

Student:

Andre Hardy

Email:

ahardy754@email.porterchester.edu

Time on Task:

4 hours, 25 minutes

Progress:

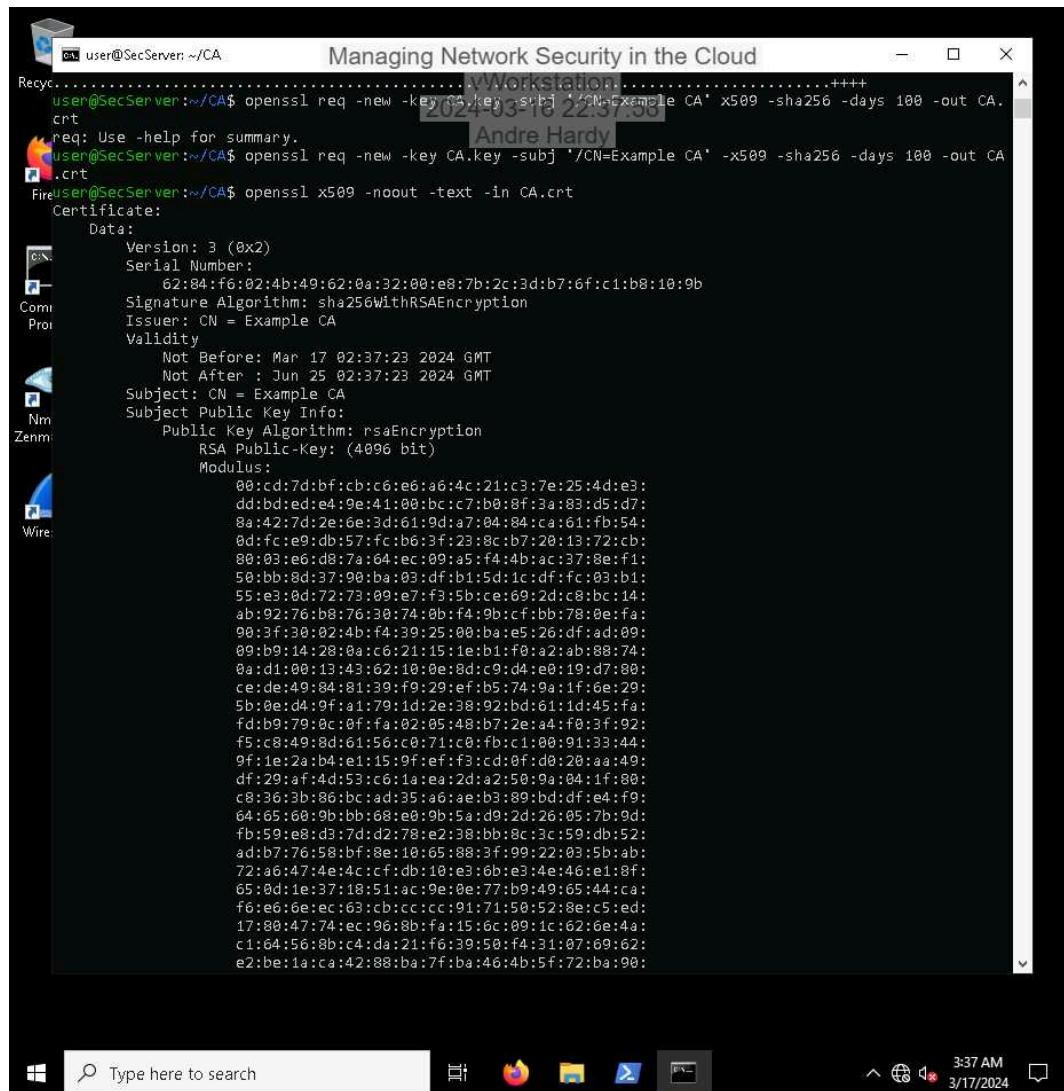
100%

Report Generated: Saturday, March 16, 2024 at 11:18 PM

Hands-On Demonstration

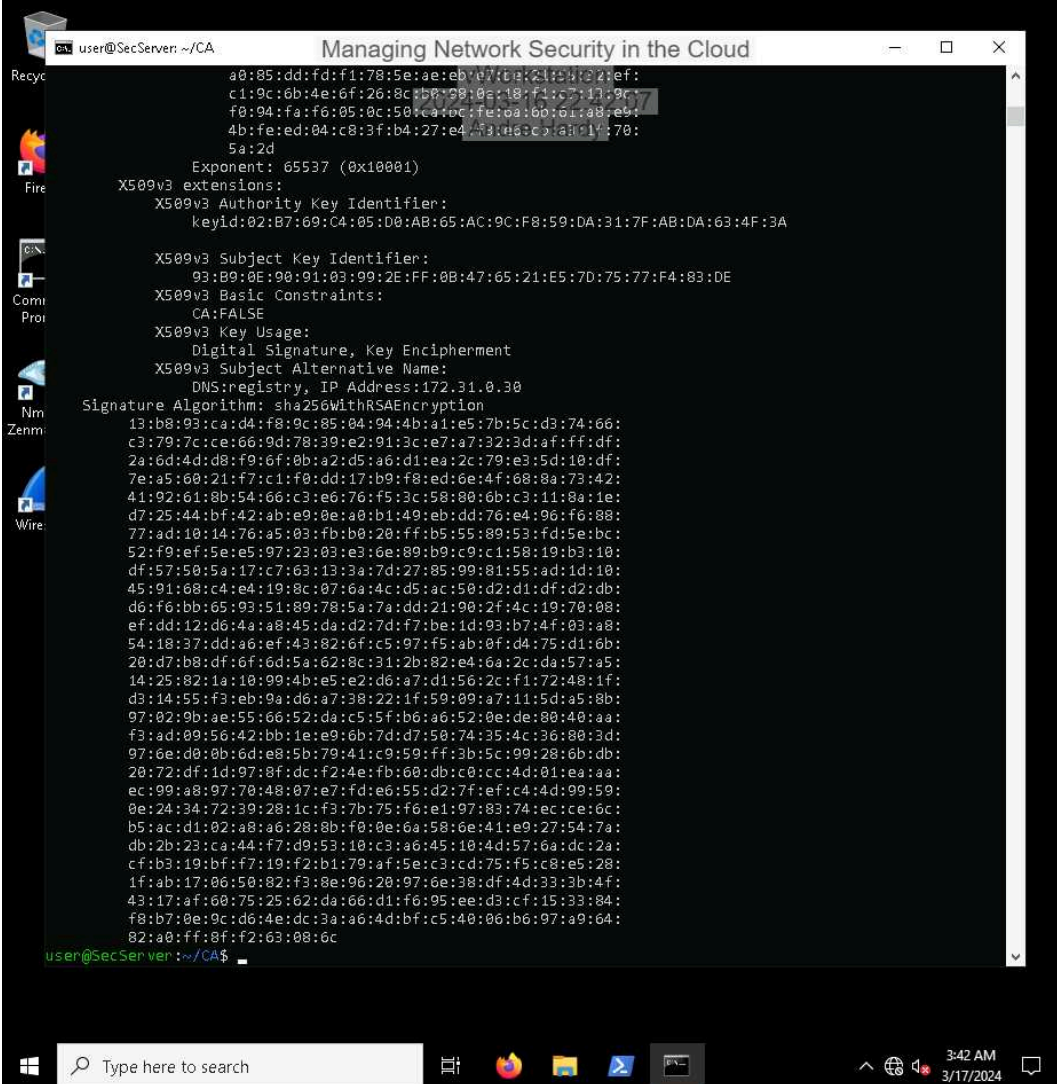
Part 1: Create Certificates for Using Mutual TLS

9. **Make a screen capture** showing at least the first 12 lines of the output for the root certificate.



```
user@SecServer: ~/CA
user@SecServer:~/CA$ openssl req -new -key CA.key -subj '/CN=Example CA' -x509 -sha256 -days 100 -out CA.crt
req: Use -help for summary.
user@SecServer:~/CA$ openssl req -new -key CA.key -subj '/CN=Example CA' -x509 -sha256 -days 100 -out CA.crt
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number:
    62:84:f6:02:4b:49:62:0a:32:00:e8:7b:2c:3d:b7:6f:c1:b8:10:9b
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: CN = Example CA
  Validity
    Not Before: Mar 17 02:37:23 2024 GMT
    Not After : Jun 25 02:37:23 2024 GMT
  Subject: CN = Example CA
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public-Key: (4096 bit)
    Modulus:
      00:cd:7d:bf:cb:c6:e6:a6:4c:21:c3:7e:25:4d:e3:
      dd:bd:ed:e4:9e:41:00:bc:c7:b0:8f:3a:83:d5:d7:
      8a:42:7d:2e:6e:3d:61:9d:a7:04:84:ca:61:fb:54:
      0d:fc:e9:db:57:fc:b6:3f:23:8c:b7:20:13:72:cb:
      80:03:e6:d8:7a:64:ec:09:a5:f4:4b:ac:37:8e:f1:
      50:bb:8d:37:90:ba:03:df:b1:5d:1c:df:fc:03:b1:
      55:e3:0d:72:73:09:e7:f3:5b:ce:69:2d:c8:bc:14:
      ab:92:76:b8:76:30:74:0b:f4:9b:cf:bb:78:0e:fa:
      90:3f:30:02:4b:f4:39:25:00:ba:e5:26:df:ad:09:
      09:b9:14:28:0a:c6:21:15:1e:b1:f0:a2:ab:88:74:
      0a:d1:00:13:43:62:10:0e:8d:c9:d4:e0:19:d7:80:
      ce:de:49:84:81:39:f9:29:ef:b5:74:9a:1f:6e:29:
      5b:0e:d4:9f:a1:79:1d:2e:38:92:bd:61:1d:45:fa:
      fd:b9:79:0c:0f:fa:02:05:48:b7:2e:a4:f0:3f:92:
      f5:c8:49:8d:61:56:c0:71:c0:fb:c1:00:91:33:44:
      9f:1e:2a:b4:e1:15:9f:ef:f3:cd:0f:d0:20:aa:49:
      df:29:af:4d:53:c6:1a:ea:2d:a2:50:9a:04:1f:80:
      c8:36:3b:86:bc:ad:35:a6:ae:b3:89:bd:df:e4:f9:
      64:65:60:9b:bb:68:e0:9b:5a:d9:2d:26:05:7b:9d:
      fb:59:e8:d3:7d:d2:78:e2:38:bb:8c:3c:59:db:52:
      ad:b7:76:58:bf:8e:10:65:88:3f:99:22:03:5b:ab:
      72:a6:47:4e:4c:cf:db:10:e3:6b:e3:4e:46:e1:8f:
      65:0d:1e:37:18:51:ac:9e:0e:77:b9:49:65:44:ca:
      f6:e6:6e:ec:63:cb:cc:cc:91:71:50:52:8e:c5:ed:
      17:80:47:74:ec:96:8b:fa:15:6c:09:1c:62:6e:4a:
      c1:64:56:8b:c4:da:21:f6:39:50:f4:31:07:69:62:
      e2:be:1a:ca:42:88:ba:7f:ba:46:4b:5f:72:ba:90:
```

15. Make a screen capture showing the X509v3 extensions of the registry certificate.



```
user@SecServer: ~/CA
Managing Network Security in the Cloud

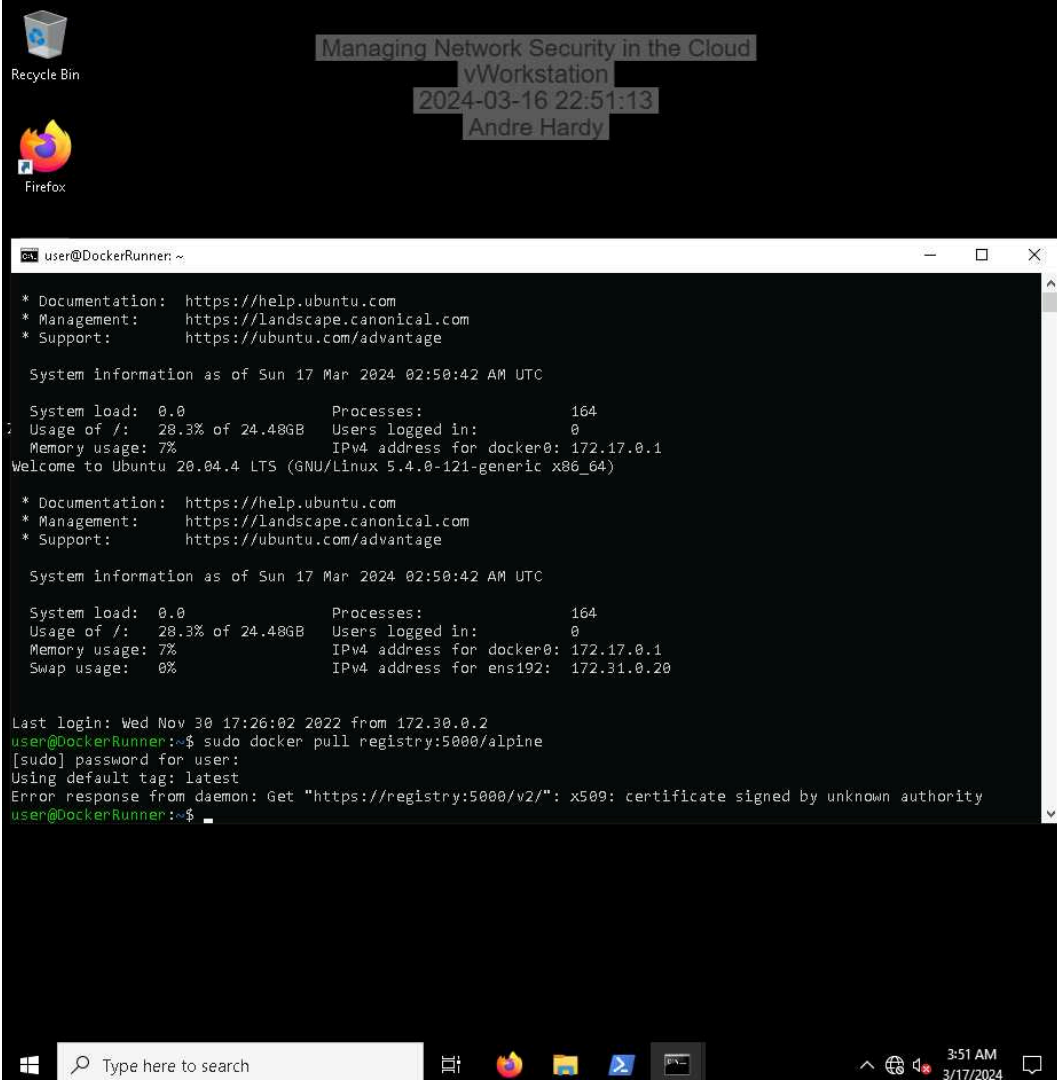
a0:85:dd:fd:f1:78:5e:ae:eb:07:6a:c1:81:02:ef:
c1:9c:6b:4e:6f:26:8c:b0:08:0a:18:f1:c7:11:9c:
f0:94:fa:f6:05:0c:50:ca:ec:fe:0a:60:61:a8:e9:
4b:fe:ed:04:c8:3f:b4:27:e4:f3:ac:c3:a3:14:70:
5a:2d
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Authority Key Identifier:
    keyid:02:B7:69:C4:05:D0:AB:65:AC:9C:F8:59:DA:31:7F:AB:DA:63:4F:3A

X509v3 Subject Key Identifier:
    93:B9:0E:90:91:03:99:2E:FF:0B:47:65:21:E5:7D:75:77:F4:83:DE
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Key Usage:
    Digital Signature, Key Encipherment
X509v3 Subject Alternative Name:
    DNS:registry, IP Address:172.31.0.30
Signature Algorithm: sha256WithRSAEncryption
13:b8:03:ca:d4:f8:9c:85:04:94:4b:a1:e5:7b:5c:d3:74:66:
c3:79:7c:ce:66:9d:78:39:e2:91:3c:e7:a7:32:3d:af:ff:df:
2a:6d:4d:d8:f9:6f:0b:a2:d5:a6:d1:ea:2c:79:e3:5d:10:df:
7e:a5:60:21:f7:c1:f0:dd:17:b9:f8:ed:6e:4f:68:8a:73:42:
41:92:61:8b:54:66:c3:e6:76:f5:3c:58:80:6b:c3:11:0a:1e:
d7:25:44:bf:42:ab:e9:0e:a0:b1:49:eb:dd:76:e4:96:f6:88:
77:ad:10:14:76:a5:03:fb:b0:20:ff:b5:55:89:53:fd:5e:bc:
52:f9:ef:5e:e5:97:23:03:e3:6e:89:b9:c9:c1:58:19:b3:10:
df:57:50:5a:17:c7:63:13:3a:7d:27:85:99:81:55:ad:1d:10:
45:91:68:c4:e4:19:8c:07:6a:4c:d5:ac:50:d2:d1:df:d2:db:
d6:f6:bb:65:93:51:89:78:5a:7a:dd:21:90:2f:4c:19:70:08:
ef:dd:12:d6:4a:a8:45:da:d2:7d:f7:be:1d:93:b7:4f:03:a8:
54:18:37:dd:a6:ef:43:82:6f:c5:97:f5:ab:0f:d4:75:d1:6b:
20:d7:b8:df:6f:6d:5a:62:8c:31:2b:82:e4:6a:2c:da:57:a5:
14:25:82:1a:10:99:4b:e5:e2:d6:a7:d1:56:2c:f1:72:48:1f:
d3:14:55:f3:eb:9a:d6:a7:38:22:1f:59:09:a7:11:5d:a5:8b:
97:02:9b:ae:55:66:52:da:c5:5f:b6:a6:52:0e:de:80:40:aa:
f3:ad:09:56:42:bb:1e:e9:6b:7d:d7:58:74:35:4c:36:80:3d:
97:6e:d0:0b:6d:e8:5b:70:41:c9:59:ff:3b:5c:99:28:6b:db:
20:72:df:1d:97:8f:dc:f2:4e:fb:60:db:c0:cc:4d:01:ea:aa:
ec:99:a8:97:70:48:07:e7:fd:e6:55:d2:7f:ef:c4:4d:99:59:
0e:24:34:72:39:28:1c:f3:7b:75:f6:e1:97:83:74:ec:ce:6c:
b5:ac:d1:02:a8:a6:28:8b:f0:0e:6a:58:6e:41:e9:27:54:7a:
db:2b:23:ca:44:f7:d9:53:10:c3:a6:45:10:4d:57:6a:dc:2a:
cf:b3:19:bf:f7:19:f2:b1:79:af:5e:c3:cd:75:f5:c8:e5:28:
1f:ab:17:06:50:82:f3:8e:96:20:97:6e:38:df:4d:33:3b:4f:
43:17:af:60:75:25:62:da:66:d1:f6:95:ee:d3:cf:15:33:84:
f8:b7:0e:9c:d6:4e:dc:3a:a6:4d:bf:c5:40:06:b6:97:a9:64:
82:a0:ff:8f:f2:63:08:6c

user@SecServer: ~/CA$
```

Part 2: Enable Zero Trust for Docker

16. **Make a screen capture** showing the unknown certificate authority error.



The screenshot shows a vWorkstation window titled "Managing Network Security in the Cloud" with a timestamp of "2024-03-16 22:51:13" and the name "Andre Hardy". Inside the window is a terminal window titled "user@DockerRunner: ~". The terminal displays system information for Ubuntu 20.04.4 LTS, including system load, memory usage, and network addresses. It then shows the command `sudo docker pull registry:5000/alpine` being executed. The output of the command is an error message: "Error response from daemon: Get \"https://registry:5000/v2/\": x509: certificate signed by unknown authority".

```
user@DockerRunner: ~
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sun 17 Mar 2024 02:50:42 AM UTC

System load: 0.0 Processes: 164
Usage of /: 28.3% of 24.4GB Users logged in: 0
Memory usage: 7% IPv4 address for docker0: 172.17.0.1
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

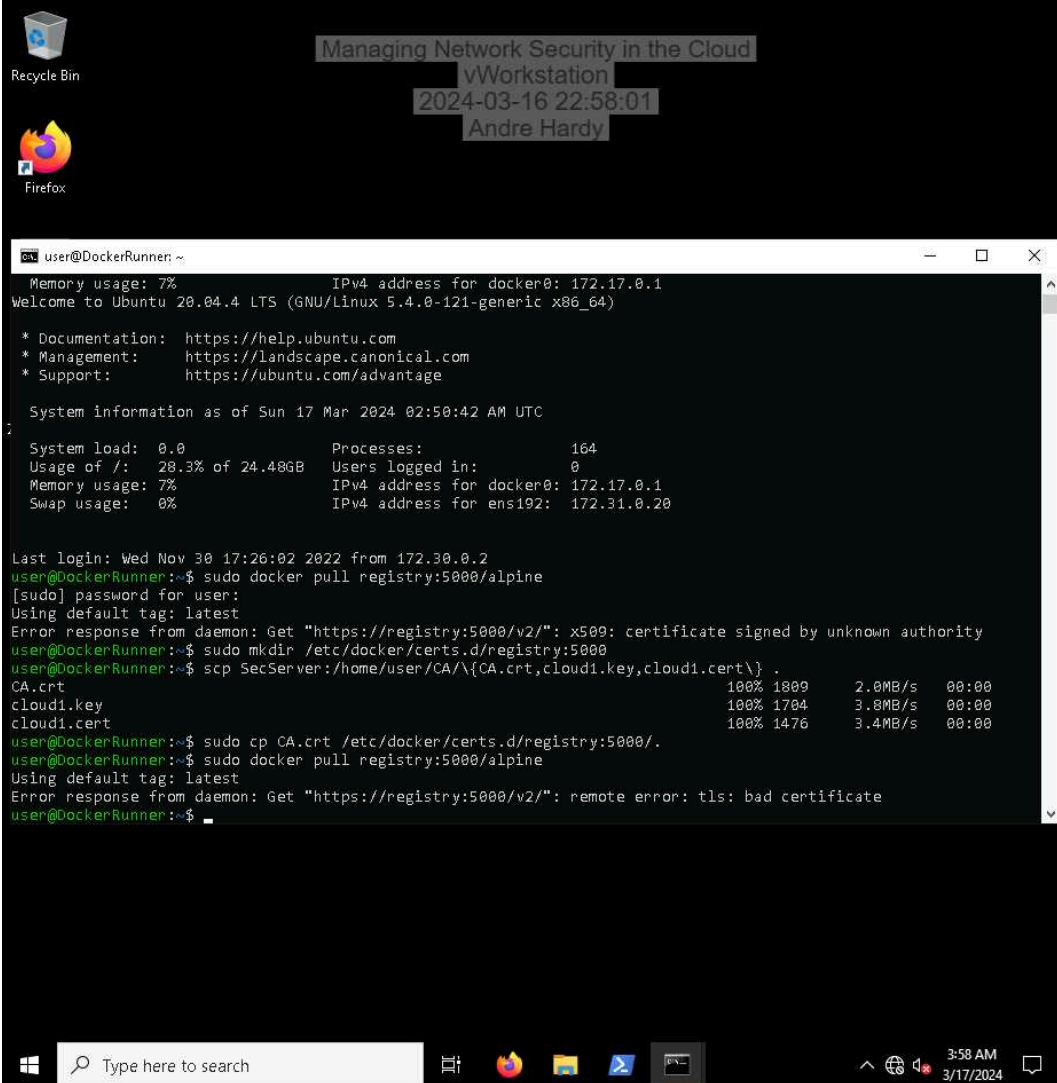
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sun 17 Mar 2024 02:50:42 AM UTC

System load: 0.0 Processes: 164
Usage of /: 28.3% of 24.4GB Users logged in: 0
Memory usage: 7% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for ens192: 172.31.0.20

Last login: Wed Nov 30 17:26:02 2022 from 172.30.0.2
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
[sudo] password for user:
Using default tag: latest
Error response from daemon: Get "https://registry:5000/v2/": x509: certificate signed by unknown authority
user@DockerRunner:~$
```

21. Make a screen capture showing the bad certificate error.



The screenshot shows a Windows desktop environment. At the top, a title bar reads "Managing Network Security in the Cloud". Below it, a "vWorkstation" window displays the date and time "2024-03-16 22:58:01" and the user name "Andre Hardy". The desktop background is dark. On the left, there are icons for "Recycle Bin" and "Firefox". A terminal window titled "user@DockerRunner: ~" is open, displaying the following text:

```
Memory usage: 7% IPv4 address for docker0: 172.17.0.1
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

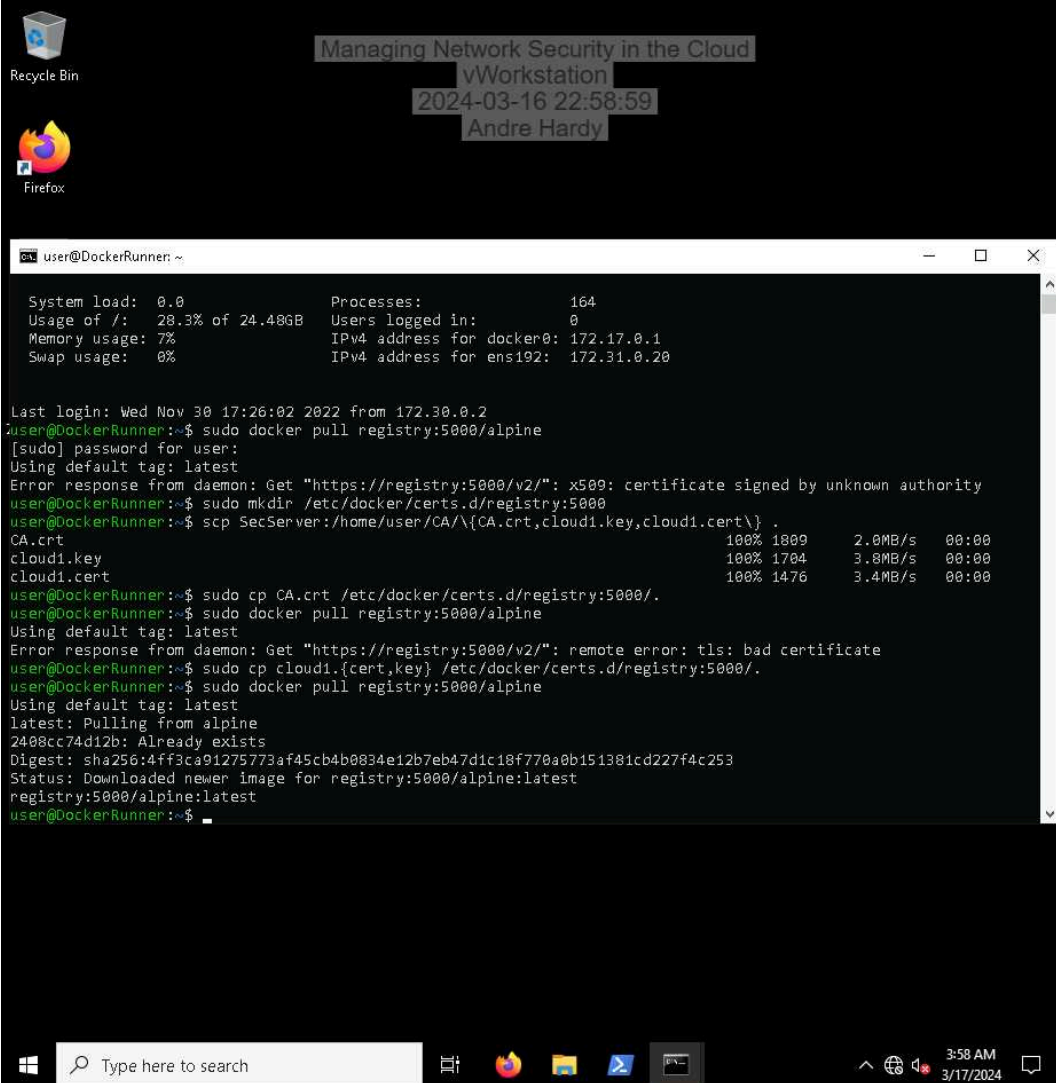
System information as of Sun 17 Mar 2024 02:50:42 AM UTC

System load:  0.0      Processes:      164
Usage of /:   28.3% of 24.4GB Users logged in: 0
Memory usage: 7%      IPv4 address for docker0: 172.17.0.1
Swap usage:  0%       IPv4 address for ens192: 172.31.0.20

Last login: Wed Nov 30 17:26:02 2022 from 172.30.0.2
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
[sudo] password for user:
Using default tag: latest
Error response from daemon: Get "https://registry:5000/v2/": x509: certificate signed by unknown authority
user@DockerRunner:~$ sudo mkdir /etc/docker/certs.d/registry:5000
user@DockerRunner:~$ scp SecServer:/home/user/CA/{CA.crt,cloud1.key,cloud1.cert} .
CA.crt                                100% 1809      2.0MB/s   00:00
cloud1.key                            100% 1704      3.8MB/s   00:00
cloud1.cert                           100% 1476      3.4MB/s   00:00
user@DockerRunner:~$ sudo cp CA.crt /etc/docker/certs.d/registry:5000/.
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
Using default tag: latest
Error response from daemon: Get "https://registry:5000/v2/": remote error: tls: bad certificate
user@DockerRunner:~$
```

The terminal window is titled "user@DockerRunner: ~". The desktop taskbar at the bottom shows the Windows logo, a search bar with the text "Type here to search", and several application icons. The system tray on the right shows the time "3:58 AM" and the date "3/17/2024".

24. **Make a screen capture** showing a successful pull of registry:5000/alpine on DockerRunner.



The screenshot shows a Windows desktop environment. At the top, there is a title bar for a window titled "Managing Network Security in the Cloud" with a subtitle "vWorkstation" and a timestamp "2024-03-16 22:58:59". Below this, the user's name "Andre Hardy" is visible. The desktop background is dark. On the left side, there are icons for "Recycle Bin" and "Firefox". The main focus is a terminal window titled "user@DockerRunner: ~". The terminal displays the following content:

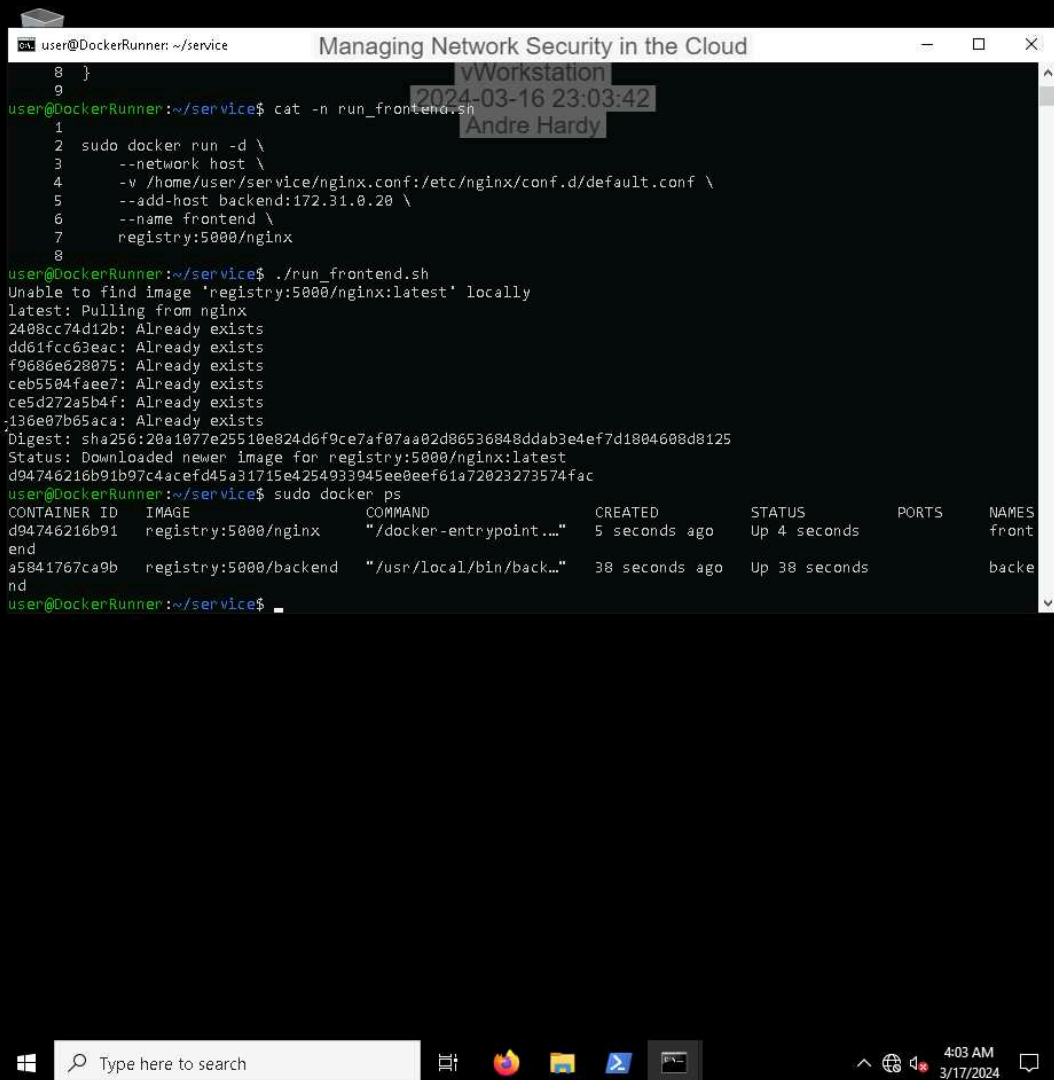
```
System load: 0.0      Processes:           164
Usage of /:  28.3% of 24.4GB  Users logged in:    0
Memory usage: 7%      IPv4 address for docker0: 172.17.0.1
Swap usage:  0%         IPv4 address for ens192: 172.31.0.20

Last login: Wed Nov 30 17:26:02 2022 from 172.30.0.2
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
[sudo] password for user:
Using default tag: latest
Error response from daemon: Get "https://registry:5000/v2/": x509: certificate signed by unknown authority
user@DockerRunner:~$ sudo mkdir /etc/docker/certs.d/registry:5000
user@DockerRunner:~$ scp SecServer:/home/user/CA/{CA.crt,cloud1.key,cloud1.cert} .
CA.crt                                100% 1809      2.0MB/s   00:00
cloud1.key                            100% 1704      3.8MB/s   00:00
cloud1.cert                           100% 1476      3.4MB/s   00:00
user@DockerRunner:~$ sudo cp CA.crt /etc/docker/certs.d/registry:5000/.
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
Using default tag: latest
Error response from daemon: Get "https://registry:5000/v2/": remote error: tls: bad certificate
user@DockerRunner:~$ sudo cp cloud1.{cert,key} /etc/docker/certs.d/registry:5000/.
user@DockerRunner:~$ sudo docker pull registry:5000/alpine
Using default tag: latest
latest: Pulling from alpine
2408cc74d12b: Already exists
Digest: sha256:4ff3ca91275773af45cb4b0834e12b7eb47d1c18f770a0b151381cd227f4c253
Status: Downloaded newer image for registry:5000/alpine:latest
registry:5000/alpine:latest
user@DockerRunner:~$
```

The terminal window is open on a Windows desktop. The taskbar at the bottom shows the Start button, a search bar with the text "Type here to search", and several application icons including Firefox, File Explorer, and Docker Desktop. The system tray on the right shows the time as 3:58 AM on 3/17/2024.

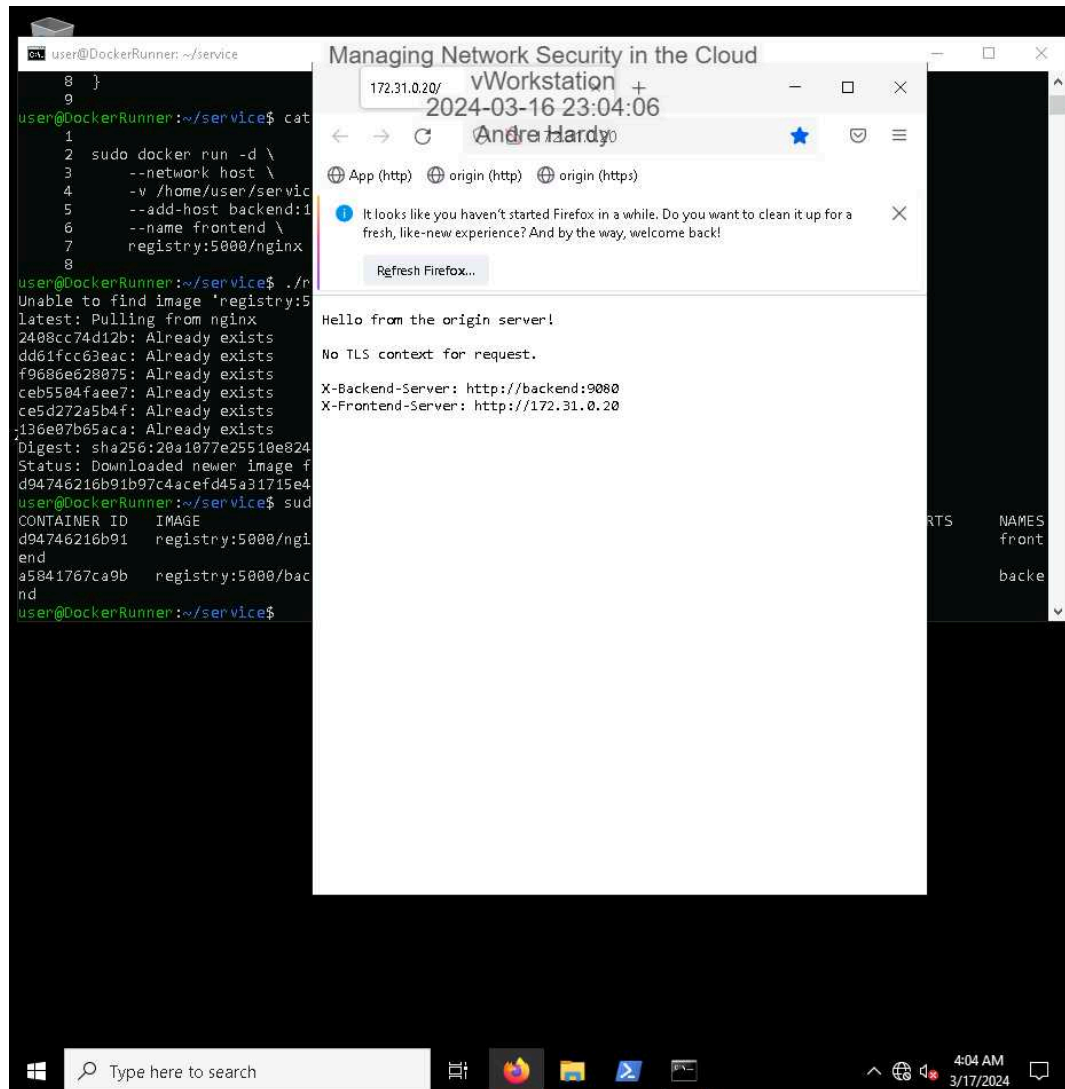
Part 3: Using Zero Trust for a Web Service

13. **Make a screen capture** showing the running frontend and backend containers.

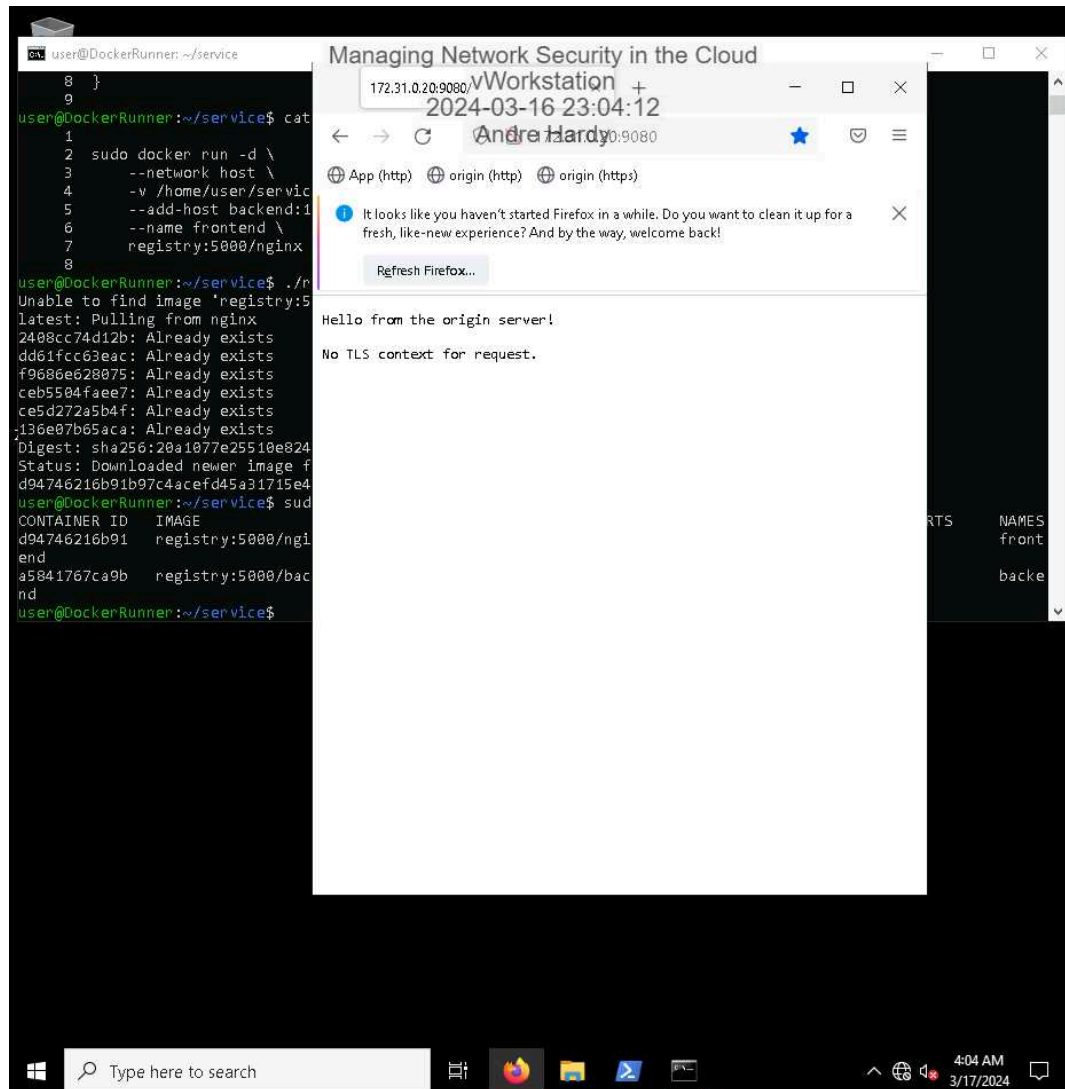


```
user@DockerRunner: ~/service
8 }
9
user@DockerRunner:~/service$ cat -n run_frontend.sh
1
2 sudo docker run -d \
3   --network host \
4   -v /home/user/service/nginx.conf:/etc/nginx/conf.d/default.conf \
5   --add-host backend:172.31.0.20 \
6   --name frontend \
7   registry:5000/nginx
8
user@DockerRunner:~/service$ ./run_frontend.sh
Unable to find image 'registry:5000/nginx:latest' locally
latest: Pulling from nginx
2408cc74d12b: Already exists
dd61fcc63eac: Already exists
f9686e628075: Already exists
ceb5504faee7: Already exists
ce5d272a5b4f: Already exists
136e07b65aca: Already exists
Digest: sha256:20a1077e25510e824d6f9ce7af07aa02d86536848ddab3e4ef7d1804600d8125
Status: Downloaded newer image for registry:5000/nginx:latest
d94746216b91b97c4acefd45a31715e4254933945ee0eef61a72023273574fac
user@DockerRunner:~/service$ sudo docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
d94746216b91   registry:5000/nginx  "/docker-entrypoint...."  5 seconds ago  Up 4 seconds             front
a5841767ca9b   registry:5000/backend "/usr/local/bin/back..." 38 seconds ago  Up 38 seconds             back
user@DockerRunner:~/service$
```

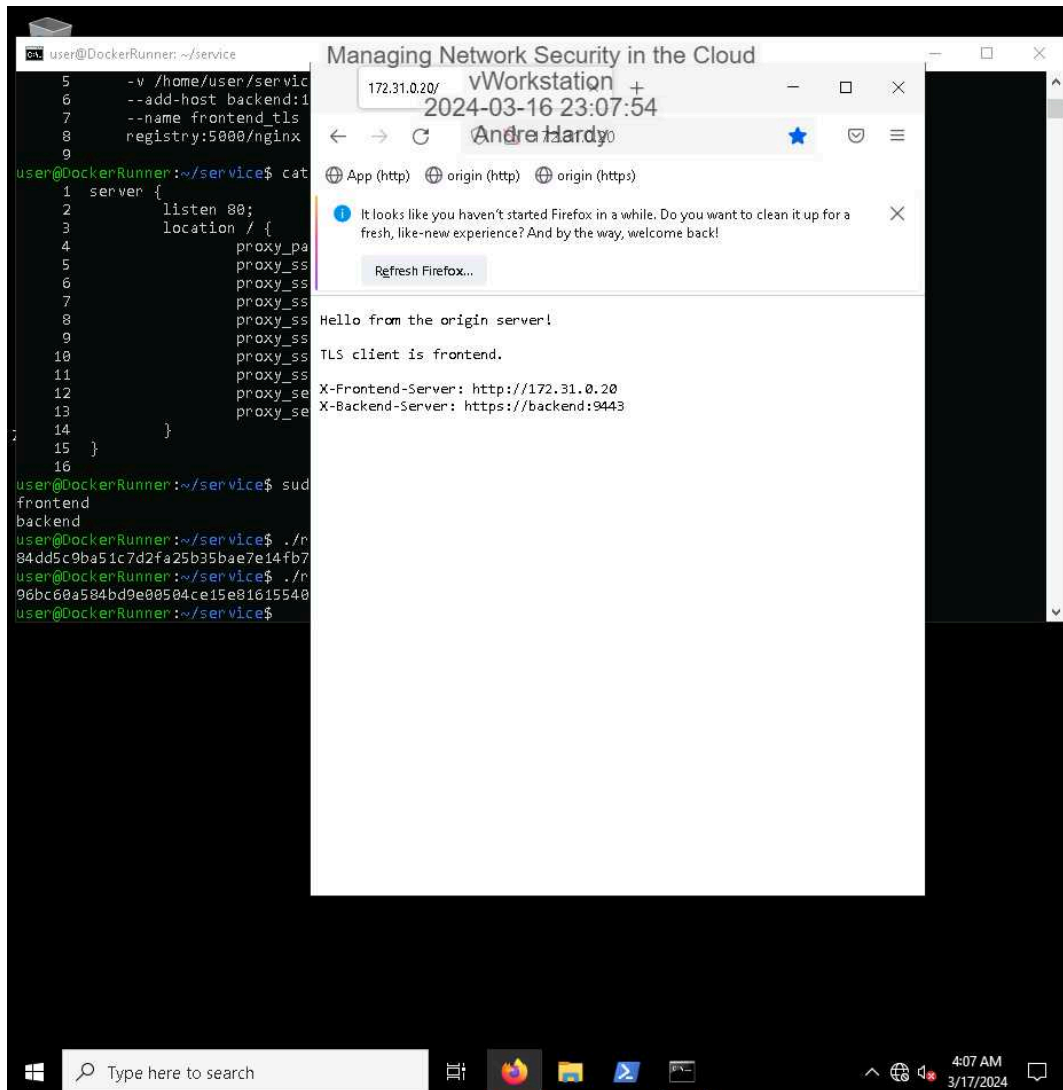
16. Make a screen capture showing the successful page load from App (http).



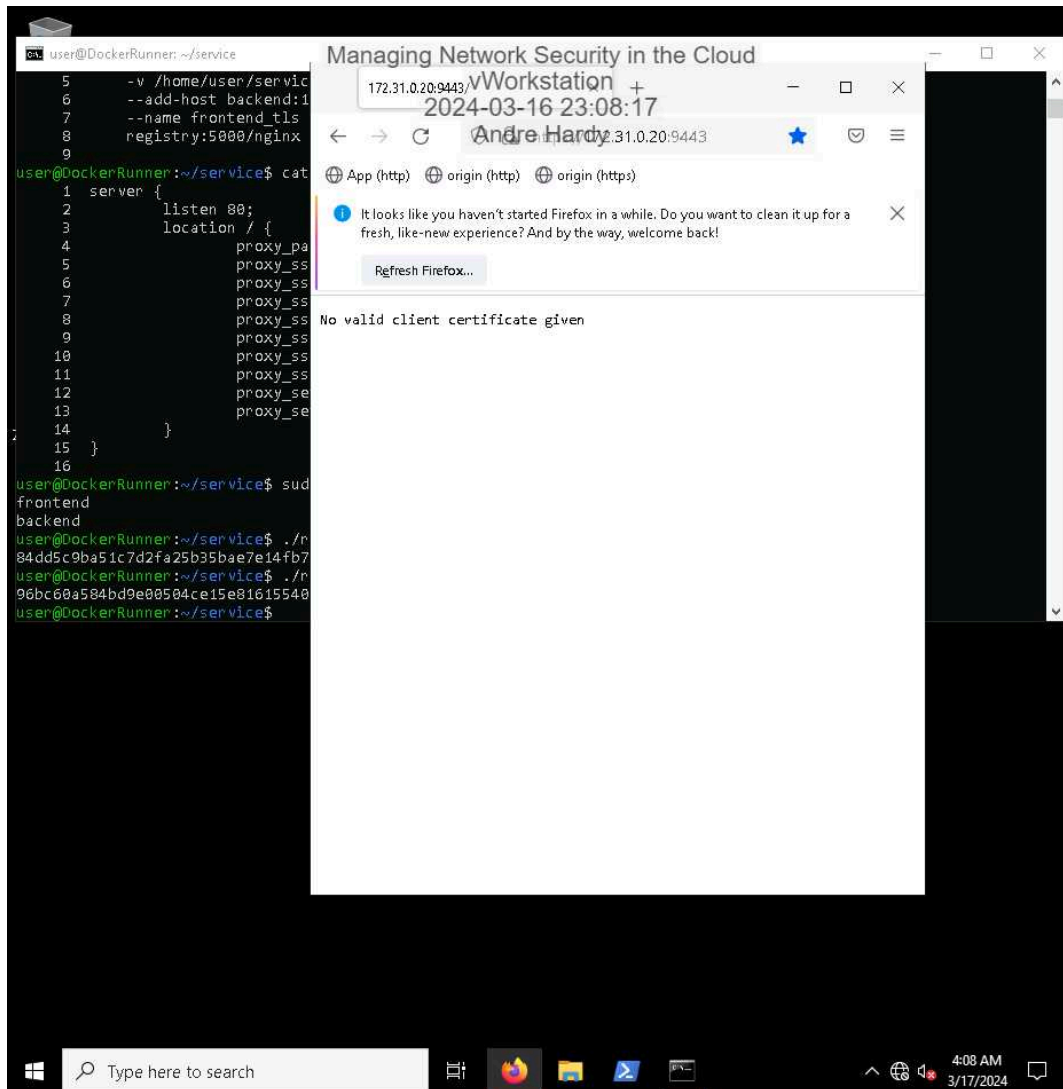
18. Make a screen capture showing the successful page load from origin (http).



30. **Make a screen capture** showing the successful page load from App (http) with mutual TLS on.



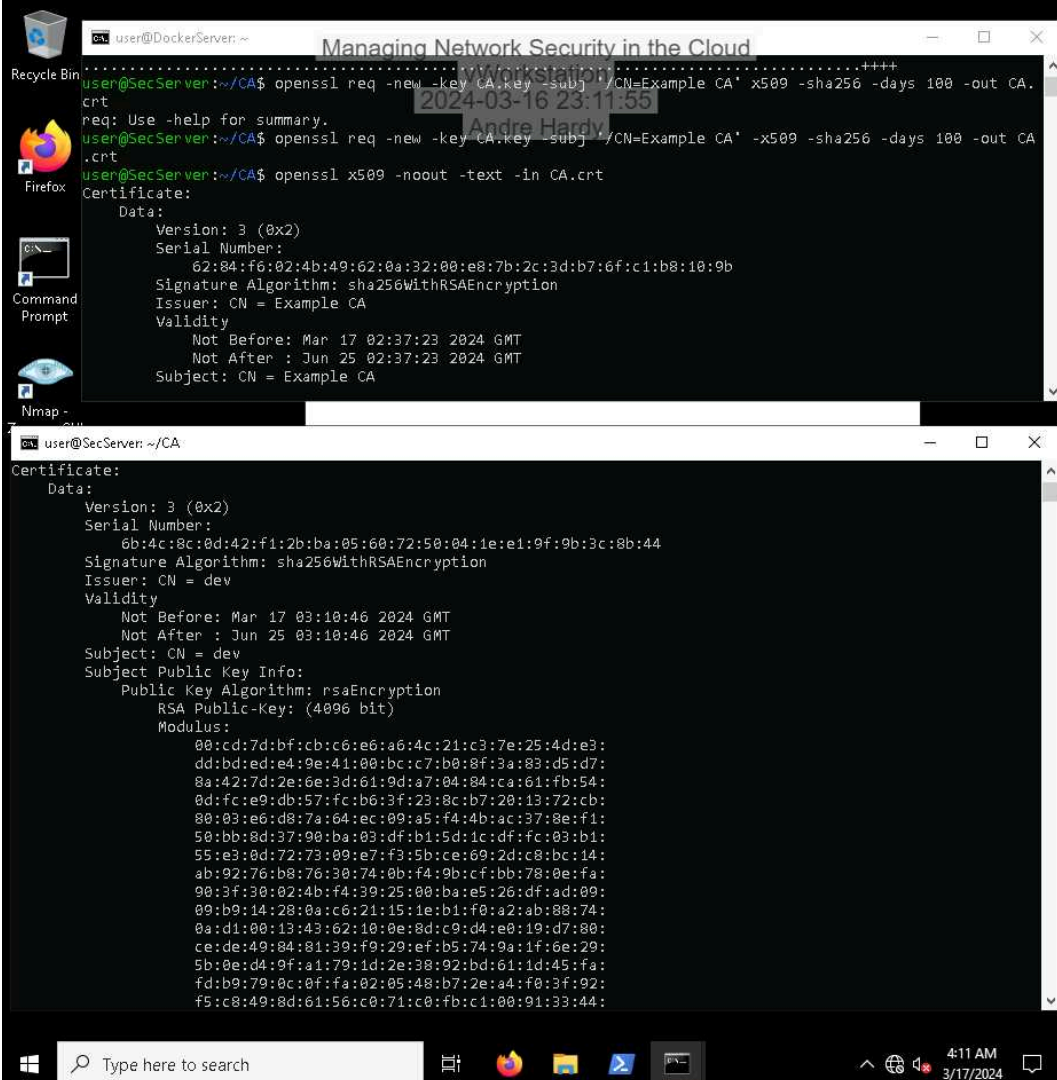
33. Make a screen capture showing the unsuccessful load from origin (https).



Challenge and Analysis

Part 1: Enable Mutual TLS for the Development Machine

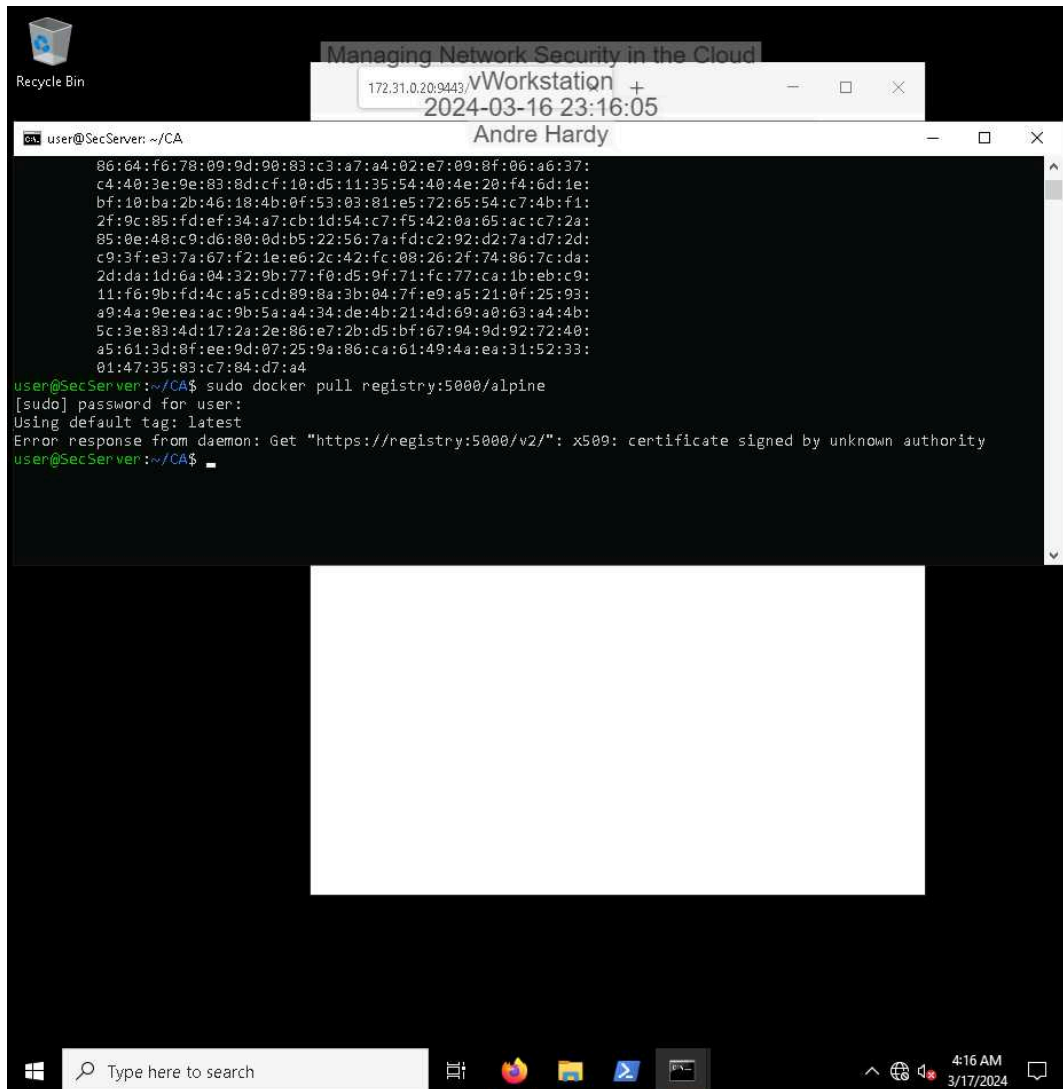
1. Make a screen capture showing at least the first 12 lines of the client certificate



```
user@DockerServer: ~$ openssl req -new -key CA.key -subj /CN=Example CA' -x509 -sha256 -days 100 -out CA.crt
req: Use -help for summary.
user@DockerServer: ~$ openssl req -new -key CA.key -subj /CN=Example CA' -x509 -sha256 -days 100 -out CA.crt
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      62:84:f6:02:4b:49:62:0a:32:00:e8:7b:2c:3d:b7:6f:c1:b8:10:9b
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN = Example CA
    Validity
      Not Before: Mar 17 02:37:23 2024 GMT
      Not After : Jun 25 02:37:23 2024 GMT
    Subject: CN = Example CA

user@SecServer: ~/CA$ openssl x509 -noout -text -in CA.crt
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      6b:4c:8c:0d:42:f1:2b:ba:05:60:72:50:04:1e:e1:9f:9b:3c:8b:44
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: CN = dev
    Validity
      Not Before: Mar 17 03:10:46 2024 GMT
      Not After : Jun 25 03:10:46 2024 GMT
    Subject: CN = dev
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public-Key: (4096 bit)
      Modulus:
        00:cd:7d:bf:cb:c6:e6:a6:4c:21:c3:7e:25:4d:e3:
        dd:bd:ed:e4:9e:41:00:bc:c7:b0:8f:3a:83:d5:d7:
        8a:42:7d:2e:6e:3d:61:9d:a7:04:84:ca:61:fb:54:
        0d:fc:e9:db:57:fc:b6:3f:23:8c:b7:20:13:72:cb:
        80:03:e6:d8:7a:64:ec:09:a5:f4:4b:ac:37:8e:f1:
        50:bb:8d:37:90:ba:03:df:b1:5d:1c:df:fc:03:b1:
        55:e3:0d:72:73:09:e7:f3:5b:ce:69:2d:c8:bc:14:
        ab:92:76:b8:76:30:74:0b:f4:9b:cf:bb:78:0e:fa:
        90:3f:30:02:4b:f4:39:25:00:ba:e5:26:df:ad:09:
        09:b9:14:28:0a:c6:21:15:1e:b1:f0:a2:ab:88:74:
        0a:d1:00:13:43:62:10:0e:8d:c9:d4:e0:19:d7:80:
        ce:de:49:04:81:30:f9:29:ef:b5:74:9a:1f:6e:29:
        5b:0e:d4:9f:a1:79:1d:2e:38:92:bd:61:1d:45:fa:
        fd:b9:79:0c:0f:fa:02:05:48:b7:2e:a4:f0:3f:92:
        f5:c8:49:8d:61:56:c0:71:c0:fb:c1:00:91:33:44:
```

2. Make a screen capture showing a successful pull of registry:5000/alpine on SecServer.



Part 2: Check Certificates Manually

1. **Make a screen capture** showing the output of the verification command for the developer certificate.

