TeamDrive

Sync your data fast & securely

TeamDrive Web Portal Administration

Release 1.0.2.0

Lenz Grimmer, Paul McCullagh

2015

CONTENTS

CHAPTER

ONE

COPYRIGHT NOTICE

TeamDrive Systems GmbH
https://www.teamdrive.com
Max-Brauer-Allee 50
22765 Hamburg, Germany
Email: info@teamdrive.com

CHAPTER TWO

TRADEMARK NOTICE

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Apache HTTP Server, Apache, and the Apache feather logo are trademarks of The Apache Software Foundation.

MySQL is a registered trademark of Oracle and/or its affiliates.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices.

VMware is a trademark or registered trademark of VMware, Inc. in the United States and/or other jurisdictions.

"Amazon Web Services", "Amazon S3" are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

"Red Hat Linux" and "CentOS" are trademarks of Red Hat, Inc. in the U.S. and other countries.

All other names and trademarks used herein are the property of their respective owners.

CHAPTER

THREE

INTRODUCTION

This document will guide you through the administration and advanced configuration of a TeamDrive Web Portal.

When managing the TeamDrive Web Portal, we assume that you have basic knowledge of:

- **Linux system administration:**
    - Adding/configuring software packages
    - Editing configurations files
    - Creating user accounts
    - Assigning file ownerships and privileges
    - Creating and mounting file systems
    - Setting up environment variables
- Apache web server: installation and configuration, adding and enabling modules, modifying configuration files
- MySQL Database: installation and configuration, administration/maintenance, using the MySQL command line client, basic SQL
- Basic knowledge of application server technology

CHAPTER

FOUR

TEAMDRIVE WEB PORTAL ADMINISTRATION

4.1 Disabling the Apache Access Log

In the default setup, Apache is used as a reverse proxy to route all calls from the TeamDrive browser App to the Docker containers. This can generate a large number of requests so there is no point in keeping the normal access log activated. We therefore recommend deactivating it in a production environment. Only the error log should be left enabled. To facilitate this, comment out the following line in the default `httpd.conf`:

```
# CustomLog logs/access_log combined
```

If problems occur, logging can be activated for a specific user (see http://httpd.apache.org/docs/2.2/mod/mod_log_config.html). e.g. all access to TeamDrive Agent using port 49153 will be logged (the required Apache logging module needs to be enabled again):

```
SetEnvIf Request_URI 49153 agent-49153
CustomLog logs/agent-49153-requests.log common env=agent-49153
```

Restart the Apache instance and check the log files for errors.

You can discover the port used by an agent by using the command:

```
[root@webportal ~]# docker ps -a | grep <username>
```

The port used will be in the 6th column of the output which has the form: `0.0.0.0:<agent-port>->4040/tcp`, e.g. `0.0.0.0:49153->4040/tcp`.

4.2 Changing an Admin User's Password

The Web Portal Administration Console can be accessed by all Admin Users by entering the correct username and password.

An existing user with administrative privileges can change his password directly via the Administration Console's login page or via the **Admin Users** page of the Administration Console.

On the login page, click on **Change Password...** to enable two input fields **New Password** and **Repeat Password** that allow you to enter the new password twice (to ensure you did not mistype it by accident). You also need to enter your username in the **Username** field and the current password in the **Password:** field above. Click **Login and Change Password** to apply the new password and log in.

You can also change your password while being logged into the Administration Console. If your user account has "Superuser" privileges, you can change the password of any admin user, not just your own one.

Click **User List** to open the user administration page.

The page will list all existing user accounts and their details.

Click the username of the account you want to modify. This will bring up the user's details page.

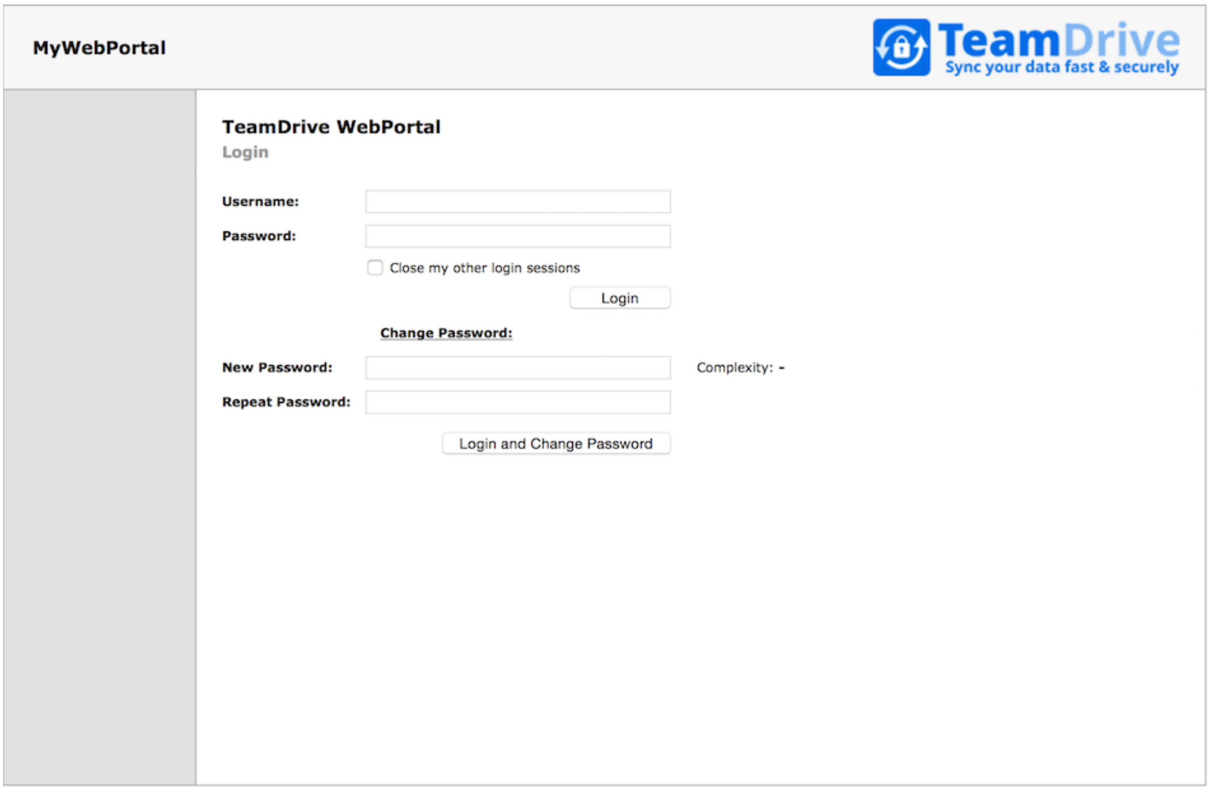
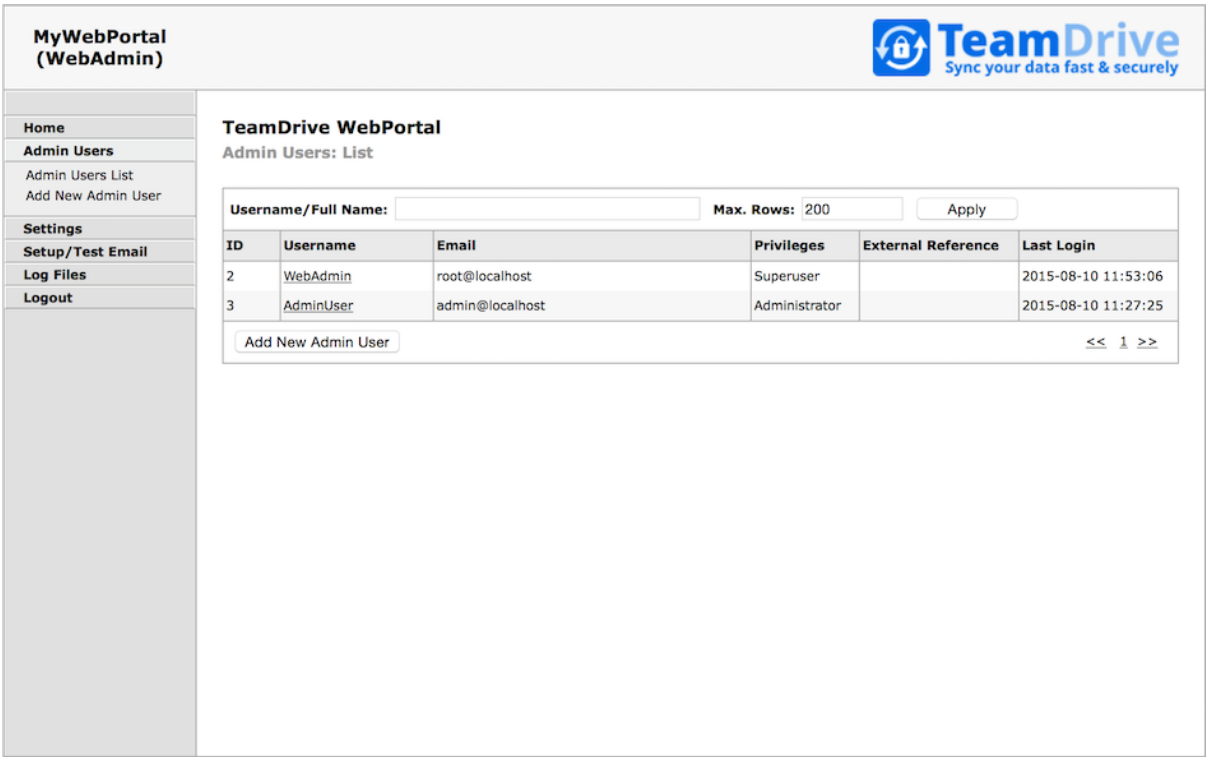


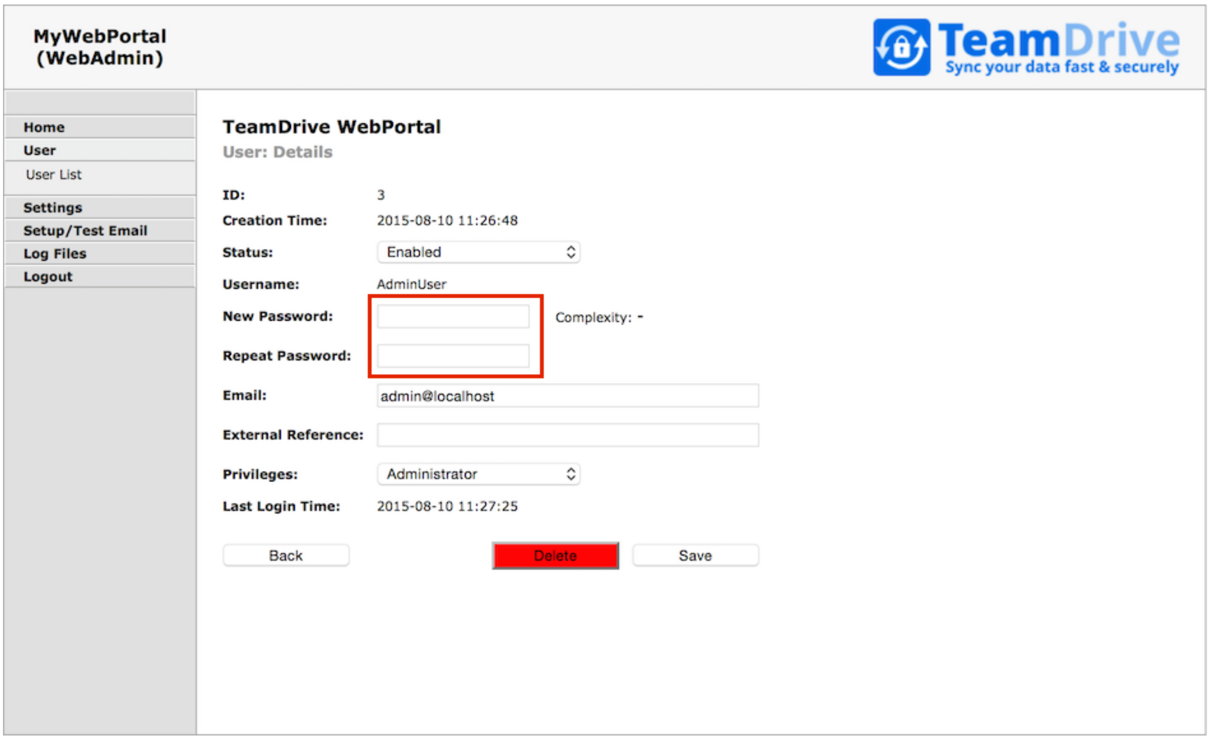
Fig. 4.1: Web Portal Administration Console: Change Password



Max. Rows: 200

ID	Username	Email	Privileges	External Reference	Last Login
2	WebAdmin	root@localhost	Superuser		2015-08-10 11:53:06
3	AdminUser	admin@localhost	Administrator		2015-08-10 11:27:25

Fig. 4.2: Web Portal Administration Console: Admin Users List



TeamDrive WebPortal

User: Details

ID: 3

Creation Time: 2015-08-10 11:26:48

Status: Enabled

Username: AdminUser

New Password:

Repeat Password:

Complexity: -

Email: admin@localhost

External Reference:

Privileges: Administrator

Last Login Time: 2015-08-10 11:27:25

Fig. 4.3: Web Portal Administration Console: User Details

To change the password, enter the new password into the input fields **New Password** and **Repeat Password** and click **Save** to commit the change.

The new password will be required the next time this user logs into the Administration Console.

In case you lost or forgot the password for the last user with Superuser privileges (e.g. the default `HostAdmin` user), you need to reset the password by removing the current hashed password stored in the MySQL Database (Column `Password`, located in Table `webportal.WP_Admin`). This can be performed using the following SQL query.

Log into the MySQL database using the `teamdrive` user and the corresponding database password:

```
[root@webportal ~]# mysql -u teamdrive -p
Enter password:

[...]

mysql> use webportal;
Database changed

mysql> SELECT * FROM WP_Admin WHERE UserName='WebAdmin'\G
*************************** 1. row ***************************
           ID: 1
       Status: 0
     UserName: WebAdmin
        Email: root@localhost
     Password: $2y$10$JIhziNetygYCeIXU3gXveue2BTqwCs4vwA6LHNUKZVt8V.U8jtkcW
  ExtReference: NULL
    Privileges: Superuser
  CreationTime: 2015-08-10 11:26:10
LastLoginTime: 2015-08-10 11:53:06
1 row in set (0.00 sec)

mysql> UPDATE WP_Admin SET Password='' WHERE UserName='HostAdmin';
Query OK, 1 row affected (0.01 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0

mysql> quit
Bye
```

Now you can enter a new password for the `HostAdmin` user via the login page as outlined above, by clicking the **Change Password** link, but leaving the **Password** field empty and only entering the new password twice, followed by clicking the **Login and Change Password** button.

4.3 Enabling Two-Factor Authentication for Superusers

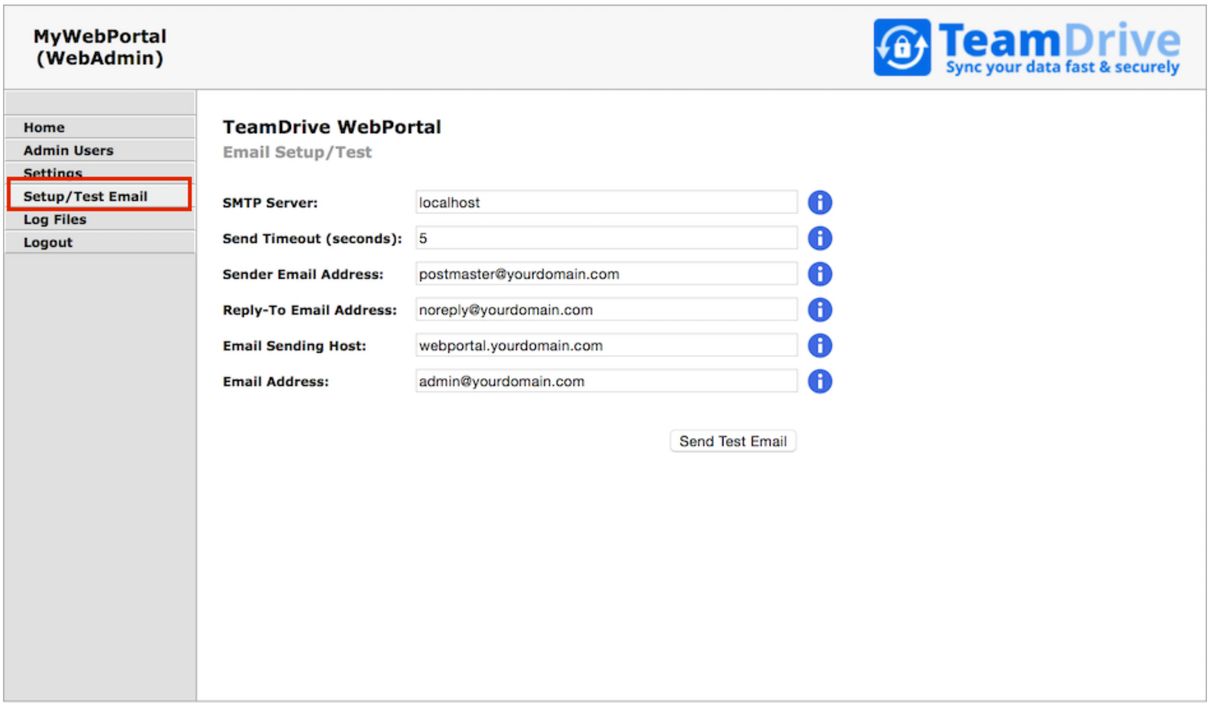
The Web Portal Administration Console supports two-factor authentication via email. In this mode, an Admin User with “Superuser” privileges that wants to log in with his user name and password needs to provide an additional authentication code that will be sent to him via email during the login process. This feature is disabled by default.

The TeamDrive Web Portal needs to be configured to send out these authentication email messages via SMTP. The Web Portal is only capable of sending out email using plain SMTP via TCP port 25 to a local or remote MTA.

If your remote MTA requires some form of encryption or authentication, you need to set up a local MTA that acts as a relay. See chapter *Installing the Postfix MTA* in the *TeamDrive Web Portal Installation Guide* for details.

Before you can enable two-factor authentication, you need to set up and verify the Web Portal’s email configuration. This can be accomplished via the Host Server’s Administration Console. You need to log in with a user account having “Superuser” privileges in order to conclude this step.

Click **Setup / Test Email** to open the server’s email configuration page.



TeamDrive WebPortal

Email Setup/Test

SMTP Server: localhost

Send Timeout (seconds): 5

Sender Email Address: postmaster@yourdomain.com

Reply-To Email Address: noreply@yourdomain.com

Email Sending Host: webportal.yourdomain.com

Email Address: admin@yourdomain.com

Fig. 4.4: Web Portal Admin Console: Email Setup / Test

Fill out the fields to match your local environment:

SMTP Server: The host name of the SMTP server accepting outgoing email via plain SMTP. Choose `localhost` if you have set up a local relay server.

Send Timout: The timeout (in seconds) that the mail sending code should wait for a delivery confirmation from the remote MTA.

Sender Email Address: The email address used as the Sender email address during the SMTP delivery, e.g. `postmaster@yourdomain.com`. This address is also known as the "envelope address" and must be a valid email address that can accept SMTP-related messages (e.g. bounce messages).

Reply-To Email Address: The email address used as the "From:" header in outgoing email messages. Depending on your requirements, this can simply be a "noreply" address, or an email address for your ticket system, e.g. `support@yourdomain.com`.

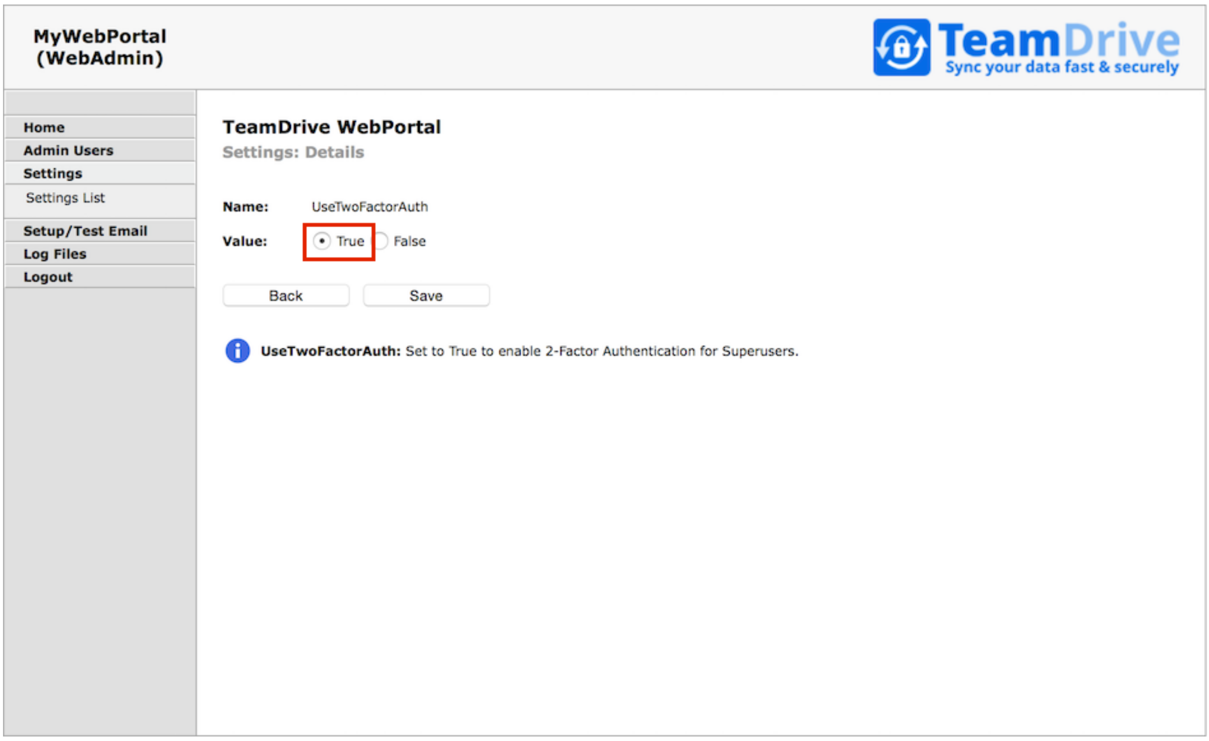
Email Sending Host: The host name used in the HELO SMTP command, usually your Web Portal's fully qualified domain name.

Email Address: The primary administrator's email address. This address is the default recipient for all emails that don't have an explicit receiving address. During the email setup process, a confirmation email will be sent to this address.

After you've entered the appropriate values, click **Send Test Email** to verify the email setup. If there is any communication error with the configured MTA, an error message will be printed. Check your configuration and the MTA's log files (e.g. `/var/log/maillog` of the local Postfix instance) for hints.

If the configuration is correct and functional, a confirmation email will be delivered to the email address you provided. It contains an URL that you need to click in order to commit your configuration changes. After clicking the URL, you will see a web page that confirms your changes.

This concludes the basic email configuration of the Web Portal. Now you can enable the two-factor authentication by clicking **Settings** -> **UseTwoFactorAuth**. Change the setting's value from `False` to `True` and click **Save** to apply the modification.



TeamDrive WebPortal

Settings: Details

Name: UseTwoFactorAuth

Value: ● True ○ False

UseTwoFactorAuth: Set to True to enable 2-Factor Authentication for Superusers.

Fig. 4.5: Web Portal Admin Console: Use Two Factor Authentication

Now two-factor authentication for the Administration Console has been enabled.

The next time you log in as a user with "Superuser" privileges, entering the username and password will ask you to enter a random secret code, which will be sent to you via email to the email address associated with your administrator account. Enter the code provdided into the input field **Authentication Code** to conclude the login process.

4.4 Changing the MySQL Database Connection Information

The Web Portal Apache module `mod_yvva` as well as the `yvvad` daemon that performs the `td-webportal` background tasks need to be able to communicate with the MySQL management database of the Web Portal.

If you want to change the password of the `teamdrive` user or move the MySQL database to a different host, the following changes need to be performed.

To change the MySQL login credentials, edit the file `/etc/td-webportal.my.cnf`. The password for the `teamdrive` MySQL user in the `[tdweb]` option group must match the one you defined earlier:

```
[tdweb]
database=webportal
user=teamdrive
password=<password>
host=127.0.0.1
```

If the MySQL database is located on a different host, make sure to modify the `host` variable as well, providing the host name or IP address of the host that provides the MySQL service. If required, the TCP port can be changed from the default port (3306) to any other value by adding a `port=<port>` option.

4.5 Configuring Active Directory / LDAP Authentication Services

If the TeamDrive users of the Web Portal are using an external Authentication Service such as Active Directory or LDAP, then the Web Portal must also be configured to use the authentication service.

Note: This section refers to the login of the TeamDrive users as apposed to the administrators of the Web Portal, which is described in the section: *Administrator Login using External Authentication* (page 12) below.

This is done by setting `AuthServiceEnabled` to `True`, and the setting the correct values for `AuthLoginPageURL` and `AuthTokenVerifyURL`.

Please refer to **Configuring External Authentication using Microsoft Active Directory / LDAP** in the **TeamDrive Registration Server Administration Guide** for details of how to setup an External authentication service. In this document we describe only the aspects that are relevant to the Web Portal.

Once you have setup LDAP or Active Directory authentication for the Registration Server it is a simple step to enable this service for the Web Portal. The page “ldap_web_login.php” has been provided for this purpose.

You must make the appropriate changes to the page as described for The “ldap_login.php” page. In the page, the text “webportal.domain.com” must be replaced with the domain name of the Web Portal. Set `AuthLoginPageURL` to the URL of the “ldap_web_login.php” page.

This URL is now the login page for the Web Portal, and the user will be automatically directed to this page if he is not already logged in.

The `AuthTokenVerifyURL` setting must be set to the “ldap_verify.php” page provided by the Active Directory / LDAP authentication implementation for the Registration Server.

Once these parameters are set correctly, and the necessary changes have been made to “ldap_web_login.php”, login using the Active Directory / ldap service should work correctly. If an login fails, first check the `/var/log/td-webportal.log` log file for errors.

4.6 Administrator Login using External Authentication

The Administration Console of the Web Portal may use External Authentication such as LDAP or Active Directory. If the administrators of the Web Portal are stored and managed by such a service then it is possible to have the user credentials checked by the server, rather than stored and checked by the Web Portal database.

There are two system settings that control this behaviour: `ExtAuthEnabled` and `ExtAuthURL`. `ExtAuthEnabled` must be set to `True`. `ExtAuthURL` specifies a URL that will perform the external authentication.

On login, if external authentication is enabled, the Web Portal will perform a HTTP POST to the URL specified by `ExtAuthURL`, passing two parameters: `username` and `password`. The page is expected to return an XML reply of the following form:

```
<?xml version='1.0' encoding='UTF-8'?>
<teamdrive>
<user>
<id>unique-user-id</id>
<email>users-email-address</email>
</user>
</teamdrive>
```

If an error occurs, for example an "Incorrect login", then the *ExtAuthURL* page must return:

```
<?xml version='1.0' encoding='UTF-8'?>
<teamdrive>
<error>
<message>error-message-here</message>
</error>
</teamdrive>
```

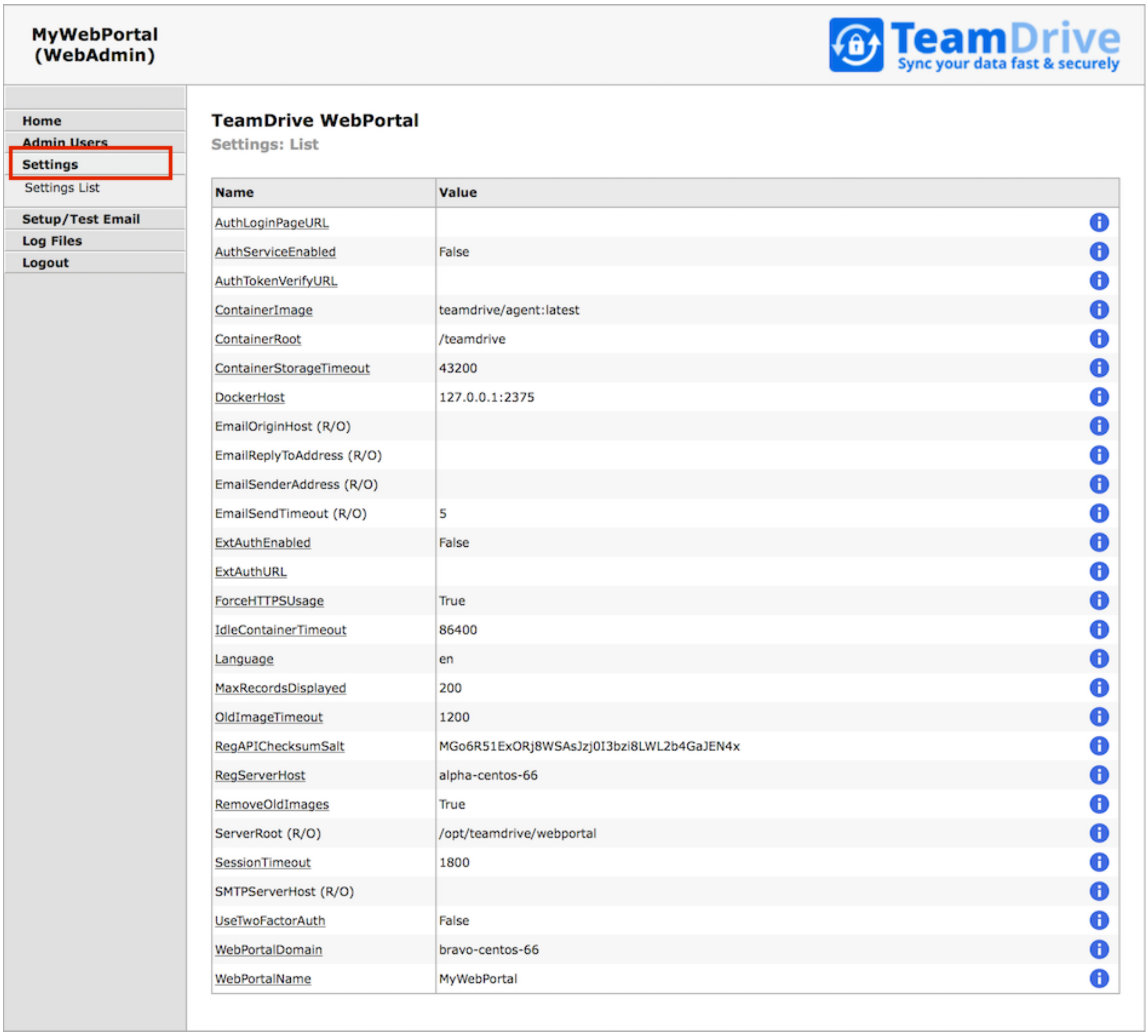
Such a page can be easily implemented in PHP, for instance. An example implementation of the `ExtAuthURL` page for LDAP and Active Directory is available upon request from TeamDrive Systems (please contact sales@teamdrive.com).

4.7 System Settings

Select the In the Settings item from the menu in order to obtain a list of all system settings available for the Web Portal.

This section describe the various settings and how that can be used to configure the Web Portal.

- **SessionTimeout**: This is the idle time in seconds after which you are required to login to the Web Portal Admin Console again.
- **WebPortalDomain**: This is the domain name (or URL) of this service.
- **WebPortalName**: This name of this service. The name is displayed in the Web Portal Admin Console. The default value is the domain name of the service The name is used for display purposes only, and may be set to any value.
- **UseTwoFactorAuth**: Set to `True` to enable 2-Factor Authentication for Superusers.
- **ServerRoot**: The installation directory of the Web Portal application.
- **ForceHTTPSUsage**: Set to `True` if the Web Portal Admin Console must be accessed using HTTPS.
- **MaxRecordsDisplayed**: This setting determines the maximum number of records that may be retrieved from the database at any a time. This parameter may only be changed by a Superuser.
- **ExtAuthEnabled**: Set this value to `True` to enable External Authentication. See *Administrator Login using External Authentication* (page 12) for details.
- **ExtAuthURL**: This is the URL that is used by the Web Portal to verify the login of an Administrator, when using External Authentication. See *Administrator Login using External Authentication* (page 12) for details.
- **AuthServiceEnabled**: Set this value to `True` to enable an Authentication Service for the TeamDrive users. This means that the users that access the Web Portal are required to login using an external login page. See *Configuring Active Directory / LDAP Authentication Services* (page 12) for details.



TeamDrive WebPortal

Settings: List

Name	Value
AuthLoginPageURL	
AuthServiceEnabled	False
AuthTokenVerifyURL	
ContainerImage	teamdrive/agent:latest
ContainerRoot	/teamdrive
ContainerStorageTimeout	43200
DockerHost	127.0.0.1:2375
EmailOriginHost (R/O)	
EmailReplyToAddress (R/O)	
EmailSenderAddress (R/O)	
EmailSendTimeout (R/O)	5
ExtAuthEnabled	False
ExtAuthURL	
ForceHTTPSUsage	True
IdleContainerTimeout	86400
Language	en
MaxRecordsDisplayed	200
OldImageTimeout	1200
RegAPIChecksumSalt	MGo6R51ExORj8WSAsJzj0I3bzi8LWL2b4GaJEN4x
RegServerHost	alpha-centos-66
RemoveOldImages	True
ServerRoot (R/O)	/opt/teamdrive/webportal
SessionTimeout	1800
SMTPServerHost (R/O)	
UseTwoFactorAuth	False
WebPortalDomain	bravo-centos-66
WebPortalName	MyWebPortal

Fig. 4.6: Web Portal Admin Console: Settings

- **AuthLoginPageURL**: This is URL of the Authentication Service login page which must be used by TeamDrive users that are registered by the external Authentication Service. See *Configuring Active Directory / LDAP Authentication Services* (page 12) for details.
- **AuthTokenVerifyURL**: This is URL is used to verify the token returned by the Authentication Service after success login by a TeamDrive user. See *Configuring Active Directory / LDAP Authentication Services* (page 12) for details.
- **SMTPServerHost**: Domain name (and port) of the SMTP server used to send emails. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **EmailSendTimeout**: Timeout in seconds, when sending an email. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **EmailSenderAddress**: The email address of the sender. This address is not directly visible to the email receiver. If an email bounces, a message will be sent to this address. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **EmailReplyToAddress**: This is the email address that will appear in the Reply-To header of the email, and will be used by the email client if the user attempts to reply to emails sent by the Web Portal. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **EmailOriginHost**: Specify the domain of the origin host, for emails sent by the server. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **EmailSettingsToConfirm**: A hash of the email settings that need to be confirmed before saving. See *Enabling Two-Factor Authentication for Superusers* (page 10) for details.
- **DockerHost**: This is the host name and port of the Docker daemon which runs the containers. See installing-docker for details.
- **ContainerImage**: This is the name of the image that must be used when creating a new container. See *Upgrading the Docker Container Image* (page 18) for details.
- **ContainerRoot**: This is the absolute path that reference the directory in which all containers will create the user data.
- **ContainerStorageTimeout**: This is the time, in minutes, that an container must be idle before its storage is removed. Zero means that the container storage is never deleted. See *Upgrading the Docker Container Image* (page 18) for details.
- **IdleContainerTimeout**: This is the time, in seconds, that a container must be idle before it is removed. Zero means that containers are never removed. See *Upgrading the Docker Container Image* (page 18) for details.
- **RemoveOldImages**: Set to `True` if containers running an old image (i.e. not equal to ContainerImage) should be removed. See *Upgrading the Docker Container Image* (page 18) for details.
- **OldImageTimeout**: This is the time, in seconds, that a container with an old image must be idle before it is removed. Zero means the container is removed, even if it is running. Note, if `RemoveOldImages` is `False`, this setting is ignored. See *Upgrading the Docker Container Image* (page 18) for details.
- **OldImageRemovalTime**: Use this setting to specify when containers with old images should be removed. You can set it to “now”, to remove the containers immediately, if set to “never”, then containers are only removed if the `OldImageTimeout` is exceeded. This value can also be set to a time (e.g. 03:00, format: hh:mm), or a date (format YYYY-MM-DD hh:mm). Note, if RemoveOldImages is `False`, this setting is ignored. See *Upgrading the Docker Container Image* (page 18) for details.
- **RegServerHost**: This is the host name of the Registration Server. See activate_web_portal for details.
- **RegAPIChecksumSalt**: This is the Registration Server API salt. It is required to authorise access to the Registration Server’s API. See activate_web_portal for details.

4.8 Web Portal Backup Considerations

The extent to which backup and failover is performed depends entirely on the service level you wish to provide.

In order to secure the configuration of the Web Portal, you must make a backup of the `webportal` MySQL database. Loss of the database will require a complete re-install of the Web Portal.

Quick recovery from failure of the Web Portal can be provided by replicating the `webportal` database to a standby machine.

You should also ensure that you have a backup of all the configuration files describe here: *List of relevant configuration files* (page 21). However, these files are rarely changed after the initial setup.

A standby Docker host is also recommended if a high level of availability is required. If the contents of the `ContainerRoot` is lost due to disk failure, or failure of the Docker host, users will have to re-enter there Spaces after they log into the Web Portal again. The only data that will be lost in this case is files that were being uploaded when the failure occurred, All other Space data is stored by the TeamDrive Hosting Service, and can be recovered from there.

In order to ensure a high level of availability, a standby Docker host may be used, and the contents of the `ContainerRoot` path can be copied to the standby system using `rsync`. Alternaives depend on the type of volume mounted at `ContainerRoot`. If the file system has sufficient redundance and can be mounted by the standby system at any time, The no further consideration are required.

Note that it is not necessary to make a backup of Docker containers, as these are automatically re-created when a user logs in.

4.9 Setting up Server Monitoring

It's highly recommended to set up some kind of system monitoring, to receive notifications in case of any critical conditions or failures.

Since the TeamDrive Web Portal is based on standard Linux components like the Apache HTTP Server and the MySQL database, almost any system monitoring solution can be used to monitor the health of these services.

We recommend using Nagios or a derivative like Icinga or Centreon. Other well-established monitoring systems like Zabbix or Munin will also work. Most of these offer standard checks to monitor CPU usage, memory utilization, disk space and other critical server parameters.

In addition to these basic system parameters, the existence and operational status of the following services/processes should be monitored:

- The MySQL Server (system process `mysqld`) is up and running and answering to SQL queries
- The Apache HTTP Server (`httpd`) is up and running and answering to http requests (this can be verified by accessing https://webportal.yourdomain.com/index.html and https://webportal.yourdomain.com/admin/index.html)
- The `td-webportal` service is up and running (process name `yvvad`)

4.10 Scaling a TeamDrive Web Portal Setup

When scaling the TeamDrive Web Portal we consider each component individually. The are four components that are relevant to this discussion: the Apache Web server, the Docker host, the MySQL Database and the Load Balancer.

The simplest configuration places all components on one machine. This is the case which is largely described in this document. In this case, the Apache Web server also fulfils the function of the Load Balancer. this is done by re-write rules which direct calls from the Web client to The associated Docker container.

Even in the case of a small scale setup, we recommend placing the Docker host on a separate system. This makes it easier to manage the resources required by Docker and the TeamDrive Agent running in the containers.

4.10.1 Apache Web Server

The Apache Web server host is responsible for the management of the Web Portal. This includes: the Login page, the Administration Console and the background tasks.

The scaling requirements of this component are relatively limited as the task do not require much resources in terms of CPU, memory or disk space.

This means that a “scale-up” of the Apache Web server host is probably quite sufficient to cope with a growing number of users.

Nevertheless, if the Web Portal access patterns require it, or simply to add redundancy it is possible to scale-out the Apache Web server, by adding additional machines that run the identical Web Portal software.

In this case a Load Balancer is required to distribute requests to the various Apache hosts. This can be done on a simple round-robin basis or according to current load since the connections are stateless.

The Web Portal service which runs the various background task should be started on all Apache hosts.

The MySQL Database must also be moved to a separate system. See below for more details.

4.10.2 MySQL Database

Load on the database, and the volume of data is minimal on the Web Portal. For this reason, it should suffice to place the MySQL database on a dedicated server as the load increases on the Web Portal. Additional CPU’s and memory can then be added to this system as required.

As mentioned above, if the Apache Web server is scaled out, then it is necessary to place the MySQL database on a separate system even if this is not required for load reasons. If this is not done then the MySQL database can remain on the same system as The Apache Web server.

4.10.3 Docker Service

The specific hardware requirements of the system running the Docker service are describe here: hardware-requirements. In this section we discuss the issues involved in scaling out the Docker service.

Depending in the usage patterns you will find it necessary to begin scale-out of the Docker service when the number of Users exceeds about 1000. In other words, a working estimate is that the Web Portal requires appropriately one Docker host per 1000 users.

A requirement for scale-out of the Docker system is software that manages a cluster of Docker hosts. There are a number of such tools available, including: Swarm, Shipyard, Google Kubernetes and CoreOS.

An important requirement of such systems is that they support the standard Docker API, which is used by the Web Portal. If this is the case, then the Web Portal will be able to start and manage containers in the cluster without regard to the number of hosts in the cluster.

The container storage used by a Docker cluster must be mounted by all hosts in the cluster. This means that the storage must be placed on a shared storage medium like an NFSv4 server or shared disk file systems like OCFS2 or GFS2. Note that concurrent access of the same volumen is required, but not concurrent access to the files on the volume. in other words, file locking is not an issue.

4.11 Upgrading the TeamDrive Web Portal

There are two aspects to upgrading the TeamDrive software used by the Web Portal: the Web Portal software, and the TeamDrive agent used by the Web Portal.

There is a dependency between two components because the Web Portal services the Web application that makes calls to the TeamDrive Agent. This means that, before the Web Portal software is updated, you must ensure that you are running the required, or latest, version of the TeamDrive Agent. Please check the Web Portal release notes (see *Release Notes - Version 1.0* (page 25)) which indicate the require version of the TeamDrive Agent.

Since the TeamDrive Agent is always backwards compatible with the Web application, you are free to use a more recent version then required. How to upgrade the TeamDrive Agent is describe in the following section: *Upgrading the Docker Container Image* (page 18).

Upgrading the TeamDrive Web Portal software is simple using The RPM package manager:

```
[root\@webportal ~]# yum update td-webportal
```

An update simply replaces the existing packages while the service is running, and the services (`httpd` and `td-webportal`) are automatically restarted afterwards.

Check the chapter *Release Notes - Version 1.0* (page 25) for the changes introduced in each new version. The release notes may also contain important notes that effect the upgrade itself.

4.12 Upgrading the Docker Container Image

The Docker container image used is stored in the `ContainerImage` setting and is set to `teamdrive/agent:latest` by default.

This means that the container image will automatically be updated whenever the image with the `latest` tag changes on your Docker host.

The upgrade of a container image cannot occur “in-place”. Instead, the old container must be removed, and a new container started which uses the new image.

During normal operation, containers are only removed when the are idle for a certain amount of time. This time is specified by the `IdleContainerTimeout` setting.

This means that if a container is in continuous use, then it will never by upgraded.

For this reason, a number of settings have been added to “force” upgrade of a container, even if the idle time-out is not exceeded. The settings that perform this task are `RemoveOldImages`, `OldImageTimeout` and `OldImageRemovalTime`.

`RemoveOldImages` must be set to `True` to enabled this functionality. It is also required that you use a container Image with an explicit version number.

Docker container images are available from the TeamDrive public Docker repostory on the docker hub. Here you will find a list of the taged images that are available:

```
https://hub.docker.com/r/teamdrive/agent/tags/
```

You can install an image using the Docker `pull` command:

```
[root@webportal ~]# docker pull teamdrive/agent:<tag>
```

Where `<tag>` is the tag of an image, for example: `4.0.12.1292-TMDR`. If you have installed the image on your Docker host, set the `ContainerImage` setting accordingly, for example: `teamdrive/agent:4.0.12.1292-TMDR`.

At this point the values of the settings `OldImageTimeout` and `OldImageRemovalTime` will take effect.

`OldImageTimeout` is the time, in seconds, that a container with an old image (an image other than `ContainerImage`) must be idle before it is removed. Zero means the container is removed immediately, even if it is running. Note, if `RemoveOldImages` is `False`, this setting is ignored.

`OldImageRemovalTime` specifies when containers with old images should be removed. Set this setting to a specific time of day (e.g. 03:00, format: hh:mm) or to a specific date (format YYYY-MM-DD hh:mm). This specifies the time when the upgrade will take place, which will force a running containers to be removed and re-created.

If you want to force upgrade immediately, set this setting to “now”. You can disable this setting by setting it to “never”. In this case, upgrade is controlled by the `OldImageRemovalTime` setting.

You will find more on the upgrade process in the description of the tasks that actually perform this functions, see background-tasks.

CHAPTER

FIVE

TROUBLESHOOTING

5.1 List of relevant configuration files

/etc/httpd/conf.d/td-webportal.httpd.conf: The configuration file that loads and enables the TeamDrive Web Portal Server-specific module for the the Apache HTTP Server: `mod_yvva.so`.

`mod_yvva.so` is responsible for providing the web-based Host Server Administration Console as well as an API used for authentication.

The file also contains various Apache "rewrite" rules required by the Web Portal.

Note: The rewrite rules in this file are disable by default. This is because it is assumed that HTTPS is always used to access the Web Portal.

Enable the rewrite rules only if you are certain that HTTP access may be used.

/etc/logrotate.d/td-webportal: This file configures how the log files belonging to the TeamDrive Web Portal are being rotated. See the `logrotate(8)` manual page for details.

/etc/td-webportal.conf: This file defines how the `td-webportal` background service is started using the `yvvad` daemon.

/etc/td-webportal.my.cnf: This configuration file defines the MySQL credentials used to access the `webportal` MySQL database. It is read by the Apache modules `mod_yvva` and the `yvvad` daemon that runs the `td-webportal` background tasks and the `yvva` command line client.

/etc/yvva.conf: This configuration file contains configuration settings specific to the Yvva Runtime Environment that effect the `mod_yyva` Apache module and the `yvva` command line shell.

5.2 List of relevant log files

In order to debug and analyse problems with the Web Portal configuration, there are several log files that you should consult:

/var/log/td-webportal.log: The log file for the Yvva runtime which provides the web-based Administration Console, the Web Portal authentication API. Errors that are incurred by the Web Portal background tasks are also written to this file.

Consult this log file when the Web Portal has issues in contacting the Registration Server, errors when handling API requests or problems with the Administration Console.

You can increase the amount of logging by changing the Yvva setting `log-level` from `notice` to `trace` or `debug` in the `yvva.conf` file:

```
log-level=trace
```

After changing `yvva.conf` you need to restart the Apache HTTP Server service using `service httpd restart`.

This log file is also used by the td-webportal background service. Checkk the log file verify that background tasks are being processed without errors.

The log file location can be configured by changing the file name passed to the log-file option in the configuration file /etc/td-webportal.conf. The log level can be increased by changing the default value notice for the log-level option to trace or debug.

Changing these values requires a restart of the td-webportal background process using service td-webportal restart.

/var/log/httpd/: The Apache HTTP Server's log files (e.g. error_log) might also contain additional relevant error messages that should be checked.

5.3 Enable Logging with Syslog

As outlined in *List of relevant log files* (page 21), the TeamDrive Web Portal logs critical errors and other notable events in a log file by default.

It is now possible to redirect the log output of the Yvva runtime components to a local syslog instance instead.

Syslog support is an essential feature for auditing, security and/or compliance reasons, as it allows you to funnel all log messages into a centralized syslog server.

This makes it easier to monitor the logs for critical events or errors and prevents tampering with the log files in case of a security breach. It also helps to maintain control over the disk space utilization on the server, as growing log files can't accidentally fill up the file system.

To enable syslog support, the log file name in the log-file setting has to be replaced with the keyword syslog. Optionally, a custom process identifier can be supplied, by appending it to the syslog keyword, using a colon as the separator, e.g. log-file=syslog:my_process_identifier. If not used, the default process identifier will be used, which is the name of the Yvva component.

To enable syslog support for the Yvva-based td-webportal background service, edit the log-file setting in file /etc/td-webportal.conf as follows:

```
log-file=syslog:webp-bkgr
```

You need to restart the td-webportal background service via service td-webportal restart in order to activate this change. If the log-level is set to debug you will now see log messages appearing in /var/log/messages:

```
Jun 23 11:57:33 localhost webp-bkgr: notice: yvvad startup
Jun 23 11:57:33 localhost webp-bkgr: notice: Using config file:
/etc/td-webportal.conf
Jun 23 11:57:33 localhost webp-bkgr: notice: No listen port
Jun 23 11:57:33 localhost webp-bkgr: notice: yvvad running in repeat 60
(seconds) mode
```

To enable syslog support for the Web Portal API and Administration Console, edit the /etc/yvva.conf file as follows:

```
log-file=syslog:webp-httd
```

You need to restart the Apache HTTP Server via service httpd restart in order to activate this change. If the log-level is set to debug you will now see log messages appearing in /var/log/messages:

```
Jun 23 12:06:04 localhost webp-httd: notice: mod_yvva 1.2.1 (May 21 2015
11:00:12) startup OK
```

5.4 Common errors

5.4.1 Web Installation: "500 Internal Server Error"

This error can be triggered by several error conditions. Check the log file `/var/log/td-webportal.log` for details.

Some common errors include:

```
[Error] -12036 (2002): Can't connect to local MySQL server through socket
'/var/lib/mysql/mysql.sock' (25)
[Error] "startup.yv" (80)
```

The local MySQL Server's socket file can't be opened. This could either be a permission problem, or the MySQL Server is simply not available. Check that MySQL is actually up and running (e.g. by running `service mysqld status`) and restart it, if necessary. If the error persists, check the MySQL error log file (usually `/var/log/mysqld.log`) for hints.

Similarly, an error like the following one indicates that a remote MySQL Server might not be answering (e.g. because of a firewall rule or because it's not running):

```
[Error] -12036 (2003): Can't connect to MySQL server on
'mysql.yourdomain.com' (107)
[Error] "startup.yv" (80)
```

If you see `Access denied` errors like the following one:

```
[Error] -12036 (1045): Access denied for user 'teamdrive'@'localhost' (using
password: YES)
[Error] "startup.yv" (32)
```

Either the username or password used to connect to the MySQL Server are wrong. Double check that the MySQL username and password provided in `/etc/td-webportal.my.cnf` are correct, e.g. by trying to connect to the MySQL server using these credentials with the `mysql` command line client.

If you see the following error when connecting to a remote MySQL Server:

```
[Error] -12036 (1130): Host 'regserver.yourdomain.com' is not allowed to
connect to this MySQL server
[Error] "startup.yv" (80)
```

Check the TeamDrive MySQL user's privileges on the remote MySQL server, e.g. by running `SHOW GRANTS FOR 'teamdrive'@'regserver.yourdomain.com';` and make sure that this user is allowed to connect to the MySQL server from the Registration Server's host.

5.4.2 Errors When Accessing the Registration Server

If the Web Portal fails to contact the Registration Server, check the `/var/log/td-webportal.log` log file, as well as `/var/log/td-regserver.log` on the Registration Server for hints.

See the Troubleshooting chapter in the Registration Server Installation Manual for details.

Note: Note that Registration Server version 3.5 or later is required by the Web Portal.

5.4.3 Errors When Accessing the Docker

If the Web Portal fails to contact the Docker daemon, first check If docker can be accessed using the command line interface, for example:

> [root@webportal install]# export DOCKER_HOST=tcp://<docker-host>:2375 [root@webportal install]# docker images

This command will list the available images. The Docker daemon must be accessible using TCP. How to configure docker for TCP access is explained here: installing-docker.

If the Web Interface does not work correctly it may be that the reference to the Docker host is not correct in the `/etc/httpd/conf.d/ssl.conf` file.

Open up this file and check that the you have followed the instructions in section configure-mod-ssl.

CHAPTER

SIX

RELEASE NOTES - VERSION 1.0

6.1 Key features and changes

This is the initial release of the Web Portal.

6.2 Change Log - Version 1.0.x

6.2.1 1.0.2 (2015-12-07)

- Fixed container language settings so that Spaces with non-ascii characters in the name now work.
- Corrected re-direct to external login pages under certain circumstances.
- Login with an email address now works.
- The Portal no longer creates containers based on the case of the input username, instead the actual username is used. This prevents the creation of duplicate containers for the same user.
- The Web Portal session will now timeout after 15 minutes idle time. The user is then required to login again.
- Implemented reset password functionality. Login after password has been forgotten now works. The user will receive a temporary password via email which is used to set a new password and login.
- Note, new re-write must be added to `/etc/httpd/conf.d/ssl.conf`':

  RewriteRule ^/requestResetPassword /yvva/requestResetPassword [PT] RewriteRule ^/tempPasswordLogin /yvva/tempPasswordLogin [PT]
- Fixed loading of favicon

6.2.2 1.0.1 (2015-10-27)

- OldImageRemovalTime setting was not visible.
- Updated WebPortal GUI to the latest 4.1.x version from the webfrontend branch.

6.2.3 1.0 (2015-10-08)

- Initial public release
- Web Portal 1.0 requires TeamDrive Agent version 4.0.12.1292 or later.

CHAPTER

SEVEN

APPENDIX

7.1 Abbreviations

PBT **PBT** is an object oriented language specifically designed for the programming of "server-side" functionality common to intra- and internet Web sites. A large share of the TeamDrive Host and Registration Server functionality is implemented in PBT. The code is parsed and executed by the Yvva application server components.

SAKH **S**erver **A**ccess **K**ey **H**TTP for TeamDrive 2.0 Clients

TDNS **T**eam **D**rive **N**ame **S**ervice

TDRS **T**eam **D**rive **R**egistration **S**erver

TDSV Same as **SAKH**, but for TeamDrive 3.0 Clients: **T**eam **D**rive **S**er**v**er

TSHS **T**eam **D**rive **S**calable **H**ost **S**torage.