

This is the Administrator reference guide for Server123 instances. Note that:

- 1 **Wherever you see 'example.com' in this document, you should substitute your own domain name**
- 2 References to the 'console' in this document refer to the Server123 administrator front-end at <https://example.com/admin>. This is distinct from the Keycloak console, which is at <https://example.com/keycloak/admin>
- 3 The 'Site Administrator' refers to the person whose details were given during configuration. See the Site Administrator section in Figure 3 of the Installation Guide. This person has a name, email address, and three passwords, all of which are referred to below
- 4 Server123 does not use any container technology, including Docker. All applications are natively installed. Java, Keycloak, Alfresco, and Redmine are installed at `/opt`; Roundcube is installed at `/var/www1`; WordPress, DokuWiki, and MediaWiki are installed at `/home/wpuser`.

Glossary

2FA	Two-Factor Authentication
CLI	Command Line Interface
GCM	Git Credential Manager
GCO	git-credential-oauth
IdP	Identity Provider
MFA	Multi-Factor Authentication
OIDC	OpenID Connect
SCM	Source code management; normally used in a Git context. The same as VCS
TOTP	Time-based One-Time Password
VCS	Version Control System (for Server123, either Git or Subversion)

CONTENTS

1	Initial setup	5
1.1	Data partition mount	5
1.2	First OIDC sign in	5
1.3	Alternative OIDC sign in	5
1.4	Additional setup	6
1.5	Initial system checks	6
2	OIDC and Keycloak.....	7
2.1	Introduction	7
2.2	OIDC sign in	7
2.3	User creation.....	9
2.4	OIDC claims	10
3	Site access.....	11
3.1	Introduction	11
3.2	Endpoints and access control	12
3.2.1	Keycloak console	13
3.2.2	Server123 console	14
3.2.3	Roundcube webmail.....	14
3.2.4	WordPress	14
3.2.5	Wikis	14
3.2.6	phpMyAdmin.....	15
3.2.7	Git and Subversion browsers	15
3.2.8	Git and Subversion command-line access.....	15
3.2.9	File share	15
3.2.10	Redmine	15
3.2.11	Alfresco.....	15
4	ssh	16
4.1	System users	16
4.2	ssh configuration.....	17
5	Email	19
5.1	Introduction	19
5.2	Initial state	20
5.3	Mail users.....	21
5.3.1	Deleting mail addresses	22
5.4	Mailboxes	23

5.4.1	Mailbox deletion	24
5.5	Mail client setup.....	24
5.5.1	Roundcube	24
5.5.2	Thunderbird.....	25
5.6	Mail tests.....	25
5.7	Message backup and deletion	26
5.7.1	Roundcube	26
5.7.2	Thunderbird.....	26
6	WordPress	27
6.1	WordPress update	27
6.2	Create your first page	28
6.3	Site Health Status.....	28
6.4	URL and directory structure.....	28
7	DokuWiki.....	29
7.1	DokuWiki Update	29
7.2	Create your first page	30
7.3	.htaccess files	30
7.4	Further reading	30
8	MediaWiki.....	30
8.1	MediaWiki Update	31
9	phpMyAdmin	31
10	VCS Repositories.....	32
10.1	Introduction	32
10.2	OAuth 2.0 Operation.....	33
10.3	Creating OAuth 2.0 CLI users	34
10.4	Repository browsing	34
10.5	OAuth 2.0 Git client configuration, Linux.....	34
10.6	OAuth 2.0 Git client configuration, Windows.....	35
10.7	Testing Git	36
10.7.1	Git fetch.....	36
10.7.2	OAuth 2.0 sign in	37
10.7.3	Git push	37
10.8	Other operations.....	38
10.8.1	Delete	38
10.8.2	Rename	38

10.8.3 Create	38
11 File share.....	39
12 Redmine.....	41
12.1 API access.....	41
13 Alfresco	42
13.1 Authentication	42
13.2 Heap settings	43
13.3 Edit in Microsoft Office	43
13.4 Edit in Google Docs	43
13.5 Using Alfresco from Microsoft Office	43
13.6 Network share.....	43
14 Server backup	44
14.1 Server files.....	44
14.2 Server state	44
14.3 rsync.....	45
14.4 burp.....	46
14.5 Enabling backups	47
14.5.1 Server	47
14.5.2 Client	47
Figure 1: Keycloak user account creation	9
Figure 2: Server endpoints.....	12
Figure 3: ssh configuration.....	17
Figure 4: Initial mail users	20
Figure 5: Initial mailboxes	20
Figure 6: Mail user creation	21
Figure 7: Mailbox creation	23
Figure 8: CLI login screen	33
Figure 9: WebDAV setup.....	39
Figure 10: GNOME files mapping.....	40
Figure 11: Registry basichostallowlist.....	42

1 INITIAL SETUP

When you have completed Stage 2 configuration the server will reboot. If you have enabled disk encryption, you will now have to enter your LUKS passphrase to mount the data partition. See 1.1 below for instructions.

To access any server functionality you will now have to carry out an 'OIDC sign in', using the procedure in 1.2 below.

1.1 DATA PARTITION MOUNT

If encryption is enabled, you must enter a LUKS passphrase whenever the server reboots. There are two possibilities here:

1. If you selected the 'Auto mount' option during configuration, the passphrase will be requested during the boot process, which will halt until a valid passphrase is supplied. You will have to do this from your VPS control console (or, if you are running on bare metal, your console or tty)
2. The server will otherwise complete the boot, but without mounting the `/data` partition. You should now browse to your site, where you will be redirected to <https://example.com/dmount> to enter your passphrase. If this is entered correctly, the `/data` partition will be mounted.

The server has reduced functionality while the `/data` partition is unmounted; incoming emails will not be accepted, for example. The sender will re-attempt transmission (generally for 2 to 5 days), but you should not leave the partition unmounted for extended periods.

1.2 FIRST OIDC SIGN IN

In normal use, you will sign into the server with an 'OIDC sign in' (see 2.2 below for more information on OIDC). You should carry out your first sign in using the Site Administrator details that you entered during configuration (see Figure 3 in the Installation Guide), with the Site Administrator's first name, and `Site Administrator Password 2`. During the first sign in, you will have to carry out these additional actions:

- 1 Configure 2FA. You will first need to install an authenticator app on your phone. Keycloak suggests using one of Google Authenticator, Microsoft Authenticator, or Free OTP. The Keycloak sign in form will display a QR code which you should scan from your app. The app should then give you a 6-digit code, which you can enter on the sign in form
- 2 Change your password
- 3 Keycloak will then send an email to the Site Administrator email address; you will need to click the confirmation link in this email.

If you cannot sign in using this procedure (if you do not want to set up 2FA, or cannot reply to a confirmation email), then you should instead use the procedure in 1.3 below.

1.3 ALTERNATIVE OIDC SIGN IN

If necessary, you can modify the sign in procedure from the Keycloak console. This will let you (among other things) disable 2FA, or the requirement to verify your email address. To do this, sign

into the Keycloak console, as the master realm admin user (see 3.2.1 below for instructions). The admin user does *not* require a 2FA login, in case this procedure is necessary. You should now:

- 1 Select the 'vserver' realm in the top-left pulldown
- 2 Click [Users](#), and select the only user (the Site Administrator)
- 3 On the [Credentials](#) tab, find the [Otp](#) row, click on the 3-dot icon, and [Delete](#)
- 4 On the [Details](#) tab, set [Email verified](#) to 'Yes'
- 5 Click [Save](#)
- 6 Log out ([Sign out](#) in the top right admin pulldown)

You should now be able to sign in as the Site Administrator without configuring 2FA/OTP, or verifying your email, as necessary.

1.4 ADDITIONAL SETUP

When you have access to the console, you should consider these further actions to complete setup:

- 1 **You will not initially be able to log in with ssh.** This is because the configuration process sets ssh login to key-only, and there are no public keys on the server. If you require ssh, you should use the console to either upload a public key, or to enable password login, for one or more of the system users. See 4 below
- 2 After configuration, the Keycloak master realm admin user can sign in using only a password (in case procedure 1.3 above was required). This is a security issue, and you should consider enabling 2FA. Instructions are given in 3.2.1 below. You should also consider restricting the IP addresses which can access the Keycloak console. This can be carried out from the Server123 console, at [Keycloak > Admin IP addresses](#)
- 3 If you have enabled encryption, you should download your drive's LUKS header, and store it somewhere safe. You can do this from the console, at [Administration > File download > LUKS header](#). You will need this file if you ever need to repair the header on your server. See the [LUKS FAQ](#) for more details
- 4 You should download and securely store the Alfresco keystore, from [Administration > File download > Alfresco keystore](#)
- 5 The server is initially set up with one mailbox, named [sysadmin](#). This mailbox can be accessed from the Roundcube webmail front-end (see 3.2.3 below), or directly, from your own email client, with password [Site Administrator Password 3](#). This password is not particularly secure, and you should consider changing it. You can do this from the Server123 console, at [Email > Mailboxes](#)
- 6 WordPress and DokuWiki, if used, may require updates. Update instructions are given in sections 6.1 and 7.1 below.

1.5 INITIAL SYSTEM CHECKS

These tests are not necessary, and should all pass on a newly-configured server. However, you may wish to repeat them:

- 1 Confirm that SSL/TLS has been correctly set up; see (5.3) in the Installation Guide
- 2 Confirm that rDNS, SPF, DKIM, and DMARC have been correctly set up; see (5.6)
- 3 Confirm that [ssh](#) has been suitably hardened with an [sshd](#) test; see (0)
- 4 Carry out a port scan with [nmap](#); see (3.1)

2 OIDC AND KEYCLOAK

2.1 INTRODUCTION

Access to your server is protected by the Apache web server, which operates as an OIDC Relying Party (RP). Apache relays end user authentication requests to an OIDC Provider (OP) and receives user identity information from that Provider. The identity information is returned as a set of *claims*.

Apache can be configured to use public Providers, such as Google, Microsoft, and Facebook. However, this is of limited use in most environments, and Apache is instead configured to use a [Keycloak](#) instance, which itself runs on your server. Keycloak maintains a local user database which is used to authenticate and authorize your users. **Any 'OIDC sign on' referred to in this manual is therefore a 'Keycloak sign on', rather than a sign on with any other Provider.**

Keycloak operates with two 'realms': the *master* realm, and the *vserver* realm. The master realm contains only a single user, named `admin`. This user can view and manage any other realm on the server instance. The admin is therefore effectively the 'superuser'.

Regular site users are placed in the *vserver* realm. After initial configuration, this realm contains a single user. This user is the Site Administrator (the Site Administrator was created during configuration; see Figure 3 in the Installation Guide).

The Keycloak console is at <https://example.com/keycloak/admin>, and can be used to manage any aspect of Keycloak functionality. However, Keycloak management is difficult and error-prone, so the Server123 console includes a simple front-end which can instead be used to manage users in the *vserver* realm (but *not* the master realm). The management functions include account creation, modification, deletion, listing, and unlocking, as well as claim assignment and password resets (see [Keycloak > User accounts](#)). These should be sufficient for normal Server123 usage, but additional operations can be carried out from the Keycloak console if necessary. However, you should take care if you use both consoles for user management, since this could lead to confusion.

2.2 OIDC SIGN IN

During an OIDC sign in, Keycloak generates a web page which asks for a username and password. An email address can be used for login, rather than a username. If 2FA is enabled for the user, Keycloak will then ask for a TOTP.

The master-realm Keycloak admin can only sign in to the Keycloak console page (see 3.2.1 below). This sign in page shows a 'KEYCLOAK' heading. The admin does not have any claims, and so is restricted to Keycloak console usage (the admin cannot, for example, access <https://example.com/mail>, which requires a `Mail` claim).

For all other site usage, users sign in to the *vserver* realm (the sign in page instead shows a 'VSERVER' heading). These users have a set of claims which allow them to access specific parts of the site.

By default, all HTTPS accesses to the site require an initial OIDC sign in, with the exceptions noted in the list below:

- 1 The WordPress, DokuWiki, and MediaWiki instances can be individually set to public access from the Server123 console (see [Others > PHP applications](#)). If public access is enabled, site

visitors will be able to access your WordPress site or wikis directly without an OIDC sign in. Note, however, that the WordPress dashboard requires an OIDC sign in (with a [WordPress](#) claim) unless you have enabled IP access

- 2 Command-line (CLI) access to Git and Subversion is carried out over HTTP and requires authentication. The auth method can be set to one of anonymous, HTTP Basic, or OIDC, from the console (see [Others > VCS](#)). The default is HTTP Basic, but any attempt to carry out a CLI access will fail until an HTTP Basic account is created. HTTP Basic accounts can be created from the console. If OIDC is instead selected, a [vcs](#) claim is required
- 3 WebDAV file sharing requires authentication. The auth method can be set to either HTTP Basic or OIDC from the console (see [Others > File share](#)). The default is HTTP Basic, because few, if any, file management programs currently support OIDC. An HTTP Basic account must be created before files can be shared; account creation can again be carried out from the console. If OIDC is instead selected, a [Files](#) claim is required.

The user has been *authenticated* if the sign in is successful. The returned access token contains a set of claims, each of which grants access to a particular resource on the server (webmail, Redmine, and so on). If a token includes a given claim, the user is *authorized* to use that resource.

2.3 USER CREATION

New Keycloak users can be created from the Server123 console ([Keycloak > User accounts > Create account](#)), by entering the following information:

User accounts [?](#)

Operation	List all accounts	<input type="radio"/>
	Create account	<input checked="" type="radio"/>
	Modify account	<input type="radio"/>
	Delete account	<input type="radio"/>
	Reset password	<input type="radio"/>
	Unlock account	<input type="radio"/>

First name

Surname

Username

Email

Password [👁](#)

Enabled ☒ Y ☐ N

2FA ☒ Y ☐ N

CLAIMS

- ☐ AdminWrite
- ☐ AdminRead
- ☐ WordPress
- ☐ Wiki
- ☐ Files
- ☐ VCS
- ☐ Mail
- ☐ Alfresco
- ☐ Redmine

Figure 1: Keycloak user account creation

When a new user is created with 2FA enabled, the user listing page (under 'List all accounts') will not initially reflect this. The console's [Keycloak > User accounts](#) page will not show 2FA as enabled until the user has set up their authenticator app and signed in.

When a new OIDC user first signs in the actions listed in 1.2 above will be taken. The default behaviour forcing a password change and email authentication can't be changed from the Server123 console. However, it can be changed from the Keycloak console, if necessary; see 3.2.1 below.

2.4 OIDC CLAIMS

OIDC users can be assigned zero or more Claims during account creation, or at a later time when modifying an account. Each claim grants access to a resource on the server. The available claims are listed in Table 1 below. The initial Site Administrator user is given all 9 claims.

Claim	Resource
AdminWrite	Allows the client to access the Server123 console (at <code>/admin</code>) and carry out operations which change the state of the system (enabling or disabling ssh, or adding new Keycloak users, for example). Any user with this claim is an 'administrator'.
AdminRead	Allows the client to view the Server123 console (at <code>/admin</code>), but not to change any state.
WordPress	Allows the client to access the WordPress site at <code>/www</code> . Note that the WordPress site may also be configured as public; if this is the case, the WordPress claim is ignored. This claim is always required to access the WordPress dashboard (at <code>/www/wp-admin</code>), whether or not public access has been set.
Wiki	Allows the client to access the Wikis at <code>/mediawiki</code> and <code>/dokuwiki</code> . The client may still be required to log into the Wiki itself. Note that the Wikis may also be configured as public; if this is the case, the Wiki claim is ignored.
Files	If OIDC authentication has been set for WebDAV file sharing, this claim allows the client to access the file share area at <code>/files</code> . The client can both read and write files. File share authentication can alternatively be set to <code>HTTP Basic</code> ; if so, this claim is ignored.
VCS	This claim allows the client to browse the Git and Subversion repositories. If OIDC authentication has been set for CLI access, this claim additionally allows a git or svn client to access the repositories. CLI authentication can alternatively be set to <code>Anonymous</code> or <code>HTTP Basic</code> ; if so, this claim is used only for browser authorisation.
Mail	Grants access to the <code>/mail</code> endpoint, which is the front-end of the Roundcube webmail app. This claim does not automatically allow access to a Dovecot mailbox; the client must additionally supply a mailbox name and a password for that mailbox.
Alfresco	Grants access to the Alfresco front-end, at <code>/alfresco</code> or <code>/share</code> . This claim does not automatically log the client into Alfresco; an Alfresco login is still required.
Redmine	Grants access to the Redmine front-end, at <code>/redmine</code> . This claim does not automatically log the client into Redmine; a Redmine login is still required.

Table 1: OIDC claims

3 SITE ACCESS

3.1 INTRODUCTION

The following ports are open:

80/http	HTTP is used only by Apache for security certificate negotiation. HTTP traffic is otherwise redirected to 443
443/https	Normal web traffic
465/submissions/smtps	Mail client message submission to Postfix (SSL/TLS)
587/submission	Mail client message submission to Postfix (STARTTLS)
993/imap	Mail client message retrieval from Dovecot
25/smtp	Postfix communication with other mail servers

In addition, a randomly-selected port is opened for ssh; see 4 below.

All regular site traffic (in other words, anything which is not ssh, or which is not controlled by your email client, or external Postfix communication) is therefore over HTTPS, because of the port 80 redirection. This is generally referred to as 'HTTP' in this document, for simplicity.

The accessible HTTP endpoints are set by the Apache configuration ([/etc/apache2/sites-enabled/vserver2.conf](#)) and are listed in 3.2 below. Access to these endpoints requires an OIDC sign in (see 2.2 above), unless noted otherwise.

You can, if necessary, carry out a port scan from another computer to confirm that no other ports are open. Nmap is suitable for this, and can be run as follows (in this case, ssh is on port 52260):

```
$ sudo nmap -v -sT -p- example.com
...
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https
465/tcp   open  smtps
587/tcp   open  submission
993/tcp   open  imaps
52260/tcp open  unknown
```

3.2 ENDPOINTS AND ACCESS CONTROL

The server endpoints are listed in Figure 2 below. This page is shown if you have carried out an OIDC sign in, but have entered an invalid address (and you have not enabled WordPress; see 6.4 below). If you have not yet signed in, you will instead be redirected to the Keycloak sign in form.

Invalid address

The primary services and endpoints provided by this server are listed below.
Note that:

- You may require additional authorisation to access one or more of these endpoints. Your sysadmin will be able to give you further information
- Alfresco, WordPress, and phpMyAdmin must be explicitly enabled before they can be accessed

Site administration	
Server123 console	https://example.com/admin
/data partition mount	https://example.com/dmount
Keycloak	
Keycloak admin console	https://example.com/keycloak/admin
Front page	https://example.com/keycloak
Logout	https://example.com/keycloak/logout
Applications	
Alfresco front page	https://example.com/alfresco
Alfresco share	https://example.com/share
DokuWiki	https://example.com/dokuwiki
File share	https://example.com/files
Git repo browser	https://example.com/gitweb
MediaWiki	https://example.com/mediawiki
phpMyAdmin	https://example.com/phpmyadmin
Redmine	https://example.com/redmine
Roundcube webmail	https://example.com/mail
Subversion repo browser	https://example.com/svnweb
WordPress admin	https://example.com/www/wp-admin

Figure 2: Server endpoints

By default, all web-accessible locations on the server are OIDC-protected, unless noted otherwise. This means that you must carry out an OIDC sign in, and have a relevant claim (see 2.4 above). However, the access controls for specific locations can be modified from the console, as detailed below.

Access control modifications are carried out by modifying the Apache configuration. Apache is reloaded after these changes, which may lead to users being logged out. If you do manually modify the Apache configuration yourself, you should take care not to change anything related to any `Define` directives.

There are two additional endpoints for git and Subversion access to the repositories, but these are accessed from CLI clients, and not a browser, and so are not listed here.

Note that:

1. The root location (/) is accessed as `https://example.com/`. In the description below, `/mediawiki`, for example, is actually `https://example.com/mediawiki`. URLs are shown here without a trailing / ; the webserver automatically adds the / if it is not present
2. Unless otherwise stated, the endpoints listed here are permanently enabled, and can be accessed at the address shown. The exceptions are the WordPress and phpMyAdmin endpoints, which can be disabled from the console. Alfresco can also be started and stopped from the console, and defaults to stopped (any access to the `/alfresco` and `/share` endpoints will therefore return a 'Service Unavailable' page).

3.2.1 KEYCLOAK CONSOLE

The Keycloak console is located at `https://example.com/keycloak/admin`, and should not be confused with the Server123 console (at `https://example.com/admin`). You should not need to use the Keycloak console in normal operation. To access the Keycloak console, log in as user `admin`, with `Site Administrator Password 1`.

By default, the Keycloak console can be accessed from any IP address. For added security, you can restrict the allowable client IP addresses from the Server123 console [Keycloak > Admin IP addresses](#) page.

2FA is *not* enabled by default for the admin user, to simplify the process of modifying the Keycloak configuration, if necessary, during first sign-on. To enable 2FA, use this procedure after logging in to the Keycloak console:

- 1 Make sure the master realm is selected in the top-left pulldown
- 2 Click [Users](#), and select user `admin`
- 3 On the [Details](#) tab, select the [Required user actions](#) pulldown
- 4 Select [Configure OTP](#)
- 5 Click [Save](#)
- 6 Log out ([Sign out](#) in the top right admin pulldown)

Now log in again. You should now be asked to activate the account by scanning a QR code in your mobile authenticator app. Your app should now give you a 6-digit code; enter this, and [Submit](#). You should be returned to the admin page. You can use a similar procedure to change the admin password, to enable email verification, and so on.

Note that, if you do enable 2FA for the admin user, and you need to use Keycloak's `kcadm` script, it will no longer work if you attempt to log in with a password. You should instead log in with the `service-cli` client, and the relevant secret.

3.2.2 SERVER123 CONSOLE

The console is located at `/admin`, but requires an `AdminRead` or an `AdminWrite` claim to access.

You should log in with the Site Administrator's first name, and `Site Administrator Password 2`. You can change the password from the `Keycloak > User accounts > Modify account` page.

3.2.3 ROUNDKUBE WEBMAIL

The webmail page is located at `/mail`, but requires a `Mail` claim to access.

After signing in, you must additionally log into Roundcube, with a `mailbox` name and password. A newly-configured system has only one mailbox, named `sysadmin`, with `Site Administrator Password 3`. This password is not particularly secure (it is also used for the initial MySQL passwords), and you should change it (from the console's `Email > Mailboxes > Modify mailbox` page).

3.2.4 WORDPRESS

WordPress is configured from the console, at `Others > PHP applications`. The site can be enabled or disabled, and can be set to either public or private access. In this context, 'site' means the WordPress front end; the WordPress dashboard is handled separately (see below). The `IP address`, `Enable`, and `Public` selections apply only to the front end.

The 'Public' selection refers only to whether or not an `OIDC` sign in is required to access the site. If it is set, no sign in is required; if it is not set, only visitors who have carried out an `OIDC` sign in, with a `WordPress` claim, are granted access. This is unrelated to any additional mechanisms that WordPress itself may have for marking pages as public or private.

By default, WordPress is not enabled. However, this is somewhat misleading: the site is still served, but will be visible only to the address given in the `IP address` field, if there is one (see the online help on the popup for address details). This allows you to develop your site without public access (you could, of course, instead set the site to private, so requiring an `OIDC` sign in).

When enabled (or when viewed from the specified IP address) the WordPress front page is at `https://example.com/www`. When the site is set for public access, `https://example.com` is redirected to `https://example.com/www`, so the WordPress front page is also visible at the site root.

The WordPress dashboard is at `/www/wp-admin`, and is always enabled *and private*. A `WordPress` claim is required for access (unless you have enabled access from your IP address). You will need to carry out an additional login as a WordPress user to access the dashboard. A newly-configured system has only one WordPress user; you should log in with the Site Administrator's first name, and `Site Administrator Password 3`.

3.2.5 WIKIS

Both wikis are always enabled, and default to private access. The wikis are available at `/dokuwiki` and `/mediawiki`, and can be independently set to public or private access from the console's `Others > PHP applications` page.

When set to private access, an `OIDC` sign in with a `Wiki` claim is required to access the relevant Wiki.

3.2.6 PHPMYADMIN

phpMyAdmin can be enabled from the console's [Others > PHP applications](#) page. By default, it is not enabled. When enabled, it is located at [/phpmyadmin](#).

An OIDC sign in with an [AdminWrite](#) claim is required to access phpMyAdmin.

3.2.7 GIT AND SUBVERSION BROWSERS

The browsers are located at [/gitweb](#) and [/svnweb](#), respectively. The browser functionality is always enabled, but requires an OIDC sign in with a [vcs](#) claim to access.

3.2.8 GIT AND SUBVERSION COMMAND-LINE ACCESS

The [/git](#) and [/svn](#) endpoints are accessed from CLI applications, and *not* from a web browser. The authentication method can be set to one of anonymous, HTTP Basic, or OIDC from the console's [Others > VCS](#) page.

When set to OIDC, an OIDC-capable CLI app will open a browser window to carry out an OIDC sign in. The user must have a [vcs](#) claim to access any repositories. No distinction is made between read and write access to the repository.

3.2.9 FILE SHARE

The [/files](#) endpoint should normally be accessed from a WebDAV-enabled file manager, but a browser can be used for viewing. The authentication method can be set to either HTTP Basic or OIDC from the console's [Others > File share](#) page.

When set to OIDC, an OIDC-capable file manager will open a browser window to carry out an OIDC sign in. The user must have a [Files](#) claim for access.

3.2.10 REDMINE

Redmine is located at [/redmine](#), and is always enabled. An OIDC sign in is required, with a [Redmine](#) claim.

3.2.11 ALFRESCO

Alfresco is located at [/alfresco](#) and [/share](#). Authentication can be set to either Password, or OIDC.

When password authentication is set, an OIDC sign in is not required, and you must log in with an Alfresco username and password. When authentication is set to OIDC an [Alfresco](#) claim is required for access.

Alfresco requires significant resources, and is not started when the system boots. You can start (or stop) it from the console's [Others > Alfresco](#) page.

4.1 SYSTEM USERS

There are two unprivileged (non-root) users:

- 1 `sysadmin`, who has `Linux user password 1`. Note that this user is unrelated to the 'Site Administrator', whose details were entered during configuration. If you need to carry out general maintenance operations, you should ssh in as this user
- 2 `wpuser`, who has `Linux user password 2`. The WordPress and wiki installations are in this user's home directory, and `wpuser` owns the WordPress and wiki files. If you need to carry out any WordPress or wiki operations you should use ssh, or a file transfer program (such as WinSCP), as this user.

Neither `sysadmin` nor `wpuser` are sudo-capable (in other words, they are not in group `sudo`), because of the security issues. Both users must carry out operations which require root permissions, but this is handled by listing the relevant scripts in the sudoers file (`/etc/sudoers.d/sysadmin`). If you need to carry out other root operations, you should `su` to root.

`wpuser` has no access to sensitive system files, and can only modify files in `/home/wpuser`. You can safely give this login to whoever is responsible for your WordPress and wiki sites. `sysadmin` *can* access sensitive files, and you should ensure that this login remains secure.

4.2 SSH CONFIGURATION

ssh can be configured from the console, at [Administration > ssh configuration](#):

ssh configuration [?](#)

The screenshot shows the 'ssh configuration' interface. At the top, there is a section for 'Enable ssh' with radio buttons for 'Y' (selected) and 'N'. Below this, there is a 'Port selection' field containing the value '52265'. Underneath the port field, there is a section for 'System account: root' with 'Ssh enable' (Y selected), 'Password enable' (N selected), and 'Source address' (Any). This is followed by a section for 'System account: sysadmin' with 'Ssh enable' (Y selected), 'Password enable' (Y selected), and 'Source address' (Any). Finally, there is a section for 'System account: wpuser' with 'Ssh enable' (Y selected), 'Password enable' (N selected), and 'Source address' (Any). At the bottom of the form are 'Submit' and 'Cancel' buttons.

Figure 3: ssh configuration

A random port is selected for ssh usage during site configuration. You will need to make a note of this port number for use with the ssh, scp, or similar commands (in this case, `ssh -p52265` or `scp -P52265`). You can alternatively set the required port yourself. You can set this to 22, or any port in the range [1024, 65535]. We suggest:

- 1 Do not use 22, or common replacement ports such as 2222. Internet-facing servers with an open port 22 will be subject to persistent penetration attempts
- 2 Use a port greater than or equal to 49152 to avoid any services which may be enabled in the future.

By default, ssh is enabled for all 3 users, but password login is disabled. If you enable password login, the relevant password is the one supplied during site configuration (`Linux root password` for root, `Linux user password 1` for sysadmin, or `Linux user password 2` for wpuser).

The best practice is to disable password logins, and to instead use a public and private key pair (key pair login is always enabled). You can use your own key pairs, or you can instead download a newly-generated key pair from the console, at the [Administration > Downloads](#) page. You can select PuTTY, OpenSSH, or OpenSSL keys (see the online help for more information on these keys). In general, however, you should select PuTTY if you have a Windows client and intend to use programs such as PuTTY or WinSCP, and OpenSSH otherwise.

When you have downloaded a new key pair, you should upload the *public* key back to the server (the server will refuse to upload private keys), from the [Administration > Uploads](#) page. The server's key pair generation code is specifically written to avoid leaving any trace of the private key on the server (it runs from RAM, does not use the command history, and so on). However, if this concerns you, you should instead generate your own key pairs elsewhere.

The console sets the required ssh configuration by modifying `/etc/ssh/sshd_config`. If you need to modify this file yourself, you should change nothing below the [DO NOT CHANGE ANYTHING BELOW HERE](#) line. If you do change this file, you should confirm that your new configuration is still sufficiently hardened. You can do this by running this command as root, and comparing the output before and after any changes:

```
vserver # sshd -T -f /etc/ssh/sshd_config | grep -iE "\
PermitRootLogin|MaxAuthTries|LoginGraceTime|PasswordAuthentication|\
PermitEmptyPasswords|KerberosAuthentication|GSSAPIAuthentication|\
X11Forwarding|PermitUserEnvironment|AllowAgentForwarding|\
AllowTcpForwarding|ClientAliveInterval|ClientAliveCountMax|\
MaxStartups|MaxSessions|AllowUsers|PermitTunnel"
```

5 EMAIL

5.1 INTRODUCTION

Mail to your server is handled by Postfix and Dovecot, with a Roundcube webmail interface. The console provides a front end for configuring Postfix and Dovecot, at [Email > Mail users](#), and [Email > Mailboxes](#).

The [Mail users](#) form is used to create email addresses at your domain. Incoming emails to these addresses can be delivered to a local mailbox, or redirected to another address, or both. There are no restrictions on the number of users that you can create.

The [Mailboxes](#) form is used to create named mailboxes on your server. Mailboxes store incoming emails, which can then be retrieved by IMAP clients. The mailboxes are located in the data partition on your server, and so will be encrypted if you selected encryption during server configuration.

In normal use, you will probably have a one-to-one correspondence between users and mailboxes. Emails to, for example, [johndoe@example.com](#) could be stored in a mailbox named [johndoe](#). However, this is not a requirement. A single mailbox can handle emails from any number of user addresses; you might have a mailbox for the entire support department, for example. When retrieving emails using an IMAP client, the client must log in with the name of the *mailbox*, and not with a specific email address. The client will then retrieve the messages in that mailbox, irrespective of the destination email address. Instructions for setting up your IMAP client are given in 5.5 below.

MESSAGE SIZE LIMIT

The maximum mail message size is left at the Postfix default, of 10240000 bytes. Message attachments are Base64-encoded, which introduces an overhead of about 33%. This means that the maximum attachment size that can be handled is about 7.4 MB. This is normal in mail systems; SMTP was not intended for sending large binary files. If you have large attachments to share, you should consider using WebDAV file sharing (see 11 below), or some alternative file share mechanism.

MAIL WHITELISTING

A pass on all the mail tests listed in 5.6 below may not be sufficient to guarantee mail delivery. The big webmail providers, in particular, will apply various heuristics in an attempt to detect spam. The most likely problem is that mails from newly-registered domains may be rejected, or classed as spam. Your recipients may therefore initially have to whitelist your domain, or mark an incoming mail as 'not spam'.

MISSED INCOMING MAILS

There are circumstances in which your server will not be able to accept incoming emails (if it is turned off, for example, or if the data partition is encrypted and is not currently mounted). This is a normal feature of all mail systems, and the SMTP protocol is designed such that the sender knows whether or not the receiver has assumed responsibility for mail delivery. The sender will periodically re-attempt message delivery if necessary. Postfix, for example, will by default attempt another delivery after 5 minutes, and will continue delivery attempts for 5 days before reporting the message as undeliverable.

5.2 INITIAL STATE

After configuration, your server has a single email address, which is `sysadmin@example.com`. The user listing (at [Email > Mail users](#)) will look as follows:

Mail users [?](#)

Operation

List all addresses

☒

Create address

☐

Modify address

☐

Delete address

☐

Addresses

	Address	Mailbox	Redirects
1	sysadmin	sysadmin	

Submit

Cancel

Figure 4: Initial mail users

This means that incoming emails to *address* `sysadmin@example.com` will be delivered to a *mailbox* which is named `sysadmin`. In this case, no redirects have been set up, so delivery is only to the mailbox. The mailboxes ([Email > Mailboxes](#)) are set up as follows:

Mailboxes [?](#)

Operation

List all mailboxes

☒

Create mailbox

☐

Modify mailbox

☐

Delete mailbox

☐

Mailboxes

	Mailbox	User name	Quota	Usage
1	sysadmin	vServer system	100 MB	10 MB

Disk available: 93.5 GB / 93.9 GB (99.7%)

Submit

Cancel

Figure 5: Initial mailboxes

This means that there is a single mailbox, which is named `sysadmin`. The user name is descriptive only, and is set to `vServer system`. This mailbox has a small quota (it can contain a maximum of 100MB of mail messages), and currently contains a small number of messages (the usage is 10 MB). Note that 'Disk available' is listed as, in this case, 93.5 GB. You should keep an eye on this figure when creating new mailboxes, and when setting mailbox quotas.

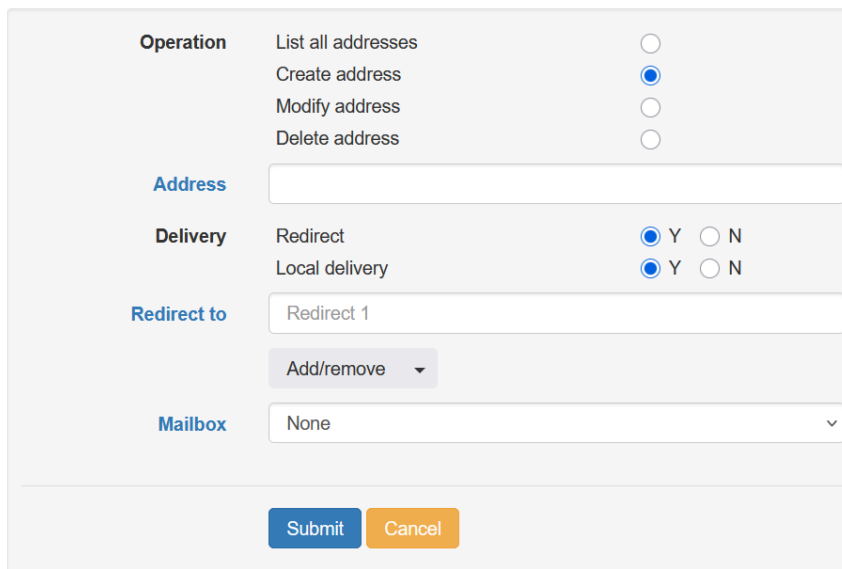
The `sysadmin` address and mailbox are used for various system-related functions (heartbeat emails, notifications of reboots, and so on). You should therefore not delete them. However, if they are accidentally deleted, you can simply re-create them.

The procedure for reading mail from this mailbox is covered in 5.5 below. Note that you need to log in with your *mailbox* name, not the 'User name' shown above. Some email clients (including Thunderbird) will require you to log in with your mailbox name at your domain (`sysadmin@example.com`), while others (including Roundcube) require only the mailbox name.

5.3 MAIL USERS

A mail user has an email address. Incoming mails to this address will be delivered to a local mailbox, or will be redirected elsewhere, or both. This information is specified as shown in Figure 6 below when creating a mail user.

Mail users



Operation	List all addresses	<input type="radio"/>
	Create address	<input checked="" type="radio"/>
	Modify address	<input type="radio"/>
	Delete address	<input type="radio"/>
Address		
<input type="text"/>		
Delivery	Redirect	<input checked="" type="radio"/> Y <input type="radio"/> N
	Local delivery	<input checked="" type="radio"/> Y <input type="radio"/> N
Redirect to	<input type="text" value="Redirect 1"/>	
	<input type="button" value="Add/remove"/>	
Mailbox	<input type="text" value="None"/>	
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>		

Figure 6: Mail user creation

ADDRESS

Enter only the *prefix* of the required address in the Address field (everything before the @ character). Similarly, if you want to modify an existing address, the names which appear in the pull-down will be the prefixes. In principle, the prefix of an email address can be almost arbitrarily complex. In practice, however, complex names generally have limited support in the wider email ecosystem, so you should keep your names simple. We recommend setting all letters to lower-case, and starting the name with a letter. The remaining characters should be chosen from [a-z](#), [0-9](#), and [_.-](#) (underscore, dot, dash). Do not use [+](#); this has a special meaning in email prefixes, which is described below.

CATCH-ALL ADDRESSES

In normal usage, you will want mails to your domain to be delivered *only* if the mail is to a recognised address prefix. However, in some cases you may wish to receive mails to *any* address at your domain. To do this, create a new address, and enter the address as [*](#). Incoming mails to your regular addresses will be processed normally, and the catch-all entry will be then used for unmatched address prefixes.

PLUSSED ADDRESSES

When you create a new address, you are actually creating a family of 'plussed addresses', rather than a single simple address. These additional addresses have a suffix which must start with a **+** character. The suffix can be arbitrarily complex, but you should use only the characters listed above to ensure deliverability. If you create address **harry**, for example, then emails to **harry+a98** will also be delivered to Harry's mailbox.

CASE SENSITIVITY

Only the domain name in an email address is case-sensitive. If you have set up an address for 'sales', then these are both valid addresses for incoming emails (although upper-case letters are not recommended, as noted above):

- **sales@example.com**
- **SalEs@example.com**

But this is not a valid address:

- **sales@Example.com**

DELIVERY

Select **Redirect to another address**, or **Local delivery**, or both.

If you select redirection, an additional field will appear to add one or more addresses to redirect to (the **Add/remove** pulldown will allow you to enter additional redirects). Assume, for example, that you want to create email address **harry@example.com**, but that these emails should simply be forwarded on to **harry1000@gmail.com**. In this case, you should enter **harry** under **Address**, and you should enter **harry1000@gmail.com** under **Redirect to**.

If you select local delivery, an additional pull-down field will appear; this field will list the names of all existing mailboxes. You should select one of these for your local deliveries. This means that you must create a mailbox *before* adding any mail addresses which will be delivered to that mailbox.

Note that you can deliver multiple addresses to a single mailbox.

5.3.1 DELETING MAIL ADDRESSES

You can delete a mail address by selecting **Delete address**. Any mails sent to this address will then be rejected by the server, and returned to the sender as undeliverable ('bounced'), unless you have opted to receive 'catch all' addresses (see above).

Note that deleting a mail address which has local delivery enabled does *not* delete any messages that were previously received by that address. These messages are stored in the relevant mailbox, and the mailbox itself must be deleted to delete any messages.

5.4 MAILBOXES

A mailbox is a storage area on your server which contains a record of your incoming email messages. You can read these messages with any IMAP-compatible client (including the Roundcube webmail client, which is already installed on your system).

Mailboxes must be named, since your users will need their mailbox name in order to log into the server and read their messages. In practice, a given user's mailbox name will probably be the same as their email address, which can be confusing. You might have a mailbox named `sales`, for example, which stores the incoming mails for the user with email address `sales@example.com`. However, you should bear in mind that the two names are not related, and that the messages for several different email addresses could potentially be stored in a single mailbox.

You can create, modify, or delete a mailbox using the [Email > Mailboxes](#) form. A mailbox has a name, a description, a password, and a disk quota. Figure 7 below shows these fields when initially creating a mailbox.

Mailboxes ?

Operation	List all mailboxes	<input type="radio"/>
	Create mailbox	<input checked="" type="radio"/>
	Modify mailbox	<input type="radio"/>
	Delete mailbox	<input type="radio"/>
Mailbox name	<input type="text"/>	
User name	<input type="text"/>	
Mailbox password	<input type="password" value="Password"/>	
Disk quota	<input type="text" value="500"/> Mbytes	
<div>Submit Cancel</div>		

Figure 7: Mailbox creation

MAILBOX NAME

This should normally be something simple and generic, such as `sales`, or `support`. The person who currently uses this mailbox may change in the future: it makes more sense to name it `support`, for example, rather than `tim.bisley`.

USER NAME

This is an optional field which describes the mailbox. You could use it to enter a name, for example, such as `Tim Bisley`. Note that this is *not* the user name which will be used to log into the mailbox.

MAILBOX PASSWORD

This is the password which will be used to log into the mailbox. This has the same restrictions as a normal user password (see the Installation Guide for supported password formats). When the mailbox has been created, the password can be changed in one of two ways:

1. Your users can change their own password from Roundcube, or
2. You can change a password from the 'Modify mailbox' selection. You will need to do this if a user loses his or her password.

When modifying an existing mailbox, you should leave the password field empty unless you intend to change it.

DISK QUOTA

Set this to the maximum allowed size of the mailbox on disk. When the mailbox has exceeded this size, incoming emails will be bounced back to the sender with an "exceeded quota" message. The mailbox user will receive warning messages before the mailbox fills, however; these are sent when the mailbox has reached 80% capacity, and then 95% capacity. The user should take action to reduce the mailbox size before it fills; see 5.7 below.

The default size of 500 megabytes should be adequate in most cases.

When you set a disk quota, make sure that you take into account your server's disk size, and the remaining disk space available. The remaining disk space can be seen on the 'List all mailboxes' form (see Figure 5).

If you do not want to enforce a quota for this mailbox, set this field to 0.

5.4.1 MAILBOX DELETION

If a mailbox is no longer required, it should be deleted to save disk space. You can do this by selecting [Delete mailbox](#). You will be asked to confirm your selection, since mailbox deletion is unrecoverable.

5.5 MAIL CLIENT SETUP

You can use any IMAP email client to read or write emails. Your server has a Roundcube webmail interface, which is described in 5.5.1 below; you can alternatively install your own IMAP client. The IMAP settings for one particular client (Thunderbird) are given in 5.5.2 below, but the settings will be the same for any other client.

5.5.1 ROUND_CUBE

[Roundcube](#) should be accessed as described in 3.2.3 above. Roundcube includes online help, which can also be found at <http://docs.roundcube.net/doc/help/1.1/>. Your users must log into Roundcube using their *mailbox* name and password. There is initially a single mailbox, named `sysadmin`, with `Site Administrator Password 3`. This mailbox receives any mail which is addressed to `sysadmin@example.com`.

Roundcube's Settings menu allows users to carry out a number of additional actions, which include:

- Users can change their own mail passwords. You will set the initial password for a given mailbox when creating the mailbox; you should ask the user to change this password when they first log in. Note that users cannot change their passwords unless they already know their current password. If a user has lost his or her password, you should create a new one from the Mailboxes menu
- Users can create new folders and message filters
- Users can create automatic message responses (for holiday auto-replies, for example)

5.5.2 THUNDERBIRD

Thunderbird can be downloaded from <https://www.thunderbird.net/>. The IMAP settings for incoming mails should be set to:

Server	example.com
Port	993
Connection security	SSL/TLS
Auth method	Normal password
User name	The user's mailbox name, but (unlike Roundcube) this is <i>at your domain</i> . If you have created mailbox 'sales', for example, then the user name should be sales@example.com . This name is used to log into the server.

The SMTP settings for outgoing mails should be set to:

Server	example.com
Port	465
Connection security	SSL/TLS
Auth method	Normal password
User name	As above

STARTTLS can instead be selected for outgoing mails, on port 587. SSL/TLS should be preferred, however, since it is less susceptible to MITM (Man In The Middle) attacks. You should restart Thunderbird if you change the security settings.

5.6 MAIL TESTS

Outgoing emails will generally not be delivered if your server's reverse DNS (rDNS) has not been set up correctly (see section 4.2 in the Installation Guide). To check your rDNS setting, go to the MxToolBox tester at <https://mxtoolbox.com/ReverseLookup.aspx>, and enter your site IP address. Click [Reverse Lookup](#). This will take a few seconds to run. On completion, it must report that this IP address points to [vserver.example.com](#) (and *not* simply [example.com](#)). There are a large number of other rDNS testers if MxToolBox is not available.

To check SPF, DKIM, and DMARC, you should send an email to one of the bigger webmail providers (from your mail client; see 5.5 above) and check their delivery report.

For Gmail, you'll find a 'More' icon (three vertical dots) at the right of the page. Click this, and select [Show original](#). This page should now show a PASS for all of SPF, DKIM, and DMARC. This can be a little more difficult with other providers, but you will, generally, have to find a way to show the original message. You can then do a search for SPF, DKIM, and DMARC, and confirm that they are shown as passing.

5.7 MESSAGE BACKUP AND DELETION

Messages can be downloaded or deleted both from Roundcube, and from your mail client (which is assumed to be Thunderbird). This will be necessary if your mailbox fills, and must be reduced in size.

5.7.1 ROUNDKUBE

Messages can be downloaded by selecting them, and then selecting 'Download' from the 'More' menu. The message(s) remain on the server after a download.

When you delete a message from the Inbox, it will be moved to 'Deleted Items' (the message is not actually 'deleted', and the mailbox remains at the same size). To reduce the mailbox size you must then empty the 'Deleted Items', or individual messages within 'Deleted Items'.

5.7.2 THUNDERBIRD

You have two options for reducing your mailbox size:

1. You can move messages to your local computer. To do this, you have to move the messages to Thunderbird's 'Local folders'. Before starting, make sure that your `Local Folders` contains an `Inbox`; create the Inbox (potentially with additional sub-folders) if not. Now select the messages you want to move. Right-click on the selected messages, and select `Move to`. Now select `Local Folders > Inbox`. Note that, unlike the Roundcube download, this operation *does* actually delete the messages from the server.
2. You can delete messages. By default, these are simply moved to your local `Deleted` folder; when you delete messages inside the Deleted folder, they will be removed from the server. This is also how Roundcube operates.

Both these options will delete messages from the server, so reducing the size of your mailbox. However, unlike Roundcube, Thunderbird does not immediately delete messages on the server, which can be confusing. By default, the mailbox will only shrink when Thunderbird closes, or when you 'Compact' the folder that you have just removed messages from. There is a procedure to set up Thunderbird for immediate removal, but this procedure does not work on Thunderbird 102. You may wish to try the instructions below if you have a later version.

To set up Thunderbird for immediate deletes, you have to run the 'Config editor'. This can currently be found from the Settings menu. If you scroll to the bottom of the page, you should find a `Config editor` button. When you're in the editor, type `expunge` into the search bar. Two settings are relevant:

1. `mail.imap.expunge_after_delete`, which defaults to `false`. Double-click on this setting to set it to `true`
2. `mail.imap.expunge_option`, which defaults to 0. Edit this, and set it to 1. Note that you have to click the 'tick' icon to save the change.

Now close and restart Thunderbird. If you have a Thunderbird version with the relevant fix, deletes from the 'Deleted' directory will now immediately shrink the mailbox.

6 WORDPRESS

Section 3.2.4 above contains instructions for enabling WordPress, and setting it for public or private access. The WordPress site itself is available at <https://example.com/www> (or <https://example.com>, if public access is enabled).

An OIDC sign in is required to access the WordPress dashboard, which is conventionally located at <https://example.com/www/wp-admin>. The dashboard can also be accessed through these URLs:

```
example.com/www/login
example.com/www/wp-login.php
example.com/www/admin
example.com/www/wp-admin
example.com/www/wp-admin/index.php
```

New WordPress users can be added from the dashboard. The user will be sent an email asking them to log in and change their password. Since the dashboard is OIDC-protected, **you must give new users a Keycloak sign in with a WordPress claim (see 2.3 above) before adding them as a new WordPress user**. This is true whether or not the WordPress site has been set for public access.

A single WordPress user was created during site configuration, with these details, and with an administrator role:

Username	Site Administrator first name
Public display name	Site Administrator full name
Email	Site Administrator email
Password	Site Administrator Password 3

The WordPress installation is at </home/wpuser/wordpress>. The database is at </data/var/lib/mysql/wordpress>, and so will be encrypted if the data partition is encrypted.

In normal use, the installation is locked down as a security precaution. In this state the WordPress files are set such that their ownership is `wpuser:www-data`, and the permissions are set such that the web server (`www-data`) can only write to files which it has a need to write to. WordPress cannot install new plugins, or update the installation, or modify the `.htaccess` file, in this state (you will get an error message such as "The update cannot be installed because some files could not be copied"). **To carry out any updates you must first reverse the lockdown; see 6.1 below.**

You should update WordPress before using it; see the instructions below.

6.1 WORDPRESS UPDATE

In order to update, you will first have to reverse the WordPress lockdown. To do this, select **Others > PHP applications** from the console. In the WordPress section, set **Unlock** to **Y**, and **Submit**.

You can now carry out any required WordPress or plugin updates by clicking on the relevant links on the dashboard. When you have completed the updates, you should lock down the site from the console, by setting **Unlock** to **N**, and clicking **Submit**.

6.2 CREATE YOUR FIRST PAGE

To confirm that the Block Editor is functional, and that you can create a new page, you should first log into the dashboard, and then:

- 1 Navigate to [Pages > Add Page](#). This will show two blocks, containing the text 'Add title', and 'Type / to choose a block'
- 2 Over-write the first box with a new title (such as 'Test')
- 3 Overwrite the second box with the new page text (such as 'This is my new page')
- 4 Click the [Publish](#) icon, and then click it again to confirm that you are ready to publish. You should see a message confirming that your test page is now live
- 5 Click on the WordPress 'W' icon to return to the dashboard
- 6 Navigate to [Settings > Reading](#)
- 7 Under 'Your homepage displays', select 'A static page', and select your new test page
- 8 Click [Save Changes](#)

Note that the active theme after configuration is 'Twenty Twenty-Three'. A number of different themes are installed; you can select a different one by navigating to [Appearance > Themes](#), and then activating one of the others.

6.3 SITE HEALTH STATUS

You can carry out a site check from the dashboard, by navigating to [Site Health Status > Site Health screen](#). This may show a number of potential issues, all of which can be ignored:

- 1 'Background updates are not working as expected', or '.htaccess is not writeable': when the site is locked down, WordPress cannot, by design, update itself. This is a security measure; the site must be unlocked to allow WordPress updates. This can also result in notifications that 'A scheduled event has failed' if the event requires deletion of a file.
- 2 'You should remove inactive plugins' (or themes): the installation includes a small number of inactive plugins and themes; you can remove these if you are not going to use them
- 3 'Page cache is not detected but the server response time is OK': the installation does not include a page cache plugin; you can, if desired, install one.

If WordPress is not enabled, and you are accessing the site from the IP address specified under [Others > PHP applications](#), then you may additionally get warnings for 'The REST API did not behave correctly' and 'Your site could not complete a loopback request'. This is because, in this state, accesses are allowed only from the given IP address, and not from the server itself.

6.4 URL AND DIRECTORY STRUCTURE

Both `home` and `siteurl` are set to `https://example.com/www` in the WordPress database (these are the back-end settings for `WP_HOME` and `WP_SITEURL` in `wp-config.php`, or 'WordPress address' and 'Site address' in the dashboard). The WordPress site is therefore visible (if enabled) at `example.com/www`. If WordPress is set to public access in the vServer console then accesses to `example.com` will be automatically redirected to `example.com/www`, so the site will be visible at both `example.com` and `example.com/www`.

The WordPress PHP code is installed at `/home/wpuser/wordpress`.

7 DOKUWIKI

The DokuWiki wiki is available at <https://example.com/dokuwiki>. The wiki can be set to public access, or can alternatively be set to require an initial OIDC sign in (see 3.2.5 above).

After configuration, a single user is created. The user details are set from the Configuration page Site Administrator fields as follows:

User	Site Administrator first name
Real Name	Site Administrator full name
Email	Site administrator email
Password	Site Administrator Password 3

This user is placed in groups 'admin, user'.

The DokuWiki installation is at </home/wpuser/dokuwiki>. The data directory is located in the data partition (at </data/var/dokuwiki/data>; see DokuWiki's [local.php](#)), to simplify backups, and to allow sensitive data to be encrypted if necessary.

In normal use, the installation is locked down as a security precaution. In this state the DokuWiki files are set such that their ownership is [wpuser:www-data](#), and the permissions are set such that the web server can only write to files which it has a need to write to. You cannot install new plugins, or update the installation, in this state. If you need to update the installation you will first have to reverse the lockdown; see 7.1 below.

You should update DokuWiki to the latest release before using it; see the instructions below.

7.1 DOKUWIKI UPDATE

Server123-0625 ships with the stable 'Librarian' release, from May 2025. If you wish to install a later release, you will first have to clear the DokuWiki lockdown. To do this, select [Others > PHP applications](#) from the console. In the DokuWiki section, set [Unlock](#) to [Y](#), and [Submit](#).

You can proceed with the update when you have released the lockdown. This is a two-stage process: you may first have to update the 'Update' plugin, and then use the plugin to update the distribution itself.

To update, browse to <https://example.com/dokuwiki>, and log in as the DokuWiki user (see above). Now click on [Admin](#) at the top of the page, and then click on [Extension Manager](#). Scroll down the new page until you find the 'DokuWiki Upgrade Plugin'. If there is an [Update](#) option, click on it. You should then get a status message at the top of the page stating that 'Plugin upgrade updated successfully'.

Click on [Admin](#) to go back to the Administration front page. Under 'Additional Plugins' there should be a 'Wiki Upgrade' link; click on this. In the new 'Wiki Upgrade' page click on [Continue](#). The upgrade is a 5-step process. None of these stages should report an error, and you should click [Continue](#) for each stage. On completion, you should get a message stating that 'Your Wiki has been updated'.

You should now lock down your new installation by reversing the unlock procedure above (set [Unlock](#) to [N](#), and click [Submit](#)).

7.2 CREATE YOUR FIRST PAGE

If you click on the DokuWiki logo at the top left you will be taken to your start page. There is a pencil icon at the right; if you hover over this, it expands to 'Edit this page'. Click on this. Enter some text and click [Save](#).

If you now log out, and log in again, this will be visible as your start page.

7.3 .HTACCESS FILES

DokuWiki is distributed with a number of `.htaccess` files which limit public access to sensitive data. However, these are not compatible with the security mechanisms used on your server, and have been replaced by various directives in the main Apache configuration, which have exactly the same effect. You can confirm this by logging out, and then attempting to access the following pages, which should all return a 'Not authorised' page:

<https://example.com/dokuwiki/conf/local.php>

<https://example.com/dokuwiki/data/pages/start.txt>

<https://example.com/dokuwiki/data/pages/wiki/dokuwiki.txt>

<https://example.com/dokuwiki/bin/plugin.php>

7.4 FURTHER READING

You can find general DokuWiki security information at these links:

<https://www.dokuwiki.org/security>

<https://www.dokuwiki.org/install:permissions>

However, you should not need to take any additional security measures, other than remembering to keep your installation locked down except when installing new plugins or upgrading.

There is a DokuWiki user forum at <https://forum.dokuwiki.org>.

8 MEDIAWIKI

The MediaWiki wiki is available at <https://example.com/mediawiki>. The wiki can be set to public access, or can alternatively be set to require an initial OIDC sign in (see 3.2.5 above).

After configuration, a single user is created. The user details are set from the Configuration page Site Administrator fields as follows:

User	Site Administrator first name
Email	Site Administrator email
Password	Site Administrator Password 3

This user is created in the `sysop` (administrators) and `bureaucrat` groups. Note that MediaWiki is not interested in the real names of users (although 'realname' extensions exist), and that usernames are always capitalised, even if you supply a lower-case name.

The MediaWiki installation is at `/home/wpuser/mediawiki`. The database is located in the data partition (at `/data/var/lib/mysql/mediawiki`) to simplify backups, and to allow sensitive data to be encrypted if necessary.

In normal use, the installation is locked down as a security precaution. In this state the DokuWiki files are set such that their ownership is `wpuser:www-data`, and the permissions are set such that the web server can only write to files which it has a need to write to. You cannot install new plugins, or update the installation, in this state.

8.1 MEDIAWIKI UPDATE

You can find your MediaWiki version at `https://example.com/mediawiki/index.php/Special:Version`. You can alternatively find this by selecting **Special pages** in the left-hand menu, and then **Version**, under the 'Data and tools' section.

Server123-0625 includes v1.43, which is the latest LTS (Long Term Support) version at the time of writing, with an End-of-life of December 2027 (see the [Version Lifecycle](#) page at the MediaWiki site).

MediaWiki does not have a straightforward upgrade procedure, and the user is expected to download the files for a new release, and then over-write the existing installation. Should you wish to upgrade, you should first read the [Manual:Upgrading](#) page, and the release notes for the new version.

The procedure for locking and unlocking the installation is identical to the WordPress and DokuWiki procedures (see 6.1 above), but note that the lock procedure can be slow, taking up to 2 minutes to complete.

9 PHPMYADMIN

phpMyAdmin provides a graphical front-end to the MariaDB database (this may be referred to as 'MySQL' elsewhere in this document, for simplicity, but is currently actually MariaDB). Any actions you can carry out from phpMyAdmin can also be carried out from the command line on the server, but you may find phpMyAdmin to be more convenient.

Access instructions are given in 3.2.6 above. Note that phpMyAdmin must be enabled from the **Administration > Others > PHP applications** page. It should be enabled only when required, and not left unnecessarily enabled. When you have carried out an OIDC sign in, you will additionally need to provide a database username and password. The available usernames are listed on the **Administration > Others > MySQL passwords** page. All usernames initially have the `Site Administrator Password 3` password.

When logged in with a given username, you will only have access to the databases relevant to that user (unless you login in with the `admin` username, which has global access). A login with username `wpuser`, for example, gives you access to the WordPress and MediaWiki databases.

The phpMyAdmin log out icon is at the top left of the page, and is an image of an open door with an arrow leaving it.

10 VCS REPOSITORIES

10.1 INTRODUCTION

The Git and Subversion repositories on your server are configured for HTTPS access¹, with a choice of one of three authentication methods:

- 1 Anonymous (for both read and write). No authentication is required in this case, and anyone who can connect to your server will be able to both read from, and write to, your repos. This method will therefore generally only be suitable for servers which are not connected to the internet
- 2 HTTP Basic. This is the traditional access method, and requires a password file on the server (`/data/etc/vcs-auth.htpasswd`). This file is created and maintained automatically from the console. Passwords in this file are hashed with [APR1/MD5](#)
- 3 OAuth 2.0 and OpenID Connect (OIDC)

The selected authentication method can be set independently for Git and Subversion, and will apply to *all* of the relevant (Git or Subversion) repos on your server. You cannot select different methods for different repos. However, it should be noted that, at the time of writing, we are not aware of any Subversion clients which support OAuth 2.0 authentication.

After initial configuration of your server both Git and Subversion will be set for HTTP Basic authentication. The password file is initially empty, and any attempt to access a repository will therefore fail.

Both Git and Subversion can be configured from the console, at [Others > VCS](#). This page also allows you to create HTTP Basic users if necessary. Changes to this page will generally be followed by an Apache restart, which could disrupt operations for a minute or so.

Note also that:

- 1 Anonymous and HTTP Basic authentication are handled automatically by Git, and do not require you to modify your normal `.gitconfig` configuration file
- 2 If you intend to use OAuth 2.0 authentication, you will have to install a [Credential Helper](#) on your client, and add a credentials section to your `.gitconfig` file. This procedure is described below. You can, if necessary, find the configuration documentation at:
https://git-scm.com/docs/git-config#_configuration_file
<https://git-scm.com/docs/gitcredentials>
- 3 An OAuth 2.0 configuration file contains a 'client secret' which is specific to your server. This can be changed if necessary (see [Keycloak > Regenerate secrets](#)). You should do this if you think that the secret has been compromised. If you change the secret, you should download the sample configuration file again (since it will contain the new secret), replace the secret in your own configuration file, carry out a Git push or fetch operation to allow your credential helper to record the secret, and then delete the downloaded file.

¹ ssh access is not supported.

10.2 OAUTH 2.0 OPERATION

The Apache web server acts as an OAuth 2.0 *Resource Server*. If a CLI client has a valid Access Token the Resource Server will allow both read and write access (for Git, these are fetch and push operations). If the client does *not* have an access token, Apache will automatically connect to an OpenID Connect *Identity Provider* (IdP). The IdP will then attempt to authorize the client; if successful, it returns an Access Token, and the Resource Server grants access to the repository, as well as returning the Access Token to the client for future use.

Apache is configured to connect to a single IdP, which is the Keycloak instance running on your server². Keycloak maintains a database of Server123 users, together with their associated claims. The Keycloak database is used both for front-end web logins, and for command-line access for CLI clients. When your CLI client connects without a valid access token, a browser window will pop up requesting a login:

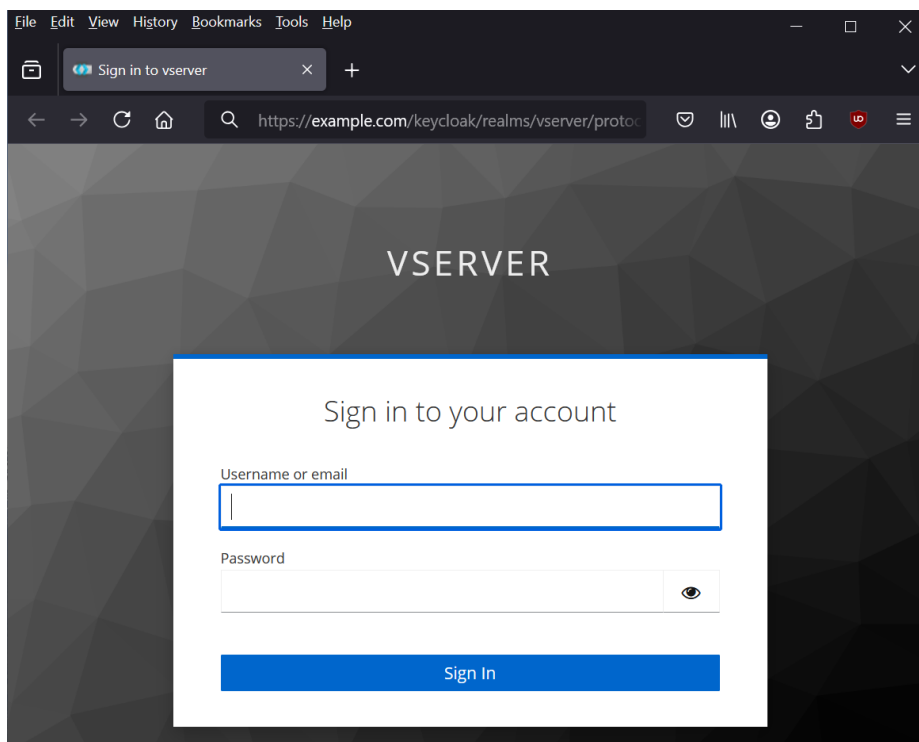


Figure 8: CLI login screen

The user will also be asked to enter a TOTP (a One Time Password) if they are configured for TOTP. If the login is successful, the user will be asked to close the new browser window, and an Access Token will be generated and returned to the client. The client will then carry out the requested operation using this token.

The sections below give practical examples for setting up OAuth 2.0 Git clients for both Linux and Windows. For both cases, you must first create one or more users who will have access to the repositories, and assign them `vcs` claims. This procedure is described in Section 10.3 below.

² It is common to allow a choice of IdPs, so that the user can log in with other accounts (Google, Microsoft, Facebook, and so on). Server123 defaults to a single IdP for simplicity.

10.3 CREATING OAUTH 2.0 CLI USERS

CLI users are created in the same way as any other Server123 users. Select [Keycloak > User accounts](#) from the console, and [Create account](#) (see Figure 1). Fill in the user details, select [Enabled](#) to enable the account, and select [2FA](#), if required, to enable TOTP. If the user only requires access to the Git and Subversion repositories, you should then select *only* the [vcs](#) claim. This claim grants the user command-line access to the repositories from CLI clients.

Instructions for setting up Git clients are given for Windows and Linux in the sections below.

10.4 REPOSITORY BROWSING

Users with a [vcs](#) claim can also use a web browser to browse any repositories on the server (this is a read-only operation). The Git repos can be found at <https://example.com/gitweb>, while the Subversion repos can be found at <https://example.com/svnweb>.

The server comes with simple preconfigured test repositories for both Git and Subversion. Both are named [test](#) (the Git repo is actually named [test.git](#) by convention, since it is a bare repository; however, for most purposes, it can simply be referred to as [test](#)). These can be browsed at the URLs above.

It should be noted that a user who has *only* a [vcs](#) claim can only do two things:

- 1 Browse the repositories
- 2 Carry out CLI operations using a client program such as git or svn (or a GUI tool such as TortoiseSVN, which relies on CLI access)

10.5 OAUTH 2.0 GIT CLIENT CONFIGURATION, LINUX

Git requires a credential helper to supply usernames and passwords (or access tokens) to a repository, and to handle secure storage of passwords. The helper used in this example is [git-credential-oauth](#) (GCO), which is included in many Linux distributions. [Git Credential Manager](#) (GCM), which is used in the Windows configuration below, can alternatively be used if necessary. However, GCM installation on Linux can be difficult, and GCO is generally preferred.

The configuration requires 3 steps:

1. Install git, if necessary
2. Install GCO
3. Create a [.gitconfig](#) file in your home directory

The commands below assume that your client is Debian or Ubuntu; replace [apt](#) with your own package management system where necessary. This procedure has been tested on Ubuntu 22.04.5 (git 2.34.1 with GCO 0.11.0) and 24.04.1 (git 2.43.0 and GCO 0.13.4).

Install git and GCO on your client as follows:

```
# apt install git
```

GCO can be installed directly for many distributions ([apt install git-credential-oauth](#) or your system's equivalent; see the GCO README for the supported list). However, you can instead install the latest GCO version if necessary, using one of the methods below:

1. Binaries are available from <https://github.com/hickford/git-credential-oauth/releases>. You will need to install a binary if your package management system is not supported
2. The GCO author maintains a PPA (Personal Package Archive) for Ubuntu users.

The PPA version can be installed as follows:

```
# add-apt-repository ppa:hickford/git-credential-oauth
# apt update
# apt install git-credential-oauth
```

You should now run `git credential-oauth` to confirm that both git and GCO have been installed (note the space character; this is a standard git command).

A basic `.gitconfig` which is sufficient for OAuth 2.0 configuration can be downloaded directly from the console. To do this, select **Administration > File download > .gitconfig**. Due to browser limitations, this may or may not be saved with a leading dot character on your computer; you will have to rename `gitconfig` to `.gitconfig` if it has been omitted. Make sure that this file is saved in your home directory.

The `.gitconfig` that you have downloaded has already been customised for your server, and will be similar to:

```
[credential "https://example.com"]
  helper          = cache --timeout 7200
  helper          = oauth
  oauthClientId    = openid-cli
  oauthScopes      = openid email
  oauthAuthURL     = /keycloak/realms/vserver/protocol/openid-connect/auth
  oauthTokenURL    = /keycloak/realms/vserver/protocol/openid-connect/token
  oauthRedirectUri = http://127.0.0.1
  oauthClientSecret = ...
```

This is sufficient for git operation, but you should, at a minimum, add your own username and email address:

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

You should now proceed to 'Testing Git' below to confirm that the configuration has been correctly set.

10.6 OAUTH 2.0 GIT CLIENT CONFIGURATION, WINDOWS

This section details one possible way to set up a Git client on Windows, using [Git for Windows](#) with the [Git Credential Manager](#) credential helper. Other options are available, but will follow the basic procedure outlined here.

Start by installing the current 'Git for Windows' from <https://gitforwindows.org/>. When you are asked to choose a credential helper, select Git Credential Manager (this is currently the default).

You should now download your sample `.gitconfig` file from the console (at **Administration > File download > .gitconfig**). Due to browser limitations, this may or may not be saved with a leading dot

character on your computer; you will have to rename `gitconfig` to `.gitconfig` if it has been omitted.

The downloaded file is shown in 10.5 above, and uses variable names which are specific to GCO. These will have to be changed for Git Credential Manager. The corrected file is shown below:

```
[credential "https://example.com"]
  helper          = manager
  provider        = generic
  oauthClientId   = openid-cli
  oauthScopes     = openid email
  oauthAuthorizeEndpoint = /keycloak/realms/vserver/protocol/openid-connect/auth
  oauthTokenEndpoint = /keycloak/realms/vserver/protocol/openid-connect/token
  oauthRedirectUri = http://127.0.0.1
  oauthClientSecret = ...
```

In other words, you must:

- 1 Replace the two helpers with a single 'manager' helper
- 2 Add a 'generic' provider line
- 3 Rename `oauthAuthURL` to `oauthAuthorizeEndpoint`
- 4 Rename `oauthTokenURL` to `oauthTokenEndpoint`

Note that the `oauthClientSecret` line does not need to be modified. The modified `.gitconfig` file must now be placed in your home directory. If you do not know where this directory is, open the 'Git bash' program that was installed with Git for Windows, and run `echo $HOME`. You can alternatively run the 'Git CMD' program, and run `echo %HOME%`. You must ensure that the configuration file is named `.gitconfig`, and not `gitconfig`.

This is a minimal configuration file. You should additionally, at least, add your own name and email address. To do this, open 'Git Bash', and run:

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

You should now proceed to 'Testing Git' below to confirm that the configuration has been correctly set.

10.7 TESTING GIT

After any configuration changes you should confirm that you can both fetch from, and push to, a remote server. The Git test repository can be used for this purpose. This repo contains a single simple C++ program.

10.7.1 GIT FETCH

To fetch from the repo, clone the project. If you have enabled OAuth 2.0, you will be asked to sign in during the clone operation (see 10.7.2 below):

```
$ mkdir vcs-git; cd vcs-git
$ git clone https://example.com/git/test
Cloning into 'test'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
$ cd test
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working tree clean
```

10.7.2 OAUTH 2.0 SIGN IN

If you have enabled OAuth 2.0 you will be asked to sign in during the clone operation. If this is the first time that you have signed in with this username, you will be asked to:

- 1 Scan a QR code to enable TOTP (if 2FA was selected for this user)
- 2 Change your password
- 3 Confirm your email address (note that the email address can be used for signing in)

However, if you have previously signed in, a browser window should be opened, to allow you to complete the sign in process (see Figure 8 above). On success, the repo will be cloned, as shown above.

Your authorisation will persist for some period of time, which will depend on your exact setup. However, it should be for some hours, at least. On expiry you will be asked to log in again when you attempt to fetch or push.

10.7.3 GIT PUSH

You will first need to modify the source code in `src/hello.cc` (by adding a comment, for example). You should now commit the change to your local copy of the repository, and then push it to the remote:

```
$ git status -s
M src/hello.cc
$ git commit -a -m "Added comment"
...
$ git push origin master
To https://example.com/git/test
  5722b9e..0aca6c3  master -> master
```

If you have enabled OAuth 2.0, you should *not* be asked to sign in again; your previous authorisation will still be valid. You can now clone the project again to a different directory, or use the repo browser, to confirm that your changes have been recorded in the remote repo.

10.8 OTHER OPERATIONS

Repositories can be created or renamed from the console, at [Others > VCS](#), and selecting the required operation.

10.8.1 DELETE

Repositories cannot be deleted from the console, because of the potential for error. Should you wish to do this, you should ssh to the server, `su` to root, and then manually delete the appropriate directory. The repos can be found at `/data/var/repositories/git` and `/data/var/repositories/svn`. The git test repo, for example, is named `/data/var/repositories/git/test.git`; removing this directory will delete the repository.

10.8.2 RENAME

Repositories are renamed simply by renaming the relevant directory, if possible (in other words, if the new directory does not already exist, and can be created).

For Subversion, the directory rename should be sufficient; see the [Red Book](#) for potential corner cases. Note that it is not necessary to regenerate the UUID.

Git directories on the server always have a `.git` suffix in their name, which is automatically added if the name supplied on the console form does not already include it. If you request a name change from 'old' to 'new', for example, the repo directory will be moved from `old.git` to `new.git`. You will need to inform your client that the repo URL has now changed. You can do this as follows:

```
$ git remote set-url <name> https://example.com/git/new
```

Where `<name>` is your repo shortname. If necessary, you can find the fetch and push shortnames by running `git remote -v`. The old URL may also be hard-wired into your `.gitconfig`. If so, it must be changed wherever it occurs.

10.8.3 CREATE

Subversion repositories can be created with a 'plain' layout, or the conventional 'trunk, branches, tags' layout (see the [Red Book](#) for details if necessary). After creating the repository, you should check out the empty repo, and then add and commit any required files for the initial import.

Git repositories are created as *bare* repositories, with `git init --bare`. If necessary, `.git` is added to the supplied name, so that the new directory always has a `.git` suffix. Bare repositories serve as an authoritative focal point for collaborative development: they handle only `fetch` and `push` operations, and are intended for use as remote central repositories. They don't have a workspace, and so can't be used for local development on the server.

11 FILE SHARE

File sharing is implemented through WebDAV, and files will normally be accessed through a file manager program (such as Windows File Explorer). The details will vary, but you will need to map a network drive, or carry out an equivalent procedure, to access the files. The shared file location on the server is `/data/var/webdav/shared` (the files will therefore be encrypted if encryption was selected during configuration), but this location is accessed externally as `/files`.

Files can be accessed with either HTTP Basic or OIDC authentication. The authentication method, and any required HTTP Basic accounts, are set from the console, at `Others > File share`:

File share [?](#)

WebDAV

Authentication

HTTP Basic ☒

OpenID Connect (OIDC) ☐

HTTP Basic auth

Operation Create account

First name Lizzie

Surname Bennet

Username lizzie

Password

Submit Cancel

Figure 9: WebDAV setup

HTTP Basic is the traditional access method, and requires a password file on the server (`/data/etc/dav-auth.htpasswd`). This file is created and maintained automatically from the form above. Passwords in this file are hashed with [APR1/MD5](#).

If OIDC is selected, an OIDC sign in is instead required (see 2.2 above), with a `Files` claim. However, as noted in (2.2), few (if any) file transfer programs currently support generic OIDC sign in.

The mechanism used to access your files will depend on your file manager program. Two examples are given below, but you should check the details for your own manager. You may also find additional information in the online help.

WINDOWS FILE EXPLORER

Open the File Explorer application, and select `This PC` from the left-hand panel. Now select the `Computer` tab in the top navigation bar, and then click on `Map Network Drive`. For the folder to share, enter `https://example.com/files`.

When you click `Finish`, a 'Windows Security' popup will appear, asking for a username and password. Enter the username and password of your HTTP Basic account. An explorer window should now

appear, showing the contents of the shared folder. You can use the normal drag and drop procedure to write files to the shared folder, to retrieve files from it, to create subdirectories, and so on.

LINUX GNOME

Open the Files application (aka 'Nautilus') by clicking on the folder icon, or selecting [Places > Computer](#) from the top bar, or equivalent. Now click on [+ Other Locations](#) in the left-hand panel. This will show a network view, and allow you to connect to a server:

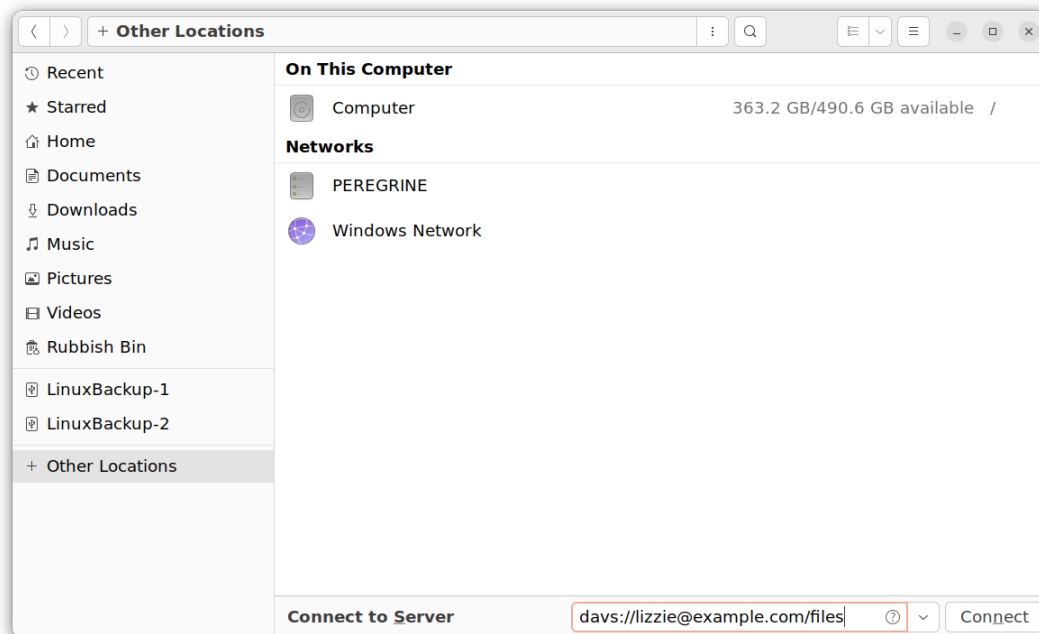


Figure 10: GNOME files mapping

In the 'Connect to Server' field, enter `davs://username@example.com/files`, where 'username' is your HTTP Basic username (in Figure 9 above, this is `lizzie`). An authorization box now pops up. Confusingly, this asks you to `Enter password for "WebDAV"`, rather than entering the password for the username. Enter your password, and a new Files window will appear, showing your shared files.

12 REDMINE

You must carry out an OIDC sign-in (3.2.10) before accessing the Redmine endpoint at `/redmine`. A single Redmine user (an administrator) was created during site configuration, with these details:

Username	Site Administrator first name
Full name	Site Administrator full name
Email	Site Administrator email
Password	Site Administrator Password 2

You will be asked to change this password when you first log in. The database is at `/data/var/lib/mysql/redmine`, and so will be encrypted if the data partition is encrypted.

12.1 API ACCESS

The Redmine API can be accessed locally, or from your client. The REST API is not enabled by default, and must first be enabled. To do this, log into Redmine as an administrator, and enable the API from the Settings menu item ([Administration > Settings > API > Enable REST web service](#)).

For local access, you will have to ssh to the server, change to the Redmine installation directory, and start another instance of Phusion Passenger as root:

```
root@vserver# cd /opt/redmine-6.0.5
root@vserver# passenger start -e production
```

This starts a standalone instance of Passenger, with nginx listening on port 3000. The console will show the log output; you should terminate Passenger with `Ctrl-C` when you no longer need it.

You can now make API requests with cURL. This request, for example, returns a list of users:

```
sysadmin@vserver$ curl -u username:password http://127.0.0.1:3000/users
```

13 ALFRESCO

13.1 AUTHENTICATION

A single Alfresco user is created during configuration. This user is an administrator, and the user details are set from the Configuration page Site Administrator fields as follows:

User	Site Administrator first name
Password	Site Administrator Password 2

There are two authentication choices:

- 1 OIDC, which is the default. In this case, an `Alfresco` claim is required for access. When you have carried out the OIDC sign in (or if you are already signed in), you will have to log in to Alfresco with your Alfresco user name and password
- 2 Basic, which allows you to log in with *only* an Alfresco username and password

Basic Authentication is required if you either (a) intend to edit your documents in Microsoft Office, or (b) intend to access Alfresco directly from a network share (for either Windows or Linux clients). Microsoft deprecated basic authentication at the end of 2024, but it can still be used by setting a group access policy, or by setting a registry key. The *'Basic authentication sign-in prompts are blocked by default in Microsoft 365 apps'* article explains this policy, and can currently be found [here](#). You can set a group policy to allow access, as described in the article, or you can instead set a registry key, as shown below:

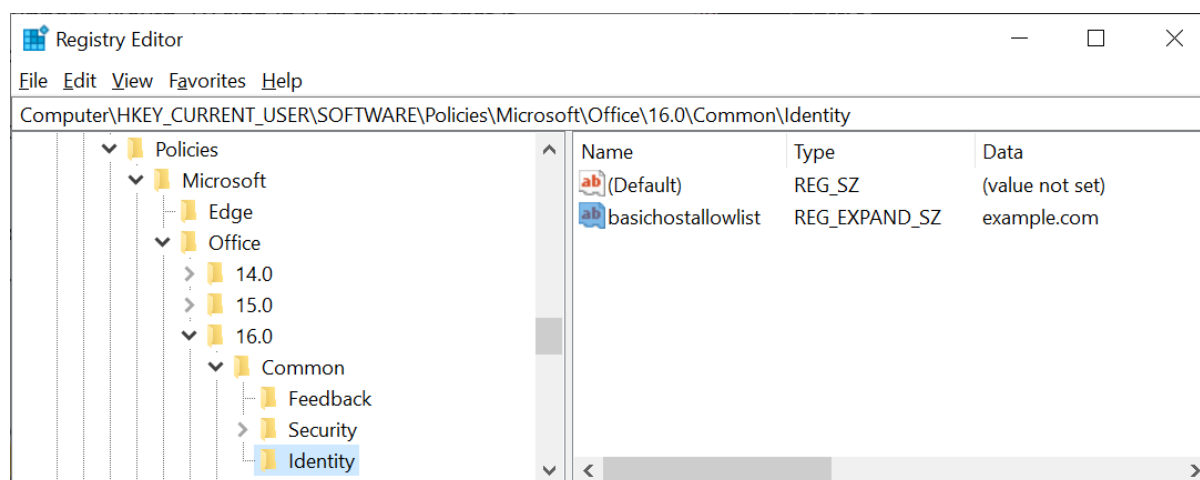


Figure 11: Registry basichostallowlist

In other words, you must add key `HKEY_CURRENT_USER\Software\Policies\Microsoft\Office\16.0\Common\Identity`, with the new entry named `basichostallowlist`, with type `REG_EXPAND_SZ`, and your own domain name as the data.

Alfresco does not run by default, and must be started (or stopped) from the console, at [Others > Alfresco](#). This form allows you to set the authentication method, and the required heap size settings (see below). The auth method and heap settings are acted on just before starting Alfresco, or just after stopping it; they cannot be changed while Alfresco is running.

Alfresco may take a minute or more to start, depending on your hardware. On completion, you can browse to the `/alfresco` and `/share` endpoints.

13.2 HEAP SETTINGS

When started, Alfresco runs four JVM instances: ACS ('Alfresco Content Services'), the Transform service, the Apache ActiveMQ message broker, and the Apache Solr search platform. The [Others > Alfresco](#) form allows you to set the heap values for these services, where 'Xms' gives the initial heap allocation, and 'Xmx' gives the maximum heap allocation. These two are normally set to the same value for servers. See the online help for more information, or the Oracle docs [here](#).

13.3 EDIT IN MICROSOFT OFFICE

If you have a Windows client, with access to Microsoft Office, *and* you have started Alfresco with Basic Authentication, then you can edit supported document types directly with MS Office from your browser. The document will have an [Edit in Microsoft Office](#) menu entry at some point; this may be to the right of the document description, under [More](#).

13.4 EDIT IN GOOGLE DOCS

It is not currently possible to edit your documents with Google Docs, following recent authentication changes at Google. If you attempt to do this, you will get a 'This app is blocked' popup from your Google sign in page.

13.5 USING ALFRESCO FROM MICROSOFT OFFICE

If you have started Alfresco with Basic Authentication then you can access your Alfresco files directly from your Microsoft Office applications. This is described in more detail in the Alfresco documentation at <https://docs.alfresco.com/microsoft-office/latest/using/>.

The file location you should use to open your content when running an Office application is <https://example.com/alfresco/aos>. You can browse down from this location to the required file, or you can enter a complete path name if it is known. The [alfresco/aos/](#) directory can also be accessed as [alfresco/webdav/](#).

Files can also be opened directly from Windows Explorer, although potentially with a slightly different URL. The sample web site design project, for example, contains a directory which contains meeting notes, at

<https://example.com/alfresco/aos/Sites/swsdp/documentLibrary/Meeting Notes>

You should also be able to use this URL directly in Windows Explorer but, if not, the conventional mapping for Explorer would be

[\\example.com@SSL\alfresco\aos\Sites\swsdp\documentLibrary\Meeting Notes](#)

The files shown can now be double-clicked to open them in Office.

13.6 NETWORK SHARE

Alfresco files can be accessed directly by mapping a network drive. Instructions are given in 11 above for Windows Explorer and Gnome Files. However, in this case no HTTP Basic account is needed, and you can use your Alfresco login directly. The share location should be set to <https://example.com/alfresco/aos> or <davs://username@example.com/alfresco/aos> for Explorer and Files, respectively, where 'username' is your Alfresco username.

14 SERVER BACKUP

This section includes recommendations for backing up your server, using either rsync or burp. Unless you write your own client-side scripts, rsync is more suited to one-off backup operations, while burp backups are rotated and entirely automated. This does, however, require you to set up a burp server on another computer.

Both rsync and burp use a single centralised file list to determine which files and directories should be backed up. You can modify this list yourself, if necessary (see 14.1 below).

Note that these backup strategies handle only the files and directories which are considered important enough to include in the file list. Their primary advantage is that they are fast (they are differential, copying only changed files) and simple, but they do *not* carry out a 'complete' backup (in other words, a backup which could be restored to a bare-metal server). However, they should be sufficient to restore a new Server123 installation to a known state.

14.1 SERVER FILES

The files which should be backed up are listed in `/data/var/www1/site/vserver.flist`, which is used for both rsync and burp. If you need to change this file (to, for example, add your own backup locations), you can download it from the console, at `Administration > File download > Backup file list`. If you modify the file you should upload it back to the server, from `Administration > File upload > Backup file list`. The uploaded file is automatically processed for burp usage.

14.2 SERVER STATE

Backing up a running server can be problematical. If the web server is running, the MySQL database is not guaranteed to be in a consistent state. Similarly, if Dovecot is running, your mailboxes may not be in a consistent state at the point at which the backup program reads them.

To handle this situation, you should log in to your server, stop these services before the backup, and then restart them after the backup. **This is carried out automatically in the procedures recommended in (14.3) and (14.4).** However, if necessary, you can do this yourself, by running the server's `backup-prepost.sh` script. This has one parameter: `pre` stops any relevant services before a backup, and `post` restarts those services. When run with `pre` the script waits (currently for 80 seconds) before returning. Note that you must first ssh to your server as user sysadmin, and that your site will show a 'Site down for maintenance' page until the services are restarted:

```
# Graceful Apache restart with a site down page, then stop MySQL and Dovecot:
sysadmin@vserver $ sudo backup-prepost.sh pre
# ...carry out the backup here
# and now start services, and restore the site
sysadmin@vserver $ sudo backup-prepost.sh post
```

If you are instead scripting this from your client, and you have set up root's public key (see 0 above), you can run the script directly from your own Linux client, as follows:

```
root@my-client # echo "backup-prepost.sh pre" | ssh -p$port $remote /bin/bash
```

In this case, `$port` is your ssh port, while `$remote` is your server. This invocation of ssh connects to the remote server, runs a Bash shell, and executes any commands found on stdin.

14.3 RSYNC

rsync is a differential file copy tool. When connecting to a remote system, it can use a remote shell program (in this case, ssh) as the transport, or it can connect to a remote rsync daemon, via TCP. The ssh option encrypts the data transfer, and is fast; the daemon/TCP option is not encrypted, is slow, and requires a port (normally 873) to be opened. However, the remote daemon option does allow the `backup-prepost.sh` script to be run automatically.

Fortunately, it is possible to combine the advantages of ssh transport, and to connect to the daemon (`rsyncd`), to run the required pre- and post- scripts. Both the commands shown below do this. `rsyncd` requires a configuration file, which is `/etc/rsyncd.conf`. This includes the definition of the src 'module', and the commands which are to be run before starting the rsync operation, and after completing it (`pre-xfer exec` and `post-xfer exec`). Note that `rsyncd` is *not* run as a daemon in this usage: it is simply started on demand when the rsync request is received.

There are many ways to use rsync. Two are given below, and have been tested. **Note that this usage requires your public key to be uploaded for user root.** In other words, it must be possible to ssh to the server as root, without supplying a password (see 0 above).

```
root@my-client # rsync -avriXPz -e "ssh -p$port" --numeric-ids --del \
                  --files-from=vserver.flist \
                  example.com::src/ \
                  /home/backups/example.com
```

In this usage:

- 1 `$port` is your server's ssh port (see 0 above)
- 2 `vserver.flist` is the locally-downloaded file list (see 14.1 above). It is not necessary to download the file list (see below), but you may wish to use a local list that you can modify yourself
- 3 The `--del` option deletes files on the receiving side which no longer exist on the server (sender). In some cases, you may wish to omit this option
- 4 `example.com` is your server; `src` is the required `rsyncd` module (in this case, it is just `/`). The double-colon enforces `rsyncd` usage, while the `ssh` enforces ssh transport
- 5 The final line is the location at which the copied files will be stored on your client. This directory will be loaded with your server's file structure, starting at the server root
- 6 You must run this as root to be able to read the required files

You can simplify this operation slightly by using the file list stored on the remote server itself, rather than downloading it first. In this case the usage is:

```
root@my-client # rsync -avriXPz -e "ssh -p$port" --numeric-ids --del \
                  --files-from=/data/var/www1/site/vserver.flist \
                  example.com::src/ \
                  /home/backups/example.com
```

These commands display (itemise) any changes carried out (`-i`). The itemisation output is complex, and difficult to understand. However, a `>` character appearing in the first column denotes that a file was received by your client because it had changed on the server. See [this blog](#) for the details. In most cases, a differential backup to an existing location will only take a minute or two to complete.

14.4 BURP

Burp ('BackUp and Restore Program') is a network backup utility. **Server123 is the burp client; the burp server is the computer where your backups will be stored.** The server must run on a Unix-based system. You can find the Burp documentation at <https://burp.grke.org/docs.html>.

To run burp, you will need to install the server on another computer, which can be identified with a domain name or an IP address. If you are backing up to a home network, you will therefore need a static IP address from your broadband provider. If this isn't possible, you may be able to get a static address from a VPN service, or a Dynamic DNS service.

Burp's behaviour is defined by `/etc/burp/burp.conf` for the client, and `/etc/burp/burp-server.conf` for the server. These can be changed if necessary, but are pre-configured as follows:

- 1 The files to be backed up are defined on the client, at `/etc/burp/clientconfdir/incexc/vserver-includes`. This file is automatically modified if you upload a file list (see 14.1 above)
- 2 Files are backed up daily, at some time between 4 and 5AM. 7 daily backups are retained, together with 4 weekly backups, and 6 multiples of 4 weeks, giving about 6 months of backups. This guarantees the ability to restore to any day within the last 168 days
- 3 The backup location is `/var/spool/burp` on the server
- 4 The server and client identify each other with SSL certificates; the server signs the client certificate when the client first connects. A password is also used, and is entered in plaintext in the two configuration files. This password was created randomly during site configuration, but you can change it if necessary (make sure you change *both* configuration files)
- 5 Burp's encryption is not enabled by default, which allows the use of delta differencing. This means that the server does not need to request an entire client file when the file changes. This also means that the backups stored on the server will not be encrypted, even if you enabled LUKS encryption during site configuration. This is because LUKS is an 'at rest' encryption scheme (see section 7 in the Installation Guide for details).

The burp client (on Server123) runs a cron job every 20 minutes (see `/var/spool/cron/crontabs/root`). This checks if the client is enabled; if so, it runs `burp -a t`, which contacts the server. The server determines whether a backup is required and, if so, initiates it. The pre- and post-scripts are run at this point (14.2), so the site will be briefly unavailable during the backup.

If you are running burp, you should occasionally check in with the backup server, to view a list of backups. To do this, ssh to your own server, and run `burp -a l` as root. If this returns only a cname message (cname from cert: `vserver.example.com`) this means that the backup server cannot be contacted, or is not running a burp server, or that there are no backups to report.

14.5 ENABLING BACKUPS

14.5.1 SERVER

Burp must be installed on your backup server. The package name may depend on your system. For Debian-based systems, for example, you will need to `sudo apt install burp`. When you have installed burp, you must copy two files from the Server123 installation to the same locations on your burp server:

- `/etc/burp/burp-server.conf`
- `/etc/burp/clientconfdir/vserver.example.com`, where `example.com` is your own domain name

Two ports must be opened. These are, by default, 7902 and 7903. This code is specific to Debian-type systems, and should be changed for your own distribution:

```
server# ufw allow 7902/tcp
server# ufw allow 7903/tcp
server# burp -c /etc/burp/burp-server.conf
server# netstat -lp | grep -i burp
tcp        0      0 localhost:7903          0.0.0.0:*               LISTEN      27434/burp
tcp        0      0 0.0.0.0:7902            0.0.0.0:*               LISTEN      27434/burp
```

Note that:

- 1 The server is run directly in this example, with `burp -c`. In normal use, a systemd unit file, or equivalent, should be created to ensure that burp is started whenever the system boots
- 2 The default ports can be changed, but you must modify both the server and client configurations (`/etc/burp/burp.conf` and `/etc/burp/burp-server.conf`)
- 3 Netstat is run to confirm that burp is listening on ports 7902 and 7903

At this point the server is waiting for a connection from `example.com`, on either of these ports.

14.5.2 CLIENT

The client configuration file (`burp.conf`) must be modified as follows:

- 1 Change `enabled=0` to `enabled=1`
- 2 The server is identified from the `server = example.com:7902` directive. Change `example.com` to the name or IP address of your burp server

Backups are now enabled, but won't commence until sometime between 4 and 5AM. To confirm that the connection is working, and that the certificate exchange completes, you should therefore run `burp -a 1` on the client (as root). See <https://burp.grke.org/docs/add-remove.html> for more details.