IdM Security Foundations

IdM Security Foundations – Project Documentation

This project sets up a **FreeIPA-based Identity Management environment** in **GCP (Qatar region)**.

We'll build a **3-node cluster** (1 Primary + 2 Replicas), configure hostnames, DNS, and then install FreeIPA for centralized identity & access management.

Plan Overview

- **VPC** → `custom-vpc`
- **Subnet** → `custom-subnet` (`10.20.0.0/24`) in `me-central1`
- **Firewall** → Allow **SSH, HTTP, HTTPS** from anywhere (`0.0.0.0/0`)
- **VMs**:
    - `idm-primary` → `e2-standard-4` (4 vCPU, 16 GB RAM, 50 GB disk)
    - `idm-replica1` → `e2-standard-2` (2 vCPU, 8 GB RAM, 30 GB disk)
    - `idm-replica2` → `e2-standard-2` (2 vCPU, 8 GB RAM, 30 GB disk)
- **OS Image** → `centos-stream-9-v20250812` from `centos-cloud`

Step 1 — Create Network

1A. Create VPC + Subnet

```
gcloud compute networks create custom-vpc --subnet-mode=custom

gcloud compute networks subnets create custom-subnet \
  --network=custom-vpc \
  --region=me-central1 \
  --range=10.20.0.0/24
```

1B. Create Firewall Rule (Allow SSH/HTTP/HTTPS)

```
gcloud compute firewall-rules create allow-ssh-http-https \
  --network=custom-vpc \
  --allow tcp:22,tcp:80,tcp:443 \
  --source-ranges=0.0.0.0/0 \
  --description="Allow SSH, HTTP, HTTPS from anywhere"
```

Step 2 — Create Virtual Machines

Primary (idm-primary)

```
gcloud compute instances create idm-primary \
  --zone=me-central1-a \
  --machine-type=e2-standard-4 \
  --subnet=custom-subnet \
  --private-network-ip=10.20.0.10 \
  --image=centos-stream-9-v20250812 \
  --image-project=centos-cloud \
  --boot-disk-size=50GB \
  --tags=idm-server \
  --metadata=enable-oslogin=FALSE
```

Replica 1 (idm-replica1)

```
gcloud compute instances create idm-replica1 \
  --zone=me-central1-a \
  --machine-type=e2-standard-2 \
  --subnet=custom-subnet \
  --private-network-ip=10.20.0.11 \
  --image=centos-stream-9-v20250812 \
  --image-project=centos-cloud \
  --boot-disk-size=30GB \
  --tags=idm-server \
  --metadata=enable-oslogin=FALSE
```

Replica 2 (idm-replica2)

```
gcloud compute instances create idm-replica2 \
  --zone=me-central1-a \
  --machine-type=e2-standard-2 \
  --subnet=custom-subnet \
  --private-network-ip=10.20.0.12 \
  --image=centos-stream-9-v20250812 \
  --image-project=centos-cloud \
  --boot-disk-size=30GB \
  --tags=idm-server \
  --metadata=enable-oslogin=FALSE
```

Step 3 — Verification

List VMs:

```
gcloud compute instances list --filter="name~'idm-'"
```

SSH into primary:

```
gcloud compute ssh idm-primary --zone=me-central1-a
```

✅ Now we have **3 servers running in Qatar region** with **SSH/HTTP/HTTPS allowed**.

Step 4 — Common Base Setup (All 3 Servers)

We'll configure **system basics** on all nodes.

4A. Update System & Install Essentials

```
sudo dnf update -y
sudo dnf install -y vim chrony bash-completion firewalld
```

4B. Set Hostnames

```
# idm-primary
sudo hostnamectl set-hostname idm-primary.lab.local

# idm-replica1
sudo hostnamectl set-hostname idm-replica1.lab.local

# idm-replica2
sudo hostnamectl set-hostname idm-replica2.lab.local
```

Check:

```
hostnamectl
```

4C. Configure `/etc/hosts`

On **all servers**, edit:

```
sudo vim /etc/hosts
```

Final content:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

10.20.0.10   idm-primary.lab.local   idm-primary idm-primary.me-central1-a.c.windy-lyceum-464103-m5.internal
10.20.0.11   idm-replica1.lab.local  idm-replica1 idm-replica1.me-central1-a.c.windy-lyceum-464103-m5.internal
10.20.0.12   idm-replica2.lab.local  idm-replica2 idm-replica2.me-central1-a.c.windy-lyceum-464103-m5.internal

169.254.169.254 metadata.google.internal  # Added by Google
```

👉 **Why?**

- Ensures FreeIPA registers our `.lab.local` FQDNs.
- Keeps GCP internal hostnames as aliases for compatibility.

Verify:

```
hostname -f
getent hosts 10.20.0.10
getent hosts 10.20.0.11
getent hosts 10.20.0.12
```

4D. Time Sync (Chrony)

```
sudo systemctl enable --now chronyd
sudo chronyc sources
```

4E. Firewall Baseline

(We already opened GCP firewall, but configure OS-level too)

```
sudo systemctl enable --now firewalld
sudo firewall-cmd --permanent --add-service=ssh
sudo firewall-cmd --permanent --add-service=ntp
sudo firewall-cmd --reload
```

✅ At this point, all servers are correctly configured for FreeIPA installation.

Step 5 — Install FreeIPA on Primary

5A. Install Packages

```
sudo dnf install -y ipa-server ipa-server-dns bind-dyndb-ldap
```

5B. Run Installer

```
sudo ipa-server-install --setup-dns
```

Answer prompts:

- Server host name → `idm-primary.lab.local`
- Domain → `lab.local`
- Realm → `LAB.LOCAL`
- Directory Manager password → `CHANGE_ME_DM_PASS`
- IPA admin password → `CHANGE_ME_ADMIN_PASS`

- Configure integrated DNS → **Yes**
- DNS forwarders → `8.8.8.8`, `1.1.1.1`
- NetBIOS → Default `LAB`
- Configure chrony → **No**
- Confirm → **Yes**

5C. Verify Installation

```
kinit admin
ipa user-find
```

👉 If successful, you should be able to authenticate as `admin` and query users.

Step 6 — Enroll Replica Hosts (idm-replica1 & idm-replica2)

6A. Install FreeIPA Client Packages

On **each replica**:

```
sudo dnf install -y ipa-client
```

6B. Run the Client Installer

```
sudo ipa-client-install --mkhomedir \
  --server=idm-primary.lab.local \
  --domain=lab.local
```

Follow prompts:

- Proceed with fixed values → `yes`
- Configure chrony with NTP server/pool → `no`
- Confirm hostname, realm, domain, IPA server, BaseDN → `yes`
- Enter IPA admin credentials → `admin` + *IPA admin password*

6C. Verify Enrollment

```
# Check hostnames
hostname

# Verify host records in FreeIPA
ipa host-show idm-replica1.lab.local
ipa host-show idm-replica2.lab.local

# If Kerberos credentials are missing:
kinit admin
ipa host-show idm-replica1.lab.local
ipa host-show idm-replica2.lab.local
```

✅ Both replicas should now be enrolled with the primary IPA server.

Step 7 — Access FreeIPA Web UI

7A. Prepare SSH Tunnel

From **Windows PowerShell**:

```
ssh -i C:\Users\path\to\openssh-file -L 443:idm-primary.lab.local:443 <idm-primary-username>@34.18.124.109
```

Explanation:

- `i <path>` → Path to OpenSSH private key
- `L 443:idm-primary.lab.local:443` → Forward local port `443` to FreeIPA server's HTTPS
- `<idm-primary-username>@34.18.124.109` → Your GCP VM login

⚠️ Keep this terminal **open** — it maintains the tunnel.

7B. Map the Hostname (Optional but Recommended)

Edit Windows hosts file (**Run Notepad as Admin**):

```
C:\Windows\System32\drivers\etc\hosts
```

Add:

```
127.0.0.1   idm-primary.lab.local
```

📌 Ensures browser resolves `idm-primary.lab.local` correctly during HTTPS redirects.

7C. Open Web UI

In your browser:

```
https://idm-primary.lab.local/ipa/ui
```

- Accept the **self-signed cert warning**
- Login page should load

7D. Sign In

- **Username:** `admin`
- **Password:** IPA admin password (set during `ipa-server-install`)

⚠️ This is **not** your Linux root/GCP/SSH password.

7E. Troubleshooting

- Verify tunnel:

```
netstat -ano | findstr 443
```

- If FreeIPA redirects to FQDN but browser fails → ensure `hosts` file mapping is in place.
- If login fails → confirm `kinit admin` works on the VM.

Step 8 — Create Groups & Users

8A. Create Groups

1. Login to Web UI → **Identity → Groups** → **Add**
2. Create groups:

Group Name	Description
admins	Full access users (IdM admins)
devs	Development team
finance	Finance team

8B. Create Users

1. Navigate → **Identity → Users → Add**
2. Add these users:

Username	First Name	Last Name	Group	Email
carol-admin	Carol	Admin	admins	carol@lab.local
alice-dev	Alice	Dev	devs	alice@lab.local
bob-finance	Bob	Finance	finance	bob@lab.local

1. **Set initial password** → e.g., `Lab1234!`

- Uncheck **“User must change password at next login”** for testing.

Step 9 — Verification (CLI Dumps)

Run on **idm-primary**:

```
echo "==================== HOSTS ===================="
ipa host-find

echo -e "\n==================== HOST GROUPS ===================="
ipa hostgroup-find

echo -e "\n==================== USERS ===================="
ipa user-find

echo -e "\n==================== GROUPS ===================="
ipa group-find

echo -e "\n==================== SUDO RULES ===================="
ipa sudorule-find
for rule in $(ipa sudorule-find --all | awk '/Rule name:/ {print $3}'); do
    ipa sudorule-show $rule --all
done

echo -e "\n==================== HBAC SERVICES ===================="
ipa hbacsvc-find

echo -e "\n==================== HBAC RULES ===================="
ipa hbacrule-find
for rule in $(ipa hbacrule-find --all | awk '/Rule name:/ {print $3}'); do
```

```
    ipa hbacrule-show $rule --all
done

echo -e "\n==================== KERBEROS TICKETS ===================="
klist
```

✅ Expected Outputs (Samples)

<details>
<summary>Hosts</summary>

```
---------------
3 hosts matched
---------------
Host name: idm-primary.lab.local
Principal: host/idm-primary.lab.local@LAB.LOCAL
...
Host name: idm-replica1.lab.local
Platform: x86_64
OS: 5.14.0-603.el9.x86_64
...
Host name: idm-replica2.lab.local
Platform: x86_64
OS: 5.14.0-603.el9.x86_64
...
Number of entries returned 3
```

</details>
<details>
<summary>Groups & Users</summary>

```
6 groups matched:
- admins → Full access
- devs → Development team
- finance → Finance team
...

4 users matched:
- admin
- carol-admin
- alice-dev
- bob-finance
```

</details>
<details>
<summary>SUDO & HBAC Rules</summary>

```
1 Sudo Rule matched:
- admins_all_sudo → Full sudo access for admins

4 HBAC Rules matched:
- allow_all → allow all
- allow_systemd-user → allow pam_systemd
- ssh_alice_only_replica2 → only alice-dev can SSH replica2
- ssh_bob_only_replica1 → only bob-finance can SSH replica1
```

</details>
<details>
<summary>Kerberos Tickets</summary>

```
Ticket cache: KCM:1001
Default principal: admin@LAB.LOCAL

Valid starting: 08/28/2025 09:24:03
```

```
Expires:       08/29/2025 09:08:38
Service principal: HTTP/idm-primary.lab.local@LAB.LOCAL
```

</details>

✅ At this stage, you have:

- Primary + replicas enrolled
- Web UI accessible
- Groups & users created
- SUDO + HBAC rules enforced
- Verified via CLI & Kerberos