# Configuring email for notifications

**In this article**

To make it easy for users to respond quickly to activity on GitHub Enterprise Server, you can configure your GitHub Enterprise Server instance to send email notifications for issue, pull request, and commit comments.

> **Who can use this feature**
> Site administrators can configure email for notifications.

## Configuring SMTP for your enterprise

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.
4. In the "Settings" sidebar, click **Email**.
5. Select **Enable email**. This will enable both outbound and inbound email. However, for inbound email to work you will also need to configure your DNS settings as described below in "Configuring DNS and firewall settings to allow incoming emails."
6. Type the settings for your SMTP server.
   - In the **Server address** field, type the address of your SMTP server.
   - In the **Port** field, type the port that your SMTP server uses to send email.
   - In the **Domain** field, type the domain name that your SMTP server will send with a HELO response, if any.
   - Select the **Authentication** dropdown, and choose the type of encryption used by your SMTP server.
   - In the **No-reply email address** field, type the email address to use in the From and To fields for all notification emails.
7. If you want to discard all incoming emails that are addressed to the no-reply email address, select **Discard email addressed to the no-reply email address**.
8. Under **Support**, select a type of link to offer additional support to your users.

   - **Email:** An internal email address.
   - **URL:** A link to an internal support site. You must include either `http://` or `https://`.

9. Test email delivery.

## Testing email delivery

1. At the top of the **Email** section, click **Test email settings**.
2. Under "Send test email to," type an address to send the test email to.
3. Click **Send test email**.

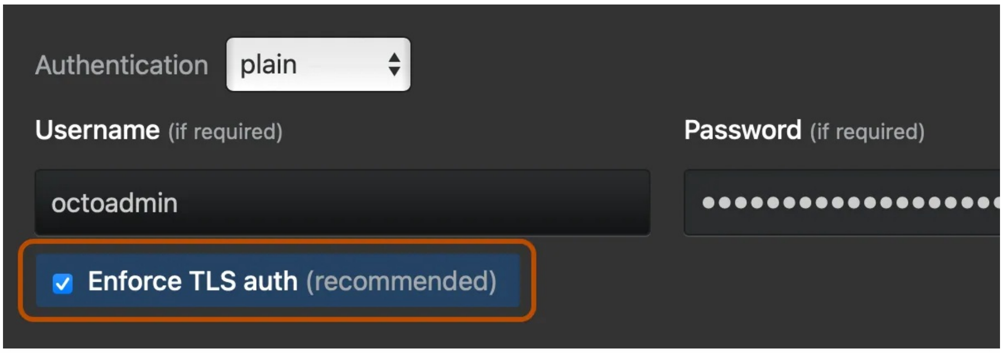
   > **Tip:** If SMTP errors occur while sending a test email—such as an immediate delivery failure or an outgoing mail configuration error—you will see them in the Test email settings dialog box.

4. If the test email fails, troubleshoot your email settings.
5. When the test email succeeds, under the "Settings" sidebar, click **Save settings**.
6. Wait for the configuration run to complete.

## Enforcing TLS for SMTP connections

You can enforce TLS encryption for all incoming SMTP connections, which can help satisfy an ISO-27017 certification requirement.

1. From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click 🚀.
2. If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
3. In the "🚀 Site admin" sidebar, click **Management Console**.
4. In the "Settings" sidebar, click **Email**.
5. Under "Authentication", select **Enforce TLS auth (recommended)**.



Authentication plain

Username (if required) octoadmin

Password (if required) ••••••••••••••••••

☑ Enforce TLS auth (recommended)

6. Under the "Settings" sidebar, click **Save settings**.

> **Note:** Saving settings in the Management Console restarts system services, which could result in user-visible downtime.

7. Wait for the configuration run to complete.

## Configuring DNS and firewall settings to allow incoming emails

If you want to allow email replies to notifications, you must configure your DNS settings.

1. Ensure that port 25 on the instance is accessible to your SMTP server.
2. Create an A record that points to `reply.[hostname]`. Depending on your DNS provider and instance host configuration, you may be able to instead create a single A record that points to `*.[hostname]`.
3. Create an MX record that points to `reply.[hostname]` so that emails to that domain are routed to the instance.
4. Create an MX record that points `noreply.[hostname]` to `[hostname]` so that replies to the `cc` address in notification emails are routed to the instance. For more information, see "Configuring notifications."

## Troubleshooting email delivery

### Create a support bundle

If you cannot determine what is wrong from the displayed error message, you can download a support bundle containing the entire SMTP conversation between your mail server and GitHub Enterprise Server. Once you've downloaded and extracted the bundle, check the entries in `enterprise-manage-logs/unicorn.log` for the entire SMTP conversation log and any related errors.

The unicorn log should show a transaction similar to the following:

```
This is a test email generated from https://10.0.0.68/setup/settings
Connection opened: smtp.yourdomain.com:587
-> "220 smtp.yourdomain.com ESMTP nt3sm2942435pbc.14\r\n"
<- "EHLO yourdomain.com\r\n"
-> "250-smtp.yourdomain.com at your service, [1.2.3.4]\r\n"
-> "250-SIZE 35882577\r\n"
-> "250-8BITMIME\r\n"
-> "250-STARTTLS\r\n"
-> "250-ENHANCEDSTATUSCODES\r\n"
-> "250 PIPELINING\r\n"
<- "STARTTLS\r\n"
-> "220 2.0.0 Ready to start TLS\r\n"
TLS connection started
<- "EHLO yourdomain.com\r\n"
-> "250-smtp.yourdomain.com at your service, [1.2.3.4]\r\n"
-> "250-SIZE 35882577\r\n"
-> "250-8BITMIME\r\n"
-> "250-AUTH LOGIN PLAIN XOAUTH\r\n"
-> "250-ENHANCEDSTATUSCODES\r\n"
-> "250 PIPELINING\r\n"
<- "AUTH LOGIN\r\n"
-> "334 VXNlcm5hbWU6\r\n"
```

```
<- "dGhpc2lzbXlAYWRkcmVzcy5jb20=\r\n"
-> "334 UGFzc3dvcmQ6\r\n"
<- "aXRyZWFsbHl3YXM=\r\n"
-> "535-5.7.1 Username and Password not accepted. Learn more at\r\n"
-> "535 5.7.1 http://support.yourdomain.com/smtp/auth-not-accepted
nt3sm2942435pbc.14\r\n"
```

This log shows that the appliance:

- Opened a connection with the SMTP server (`Connection opened: smtp.yourdomain.com:587`).
- Successfully made a connection and chose to use TLS (`TLS connection started`).
- The `login` authentication type was performed (`<- "AUTH LOGIN\r\n"`).
- The SMTP Server rejected the authentication as invalid (`-> "535-5.7.1 Username and Password not accepted.`).

## Check your GitHub Enterprise Server instance logs

If you need to verify that your inbound email is functioning, you can review `/var/log/mail.log` and `/var/log/mail-replies/metroplex.log` on your instance.

`/var/log/mail.log` verifies that messages are reaching your server. Here's an example of a successful email reply:

```
Oct 30 00:47:18 54-171-144-1 postfix/smtpd[13210]: connect from st11p06mm-
asmtp002.mac.com[17.172.124.250]
Oct 30 00:47:19 54-171-144-1 postfix/smtpd[13210]: 51DC9163323: client=st11p06mm-
asmtp002.mac.com[17.172.124.250]
Oct 30 00:47:19 54-171-144-1 postfix/cleanup[13216]: 51DC9163323: message-id=
<b2b9c260-4aaa-4a93-acbb-0b2ddda68579@me.com>
Oct 30 00:47:19 54-171-144-1 postfix/qmgr[17250]: 51DC9163323: from=
<tcook@icloud.com>, size=5048, nrcpt=1 (queue active)
Oct 30 00:47:19 54-171-144-1 postfix/virtual[13217]: 51DC9163323: to=<reply+i-1-
1801beb4df676a79250d1e61e54ab763822c207d-5@reply.ghe.tjl2.co.ie>, relay=virtual,
delay=0.12, delays=0.11/0/0/0, dsn=2.0.0, status=sent (delivered to maildir)
Oct 30 00:47:19 54-171-144-1 postfix/qmgr[17250]: 51DC9163323: removed
Oct 30 00:47:19 54-171-144-1 postfix/smtpd[13210]: disconnect from st11p06mm-
asmtp002.mac.com[17.172.124.250]
```

Note that the client first connects; then, the queue becomes active. Then, the message is delivered, the client is removed from the queue, and the session disconnects.

`/var/log/mail-replies/metroplex.log` shows whether inbound emails are being processed to add to issues and pull requests as replies. Here's an example of a successful message:

```
[2014-10-30T00:47:23.306 INFO (5284) #] metroplex: processing <b2b9c260-4aaa-
4a93-acbb-0b2ddda68579@me.com>
[2014-10-30T00:47:23.333 DEBUG (5284) #] Matched
/data/user/mail/reply/new/1414630039.Vfc00I12000eM445784.ghe-tjl2-co-ie
[2014-10-30T00:47:23.334 DEBUG (5284) #] Moving
/data/user/mail/reply/new/1414630039.Vfc00I12000eM445784.ghe-tjl2-co-ie =>
/data/user/incoming-mail/success
```

You'll notice that `metroplex` catches the inbound message, processes it, then moves the file over to `/data/user/incoming-mail/success`.

## Verify your DNS settings

In order to properly process inbound emails, you must configure a valid A Record (or CNAME), as well as an MX Record. For more information, see "Configuring DNS and firewall settings to allow incoming emails."

## Check firewall or AWS security group settings

If your GitHub Enterprise Server instance is behind a firewall or is being served through an AWS security group, make sure port 25 is open to all mail servers that send emails to `reply@reply.[hostname]`.

## Contact support

If you're still unable to resolve the problem, contact us by visiting GitHub Enterprise Support. Please attach the output file from `http(s)://[hostname]/setup/diagnostics` to your email to help us troubleshoot your problem.