Gua paso a paso: Implementacin de SalasJuntas en Apache (Linux)

Objetivo
Implementar la solucin completa de **SalasJuntas** en un servidor Linux usando **Apache** como servidor web p
ara el frontend y **reverse proxy** para la API Node.js.

---

1. Requisitos previos

1.1 Infraestructura
- Servidor Linux (Ubuntu/Debian recomendado).
- Acceso con usuario con permisos sudo.
- Dominio apuntando al servidor (ejemplo: ‘salas.midominio.com‘).

1.2 Software requerido
- Apache 2.4+
- Node.js 20+
- npm 10+
- systemd
- rsync

Instalar paquetes base:

‘‘‘bash
sudo apt update
sudo apt install -y apache2 nodejs npm rsync
‘‘‘

Habilitar mdulos Apache necesarios:

‘‘‘bash
sudo a2enmod proxy proxy_http rewrite headers ssl
‘‘‘

---

2. Obtener el cdigo

Crear carpeta de despliegue y clonar:

‘‘‘bash
sudo mkdir -p /opt/salasjuntas
sudo chown -R $USER:$USER /opt/salasjuntas
cd /opt
git clone <URL_DEL_REPOSITORIO> salasjuntas
cd /opt/salasjuntas
‘‘‘

Si no usas git, copia el cdigo manualmente a ‘/opt/salasjuntas‘.

---

3. Configuracin de Azure AD / Microsoft Graph

1. Crear o usar una App Registration en Azure Portal.
2. Configurar Redirect URI tipo Web:
   - Produccin: ‘https://TU_DOMINIO/‘
3. Crear ‘Client Secret‘.
4. Asignar permisos Graph:

   - ‘Calendars.Read‘
   - ‘User.Read‘
   - ‘Places.Read.All‘
5. Ejecutar **Grant admin consent**.

Guardar los valores:
- Tenant ID
- Client ID
- Client Secret

---

4. Configurar variables de entorno

Crear ‘.env‘ desde el ejemplo:

“‘bash
cd /opt/salasjuntas
cp .env.example .env
nano .env
“‘

Ejemplo recomendado para produccin:

“‘env
PORT=4000
FRONTEND_URL=https://TU_DOMINIO
GRAPH_ENABLED=true
TIMEZONE=America/Mexico_City
AZURE_TENANT_ID=<tenant_id>
AZURE_CLIENT_ID=<client_id>
AZURE_CLIENT_SECRET=<client_secret>
AZURE_REDIRECT_URI=https://TU_DOMINIO/
ROOM_EMAILS=sala1@empresa.com,sala2@empresa.com
“‘

Notas:
- ‘GRAPH_ENABLED=true‘ usa Microsoft Graph real.
- ‘GRAPH_ENABLED=false‘ usa datos mock.

---

5. Verificar integridad del repositorio (recomendado)

Antes de instalar dependencias, valida que no existan conflictos de merge ni JSON invlido:

“‘bash
cd /opt/salasjuntas
node scripts/validate-repo.mjs
“‘

Si el script reporta conflictos (‘<<<<<<<‘, ‘=======‘, ‘>>>>>>>‘) o JSON invlido, corrige esos archivos y vue lve a ejecutar.

---

6. Instalar dependencias y construir frontend

“‘bash
cd /opt/salasjuntas

npm install
npm run build -w frontend
‘‘‘

Si obtienes ‘npm ERR! EJSONPARSE‘ en ‘frontend/package.json‘:

1. Edita el archivo y elimina marcadores de conflicto:
   - ‘<<<<<<< ...‘
   - ‘=======‘
   - ‘>>>>>>> ...‘
2. Deja el archivo como JSON vlido (sin comentarios y con comillas dobles).
3. Ejecuta nuevamente:

‘‘‘bash
node scripts/validate-repo.mjs
npm install
‘‘‘

---

7. Publicar frontend esttico en Apache

‘‘‘bash
sudo mkdir -p /var/www/salasjuntas/current
sudo rsync -av --delete /opt/salasjuntas/frontend/dist/ /var/www/salasjuntas/current/
‘‘‘

La app incluye reglas ‘.htaccess‘ para rutas SPA.

---

8. Configurar API Node con systemd

Copiar unidad de servicio:

‘‘‘bash
sudo cp /opt/salasjuntas/deploy/systemd/salasjuntas-api.service /etc/systemd/system/salasjuntas-api.service
‘‘‘

Revisar rutas y usuario (si aplica):

‘‘‘bash
sudo nano /etc/systemd/system/salasjuntas-api.service
‘‘‘

Activar y arrancar servicio:

‘‘‘bash
sudo systemctl daemon-reload
sudo systemctl enable --now salasjuntas-api
sudo systemctl status salasjuntas-api
‘‘‘

Ver logs en vivo:

‘‘‘bash
journalctl -u salasjuntas-api -f
‘‘‘

---

9. Configurar VirtualHost Apache

Copiar archivo base:

‘‘‘bash
sudo cp /opt/salasjuntas/deploy/apache/salasjuntas.conf /etc/apache2/sites-available/salasjuntas.conf
‘‘‘

Editar y ajustar dominio/rutas:

‘‘‘bash
sudo nano /etc/apache2/sites-available/salasjuntas.conf
‘‘‘

Verifica:
- ‘ServerName‘ correcto.
- ‘DocumentRoot /var/www/salasjuntas/current‘
- ‘ProxyPass /api http://127.0.0.1:4000/api‘
- ‘ProxyPassReverse /api http://127.0.0.1:4000/api‘

Activar sitio y recargar Apache:

‘‘‘bash
sudo a2dissite 000-default.conf
sudo a2ensite salasjuntas.conf
sudo apachectl configtest
sudo systemctl reload apache2
‘‘‘

---

10. Habilitar HTTPS con Let?s Encrypt (recomendado)

‘‘‘bash
sudo apt install -y certbot python3-certbot-apache
sudo certbot --apache -d TU_DOMINIO
sudo certbot renew --dry-run
‘‘‘

---

11. Validacin final

1. Abrir ‘https://TU_DOMINIO‘.
2. Validar que cargue la UI y selector de salas.
3. Validar endpoint salud:

‘‘‘bash
curl -i https://TU_DOMINIO/api/health
‘‘‘

4. Si Graph est activo, probar botn **Conectar Microsoft 365**.
5. Confirmar que se listan eventos reales de salas.

---

12. Operacin y mantenimiento

Reiniciar API:

‘‘‘bash
sudo systemctl restart salasjuntas-api
‘‘‘

Ver estado API:

‘‘‘bash
sudo systemctl status salasjuntas-api
‘‘‘

Logs de API:

‘‘‘bash
journalctl -u salasjuntas-api -f
‘‘‘

Recargar Apache:

‘‘‘bash
sudo systemctl reload apache2
‘‘‘

---

13. Troubleshooting rpido

401 en ‘/api/rooms‘
- Verificar login OAuth y token vigente.
- Verificar ‘GRAPH_ENABLED=true‘ y credenciales vlidas.

‘redirect_uri_mismatch‘
- ‘AZURE_REDIRECT_URI‘ debe coincidir exactamente con Azure.

404 al refrescar rutas frontend
- Confirmar ‘.htaccess‘ en ‘/var/www/salasjuntas/current/‘.
- Confirmar ‘AllowOverride All‘ en Apache.
- Confirmar ‘mod_rewrite‘ habilitado.

API no levanta con systemd
- Revisar ‘journalctl -u salasjuntas-api -f‘.
- Revisar ruta de Node en ‘ExecStart‘.
- Revisar formato de ‘/opt/salasjuntas/.env‘.