**SSO demo project guide**

# **Red Hat installation**

Create HyperV virtual machine and set ISO image (before download RedHat ISO image from RedHat site

* + Download RHEL ISO:
    - Register for the Red Hat Developer Program to download the RHEL ISO for free (for development and testing purposes).
    - <https://developers.redhat.com/products/rhel/download>
  + Create a New Virtual Machine in Hyper-V:
    - Open Hyper-V Manager.
    - Click New > Virtual Machine and follow the wizard to create a new VM.
    - Specify the Name and Location of the VM.
    - Choose Generation 1 for broader compatibility.
  + Configure VM Settings:
    - RAM: Allocate at least 4 GB of RAM (6 GB or more is recommended for better performance).
    - Processor: Assign an appropriate number of virtual processors based on the resources of your host machine.
    - Networking: Connect the VM to a virtual switch to allow network access.
  + Attach the RHEL ISO File:
    - In the Connect Virtual Hard Disk step, choose Create a virtual hard disk and allocate sufficient storage (at least 40 GB).
    - Go to Installation Options and select Install an operating system from a bootable CD/DVD-ROM.
    - Browse and attach the RHEL ISO file you downloaded earlier.
  + Configure Boot Settings:
    - Ensure the DVD Drive is the first boot device in the Firmware section of the VM settings.
  + Start the VM:
    - Right-click the newly created VM in Hyper-V Manager and click Connect, then select Start.
    - The VM should boot from the RHEL ISO, and the installation process will begin.

RedHat official instructions:

<https://www.youtube.com/watch?v=5A_BucrI34A>

<https://www.linuxtechi.com/how-to-install-freeipa-server-on-rhel/>

## 

## **Setting hostname**

ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.0.43 netmask 255.255.255.0 broadcast 192.168.0.255

inet6 fda5:8943:8dcd:5147:215:5dff:fe00:2804 prefixlen 64 scopeid 0x0<global>

inet6 fe80::215:5dff:fe00:2804 prefixlen 64 scopeid 0x20<link>

ether 00:15:5d:00:28:04 txqueuelen 1000 (Ethernet)

RX packets 2935303 bytes 4325927621 (4.0 GiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 710821 bytes 67643458 (64.5 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 31209 bytes 10857313 (10.3 MiB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 31209 bytes 10857313 (10.3 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[pcaric@localhost ~]$ !2

sudo vim /etc/hosts

[pcaric@localhost ~]$ sudo cat /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.0.43 idm.example.test idm

[pcaric@localhost ~]$

[pcaric@localhost ~]$ nslookup idm.example.test

Server: 127.0.0.1

Address: 127.0.0.1#53

Name: idm.example.test

Address: 192.168.0.43

# 

# **Update OS and installing Freeipa and DNS**

**sudo dnf update -y**

**sudo dnf install freeipa-server freeipa-server-dns -y**

**sudo ipa-server-install --setup-dns**

Setup complete

Next steps:

1. You must make sure these network ports are open:

TCP Ports:

\* 80, 443: HTTP/HTTPS

\* 389, 636: LDAP/LDAPS

\* 88, 464: kerberos

\* 53: bind

UDP Ports:

\* 88, 464: kerberos

\* 53: bind

\* 123: ntp

Firewall enabling:

sudo firewall-cmd --permanent --add-port={80/tcp,443/tcp,389/tcp,636/tcp,88/tcp,464/tcp,53/tcp,88/udp,464/udp,53/udp,123/udp}

sudo firewall-cmd --reload

# TCP Ports

sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT # HTTP

sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT # HTTPS

sudo iptables -A INPUT -p tcp --dport 389 -j ACCEPT # LDAP

sudo iptables -A INPUT -p tcp --dport 636 -j ACCEPT # LDAPS

sudo iptables -A INPUT -p tcp --dport 88 -j ACCEPT # Kerberos

sudo iptables -A INPUT -p tcp --dport 464 -j ACCEPT # Kerberos

sudo iptables -A INPUT -p tcp --dport 53 -j ACCEPT # DNS

# UDP Ports

sudo iptables -A INPUT -p udp --dport 88 -j ACCEPT # Kerberos

sudo iptables -A INPUT -p udp --dport 464 -j ACCEPT # Kerberos

sudo iptables -A INPUT -p udp --dport 53 -j ACCEPT # DNS

sudo iptables -A INPUT -p udp --dport 123 -j ACCEPT # NTP

Backup certificate:

sudo cp /root/cacert.p12 //home/pcaric/certbackup/cacert.p12

Checking after installation and setting firewall:

sudo systemctl

status ipa sudo systemctl status named-pkcs11

sudo systemctl status dirsrv.target

sudo systemctl status httpd

sudo systemctl status krb5kdc

sudo systemctl status kadmin

# **Kerberos settings**

## **Setting Kerberos ticket for admin**

pcaric@localhost ~]$ kinit admin

[pcaric@localhost ~]$ klist

Ticket cache: KCM:1000

Default principal: admin@EXAMPLE.TEST

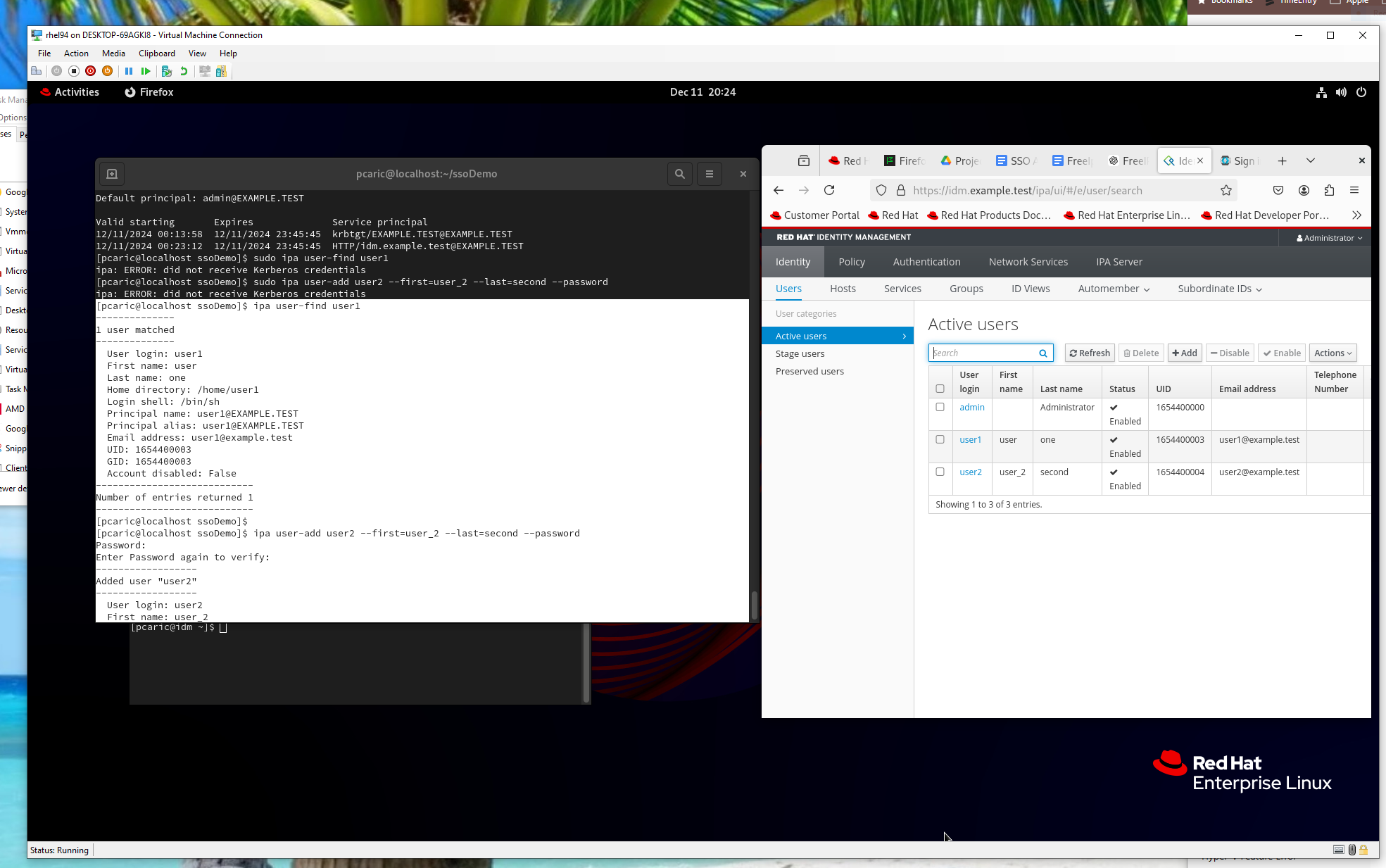
Valid starting Expires Service principal

12/11/2024 00:13:58 12/11/2024 23:45:45 krbtgt/EXAMPLE.TEST@EXAMPLE.TEST

12/11/2024 00:23:12 12/11/2024 23:45:45 HTTP/idm.example.test@EXAMPLE.TEST

[pcaric@localhost ~]$

## **Accessing Kerberos via web page**



<https://idm.example.test/ipa/ui>

Add user1 via web page, it can also be added via terminal:  
ipa user-add user1 --first=user\_1 --last=second --password

and checked with:  
pa user-find user1

Checking with ldapsearch:

ldapsearch -x -h idm.example.test -D "cn=Directory Manager" -w paola123 -b "cn=users,cn=accounts,dc=example,dc=test"

# **Setting web applications**

[pcaric@localhost ~]$ sudo mkdir -p /var/www/app1.com/public\_html

[pcaric@localhost ~]$ sudo mkdir -p /var/www/app2.com/public\_html

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$ sudo vim /var/www/app1.com/public\_html/index.html

[pcaric@localhost ~]$ sudo vim /var/www/app2.com/public\_html/index.html

[pcaric@localhost ~]$

[pcaric@localhost conf.d]$ cat /var/www/app1.com/public\_html/index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Keycloak Demo App 1</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/keycloak-js/21.0.2/keycloak.min.js"></script>

</head>

<body class="bg-light">

<div class="container py-5">

<div class="row justify-content-center">

<div class="col-md-6 text-center">

<h1 class="mb-4">Welcome to the Demo App 1</h1>

<div>

<button id="loginBtn" class="btn btn-primary btn-lg mb-3"

onclick="window.location.href = '/login'">Login</button>

</div>

</div>

</div>

</div>

</body>

</html>

do same for the app2  
  
  
Web page after login: app1after.html

[pcaric@localhost conf.d]$ cat /var/www/app1.com/public\_html/app1after.html  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Keycloak Demo App 1</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/keycloak-js/21.0.2/keycloak.min.js"></script>

</head>

<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

<div class="container-fluid">

<a class="navbar-brand" href="#">You successfully log in at Demo App 1</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">

<span class="navbar-toggler-icon"></span>

</button>

</div>

</nav>

</body>

</html>  
  
do same for the app2

## **Setting Apache server, reverse proxy with certificates**

[pcaric@localhost ~]$ sudo mkdir -p /etc/ssl/app2

[pcaric@localhost ~]$ sudo mkdir -p /etc/ssl/app1

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/app1/privkey.pem -out /etc/ssl/app1/fullchain.pem

...+.+......+.....+...+.+...+++++++++++++++++++++++++++++++++++++++\*..+.+.........+...+.....+......+++++++++++++++++++++++++++++++++++++++\*.+......+...................+...+............+..+.........+....+......+......+..............+.......+..+......+....+..+......+.........+......+..........+..+.+......+...........+............+...+...+....+...+..+...............+.+...........+...+...+....+...+........+...+......+......+.......+......+.....+.+........+...+....+.....+....+...........+...+.+.....+.........+...+.........+.+...+............+.....+....+...+........+...+....+........+.........+.+........+.+.....+.+..+.........................+...+........+......+.+.....+.......+..+......+...+.+...+.....+...+....+...+..+....+.....+.......+..+...+............+.......+...+...+.....+...............+............+.+............+..+.+.....+....+...+..+............+....+......+.....+.+.....+...+..........+...........+......+....+..................++++++

..+.........+.+++++++++++++++++++++++++++++++++++++++\*........+++++++++++++++++++++++++++++++++++++++\*..+....+.....+.++++++

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:HR

State or Province Name (full name) []:

Locality Name (eg, city) [Default City]:

Organization Name (eg, company) [Default Company Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) []:app1.com

Email Address []:

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$

[pcaric@localhost ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/app2/privkey.pem -out /etc/ssl/app2/fullchain.pem

......+..+...+...+....+...+........+......+.+++++++++++++++++++++++++++++++++++++++\*........+.....+....+...........+.+..+......+.........+.......+...........+.........+..........+.....+....+......+...+..................+..+................+..............+.......+...+...+...+++++++++++++++++++++++++++++++++++++++\*.+...+.........+...+..................+..+......+.+.....+.+.....+...+..........+.........+..+.............+......+.....+............+...+..........+..............+.+.....+.+.....+...+............+....+.........+..+...+.+......+...+..+......+..........+..+......+.+......+..+.+.....+.+.....+.+............+..+..........+...+..................+..+..................+...++++++

...+...+.+.....+...............+.+..+.+..+............+++++++++++++++++++++++++++++++++++++++\*.+....+......+........+.+......+.....+...+...+...+......+++++++++++++++++++++++++++++++++++++++\*....+.................+..................+...+............................+..+.......+..+......+.........+...................+.........+........+...+....+...+...+..+..........+..+......+......+...+......+.+........+...+.........................+.....+............+.+......+......+...+.....................+..................+..+...+...+...+....+...+...+.........++++++

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:HR

State or Province Name (full name) []:

Locality Name (eg, city) [Default City]:

Organization Name (eg, company) [Default Company Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) []:app2.com

Email Address []:

## **Setting apache configuration - after Keycloak is configured**

Install module for Apache for OIDC:

sudo dnf install mod\_auth\_openidc -y

<VirtualHost \*:80>

ServerName app1.com

Redirect permanent / https://app1.com/

</VirtualHost>

<VirtualHost \*:443>

ServerName app1.com

ServerAlias www.app1.com

DocumentRoot /var/www/app1.com/public\_html

<Directory /var/www/app1.com/public\_html>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

SSLEngine on

SSLCertificateFile /etc/ssl/app1/fullchain.pem

SSLCertificateKeyFile /etc/ssl/app1/privkey.pem

# Keycloak settings

OIDCProviderMetadataURL http://idm.example.test:9080/realms/PaolaCompany/.well-known/openid-configuration

OIDCClientID app1

OIDCClientSecret DQevstxSoBvijSnTaV10jYCrfFIvHgWC

OIDCRedirectURI https://app1.com/app1after.html

OIDCResponseType code

OIDCScope "openid email profile"

# Mandatory Crypto Passphrase for OIDC

OIDCCryptoPassphrase a0f256f445326a385ed81b9eb8c296c4eedbbf1aa164545c14597ddcd8a3b861

<Location /login>

AuthType openid-connect

Require valid-user

</Location>

#ProxyPreserveHost On

#ProxyPass / http://app1.com

#ProxyPassReverse / http://app1.com

ErrorLog /var/log/httpd/app1-error.log

CustomLog /var/log/httpd/app1-access.log combined

</VirtualHost>

From the Keycloak Client / app1 / credentials copy secret for   
OIDCClientSecret

Generating OIDCCryptoPassphrase:

openssl rand -hex 32

now copy / paste this for OIDCCryptoPassphrase

app2 credentials from Keyclaok clien2 (app2): 07hgI01ynmiBEQBadRAdyhpLoATVPLlH

PassPhrase for app2: f7c73a7824b090c347bdaae11ba465c320025340c29ae6cebc1b1f781f83571a

[pcaric@localhost ~]$ sudo apachectl configtest

Syntax OK

# **Keycloak installation**

Enable port 9080 (for Keycloak) in the RedHat firewall to be able to access from the host

sudo firewall-cmd --add-port=9080/tcp

**IMPORTANT SETTINGS ON THE WINDOWS HOST**

c:/windows/system32/drivers/etc/hosts

add this line in the file above  
192.168.0.43 app1.com app2.com idm.example.test

(192.168.0.43 is the IP of the HyperV VM so it can be different)

this has to be done in order to be able to access:  
app1.com (Web app1)

app2.com (Web app2)

idm.example.com:9080 (Keycloak)

from the host’s browser

**DOWNLOAD KEYCLOAK:**

<https://medium.com/devsecops-community/how-to-install-keycloak-on-linux-full-setup-dev-production-modes-and-ad-integration-6429d75fbd62>

**DOWNLOAD KEYCLOAK 23 OR SOME NEWER VERSION:**

wget https://github.com/keycloak/keycloak/releases/download/23.0.3/keycloak-23.0.3.tar.gz

tar -xf keycloak-23.0.3.tar.gz

export KEYCLOAK\_ADMIN=admin

export KEYCLOAK\_ADMIN\_PASSWORD=admin

cd keycloak-23.0.3

**START KEYCLOAK IN SEPARATE TERMINAL TO MONITOR LOGS**

bin/kc.sh start-dev –http-port=9080

**ACCESS KEYCLOAK FROM THE HOST**

<http://idm.example.test:9080>

## **Configure Keycloak to Use FreeIPA as an LDAP User Federation Provider**

Integrating FreeIPA with Keycloak allows Keycloak to authenticate users from FreeIPA. We'll set up **LDAP User Federation** in Keycloak.

* **Access Keycloak Admin Console:**
  + **Open Your Browser** and navigate to: http://idm.example.test:9080
  + **Log In to Keycloak:**
    - **Username:** admin
    - **Password:** admin
* **Create the "PaolaCompany" Realm:**
  + **Add a New Realm:**
    - In the top-left corner, click on the current realm (e.g., master) dropdown.
    - Click **"Add Realm"**.
  + **Configure Realm Details:**
    - **Name:** PaolaCompany
    - Click **"Create"**.
  + Create client:
    - root url: <https://app1.com>
    - home url: <https://app1.com>
    - valid redirect url: <https://app1.com/app1after.html>
    - valid post logout: <https://app1.com/app1after.html>
    - web origins & admin: <https://app1.com>
  + Capability config: Set Client authentication to ON and SAVE
  + After saving there will be Credentials tab. Go there and copy Client secret and paste
  + in the Apache conf file for OIDCClientSecret.
* **Add LDAP User Federation Provider:**
  + **Navigate to User Federation:**
    - In the left-hand menu, under the PaolaCompany realm, click **"User Federation"**.
  + **Add LDAP Provider:**
    - Click **"Add provider"** and select **"ldap"** from the dropdown.
  + **Configure LDAP Settings:**
    - **Edit Mode:** READ\_ONLY  
      *(This ensures Keycloak does not modify FreeIPA data.)*
    - **Vendor:** Other
    - **Username LDAP Attribute:** uid
    - **RDN LDAP Attribute:** uid
    - **UUID LDAP Attribute:** entryUUID
    - **User Object Classes:** inetOrgPerson
    - **Connection URL:** ldap://idm.example.test  
      *(Use ldaps://idm.*example*.test if you have TLS configured.)*
    - **Users DN:** cn=users,cn=accounts,dc=example,dc=test
    - **Bind DN:** uid=admin,cn=users,cn=accounts,dc=example,dc=test
    - **Bind Credential:** Enter the **Directory Manager** password you set during FreeIPA installation.
    - **Use Truststore SPI:** ldapsOnly if using ldaps://, otherwise never.
    - **Connection Pooling:** Enable or disable based on your needs (default is typically fine).
    - **Cache Settings:** Adjust as needed, but defaults are suitable for most scenarios.
  + **Test Connection and Authentication:**
    - Click **"Test connection"** to ensure Keycloak can reach FreeIPA.
    - Click **"Test authentication"** to verify the Bind DN and password.
  + **Save the Provider:**
    - Once both tests are successful, click **"Save"**.
  + **Synchronization:**
    - After saving, you can **synchronize all users** from FreeIPA to Keycloak by clicking **"Synchronize all users"**.
* **Verify User Federation:**
  + **Navigate to Users:**
    - Click **"Users"** under the PaolaCompany realm.
  + **Check for Imported Users:**
    - You should see admin and user1 listed, reflecting the users from FreeIPA.

# **Restart apache**

pcaric@localhost ~]$ sudo systemctl restart httpd

[pcaric@localhost ~]$ sudo systemctl status httpd

● httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)

Drop-In: /etc/systemd/system/httpd.service.d

└─ipa.conf

Active: active (running) since Wed 2024-12-11 09:57:22 CET; 12s ago

Docs: man:httpd.service(8)

Process: 194343 ExecStartPre=/usr/libexec/ipa/ipa-httpd-kdcproxy (code=exited, status=0/SUCCESS)

Main PID: 194345 (httpd)

Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"

Tasks: 230 (limit: 35692)

Memory: 337.4M

CPU: 4.914s

CGroup: /system.slice/httpd.service

├─194345 /usr/sbin/httpd -DFOREGROUND

├─194347 /usr/sbin/httpd -DFOREGROUND

├─194348 /usr/sbin/httpd -DFOREGROUND

├─194352 "(wsgi:kdcproxy)" -DFOREGROUND

├─194353 "(wsgi:kdcproxy)" -DFOREGROUND

├─194354 "(wsgi:ipa) " -DFOREGROUND

├─194356 "(wsgi:ipa) " -DFOREGROUND

├─194357 "(wsgi:ipa) " -DFOREGROUND

├─194358 "(wsgi:ipa) " -DFOREGROUND

├─194359 /usr/sbin/httpd -DFOREGROUND

├─194360 /usr/sbin/httpd -DFOREGROUND

└─194372 /usr/sbin/httpd -DFOREGROUND

Dec 11 09:57:21 idm.example.test systemd[1]: Starting The Apache HTTP Server...

Dec 11 09:57:22 idm.example.test ipa-httpd-kdcproxy[194343]: ipa: INFO: KDC proxy enabled

Dec 11 09:57:22 idm.example.test ipa-httpd-kdcproxy[194343]: ipa-httpd-kdcproxy: INFO KDC proxy enabled

Dec 11 09:57:22 idm.example.test systemd[1]: Started The Apache HTTP Server.

Dec 11 09:57:22 idm.example.test httpd[194345]: Server configured, listening on: port 443, port 80

[pcaric@localhost ~]$

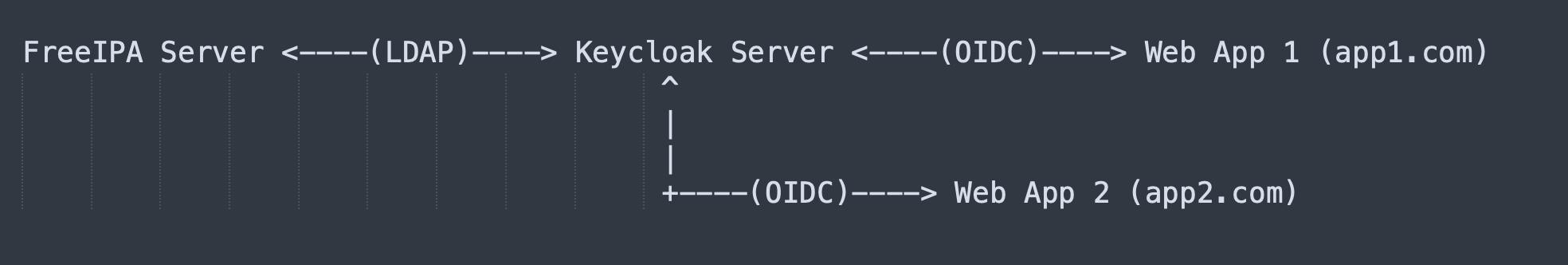
# **Testing SSO call flow**

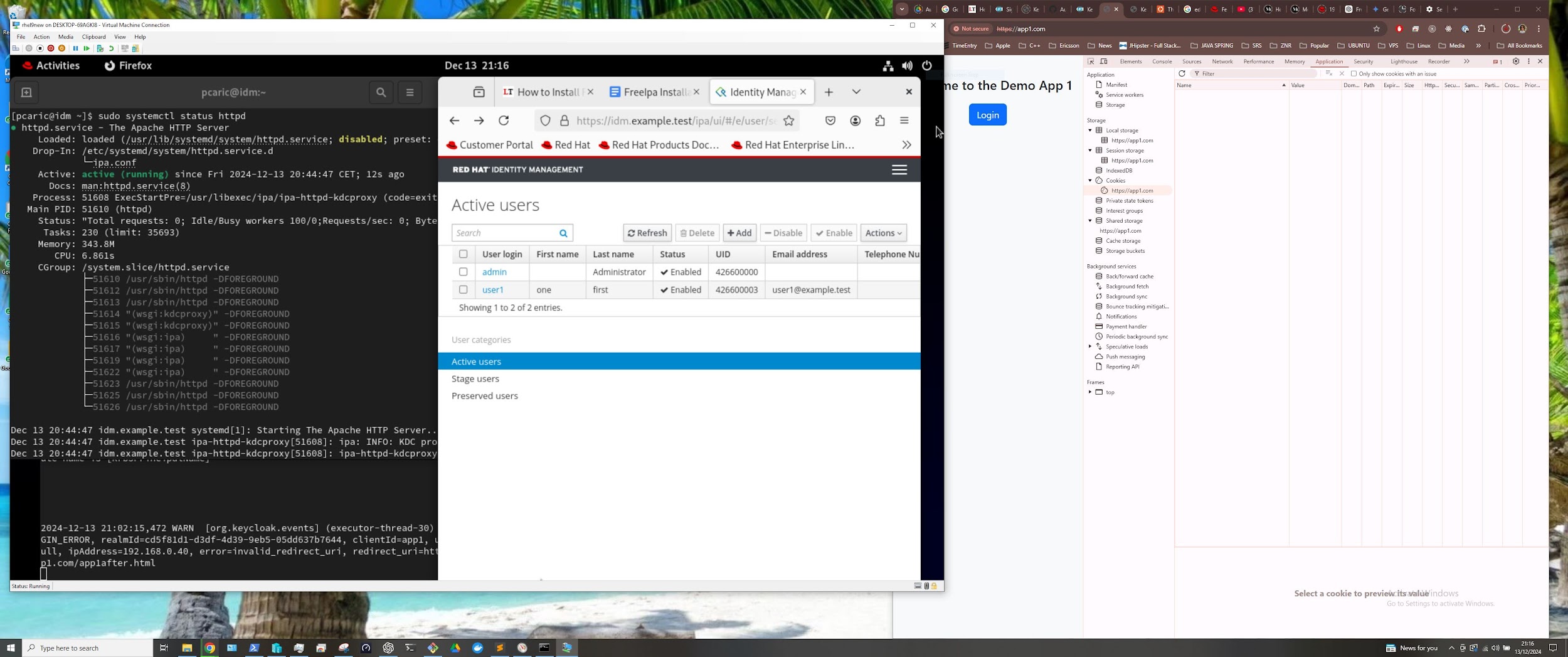
**ON THE VM, TERMINALS**

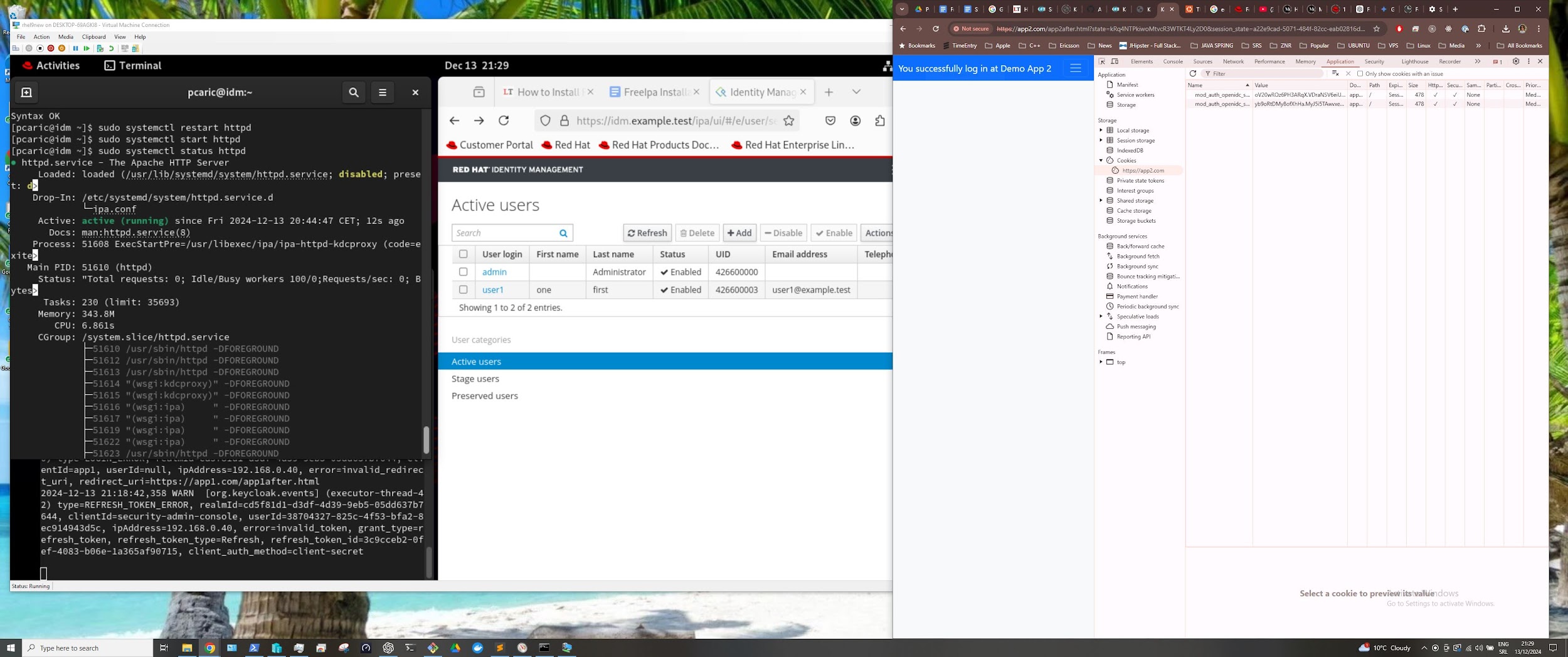
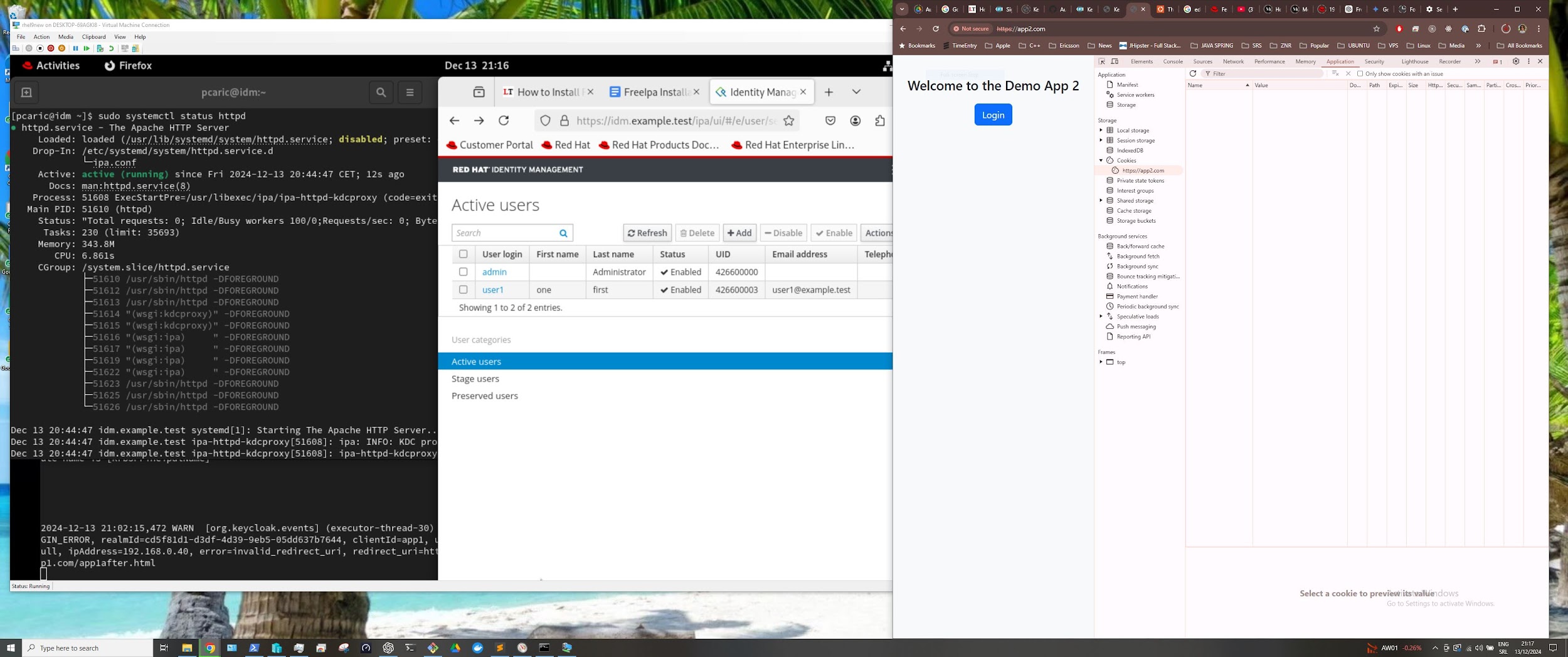
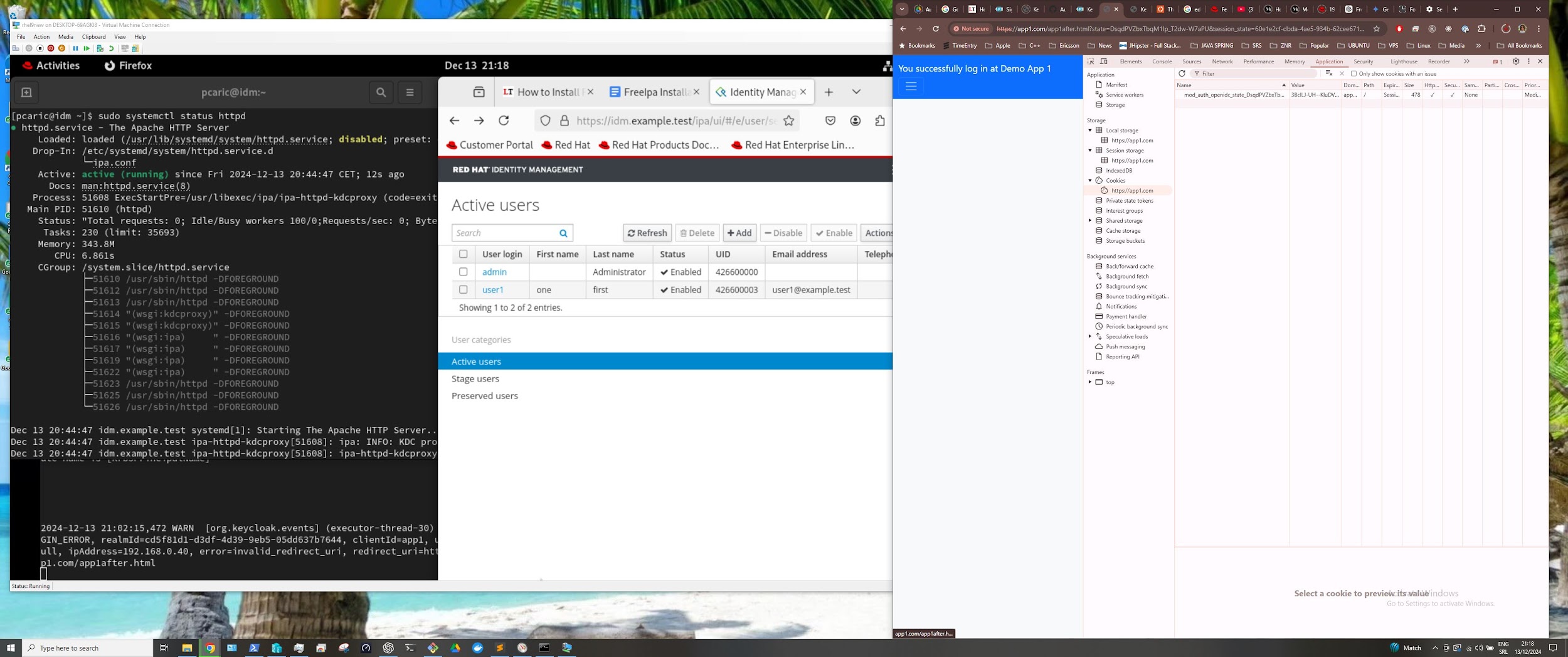
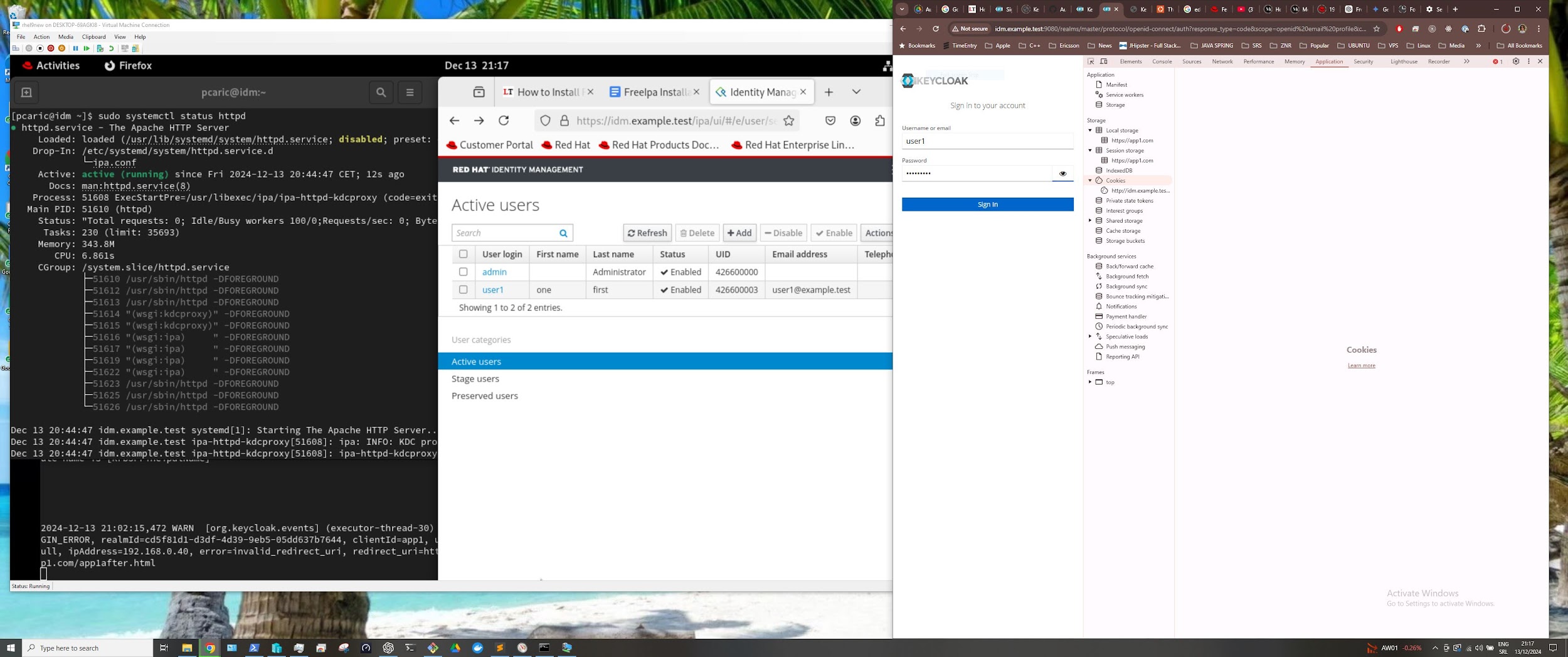
1. start Keyclaok in separate terminal: bin/kc.sh start-dev --http-port=9080
2. set keycloak proxy: sudo firewall-cmd --add-port=9080/tcp

**ON THE HOST MACHINE, WEB BROWSER**

1. access keycloak from the host: http://idm.example.test:9080
2. go to the <https://app1.com>
3. clock on the login button
4. insert credentials user1 / password1 (set before via Kerberos)
5. keycloak will use ldap to check the users’s credentials stored in the FreeIPA server
6. since user is valid keycloak will distribute token for authentication
7. token is saved in the browser’s storage - cookie
8. now go to the other app <https://app2.com> and click on the login button
9. this time user will not need to fill credentials, he will be forwarded to the app2 web







**Materials and links used for this**

* <https://centlinux.com/install-freeipa-on-rocky-linux/>
* <https://www.freeipa.org/page/Quick_Start_Guide>
* <https://infotechys.com/install_and_configure_ipa_idm_on_rhel_9/>
* <https://medium.com/devsecops-community/how-to-install-keycloak-on-linux-full-setup-dev-production-modes-and-ad-integration-6429d75fbd62>
* <https://www.reddit.com/r/FreeIPA/comments/y5icj2/keycloak_integration/>
* ChatGpt <https://chatgpt.com/> and Gemini <https://gemini.google.com/> for consulting and reviewing tutorials