How to set up a staging server to test new Mautic releases

Updating to the latest version of Mautic doesn't necessarily have to be a risky or time-consuming endeavor. Aside from new features, bug fixes - especially security-related bug fixes - are a good reason to upgrade to the latest Mautic release. The community may not be able to help you with an issue as effectively if you are still running an outdated version of Mautic.

Using a "staging" server - a replica of your production Mautic instance - to test whether an update is successful before applying it to production, is a best practice to prevent unexpected errors with your live marketing database.

In this article, you will learn how to clone your Mautic installation to a staging server and manually perform a Mautic update to the latest stable version.

The article assumes that you are running Mautic on a VPS or dedicated server (not shared hosting), where you have root access to the Linux command line to update Mautic. It also assumes that the MySQL database is on the same server as Mautic. Before getting started, make sure you have access to the following credentials handy:

- Root access to the Mautic server (root password and/or SSH key)
- Username and password to the provider dashboard (e.g. Amazon AWS, DigitalOcean, Linode, etc.) where your Mautic instance is hosted
- Access to where your DNS settings are managed. Usually your domain registrar (e.g. GoDaddy, NameCheap, etc.) - if you are using the registrar's DNS service (e.g. nsXX.domaincontrol.com), or hosting company - if you have pointed your domain to your hosting provider's nameservers.

If your hosting provider offers a “snapshot” or “image” option that is the simplest way to clone Mautic along with all of its dependencies (e.g. PHP and its related modules). This ensures that you are testing against the same web server and PHP versions on staging as in production. This way you can accurately see how the new version of Mautic would behave according to your specific server configuration.

Create a production server snapshot

Amazon AWS refers to this functionality as creating an EBS snapshot of the root volume of your EC2 instance (virtual server) running Mautic. Other providers, such as DigitalOcean call this taking a snapshot image of your Mautic Droplet. Either way, the hosting provider should allow you to launch a new virtual server based on the snapshot that is a clone of your original server. Prior to taking the snapshot however, you should ensure that you have disabled cron jobs by commenting out the Mautic-related cron jobs in the user-specific cron tab:

crontab -u www-data -e

or the system-wide cron tab

/etc/crontab

To comment out a crontab entry, simply add the # symbol at the beginning of the line.

This ensures that the staging server will not actually send out segment emails or trigger actions on published campaigns once it is switched on, which would be duplicated on your live server as well. After taking the snapshot you can safely remove the # symbols to re-enable cron jobs on your live instance.

Create a staging server based on the snapshot

Next, deploy the staging server using the snapshot that you have previously taken. The process for this will slightly vary based on the provider.

In AWS you would create a new EBS volume from the snapshot, then attach it as the root volume to a separate EC2 instance.

In DigitalOcean, you would create a new droplet and select the snapshot as the image that should be used as the droplet's boot disk.

Point a separate 'A' record at the staging server

Note down the public IP address of the new, staging server, and for Amazon AWS, ensure that it has been added to a security group that allows inbound traffic on HTTP (80) and HTTPS (443) to the server.

From your domain registrar or managed DNS service (e.g. Route 53, Dyn, No-IP), create a new A record that points a staging subdomain (e.g. staging-mautic.example.com) at the public IP address of the staging server.

If you can configure the TTL (time to live) value you should set it to as low as possible (e.g. 300 seconds) so that the new record propagates across the Internet as rapidly as possible.

Modify staging server (e.g. Apache) configuration

After booting up the staging server, you're still not quite done yet, because the new server is still listening for HTTP(S) requests from the domain or subdomain of the production instance. You need to change the VirtualHost URL in Apache or Nginx, as well as the site_url value in **/path/to/mautic/app/config/local.php**.

Most Mautic installations in the wild are running Apache. If you are using Ubuntu, your Apache configuration file is either located in /etc/apache2/sites-available or /etc/apache2/conf.d. If you are using CentOS or RHEL, your

config file is located in /etc/httpd/conf.d by default.

Find the configuration file(s) with the ServerName value corresponding to your production Mautic URL and modify any instances of *ServerName* to the subdomain that you will be using for testing. If you are running Mautic on HTTPS, you should make sure to modify *ServerName* for both the port 80 and port 443 VirtualHost blocks. Depending on how it was set up, these VirtualHost blocks may be contained in two separate Apache config files.

If using HTTPS, also make sure to modify *SSLCertificateFile*, *SSLCertificateKeyFile*, and *SSLCertificateChainFile* values in the port 443 VirtualHost to refer to a path containing a valid SSL certificate for the staging subdomain.

If using Let's Encrypt to obtain certificates, the directory structure is /etc/letsencrypt/live/subdomain.example.com/. Refer to the Let's Encrypt Certbot documentation for instructions how to obtain a valid certificate for the staging subdomain.

Also, if you set up any 301 or 302 redirects in your VirtualHost files to redirect HTTP to HTTPS, remember to update those redirects with the staging subdomain as well.

```
<VirtualHost *:80>ServerName staging-mautic.example.com…</VirtualHost
><IfModule mod_ssl.c><VirtualHost *:443>ServerName staging-mautic.exa
mple.com…SSLCertificateFile /etc/letsencrypt/live/staging-mautic.exam
ple.com/cert.pemSSLCertificateKeyFile /etc/letsencrypt/live/staging-m
autic.example.com/privkey.pemSSLCertificateChainFile /etc/letsencrypt
/live/staging-mautic.example.com/chain.pem</VirtualHost></IfModule>
```

After making all of the changes above, restart the Apache server with the following commands.

Ubuntu: sudo service apache2 reload

CentOS: sudo systemctl restart httpd

Modify site_url in Mautic config on staging server

Using a text editor, change the site_url value in /path/to/mautic/app/config/local.php to reflect the staging site URL. To prevent permissions errors, you should modify this file as the web server user (www-data on Ubuntu, apache on CentOS/RHEL) using the command (replacing user in both cases below with the appropriate web server user):

sudo -u username nano /path/to/mautic/app/config/local.php

Next, clear the Mautic cache so that the staging site URL takes effect.

sudo -u username php /path/to/mautic/bin/console cache:clear

Test the Mautic update on the staging server

On the staging server, run the following commands to perform a Mautic update to the latest stable version, and bring the database schema into alignment with the latest version. If prompted whether to proceed at the doctrine migration or schema update steps, respond ‘y’ for yes (replacing user in all cases below with the appropriate web server user).

Updating to the latest version of Mautic doesn't necessarily have to be a risky or time-consuming endeavor. Aside from new features, bug fixes - especially security-related bug fixes - are a good reason to upgrade to the latest Mautic release. The community may not be able to help you with an issue as effectively if you are still running an outdated version of Mautic.

Using a "staging" server - a replica of your production Mautic instance - to test whether an update is successful before applying it to production, is a best practice to prevent unexpected errors with your live marketing database.

In this article, you will learn how to clone your Mautic installation to a staging server and manually perform a Mautic update to the latest stable version.

The article assumes that you are running Mautic on a VPS or dedicated server (not shared hosting), where you have root access to the Linux command line to update Mautic. It also assumes that the MySQL database is on the same server as Mautic. Before getting started, make sure you have access to the following credentials handy:

- Root access to the Mautic server (root password and/or SSH key)
- Username and password to the provider dashboard (e.g. Amazon AWS, DigitalOcean, Linode, etc.) where your Mautic instance is hosted
- Access to where your DNS settings are managed. Usually your domain registrar (e.g. GoDaddy, NameCheap, etc.) - if you are using the registrar's DNS service (e.g. nsXX.domaincontrol.com), or hosting company - if you have pointed your domain to your hosting provider's nameservers.

If your hosting provider offers a "snapshot" or "image" option that is the simplest way to clone Mautic along with all of its dependencies (e.g. PHP and its related modules). This ensures that you are testing against the same web server and PHP versions on staging as in production. This way you can accurately see how the new version of Mautic would behave according to your

specific server configuration.

Create a production server snapshot

Amazon AWS refers to this functionality as creating an EBS snapshot of the root volume of your EC2 instance (virtual server) running Mautic. Other providers, such as DigitalOcean call this taking a snapshot image of your Mautic Droplet. Either way, the hosting provider should allow you to launch a new virtual server based on the snapshot that is a clone of your original server. Prior to taking the snapshot however, you should ensure that you have disabled cron jobs by commenting out the Mautic-related cron jobs in the user-specific cron tab:

crontab -u www-data -e

or the system-wide cron tab

/etc/crontab

To comment out a crontab entry, simply add the # symbol at the beginning of the line.

This ensures that the staging server will not actually send out segment emails or trigger actions on published campaigns once it is switched on, which would be duplicated on your live server as well. After taking the snapshot you can safely remove the # symbols to re-enable cron jobs on your live instance.

Create a staging server based on the snapshot

Next, deploy the staging server using the snapshot that you have previously taken. The process for this will slightly vary based on the provider.

In AWS you would create a new EBS volume from the snapshot, then attach it as the root volume to a separate EC2 instance.

In DigitalOcean, you would create a new droplet and select the snapshot as the image that should be used as the droplet's boot disk.

Point a separate 'A' record at the staging server

Note down the public IP address of the new, staging server, and for Amazon AWS, ensure that it has been added to a security group that allows inbound traffic on HTTP (80) and HTTPS (443) to the server.

From your domain registrar or managed DNS service (e.g. Route 53, Dyn, No-IP), create a new A record that points a staging subdomain (e.g. staging-mautic.example.com) at the public IP address of the staging server.

If you can configure the TTL (time to live) value you should set it to as low as possible (e.g. 300 seconds) so that the new record propagates across the Internet as rapidly as possible.

Modify staging server (e.g. Apache) configuration

After booting up the staging server, you're still not quite done yet, because the new server is still listening for HTTP(S) requests from the domain or subdomain of the production instance. You need to change the VirtualHost URL in Apache or Nginx, as well as the site_url value in **/path/to/mautic/app/config/local.php**.

Most Mautic installations in the wild are running Apache. If you are using Ubuntu, your Apache configuration file is either located in /etc/apache2/sites-available or /etc/apache2/conf.d. If you are using CentOS or RHEL, your config file is located in /etc/httpd/conf.d by default.

Find the configuration file(s) with the ServerName value corresponding to your production Mautic URL and modify any instances of *ServerName* to the subdomain that you will be using for testing. If you are running Mautic on HTTPS, you should make sure to modify *ServerName* for both the port 80 and

port 443 VirtualHost blocks. Depending on how it was set up, these VirtualHost blocks may be contained in two separate Apache config files.

If using HTTPS, also make sure to modify *SSLCertificateFile*, *SSLCertificateKeyFile*, and *SSLCertificateChainFile* values in the port 443 VirtualHost to refer to a path containing a valid SSL certificate for the staging subdomain.

If using Let's Encrypt to obtain certificates, the directory structure is /etc/letsencrypt/live/subdomain.example.com/. Refer to the Let's Encrypt Certbot documentation for instructions how to obtain a valid certificate for the staging subdomain.

Also, if you set up any 301 or 302 redirects in your VirtualHost files to redirect HTTP to HTTPS, remember to update those redirects with the staging subdomain as well.

```
<VirtualHost *:80>ServerName staging-mautic.example.com…</VirtualHost
><IfModule mod_ssl.c><VirtualHost *:443>ServerName staging-mautic.exa
mple.com…SSLCertificateFile /etc/letsencrypt/live/staging-mautic.exam
ple.com/cert.pemSSLCertificateKeyFile /etc/letsencrypt/live/staging-m
autic.example.com/privkey.pemSSLCertificateChainFile /etc/letsencrypt
/live/staging-mautic.example.com/chain.pem</VirtualHost></IfModule>
```

After making all of the changes above, restart the Apache server with the following commands.

Ubuntu: sudo service apache2 reload

CentOS: sudo systemctl restart httpd

Modify site_url in Mautic config on staging server

Using a text editor, change the site_url value in /path/to/mautic/app/config/local.php to reflect the staging site URL. To prevent permissions errors, you should modify this file as the web server user (www-data on Ubuntu, apache on CentOS/RHEL) using the command (replacing user in both cases below with the appropriate web server user):

sudo -u username nano /path/to/mautic/app/config/local.php

Next, clear the Mautic cache so that the staging site URL takes effect.

sudo -u username php /path/to/mautic/bin/console cache:clear

Test the Mautic update on the staging server

On the staging server, run the following commands to perform a Mautic update to the latest stable version, and bring the database schema into alignment with the latest version. If prompted whether to proceed at the doctrine migration or schema update steps, respond ‘y’ for yes (replacing user in all cases below with the appropriate web server user).

> Note: if you are updating to a release other than Stable (e.g. beta release) you will need to change the stability setting in the /path/to/mautic/app/config/local.php file. Look for 'update_stability' and change to the relevant setting (e.g. 'update_stability' => 'beta'). If you haven't previously saved the configuration settings in Mautic (settings > configuration) you may not see these settings until you save in the user interface for the first time.

sudo -u username php /path/to/mautic/bin/console mautic:update:find sudo -u username php /path/to/mautic/bin/console mautic:update:apply sudo -u username php /path/to/mautic/bin/console doctrine:migration:status sudo -u username php /path/to/mautic/bin/console doctrine:migration:migrate sudo -u username php /path/to/mautic/bin/console doctrine:schema:update --dump-

sql sudo -u username php /path/to/mautic/bin/console doctrine:schema:update sudo -u username php /path/to/mautic/bin/console cache:clear

Try the latest version of Mautic on the staging server

If you got the message in the console that Mautic has been updated to the latest version and you cleared the cache one last time after all of the commands in the previous step, you're ready to test the latest version of Mautic.

Visit the staging Mautic URL in your web browser, login with your existing credentials, and verify the bottom right hand corner of the dashboard reflects a newer version of Mautic.

You should also make sure that no server errors are displayed by clicking on the "gear" icon in the top right-hand corner of the dashboard, then selecting System Info. Then, navigate to Error Log to ensure nothing unexpected is displayed there.

If you can navigate around the pages of the Mautic dashboard and check that features such as the visual editor are working properly, you may be ready to attempt an upgrade on your production server - after taking a backup, of course.

If you wanted to re-enable cron jobs (by uncommenting the lines in the crontab file) on the staging server to see the outgoing emails that would be generated by Mautic with the new version running in production, you would need to modify the email gateway settings to a test SMTP server such as MailHog, to prevent any live emails from actually being deployed.

Please note this does not capture other channels such as push notification, browser notifications and text messages.

Otherwise, if there are errors, consider visiting the Mautic Community Forum for advice and troubleshooting tips from other community members, or hiring a Mautic support specialist or consultant to repair any server configuration issues before attempting the update on your live instance.

Online URL:
https://kb.mautic.org/article/how-to-set-up-a-staging-server-to-test-new-mautic-releases.html

SyntaxHighlighter.config.stripBrs=true; SyntaxHighlighter.all();