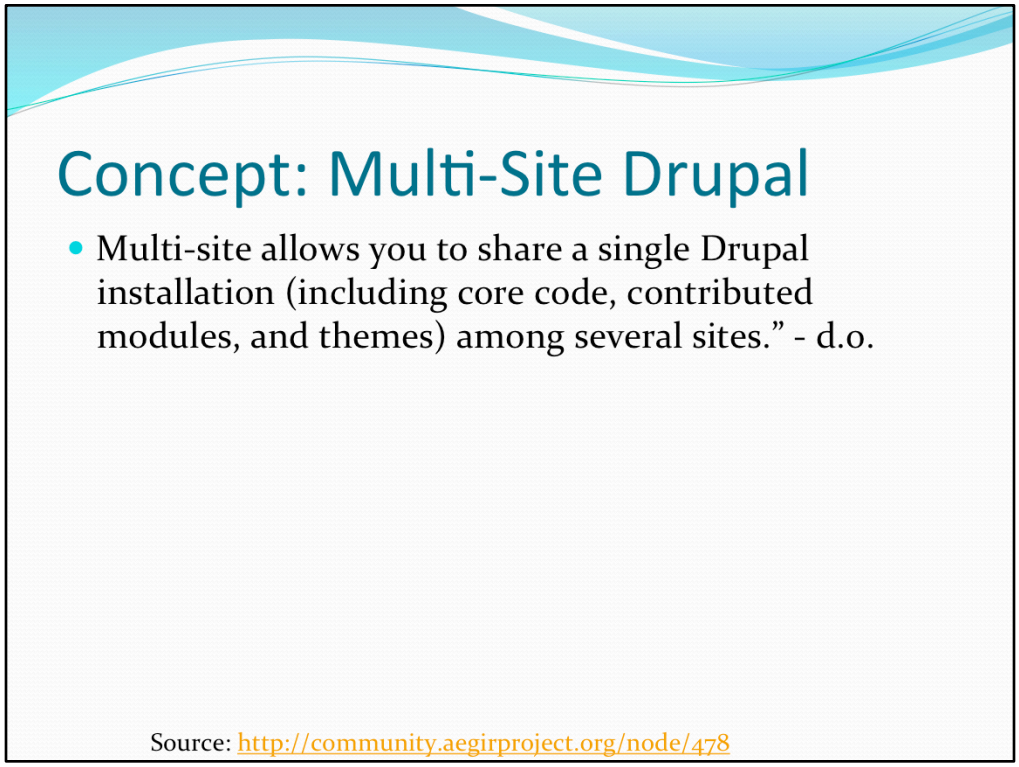
Aegir fully supports debian based systems. Using the proper repositories you can use apt-get install aegir to install it. At the UO we run Red Hat, so it was a bit more involved than one command. However, all notes have been posted on the projects website under manual installation.



Platform: A copy of drupal core that your sites exist on. e.g. d5, d6, 7, OpenAtrium, Pressflow…

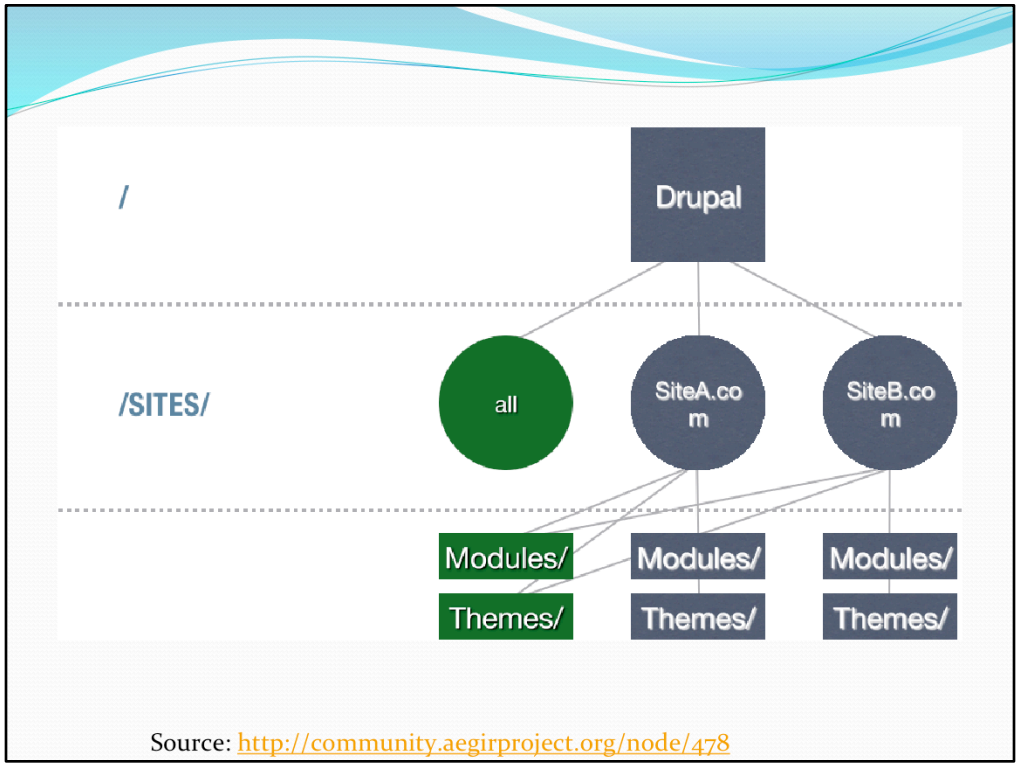
In Aegir, platforms allow site administrators to quickly view installed modules, migrate sites to new platforms(versions of drupal), rollback, etc… They can be either ssl enabled or not.

Now, lets talk a bit about multi-site drupal. Who here has a multi-site instance currently running?



Each site gets a directory in the sites/ directory
Each site has its own database, therefore its own content/users/settings/etc.
Only upgrade the code once

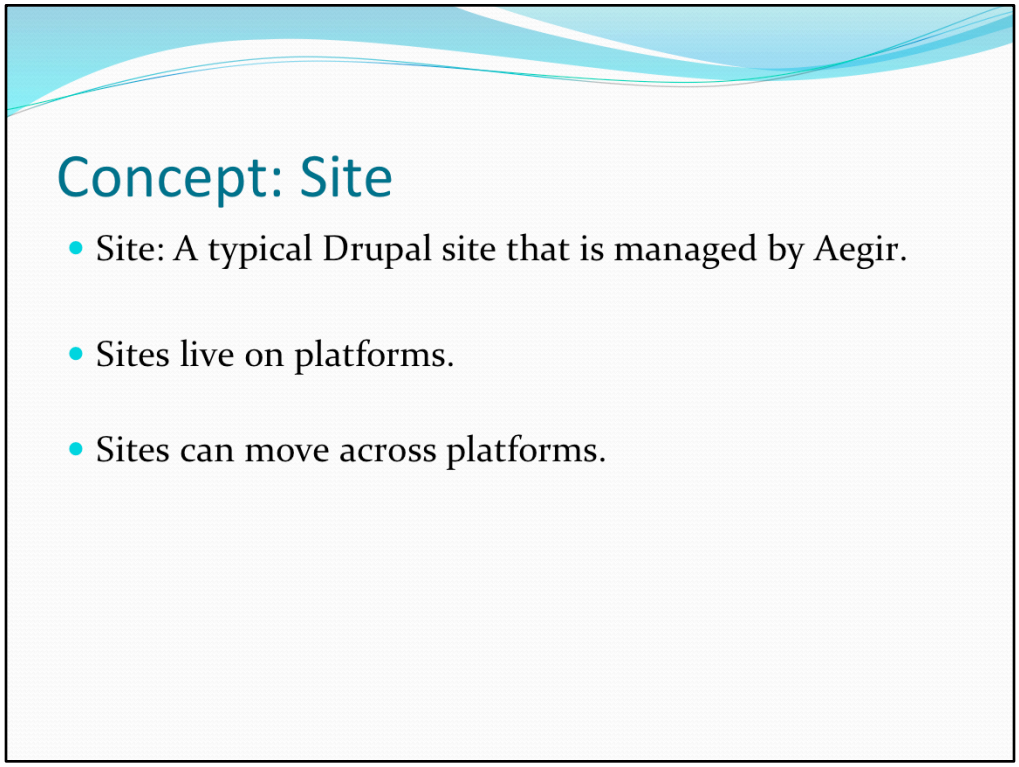
The platform contains all shared code, and each site overrides where necessary. Next is a clever diagram created by an Aegir community member that shows this well.



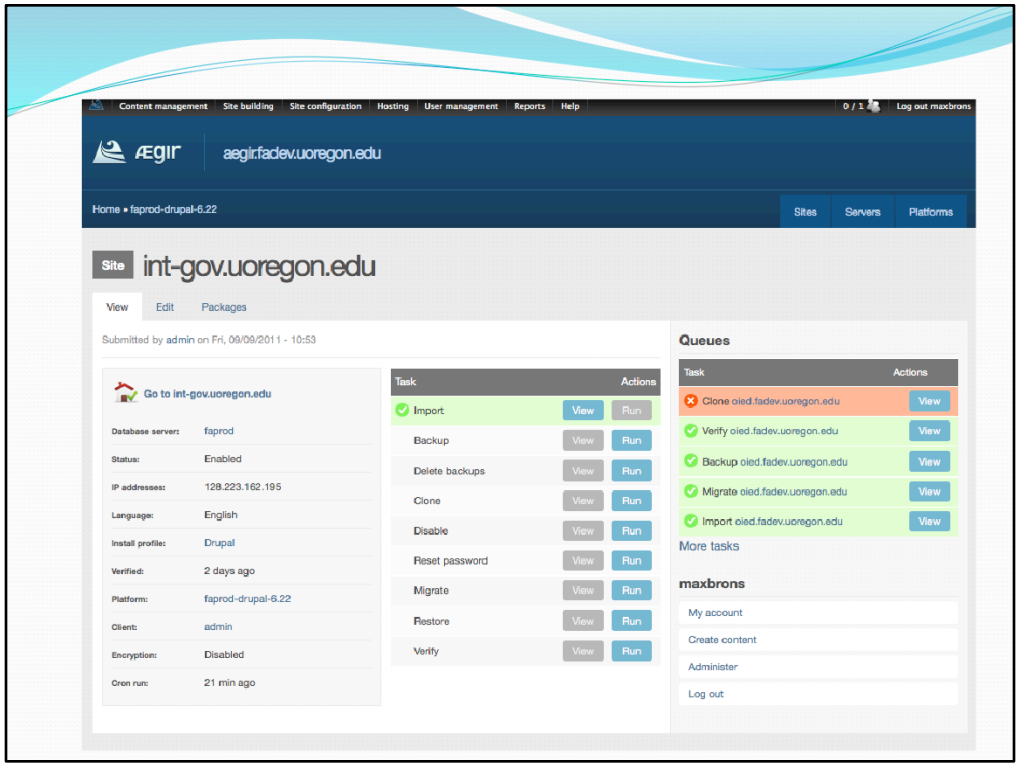
Source: http://community.aegirproject.org/node/478

Examples of shared modules and themes would be Views, CCK, and Zen.

Site A may have a custom module and Site B might have a custom theme not supported by the hosting company. This provides a relatively safe sandbox per site.

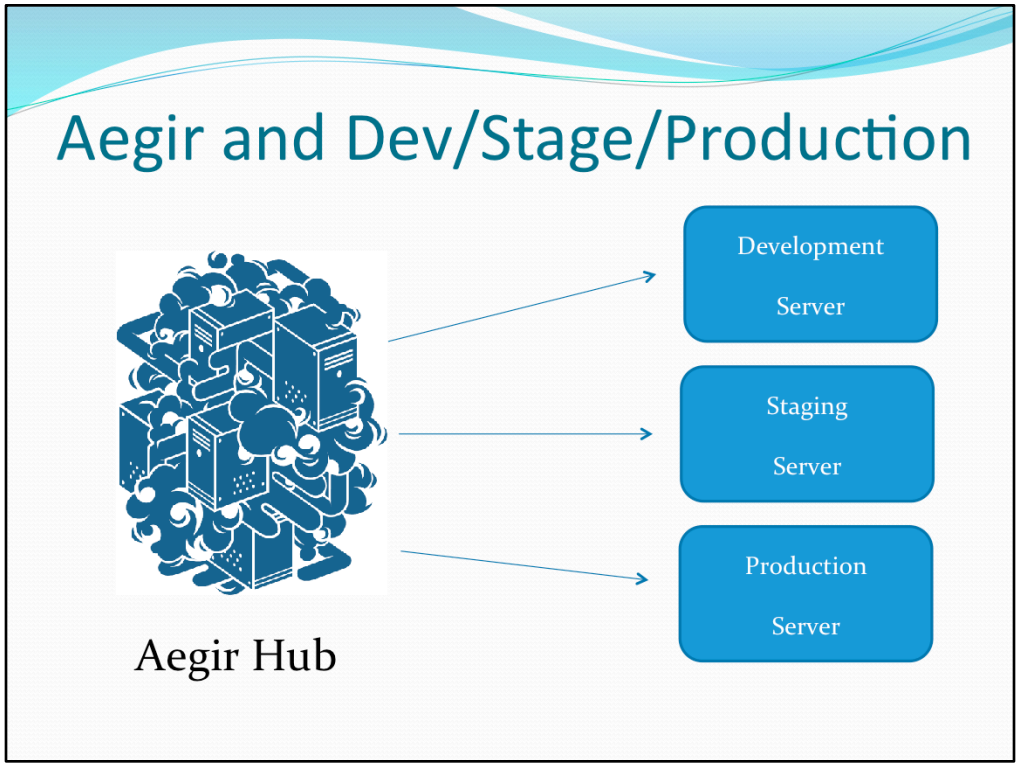


When using Aegir, all sites exist on platforms. The advantage of this is that you can migrate sites between platforms which essentially upgrades components as necessary.


Content management
Site building
Site configuration
Hosting
User management
Reports
Help
0 / 1
Log out maxbrons
Ægir
aegir.fadev.uoregon.edu
Home » faprod-drupal-6.22
Sites
Servers
Platforms
Site
int-gov.uoregon.edu
View
Edit
Packages
Submitted by admin on Fri, 09/09/2011 - 10:53
Go to int-gov.uoregon.edu
Database server: faprod
Status: Enabled
IP addresses: 128.223.162.195
Language: English
Install profile: Drupal
Verified: 2 days ago
Platform: faprod-drupal-6.22
Client: admin
Encryption: Disabled
Cron run: 21 min ago
Task
Actions
Import
Backup
Delete backups
Clone
Disable
Reset password
Migrate
Restore
Verify
View
Run
Queues
Clone oied.fadev.uoregon.edu
Verify oied.fadev.uoregon.edu
Backup oied.fadev.uoregon.edu
Migrate oied.fadev.uoregon.edu
Import oied.fadev.uoregon.edu
More tasks
maxbrons
My account
Create content
Administer
Log out

Concept: Hub & Spoke

- There is one central Aegir host (Hub)
- Host pushes out to other servers (spokes)

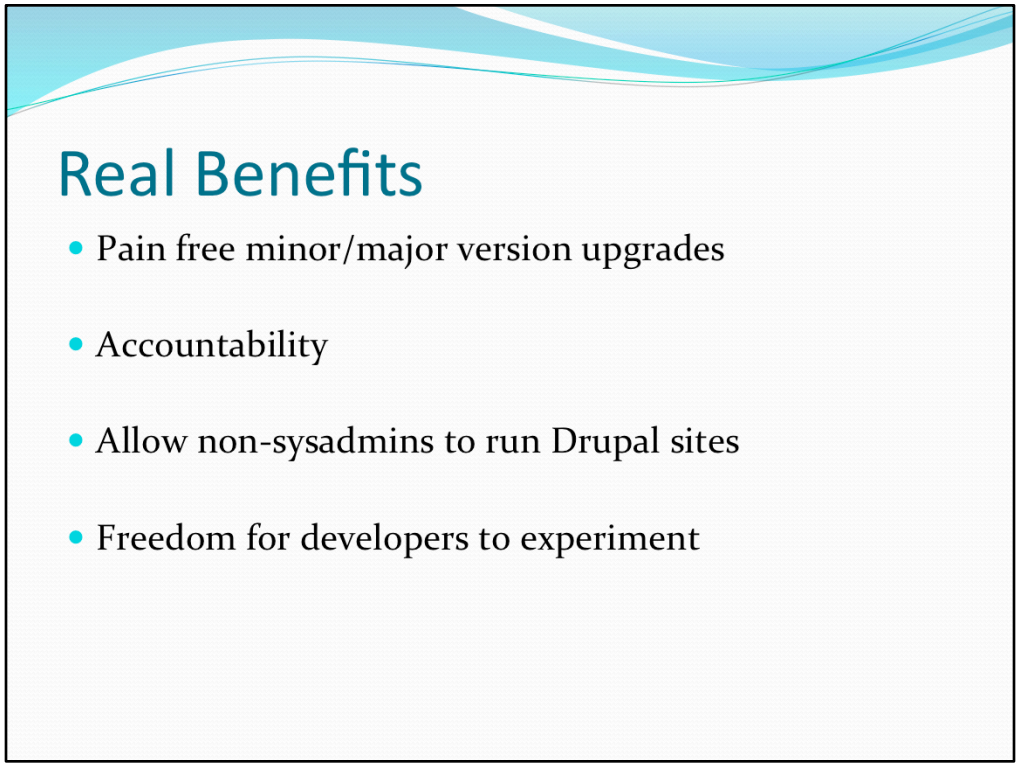


Blue computer cluster courtesy of http://www.aegirproject.org/

First let me say that Aegir is not a deployment tool out of the box. It is intended as a hosting management tool. However, for certain types of sites it works wonderfully for staging/production work. With a bit of effort, shell scripts can be used to do a true stage/prod environment.

How Aegir does this is the clone feature. It takes a site on an existing platform and clones it onto another platform. It does a great job of moving content, updating non-hardcoded links, and linking it to the proper database. This of course really only works for sites that are not being continuously edited or have a predictable rate of traffic. For large sites, the quest continues.

Here is how it works:



Upgrades:
Minor upgrades go easily. The benefit of Aegir is that you can see which modules besides core will be updated. You can also easily roll back to your prior platform if the upgrade does not work. Major upgrades are facilitated by Aegir but still manual work has to be done. An example is a drupal 5 to drupal 6 upgrade. Once in aegir you simply migrate the site to the new platform and then upgrade the modules as normal. Themes can be a real bear still.

Accountability:
Aegir allows a system administrator to see all of the drupal sites in one location. You can see what sites are on platforms that still have security alerts and move those to secure platforms. This helps facilitate accountability on the developer side.

Non-sysadmins:
A user can login and create a new drupal site on platforms that they have access too. If they have a good set of modules that are installed on the platform they should be able to develop and work on the site fairly insulated from the system administration duties. Within Aegir they can backup sites, disable sites, restore, reset passwords.

Freedom:
Developers can spin up sites with ease, they can move them to different platforms, they have the ability to focus on creating the site rather than installing it and all necessary modules. Combined with features or makefiles this is extremely powerful.

Git

Aegir and git work together to pull latest modules, core, etc… however, running a git repository of the code on the main hub seems to have iffy results. You cannot run it on a spoke as it will cause errors when the site is verified. The git repo must be owned by the Aegir user and have 775 permissions.

So far, the most success has been had using git locally, and then pushing and pulling to/from a gitosis repository. This way, if the sites git repo is wiped out, it still exists on gitosis.

.htaccess

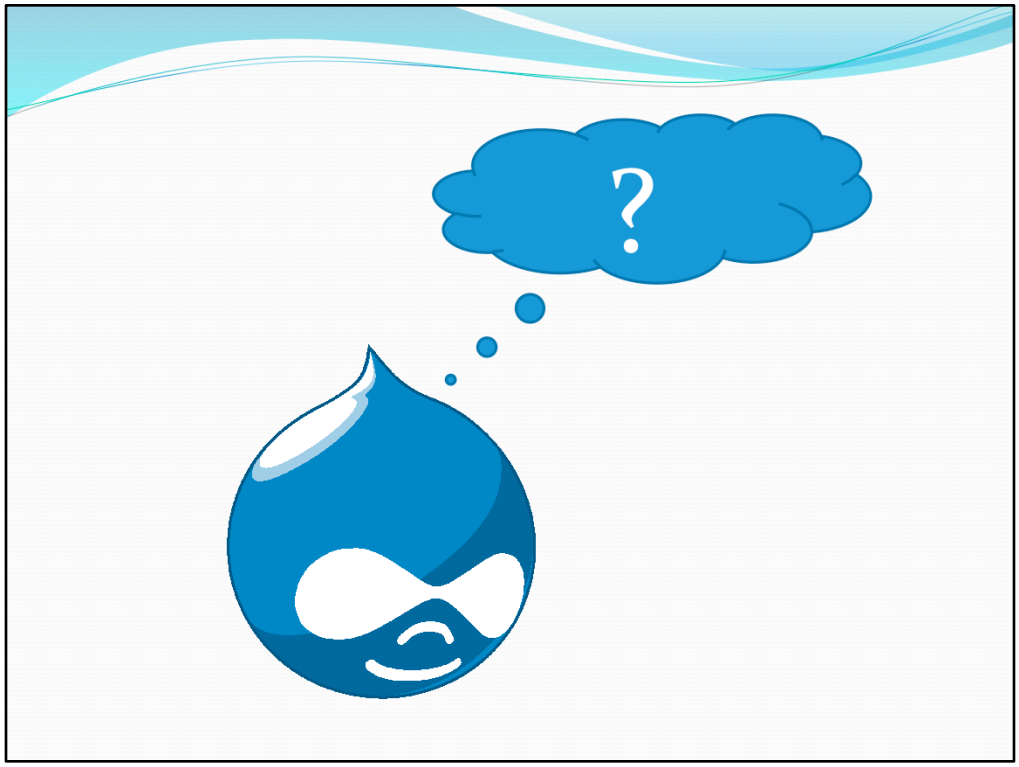
You cannot by default run modules that rely on .htaccess. There are workarounds for this we have discovered. You need to modify the .drush/provision script to allow for individual site .htaccess files. Also, local.settings.php can be used to expose external databases.

Permissions

Multiple developers on the aegir master box need to be able to do things inside of the aegir owned directories. Making them members of the aegir group allows some functionality to come through. Chown, mv, git, and drush are necessary sudo commands.

Demo

- Spin up a new site
- Move it to the staging server



What a remote server requires

- An aegir system account with shell access
  - sudo useradd -b /var -m -G apache aegir
- RSA keys from your Aegir hub.
  - ssh-copy-id_ aegir@serverName
- Aegir user allowed sudo privilege to restart apache
  - aegir ALL=(ALL) NOPASSWD: /usr/sbin/apachectl

Let Aegir know about the server

- Add a server using the Aegir interface.
- Clone an existing platform or create a new one.