TeamDrive Web Portal Administration

Release 2.0.1.0

Paul McCullagh, Eckhard Pruehs

2019

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.

“Docker” is a trademark or registered trademark of Docker, Inc.

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.

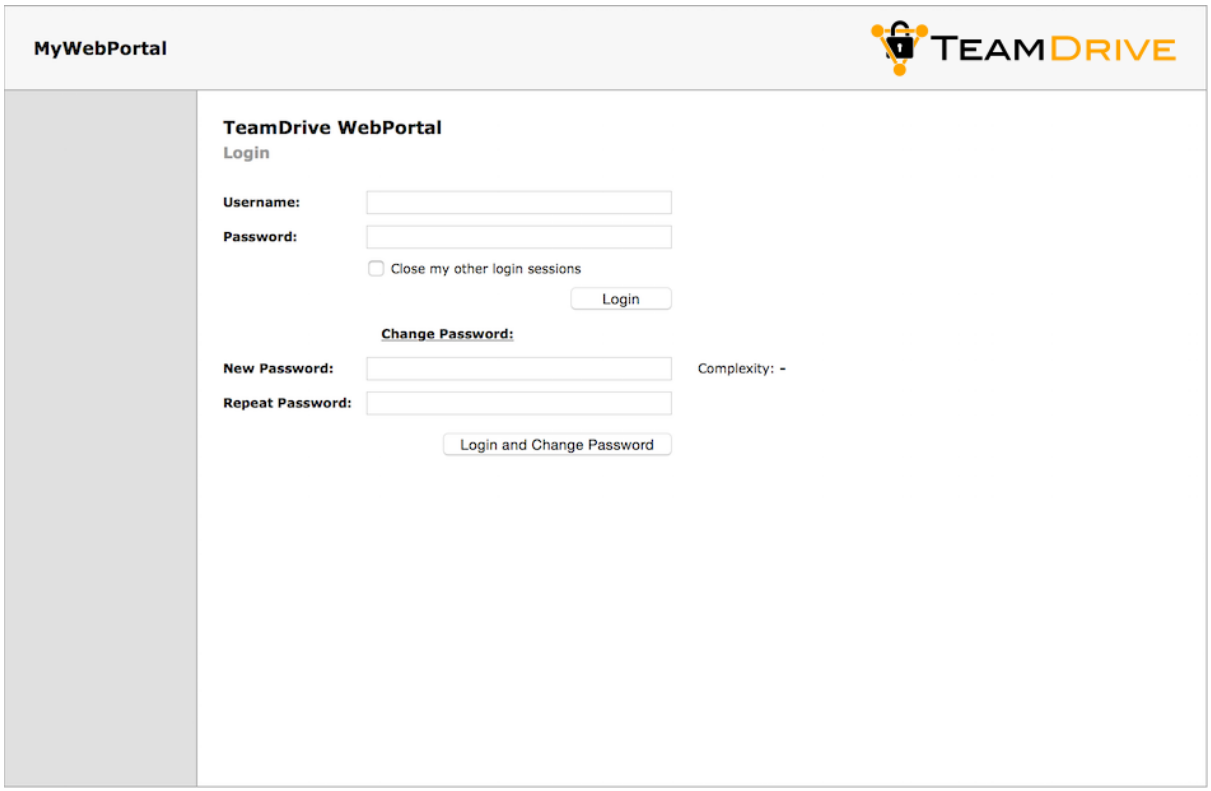

Fig. 4.1: Web Portal Administration Console: Change Password

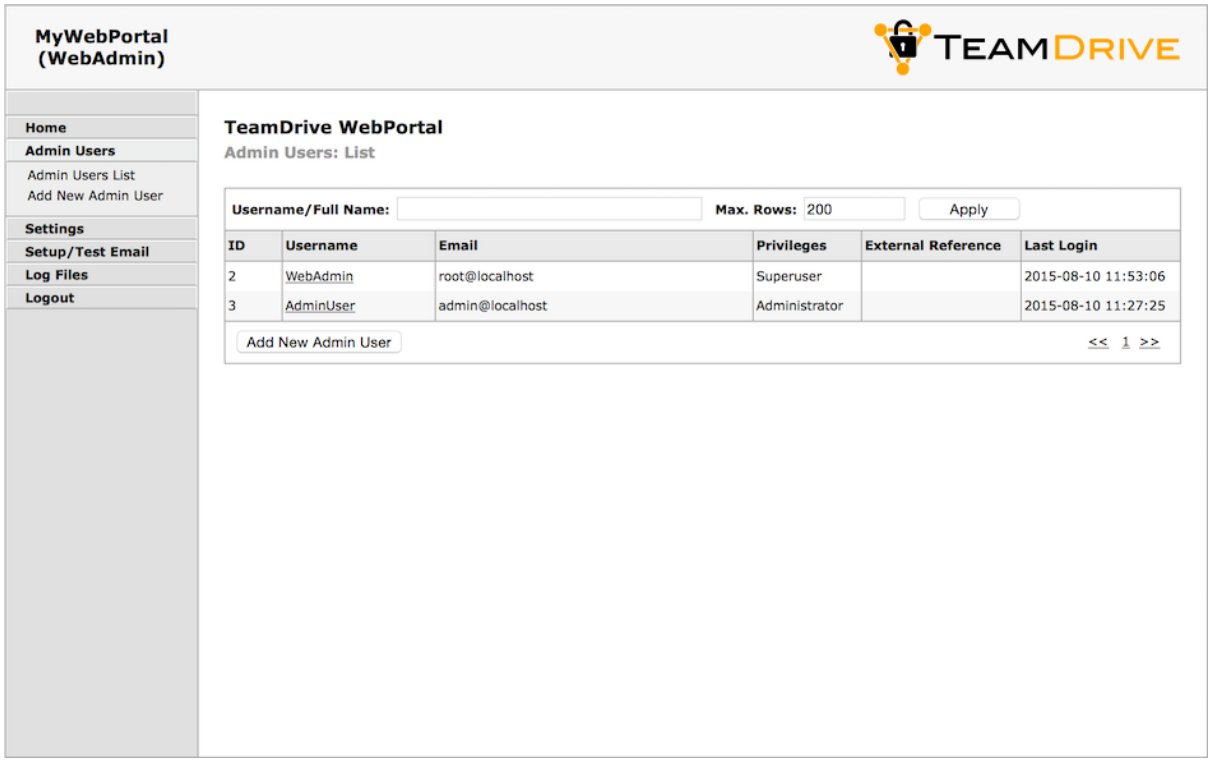

Fig. 4.2: Web Portal Administration Console: Admin Users List

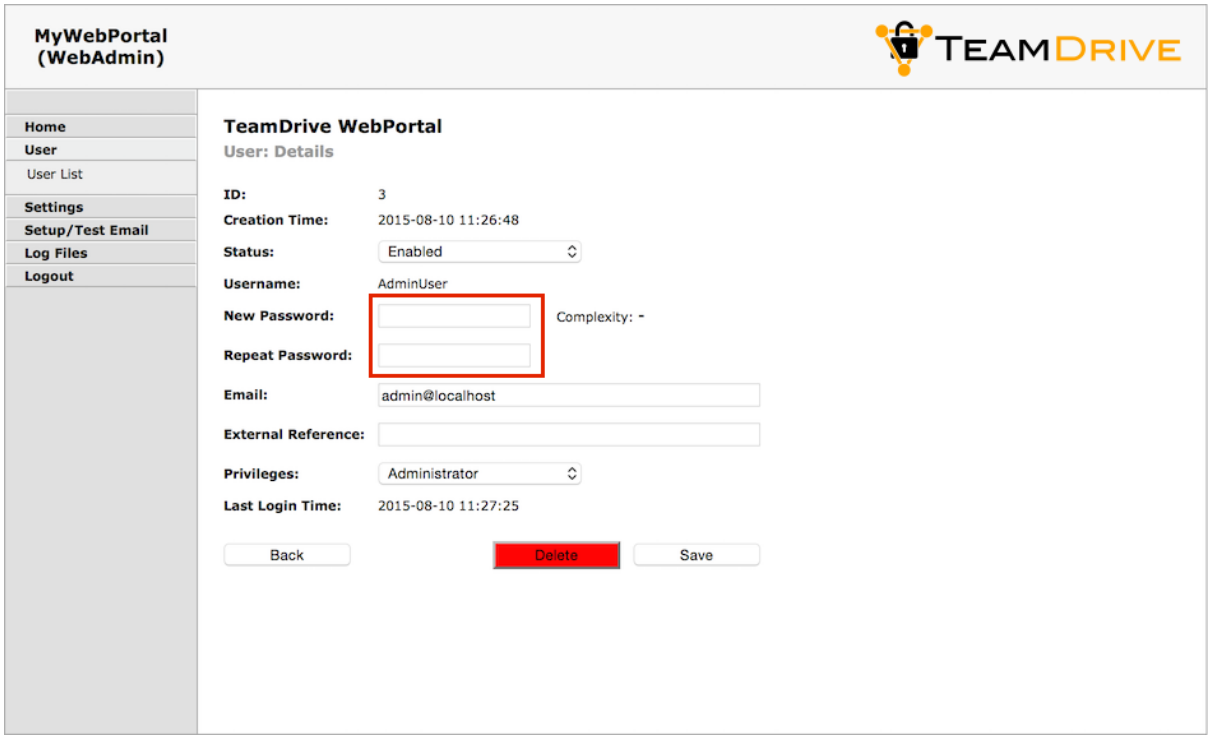

Fig. 4.3: Web Portal Administration Console: User Details

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 How to Enable Two-Factor Authentication

Two-factor authentication (2FA) can be enabled at two different levels:

- 2FA for the Web Portal Administrators
- 2FA for the users of the Web Portal

How to enable two-factor authentication for administrators is described in the section below (*Enabling Two-Factor Authentication for Administrators* (page 11)).

Two-factor authentication (2FA) for the Web Portal users requires the Registration Server version 3.6 or later. 2FA is implemented by the Registration Server using the Google Authenticator App (https://support.google.com/accounts/answer/1066447?hl=en).

To enable 2FA for users, set `AuthServiceEnabled` to `True`, and leave the associated settings: `AuthLoginPageURL`, `AuthTokenVerifyURL` and `RegistrationURL` blank.

As described in *Web Portal Settings* (page 21), these settings default to login and registration pages provided by the Web Portal. The Web Portal pages redirect to the associated pages provided by the Registration Server.

On the Registration Server the pages, can be optionally customised using the template system. The templates to be modified are: `portal-login`, `portal-lost-pwd`, `portal-register`, `portal-activate`, `portal-login-ok`, `portal-goog-auth-setup`, `portal-goog-auth-login`, and `portal-goog-auth-login-ok`.

If you would like to allow users to register directly via the Web Portal, then set `RegistrationEnabled` to `True`.

In order to setup two-factor authentication, users must be directed to the page:

```
https://webportal.yourdomain.com/portal/setup-2fa.html
```

This page provides instructions of how to configure Google Authentication for the user's account.

Note: Please check the apache ssl.conf for the additional RewriteRule in case you updated from WebPortal 1.0.5 to a newer version:

RewriteRule ^/portal(.*)$ /yvva/portal$1 [PT]

See configure-mod-ssl for details.

On the Registration Server you must add the domain name of the Web Portal (as specified by `WebPortalDomain`) to Provider setting `API_WEB_PORTAL_IP`. Modify this setting by adding the domain name on a line beneath the IP Address of the Web Portal which you have already set (as described in associate_portal_provider).

If the Web Portal is used by several Providers, only modify the `API_WEB_PORTAL_IP` setting of one of the Providers. This will be the default Provider for users that register directly via the Web Portal.

4.4 Enabling Two-Factor Authentication for Administrators

The Web Portal Administration Console supports two-factor authentication via email. In this mode, an administrator with "Superuser" privileges that logs-in with his username and password must 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.

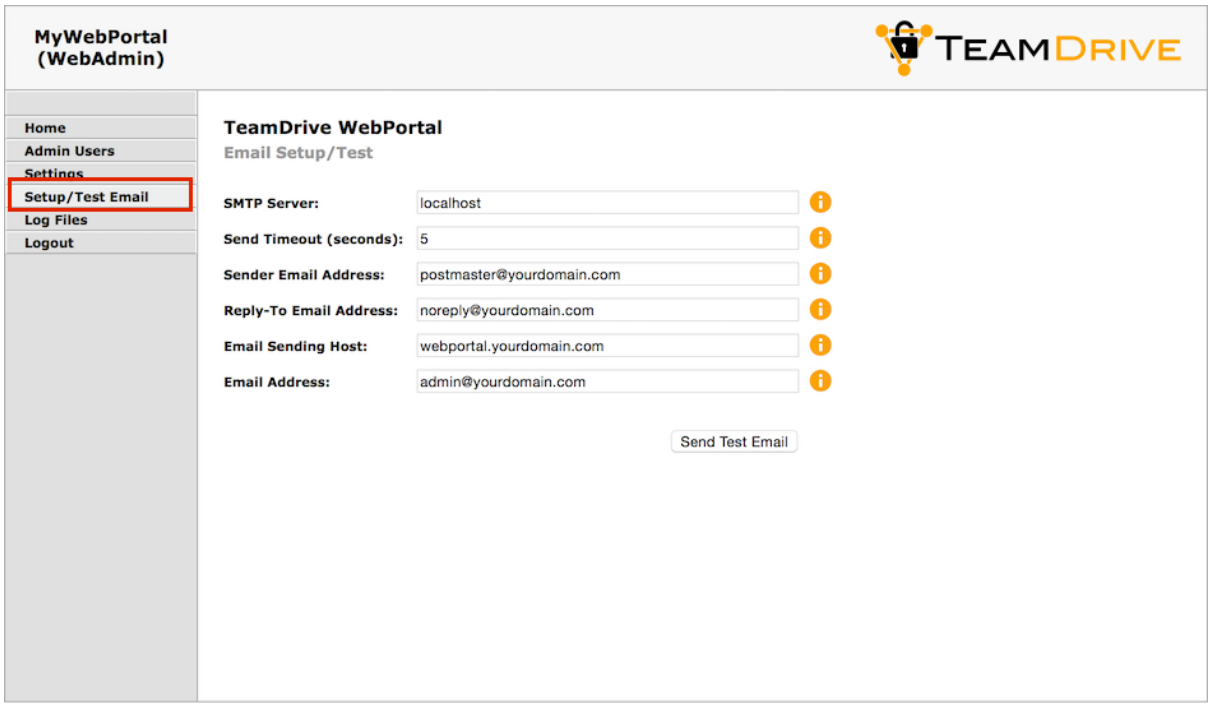

Fig. 4.4: Web Portal Admin Console: Email Setup / Test

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.

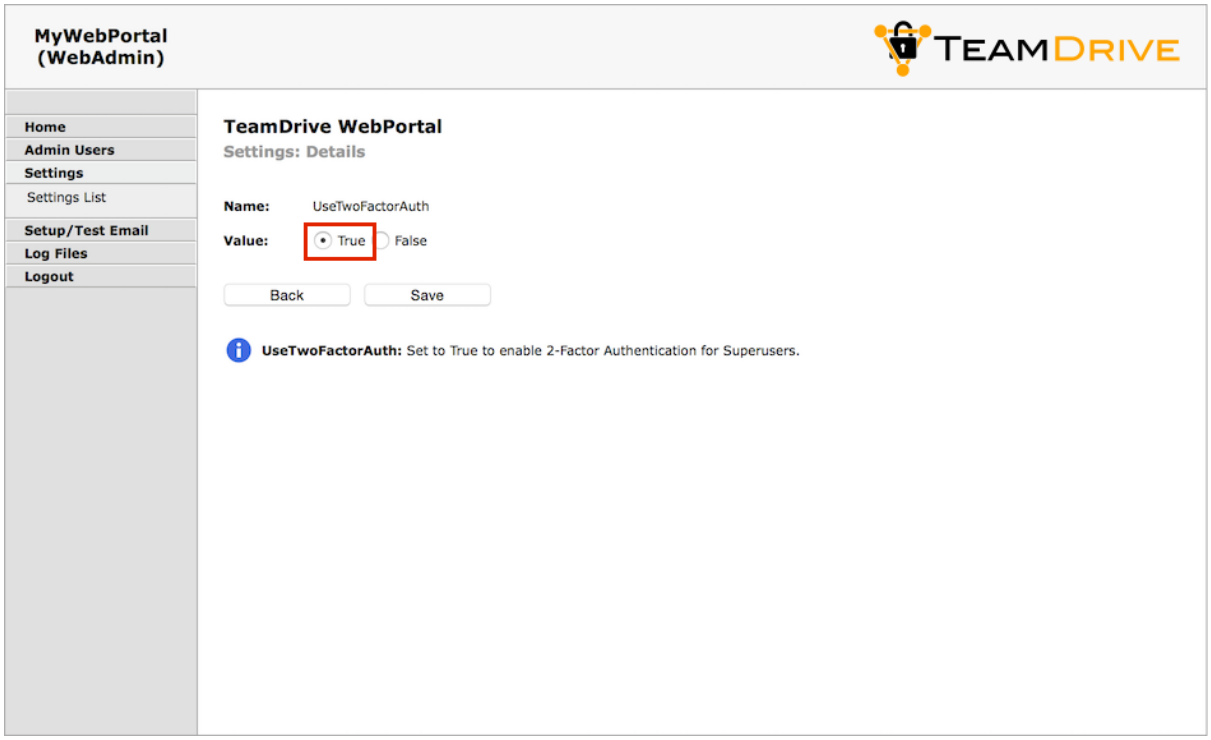

Fig. 4.5: Web Portal Admin Console: Use Two-Factor Authentication

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.

4.5 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.

4.6 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 13) below.

This is done by setting `AuthServiceEnabled` to `True`, and then setting the correct values for `AuthLoginPageURL` and `AuthTokenVerifyURL`. A Web Portal may only be connected to a single Authentication Service.

Note: Once external authentication service has been activated, it is no longer possible to use the Web Portal for regular login of users. This is because the user will always be redirected to the `AuthLoginPageURL` page.

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.

In particular, you must set `$webportal_domain` value in the `ldap_config.php` page (Registration Server 3.6).

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.

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 the login fails, first check the `/var/log/td-webportal.log` log file for errors.

4.7 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).

4.8 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: config_files. 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 their Spaces after they log into the Web Portal again. The only data that will be lost in this case are 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`. Alternatives depend on the type of volume mounted at `ContainerRoot`. If the file system has sufficient redundancy and can be mounted by the standby system at any time, Then 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. There 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 fulfills 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 volume is required, but not concurrent access to the files on the volume. in other words, file locking is not an issue.

With Web Portal version 1.0.10 the support for Docker Swarm is approved, but only for the legacy standalone Swarm setup (https://docs.docker.com/swarm/overview/), because of the different service model in the Docker Engine v1.12.0 using the swarm mode.

The *DockerHost* (page 24) setting must be changed to the Swarm port (default 2377).

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. The Web Portal requires a `MinimumAgentVersion` and will make sure that you are running the required 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 than required. How to upgrade the TeamDrive Agent is described in the following section: *Upgrading the Docker Container Image* (page 16).

Upgrading the TeamDrive Web Portal depends on using the standard or a White Label version. Updating the main Web Portal component is identical in both cases. Download the updated repository:

```
[root@webportal ~]# wget -O /etc/yum.repos.d/td-webportal.repo \
http://repo.teamdrive.net/td-webportal.repo
```

and update the Web Portal packages using the RPM package manager:

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

An update simply replaces the existing packages while the service is running, and the services (`httpd` and `td-webportal`) are automatically restarted afterwards. After the packages are updated proceed with the next chapter to update the Docker Container Images.

Check the chapter releasenotes 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 the minimum required agent version by default (see `MinimumAgentVersion`).

This means that the container image will automatically be updated when you manually increase the `ContainerImage` or a newer version of the Web Portal requires a newer `MinimumAgentVersion`.

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 they 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 timeout is not exceeded. The settings that perform this task are `RemoveOldImages`, `OldImageTimeout` and `OldImageRemovalTime`.

`RemoveOldImages` must be set to `True` to enabled this functionality.

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

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

You can install or update an image using `yvva` command line. This will automatically check the Docker hub for the newest published agent:

```
[root@webportal ~]# yvva
Welcome to yvva shell (version 1.3.8).
Enter "go" or end the line with ';;' to execute submitted code.
For a list of commands enter "help".

UPGRADE COMMANDS:
-----------------
To upgrade from the command line, execute:
yvva --call=upgrade_now --config-file="/etc/yvva.conf"

upgrade_now;;
Perform upgrade changes to the Docker image and/or database (this command cannot␣
↪be undone).
```

and in case of using a White Label version of TeamDrive, please follow the steps described in the chapter creating-white-label-agent-image. In case of upgrading from version 1.1 or below to a newer version, please follow the steps in *Upgrading a White Label installation from Version 1.1 to 1.2* (page 18)

Note: If outgoing requests has to use a proxy server, follow the Docker documentation https://docs.docker.com/engine/admin/systemd/#http-proxy to set a proxy for Docker. Restart the Docker service after adding the proxy configuration.

The successfull updated image will set the `ContainerImage` setting accordingly, for example: `teamdrive/agent:4.5.5.1838`.

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 container 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.

4.13 Upgrading a White Label installation from Version 1.1 to 1.2

Note: This step is only necessary when updating from a version 1.1 or below to version 1.2 (or later) to define your *White Label* (page 28) settings. Once you set your White Label settings, the update process is identical to the normal update process with just executing `upgrade_now;;` in the `yvva` command line.

The White Label GUI Web Portal RPM is no longer necessary and the existing package must be removed. Search for the old installed packages:

```
[root@webportal ~]# rpm -qa | grep "webportal-clientui"
```

and remove all listed packages using:

```
[root@webportal ~]# rpm -e <full package name>
```

Now download the updated repository:

```
[root@webportal ~]# wget -O /etc/yum.repos.d/td-webportal.repo \
http://repo.teamdrive.net/td-webportal.repo
```

and update the Web Portal packages using the RPM package manager:

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

The Web Portal version 1.2 or later is capable of building the White Label Docker image automatically. The description below assumes you are using a customised version of the TeamDrive Agent executable, or the Web-GUI.

Use the `yvva` command line (see below) or the Web Admin to fill in your *White Label* (page 28) product information.

The following required values are necessary to build the White Label Docker image:

- `UseWhiteLabelDockerImage`: Set this value to `True`.
- `WhiteLabelBinaryName`: The binary name of the linux agent ending with `.bin`.
- `WhiteLabelProductName`: The first part of the Agent archive (the tar.gz file). This value should be all lower-case. If not, please contact TeamDrive support.
- `WhiteLabelProviderCode`: Your 4 letter Provider code.

In addition to set these values, it may be necessary to modify the following settings:

- `WhiteLabelAgentDownloadURL`: By dafault this is the link to the TeamDrive download portal:

  ```
  http://s3download.teamdrive.net/{VERSIONSHORT}/{PROVIDERCODE}/linux-x86_64/
  ↪{PRODUCTNAME}_agent_{VERSION}_el7.x86_64.tar.gz
  ```

  See *WhiteLabelAgentDownloadURL* (page 28) for a detailed description of this value. If your Agent archive is not located on the TeamDrive portal, then you should set the value accordingly.

  The **{VERSION}** placeholder will be replaced by the highest version specified by the `ContainerImage` and `MinimumAgentVersion` settings.

- `WhiteLabelDISTRIBUTOR`: The content for the DISTRIBUTOR file for the agent. If left empty the DISTRIBUTOR file from the Agent archive (.tar.gz file) will be used.

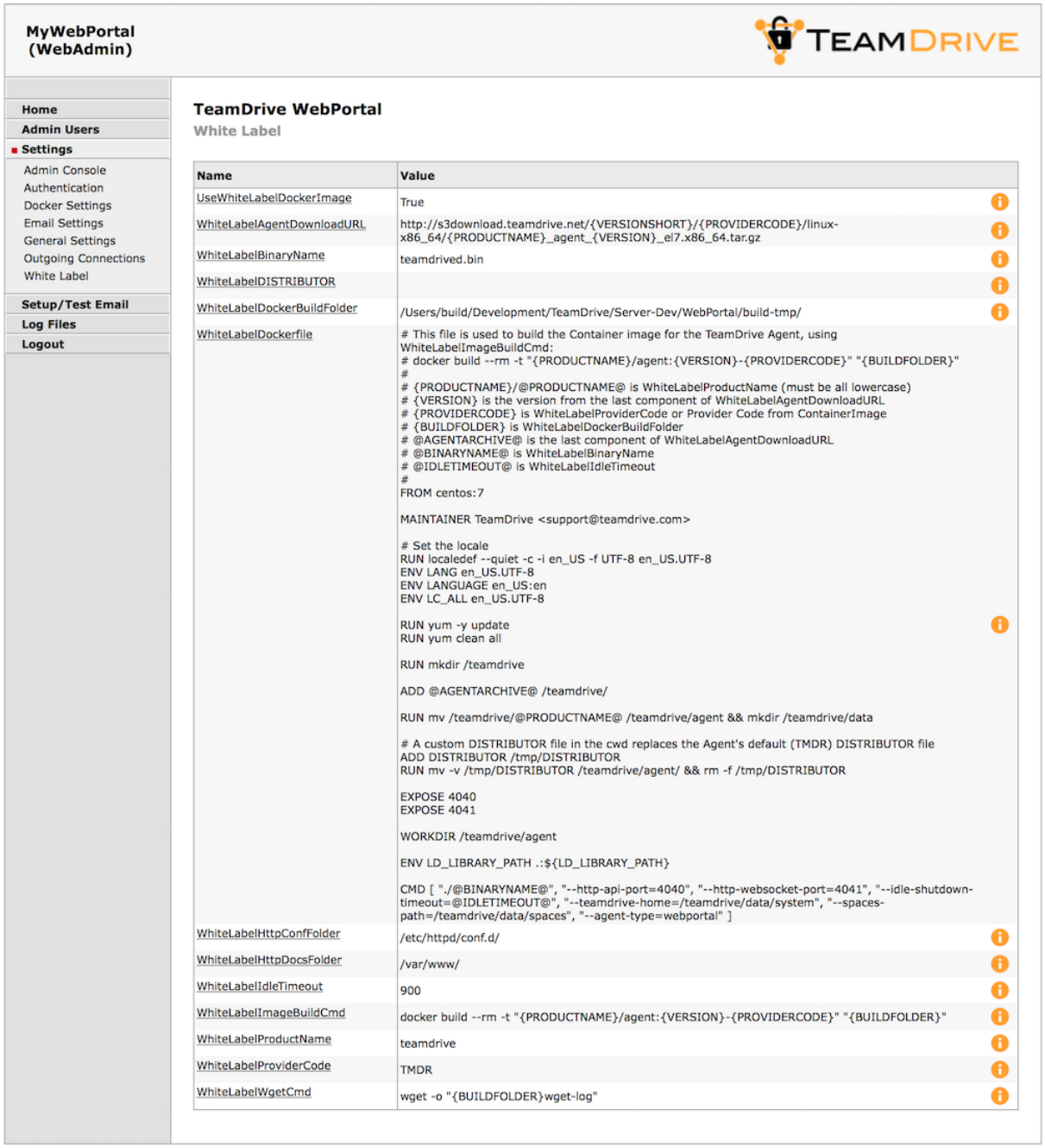

Fig. 4.6: Web Portal Admin Console: White label settings

- `WhiteLabelINIFileSettings`: Additional client settings which will be written to the /etc/teamdrive.ini file during the docker iamge build process.

To set your White Label settings using the `yvva` command line: Start yvva as `root` user and replace the following placeholders `<your-....>` with your values:

```
AppSetting:setSetting("UseWhiteLabelDockerImage", $true);;
AppSetting:setSetting("WhiteLabelBinaryName", "<your-binary-name>");;
AppSetting:setSetting("WhiteLabelProductName", "<your-product-name>");;
AppSetting:setSetting("WhiteLabelProviderCode", "<your-provider-code>");;
```

to verify your values execute:

```
print AppSetting:getSettingAsBool("UseWhiteLabelDockerImage");;
print AppSetting:getSettingAsString("WhiteLabelBinaryName");;
print AppSetting:getSettingAsString("WhiteLabelProductName");;
print AppSetting:getSettingAsString("WhiteLabelProviderCode");;
```

The optional parameters, `WhiteLabelAgentDownloadURL` and `WhiteLabelDISTRIBUTOR` can be a set in a similar manner (line breaks are permitted in strings).

However, it is easier to change settings like `WhiteLabelDISTRIBUTOR` in the Web Admin.

After these values have been set correctly, you can build a new Docker images by starting the `yvva` command line, and running the following command:

```
upgrade_now;;
```

The Web Portal will try to download the required White Label Agent archive version and build the new Docker image. The single steps will be logged to the console and will also show error messages if a step fails.

In case that the download fails or if you want to skip the download step, place your Agent archive in the `WhiteLabelDockerBuildFolder` and the update process will use it to create the Docker image. The image will be used to retrieve and update the Web-GUI as required.

More information on the process is provided in the section creating-white-label-agent-image.

CHAPTER

FIVE

WEB PORTAL SETTINGS

This chapter lists and describes the available configuration options for the TeamDrive Web Portal.

You can review and modify most of these via the TeamDrive Web Portal Admin Console by clicking **Settings**. Some settings are marked as read-only ("R/O"), they can not be changed.

The settings are grouped into sections:

5.1 Admin Console

5.1.1 ExtAuthEnabled

Set this value to `True` to enable External Authentication. See *Administrator Login using External Authentication* (page 13) for details.

5.1.2 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 13) for details.

5.1.3 Language

This is the default language used by the Web Portal Admin Console.

5.1.4 UseTwoFactorAuth

Set to `True` to enable two-factor authentication for Superusers.

Note that this setting only applies to the user of the Web Portal Admin Console. The setting has nothing to do with the use of two-facter authentication used by the users of the portal. This is described in the section: *How to Enable Two-Factor Authentication* (page 10).

5.2 API

5.2.1 APIAccessList

A list of IPs which are allowed to access the API of the Web Portal.

5.2.2 APIChecksumSalt

To detect "man in the middle" attacks when sending API requests to the Web Portal, a random "salt value" is generated during the initial installation. The sender must add this salt value to his request before calculating the MD5 hash value of the API request content which will be sent to the Web Portal.

The checksum will be included in the URL, so that the Web Portal can check if the content was modified during the transport.

This setting is read-only and can not be changed via the Admin web interface.

5.3 Authentication

5.3.1 AuthLoginPageURL

This is URL of the login page which is used to login using the external Authentication Service. See *Configuring Active Directory / LDAP Authentication Services* (page 13) for details.

When `AuthServiceEnabled` is `True`, the Web Portal login page: `https://webportal.yourdomain.com/portal/login.html`, redirects to the page specified by this setting.

If `AuthServiceEnabled` is `True`, but this setting has no value, then the Portal Login page provided by the Registration Server (version 3.6 or later) is used by default.

The Registration Server Portal Login page also allows the use of Two-factor authentication using the Google Authentication App. In this case, Two-factor authentication can be setup using the page: `https://webportal.yourdomain.com/portal/setup-2fa.html`, which redirects to the web-page that provides this service on the Registration Server.

The Registration Server Portal pages are customisable using the templates provided. Details are available in the Registration Server documentation.

5.3.2 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 13) for details.

When `RegistrationEnabled` is set to `True`, you must ensure that `AuthLoginPageURL` (see *AuthLoginPageURL* (page 22)) is set correctly.

In order to permit user registration via an Authentication Service you must also set `RegistrationEnabled` to `True` (see *RegistrationEnabled* (page 26)). In this case, you must also set `RegistrationURL` (see *RegistrationURL* (page 27)) correctly.

5.3.3 AuthTokenVerifyURL

This 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 13) for details.

BY default, this setting is set to the Web Portal verification URL: `https://webportal.yourdomain.com/portal/verify.html`

5.3.4 LicenseBuyURL

This URL will be displayed for a user, if **LicenseProfessionalRequired** is set and the user has no professional license.

5.3.5 LicenseProfessionalRequired

Login at the Web Portal requires a professional license for the user.

5.3.6 UseEmbeddedLogin

This setting determines whether the Web Portal uses the embedded, or non-embedded form of external login / registration.

External authentication can be embedded in the TeamDrive Web GUI, or can the external authentication pages can be used directly. Set `UseEmbeddedLogin` to `True` in order to use the embedded login form.

By default, `UseEmbeddedLogin` is set to `False` if you upgrade from a previous version of the Web Portal that was using external authentication, otherwise, the default is `True`.

Accessing the Web Portal domain, for example: `https://webportal.yourdomain.com`, will automatically present the login in the embedded or non-embedded form, as specified by `UseEmbeddedLogin`.

You can now use "explicit" links to the login page in order to set the default provider code and language, for the login or registration.

For the non-embedded login form use the following explicit link:

```
https://webportal.yourdomain.com/portal/login.html?dist=CODE&lang=LG
```

and for the embedded login form use the following explicit link:

```
https://webportal.yourdomain.com/extauth/login.html?dist=CODE&lang=LG
```

where `CODE` is the provider code, and `LG` is the language code, for example `en` or `de`.

Note that the external authentication service must be able to handle the specified provider code and language.

5.4 Docker Settings

5.4.1 ContainerImage

This is the name of the image that must be used when creating a new container. See *Upgrading the Docker Container Image* (page 16) for details.

Note that if the `MinimumAgentVersion` specifies a TeamDrive agent version that is higher than the version of the Agent specified by `ContainerImage`, then the container image used will be determined by `MinimumAgentVersion`.

5.4.2 ContainerRoot

This is the absolute path that reference the directory in which all containers will create the user data.

5.4.3 ContainerStorageTimeout

This is the time, in minutes, that a 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 16) for details.

5.4.4 CurrentGUIVersion

The version of the installed GUI package. The update process will retrieve or build a new Docker container (see update process for details). The GUI package will be extracted from this container and the HTML pages, images and javascript code will be located in the apache document root. The GUI version should be identical to the `ContainerImage` version.

5.4.5 DockerHost

This is the host name and port of the Docker daemon which runs the containers. See installing-docker for details.

5.4.6 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 16) for details.

5.4.7 MinDockerDataSpaceAvailable

A minimum value in GB for the available Docker data space on the storage (see https://docs.docker.com/engine/userguide/storagedriver/device-mapper-driver/#/configure-docker-with-devicemapper) If the minimum value is reached, no more Docker container for new users will be created. Set to 0 to disable checking the available Docker data space.

5.4.8 MinDockerMetaDataSpaceAvailable

A minimum value in GB for the available Docker meta data space on the storage (see https://docs.docker.com/engine/userguide/storagedriver/device-mapper-driver/#/configure-docker-with-devicemapper) If the minimum value is reached, no more Docker container for new users will be created. Set to 0 to disable checking the available Docker meta data space.

5.4.9 MinimumAgentVersion

This setting is specifies the minimum TeamDrive Agent version that is required by the Web Portal. The setting may not be modified. If The current image used by containers has a Agent version that is earlier than `MinimumAgentVersion`, then upgrade of the containers will be forced by the Web Portal. This means that users may experience a spontaneous logout.

Following upgrade, `ContainerImage` will be set to the required image.

5.4.10 MaxActiveContainer

A parameter to limit the currently active users. Set to 0 to disable the limitation.

5.4.11 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 16) for details.

5.4.12 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 16) for details.

5.4.13 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 16) for details.

5.5 Email Settings

5.5.1 EmailOriginHost

Specify the domain of the origin host, for emails sent by the server. See *Enabling Two-Factor Authentication for Administrators* (page 11) for details.

5.5.2 EmailSendTimeout

Timeout in seconds, when sending an email. See *Enabling Two-Factor Authentication for Administrators* (page 11) for details.

5.5.3 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 Administrators* (page 11) for details.

5.5.4 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 Administrators* (page 11) for details.

5.5.5 EmailSettingsToConfirm

A hash of the email settings that need to be confirmed before saving. See *Enabling Two-Factor Authentication for Administrators* (page 11) for details.

5.5.6 SMTPServerHost

Domain name (and port) of the SMTP server used to send emails. See *Enabling Two-Factor Authentication for Administrators* (page 11) for details.

5.6 General Settings

5.6.1 AllowedProviders

This is a list of Provider codes of the users that may login to the Portal. If empty, any user may login to the Portal.

Note: Changes to the list will not be recognized by running container instances. You have to stop all running instances manually.

5.6.2 ForceHTTPSUsage

Set to `True` if the Web Portal Admin Console must be accessed using HTTPS.

5.6.3 MaxRecordsDisplayed

This setting determines the maximum number of records that may be retrieved from the database at any time. This parameter may only be changed by a Superuser.

5.6.4 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.

Before you can change‘‘RegAPIChecksumSalt‘‘ you may have to set `RegServerHost` to blank (the empty string), if it references a server that does not math the new `RegAPIChecksumSalt` value.

After changing `RegAPIChecksumSalt` you can then set `RegServerHost` correctly.

5.6.5 RegistrationEnabled

Set to `True` in order to allow users to register directly From the Web Portal. By default this value is set to `False`.

The setting `RegistrationURL` (see *RegistrationURL* (page 27)) specifies the URL that provides the registration page.

When `RegistrationEnabled` is set to `True` there are 2 possibilities, depending on whether `AuthServiceEnabled` (*AuthServiceEnabled* (page 22)) is set to `True` or `False`.

If `AuthServiceEnabled` is `True`, then registration uses the external Authentication Service mechanism which results in the user being logged-in, immediately after registration.

When `AuthServiceEnabled` is `True`, it is possible to use the customisable registration page provided by the Registration Server (version 3.6 or later). In this case ‘‘RegistrationURL‘‘must not be set (see *RegistrationURL* (page 27)) .

If `AuthServiceEnabled` is `False`, then the TeamDrive Agent Web-GUI provides a “Register Now” button which references this page specified by `RegistrationURL`, in the login dialog.

In this case, the page referenced by `RegistrationURL` is a custom developed web-page which performs registration using the Registration Server API and then redirects to the Web Portal login page: `https://webportal.yourdomain.com/portal/login.html`.

5.6.6 RegistrationURL

This URL references a Web-page where a user can register as a TeamDrive user. Alternatively, if an external Authentication Service is being used this page allows users to register with this service.

This page will only be used of `RegistrationEnabled` is set to `True`.

The Web Portal register page: `https://webportal.yourdomain.com/portal/register.html`, automatically redirects to the page.

If `RegistrationEnabled` is `True`, but this setting has no value, then the Portal Registration page provided by the Registration Server (version 3.6 or later) is used by default. In this case, `AuthServiceEnabled` (see *AuthServiceEnabled* (page 22)) must be set to `True`.

If `RegistrationEnabled` is `True` and `AuthServiceEnabled` is `False` then this setting must reference a custom developed web-page which performs registration using the Registration Server API and then redirects to the Web Portal login page: `https://webportal.yourdomain.com/portal/login.html`.

5.6.7 RegServerHost

This is the domain name of the Registration Server. See activate_web_portal for details.

Before you can set `RegServerHost` you may have to set `RegAPIChecksumSalt` to blank (the empty string) if the current value of `RegAPIChecksumSalt` does not match the Registration Server.

After changing `RegServerHost` you can then set `RegAPIChecksumSalt` correctly. This is necessary because the Web Portal will attempt to verify the Registration Server connection and will fail if `RegAPIChecksumSalt` is set to an incorrect value.

5.6.8 ServerRoot

The installation directory of the Web Portal application.

5.6.9 SessionTimeout

This is the idle time in seconds after which you are required to login to the Web Portal Admin Console again.

5.6.10 WebPortalDomain

This is the domain name (or URL) of this service.

5.6.11 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.

5.7 Outgoing Connections

5.7.1 UseProxy

Set this value to `True` in order to enable the use of a proxy for all outgoing connections of the Web Portal, and the TeamDrive Agent running in the Docker environment.

5.7.2 ProxyHost

This is the domain name (or IP address) and port number of the proxy to be used for outgoing connections. If not set, the `UseProxy` setting will be ignored.

Note that this setting is used for both HTTP and HTTPS connections.

5.7.3 NoProxyList

This is a comma separated list of domains and IP addresses that are to be contacted without the use of a proxy.

5.7.4 ConnectionTimeout

The timeout in milliseconds when making outbound connections. The default is 30 seconds.

5.8 White Label

The White Label settings are used to build a custom Docker Container image for use with the Web Portal.

5.8.1 UseWhiteLabelDockerImage

Set to `True` in order to build a White Label Docker image for use with the Web Portal.

5.8.2 WhiteLabelAgentDownloadURL

This URL is used to download the TeamDrive Agent archive (.tar.gz file).

By default the URL refers to the TeamDrive download portal:

```
http://s3download.teamdrive.net/{VERSIONSHORT}/{PROVIDERCODE}/linux-x86_64/
↪{PRODUCTNAME}_agent_{VERSION}_el7.x86_64.tar.gz
```

Before usage, the following substitutions are made:

- **{PRODUCTNAME}** is set to `WhiteLabelProductName`, after converting to all lowercase letters.
- **{PROVIDERCODE}** is set to the value of the `WhiteLabelProviderCode` setting.
- **{VERSION}** is set to the version of the Agent being built.
- **{VERSIONSHORT}** a short version of the version number of the archive, which does not include the "patch" number. Version numbers have the form: <major>.<minor>.<patch>.<build>

If you have your own download portal, you can remove the placeholders as required.

If the required TeamDrive Agent archive is found in the build folder (`WhiteLabelDockerBuildFolder`) the Web Portal will not attempt to download the archive.

5.8.3 WhiteLabelBinaryName

`WhiteLabelBinaryName` is the name of TeamDrive Agent binary executable which is started when launching a Docker container. The executable is included in the Agent archive (.tar.gz file).

By default, this value is "teamdrived.bin".

5.8.4 WhiteLabelDISTRIBUTOR

This is the contents of the signed White Label DISTRIBUTOR file. This value replaces the contents of the DISTRIBUTOR file included in the Agent archive.

By default this value is empty, which means that the DISTRIBUTOR file in the Agent archive is used.

Please notice, that only signed DISTRIBUTOR files will be accepted. The signature will be checked during the creation of the docker image and at each start of the agent.

Own settings should be set using `WhiteLabelINIFileSettings`.

The default contents for the TeamDrive Agent are as follows:

```
code=TMDR
reg-server-list-url=http://reg.teamdrive.net/pbas/td2as/lis/regserverlist.htm
reg-server-name=TeamDriveMaster
reg-server-url=http://reg.teamdrive.net/pbas/td2as/reg/
notification-url=http://notification.teamdrive.net/pbas/td2as/reg/
media-server-url=http://media.teamdrive.net/pbas/td2as/reg/
update-program-url=http://reg.teamdrive.net/pbas/td2as/upd/update.xml
balance-url=http://balance.teamdrive.net/pbas/td2as/bal/balance.xml
log-upload-url=http://logupload.teamdrive.com/upload.php
redirector-url=http://www.teamdrive.com/redirector.php
ping-url=http://ping.teamdrive.net/ping.xml

enable-provider-panel-android=false
enable-provider-panel-ios=false
enable-provider-panel-linux=true
enable-provider-panel-mac=true
enable-provider-panel-win=true
```

5.8.5 WhiteLabelINIFileSettings

Additional client settings which will be written to the /etc/teamdrive.ini file during the creation of the docker image.

5.8.6 WhiteLabelDockerBuildFolder

This is the folder in the filesystem where the files are created during the Docker image build process.

If the Agent archive cannot be downloaded then it may be copied manually to this directory before the build is initiated (see installing-agent-image).

The DISTRIBTOR file and the Dockerfile used to build the Docker image are created in this directory. Since these files are not deleted you can check the contents after the build is completed.

5.8.7 WhiteLabelDockerfile

The contents of the Dockerfile used by docker to build a new TeamDrive Agent image, as described in the Docker documentation: https://docs.docker.com/engine/reference/builder/.

A number of replacements are performed before the file is used:

- **@AGENTARCHIVE@** is set to the last component of the `WhiteLabelAgentDownloadURL` setting.
- **@BINARYNAME@** is set to the value of the `WhiteLabelBinaryName` setting.
- **@IDLETIMEOUT@** is set to the value of the `WhiteLabelIdleTimeout` setting.
- **@PRODUCTNAME@** is set to `WhiteLabelProductName`, after converting to all lowercase letters.

After substitution, the Web Portal uses the value of `WhiteLabelImageBuildCmd` to call docker to create the image.

5.8.8 WhiteLabelHttpDocsFolder

This must be set to the path to the Apache documents folder. By default, the value is "/var/www/". There is no need to change this setting if you are running the Web Portal on CentOS 6 or CentOS 7.

5.8.9 WhiteLabelHttpConfFolder

The path to the Apache folder for configuration files, "/etc/httpd/conf.d/" by default. There is no need to change this setting if you are running the Web Portal on CentOS 6 or CentOS 7.

5.8.10 WhiteLabelIdleTimeout

This is a timeout value in seconds that determines when the TeamDrive Agent running in a container will automatically shutdown, stopping the container. The default value is 15 minutes.

Since this value is hardcoded in when the Docker image is build, changing this take effect after building and a new custom Docker image.

5.8.11 WhiteLabelImageBuildCmd

This is a shell command which calls the docker executable to build a new Docker image. The only reason to change this setting is to determine the "docker" executable to be used by specifying the path of the executable.

Before usage, the following substitutions are made:

- **{BUILDFOLDER}** is set to the value of the `WhiteLabelDockerBuildFolder` setting.
- **{PRODUCTNAME}** is set to `WhiteLabelProductName`, after converting to all lowercase letters.
- **{PROVIDERCODE}** is set to the value of the `WhiteLabelProviderCode` setting.
- **{VERSION}** is set to the version of the Agent being built.

5.8.12 WhiteLabelWgetCmd

This is a shell command calls the wget executable to download the TeamDrive Agent archive. Additional arguments (e.g. -O, -e and the download URL) will be added to this command as required.

The only reason to change this setting is to determine the "wget" executable to be used by add a path, or to specify a different location for the log file.

Before usage, **{BUILDFOLDER}** is set to the value of the `WhiteLabelDockerBuildFolder` setting.

If the wget call fails, check the "wget-log" log file for details.

5.8.13 WhiteLabelProviderCode

This is your White Label 4 letter Provider code. By default, the Provide code is "TMDR".

5.8.14 WhiteLabelProductName

Your White Label linux Product name. The default Product name is "teamdrive".

Note that the Product name is required to be all lowercase letters.

This value is the first part of the name of the Agent archive (.tar.gz file) which contains the binary of the TeamDrive Agent, as specified by the last component of the `WhiteLabelAgentDownloadURL` setting, for example: "teamdrive_agent_4.5.5.1838_el7.x86_64.tar.gz".

When the Agent archive is unpacked, the Web Portal assumes that the top-level directory is the same as the value of this setting. In addition, when upgrading, the Web Portal will create a Docker image with a name of the form:

```
<white-label-product-name>/agent:<version-number>-<provider-code>.
```

The Web Portal also uses the image name to search the Docker hub before building a custom image.

CHAPTER

SIX

TROUBLESHOOTING

6.1 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 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 module `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.

6.2 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, and 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. Check the log file to 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.

6.3 Enable Logging with Syslog

As outlined in list of relevant log files, 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
```

6.4 Common errors

6.4.1 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
'webportal.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 'webportal.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`@`webportal.yourdomain.com` ``; and make sure that this user is allowed to connect to the MySQL server from the Registration Server's host.

6.4.2 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.

6.4.3 Errors When Accessing 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 you have followed the instructions in section configure-mod-ssl.

CHAPTER

SEVEN

RELEASE NOTES - VERSION 2.0

Note: Please follow the new update process described in chapter *Upgrading the TeamDrive Web Portal* (page 16). The former separate GUI rpm package is not longer necessary. The standard Web Portal will update the docker Container image from the docker hub during the update step and will extract and update the files necessary for the GUI from this image. A white label Web Portal needs the white label agent .tar.gz to build a white label docker container image.

7.1 Key features and changes

- Increased MinimumAgentVersion to 4.6.7.2328
- External authentication supports both login and registration. This feature can be activated by setting `AuthServiceEnabled` to True. To allow registration set `RegistrationEnabled` to True. If no `AuthLoginPageURL` or `RegistrationURL` page is specified then the Web Portal will use the "portal pages", provided by the Registration Server.
- External authentication can be embedded in the TeamDrive Web GUI, or can the external authentication pages can be used directly. A new setting: `UseEmbeddedLogin`, must be set to `True` in order to use the embedded login form.

  By default, `UseEmbeddedLogin` is set to `False` if you upgrade from a previous version of the Web Portal that was using external authentication. Otherwise, the default is `True`. This is to ensure backwards compatibility, with previous versions that only supported the non-embedded form.

  Accessing the Web Portal domain, for example: `https://webportal.yourdomain.com`, will automatically present the login in the embedded or non-embedded form, as specified by `UseEmbeddedLogin`.
- You can now use "explicit" links to the login page in order to set the default provider code and language, for the login or registration.

  For the non-embedded login form use the following explicit link:

  ```
  https://webportal.yourdomain.com/portal/login.html?dist=CODE&lang=LG
  ```

  and for the embedded login form use the following explicit link:

  ```
  https://webportal.yourdomain.com/extauth/login.html?dist=CODE&lang=LG
  ```

  where `CODE` is the provider code, and `LG` is the language code, for example `en` or `de`.

  Note that the external authentication service must be able to handle the specified provider code and language.

7.2 Administration Console

- Added a Container list page, which can be used to search for containers of a particular user and type. The container details page allows you to stop, start and delete containers.

  Note that deleting a container will remove all the container data as well. This means that Web Portal users will find all spaces deactivated on next login. If the user looses his password he will also loose access to his data, unless he has a TeamDrive installation elsewhere.

7.3 Change Log - Version 2.0

7.3.1 2.0.1 (2019-06-11)

- Fixed problems the on demand creation and starting of containers that have been deleted (WEBCLIENT-304).

7.3.2 2.0.0 (2019-04-25)

- Initial release of Web Portal 2.0.

CHAPTER

EIGHT

RELEASE NOTES - VERSION 1.2

Note: Please follow the new update process described in chapter *Upgrading the TeamDrive Web Portal* (page 16). The former separate GUI rpm package is not longer necessary. The standard Web Portal will update the docker Container image from the docker hub during the update step and will extract and update the files necessary for the GUI from this image. A white label Web Portal needs the white label agent .tar.gz to build a white label docker container image.

8.1 Key features and changes

- Simplified installing and updating the web portal and docker container for standard and white label configuration.
- Increased MinimumAgentVersion to 4.5.2.1775 to support PointInTime-Recovery and Read-Confirmations

8.2 Change Log - Version 1.2

8.2.1 1.2.3 (YYYY-MM-DD)

- Reset of Admin User's password as described in the documentation (i.e. by setting the password to blank in the database) was not working (WEBCLIENT-259).
- Added a illustrated overview of the Web Portal to the documentation, showing the connection to other components in the TeamDrive system (see introduciton_to_the_teamdrive_web_portal).

8.2.2 1.2.2 (2018-11-06)

- The Web Portal supports now the Docker Community and Enterprise Edition and also still the old Commercially Supported version with the latest version 1.13 (from January 2017). Please notice, that the Docker CS version will be still maintained, but not further developed any more. Check the docker installation chapter for the differences between Docker CS and CE/EE installation.
- The Web Portal will only allow using signed DISTRIBUTOR files like the standard client. The signature will be checked during the creation of the docker image and at each start of the agent. Additional client settings must be moved to the new setting `WhiteLabelINIFileSettings`. If settings are still required in the DISTRIBUTOR file it must be signed by TeamDrive Systems for you.
- The current agent supports now web-sockets to refresh data in the browser without refreshing the page itself. To support web-socket connections, the apache module `proxy_wstunnel_module` must be enabled (See configure-apache-24 for details)
- Increased MinimumAgentVersion to 4.6.4.2183

8.2.3 1.2.1 (2017-11-29)

- Increased MinimumAgentVersion to 4.5.5.1838
- Upgrade will change the `WhiteLabelAgentDownloadURL` setting from ".../{PRODUCTNAME}_agent_{VERSION}_x86_64.tar.gz" to ".../{PRODUCTNAME}_agent_{VERSION}_el7.x86_64.tar.gz". this is done because the TeamDrive agent is now built in 2 versions: "el6" are built for CentOS 6, and "el7" versions are built for CentOS 7. It is assumed that the Web Portal is run on a CentOS 7 platform. If this is not the case, then you must manually change this setting to ".../{PRODUCTNAME}_agent_{VERSION}_el6.x86_64.tar.gz" (WEBCLIENT-255).
- Updated documentation to include new TeamDrive CI (WEBCLIENT-254).
- The Web Portal external authentication now handles transitioning to a new User Secret generation algorithm as implemented by Registration Server version 3.7.6.
- Bug fix: boolean settings were not correctly pre-selected.
- Several improvements have been made to the upgrade procedure which generates a new Docker image. The setting `WhiteLabelAgentDownloadURL` can now be left blank, of the Agent archive (.tar.gz file) has been placed manually in the build folder (`WhiteLabelDockerBuildFolder`).

  If `ContainerImage` is set to image with a version number higher than the `MinimumAgentVersion`, then the Web Portal will build an image for the version specified by `ContainerImage`.
- Version 1.2.1 requires YVVA runtime version 1.4.4.

8.2.4 1.2.0 (2017-08-14)

- Initial 1.2 release.

CHAPTER

NINE

RELEASE NOTES - VERSION 1.1

Note: When updating from an older version of the Web Portal, remove the DOCKER_HOST setting in the apache config file `/etc/sysconfig/httpd`. It is not longer necessary.

If you update docker to version 1.12.6 the docker service might not start anymore as described in the docker release notes: https://github.com/docker/docker/releases/tag/v1.12.6 Please remove the file `/etc/systemd/system/docker.service.d/web-portal.conf` and add the `--host=tcp://0.0.0.0:2375` instead to the OPTIONS parameters in `/etc/sysconfig/docker` as described in the docker configuration installing-docker chapter.

9.1 Key features and changes

- Added professional license required check (WEBCLIENT-233)
- Added setting to limit currently active users (WEBCLIENT-234)
- Added setting for minimum docker available data and meta data space. If minimum is reached, no more docker container will be created for new users (WEBCLIENT-235)
- Settings are now displayed in groups in the Admin Console (WEBCLIENT-237).
- Increased MinimumAgentVersion to 4.3.2.1681 to support space web access settings (TDCLIENT-2184). The webportal docker agent will be started with an additional setting agent-type=webportal to distinguish a standard and a webportal agent
- Added settings to support a Proxy for outgoing connections: `UseProxy`, `ProxyHost` and `NoProxyList` (WEBCLIENT-242). See *Outgoing Connections* (page 27) for details.
- Added the `ConnectionTimeout` setting which specifies a timeout for outgoing connections (see *Outgoing Connections* (page 27)).
- Added support for Docker Swarm. Docker Swarm is a native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host. Please notice, that only the legacy standalone Swarm is supported (https://docs.docker.com/swarm/overview/), because of the different service model in the Docker Engine v1.12.0 using the swarm mode. Change the `DockerHost` Web Portal setting from the standard docker port 2375 to the swarm port 2377 to switch from the standard docker API access to the swarm API access (WEBCLIENT-245).

9.2 Change Log - Version 1.1

9.2.1 1.1.0 (2017-04-10)

- Initial 1.1 release.

CHAPTER

TEN

RELEASE NOTES - VERSION 1.0

10.1 Key features and changes

This is the initial release of the Web Portal.

10.2 Change Log - Version 1.0

10.2.1 1.0.9 (2017-02-10)

- Increased MinimumAgentVersion to 4.3.1.1656 to fix a bug when login with email address and magic usernames.
- Revised chapter Web Portal Virtual Appliance with CentOS 7 and docker direct-lvm storage

10.2.2 1.0.8 (2017-02-07)

Note: After updating docker to version 1.12.6 the docker service might not start anymore as described in the docker release notes: https://github.com/docker/docker/releases/tag/v1.12.6 Please remove the file `/etc/systemd/system/docker.service.d/web-portal.conf` and add the `--host=tcp://0.0.0.0:2375` instead to the OPTIONS parameters in `/etc/sysconfig/docker` as described in the docker configuration installing-docker chapter.

- Removed support for CentOS 6
- Fixed docker configuration
- Fixed PDF creation for this documentation
- Fixed download links for VM-Ware images

10.2.3 1.0.7 (2016-11-10)

- Increased MinimumAgentVersion to 4.2.2.1579 to support email notifications
- Fixed docker configuration
- Fixed apache 2.4 configuration

10.2.4 1.0.6 (2016-07-11)

Note: Updating the Host Server on CentOS 7 with "yum update" might update the apache to a newer version. This update could re-install the deleted "conf"-files in the folder `/etc/httpd/conf.modules.d/` and will prevent starting the apache. Please follow the modified instruction to disable all modules in the "conf"-files instead of deleting them as described in configure-apache-24

- Improved Docker installation documentation (WEBCLIENT-219, WEBCLIENT-223).
- The Web Portal now checks if the user is authorised to access a Web Portal. A user is authorised to access a Web Portal if the Provider setting: `ALLOW_WEB_PORTAL_ACCESS` is set to `permit` or `ALLOW_WEB_PORTAL_ACCESS` is set to `peruser` and the user's "Web Portal Access" capability bit is set (a user-level setting).

  When using external authentication, the same check is done if the Registration Server is version 3.6 or later. When using a Registration Server 3.5 or earlier, the Web Portal will not check the user's Web Portal access permissions (in the case of external authentication).
- Added setting `AllowedProviders` which is a list of Provider codes of the users that are allowed to login to the Web Portal.

  An input field on the setup page allows this variable to set during installation of the Web Portal.
- The URL `https://webportal.yourdomain.com/portal/authservice.html` is now the target URL for external Authentication Services acting on behalf of the Web Portal.

  In other words, in successful authorisation by an external Authentication Service, the user is redirected back to this page.

  The Web Portal will may add certain arguments to `AuthLoginPageURL` and `RegisterURL` pages:

  - "portal=true": This argument is always added to the URL. This is useful, in the case when the same Authentication Service is called by the TeamDrive Client and the Web Portal. The argument can be used to determine whether to redirect on successful login or not.
  - "cookie=?": This argument will be added if the Authentication Service provided a cookie after the last successful login. The cookie is stored by the TeamDrive Agent.
  - "error=?": This argument indicates that the Web Portal encountered an error after successful authorisation by the Authentication Service. It is a base-64 (URL) encoded string containing the error message. The error should be displayed in the login page served by the Authentication Service.
- Support CentOS 7 with Apache 2.4
- Increased MinimumAgentVersion to 4.2.0.1470 to support the space activities
- Added setting `RegistrationEnabled` (default `False`). This value must be set to `True` to allow registration of users directly via the Web Portal.
- Added login and registration pages: All of these pages redirect to the associated pages on the Registration Server. After login, or registration, the Registration Server redirects back to the Web Portal.
  - `https://webportal.yourdomain.com/portal/login.html` This page allows users to login using two-factor authentication, if this has been configured. `/portal/login.html` is now the default for the `AuthLoginPageURL` setting.
  - `https://webportal.yourdomain.com/portal/register.html` Using this page a user can register as a TeamDrive user without installing the TeamDrive Client. After registration the user has access to the Web Portal. `/portal/register.html` is now the default for the `RegisterURL` setting.
  - `https://webportal.yourdomain.com/portal/lost_pwd.html` This page sends a temporary password to the user and allows the user to login and set a new password. The page is linked from `/portal/login.html`.

  - `https://webportal.yourdomain.com/portal/setup-2fa.html` Using this page the user can configure two-factor authentication using the Google Authenticator App.
  - *The default of the ''AuthTokenVerifyURL'* setting is now: `https://webportal.yourdomain.com/portal/ve`

10.2.5 1.0.5 (2016-02-16)

- Fixed a problem on login with a user registered via the Registration Server API using email address as identification (WEBCLIENT-205).
- Use the -v option when removing containers. This ensures that the container volume is also removed (WEBCLIENT-204).

10.2.6 1.0.4 (2016-02-09)

- Framework synced with Host- and Reg-Server

10.2.7 1.0.3 (2016-02-02)

- Added setting `MinimumAgentVersion` which specifies the minimum version of the TeamDrive Agent that will work with the Web Portal. Upgrade to this version of the Agent is forced as soon as the new version of the Web Portal is online (WEBCLIENT-194).
- Updated documentation for Docker version 1.7.1
- Fixed Internet explorer caches API calls. (WEBCLIENT-186)
- Added description about the dependencies between Webportal, Provider and Reg-Server and normal and external Authentication. (WEBCLIENT-176)
- The performExternalAuthentication redirects to http:// instead of https://. (WEBCLIENT-182)
- The getLoginInformation() API call now returns "registerUrl" if the setting RegistrationURL, is set on the Web Portal. (WEBCLIENT-179)
- Redirect to the login page when a request to an agent returns a 503 code. This requires a manual update to the ssl.conf, refer to the documentation on server installation and configuration. (WEBCLIENT-198)

10.2.8 1.0.2 (2015-12-07)

- Fixed container language settings so that Spaces with non-ascii characters in the name now work.
- Corrected redirect 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

10.2.9 1.0.1 (2015-10-27)

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

10.2.10 1.0 (2015-10-08)

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

CHAPTER

ELEVEN

APPENDIX

11.1 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, Registration Server and Webportal Server functionality is implemented in PBT. The code is parsed and executed by the Yvva application server components.

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

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

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