WebHare can send outgoing mail through Amazon AWS SES, and process bounces/complaints through SNS. This guide will use the Amazon 'aws' command line utility, but you can also do the process through the web interface.

SMTP Account

Configure a SMTP account with AWs in the AWS console. It will provide you with the email/smtp server name, username and password.

In WebHare, set mail settings: SMTP server (eg email-smtp.eu-central-1.amazonaws.com), port (587), username and password.

Bounces and feedback

To prepare your WebHare for SES, you need to ensure it can process bounces. You need to do this once per installation that will use SES.

Before you begin you'll need to find your WebHare server's SNS endpoint (in WebHare's mail configuration) and set up your SES/SNS credentials.

```
# Set up a topic name - use eg 'myserver-email'
aws sns create-topic --name <TOPIC>

# The above command will return the topic's full ARN which you'll need
# to specify to --topic-arn.
aws sns subscribe --topic-arn <TOPIC-ARN> --protocol https --notification-endpoint <SNS ENDPOI
```

Verify your identity

To be allowed to actually send, you'll need to verify each identity you'll be using at either the email or domain level. To verify an email address:

```
# Specify the email address to verify
aws ses verify-email-identity --email-address <EMAIL>
```

and follow the email instructions you receive.

You can also verify a full domain, but that'll also require some DNS setup:

```
aws ses verify-domain-identity --domain <DOMAIN>
```

```
aws ses  get-identity-dkim-attributes --identities <DOMAIN>
```

Configure the identity

Whether you choose a mail or domain entity, you now need to configure this identity to send its delivery information to this topic (and not just mail it back to you). To link the identity to your

SNS topic:

```
# Receive all three notification types
aws ses set-identity-notification-topic --identity <IDENTITY> --notification-type Bounce --sns-topic
aws ses set-identity-notification-topic --identity <IDENTITY> --notification-type Complaint --sns-top
aws ses set-identity-notification-topic --identity <IDENTITY> --notification-type Delivery --sns-topic

# Enable header information (WebHare needs the Message-IDs)
aws ses set-identity-headers-in-notifications-enabled --identity <IDENTITY> --notification-type Bou
aws ses set-identity-headers-in-notifications-enabled --identity <IDENTITY> --notification-type Cor
aws ses set-identity-headers-in-notifications-enabled --identity <IDENTITY> --notification-type Deli

# Disable email feedback
aws ses set-identity-feedback-forwarding-enabled --identity <IDENTITY> --no-forwarding-enabled
```

If none of the above steps returned an error, you're almost done. Set up WebHare's mailrouting, and if you're setting this up for the newsletter module (Pronuntio), don't forget to disable its own bounce handling for each relevant account.

Testing

Amazon provides simulator email addresses you can use to test your configuration. For example, emailing bounce@simulator.amazonses.com should allow you to see the Bounce status in the managed queue mail details.

WebHare 4.28 offers a simpler <filetype> definition syntax which makes prebuilt files obsolete.

A quick guide to switch from prebuilt types to filetypes:

Change

```
1    <prebuiltpage type="dynamic" tag="prebuilt:type" library="lib.whlib"
2                  webpageobjectname="Page" />
```

to

```
1    <filetype namespace="http://example.net/xmlns/filetype" kind="virtualfile">
2     <dynamicexecution webpageobjectname="lib.whlib#Page" />
3    </filetype>
```

And set up a conversion script:

```
<?wh

LOADLIB "mod::system/lib/database.whlib";
LOADLIB "mod::system/lib/migrations.whlib";

OpenPrimary()->BeginWork();
ConvertPrebuiltFiles("prebuilt:type", "http://example.net/xmlns/filetype");
GetPrimary()->CommitWork();
```

URL History

When you move content in the Publisher application WebHare can automatically redirect visitors to the new URL. This is done by recording all URLs to which content has been published in the so called *URL History.*

This URL history is not archived or copied when content is synchronized. If you intend to move a site to a different WebHare server and restructure it there before moving it back to the production environment, you may lose the URL history if you're not careful. To avoid this, you need to manually save the URL history on the original publication server and restore it after migration using the Publisher's 'whfstool' on the WebHare command line. This converts the URL history into contenttype data that can be migrated.

You can enable the webfeature "Show 'saved URL history'" to enable an additional file properties tab that will display the URL history. This allows editors to see and manipulate the URL history manually on their testing environment.