FreePBX and SIP Lab

Overview

Project Oyster includes a dedicated FreePBX virtual machine for learning and testing IP telephony. FreePBX provides a web-based management interface for Asterisk, which handles telephony services and call processing.

The lab currently supports internal SIP calls between two Yealink T43U phones.

Virtual Machine

- **VM name:** `FreePBX`
- **IP address:** `192.168.90.7`
- **Web access:** `http://freepbx.oyster.lab` or the configured NPM hostname
- **SIP signalling:** UDP 5060, subject to the active PJSIP configuration

Confirm the actual configured ports and transport before troubleshooting or making firewall changes.

Extensions and Endpoints

Extension	Display Name	Endpoint
100	Test1	Yealink T43U
101	Test2	Yealink T43U

The two phones have been provisioned using MAC-specific configuration files hosted by the FreePBX VM.

Provisioning directory:

```
/var/www/html/provisioning/
```

The filenames are case-sensitive on Linux. For example, the configuration file for MAC address `80:5E:0C:73:60:37` is:

```
805e0c736037.cfg
```

Extension passwords and other secrets must not be stored in this document.

Basic Call Flow

```
Yealink T43U (Extension 100)
           |
           | SIP signalling
           v
        FreePBX
           ^
           | SIP signalling
           |
Yealink T43U (Extension 101)
```

SIP signalling establishes and controls the call. RTP typically carries the actual voice media. A call may therefore connect successfully while audio fails if RTP traffic is blocked or incorrectly routed.

Useful Troubleshooting Commands

Check Asterisk service status on the FreePBX VM:

```
sudo systemctl status asterisk
```

Start Asterisk if it is stopped:

```
sudo systemctl start asterisk
```

Review recent service logs:

```
sudo journalctl -u asterisk -n 100 --no-pager
```

Check whether UDP 5060 is listening:

```
sudo ss -lunp | grep 5060
```

These commands should be run inside the FreePBX VM, not on `infra01` or the Proxmox host.

Known Recovery Observation

During a previous network interruption, the FreePBX VM became reachable again but Asterisk remained inactive. Starting Asterisk manually restored telephony service.

This indicates a recovery behaviour that needs further investigation. Verify why Asterisk did not start automatically after the interruption or restart, and test the service's startup behaviour after making any changes.

Future Work

- Obtain authorized test DID and SIP trunk details from the provider.
- Configure and test inbound and outbound calls.
- Document the SIP transport, codecs, RTP port range, and number format once confirmed.
- Verify Asterisk starts automatically after a VM reboot.
- Back up the FreePBX VM before major telephony changes.
- Keep SIP credentials and extension secrets out of documentation and screenshots.

Security Notes

Do not expose SIP or RTP ports to the public internet without an explicit design and appropriate restrictions. Use test numbers and provider-approved credentials only. Keep the lab isolated from production telephony unless a connection has been deliberately authorized.

Revision #1
Created 2026-10-09 14:09:10 UTC by Admin
Updated 2026-10-09 14:09:28 UTC by Admin