

PRÉ-REQUISITOS · MIGRAÇÃO

Pré-requisitos: Migração Azure DevOps para GitLab™ self-managed

Documento enviado após a escolha do serviço PS-MIG-03; o que o cliente provisiona, as credenciais que entrega à Pointer, as informações a enviar e o checklist de prontidão.

CÓDIGO**PR-MIG-03****TIPO****Pré-requisitos****VERSÃO****1.2****DATA****1 de outubro de 2026**

O3S Consultoria e Tecnologia da Informação Ltda.
contato@pointertech.digital · +55 (61) 99874-4648 · pointertech.digital

GITLAB is a trademark of GitLab Inc. in the United States and other countries and regions.

**Select
Partner**

PRÉ-REQUISITOS · PR-MIG-03 · OFERTA PS-MIG-03

Pré-requisitos: Migração Azure DevOps para GitLab self-managed

Documento enviado ao cliente depois da escolha do serviço **PS-MIG-03 — Migração Azure DevOps para GitLab self-managed**. Lista o que o cliente provisiona, as credenciais que entrega, as informações a enviar à Pointer, o checklist de prontidão e a matriz do que é migrado. O prazo de devolução das informações e do checklist é definido no kick-off.

Os itens do checklist marcados como **Preflight automático** são conferidos pelo pipeline Pointer contra a instância de destino real antes da primeira onda; os demais são conferidos na reunião de prontidão ou pelo próprio cliente. Nenhuma onda é executada com erro no preflight.

Conteúdo

Origem e destino suportados

Configuração da instância de destino

Organização e projetos no Azure DevOps

Host de migração

Runner do engajamento

Rede

Usuários e contribuições

Governança da migração

Critérios de aceite por onda

Entrega das credenciais

Credenciais que o cliente entrega à Pointer

Informações a enviar

Matriz de itens migrados

Checklist de prontidão

1. Origem e destino suportados

Item	Suportado
Origem	Azure DevOps Services, repositórios Git. Azure DevOps Server: sob consulta (escopo e condições definidos em proposta específica, com execução piloto). Repositórios TFVC: fora do escopo
Destino	Somente GitLab self-managed do cliente, inclusive instância implantada pela Pointer
Método	A plataforma GitLab não tem importador nativo de Azure DevOps. O GitLab Congregate lê o Azure DevOps no host de migração (APIs e Git), monta arquivos de exportação no formato GitLab e a instância de destino os importa por arquivo
Estrutura no destino	Grupo pai + projeto do Azure DevOps + repositório; o grupo do projeto é criado pelo pipeline antes da importação
Versões	Azure DevOps Services: serviço. Azure DevOps Server: versões definidas na proposta e conferidas na execução piloto
Destinos fora do escopo	GitLab.com e GitLab Dedicated como destino: roadmap
Outras origens	GitLab (PS-MIG-01), GitHub (PS-MIG-02), Bitbucket Cloud (PS-MIG-04); Bitbucket Server/Data Center e AWS CodeCommit: roadmap

- Com origem Azure DevOps não há arquivamento da origem, reescrita de referências nos repositórios nem contagem independente origem × destino; o pipeline verifica cada importação pelo status e pelas estatísticas da importação.
- Azure DevOps Server: o runbook do GitLab Congregate prevê uma aplicação própria para listar os usuários.

2. Configuração da instância de destino

Item	Especificação	Obrigatório
Fonte de importação <i>GitLab export</i>	Habilitada nas configurações de importação e exportação da instância (<i>Admin → Settings → General → Import and export settings</i> , fontes de importação). Em GitLab self-managed nenhuma fonte de importação vem habilitada por padrão. O preflight reprova se ausente	Sim
Grupo pai	Grupo existente, privado ou interno, que recebe a migração. O preflight reprova grupo inexistente ou público	Sim
Token de administrador	A importação por arquivo com associação de autoria por e-mail e a criação de usuários exigem token de administrador no destino	Sim

3. Organização e projetos no Azure DevOps

- O e-mail de cada usuário no Azure DevOps (o GitLab Congregate usa o `mailAddress` da Graph API) é igual ao e-mail do usuário no destino.
- O GitLab Congregate lista só os usuários dos times do projeto no Azure DevOps.
- Por padrão são migrados todos os pull requests e todos os work items de backlog de cada projeto da onda; o pipeline não aplica filtro de data.
- O GitLab Congregate clona os repositórios por Git com o PAT; se o clone com PAT falhar, o runbook do GitLab Congregate orienta um *credential helper* no host de migração.
- Variáveis secretas e variable groups ligados ao Azure Key Vault não são migrados e são recriados no destino.
- Durante a janela da onda, os repositórios e work items da onda não recebem alterações no Azure DevOps.

4. Host de migração

Item	Especificação	Obrigatório
Máquina	Linux x86_64 dedicada ao engajamento, sem outros serviços; o usuário do runner participa do grupo <code>docker</code> , o que equivale a root no host. Um host por cliente	Sim
Recursos de partida	4 vCPU, 16 GB de RAM e 100 GB de disco SSD (imagem do GitLab Congregate com cerca de 3,3 GB; o banco MongoDB da ferramenta cresce com a listagem da origem). A documentação do GitLab Congregate (página <i>Migration Workflows</i>) pede, no mínimo, ambiente com Docker e 4 CPU, 8 GB de RAM e 20 GB de disco	Sim
Contêineres	Docker Engine e Docker Compose v2; a ausência reprova o início de cada job	Sim
Swap	2 GB de swap, conforme a documentação do GitLab Congregate	Não
Disco para exportações	Os arquivos de exportação são montados no host; disco dimensionado para os maiores repositórios da onda, conforme a orientação da documentação do GitLab Congregate de dimensionar o disco para as exportações	Sim
Certificados	Certificado válido no destino. CA corporativa em arquivo no host (arquivo inexistente reprova o job), somada às raízes do contêiner do GitLab Congregate	Condicional: CA interna
Proxy	Proxy de saída informado no plano de migração e configurado também no daemon Docker	Condicional: proxy corporativo
Acompanhamento	A interface do GitLab Congregate e o Flower ficam só em <code>127.0.0.1</code> do host; o acompanhamento é por túnel SSH (usuário@host)	Não

- No encerramento, o pipeline remove a stack e todos os dados do engajamento no host (listagens, exportações, logs e resultados).

5. Runner do engajamento

Item	Especificação	Obrigatório
GitLab Runner	Instalado no host de migração, executor shell, registrado no projeto do engajamento na GitLab.com da Pointer com o token de registro criado nesse projeto	Sim
Configuração	Runner de projeto com a tag do engajamento; <i>Run untagged jobs</i> desmarcado; <i>Protected e Lock to current projects</i> marcados; tempo máximo de job de 8 horas ou mais	Sim
Usuário do runner	<code>gitlab-runner</code> no grupo <code>docker</code> ; em Ubuntu, remover <code>/home/gitlab-runner/.bash_logout</code> (o <code>clear_console</code> encerra jobs shell)	Sim

Item	Especificação	Obrigatório
Encerramento	Runner removido do projeto e desinstalado do host	Sim

6. Rede

Conexão	Protocolo e observação	Obrigatório
Host de migração → Azure DevOps (APIs e repositórios Git)	HTTPS: <code>https://dev.azure.com/<organização></code> no Azure DevOps Services ou o endereço do Azure DevOps Server	Sim
Host de migração → instância de destino	HTTPS: envio dos arquivos de exportação e chamadas de API	Sim
Host de migração → gitlab.com	HTTPS: runner e projeto do engajamento	Sim
Host de migração → registry.gitlab.com	HTTPS: imagens do GitLab Congregate e da plataforma Pointer. Para pull de registry.gitlab.com, a documentação da GitLab.com lista também <code>*.storage.googleapis.com</code> e <code>*.cdn.registry.gitlab-static.net</code>	Sim
Host de migração → Docker Hub	HTTPS: imagens do MongoDB e do Redis	Sim
Estação do engenheiro → host de migração	SSH, para o túnel de acompanhamento	Não

- A instância de destino não precisa alcançar o Azure DevOps: o host de migração lê o Azure DevOps e o destino importa por arquivo.
- O host de migração só faz conexões de saída. Espelhos internos equivalentes podem substituir gitlab.com, registry.gitlab.com e Docker Hub.
- Operação sem acesso à internet (air-gapped) está no roadmap.

7. Usuários e contribuições

- Não há usuários placeholder: autoria e memberships são associadas pelo e-mail no momento da importação. Usuário inexistente no destino nesse momento fica com a autoria na conta da importação, sem reatribuição depois.
- Opção 1: o GitLab Congregate cria no destino, antes da primeira onda, os usuários dos times dos projetos do Azure DevOps (senha aleatória ou redefinição pelo usuário; inativos só quando pedido), e o pipeline confere cada um pelo e-mail.
- Opção 2: usuários provisionados no destino por LDAP ou SAML, com o mesmo e-mail do Azure DevOps, antes da primeira onda.
- Depois da importação, o GitLab Congregate remove os membros diretos dos projetos importados; para as memberships, a matriz do GitLab Congregate orienta SAML (JIT), SCIM ou SAML Group Links no destino.
- A conta dona do token do destino é a conta da importação.

8. Governança da migração

- Aprovador de cada onda e do rollback, do lado do cliente.
- Janela de manutenção de cada onda e congelamento de escrita no Azure DevOps durante a janela.
- Política de usuários (criação pelo GitLab Congregate ou provisionamento por LDAP ou SAML) concluída antes da primeira onda.
- Comunicação aos usuários sobre o fim do uso dos repositórios no Azure DevOps (o pipeline não arquiva a origem Azure DevOps).
- Responsável por cada item de tratamento manual da matriz, inclusive a conversão dos pipelines YAML e a recriação das variáveis secretas.

9. Critérios de aceite por onda

- Antes da primeira onda: todos os usuários com situação "existe no destino" no relatório de usuários, quando a criação pelo GitLab Congregate está no plano.
- Relatório da onda sem erro: todo repositório existe no destino e nenhuma importação terminou com falha.
- Alertas (relações com falha, diferença entre objetos lidos e importados) analisados e aceitos ou corrigidos, com registro no registro de decisões.
- Validação funcional pelo cliente nos projetos críticos da onda: clone, merge requests, issues e autoria.

10. Entrega das credenciais

Cada credencial é cadastrada como variável de CI/CD do projeto do engajamento, com as opções *Protected* e *Masked and hidden* e escopo restrito ao ambiente de migração, que é protegido e exige aprovação para a migração e o rollback de cada onda. O nome do usuário do Azure DevOps também fica em variável do ambiente, fora do repositório do engajamento. O pipeline lista, sem valores, as variáveis exigidas pelo plano, e o preflight reprova se faltar alguma. A configuração do GitLab Congregate com os tokens só existe durante cada job no host de migração.

Tokens nunca são colados em chat, issue ou arquivo; se isso ocorrer, o token é revogado e outro é emitido. No encerramento, os tokens usados na origem e no destino são revogados e as contas de administrador criadas para o engajamento são desativadas.

11. Credenciais que o cliente entrega à Pointer

Nenhum valor secreto é enviado por e-mail, chat, issue ou arquivo. Cada credencial é cadastrada como variável de CI/CD protegida no projeto do engajamento, com escopo do ambiente, e revogada no encerramento.

Credencial	Finalidade	Permissões	Formato e validade	Obrigatório
Token de acesso pessoal da instância de destino	Usado pelo GitLab Congregate (criação de usuários, importação por arquivo e pós-importação), pela conferência de usuários, pela verificação das importações e pelo rollback.	Escopo <code>api</code> , e também <code>admin_mode</code> quando o Admin Mode estiver habilitado ou quando for usada uma conta de serviço (runbook do GitLab Congregate). Usuário administrador da instância: a criação de usuários sem administrador reprova no preflight.	Personal access token da instância GitLab de destino, cadastrado como variável de CI/CD <code>PS_MIG_DESTINO_TOKEN</code> (Protected, Masked and hidden, escopo do ambiente de migração) Expiração na data estimada da última onda, conforme a documentação do GitLab Congregate; o preflight alerta com menos de 7 dias. Revogado no encerramento.	Sim
Personal access token (PAT) do Azure DevOps	Usado pelo GitLab Congregate para listar organização, projetos, repositórios, times e usuários, clonar os repositórios, ler pull requests, work items e anexos e montar a exportação.	Pipeline Pointer: leitura em Code, Project and Team, Work Items e Graph. Documentação do GitLab Congregate: escopo Read (ou Full access) e usuário do token administrador da organização (por exemplo, com o papel Azure DevOps Administrator do Microsoft Entra). O papel exigido na organização é definido na reunião de prontidão.	PAT do Azure DevOps, cadastrado como variável de CI/CD <code>PS_MIG_ORIGEM_TOKEN</code> (Protected, Masked and hidden, escopo do ambiente de migração) Expiração na data estimada da última onda. Revogado no encerramento.	Sim
Usuário do Azure DevOps dono do PAT	Identificação do usuário do PAT, exigida pelo plano de migração com origem Azure DevOps; sem ela, o preflight reprova.	Mesmo usuário do PAT.	Texto, cadastrado como variável do ambiente de migração <code>PS_MIG_ORIGEM_USUARIO</code> (fora do repositório, por ser dado pessoal) Até o encerramento	Sim

12. Informações a enviar

Informações sem sigilo, devolvidas pelo cliente antes da reunião de prontidão. Os campos marcados como obrigatórios são conferidos pela validação automática do pipeline.

Origem

Informação	Exemplo	Obrigatório	Observação
Tipo de origem	Azure DevOps Services	Sim	Azure DevOps Services ou Azure DevOps Server (sob consulta).
URL da organização	<code>https://dev.azure.com/orgao-exemplo</code>	Sim	No Azure DevOps Server, o endereço da coleção.
Projetos do Azure DevOps no escopo	sistemas-internos, portal	Sim	
Tipos de work item em uso	Epic, Feature, User Story, Task, Bug	Sim	Epic e Feature não migram no fluxo atual.

Informação	Exemplo	Obrigatório	Observação
Repositórios TFVC existentes	Não	Sim	TFVC fica fora do escopo.

Destino

Informação	Exemplo	Obrigatório	Observação
URL da instância de destino	https://gitlab.exemplo.gov.br	Sim	
Versão e nível de assinatura do destino	19.4.1, Premium	Sim	
Grupo pai no destino	migrados-ado	Sim	Grupo privado ou interno existente.

Host de migração e rede

Informação	Exemplo	Obrigatório	Observação
Nome e sistema operacional do host de migração	host-migracao.exemplo.gov.br, Ubuntu 24.04	Sim	
Proxy de saída	https://http://proxy.exemplo.gov.br:3128; exceções: .exemplo.gov.br	Condicional: proxy corporativo	HTTPS, HTTP e exceções.
Caminho da CA corporativa no host	/etc/pki/ca-trust/source/anchors/ac-exemplo.pem	Condicional: CA interna	Caminho absoluto de arquivo existente no host.
Acesso SSH para acompanhamento	usuario@host-migracao.exemplo.gov.br	Não	Túnel para a interface do GitLab Congregate.

Usuários

Informação	Exemplo	Obrigatório	Observação
Política de usuários	Criação pelo GitLab Congregate antes das ondas	Sim	Criação pelo GitLab Congregate ou provisionamento por LDAP ou SAML com o mesmo e-mail.
Senha dos usuários criados	Senha aleatória	Condicional: criação de usuários	Senha aleatória (padrão) ou redefinição de senha pelo usuário.
Incluir usuários inativos	Não	Condicional: criação de usuários	
Confirmação de e-mails iguais no Azure DevOps e no destino	Sim	Sim	

Ondas

Informação	Exemplo	Obrigatório	Observação
Projetos do Azure DevOps de cada onda	piloto: portal	Sim	O mesmo projeto não se repete entre ondas. Também por planilha (colunas onda e projeto).
Nome, data e descrição de cada onda	piloto; 10/11/2026; projeto de baixo risco	Sim	A primeira onda é a piloto.
Janela de manutenção de cada onda	10/11/2026, 19h às 23h	Sim	
Aprovador de cada onda e do rollback	Nome e cargo	Sim	
Janela de rollback	24 horas	Não	De 1 a 720 horas; padrão 24.
Confirmação do congelamento de escrita no Azure DevOps durante a janela	Sim	Sim	

Pós-migração

Informação	Exemplo	Obrigatório	Observação
Responsáveis pelos itens de tratamento manual	Pipelines YAML e variáveis secretas: equipe de plataforma	Sim	Conforme a matriz.

Contatos

Informação	Exemplo	Obrigatório	Observação
Patrocinador	Nome, e-mail, telefone	Sim	
Ponto focal técnico	Nome, e-mail, telefone	Sim	
Administrador da organização no Azure DevOps	Nome, e-mail	Sim	

Informação	Exemplo	Obrigatório	Observação
Administrador da instância de destino	Nome, e-mail	Sim	
Infraestrutura, rede e segurança	Nome, e-mail	Sim	Host, firewall, proxy e certificados.

13. Matriz de itens migrados

- Nativo** Migrado pelo método oficial (importação por arquivo da instância de destino, a partir da exportação montada pelo GitLab Congregate), disparado pelo pipeline
- Pipeline Pointer** Tratado pelo pipeline Pointer após a importação (pós-importação do GitLab Congregate)
- Ajuste manual** Exige ajuste manual após a migração (com responsável definido no plano)
- Não migra** Não é migrado; recriação fora do escopo, salvo acordo

O tratamento de cada item segue a documentação oficial do método de migração e é conferido na onda piloto do engajamento.

Repositório

Item	Tratamento	Observação
Repositório Git (commits, branches, tags)	Nativo	Histórico e autoria (nome e e-mail) dos commits.
Políticas de branch (mínimo de revisores, resolução de comentários, tipos de merge, build validation)	Ajuste manual	O GitLab Congregate só migra políticas de branch com uma opção que o pipeline não habilita.
Repositórios TFVC	Ajuste manual	A documentação GitLab informa que não há ferramenta da GitLab para migrar de TFVC para Git. Conversão para Git fora do pipeline e fora do escopo desta oferta.

Pull requests → merge requests

Item	Tratamento	Observação
Pull requests (autor, responsáveis, revisores, quem fez o merge, quem fechou, comentários)	Nativo	Todos os pull requests do projeto (o pipeline não aplica filtro de data).
Anexos de pull requests e de work items	Pipeline Pointer	Etapa fixa do pós-importação do GitLab Congregate: baixa do Azure DevOps, envia ao destino e reescreve os links.
Vínculo entre work item e pull request	Nativo	Referências cruzadas nas descrições (seções de issues e merge requests relacionados), geradas na exportação; a matriz do GitLab Congregate classifica como parcial.

Work items → issues

Item	Tratamento	Observação
Work items de backlog (Bug, Task, User Story, Issue, Product Backlog Item, Requirement, Impediment)	Nativo	Viram issues do projeto com título, descrição convertida de HTML para Markdown, estado (New e Active → aberta; Resolved e Closed → fechada), autor, responsável, comentários, e tipo e tags como labels.
Work item Epic	Não migra	No fluxo do pipeline (ondas de projeto, com o grupo do projeto criado antes), o Epic não vira issue nem épico. A documentação do GitLab Congregate descreve Epic e Feature como épicos de grupo, tratados na importação de grupo, que não roda quando o grupo já existe.
Work item Feature	Não migra	Nível de portfólio, com o mesmo tratamento do Epic no GitLab Congregate.
Campos personalizados e tipos de work item personalizados	Não migra	O pipeline não habilita as opções do GitLab Congregate para campos e tipos personalizados.
Azure Test Plans	Não migra	Sem equivalente na plataforma GitLab, segundo a matriz do GitLab Congregate.

CI/CD

Item	Tratamento	Observação
Pipelines YAML (Azure Pipelines)	Ajuste manual	Conversão para GitLab CI/CD fora da migração; o conversor de pipelines do GitLab Congregate não é usado pelo pipeline Pointer.
Variáveis secretas e variable groups ligados ao Azure Key Vault	Ajuste manual	Não migrados; recriados manualmente no destino (documentação do GitLab Congregate).

Pacotes e registries

Item	Tratamento	Observação
Imagens do Azure Container Registry	Ajuste manual	Fora do fluxo principal do GitLab Congregate; a cópia de imagens do pipeline atende só origem GitLab.

Usuários e permissões

Item	Tratamento	Observação
Permissões e memberships do Azure DevOps	Ajuste manual	Não suportado pelo GitLab Congregate, que orienta SAML (JIT), SCIM ou SAML Group Links; o GitLab Congregate remove os membros diretos dos projetos após a importação.

Notas da matriz

- **A confirmar na onda piloto:** variable groups com valores não secretos (no GitLab Congregate viram variáveis de CI/CD de grupo na importação de grupo, que o fluxo atual do pipeline não executa); wikis de projeto e de código (tratadas na importação de grupo do GitLab Congregate; o destino da wiki de código diverge entre os documentos do GitLab Congregate); feeds do Azure Artifacts (a matriz do GitLab Congregate classifica como parcial e em andamento; fora do pipeline).
- Limitação do fluxo do pipeline: o work item Epic não vira issue nem épico.
- Azure DevOps Server: toda a matriz é conferida na onda piloto.

14. Checklist de prontidão

Itens conferidos antes da execução. "Preflight automático" indica verificação feita pelo pipeline contra o ambiente real; os demais são conferidos na reunião de prontidão ou pelo cliente.

☐	Item	Verificado por
☐	Docker Engine e Docker Compose v2 presentes no host; CA informada existente no host (verificado na preparação do host).	Preflight automático
☐	Variáveis de CI/CD exigidas pelo plano cadastradas, inclusive o usuário do Azure DevOps.	Preflight automático
☐	Instância de destino acessível e token do destino válido, com papel de administrador.	Preflight automático
☐	Token do destino com validade maior que 7 dias.	Preflight automático
☐	Fonte de importação <i>GitLab export</i> habilitada no destino.	Preflight automático
☐	Grupo pai no destino existente e privado ou interno.	Preflight automático
☐	Configuração do GitLab Congregate validada.	Preflight automático
☐	PAT do Azure DevOps com os escopos de leitura exigidos e papel na organização definido.	Reunião de prontidão
☐	Runner online, com a tag do engajamento, protegida, travado no projeto e com tempo máximo de job de 8 horas ou mais.	Reunião de prontidão
☐	Plano de ondas com onda piloto, janelas, aprovadores e janela de rollback definidos.	Reunião de prontidão
☐	Política de usuários definida e e-mails iguais no Azure DevOps e no destino.	Reunião de prontidão
☐	Azure DevOps Server: onda piloto definida.	Reunião de prontidão
☐	Usuários criados ou provisionados no destino antes da primeira onda.	Cliente
☐	Host de migração provisionado conforme a especificação, com saída HTTPS para Azure DevOps, destino, gitlab.com, registry.gitlab.com e Docker Hub.	Cliente
☐	PAT do Azure DevOps e token do destino emitidos com escopos, papéis e validade exigidos e cadastrados como variáveis protegidas.	Cliente
☐	Congelamento de escrita comunicado aos usuários do Azure DevOps para a janela de cada onda.	Cliente