```
def test_sign_in_with_valid_credentials_and_email_mfa(monkeypatch):
    """
    Tests successful sign in using valid credentials with EMAIL MFA, using a mocked IMAP email code.
    """
    from signIn import sign_in
    from seleniumrequests import Chrome
    from selenium.webdriver.common.by import By
    class DummyElement:
        def __init__(self):
            self._text = "EMAIL Code"
        def clear(self): pass
        def send_keys(self, val): pass
        def click(self): pass
        @property
        def text(self):
            return self._text
        def find_element(self, by, selector):
            return self
    class DummyWebDriver(Chrome):
        def __init__(self):
            self.current_url = 'https://mint.intuit.com/'
        def implicitly_wait(self, val): pass
        def get(self, url): self.current_url = url
        def find_element(self, by, value): return DummyElement()
    def mock_get_email_code(*args, **kwargs): return '123456'
    monkeypatch.setattr('signIn.get_email_code', mock_get_email_code)
    d = DummyWebDriver()
    result = sign_in(
        email="test@example.com",
        password="ValidPass!123",
        driver=d,
        mfa_method="EMAIL",
        mfa_token=None,
        mfa_input_callback=None,
        imap_account="acct",
        imap_password="pwd",
        imap_server="imap.example.com"
    )
    assert result is None or isinstance(result, str)
```

---Extra Details---
Filename: signIn.py
Description: Test signing in with valid credentials using EMAIL multi-factor authentication, mocking the IMAP email code retrieval.
Score: 100%
Alignment: 100%
Validation Notes: Checks main sign_in logic with valid email, password, and MFA via email; ensures smooth run with mocked dep