SSP - Self Service Password

Sistema de troca de senha para serviços integrados ao LDAP

- Implantação do LDAP Tool Box Self Service Password (SSP) via Docker

Implantação do LDAP Tool Box Self Service Password (SSP) via Docker

IFSP Salto — VM `10.114.50.200` (Ubuntu Server 22.04) | LDAP: `ldap.slt.ifsp.edu.br` (`10.114.50.57`)

Este guia cobre: instalação via Docker, LDAP, captcha, recuperação de senha por e-mail, bloqueio por tentativas e integração com Postfix. Baseado na documentação oficial (instalação e Docker Hub).

> A instalação do Docker foi desconsiderada, conforme solicitado — a VM já o possui.

---

1. Visão geral da arquitetura

```
[Usuário] --HTTPS--> [Nginx no host, TLS]
                            |
                            v
                   [container ssp:80] --LDAP(389)--> ldap.slt.ifsp.edu.br (10.114.50.57)
                            |
                            v (rede interna Docker "ssp_net", porta 587)
                   [container postfix] --> relay/smarthost institucional
```

A imagem oficial do SSP só expõe a aplicação em HTTP (porta 80) internamente — não há suporte documentado a TLS dentro do container. Por isso, o guia usa um **reverse proxy no host** (Nginx) para TLS. O Postfix roda em **outro container** (`boky/postfix`), na mesma rede Docker `ssp_net` do SSP — os dois se falam pelo nome do serviço, sem precisar tocar no host. É a imagem mais usada e madura para "Postfix como relay em container" (766+ estrelas no GitHub, focada exatamente

nesse caso de uso — enviar e-mails de dentro do Docker sem gerenciar um MTA completo).

---

2. Estrutura de diretórios no host

bash

```
sudo mkdir -p /opt/ssp/{conf,data/ratelimit,data/postfix-spool}
cd /opt/ssp
```

Copie os arquivos `docker-compose.yml` e `conf/config.inc.local.php` (anexos) para dentro dessa estrutura.

---

3. docker-compose.yml

Veja o arquivo `docker-compose.yml` anexo. Pontos-chave:

- Monta `./conf` em `/var/www/conf` (caminho oficial de configuração da imagem).
- Monta `./data/ratelimit` em `/var/www/ratelimit`, um diretório **gravável** para persistir o controle de bloqueio por tentativas entre reinícios do container (por padrão ele usaria `/tmp`, que é perdido a cada restart).
- Sobe também o serviço `postfix` (imagem `boky/postfix`), na mesma rede `ssp_net` — é para ele que o SSP manda os e-mails de recuperação (detalhes na seção 8).
- Publica apenas `0.0.0.0:8090:80` do serviço `ssp` — a aplicação só fica acessível localmente; quem expõe para a rede é o Nginx com TLS. O Postfix não publica nenhuma porta para o host, só é alcançável de dentro da rede `ssp_net`.

Ajuste permissões do diretório de rate-limit para o usuário do Apache dentro do container (normalmente `www-data`, uid `33` na imagem padrão; `82` na `alpine-latest`):

bash

```
# confira o uid depois de subir o container:
docker exec -it ssp id www-data
# depois ajuste no host, por exemplo para uid 33:
sudo chown -R 33:33 /opt/ssp/data/ratelimit
```

Subir o serviço:

bash

```
cd /opt/ssp
docker compose up -d
docker compose logs -f
```

---

4. Configuração do SSP (`conf/config.inc.local.php`)

Veja o arquivo `config.inc.local.php` anexo — já preenchido com os dados que você passou. **Antes de subir em produção, ajuste obrigatoriamente:**

Parâmetro	O que fazer
`$ldap_bindpw`	Senha real da conta `cn=admin,dc=slt,dc=ifsp,dc=local`. **Não deixe em texto puro no repositório** — veja a nota de segurança abaixo.
`$ldap_filter`	Confirme o `objectClass` real dos usuários no seu diretório (`posixAccount`, `inetOrgPerson`, etc.)
`$mail_attributes`	Confirme o nome do atributo de e-mail no schema (geralmente `mail`)
—	(nenhum pendente no relay de e-mail — credencial já confirmada e preenchida, seção 8)

Sobre `$ldap_binddn` / `$who_change_password = "manager"`

Como a recuperação por e-mail (token) precisa trocar a senha **sem o usuário informar a senha antiga**, o SSP precisa de uma conta com poderes de administrador — por isso usamos `cn=admin,...` como bind e `$who_change_password = "manager"`. Isso é obrigatório para a funcionalidade de recuperação por e-mail funcionar.

Nota de segurança sobre a senha do admin LDAP

Evite deixar `$ldap_bindpw` em texto puro no arquivo versionado. Duas opções:

1. Restrinja permissões do arquivo (`chmod 640`, dono root, grupo do serviço) e mantenha fora de qualquer repositório Git.
2. Ou use uma conta de serviço LDAP dedicada, com permissão apenas de escrita no atributo de senha (`userPassword`) dos usuários da OU relevante — mais seguro do que usar o `cn=admin` completo. Recomendo fortemente essa segunda opção para produção.

---

5. Captcha

Já habilitado no `config.inc.local.php`:

php

```
$use_captcha = true;
$captcha_class = "InternalCaptcha";
```

O `InternalCaptcha` é gerado localmente via `php-gd` (já incluso na imagem oficial) e **não depende de nenhum serviço externo** — ideal para uma rede interna do campus. Ele aparece em todos os formulários (troca de senha, token de recuperação, perguntas). Se no futuro quiser Google reCAPTCHA ou Friendly Captcha, basta trocar `$captcha_class` — os parâmetros extras estão comentados no fim do arquivo de config para referência.

---

6. Recuperação de senha por e-mail

Já configurado:

php

```
$use_tokens = true;
$mail_address_use_ldap = true;   // busca o e-mail direto no LDAP, não pergunta ao usuário
$crypt_tokens = true;
$token_lifetime = "3600";        // link válido por 1h
$obscure_usernotfound_sendtoken = true;  // não revela se o login existe (evita enumeração de contas)
```

Fluxo: usuário informa o login → SSP busca o e-mail cadastrado no atributo `mail` do LDAP → envia um link com token → usuário clica e define nova senha.

---

7. Bloqueio por tentativas

O SSP tem um mecanismo de rate limit por login **e** por IP:

php

```
$use_ratelimit = true;
$ratelimit_dbdir = '/var/www/ratelimit';  // volume persistente (seção 3)
$max_attempts_per_user = 3;   // bloqueia após 3 tentativas do mesmo login
$max_attempts_per_ip = 5;     // bloqueia após 5 tentativas do mesmo IP
$max_attempts_block_seconds = "300";  // bloqueio de 5 minutos
```

Isso cobre a troca de senha, o envio de token por e-mail e (se usado) o envio por SMS/perguntas — protegendo contra força bruta e contra abuso do envio de e-mails de recuperação.

Importante: esse controle vive em arquivos temporários dentro do container (por isso montamos um volume dedicado). Se você rodar múltiplas réplicas do SSP atrás de um load balancer no futuro, vai precisar de sessões persistentes (sticky sessions) ou migrar para outra estratégia — não é o caso aqui com um único container.

Um bloqueio complementar, no lado do LDAP (via overlay `ppolicy` do OpenLDAP, atributo `pwdFailureTime`), pode ser adicionado depois habilitando `$ldap_use_ppolicy_control = true;` — mas isso exige configurar o overlay `ppolicy` no próprio servidor `ldap.slt.ifsp.edu.br`, o que está fora do escopo desta VM. O rate limit do SSP já atende ao requisito pedido.

---

8. Integração com Postfix (também em container)

O `docker-compose.yml` já inclui um segundo serviço, `postfix`, usando a imagem `boky/postfix` — um relay Postfix "pronto para uso" pensado exatamente para rodar ao lado de outra aplicação em Docker. Ele **não** expõe porta nenhuma para fora: o container `ssp` fala com ele direto pelo nome do serviço (`postfix`) dentro da rede `ssp_net`.

O relay externo já está configurado com os dados oficiais do manual de e-mail do IFSP — "Configurações SMTP para e-mail de sistemas (não-responda.ifsp.edu.br)":

Parâmetro	Valor
Host	`nao-responda.ifsp.edu.br`
Porta	`587` (STARTTLS) — preferida pelo manual; `465` (SSL/TLS implícito) é o fallback
Autenticação	usuário `slt` (confirmado pelo sistema legado `email.class.php`, que já enviava com sucesso por esse relay)
Segurança	TLS obrigatório

O que já está configurado no `docker-compose.yml`

yaml

```
postfix:
  image: boky/postfix:latest
  environment:
    - ALLOWED_SENDER_DOMAINS=nao-responda.ifsp.edu.br
    - POSTFIX_myhostname=ssp-postfix.slt.ifsp.edu.br
    - RELAYHOST=[nao-responda.ifsp.edu.br]:587
    - RELAYHOST_USERNAME=slt
    - RELAYHOST_PASSWORD_FILE=/run/secrets/relayhost_password
    - POSTFIX_smtp_tls_security_level=encrypt
  secrets:
    - relayhost_password
  volumes:
    - ./data/postfix-spool:/var/spool/postfix
  networks:
    - ssp_net

secrets:
  relayhost_password:
    file: ./secrets/relayhost_password.txt
```

- `ALLOWED_SENDER_DOMAINS=nao-responda.ifsp.edu.br`: precisa bater com o domínio da conta autenticada — é esse domínio que o relay aceita como remetente válido vindo deste Postfix.
- **Senha da conta** `slt` — já preenchida no `secrets/relayhost_password.txt` anexo (extraída do `email.class.php` do sistema legado). Ela fica fora do `docker-compose.yml` (que pode ir para um repositório) e é passada via **Docker secret**:

bash

```
cd /opt/ssp
mkdir -p secrets
# copie o secrets/relayhost_password.txt anexo para dentro dessa pasta
chmod 600 secrets/relayhost_password.txt
echo "secrets/relayhost_password.txt" >> .gitignore
```

> **Importante:** essa senha está em texto puro no arquivo — trate como qualquer credencial de produção. Restrinja o acesso ao diretório `/opt/ssp/secrets` (só root), nunca comite em repositório público, e considere pedir à equipe de e-mail do IFSP para rotacioná-la já que estava hardcoded num sistema legado (`email.class.php`) — se aquele arquivo circulou por outros lugares, a senha pode já estar exposta.

- **Porta 465 (fallback)**: se a 587 estiver bloqueada na rede do campus, troque `RELAYHOST=[nao-responda.ifsp.edu.br]:465` e adicione mais uma variável (a imagem `boky/postfix` aceita qualquer opção do Postfix via `POSTFIX_<nome>`):

yaml

```
  - POSTFIX_smtp_tls_wrappermode=yes
```

(465 é SSL/TLS implícito desde a conexão — sem esse modo o Postfix tentaria STARTTLS na 465 e falharia.)

- `mynetworks` já cobre por padrão as faixas privadas usuais do Docker (`10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`), então normalmente não é preciso mexer nisso.

Alinhamento do remetente (`$mail_from`)

Já ajustado no `config.inc.local.php` anexo:

php

```
$mail_from = "slt@nao-responda.ifsp.edu.br";
```

Servidores institucionais (o IFSP usa Zimbra) costumam rejeitar ou sobrescrever o campo "From" quando ele não corresponde à conta autenticada no SMTP, a menos que exista uma permissão de "enviar como" liberada para outro endereço. Se quiserem manter um remetente mais amigável como `senha@slt.ifsp.edu.br`, é preciso confirmar com a equipe de e-mail do IFSP se essa permissão pode ser concedida para a conta `slt@nao-responda.ifsp.edu.br`.

Subindo e testando

bash

```
cd /opt/ssp
docker compose up -d
docker compose logs -f postfix
```

Teste de envio isolado, de dentro do container SSP:

bash

```
docker compose exec ssp sh -c "apk add --no-cache curl 2>/dev/null; \
  echo -e 'Subject: Teste SSP\n\nMensagem de teste' | \
  curl -s smtp://postfix:587 --mail-from slt@nao-responda.ifsp.edu.br --mail-rcpt
seuemail@slt.ifsp.edu.br --upload-file -"
```

Ou, mais simples, direto no container do Postfix:

bash

```
docker compose exec postfix sh -c "echo 'Teste SSP' | mail -s 'Teste'
seuemail@slt.ifsp.edu.br"
docker compose logs postfix
```

Se a autenticação falhar, o log do Postfix mostra o motivo (usuário/senha incorretos, TLS recusado, etc.) — confira `docker compose logs postfix` logo após o teste.

Como o container SSP acessa o Postfix

No `config.inc.local.php` (já ajustado no arquivo anexo):

php

```
$mail_smtp_host = 'postfix';   // nome do serviço no docker-compose.yml
$mail_smtp_port = 587;         // porta de submission da imagem boky/postfix (interna, entre
os dois containers)
$mail_smtp_auth = false;
```

Essa porta 587 é **interna** (SSP → container Postfix, sem senha, protegida só pela rede Docker) — não confundir com a porta 587 do relay externo (`nao-responda.ifsp.edu.br`, essa sim autenticada com TLS).

Como os dois containers estão na mesma rede `ssp_net`, o Docker resolve o nome `postfix` automaticamente via DNS interno — não é preciso IP fixo nem `extra_hosts`.

Persistência da fila

O volume `./data/postfix-spool` guarda a fila de e-mails do Postfix entre reinícios do container (evita perder mensagens em trânsito num restart). Crie o diretório antes de subir:

bash

```
mkdir -p /opt/ssp/data/postfix-spool
```