

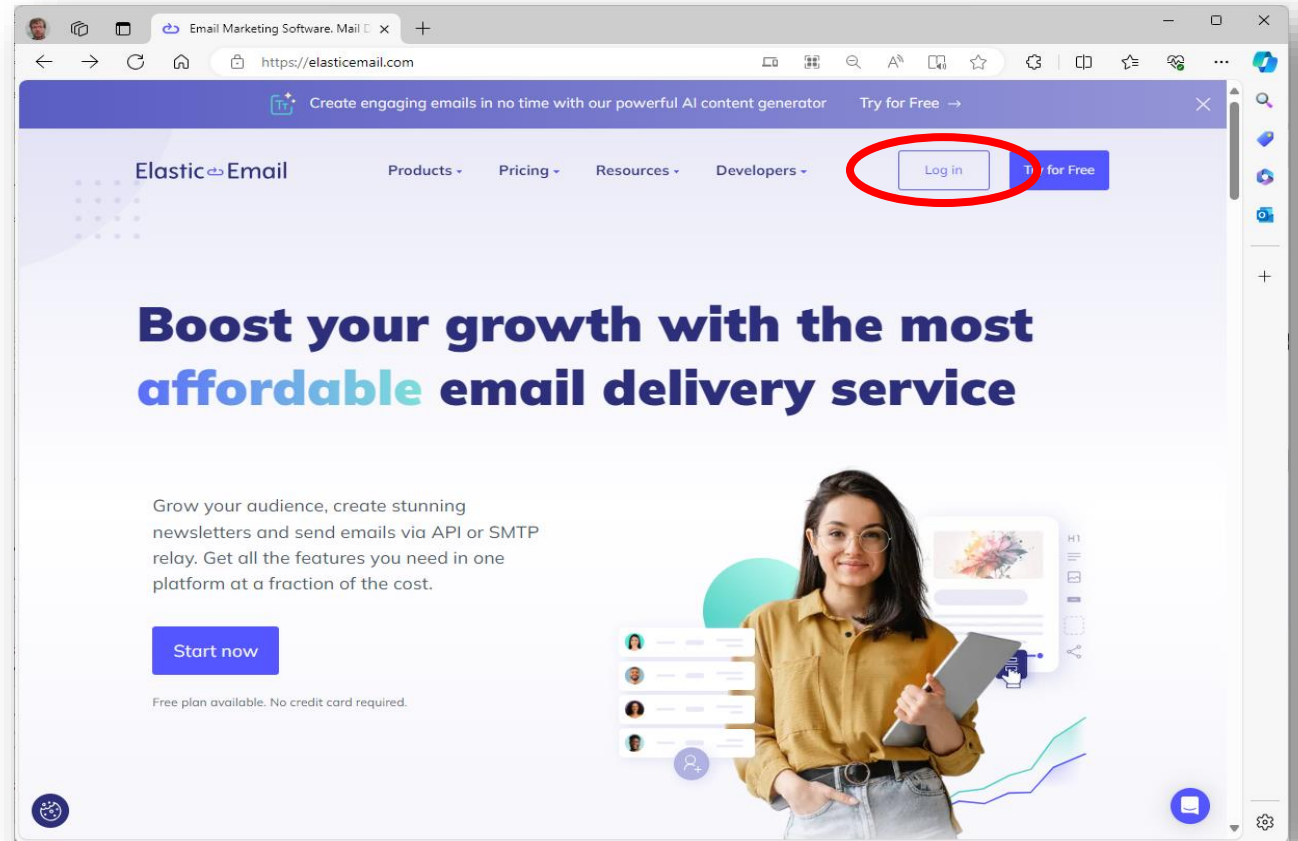
SuniTAFE

ICTPRG302 Apply introductory
programming techniques

**Send email from Python
using AWS-Cloud9, SMTP and elasticemail.com**

elasticemail.com – Login

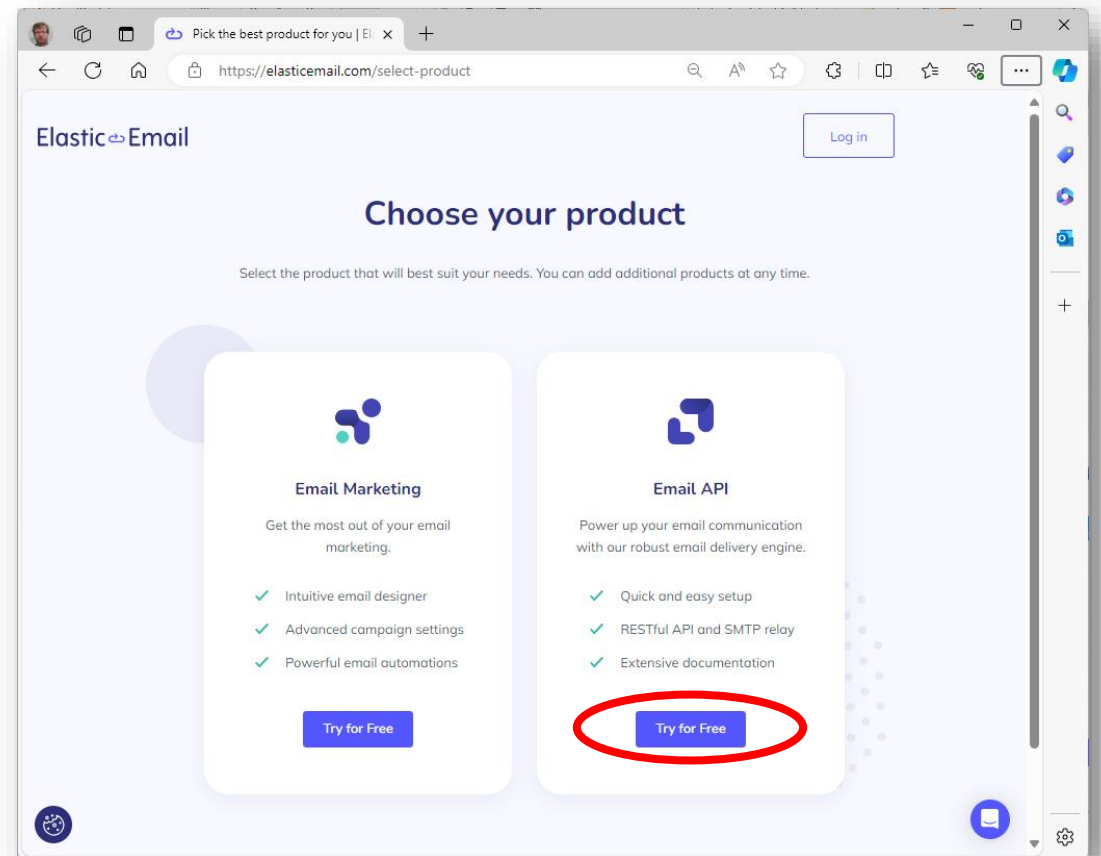
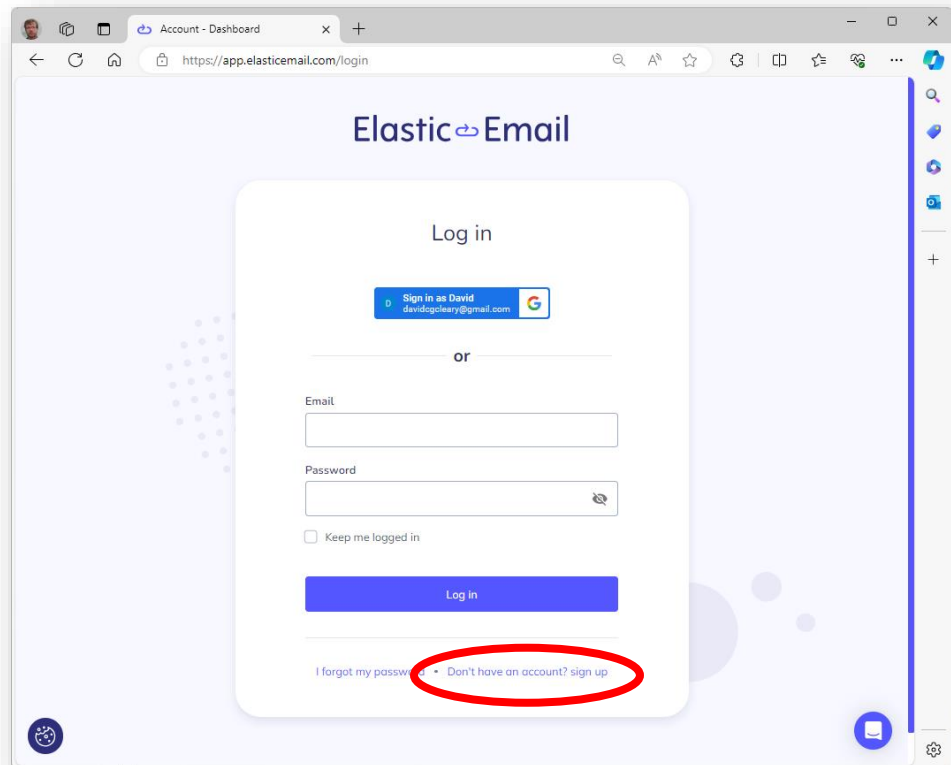
1. *Before you continue, ensure that you have a Gmail account*
2. Go to, <https://elasticemail.com>
3. Select, [Log in]



elasticemail.com – Login

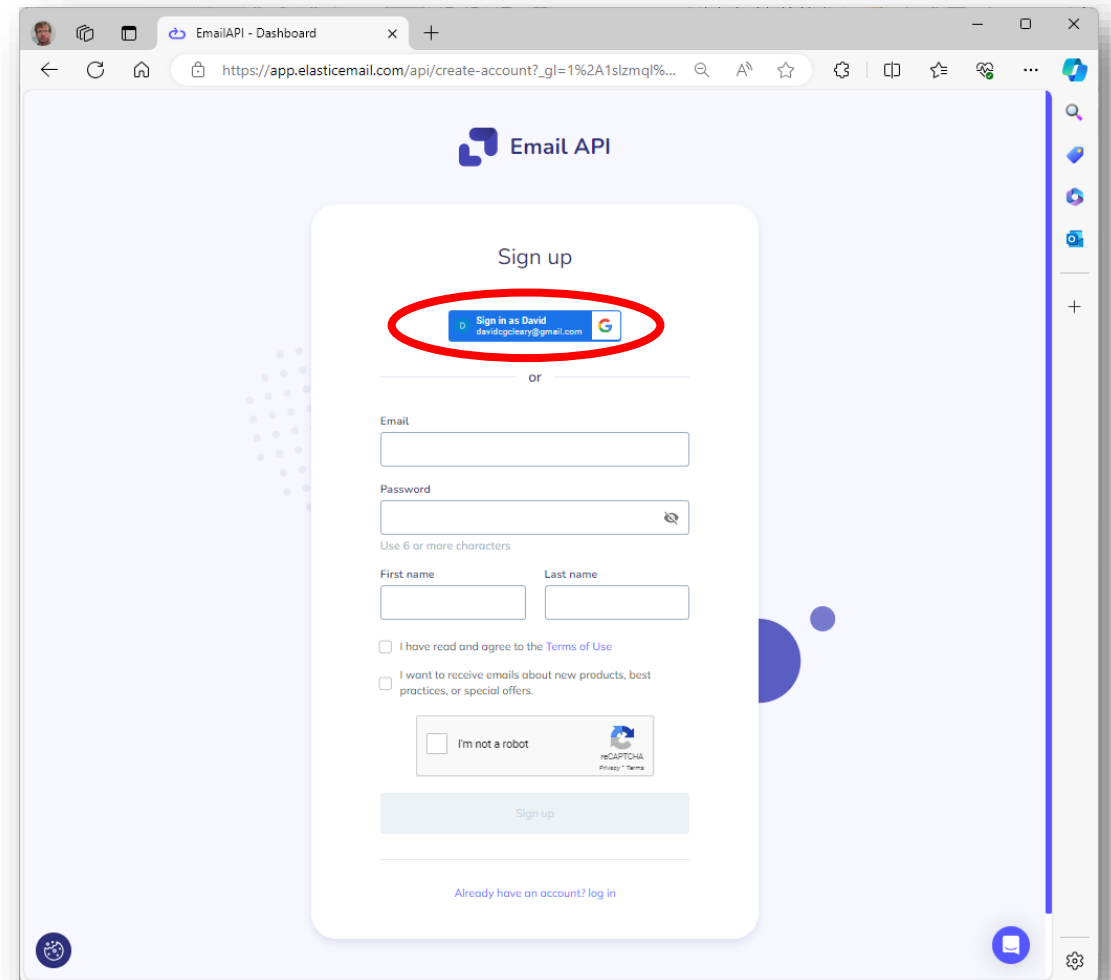
4. Select, **[Don't have an account? sign up]**

5. Select, **Email API | [Try for Free]**



elasticemail.com – Login

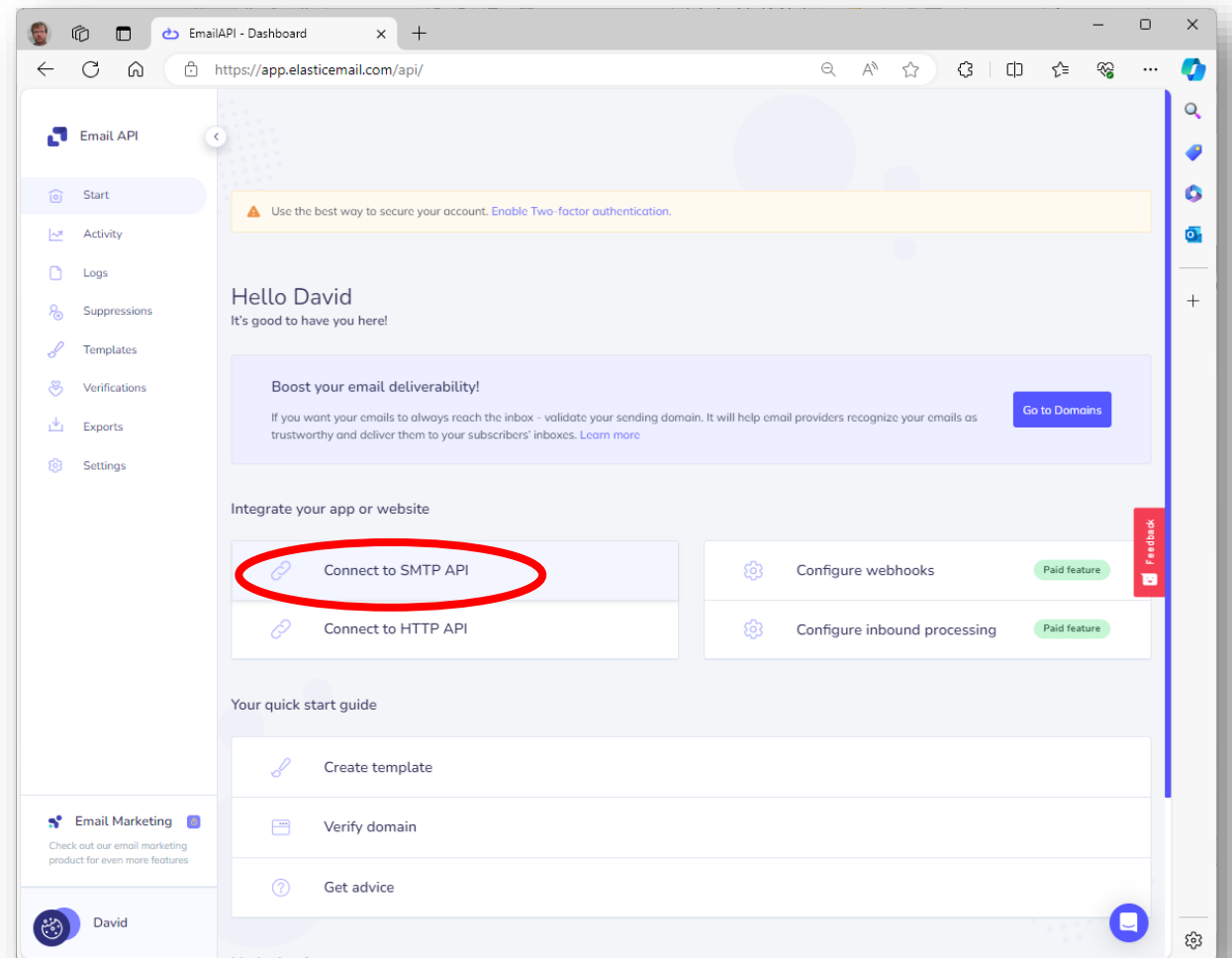
6. Sign up using your Gmail account



The screenshot shows the 'Email API' sign-up page. The 'Sign in as David' button, which includes a Google icon and the email 'davidcgc@icloud.com', is circled in red. Below this button is an 'or' separator, followed by input fields for 'Email', 'Password', 'First name', and 'Last name'. There are also checkboxes for 'I have read and agree to the Terms of Use' and 'I want to receive emails about new products, best practices, or special offers.' A reCAPTCHA widget is present with the text 'I'm not a robot'. At the bottom of the form is a 'Sign up' button and a link for 'Already have an account? log in'.

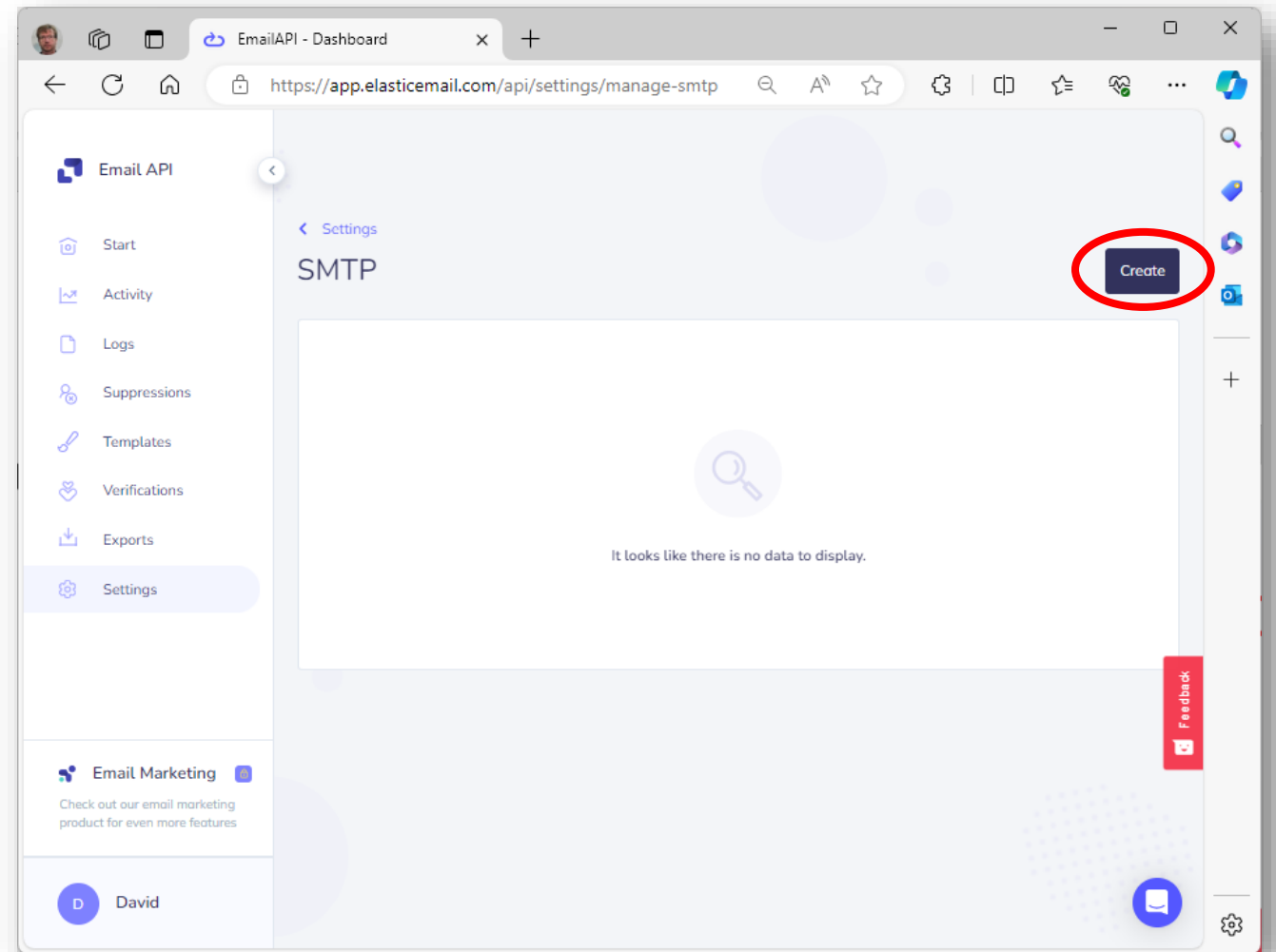
elasticemail.com – Manage SMTP settings

7. Select, **[Connect to SMTP API]**



elasticemail.com – Manage SMTP settings

8. Select, **[Create]**



elasticemail.com – Manage SMTP settings

9. For **Username**, enter your ***Gmail email address***
10. Select, [**Set access restrictions**]
11. For **Expires**, enter ***31 Dec 2024***
12. Select, [**Create**]

The screenshot shows the 'EmailAPI - Dashboard' in a web browser. The URL is <https://app.elasticemail.com/api/settings/create-smtp>. The page title is 'SMTP'. On the left sidebar, the 'Settings' menu item is highlighted. The main content area contains the following form elements:

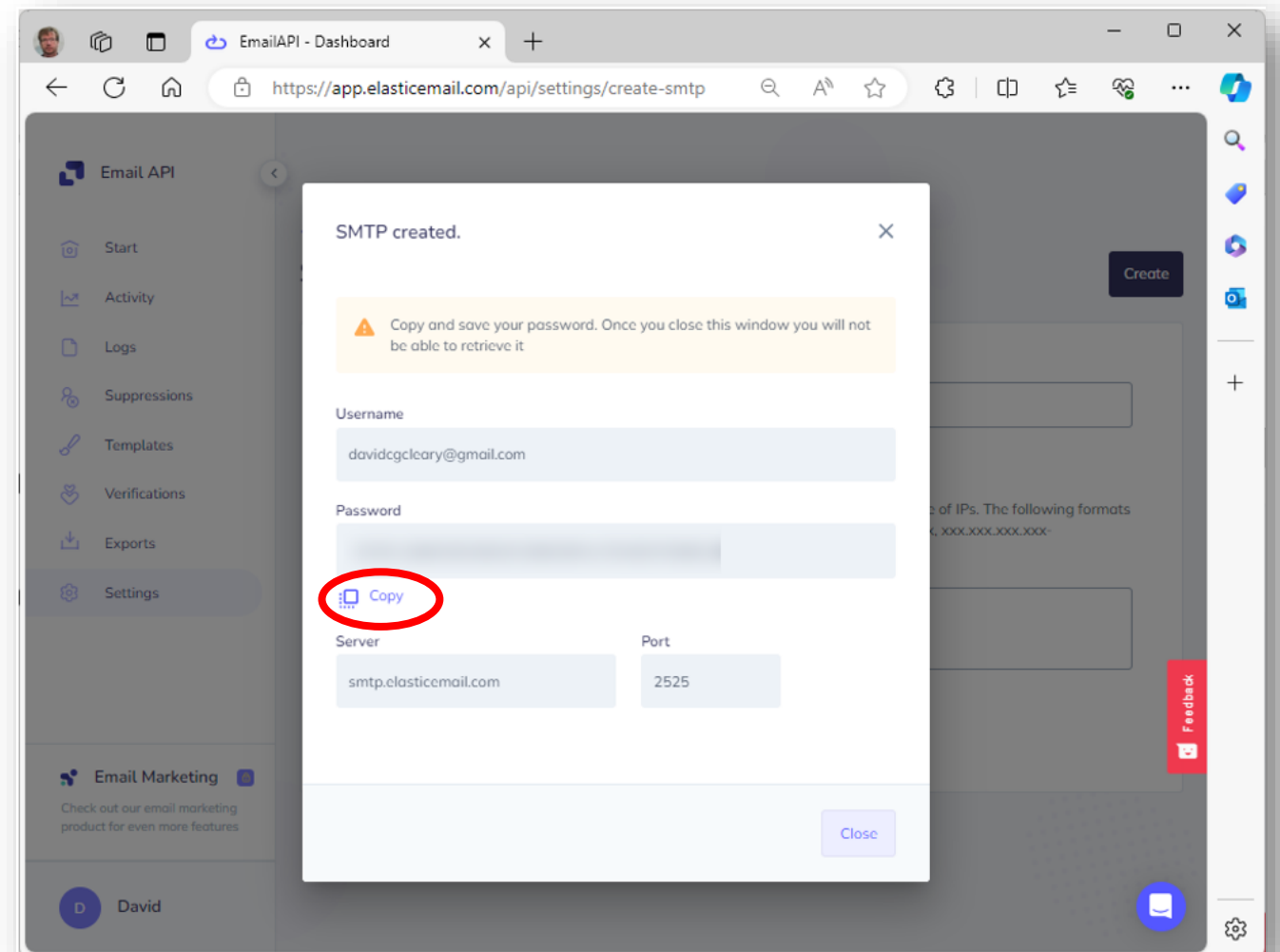
- Username:** A text input field containing 'davidcgcleary@gmail.com'.
- Set access restriction (optional):** A link that has been selected, highlighted with a red circle.
- Expires:** A date input field containing '31 Dec 2024'.
- Create:** A blue button in the top right corner, highlighted with a red circle.

Below the 'Set access restriction' link, there is explanatory text: 'To further secure your connection you can optionally limit access to a single IP or range of IPs. The following formats are supported and delimited by a comma for multiple IPs or IP Ranges: xxx.xxx.xxx.xxx, xxx.xxx.xxx.xxx-yyy.yyy.yyy.yyy, xxx.xxx.xxx.xxx/xx and IPv6.' Below this text is an empty text area for input.

elasticemail.com – Manage SMTP settings

13. Select, **[Copy]** to copy and save your password

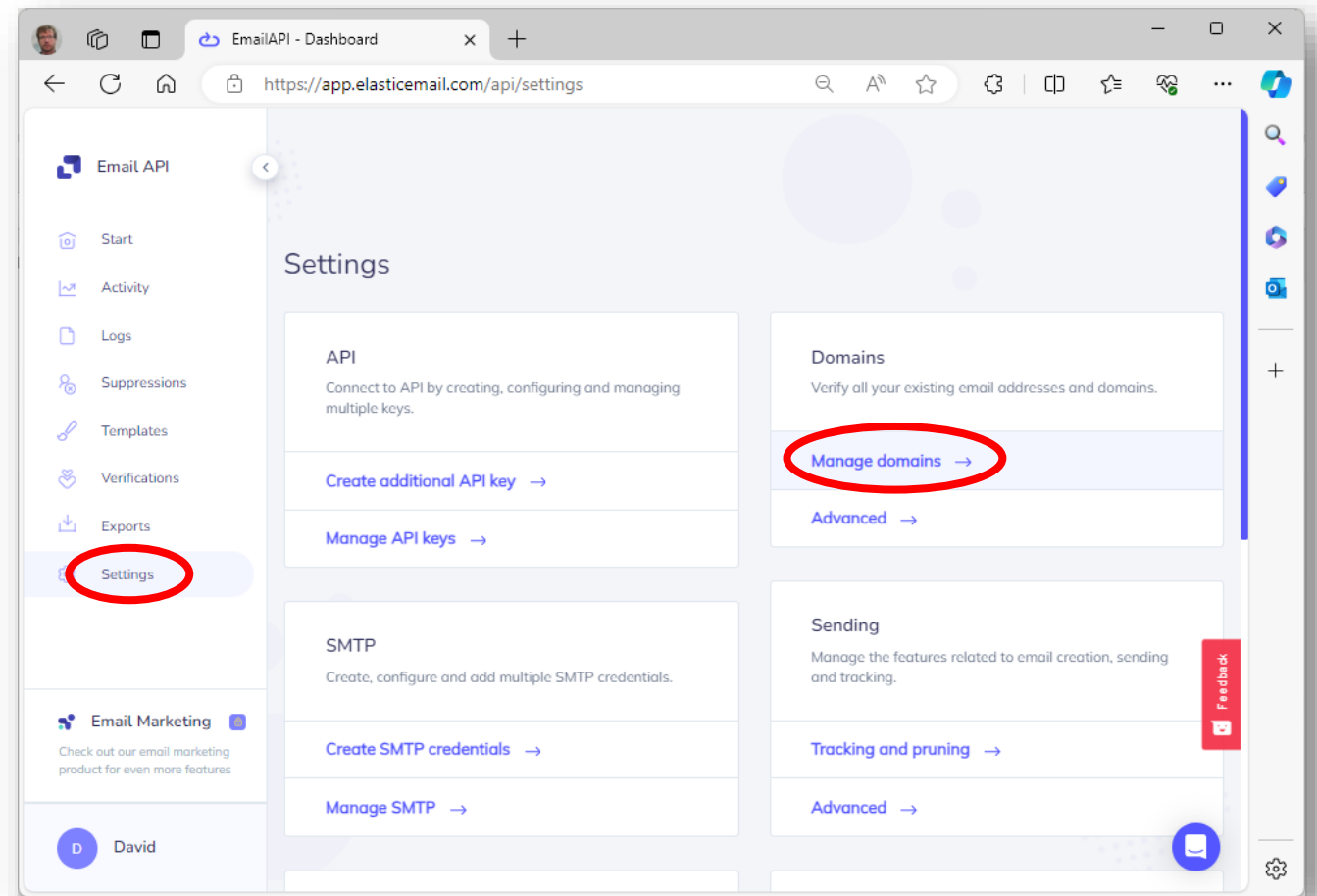
Once you close this window you will not be able to retrieve the password again.



elasticemail.com – Verify sender email address

14. Select, **[Settings]**

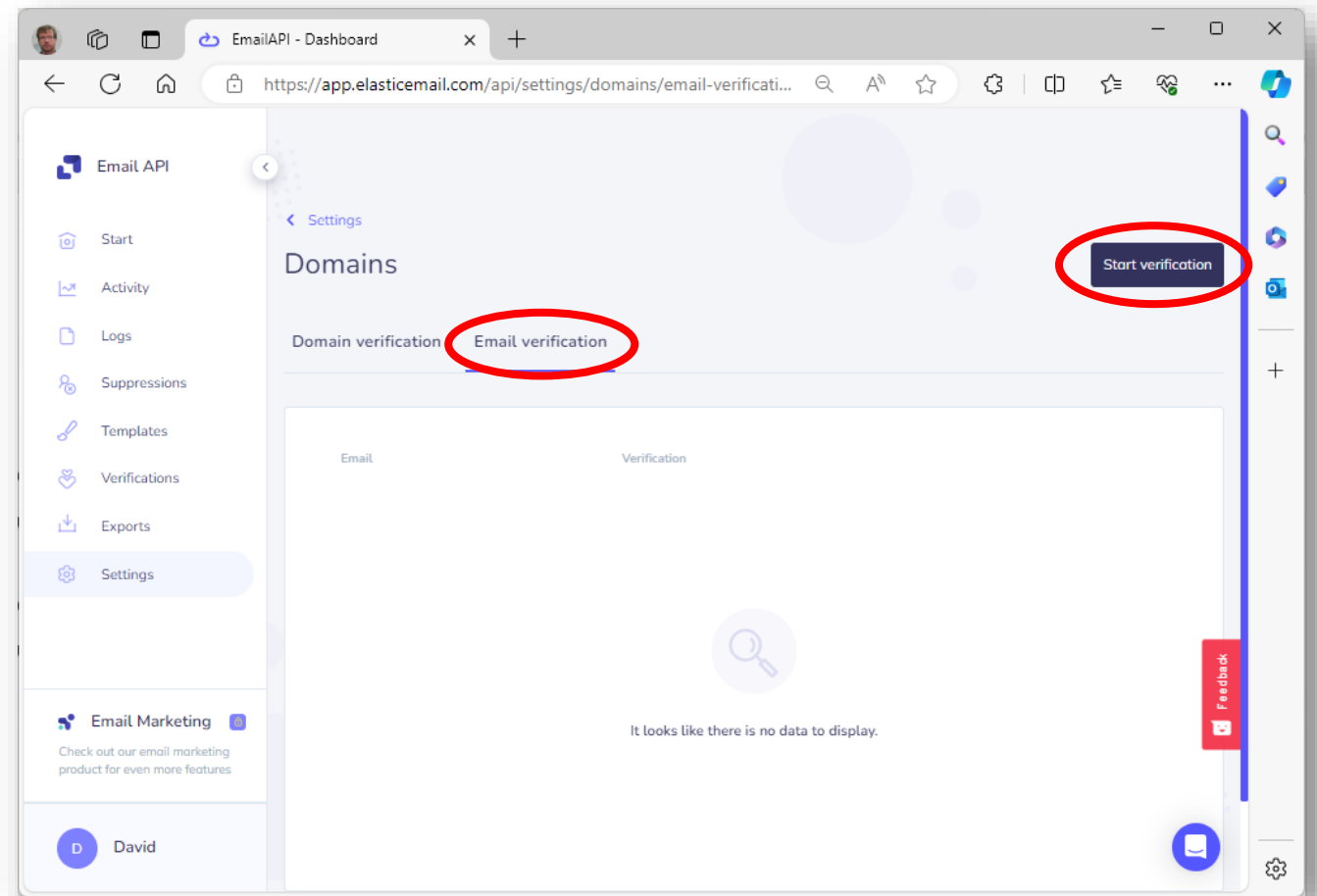
15. Select **[Manage domains]**



elasticemail.com – Verify sender email address

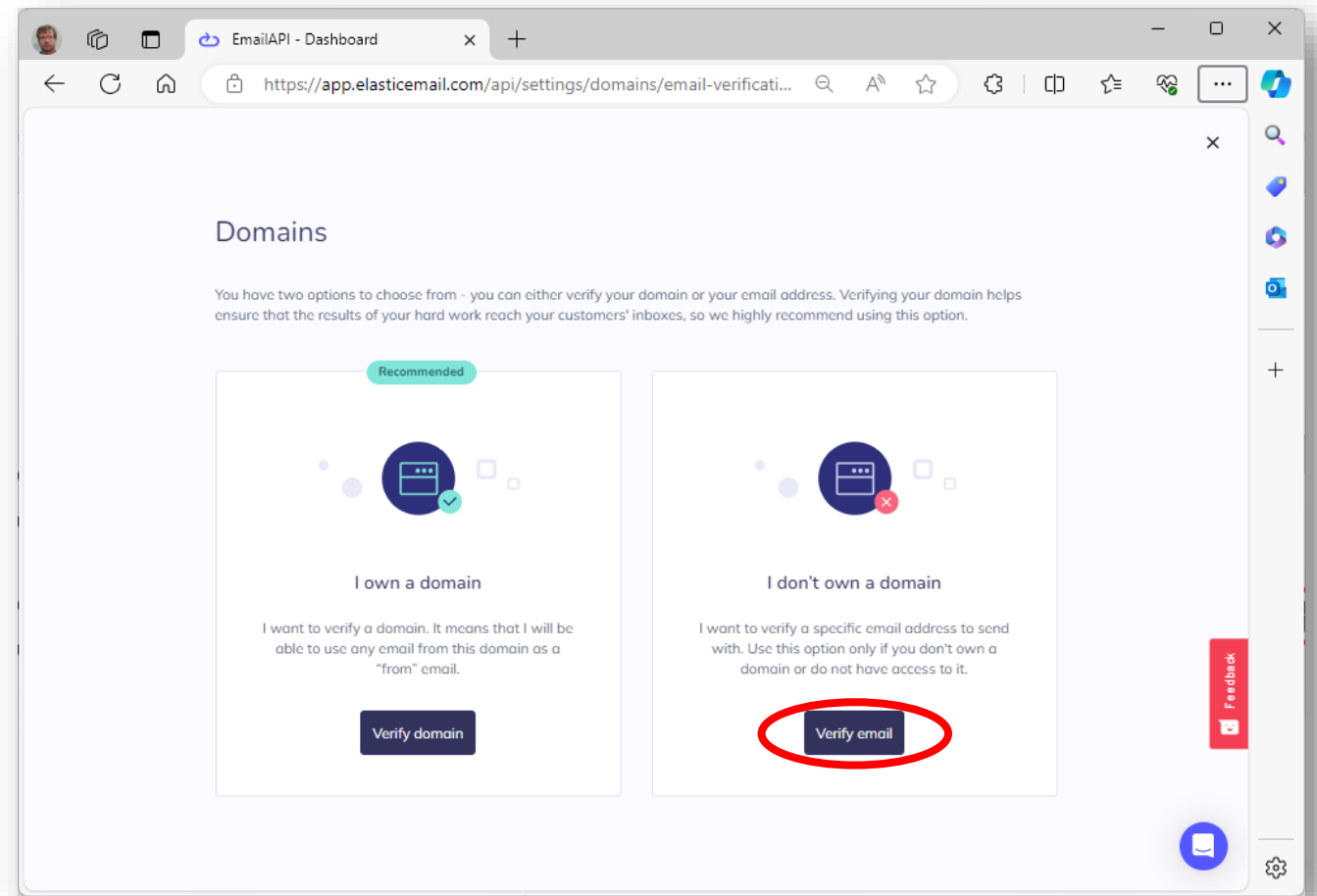
16. Select, [**Email verification**]

17. Select [**Start verification**]



elasticemail.com – Verify sender email address

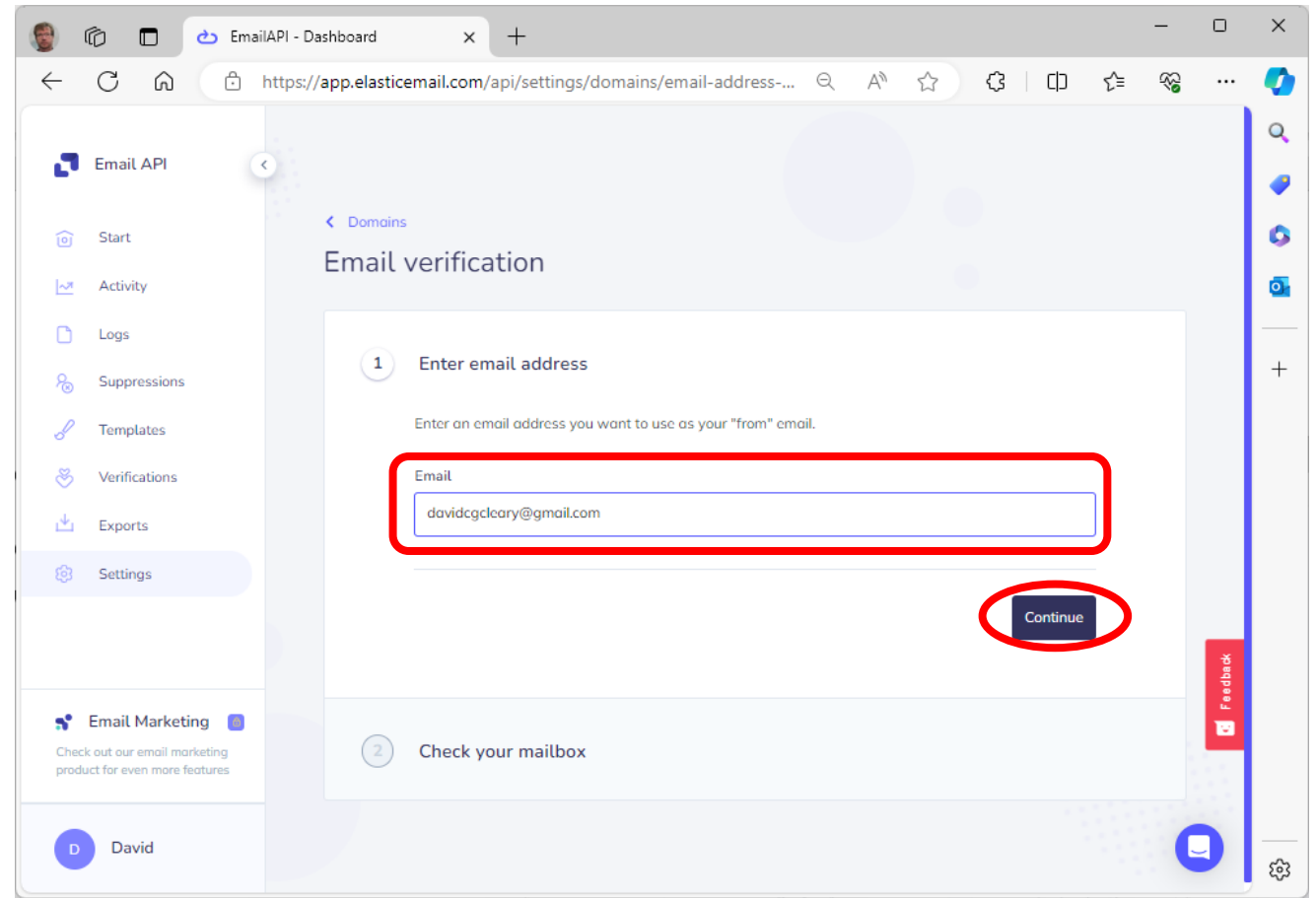
18. Select, **[Verify email]**



elasticemail.com – Verify sender email address

19. For **Email**, enter your
Gmail email address

20. Select **[Continue]**

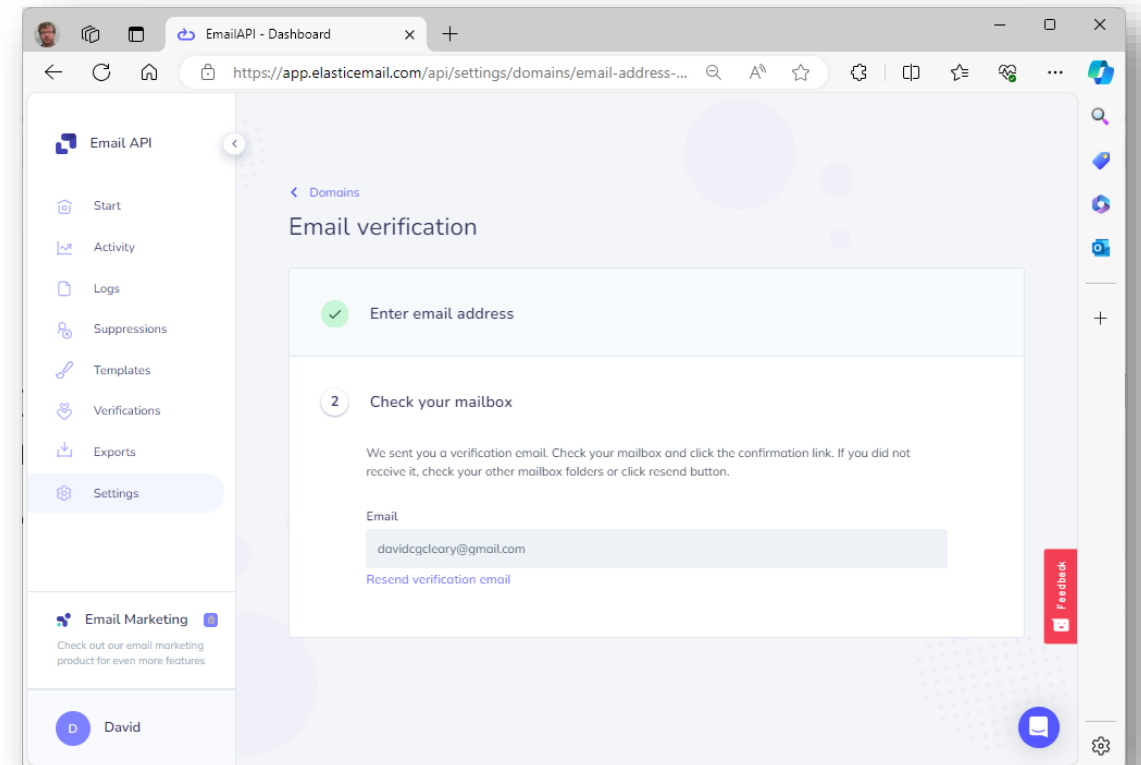
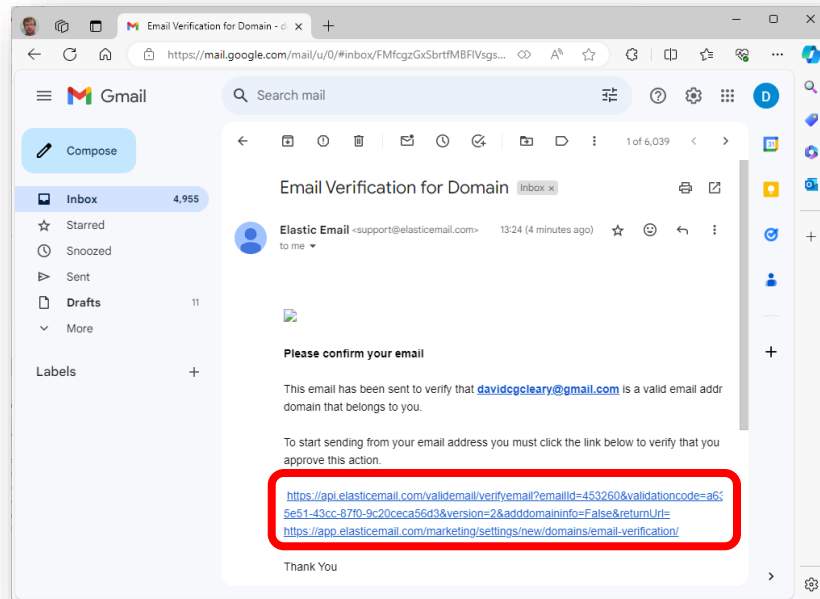


The screenshot shows the EmailAPI dashboard in a web browser. The left sidebar contains navigation links: Email API, Start, Activity, Logs, Suppressions, Templates, Verifications, Exports, and Settings (highlighted). Below these are links for Email Marketing and a user profile for David. The main content area is titled 'Email verification' and shows a two-step process. Step 1, 'Enter email address', is active and contains a text input field with the email 'davidcgcleary@gmail.com' and a 'Continue' button. Both the input field and the button are circled in red. Step 2, 'Check your mailbox', is partially visible below. The browser's address bar shows the URL 'https://app.elasticemail.com/api/settings/domains/email-address-...'.

elasticemail.com – Verify sender email address

21. Check your Gmail inbox for an email with subject **Email Verification for Domain**

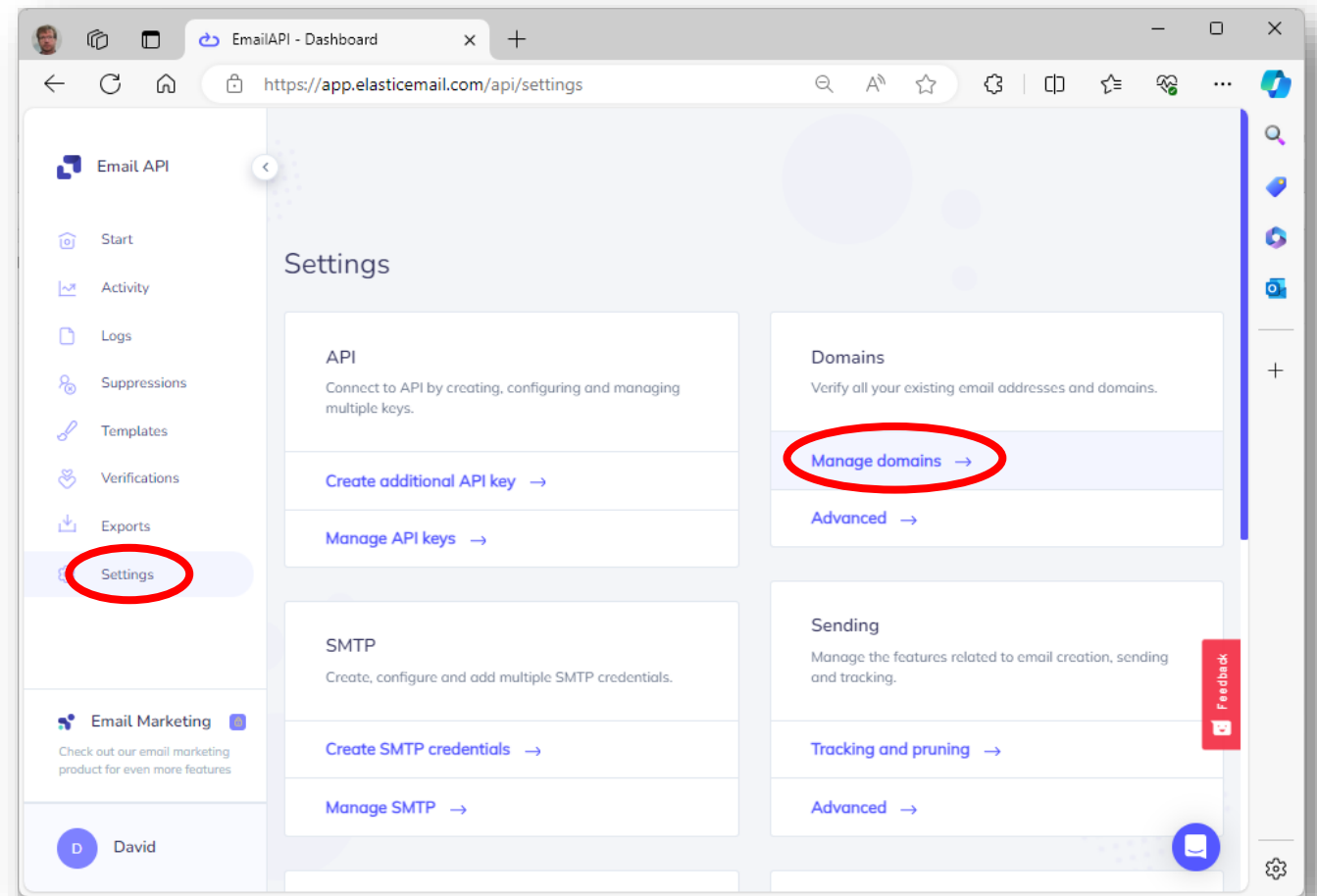
22. Click on **the link** in the email to verify the sender email address



elasticemail.com – Verify sender email address

23. Select, **[Settings]**

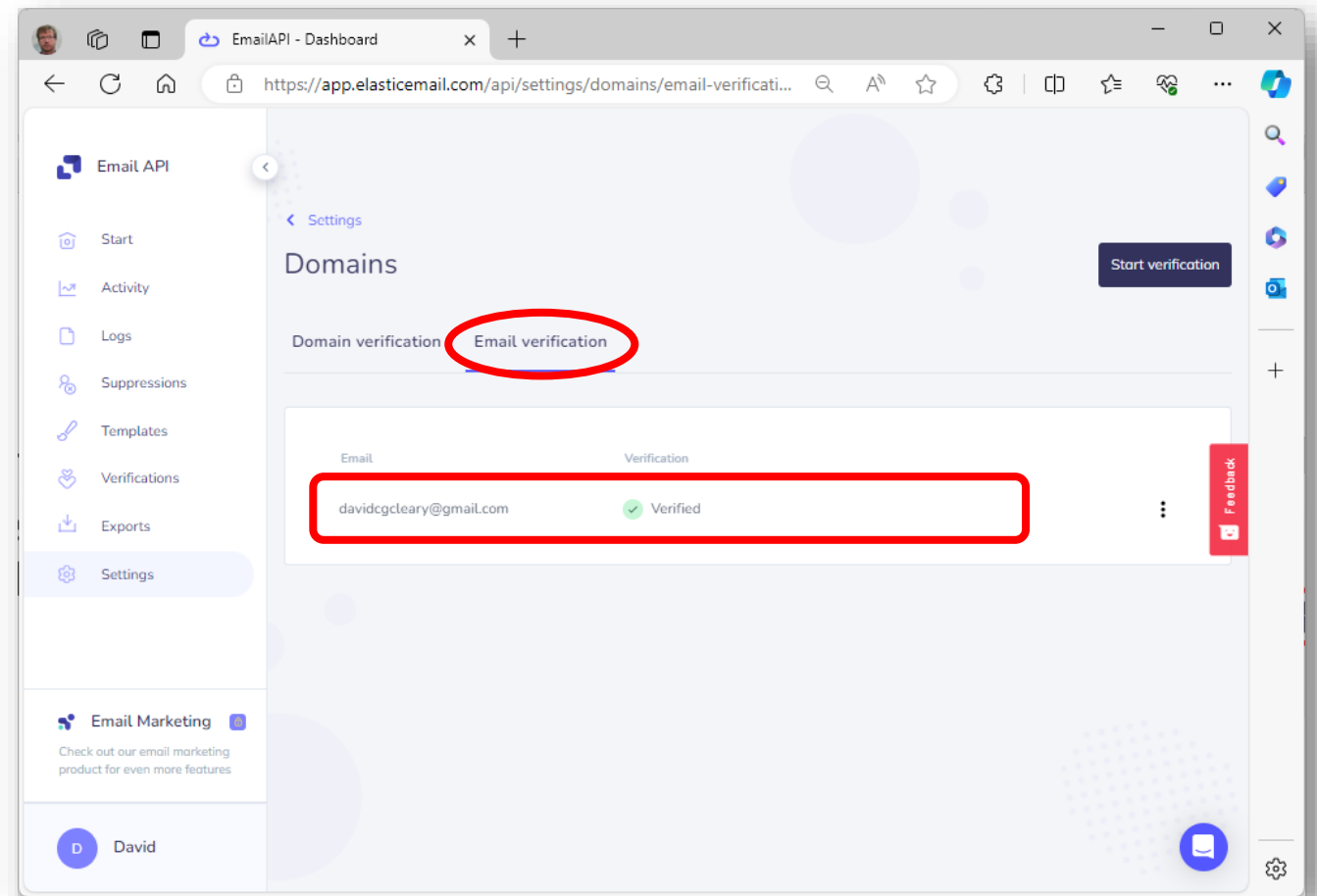
24. Select, **[Manage domains]**



elasticemail.com – Verify sender email address

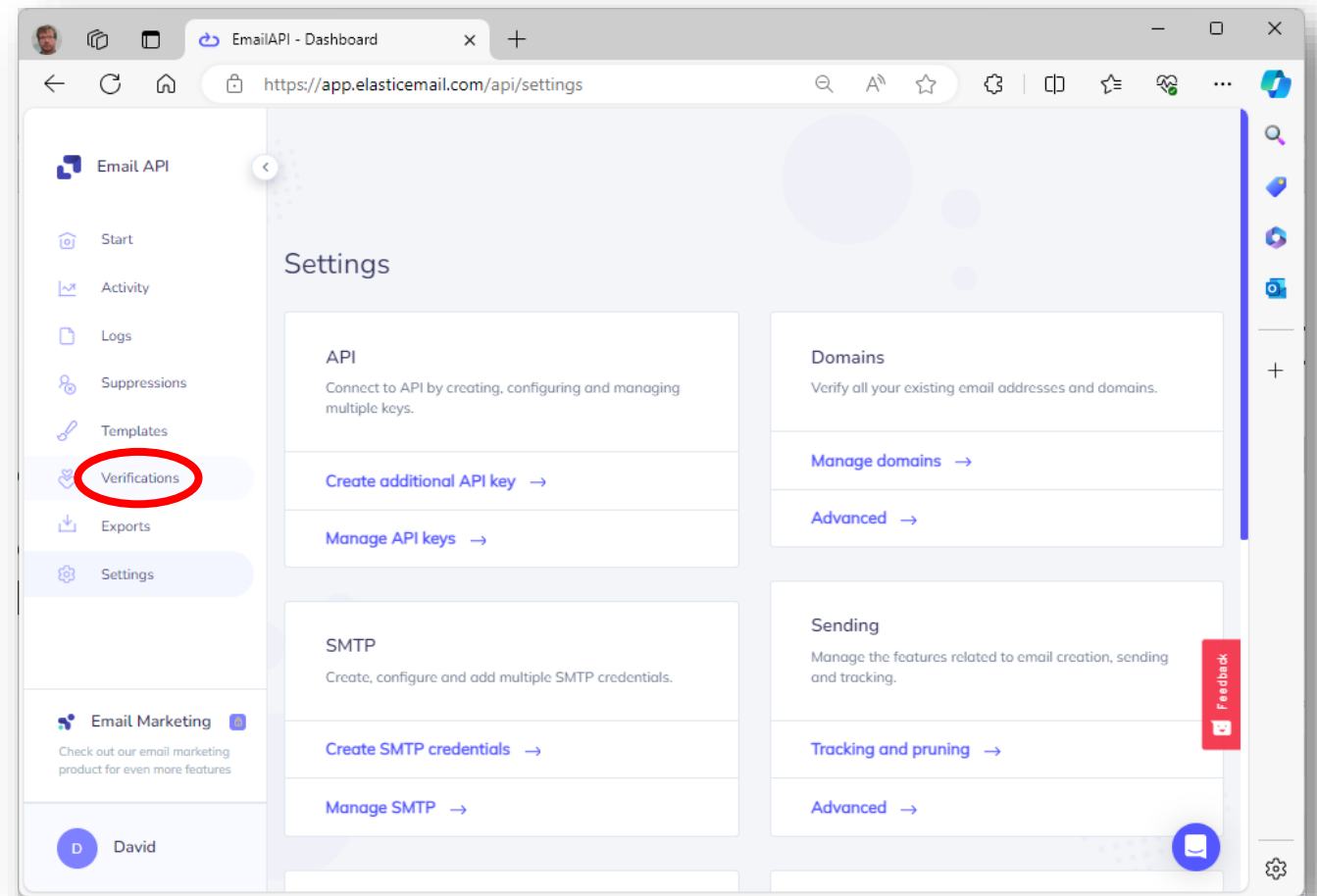
25. Select, [**Email verification**]

26. The verified sender email address is displayed



elasticemail.com – Verify recipient email address

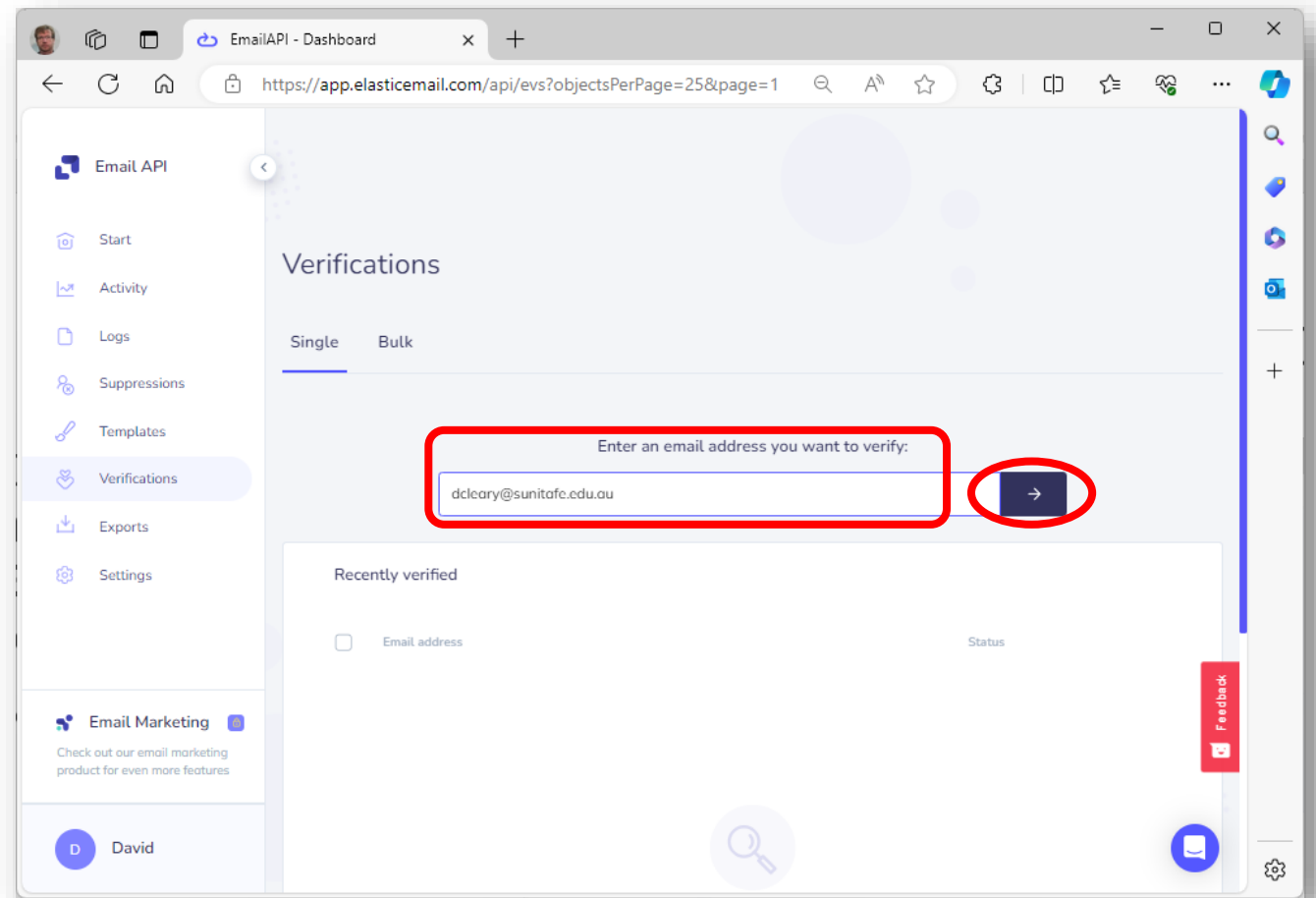
27. Select **[Verifications]**



elasticemail.com – Verify recipient email address

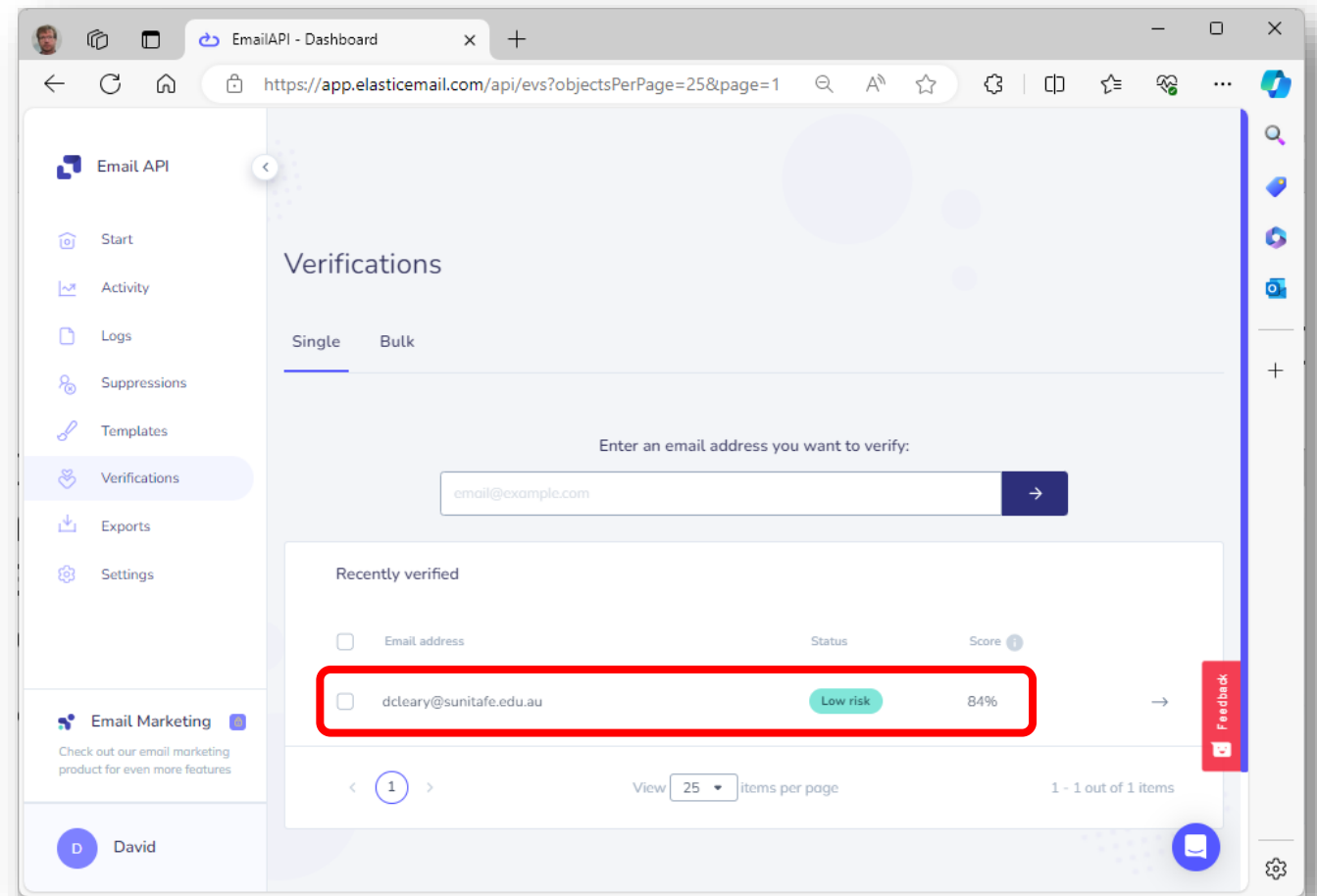
28. For **Enter an Email address you want to verify**, enter your *SuniTAFE email address*

29. Select [→]



elasticemail.com – Verify recipient email address

30. The verified recipient email address is displayed



Python – Send email using smtp.elasticemail.com

31. Define SMTP settings in backupcfg.py

```
# SMTP settings
smtp = {"sender":      "davidcgcleary@gmail.com", # elasticemail.com verified sender
        "recipient":  "dcleary@sunitafe.edu.au", # elasticemail.com verified recipient
        "server":     "smtp.elasticemail.com",   # elasticemail.com SMTP server
        "port":       2525,                      # elasticemail.com SMTP port
        "user":        "davidcgcleary@gmail.com", # elasticemail.com user
        "password":    "AAAABBBBXXXXYYYYZZZZ"}   # elasticemail.com password
```

Python – Send email using smtp.elasticemail.com

32. Define sendEmail() function in backup.py

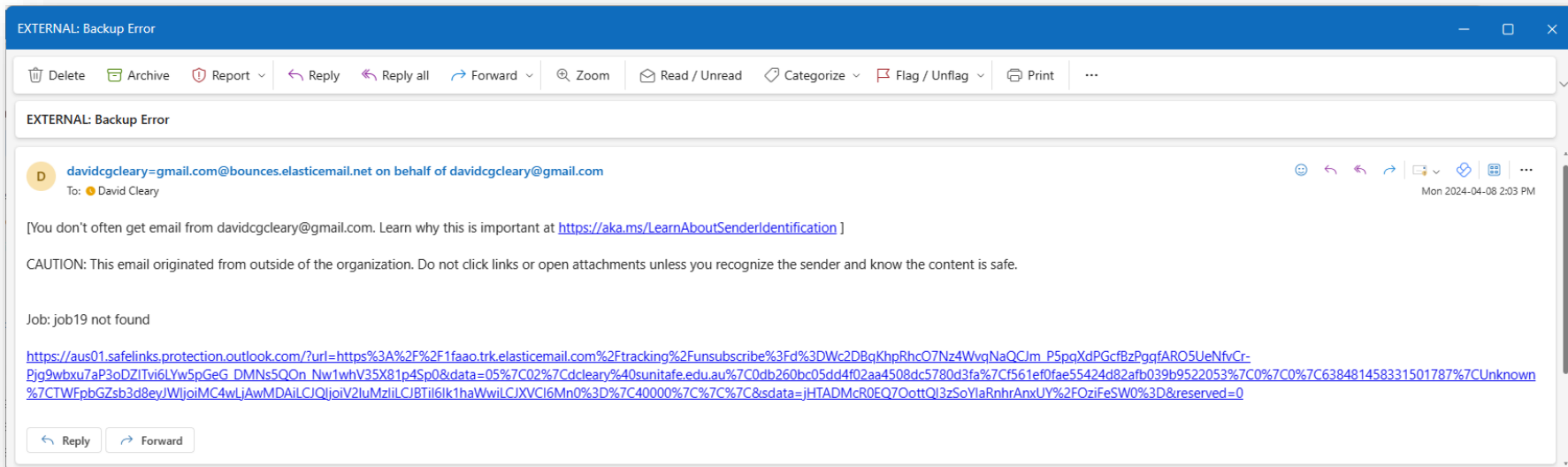
```
def sendEmail(message, dateTimeStamp):  
    """  
    Send an email message to the specified recipient.  
    Parameters:  
        message (string): message to send.  
        dateTimeStamp (string): Date and time when program was run.  
    """  
    # create email message  
    email = 'To: ' + smtp["recipient"] + '\n' + 'From: ' + smtp["sender"] +  
            '\n' + 'Subject: Backup Error\n\n' + message + '\n'  
  
    # connect to email server and send email  
    try:  
        smtp_server = smtplib.SMTP(smtp["server"], smtp["port"])  
        smtp_server.ehlo()  
        smtp_server.starttls()  
        smtp_server.ehlo()  
        smtp_server.login(smtp["user"], smtp["password"])  
        smtp_server.sendmail(smtp["sender"], smtp["recipient"], email)  
        smtp_server.close()  
    except Exception as e:  
        print("ERROR: Send email failed: " + str(e), file=sys.stderr)
```

Python – Send email using smtp.elasticemail.com

33. Run backup.py program, to send email

```
bash - "ip-172-31 x"
dclearly:~/environment/ictprg302-Python (main) $ backup.py job19
ERROR: Job: job19 not found
dclearly:~/environment/ictprg302-Python (main) $
```

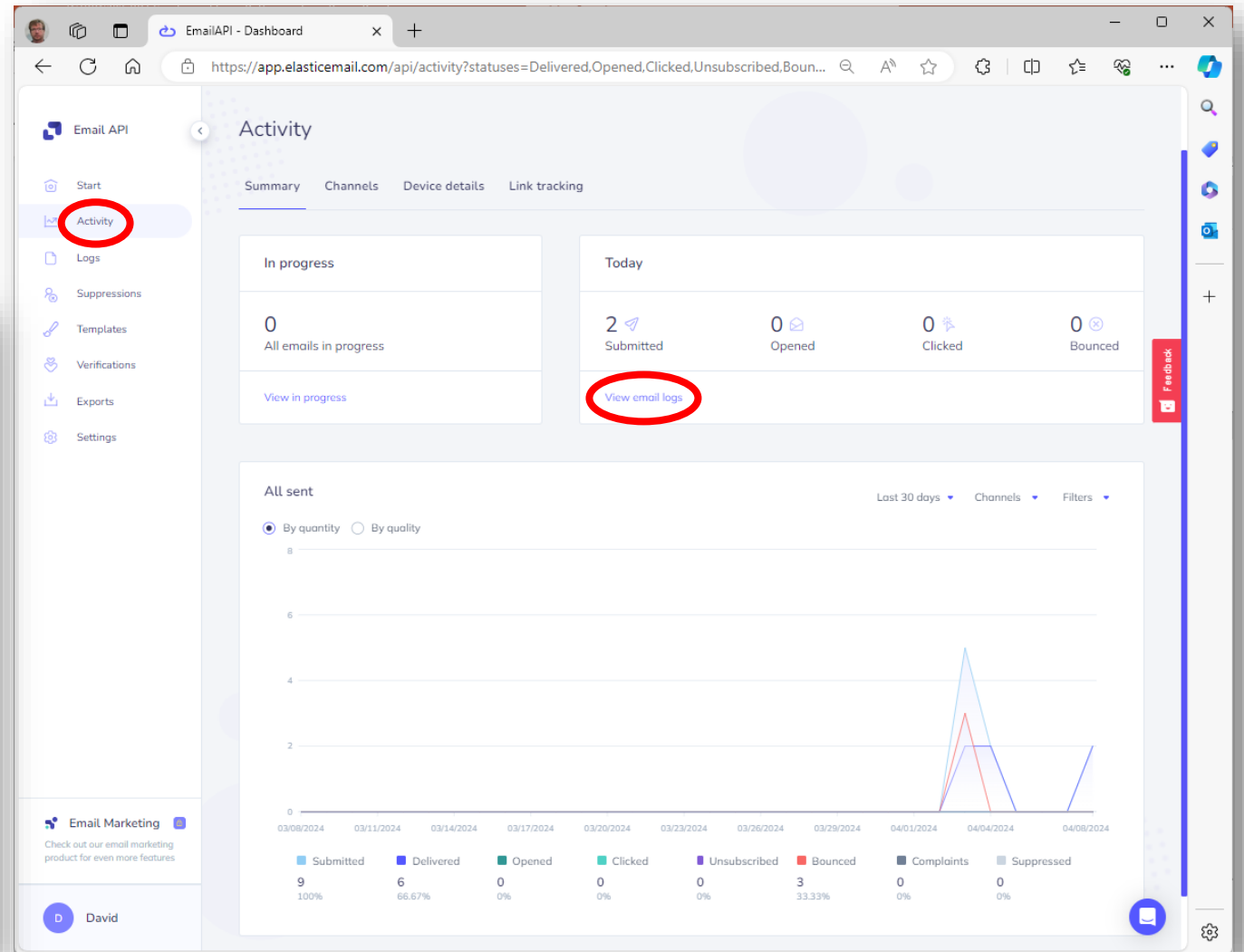
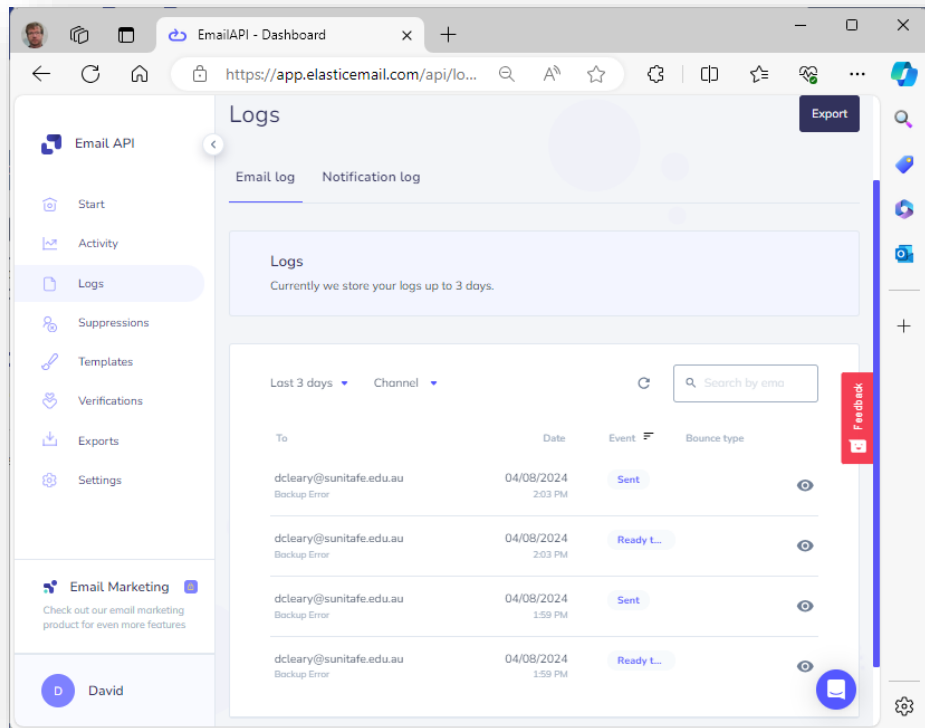
34. View the received email



elasticemail.com – Report emails sent

35. Select, **[Activity]**

36. Select, **[View email logs]**



Python – Delete password details

It is imperative that you delete the elasticmail.com password details from your Python code before you upload that code to github.com.

If your github.com repository is public, the elasticmail.com password could be accessed and used to send spam emails from your account.

31. Modify SMTP settings in backupcfg.py

```
# SMTP settings
smtp = {"sender":      "davidcgcleary@gmail.com", # elasticmail.com verified sender
        "recipient":  "dcleary@sunitafe.edu.au", # elasticmail.com verified recipient
        "server":     "smtp.elasticmail.com",    # elasticmail.com SMTP server
        "port":       2525,                      # elasticmail.com SMTP port
        "user":        "davidcgcleary@gmail.com", # elasticmail.com user
        "password":    ""}                      # elasticmail.com password
```