
OKF Operations Manual

Release 1.0

Open Knowledge Foundation

March 29, 2016

1	Getting help	3
1.1	If you work for Open Knowledge	3
1.2	If you do NOT work for Open Knowledge	3
1.3	How to write a good support request	3
2	Guides	5
2.1	Requesting a Domain	5
2.2	Renewing Domains	5
2.3	Email alias management	6
2.4	Google Analytics	7
2.5	Managing mailing lists	8
2.6	Managing a site in the Blogfarm	10
2.7	Request Tracker	11
3	Sysadmin pages	15
3.1	Is this an emergency?	15
3.2	Documentation and notes	16
3.3	Who are we?	36
4	What is this?	37
4.1	Getting started with the Ops Manual, reStructuredText, and Sphinx	37
5	Indices and tables	39
5.1	README	39

Welcome. This is the OKF Operations Manual, where you will find documentation and advice on how the Open Knowledge Foundation (OKF) operates. While primarily intended to be used by our staff, the content here may also be useful for those affiliated with our Working and Local Groups, as well as our broader community.

If you need help from the sysadmin team, please read the [Getting Help](#) documentation.

Getting help

It's important you know how to request support from the Open Knowledge sysadmin team. If your support request is made through the right channels with the right information included, it will get dealt with much more quickly than if it is not.

1.1 If you work for Open Knowledge

If you work for the Open Knowledge, you should email sysadmin@okfn.org.

Please see below for a request template.

1.2 If you do NOT work for Open Knowledge

If you are affiliated with the Open Knowledge Foundation community (our Local Groups, Working Groups, Chapters, or similar) you should, in the first instance, email sysadmin@community.okfn.org.

Please see below for a request template.

1.3 How to write a good support request

No matter who you contact in the first instance, you are much more likely to get a timely and helpful response if you ensure that you provide the **right amount of information** when submitting your request.

If you don't give us enough information, it make take a long chain of back-and-forth emails to understand what needs to be done. Given that system administrators tend to attend to support emails in short bursts between actually processing requests, this could dramatically extend the amount of time required to process your request.

In particular, **it's important that you make the following clear** in your request:

1. Who you are and what project or group (if any) you represent,
2. What you would like the sysadmin team to do,
3. Any deadline for processing your request.

If you are requesting something which has non-trivial costs associated with it, please ensure you include the name of someone authorized to sign off on such requests, such as a Unit Head. Please not that we don't normally have any spare server/website capacity as we try to run as lean as possible

If you are reporting an outage or a technical problem, please also make clear:

1. What you did or tried to do, in enough detail to be able to reproduce the problem.
2. What you expected to happen.
3. What happened instead. Include any relevant logs or screenshots.

You may find the following template useful when submitting your request:

```
Dear sysadmin,  
  
Please could you <insert request here...>  
  
Group: (e.g. LTP, Knowledge, OKF Belgium, etc.)  
Project: (e.g. Open Data Index, CKAN, OpenSpending)  
Ideally done by:  
Must be done by:  
Cost code:  
Signed off by: (if you name someone, you must CC them)  
  
Regards,  
Joe Bloggs
```

You may of course omit any fields you do not see as relevant for your request.

This part of the operations manual contains documentation on tools and processes used by the Open Knowledge Foundation. Available topics are listed below:

2.1 Requesting a Domain

To request a new domain from the sysadmin team, please send an email into sysadmin@okfn.org with the following details:

- Name(s) of domain(s) required
- How long you want to register the domain for (default, 1 year)
- Does the domain need to be set to auto-renew
- What role is the owner of the domain (so we know who to ask about renewal)
- If this is for a specific project

When we say role, an example of a role would be say 'Network Team Lead' or similar. This is so we don't end up with a number of domains 'owned' by Alice which are then orphaned if Alice were to leave the organisation.

Auto-renew means that the domain will be automatically renewed when it comes up for renewal for a year. If you want to keep the domain, say yes. Otherwise we'll assume you don't want it to auto-renew.

If we don't know who to ask about a domain and it's not set to auto-renew the domain will normally be allowed to expire, so if you don't want that to happen, please tell us. A reasonable number of the domains that are registered are never actually used, so if a domain hasn't been used in a year and we've not been asked to auto-renew it, we won't.

Unless you want the DNS to be setup immediately, we will put a OKFN holding page on the domain. This is the holding page - <http://okfnlabs.org/holding-page/>

2.2 Renewing Domains

As an organisation we have a reasonable number of domains, while an individual domain might be cheap to renew, we have enough that it costs us a fairly large sum of money each year. Some of these domains are bought for projects and then not used. If a domain isn't in use and hasn't been used for a year, unless you've requested it to be auto-renewed, we will allow the domain to be dropped. So if you want to keep a domain, please make sure you follow the above process when requesting them.

2.3 Email alias management

2.3.1 Introduction

This document contains a description of how to manage our large set of email domains and aliases. **Members of the coordination team** (management + unit heads), as well as the sysadmin team, can edit aliases through the interface at FastMail, our front-line email provider.

2.3.2 How to add an email alias

1. Visit the [FastMail Virtual Domains configuration page](#) (log in from LastPass)
2. Fill out the form as shown below. For this example, we are adding an alias so that `joe.bloggs@gmail.com` receives mail sent to `joe.bloggs@okfn.org`. Please ensure that the penultimate checkbox (“SRS”) is checked as shown below.

3. When you're done click the “Add” button. You should see a confirmation like the following at the top of the page:

4. If the domain you wish to receive email at is not in the dropdown list, please submit your request to sysadmin@okfn.org.

2.3.3 How to remove an email alias

1. Visit the [FastMail Virtual Domains configuration page](#) (log in from LastPass)
2. Use the search box to find the alias. NB: enter only the username part of the email address. That is, if you wish to delete `info@okfn.org`, just search for “info”.
3. Click the “Del” button on the right hand side of the row containing the alias you wish to delete. You will see a confirmation similar to that shown above for alias creation.

2.3.4 How to search for an email address across all aliases

1. The Fastmail web interface search, does not let you search for email addresses that may be part of an alias,

- An optional method to look for email addresses across all aliases is, to set the 'Aliases to display' to its maximum value.

The screenshot shows a web form for configuring email aliases. At the top, there is a search bar with '@ okfn.org' and an 'Add' button. Below this, the 'Aliases to display' section has radio buttons for 20, 50, 100, and 500. The current selection is 20. To the right, it indicates '1-404 of 404' aliases. At the bottom, there is a 'Save Changes' button with a checkmark icon.

- The 'Aliases to display' option is located at the bottom of the 'Virtual Aliases' list.
- Once all the Aliases are displayed use CTRL+F, the browser search to find an email address or alias.

2.3.5 How to remove or add a user to an email alias

- Visit the [FastMail Virtual Domains configuration page](#) (log in from LastPass)
- Use the search box to find the alias. NB: enter only the username part of the email address. That is, if you wish to modify info@okfn.org, just search for "info".
- Add a user's email or remove a user's email on the second column, and click the "Save Changes" button at the bottom of the table. You will see a confirmation similar to that shown above for alias creation.

2.4 Google Analytics

2.4.1 Adding a New Site to Google Analytics

When there is a request to create a new Google Analytics tracking ID, it can be done with the following steps.

- Login to Google Analytics with a google account that has administrator access.
- From the home screen of Google Analytics, select the admin button near the top right of the screen. (if when you log in you do not land on the home screen, you should click the home button on the top left. These navigation instructions get quite confusing if starting from a different page).
- You should now be on the Account Administration screen with a table listing of the accounts your login has access to. Within this table click on "OKFN 2".
- You should now be on the Account Administration screen with a table listing of the accounts your login has access to. Within this table click on "Create a new property". This should be at the bottom of the list.
- Enter the website name (e.g. Open Development Toolkit) and website URL (e.g. <http://opendevtoolkit.net>). Change the Reporting Time Zone to United Kingdom and select "GMT (no daylight saving)".
- Click on "Get Tracking ID". This will get you to a page where the tracking ID will be prominently displayed.
- Go back to the Admin page and select the new Property. The view on the third column will be called "All Website Data". Click on "View Settings" and rename that to the property name.
- Go back to the Admin page and select the new Property. Click on User Management. At the bottom of the page, under "Add permissions for", grant the users of the project all permissions to this property.

2.4.2 Google Analytics (global tracking filter)

Each site is allowed a unique Google Analytics tracking code as per the above Google Analytics (single site). On top of this, a global OKF tracking code (UA-33874954) is added to every page to log data aggregate data across the entire blogfarm. No additional configuration is required to include traffic for a new site in this aggregate account, however it is possible to create a filter on this aggregate account to allow drill down reports of traffic for each site that is tracked. Administrators of the global tracking code may add this filter using the following steps.

1. Login to Google Analytics with a google account that has administrator access to the global profile
2. From the home screen of Google Analytics, select the admin button near the top right of the screen. (if when you log in you do not land on the home screen, you should click the home button on the top left. These navigation instructions get quite confusing if starting from a different page).
3. You should now be on the Account Administration screen with a table listing of the accounts your login has access to. Within this table click on “OKFN 2”.
4. Then select “OKFN Aggregate” from the Properties table on the next screen.
5. Here you should see a list of “Profiles” within this property. Do not edit the existing profiles, simply select “New Profile”
6. Leave “Web Site” selected under the “what data should this profile collect” question.
7. For the “Reporting Profile Name”, enter the name of the site
8. Select a relevant reporting timezone
9. Click “Create Profile”
10. On the resulting new profile screen, select the “Filters” tab.
11. Click “New Filter”
12. Select “Apply Existing Filter”
13. In the “Available Filters” list, scroll down and select the first “Add Hostname” item and click “Add” to add it to the “Selected Filters” list
14. Click “Save”
15. Return to the Filters List (by clicking save if you have landed on the “Edit Filter” page).
16. Click “New Filter” again
17. This time leave the “Create New Filter” option selected
18. In the filter name, insert “Include Domain (example.com)” where the item in parenthesis is the url of the site being filtered. This is just for identification purposes.
19. Select the “Custom Filter” option
20. Then select the “Include” option
21. Then from the “Filter Field” dropdown select “Hostname”
22. Then in the filter pattern enter the domain, escaping periods with backslashes (eg publicdomainreview\.org or blog\.okfn\.org)
23. Leave case sensitive as “No”
24. Click “Save”
25. It may take some hours for traffic stats to appear for this profile in the reporting section

2.5 Managing mailing lists

2.5.1 Introduction

This document contains a description of how we manage a large set of community mailing lists, and how you can perform common operations required of a list owner or moderator.

2.5.2 The basics

Our list server is known as `lists.okfn.org`, and can be visited on the web at <https://lists.okfn.org>.

A mailing list called `widgets-discuss` will have an email address of the form `widgets-discuss@lists.okfn.org`. List owners can always be contacted by emailing `widgets-discuss-owner@lists.okfn.org`.

You can link to the `widgets-discuss` email address directly at <https://lists.okfn.org/mailman/listinfo/widgets-discuss> (please note that this particular link will not work, as we don't – yet – have a widget discussion list).

2.5.3 List administrators and moderators

List administrators and moderators are responsible for the smooth operation of the list. Their tasks may include:

- Helping users who have difficulty subscribing to or unsubscribing from their lists.
- Banning abusive users.
- Filtering spam that's sent to the list.
- Ensuring that the list description and settings are kept up-to-date.

In particular, list administrators (also known as list owners) can do all of these things and more. List moderators have a single role: reviewing the moderation queue of the mailing list and letting through legitimate emails while rejecting spam.

2.5.4 Obtaining a mailing list

If you need a mailing list for your project, you need to make a few decisions about your list, and then you can contact the Open Knowledge Foundation sysadmin administrators for help. You should decide:

1. What you want your list to be called. A list to discuss geographic data might be called `geodata-discuss`, while a local group list for Spain might be called `okfn-es`. See the list names already taken at <https://lists.okfn.org> for ideas.
2. Do you want your list to be public or private? Should anyone be able to subscribe, or do you want to approve new sign-ups on a case-by-case basis?
3. Do you want your list to be publicly advertised in the list of lists at <https://lists.okfn.org>?
4. Who will be your list owners? Ideally you should have at least two.
5. Do you want to designate additional list moderators?

When you have answers to all these questions, you should contact the system administrators with your request. Details on how to do this can be found here on the Ops Manual: [Getting help](#).

2.5.5 Administering or moderating a mailing list

Say your list is called `widgets-discuss`. You can visit the list information page at <https://lists.okfn.org/mailman/listinfo/widgets-discuss>. From there, you can select either the “administrative interface” or the “moderation interface” links at the bottom of the page.

If you are a list administrator or a list moderator, you should have been emailed a password allowing you to administer or moderate the list. Please check your email archive for this email (it will come from `mailman@lists.okfn.org` and have a subject like “Your new `widgets-discuss` list password” or “Your new `widgets-discuss` moderator password”).

If you still can't find the password and think you can have it, please contact the system administrators: [Getting help](#).

2.5.6 Moderating mailing list spam mail

If the list you are moderating receives spam, you can forward these spam mails from the moderation interface to our bayes based spam learning address - `learn-spam@lists.okfn.org`

This will help our mailman spam filter to 'learn' and classify spam mails correctly.

2.5.7 Mailing list actions that require sysadmin help

You may request help from sysadmins for these mailing list queries

- Create a new list @lists.okfn.org
- Create a list on a different @domain
- Delete a list
- Keep list of subscribers
- Keep list archive
- Rename a list (copy subscribers over to a new list and archive the old one)
- Mailing list archives in a tarball/zip
- Reset the Admin password
- Export list subscribers as csv

2.6 Managing a site in the Blogfarm

This article contains guidelines for site owners how to activate popular features on OKFN's Wordpress blogfarm at WPEngine.

2.6.1 Activate Akismet / Basic spam protection

Akismet is a comment and trackback spam prevention plugin that ships with WordPress. Site administrators can implement basic spam protection on their site by activating and configuring this plugin using the following steps.

1. An API/Licence key is required for Akismet to function, you register for one at their website <https://akismet.com/signup/>
2. Log into the dashboard for your site (<http://site.okfn.org/wp-admin>)
3. Navigate to Plugins and activate Akismet
4. Navigate to Plugins > Akismet
5. Enter your API/Licence key then save

2.6.2 Disqus Comments

As an alternative to the built-in WordPress comments feature, the Disqus comments is a third party commenting system that may have more suitable features and spam prevention methods. Site admins can enable Disqus comments on their site using the following steps.

1. Signup at the Disqus website <https://disqus.com/admin/signup/>

2. Make note of the Site Shortname entry, and the login credentials of the Primary Moderator account you created
3. Log into your site dashboard (<http://site.okfn.org/wp-admin>)
4. Navigate to Plugins and activate the Disqus Comment System plugin
5. Navigate to Comments > Disqus
6. Login with the primary moderator account you created in the Disqus signup
7. Select the correct website profile and save
8. Disqus comments should now be active, verify this on your site. The Comments > Disqus menu option should now provide a portal to moderate/manage comments and settings

2.6.3 Google Analytics (single site)

Google Analytics tracks visitors of your site. Site administrators can enable this on their site using the following steps/.

1. A Google Analytics tracking code is required, Google has existing help resources explaining how to do that - <https://support.google.com/analytics/answer/1008015?hl=en>
2. Log into your admin dashboard
3. Navigate to Settings > Google Analytics
4. Enter your tracking ID, then save.

NOTE: only available if you have rights to demonstrate you are the owner of the site. (ftp access, template edition rights, etc)

2.6.4 Domain Redirections

Redirections should *only* be managed on the WPEngine domain panel <https://my.wpengine.com/installs/okf/domains>

Click “Add Domain Redirect to this Domain” to add a new redirection for that domain.

2.7 Request Tracker

Request Tracker or RT as it is more commonly known is request tracking software. Open Knowledge uses it to track requests for work from internal and external users of our IT systems.

RT can be found at <http://rt.okfn.org>. Even if you don’t have an account with RT you can email requests into RT and these will be logged as tickets which the relevant team can act upon.

2.7.1 Guides

Dashboards

Dashboards are a useful way of setting up a page either for yourself or your team which shows a specific subset of tickets. If you want a specific dashboard send a ticket into the sysadmin team with the following information:

- Name of the dashboard
- Tickets you want in the dashboard
- Which group the dashboard is for

There is a lot of extra useful information about dashboards on the RT website:

<http://www.bestpractical.com/docs/rt/4.2/dashboards.html>

Quick Resolve Bookmarklet

If you find yourself with a lot of tickets which all need to be resolved in a hurry, using a bookmarklet can save a lot of time. Simply create a bookmark with the following javascript code in it.

```
javascript:self.location=self.location+'&Status=resolved'
```

How to log a ticket via email

Currently the sysadmin and finance teams use RT. To log a call with the sysadmin team use the following email address:

sysadmin@okfn.org

To log a ticket with the finance team, please email:

payments@okfn.org

You will receive an automatic email from RT. This will contain your ticket ID. If you reply to this email it will update your ticket. RT knows how to file your reply by the subject. So it is important that you do not change the subject. The key part of the subject is this:

```
[sysadmin #112233]
```

If you do not have the part between [and] then RT will create a new ticket. This can be confusing if someone has cced sysadmin@okfn.org in (for information) and then when you hit reply all, you don't have the right subject and RT creates another ticket. If you do not have a request, please do not email sysadmin@okfn.org, as it goes directly into RT. If you know a ticket has been created, sending an email with the right ID into RT will file the response in the right place.

An example of a subject for payments would be:

```
[[payments@okf] #12246]
```

Making changes to a ticket through the web interface

If you have an account on rt.okfn.org then there are a lot of extra things you can do to a ticket. It's possible to merge tickets, link them together, set reminders. As well as changing the owner, queue and priority.

It's not possible to delete a reply or a comment. So please be careful when replying to a ticket, as once you've replied, that reply is there for ever. Never say anything on RT that you wouldn't want the requestor to read.

History

RT records every action on a ticket and you can view the full trail of actions in the history section. This is useful to track how much time people have been working on a ticket or what actions have been taken on a ticket. You probably won't use it often, but it's very useful when you do.

Basics

You'll use the basics section quite a lot and it's quite straightforward. This is where you change the owner, queue, subject, status, priority and a few other things. You can use the priority field however you want to, but it's worth discussing how you want to use it within your team and sticking to it. Obviously it's a handy way of prioritising work. If the request is urgent, then give it a 5. If it's something you'd like to do, but isn't that crucial, give it a 3. As long as you and your team are consistent, then this can be a really useful way of managing your workload.

Links

The links section (top right hand corner of the UI) is where you can merge tickets. You can also link tickets up in two ways.

If you want to merge a ticket, simply add the ID of the ticket and save. This cannot be undone, so make sure you're merging the right two tickets.

There are quite a few ways to link tickets together, but the most useful is probably 'Refers to' which simply says the ticket refers to another ticket. If you're not sure about how this works, feel free to log a couple of test tickets and link them up.

Jumbo

Jumbo displays everything so if you have multiple changes to various parts of a ticket, this is useful. But it can contain a lot of information.

Reminders

You can set a reminder to remind you to check on a ticket on a certain day. This is a really useful feature and well worth using. If someone has a request they want done on a specific day, just set a reminder for that day and RT will remind you on the day.

Searching for tickets

If you know the number of the ticket, this is really easy, just put that into the search box which is on all pages. If you put 'fulltext:some text' in this box, RT will do a full text search of all tickets.

If you want something a bit more complex, there is the Advanced Search page, which will allow you to search in great detail for tickets. Here is a short overview of what is possible. It's worth spending a little bit of time learning how to search for tickets as it's a very flexible and powerful interface.

To get to the advanced search interface click, Search > Tickets > New Search.

Once you're on the page, here's a quick overview of it. On the left, you've got the search criteria. On the right you have the details of the search that you've built. Below the current search box, there is a 'Saved searches' box which can be used to save specific searches and load saved searches, which can save you a lot of time. At the bottom on the left you can set the sort options for the search. On the bottom left you can select the display columns of the search.

The section where you build your search is the Add Criteria area. As you can see, you can search by a large list of criteria. If you know that the tickets you want are from bob@okfn.org and they are in the payments queue, you simply put Queue 'is' Payments in that dropdown. Then in the Requestor Email 'is' bob@okfn.org, then click 'Add these terms and search'.

Saving a search

Once you've got a query you're happy with, you can save it. Simply go to the Saved searches section, set the privacy (if it's just for you leave on 'My saved searches'), but if you want something others can use, select the right area. Fill in the description and click save.

To use a saved search, just go to 'Load saved search' and select the search you want and click load. Then click the search button in bottom right.

Replying and Commenting

There are two ways to add your input to a call. Replying and commenting. When you reply to a ticket, that reply is emailed to the requestor and all others CCed or AdminCCed into a call. If you add a comment onto a ticket it is not emailed out to everyone. It is possible to see who's been sent a reply by looking at the 'Show Email' link on the reply/comment on the ticket in the web interface.

If you email a ticket, that is always added as a reply. So your email will go to everyone on the ticket.

It is important that you understand the difference and also that it isn't possible to delete a reply or comment. So before you reply, make sure you're happy with everyone seeing the reply and that you can't remove it afterwards.

2.7.2 Current RT email addresses

Currently the sysadmin and finance teams use RT. To log a call with the sysadmin team use the following email address:

sysadmin@okfn.org

To log a ticket with the finance team, please email:

payments@okfn.org

2.7.3 Official Documentation

The official documentation from Best Practical the makers of RT can be found here:

<http://www.bestpractical.com/docs/rt/4.2/>

Sysadmin pages

This part of the operations manual contains documentation by and for the OKF's system administrators. If you have any questions about this content, please contact one of the sysadmin team: *Who are we?*

3.1 Is this an emergency?

In order to better serve our users, we prioritise our work, primarily on the basis of information that you provide us when [submitting your requests for help](#). But this leads to a problem, which is that requestors develop a tendency to slightly overstate the urgency of their request, in the hope that it will be processed more quickly. Everyone does this, even us!

The end-state of this situation isn't pretty. Soon everything is "urgent" or "an emergency," which has several problems:

- real emergencies get missed
- system administrators are constantly interrupt-driven, and are unable to effectively manage their workload
- any work which is not "urgent" takes a disproportionately long time to be processed

So, we need a way to define what an emergency is. This is difficult to do for the Open Knowledge Foundation given the varied nature of our work. As such, we're going to provide two lists which help explain what is, and isn't, an "emergency".

3.1.1 It might be an emergency if...

We distinguish between the following situations:

"code red" Situations which are currently preventing our users from using services which we are reasonably expected to provide them, or which are stopping our staff from conducting their business.

"code yellow" Any situation which, if not addressed, could lead to a "code red".

In general:

1. Any unscheduled outage or breakage of any public-facing system is an emergency.
2. Any unscheduled outage or breakage of any system that is critical for you to do your job is an emergency.

Here are a few emergencies:

- a public website hosted by us is unavailable, unresponsive, or broken
- a mailing list is unusable, or emails aren't reaching members
- you can't get to your work email

- hackers or spammers have taken control of one of our web properties
- a **project partner or funder** has unilaterally changed the circumstances surrounding a project such that our deadlines have changed **in a way that we have no control over**, and thus requests for changes to sysadmin-controlled systems need to be processed faster than usual ¹

3.1.2 It's probably not an emergency if...

We don't want to be seen as **BOFHs**, but the cardinal rule here is:

Poor planning on your part does not automatically constitute an emergency on mine.

That means that you should not second-guess the system administrators' workload. Please trust that we prioritise work such that everyone is happiest over the long run.

Here are a few plausible situations in which flagging a request as "urgent" is not usually acceptable:

- you or your project team failed to identify all the dependencies of launching your new project, and now it needs launching "urgently" with support from the sysadmin team ²
- you want something done, and used to be able to do it yourself, but now it's looked after by the sysadmin team
- it's Friday afternoon, and you just remembered something you promised someone by Monday that needs sysadmin intervention
- you are a senior staff member, so your request is more important and should be processed more quickly than others

Ultimately, we want to provide the best possible service to our users, just as you do. We will of course still be happy to bend the rules for you every once in a while. **We are all human, and we're happy to help you with an "avoidable emergency" every once in a while.** Just please try hard not to make a habit of it: remember the story of the Boy Who Cried Wolf.

Footnotes:

3.2 Documentation and notes

3.2.1 Infrastructure overview

Open Knowledge operates many different services for many different people. This document is intended to provide a broad overview of the infrastructure underlying these services and how it is managed.

The infrastructure operated by Open Knowledge (and specifically that managed by the Sysadmin team) can broadly be divided into two categories:

1. *Self-hosted infrastructure*
2. *Externally-hosted infrastructure*

This document describes these two areas below:

¹ Please note that the critical feature here is that there is no way you could reasonably have been expected to know the situation was going to change. A similar situation in which the change of deadline comes from within your project team is not comparable.

² It's worth reminding you that we're always happy to consult at an early stage on what things might need to happen with our involvement to make your project a success! Email sysadmin-team@okfn.org with your questions.

Self-hosted infrastructure

Machine hosting

We maintain a reasonable number of (primarily Ubuntu Linux) servers across a number of hosting providers. Without exception, these machines should be managed by our Ansible configuration management system.

Please refer to the [Ansible inventory](#) for the canonical list of managed hosts and their whereabouts.

Mailing lists

We maintain a [Mailman](#) instance which manages a wide variety of public and private mailing lists on the `lists.okfn.org` mail domain.

Etherpad

Etherpad is an online document editor providing collaborative real-time authoring. We host an instance at <http://pad.okfn.org>.

Wiki

There is a somewhat under-loved MediaWiki instance available at <http://wiki.okfn.org>

Externally-hosted infrastructure

Mail routing

We use [FastMail](#) as our front-line mail router for almost all mail domains associated with OKF (`okfn.org`, `ckan.org`, etc.). One notable exception is the mailing list server (`lists.okfn.org`) described above: *Mailing lists*

Staff email

Open Knowledge Central staff email, calendars, etc. (often collectively known as “groupware”) is currently provided by [Google Apps for Business](#).

We also now have [Google Vault](#). Which can be used to ‘save’ emails from email accounts which aren’t in use.

WordPress “farm”

A multi-domain [WordPress](#) “farm” is operated for us by [WPEngine](#). This hosts, among others, `blog.okfn.org`, as well as many other project and local group sites. Please note that this **does not** host the `okfn.org` website.

3.2.2 Repositories

We attempt, as far as is reasonably possible given the tools and time available, to abide by the principle of “Infrastructure as Code” – all parts of our infrastructure configuration should be easily reproducible in an automated fashion.

The following repositories are used to manage the OKF infrastructure:

infra

<https://github.com/okfn/infra> is the main repository we use to manage our infrastructure. It includes configuration and data for the [Ansible](#) configuration management system.

credentials

<https://github.com/okfn/credentials> is a private repository containing secrets, keys, and other things the rest of the world shouldn't know. The contents of this repository are encrypted using [git-crypt](#).

sysadmin-oz

<https://github.com/okfn/sysadmin-oz> contains a simple tool to optimize DNS zones according to our internal DNS standards.

3.2.3 List of infrastructure services

Blogfarm

OKF has a blogfarm at specialized Wordpress hoster [WPEngine](#). It is a wordpress multisite installation that serves numerous properties from a single codebase. It can provide sites at **{name}.okblogfarm.org** as well as at any domain name (via Wordpress' domain alias feature).

Amongst the sites it currently hosts are the [OKFN blog](#), local OKFN chapters like [OKFN Belgium](#) or [OKFN Germany](#), [CKAN](#), the [Open Knowledge & Data Festival](#), [OpenGLAM](#), [Digitised Manuscripts to Europeana](#) and many others.

WPEngine Overview

Some basic instructions outlining the code deployment procedures for our blog farm hoster WPE ([WPEngine - http://wpengine.com/](http://wpengine.com/)).

WPE code deployment is done through their git post-recv hook. In essence it's as simple as committing code changes to the repository and doing a git push back to the WPE remote. More detailed instructions can be found in their documentation about this - <http://git.wpengine.com/>.

Access to the WPE git repos is controlled by public keys. You'll need to contact the OKF Sysadmin team to have your public key granted access before proceeding. Once this has been done, confirm access by:

```
$ ssh git@git.wpengine.com info
```

You should get a response back listing the repositories you have access to. At a minimum these should be:

```
production/okf
staging/okf
```

Consider these as two separate git remotes containing the code for the production web dir and staging web dir respectively.

Git Workflow

We maintain the code on our github repository.

```
$ git clone git@github.com:okfn/wordpress.git wordpress
$ cd wordpress
```

This repository uses submodules for the various plugins and themes. So you should initialize and update these:

```
$ git submodule update --init
```

The recommended workflow is to add remotes for production and staging:

```
$ git remote add staging git@git.wpengine.com:staging/okf.git
$ git remote add prod git@git.wpengine.com:production/okf.git
```

The wordpress repository has two branches, master and prod. Changes need to land in master, be pushed to staging, and then merged into the prod branch.

Deploying changes is as simple as pushing your commits back to the staging or production. So something like the following. It's advised that you do this via a terminal window so that you can see descriptive deployment progress output, along with any custom errors. When you push to staging, it's recommended that you push to origin at the same time so we keep our github repo up-to-date:

```
$ git commit
$ git push staging master
$ git push origin master
```

To push to production, switch to the prod branch, merge master into it, and push to prod:

```
$ git checkout prod
$ git merge master
$ git push prod prod
$ git push origin prod
```

Sometimes, we've noticed that pushes to WPEngine didn't work as expected. At that point, make a small change, push to staging or prod (wherever you find an inconsistency) and check again. If it worked, revert that commit and push again. This is the only time when *git push -f* is the right thing to do. Any other time you think you need to do a *git push -f*, please check with the sysadmin team first:

```
$ git commit
$ git push staging master
$ git reset HEAD~
$ git push -f staging master
```

Sub Modules Note that submodules are referenced to a master repository via commit ID's, rather than branches. As such, changes to submodules need to be committed to the master repo as well. That is, if you commit a change to a submodule, the latest commit for that repository has progressed, and that new status needs to be reflected in the master repo. If you make changes to the submodule outside of the main repo working directory, you'll need to git pull that submodule into the main working dir and commit it's new pointer to that main repo.

When you deploy to WPE, for any submodule that the master repository knows about, the commit for that submodule must be accessible to WPE. This means 2 things: 1) The repo must be publicly readable. WPE does not provide a public key for you to grant access to private repos. 2) The commit to retrieve must be available on that remote (pushed)

The latter is often forgotten. Use the following one liner to push all submodules from your master repo root dir:

```
$ git submodule foreach git push master origin
```

It's typically easier to work on all files (main repo and submodules) within the same working directory so that you can commit to both the sub and main repos quickly.

If other users make changes to the sub-modules, you'll need to retrieve these to your own local dir also. So before making any changes, it's good practice to do the following:

```
$ git pull origin master
$ git submodule foreach git pull origin master
```

Staging

The canonical URL for staging is <http://okf.staging.wpengine.com>. It is recommended you login to the staging site and look at the [sites](#) on the network admin panel. This should give you the right URL for other staging sites.

Contact

- For **non-technical issues** with the OKFN blog farm, e.g. with the content of one of its sites, contact either the owner of a specific blog site, or <blog@okfn.org>
- For **technical problems** with the OKFN blog farm, notify either the owner of a specific blog site, or <sysadmin@okfn.org>.

There is also a mailing list <sysadmin-coord@lists.okfn.org> for general technical discussions.

Tickets (general)

- To raise a ticket in RT tracker, simply send a mail to <sysadmin@okfn.org>
- If our blogfarm hoster WPEngine has to look into an issue, we raise a ticket in their support platform (see below)
- There is [Issue tracker for http://okfn.org/](#) which tracks non-sysadmin issues with okfn.org, powered by Django CMS.
- There is [Issue tracker for OKFN sites](#), which tracks non-sysadmin issues on other OKFN sites.
- Themes and plugins might have their own repo trackers, e.g. the “[Wordpress OKFN general-purpose theme \(v2\)](#). Based on [Bootstrap](#) and [Buddypress](#) “

Tickets at WPEngine

 How to file a ticket at WPEngine:

- If you haven't done so yet, register at [WPEngine's Zendesk platform](#). You have to use your @okfn.org mail address.
- Log into [WPEngine's Zendesk platform](#). You should see “OKFN.org” organisation.
- Click on “[SUBMIT A REQUEST](#)”.

Their core support times are 9:00-18:00 US Central Time (WPE sits in Austin, Texas. CST=UTC-6, CDT=UTC-5). That is usually 15:00-24:00 UK time. About out-of-hours support, [they write](#):

After hours Support and phone support focus primarily on emergency and high-priority issues. But we are getting closer and closer to being able to offer 24/7 Support. There are multiple team members here, answering phones and cleaning up tickets at virtually all hours of the day, every day. For now you can be confident that, when you have a problem, there will be someone here to help.

Notes:

- We are currently setup as a ZenDesk “shared organization” (as opposed to “non-shared”): everyone who registers with a @okfn.org address can access all our tickets.
- There is a “[Subscribe](#)” button at “[OKFN.org](#)” ==> “[Open requests](#)“

How to report failing blog pages If blog pages fail please report it to us! But in order to look into the failure, we need some details. Please collect the following data and send it to <sysadmin@okfn.org>:

- The **failing URL**. If the issue affects several URLs or sites please mention a couple of them (at least three).
- The exact **time and date** of the failure, with timezone.
- Your **IP address**. Click [here](#) to find out.
- What is the **false result** (as opposed to the expected result)? E.g. copy&paste the error message.
- (If you have a login to the site): Does it make a difference whether you are logged in or not?
- Make sure you are logged out. Does the error disappear when you circumvent WPEngine's cache? (Append a random querystring like "?q=5971" to the URL)
- (If yes, and you are a site admin) Does it help when you clear the WPEngine cache? (see section "Caching" below)

Don't bother to report failing blog pages if you don't have the time to provide those details - there is nothing we can do without them.

Caching

One can check whether a page was delivered from a cache by looking at the headers (e.g. using `curl -I` or the Firefox Add-on [Live HTTP Headers](#)). Example of a cached page:

```
$ curl -sI http://okblogfarm.org/ | grep '^X-Cache'
X-Cacheable: SHORT
X-Cache: HIT: 3
X-Cache-Group: normal
```

One can view any page bypassing the cache by appending a unique (e.g. random) query string. E.g.:

```
$ curl -sI http://okblogfarm.org/?nocache=00012 | grep '^X-Cache'
X-Cacheable: SHORT
X-Cache: MISS
X-Cache-Group: normal
```

WPEngine uses Varnish, which caches aggressively. If this cache needs to be busted, it needs to be ticketed with them.

How to: create a new blog...

... as **{name}.okfn.org** Requirements:

- You will need to be a Network Admin on okfn.org
- You will need control over the DNS records for *okfn.org*
- You will need access to the [WPEngine control panel](#) (see below).

Basic install:

1. Login and go to Network Admin - <http://okblogfarm.org/wp-admin/network/>
2. Select Add Site
 - For WG sites name after working group e.g. for economics wg would be economics.okblogfarm.org
 - Put your username/email for admin role
 - Test <http://{name}.okblogfarm.org/>, it should work now.

3. Add users to site as appropriate
4. Leave the “Network Admin” area. Instead, go to the admin area of your new blog at <http://{name}.okblogfarm.org/wp-admin/>
5. Activate and configure standard plugins:
 - Akismet
 - Google Analytics (see Google Analytics in Settings)
6. Go to the [domain admin page](#). Add the site ID of your new site and the domain name if it needs to be <http://{name}.okfn.org/>, tick the *Primary* checkbox and submit the form.
7. Log into the [WPEngine control panel](#) then, add the new site hostname under [Domains](#) (you might want to add redirects from www - optional)
8. Create a DNS CNAME record for [{{name}}.okfn.org](#) pointing to [blogfarm.okserver.org](#). Wait for the record to propagate and test.
9. (Optional) Configure theme. The default Open Knowledge Foundation theme is maintained at <https://github.com/okfn/wordpress-theme-okfn>.
10. (Optional) Activate & configure additional plugins. Do this on a site-by-site basis, do **not** use ‘Network Activate’

Remark:

- Commonly used files should be hosted on Amazon S3 bucket <http://assets.okfn.org>. The process for uploading is documented at <https://bitbucket.org/okfn/m.okfn.org/src/d7625d7066d0/m.okfn.org/README.txt>

... as **mydomain.org**

- You will need control over the DNS records for *mydomain.org* (see Sysadmin/DomainServices)
 - You will need access to the [WPEngine control panel](#) (see below).
 - You will need to be a Network Admin on okfn.org.
1. Setup a new site as {name}.okfn.org as described in the previous paragraph.
 2. Log into the [WPEngine control panel](#) then, add the new site hostname under [Domain](#) (you might want to add redirects from www - optional)
 3. Temporarily add the blog farm’s IP address “178.79.130.212 *mydomain.org*” to your `/etc/hosts` and test <http://mydomain.org/>.
 4. Create a DNS CNAME record (see Sysadmin/DomainServices) for *mydomain.org* (and *www.mydomain.org*) pointing to [blogfarm.okserver.org](#) or its IP address 178.79.130.212. If the domain is at DME, make it a “A-NAME” to [blogfarm.okserver.org](#). Wait for the record to propagate and test.

How to: add or modify a theme/plugin

Caveats There are certain things to be aware of when managing a wordpress installation at WPEngine:

- **Do not try to modify/update/upgrade the Wordpress core.** It is maintained by WPEngine
- **Make minimal use of session cookies.** The presence of a session cookie might circumvent caches.
- **Some PHP methods might be restricted or not available at all.** WPEngine might apply strict security policies and restrict what PHP can do. That could break plugins/themes.

HOWTO: “Dead man’s switch” monitoring

Monitoring things that are supposed to be working all the time (like a website) is relatively easy to do. You simply check every few minutes that the website is up, or that the server responds to pings.

Monitoring things that are supposed to happen periodically (like daily, weekly, or monthly backups) is a little harder. One of the best ways to do this is a [dead man’s switch](#), also known as a dead man’s handle. In a DMS system, an external service expects to be notified every time the event occurs, and raises the alarm if no report is received in a given time period.

There is a simple web service, [Dead Man’s Snitch](#), which allows for easy creation of an unlimited number of DMS-type monitoring checks for \$19 per month. We use this in a number of places to ensure that periodic events are working. In the event that a task fails to report, Dead Man’s Snitch will email our ticket address, notifying us of the failure.

Creating a DMS monitoring check

1. Log in to <https://deadmanssnitch.com/> using the account details stored in LastPass.
2. Create a new “snitch,” specifying the frequency at which the job should run. You will be given a URL that looks something like <https://nosnch.in/5e5879c9be>. This is the URL that you have to request (usually programmatically) in order to report in for a given job.
3. Make sure that the URL is called if and only if the job is successful. For example, here is a shell script that dumps PostgreSQL databases and calls DMS

```
#!/bin/sh

# This ensures that if any command in the script exits abnormally (i.e.
# with a non-zero exit code), the script will immediately abort.
#
# This is important, because it means that the request to Dead Man's
# Snitch will not happen if pg_dump or s3cmd fails.
set -e

# It's also a good idea to detect and treat the use of undefined
# environment variables as an error:
set -u

DUMPNAME="$(date +%Y%m%d%H%M%S)-mydatabase.dump"

# Dump the PostgreSQL database to a time
pg_dump mydatabase >"$DUMPNAME"

# Upload the dump to S3
s3cmd put "$DUMPNAME" s3://backupbucket/postgres/

# Record the file size
DUMPSIZE=$(ls -lhd "$DUMPNAME" | awk '{ print $5 }')

# If we got this far, all is well, and we should report to DMS. The
# message part is completely optional, but it can be useful to return
# basic stats (such as size of backup, or how long the task took), which
# are then displayed on the DMS web interface.
curl -sSL https://nosnch.in/5e5879c9be \
    -d "m=size: $DUMPSIZE" \
    >/dev/null
```

Domain registration and naming

Overview

Domain names are handled on two, technically independent levels:

- **Domain Registrar:** To register new domains, to make sure they point to the correct DNS servers, and to renew already registered domains. We use [gandi.net](#) (TLD price list).
- **DNS Service:** To administer DNS records (A/CNAME/MX/...) for domains which we have already registered. We use [DNS Made Easy](#) (“DME”).

The connection between the two levels is that you have to tell the registrar which DNS service you intend to use for a specific domain. These settings are called the **WHOIS NS records**.

Remark: Usually the order doesn't matter, but for some TLDs (e.g. .DE), a new domain has first to be configured properly at the DNS service before it can be registered or transferred.

1a: How to transfer an existing domain to OKF

For the current domain owner: So you are the current owner of a domain and you want to transfer it to OKF. You (or your domain admin) will have to prepare the domain transfer OR THE TRANSFER IT WILL FAIL. This is what you need to do:

- Get the **domain (transfer) secret** or **Auth(Info)Code** for the domain.
- Make sure the domain is not locked (some registrar call this “*disable Theft protection*”)
- Make sure any form of owner pseudonymization/anonymisation (“*Privacy protection*”) is disabled for the domain. Your mail address as domain owner should be visible in the domains **WHOIS record**.
- Make sure that very owner mail address is functional. The registry will send a confirmation request to this address.
- In case the DNS servers for the domain are not already ns{10,11,12,13,14,15}.dnsmadeeasy.com. write down the list of all DNS records (A, CNAMEs, MX etc) for this domain. This list is called the **zone file** for the domain.

When everything is prepared, do this:

- Send the above !AuthCode and (if required) the zone file to sysadmin@okfn.org.
- Watch mail for the confirmation request from Gandi and follow the confirmation link
- Watch mail for a notification from your current registrar. Usually if you don't do anything, the transfer will proceed after a couple of days automatically. Sometimes a confirmation link is included which allows you to speed up the transfer, but ...
 - **BEWARE: Some dodgy registrars send transfer notifications containing a “click here to cancel the transfer” link - DO NOT FOLLOW THAT LINK, the transfer would get cancelled immediately without any further questions..**

For the OKF domain admin:

- Verify the above requirements for domain transfer are given (not locked, not anonymized)
- If the domain is not yet served by the DME name servers: add the domains to DME (see below)
- Log into [gandi.net](#) with our handle OS1535-GANDI and transfer the domain in:
 - Using the “bulk transfer” option might speed up things

- Set our handle OS1535-GANDI (Open Knowledge Foundation, sysadmin@okfn.org) as new owner.
- Set DNS servers to ns{10,11,12,13,14,15}.dnsmadeeasy.com
- When the confirmation request to the future owner hits sysadmin@okfn.org, confirm it. If a transfer fails, log into Gandi, go to our account, *Orders in Progress*, click on *Detail* next to the *Failed* message and read the error message carefully. Usually it is telling.—

1b: How to register a new domain for OKF

This procedure will secure a *formerly non-existing* domain for OKF. And it will prepare the next step (setting DNS records) by telling the register what DNS will be used for To register a new domain, log in to the Gandi account, details are in lastpass. Domain registration is straightforward.

- Enter the domain name you want to register and click on Search for a domain name. This should bring up the results.
- Make sure the automatically selected domain name has the right address (.org or .com etc) and add to cart.
- Choose 1yr duration and make sure it has the correct contact details
- Pay using the pre-paid account.
- Now set the DNS servers to be the DME name servers:
 - In the “Domains” dashbord, click on the new domain name
 - In the bottom right section “dns Name servers”, click on “Modify servers”
 - Set the DME name servers:
 - * DNS1=ns10.dnsmadeeasy.com
 - * DNS2=ns11.dnsmadeeasy.com
 - * DNS3=ns12.dnsmadeeasy.com
 - * DNS4=ns13.dnsmadeeasy.com
 - * DNS5=ns14.dnsmadeeasy.com
 - * DNS6=ns15.dnsmadeeasy.com
 - * DNS7= [empty]
 - Press “Submit”
- Sign out.

2: How to add a domain to our DNS service

To add a new domain (e.g. “okfn.it”) to DME’s DNS service (which can be done before or after registration of the domain):

- Login into <https://cp.dnsmadeeasy.com/> as user *openknowledgefoundation*. Ask the Sysadmin/sysadmin team for the password.
- In the top right navigation, click on the leftmost “DNS”, then “Managed DNS”.
- Click on “Add Domains” and follow the dialog. (**ToDo: document the details**)

Note that DME allows so-called “ANAME records”. That are “internal CNAMEs that look from outside like A-records”. Very handy!

This is how you can test a DNS record without poisoning your local DNS cache with old data:

```
host -t a $DOMAIN_NAME_WITH_TAILING_DOT ns10.dnsmadeeasy.com.
```

e.g.

```
host -t a www.bibsoup.net. ns10.dnsmadeeasy.com.
```

You can observe a DNS record until you see the change:

```
watch host -t a $DOMAIN_NAME_WITH_TAILING_DOT ns10.dnsmadeeasy.com.
```

Now the change should show up in the global DNS system (provided the domain was properly delegated at its registrar to the DME DNS servers). If your local DNS cache isn't storing an outdated record, you should see the change now:

```
host -t a $DOMAIN_NAME_WITH_TAILING_DOT
```

Alternatively you could verify that “ping” is now using the correct IP address:

```
ping $DOMAIN_NAME
```

Remarks:

- “Always *test* DNS changes as described above! “
- Do not forget tailing dots when you create CNAMEs, otherwise e.g. `www.bibsoup.net.` might end up pointing to `s065.okserver.org.bibsoup.net.` instead of `s065.okserver.org.`

Email aliases and forwarders

Email aliases and forwarding is managed on [fastmail.fm](#).

How to add a new domain

Set the MX records to the new domain as below:

```
20 in2-smtp.messagingengine.com.
10 in1-smtp.messagingengine.com.
```

Now add the new domain in the [FastMail Virtual Domains configuration page](#) (log in from LastPass)

The configuration in fastmail should look like below.

yourtopia.net	N/A	Auto (Int) ▼	Us	No, reject ▼	Del
okfn.be	N/A	Auto (Int) ▼	Us	No, reject ▼	Del
okfestival.org	N/A	Auto (Int) ▼	Us	No, reject ▼	Del

If you cannot set Routing to Auto (Int) at the time of creation, submit the form and you will be able to change it.

Server hosting and building new server

How to spin up a new cloud instance

- **Preparation:** Decide which hoster & datacenter the new server shall live

in. Find the next free server number and set the new server's hostname.

- **Deployment (Amazon EC2):** Spin up a new instance using our AWS script, add a name tag and enable the “termination protection”. See also section *Access credentials* below.
- **DNS:** Add an A record for hostname and IP address.
- **Bootstrap:** To bootstrap the machine, use bootstrap.yml in the *infra* repository:

```
ansible-playbook bootstrap.yml --extra-vars="hostip=<newip> host=<newhostname> user=ubuntu" --ve
```

- **Monitoring:** Adding the machine to the respective host group for monitoring.

Access credentials

Credentials for the web interface should be in LastPass. Private keys and other shared files should be in the *okfn/credentials* <<https://github.com/okfn/credentials>> repo.

Mailing list hosting

- Runs on [GNU Mailman](#)
- Deployed at: <http://lists.okfn.org/>
- Details of setup/configuration at: [MailmanHowto](#)
- List creation password: std low-sec pwd

How to add a new mailing list domain

- **NOTE:** While its possible to create multiple domains in mailman, its upto the admin/users to ensure lists with duplicate names are not created.
- The current default is @lists.okfn.org
- To add more domains, append the domain name into `ansible/inventory/host_vars/s116.okserver.org.yml` in the relevant parts of the file

How to Rebuild list archives

mmarch

How to Rename a List

See mailman documentation :

- [How to rename a list](#)
- [How do I change the name of \(rename\) a list?](#)

NOTE: Some files and dirs need to be readable and/or writeable by user/group www-data, others by user/group “list”.

How to remove an individual post

We do not remove emails from Open Knowledge list archives as a matter of course. As the email has already been sent to the list, so to all subscribers, it is already public. Also constantly removing emails places a lot of extra work on the team.

It is possible to remove the email from the list archives, however this breaks all links in the list archive. Sysadmins will remove messages only for serious privacy reasons; the current policy is that all requests are rejected (unless there is a court order or equivalent legal obligation, of course).

This is quite painful. See also: [How can I remove a post from the list archive or remove an entire archive?](#)

Log into our mailman host (currently *s116.okserver.org*) and follow the below steps:

```
# What list do we want to edit?
ML="sysadmin-coord"

# Make a backup of the mbox file
cp -a /var/lib/mailman/archives/private/${ML}.mbox/${ML}.mbox.`SAVE-date +%Y%m%d`

# Make a working copy of the mbox file
cp -a /var/lib/mailman/archives/private/${ML}.mbox/${ML}.mbox.work

# Enter the mbox and remove the posting.
#BEWARE: *REMOVING* the post will cause renumbering of the subsequent posts in the archive.
#You might want to *EDIT* the post instead and remove its content. Not sure whether this can be done

mutt -f /var/lib/mailman/archives/private/${ML}.mbox/${ML}.mbox.work
# Remove the new headers mutt has left behind:
# BEWARE 1: Maybe other statuses are also possible, e.g. "R". Make sure they are removed as well
# BEWARE 2: This would also remove such lines from mail BODIES which would be wrong!

egrep -v '^(Status: O|Content-Length: [0-9]*|Lines: [0-9]*)$' /var/lib/mailman/archives/private/${ML}.mbox

# Check that we really only removed the one post:
diff /var/lib/mailman/archives/private/${ML}.mbox/${ML}.mbox.{work}

#Is all fine? The proceed:
#Remove the working copy
rm /var/lib/mailman/archives/private/${ML}.mbox/${ML}.mbox.work

#move the old HTML archive:
mv /var/lib/mailman/archives/private/${ML}{,.`SAVE-date +%Y%m%d`}

#re-create html archive:
sudo -u www-data mkdir -m 775 /var/lib/mailman/archives/private/${ML} sudo -u list /usr/lib/mailman/
```

How to remove a user from a mailinglist via cli

```
remove_members -n -N [addr1...]
```

How to add a user into a mailing list via cli

```
echo 'person@example.com' | add_members -r -wn listname
```

How to export mailinglist users

To list users of a particular list, with full names - '*Full Name*':

```
list_members --fullnames listname
```

How to upgrade mailman

Mailman does not like to be upgraded while it has messages in its queues. Therefore you should follow this procedure:

- Stop mailman.
- If you want, make a backup of the queues:

```
/var/lib/mailman/qfiles/{bad,shunt}/
```

- Remove “bad” and “shunt” queued messages:

```
sudo /usr/lib/mailman/cron/cull_bad_shunt
```

- Check whether there are still queue files:

```
sudo find /var/lib/mailman/qfiles/ -type f
```

If there are no more queued messages, you can upgrade mailman now. Otherwise proceed:

- Prevent the MTA (postfix in our case) from passing new postings to mailman, but make sure it still accepts mails *from* mailman. I am not sure whether stopping the postfix service would work, so instead i block port 25 temporarily:

```
sudo iptables -A INPUT -m state --state NEW -p tcp --dport 25 -i ! lo -j REJECT
```

- Start mailman and wait until the queues are empty:

```
sudo watch 'find /var/lib/mailman/qfiles/ -type f | wc -l'
```

- Stop mailman. Revert the above step that stopped postfix from passing messages to mailman, e.g. start postfix, or remove any block:

```
sudo iptables -D INPUT -m state --state NEW -p tcp --dport 25 -i ! lo -j REJECT
```

Now it should be safe to upgrade mailman.

Mailman Troubleshooting

Important folders

- Mailman folder:

```
/var/lib/mailman/
```

- Mailman private archives (all lists, mbox files):

```
/var/lib/mailman/archives/
```

- Mailman public archives (lists available via the web interface, html files):

```
/var/lib/mailman/archives/public
```

- Individual list config (stored in python pickle format):

```
/var/lib/mailman/lists/${list-name}/config.pck
```

Reading list config:

```
/var/lib/mailman/bin/dumpdb /var/lib/mailman/lists/${list-name}/config.pck
```

Modifying a list config

- Create a config file with content like 'key=value' pairs, key and value pairs can be read from the .pck file.
- In the config given below, we're modifying the list footer to include an unsubscribe link.

```
cat /root/mailman_list_config
mlist.personalize=1 mlist.msg_footer='_____ \n%(real_name)
mlist.digest_footer='_____ \n%(real_name)s mailing list\n%
# Apply the config to the list
/usr/sbin/config_list -i mailman_list_config ${list-name}
```

Cleaning the Postfix Queue on the Mailman Server

Occasionally, postfix on the mailman server will have a large queue because of rejections. Usually, there might a spam user who was sent emails which were rejected several times ending up in the queue.

When this happens, run the following command to get a list of users and the number of emails in the queue for them:

```
postqueue -p | grep '@' | grep -v bounces | sort | uniq -c
```

Get the most offending user from this list and delete them from the queue with the *postfix_queue_del.pl* script:

```
postfix_queue_del.pl spammer@example.com
```

Renaming lists

See this FAQ: <http://wiki.list.org/pages/viewpage.action?pageId=4030617>

Regenerate list archive HTML

In some cases, the [HTML archive](<https://lists.okfn.org/pipermail/<list-name>/>) because a post had to be removed from the list or, Mailman doesn't stops updating the HTML archive,

To regenerate the HTML archive

```
/usr/lib/mailman/bin/arch --wipe <list-name>
```

```
--wipe clears out the old HTML content
```

Bulk operations on mailing lists

Nick asked if we could normalise all the names of the mailing lists to lowercase. It turns out to be pretty easy. Create a python file like this, and call it *lowername.py*:

```
def lowername(mlist):
    mlist.real_name = mlist.real_name.lower()
    mlist.Save()
```

Then this can be run with the *withlist* program:

```
export PYTHONPATH=`pwd`
ls /var/lib/mailman/lists | while read list; do
    withlist -l -r lowername $list
done
```

Common list moderator password

We needed a better way to manage moderator passwords across lists, since mailman keeps one password - for multiple moderators on a list and this results in passwords being required to be reset and additional overhead.

The solution we came up with, was to have a shared password for all list moderators, and have it reset once every six months.

The script used to reset the moderator password across all lists is `[usr/lib/mailman/bin/change_mod_pw]` (https://github.com/okfn/infra/blob/master/bin/change_mod_pw)

- To update the moderator password for all lists:

```
/usr/lib/mailman/bin/change_mod_pw -a -p <newpasswd>
```

- To update the moderator password for a single list:

```
/usr/lib/mailman/bin/change_pw -l <listname> -p <password>
```

Run with `-help` for other options, you may require `-q` if the moderators do not need to be informed.

Archive a Mailing List

- Disable delivery to the mailing list using the aliases file. This is done by taking the mailing list address to `/dev/null` in `src/infra/ansible/roles/postfix/templates/s116.okserver.org/aliases` and then running `ansible`. It is a good idea to also redirect the meta-addresses, `-owner` and `-request` to `/dev/null` as well.
- Disable public advertisement of the list either by using the list administrative interface or by doing:

```
/usr/lib/mailman/bin/config_list -o /tmp/foo.py foo-list
sed -i 's@^advertised = 1$@advertised = 0@' /tmp/foo.py
sed -i 's@^emergency = 0$@emergency = 1@' /tmp/foo.py
/usr/lib/mailman/bin/config_list -i /tmp/foo.py foo-list
```

the second `sed` command is a belt and braces approach to disabling posting to the list, in addition to the aliases trick.

- Unsubscribe all members from the list:

```
/usr/lib/mailman/bin/list_members foo-list | xargs /usr/lib/mailman/bin/remove_members foo-list
```

Import a mailinglist into the OKF mailman server

In some cases, a mailman list needs to be imported into the OKF mailman server,

- To import a single list, from a remote host we require:

```
/var/lib/mailman/archives/private/<listname>.mbox  
/var/lib/mailman/lists/<listname>/config.pck
```

- Once the files are copied over to the OKF mailman server
 - We create a new list, the admin user doesn't matter as of now:

```
/usr/sbin/newlist
```

- Copy the mbox file:

```
cp /tmp/<listname>.mbox /var/lib/mailman/archives/private/<listname>.mbox/<listname>.mbox
```

- Generate HTML archives:

```
/var/lib/mailman/bin/arch --wipe <listname>
```

- Copy over the config.pck file which contains the users info and list config:

```
cp /tmp/config.pck /var/lib/mailman/lists/<listname>/config.pck
```

- Update the list url in config.pck:

```
/var/lib/mailman/bin/withlist -l -r fix_url <listname> https://lists.okfn.org/mailman/
```

- The admin interface, html archives should now be accessible with all its users:

```
https://lists.okfn.org/mailman/admin/<listname>  
https://lists.okfn.org/pipermail/<listname>
```

Monitoring and Graphs

We use a bunch of tools for monitoring - which are [Nagios](#), [check_mk](#)

Each of these are managed as a role in the [Ansible OKF infra repo](#)

Host and service monitoring

[Nagios](#) is setup to monitor hosts and services.

[check_mk](#) is used along with [nagios](#), since [check_mk](#) allows managing [nagios](#) config as a template. Checks are mainly passive, thus reducing load on the [nagios](#) server. The [check_mk](#) agent is installed on each host, which adds passive checks for host services.

All [nagios](#) config is managed using [Ansible](#) from within the [check-mk-server](#) and [nagios-server](#) roles.

Add a host into Nagios To add a host for monitoring, the host just needs to be listed under the right group in the [Ansible Inventory](#). Which is either [ckan](#), [openspending](#) or [sysadmin-server](#). Normally add the server in to it's own relevant group and then add that group into the [sysadmin-server](#), [ckan](#) or [openspending](#), depending as to what's needed.

```
https://github.com/okfn/infra/blob/master/ansible/inventory/hosts
```

Once the host is added into the inventory, running the [check_mk-server](#) play on the target host, which should take care of the rest.

```
ansible-playbook main.yml --tags="check_mk_config_main" -i inventory/hosts -c  
ssh -vvv
```

Remove a host from Nagios To remove a host from nagios, it must be removed from the Ansible [inventory](#).

We also need to remove any `.yml vars` file for the host.

The next step is to update the `check_mk` config, by running:

```
ansible-playbook main.yml --tags="check_mk_config_main" -i inventory/hosts -c ssh -vvv
```

Other monitoring related Ansible flags We have a bunch of flags in Ansible to simplify managing checks for each host, they are documented here in the [OKF infra repo](#)

Request Tracker

We use Request Tracker (rt.okfn.org) as our ticketing system for a number of services within Open Knowledge. This is a short overview of the current sysadmin queues and how they're to be managed.

sysadmin-triage

All tickets come into `sysadmin-triage` (apart from the ones which go directly to alerts). This queue should be checked on a regular basis and the tickets need to be moved into the relevant queues below.

sysadmin

Anything that's not Wordpress, Community or an Alert. So basically it's server or software related.

sysadmin-alert

All Nagios alerts go into here. Currently they aren't auto-closed, but it's something we're hoping to do.

sysadmin-wordpress

All wordpress related tickets go here.

sysadmin-community

All community issues go into here. Mostly end-user issues. If it's a server issue, it should be in `sysadmin`.

Prevent RT outgoing emails to a specified user

Some email addresses from which RT receives email are `noreply@` email addresses. By default, the emails which RT sends these users will usually create additional noise in the form of "Undelivered Mail" notifications.

The easiest way to prevent these is to remove the email address from the autogenerated user. As long as the user's username still matches the email address from which tickets are created, RT will correctly associate tickets with that user, but won't attempt to email them.

1. Go to *Admin* → *Users* → *Select* in the global navigation. (If you do not see an *Admin* button, you are not an RT administrator and will not be able to perform this task. Please contact someone who is.)

2. Find the user by putting their email address in the *Go to user* box and submitting the form.
3. Remove the user's email address from the *Email* form field.
4. Save your changes using the *Save Changes* button at the bottom right of the form.

Wikifarm

This article is about our MediaWiki-based wikifarm.

The Mediawiki instance hosts multiple wiki sites, its based on: http://www.mediawiki.org/wiki/Manual:Wiki_family#Scenario_2:_Quick_up

Add a new wiki site to the farm

1. Create a new mysql db for the new site with the name `wiki_<domain>`

```
#mysql -u root -p
mysql> CREATE DATABASE wiki_<domain>;
mysql> CREATE USER wiki_<domain> IDENTIFIED BY <pass>;
mysql> GRANT ALL ON wiki_<domain>.* TO wiki_<domain>;
mysql> quit
```

New Wiki site configuration

- On the wikifarm host, we create a new configuration file for the site.
- You can use the existing `LocalSettings-wiki.okfn.org.php` config as a template.
- Ideally set the name of the config file as `LocalSettings-<domain>.php`.

1. Update the db credentials we setup in the previous step.

```
# vi /var/www/wiki/LocalSettings-<domain>.php

$wgDBtype           = "mysql";
$wgDBserver         = "localhost";
$wgDBname           = "wiki_dm2e_eu";
$wgDBuser           = "wiki_dm2e_eu";
$wgDBpassword       = "w1k1dm2e";
```

2. Make sure that all required extensions similar to `wiki.okfn.org` are enabled.

```
require_once( "$IP/extensions/AntiBot/AntiBot.php" );
require_once( "$IP/extensions/AntiSpooof/AntiSpooof.php" );
require_once( "$IP/extensions/SimpleAntiSpam/SimpleAntiSpam.php" );
require_once( "$IP/extensions/TitleBlacklist/TitleBlacklist.php" );
require_once( "$IP/extensions/SpamBlacklist/SpamBlacklist.php" );
```

3. Once you are done with the instance's configuration file set it up to be accessible - add a case statement for the new site.

```
vi /var/www/wikifarm/wiki/LocalSettings.php

case "<wiki domain>.name":
    require_once LocalSettings-<domain>.php;
    break;
```

4. Run the Mediawiki update script:

```
cd /var/www/wikifarm/wiki/maintenance
php update.php --conf /var/www/wikifarm/wiki/LocalSettings-<domain>.php
```

If update.php fails, LocalSettings-<domain>.php might need fixing - some extensions might not be present or have other names.

Setup DNS records

Dns records for the domain will need to be updated to point to this host.

Nginx config

Once the wiki instance is setup, the nginx config which is maintained in ansible for the wiki host will need to be updated.

3.2.4 Step-by-step HOWTOs for common problems

HOWTO: Decommissioning a server

Decommissioning a virtual or physical server is straightforward, but it is important to follow a documented process to ensure that unnecessary alarms are not raised and that important data is saved.

1. Ensure that users are given appropriate notice.

It is important that servers and services don't disappear unexpectedly if people rely upon them. Make sure that any users of the machine are given plenty of time to retrieve data or raise issues with the machine's decommissioning.

Find out what services are hosted on the machine and make posts to appropriate mailing lists to announce the scheduled retirement of the machine.

2. Take appropriate backups of machine state.

Unless there is a good reason to do otherwise, it is probably a good idea to back up `/etc`, `/home`, `/var/lib/{mysql, postgresql}` and similar kinds of data. This is not an exhaustive list.

Backups should usually be simple tarballs of the data in question, and should be uploaded to:

```
s3://attic.okfn.org/archive/<hostname>-<backupname>-<yyyymmdd>.tar.gz
```

For example:

```
s3://attic.okfn.org/archive/s123.okserver.org-etc-20140401.tar.gz
```

3. Schedule downtime for the machine in Nagios.

This can be done using the Nagios user interface, or you can get Mowgli to do it. For example, to schedule `s123.okserver.org` for one day's downtime:

```
!down s123 1d
```

4. Power down the machine. The exact process to do this will be vendor-specific.

5. Ensure that the right machine has gone offline according to Nagios. This sounds stupid, but do it!

6. Destroy the machine. The exact process to do this will be vendor-specific.

7. Remove the machine from the Ansible inventory.
8. Run Ansible on the monitoring machine(s) to remove any monitoring for that host. It's quickest to just run the `check-mk-server` play, so use `-t` to do that. In the `infra` repository:

```
cd ansible/  
ansible-playbook -i inventory/hosts -t check-mk-server -l monitoring main.yml
```

This will force `check_mk` to regenerate its inventory and update Nagios.

9. Remove any remaining entries relating to the machine in question from DNS.

3.2.5 Onboarding a sysadmin

Things that need doing when a new sysadmin comes on board:

- reencrypt credentials repo with their GPG key
- add to sysadmin group and run ansible across all machines
- WPENGINE: - my.wpengine.com account - superadmin privileges on the blogfarm
- Add to sysadmin-team in Google Groups
- Google analytics
- Lastpass Superadmin
- Google Apps admin
- Instruct to subscribe to `sysadmin-coord@`, `outages@`, and `sysadmin-announce@`
- Pagerduty invite
- AWS IAM account
- rtadmin on `rt.okfn.org`

3.3 Who are we?

- Nigel Babu: Senior sysadmin
- Matthew Moore: Sysadmin

At various times, the following people have provided advice and support to the sysadmin team:

- Neal Bastek
- Joel Rebello
- James Casbon
- Michelle Heydon
- Rufus Pollock
- Nick Stenning
- Nils Toedtman
- Will Waites

What is this?

The web pages you’re viewing are maintained as a [Sphinx](#) project on [GitHub](#). Anyone with appropriate permissions on GitHub can edit the content, and there is a short guide to getting started:

4.1 Getting started with the Ops Manual, reStructuredText, and Sphinx

4.1.1 Basics

The Ops Manual is maintained as a [Sphinx](#) project on [GitHub](#). This means that the pages you’re viewing are plain text files (actually, they’re in a format called [reStructuredText](#), of which more later) which live in a [git repository](#). Editing the manual is as simple as clicking the “Edit on GitHub” link which appears in the sidebar of each page. When you click “Commit changes,” your changes will be built and pushed to the live copy of the manual that you’re looking at. It might take a minute or two before your changes are live.

Alternatively, if you’re more comfortable on the command line, you can check out the repository and work on the files on your own computer:

```
git clone git@github.com:okfn/opsmanual
cd opsmanual
vim opsmanual-getting-started.rst
git commit -am "Improve the getting started guide"
git push
```

You can also build the project locally by running `make html` from the root of the project. The built site will be placed in `_build/html`. You don’t need to do this, but if you’re making large edits it can help you preview your changes before pushing.

4.1.2 Ops Manual Structure

The manual is divided into a few sections, each broadly aimed at a particular group in a particular role. Existing sections include:

guides/ Guides to getting stuff done at the Open Knowledge Foundation. Most people will want to start here.

sysadmin/ Internal documentation and content aimed primarily at members of the OKF sysadmin team.

The tables of contents that form the main navigational structure of the site are maintained in `index.rst` files at various points. See the `toctree` directives in, for example, `index.rst` or `2nd-line/index.rst`. If you’re adding a new

page you should ensure it's included in at least one `toctree` so it's discoverable from the root of the site. (Although even if you don't add it to a ToC, it will still show up in the site search.)

4.1.3 reStructuredText

The manual pages are written in a format called `reStructuredText`, or rST. At its simplest, rST is much like the `Markdown` format with which you may be more familiar. You can view the rST source of any page on this site by clicking the “Show Source” link in the left-hand column.

Some useful resources for getting started editing rST documents:

- [Quick reStructuredText](#)
- [rST demonstration document \(and its HTML output\)](#)
- [Markdown to rST converter](#)

A note on linking

If you want to link to another part of the manual, you can use the `:doc:` Sphinx role ([more documentation about this here](#)) to reference another file in the repository. For example, you can say:

```
More information on widgets can be found in :doc:`manufacturing/widgets`.
```

This will insert a reference to the `manufacturing/widgets.rst` file relative to the current file.

If you want to reference a particular part of another document, you can do so by inserting a label:

```
.. _a-label:
```

Note the `..`, the opening `_` and the closing `:`. The label goes before the heading you want to link to. It is **ESSENTIAL** that this label is followed by a blank line.

You can then refer to the label from anywhere else in the manual using the `:ref:` Sphinx role ([more documentation about this here](#)):

```
You should refer to :ref:`Whatever you want <a-label>` for more information on this subject.
```

4.1.4 The Build

The content you push to GitHub constitutes the “source format” of the manual. This source content is compiled by `Sphinx` into the web pages you are currently viewing. This is done by a `Travis CI` job. The output of this job will also inform you of any rST syntax errors or other problems `Sphinx` encountered when building the site.

The build artefacts (a static HTML website) are copied into the `gh-pages` branch of the git repository and pushed to GitHub. The manual is then served at:

<http://ops.okfn.org/>

Indices and tables

- [search](#)
- [sitemap](#)

5.1 README

[Go here](#)

Playing with this locally, you probably want to use make:

```
$ make html
```