

POINTER DEVSECOPS · PROFESSIONAL SERVICES

Catálogo de Serviços de Professional Services para a plataforma GitLab™

Implantação, migração e avaliação da plataforma GitLab executadas por pipelines próprios da Pointer, em data center do cliente ou em nuvem

CÓDIGO

PS-00

TIPO

Catálogo

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.



GitLab
Select
Partner

PS-00 · POINTER DEVSECOPS · PROFESSIONAL SERVICES

Catálogo de Serviços

Implantação, migração e avaliação da plataforma GitLab executadas por pipelines próprios da Pointer, em data center do cliente ou em nuvem

Conteúdo

Apresentação

A Pointer

Portfólio de serviços

Onde implantamos: data center próprio ou nuvem

Metodologia

Como entregamos: a plataforma Pointer de Professional Services

Arquiteturas suportadas

Origens e destino de migração

Avaliações: maturidade, Customer Success Plan e levantamento da plataforma

Capacitação

Sob consulta e roadmap

Documentos complementares

Próximos passos

1. Apresentação

A Pointer DevSecOps Professional Services executa a implantação, a migração, a avaliação, a capacitação e o onboarding da plataforma GitLab, e publica o painel executivo de métricas DORA, para órgãos públicos e empresas privadas. Cada serviço é entregue por um **pipeline próprio de Professional Services**, construído sobre as ferramentas oficiais da GitLab (GitLab Environment Toolkit, Helm chart, Linux package, Direct Transfer e Congregate) e versionado em um catálogo interno de componentes de CI/CD.

Este catálogo descreve os serviços, as variantes de arquitetura, as origens e o destino de migração suportados, a metodologia de execução e os documentos complementares de cada oferta: *data sheet*, pré-requisitos e estudo de custos de infraestrutura.

2. A Pointer

A Pointer (O3S Consultoria e Tecnologia da Informação Ltda.), empresa do Extreme Group, atua em DevSecOps, estratégia multinuvem e segurança, da consultoria à capacitação. A vertical DevSecOps é parceira GitLab Select.

RAZÃO SOCIAL	O3S Consultoria e Tecnologia da Informação Ltda.
PARCERIA	Parceira GitLab Select
CERTIFICAÇÕES GITLAB DA EQUIPE	Certificações oficiais GitLab de serviços profissionais (implantação e migração)
CONTATO	contato@pointertech.digital · +55 (61) 99874-4648 · pointertech.digital

3. Portfólio de serviços

As ofertas se organizam em cinco famílias: implantação, migração, avaliação, capacitação e operação assistida. Cada oferta tem uma *data sheet* com escopo, arquitetura, entregáveis e premissas, e um documento de pré-requisitos, enviado depois da escolha do serviço, com o que o cliente provisiona e o que informa à Pointer.

Implantação

Código	Oferta	Variantes e estado	Pré-requisitos
PS-IMP-01	Implantação da plataforma GitLab em servidores Linux GitLab self-managed com Linux package, instalado pelo GitLab Environment Toolkit em máquinas virtuais existentes do cliente ou criadas pelo pipeline na conta AWS ou Azure do cliente, nas arquiteturas de referência de 1.000 a 50.000 usuários	● Disponível × 4 ○ Sob consulta × 3	PR-IMP-01
PS-IMP-02	Implantação da plataforma GitLab em Kubernetes GitLab self-managed pelo GitLab Helm chart ou pelo GitLab Operator, aplicados pelo GitLab Environment Toolkit em cluster Kubernetes existente do cliente ou, no Cloud Native Hybrid na AWS, em EKS criado pelo pipeline, nas arquiteturas Cloud Native e Cloud Native Hybrid	● Disponível × 7 ○ Sob consulta × 3	PR-IMP-02

Migração

Código	Oferta	Variantes e estado	Pré-requisitos
PS-MIG-01	Migração GitLab para GitLab self-managed Grupos e projetos de GitLab.com ou de GitLab self-managed migrados em ondas para a instância GitLab self-managed do cliente, por Direct Transfer orquestrado pelo GitLab Congregate	● Disponível × 3	PR-MIG-01

Código	Oferta	Variantes e estado	Pré-requisitos
PS-MIG-02	Migração GitHub para GitLab self-managed Repositórios, issues, pull requests e releases do GitHub importados em ondas para a instância GitLab self-managed do cliente, pelo importador GitHub da instância de destino disparado pelo GitLab Congregate	● Disponível x 1 ○ Sob consulta x 1	PR-MIG-02
PS-MIG-03	Migração Azure DevOps para GitLab self-managed Repositórios Git, pull requests e work items de backlog do Azure DevOps Services migrados em ondas para a instância GitLab self-managed do cliente, por exportação montada pelo GitLab Congregate e importação por arquivo	● Disponível x 1 ○ Sob consulta x 1	PR-MIG-03
PS-MIG-04	Migração Bitbucket Cloud para GitLab self-managed Repositórios e pull requests de um workspace do Bitbucket Cloud importados em ondas para a instância GitLab self-managed do cliente, pelo importador Bitbucket Cloud da instância de destino disparado pelo GitLab Congregate	● Disponível x 1	PR-MIG-04

Avaliação

Código	Oferta	Variantes e estado	Pré-requisitos
PS-AVA-01	Avaliação de maturidade DevSecOps e Customer Success Plan Teste de Maturidade por persona e capacidade da plataforma GitLab 19.x, Customer Success Plan e pesquisa de satisfação, com resultados em site exclusivo do cliente	● Disponível x 3 ○ Sob consulta x 1	PR-AVA-01
PS-AVA-02	Levantamento da plataforma Inventário, métricas, ondas e estimativa de esforço de migração coletados por API, somente leitura, na plataforma atual	● Disponível x 6 ○ Sob consulta x 3	PR-AVA-02

Código	Oferta	Variantes e estado	Pré-requisitos
PS-D0R-01	Painel executivo de métricas DORA Frequência de deploy, lead time de mudanças, tempo de restauração e taxa de falha da instância GitLab do cliente, por escopo, com evolução histórica, para diretoria e alta gestão sem usuário no GitLab	● Disponível × 2 ○ Sob consulta × 1	PR-D0R-01

Capacitação

Código	Oferta	Variantes e estado	Pré-requisitos
PS-CAP-01	Capacitação na plataforma GitLab Sete trilhas por perfil com 28 módulos alinhados à versão 19.4, exercícios práticos por aluno com correção automática, avaliação pré e pós e certificados verificáveis	● Disponível × 7 ○ Sob consulta × 6	PR-CAP-01

Operação assistida

Código	Oferta	Variantes e estado	Pré-requisitos
PS-ONB-01	Onboarding do cliente na plataforma GitLab Página de boas-vindas à plataforma GitLab contratada: portais oficiais da GitLab, abertura e acompanhamento de chamados, SLAs do Priority Support, escopo do suporte da GitLab e o apoio da Pointer em cada etapa	● Disponível × 2	PR-ONB-01
PS-OPS-01	Operação assistida Acompanhamento da plataforma GitLab implantada pelo pipeline Pointer, com ajustes por merge request e rotinas de dia 2 versionadas	● Disponível × 12 ○ Sob consulta × 2 —	

● Disponível Ofertado nas condições deste documento.

○ Sob consulta Escopo e condições definidos em proposta específica, com execução piloto.

4. Onde implantamos: data center próprio ou nuvem

A implantação é feita sobre **infraestrutura já existente do cliente** ou sobre **infraestrutura criada pelo pipeline na conta AWS ou Azure do cliente**. Em infraestrutura existente, o pipeline não usa API de provedor de nuvem: recebe máquinas virtuais Linux (inventário estático, acesso

por SSH) ou um cluster Kubernetes existente. A instalação executa a partir de um runner com acesso à rede do cliente.

- **Data center próprio (on-premises):** VMs em qualquer hipervisor (por exemplo VMware, Hyper-V, KVM) ou servidores físicos, desde que com sistema operacional suportado, SSH e sudo; clusters Kubernetes de distribuição conforme (por exemplo RKE2, OpenShift, K3s).
- **Nuvem pública, recursos existentes:** VMs ou clusters já criados na conta do cliente (por exemplo AWS, Azure ou Google Cloud), inclusive serviços gerenciados de Kubernetes, PostgreSQL, Redis e object storage onde a arquitetura permite.
- **Nuvem pública, infraestrutura criada pelo pipeline:** com os módulos Terraform oficiais do GitLab Environment Toolkit, sem modificação — na AWS, VMs das arquiteturas Linux e Cloud Native Hybrid com EKS, rede e buckets; no Azure, VMs das arquiteturas Linux. O pipeline usa a API do provedor com a identidade fornecida pelo cliente, guarda o estado do Terraform no projeto do engajamento e só destrói recursos por decisão registrada em merge request. Google Cloud (VMs e GKE) e AKS criados pelo pipeline: sob consulta.
- **Rede:** o runner do engajamento precisa de saída HTTPS para gitlab.com, registry.gitlab.com e packages.gitlab.com (diretamente ou por proxy). A operação sem acesso à internet (*air-gapped*) está no roadmap.

5. Metodologia

Todo engajamento segue a metodologia da Pointer em quatro fases, executadas com a governança descrita a seguir e com os entregáveis registrados no repositório do engajamento.

Fase 1 Assessment • Reunião de kick-off: escopo, papéis, canais, calendário e critérios de aceite • Discovery técnico com roteiro estruturado e, quando contratado, Levantamento da Plataforma por API • Definição da arquitetura (arquitetura de referência GitLab) ou do plano de ondas de migração • Envio do documento de pré-requisitos e acompanhamento do checklist de prontidão • Criação do projeto do engajamento a partir do template e validação automática da configuração	Fase 2 Implantação • Preflight contra o ambiente real do cliente (conectividade, recursos, versões, serviços externos) • Execução pelo pipeline em ambiente protegido, com aprovação por etapa • Verificação automática pós-implantação ou pós-onda e registro das evidências • Tratamento de falhas com diagnóstico automático, reexecução ou rollback, conforme o serviço	Fase 3 Otimização • Operação assistida no período acordado (acompanhamento de uso, correções e ajustes de configuração; chamados do cliente na ferramenta de chamados da Pointer, com prazo de primeira resposta por severidade) • Teste de backup e de restauração, e rotinas de dia 2 disponíveis no pipeline • Ajustes de runners, integrações e políticas conforme o escopo contratado	Fase 4 Transferência de conhecimento • Documento as-built gerado pelo pipeline e revisão com a equipe do cliente • Sessões de transferência de conhecimento sobre a arquitetura, o pipeline e as rotinas de dia 2 • Entrega do repositório do engajamento (configuração como código), com opção de espelhamento na instância GitLab do cliente • Aceite formal e relatório de encerramento
--	--	---	---

Fase	Objetivo	Saídas
Assessment	Confirmar requisitos, arquitetura e prontidão antes de qualquer alteração no ambiente.	Plano do engajamento · Configuração do engajamento validada pelo pipeline · Checklist de prontidão
Implantação	Executar a implantação ou a migração pelo pipeline Pointer, com verificação automática.	Relatórios de preflight e verificação · Relatório de cada onda de migração
Otimização	Estabilizar e ajustar o ambiente entregue.	Registro de ajustes · Evidência de backup
Transferência de conhecimento	Entregar a operação ao cliente com documentação e capacitação.	As-built · Termo de aceite · Relatório de encerramento

Papéis

Papel	Responsabilidade
Pré-venda e proposta (Pointer)	Reunião de discovery, escolha da oferta e da variante, proposta técnico-comercial e envio do documento de pré-requisitos após a escolha do serviço
Gerente do engajamento (Pointer)	Plano, cronograma, riscos, mudanças de escopo, status semanal e aceite
Coordenação do engajamento (Pointer)	Agenda de sessões e janelas, convocação dos rituais, registro de riscos, ações, pendências e decisões e controle das pendências do cliente
Arquitetura técnica (Pointer)	Arquitetura de referência ou plano de ondas de migração, revisão técnica das mudanças de configuração e decisões técnicas do engajamento
Engenheiro de Professional Services (Pointer)	Configuração do engajamento, execução dos pipelines e evidências
Patrocinador (cliente)	Decisões de escopo, prioridade e aceite
Ponto focal técnico (cliente)	Pré-requisitos, acessos, janelas de mudança e validação funcional
Infraestrutura, rede e segurança (cliente)	VMs ou cluster, DNS, certificados, regras de firewall e contas de serviço

Uma mesma pessoa pode acumular papéis da Pointer; a designação de cada papel é registrada no kick-off.

Rituais de governança

- Kick-off com escopo, papéis e critérios de aceite
- Status semanal com andamento, riscos, bloqueios e próximos passos
- Mudança de escopo por solicitação formal, avaliada em prazo e esforço antes da execução
- Aceite por entregável e encerramento com relatório final
- Retrospectiva interna da equipe Pointer e retrospectiva com o cliente na reunião de encerramento
- Pesquisa de satisfação do engajamento (CSAT) respondida pelo cliente no encerramento

Registros e aceite

- **Registro RAID** (riscos, ações, pendências e decisões) no projeto do engajamento: riscos, ações e pendências como issues com rótulos próprios e quadro de acompanhamento; decisões no registro de decisões do repositório. Revisado no status semanal.
- **Aceite por entregável:** o cliente registra o aceite ou as objeções de cada entregável no prazo definido na proposta (padrão: 5 dias úteis após a entrega). Sem manifestação nesse prazo, o entregável é considerado aceito (aceite passivo).

6. Como entregamos: a plataforma Pointer de Professional Services

Os serviços de implantação, migração e avaliação são executados por **componentes de CI/CD versionados**, mantidos pela Pointer em um catálogo interno e consumidos pelo projeto de cada engajamento.

Configuração como código

Cada engajamento tem um repositório próprio com a configuração do ambiente (arquivos YAML), revisada por merge request e validada automaticamente pelo pipeline.

Isolamento por cliente

Um projeto, um runner e um conjunto de variáveis por cliente. O pipeline de um cliente não acessa dados nem ambientes de outro.

Execução na rede do cliente

Os jobs que instalam e alteram a plataforma rodam em um runner com acesso à rede do cliente; a criação de infraestrutura na nuvem usa só a API do provedor, em runner hospedado ou em runner indicado pelo cliente. Credenciais ficam em variáveis protegidas com escopo de ambiente.

Versões fixas e reproduzíveis

Cada engajamento usa uma versão fixa dos componentes e das imagens. Uma nova versão só entra por merge request.

Evidência em cada etapa

Preflight, verificação, relatórios de onda e as-built são gerados pelo próprio pipeline e ficam registrados no projeto do engajamento.

Versões testadas

Cada versão dos componentes passa por testes automatizados e os fluxos ofertados são executados de ponta a ponta antes de entrar no catálogo.

Ferramentas

Ferramenta	Origem	O que é	Como a Pointer usa
GitLab Environment Toolkit (GET)	Ferramenta oficial da GitLab	Conjunto de playbooks Ansible e módulos Terraform mantido pela GitLab para implantar as arquiteturas de referência GitLab.	Imagem oficial do GET, sem modificações, executada pelo pipeline com inventário estático gerado a partir da configuração do engajamento; extensões por Custom Config e Custom Tasks. Quando o pipeline cria a infraestrutura, usa os módulos Terraform oficiais do GET para AWS e Azure.

Ferramenta	Origem	O que é	Como a Pointer usa
Linux package (Omnibus)	Pacote oficial da GitLab	Pacote de instalação da plataforma GitLab para servidores Linux.	Instalado pelo GET em cada VM, na versão definida no engajamento.
GitLab Helm chart	Chart oficial da GitLab	Chart Helm para instalar a plataforma GitLab em Kubernetes.	Aplicado pelo GET nas topologias Cloud Native Hybrid e Cloud Native, com os valores gerados pelo pipeline, diretamente ou pelo GitLab Operator.
GitLab Runner e agente GitLab para Kubernetes	Ferramentas oficiais da GitLab	Executor dos jobs de CI/CD e agente que conecta clusters Kubernetes à instância.	Frota de runners do cliente criada na instância pela API e instalada em VMs ou em Kubernetes; agentes registrados e instalados nos clusters do cliente, com o acesso de CI/CD definido na configuração.
Direct Transfer	Recurso nativo da plataforma GitLab	Migração de grupos e projetos entre instâncias GitLab pela API, executada pela instância de destino.	Método padrão das migrações GitLab → GitLab, disparado pelo Congregate a partir do pipeline.
Importadores da plataforma GitLab	Recurso nativo da plataforma GitLab	Importadores de projetos do GitHub e do Bitbucket, executados pela instância de destino.	Disparados pelo Congregate nas migrações de GitHub e Bitbucket.
GitLab Congregate	Ferramenta da GitLab Professional Services	Ferramenta de automação de migrações em escala mantida pela equipe de Professional Services da GitLab.	Versão fixada por digest, executada em um host de migração dedicado; o pipeline Pointer gera a configuração, dispara cada onda e trata as etapas posteriores à importação.
psctl	Ferramenta Pointer	Linha de comando da plataforma Pointer: valida a configuração do engajamento, gera o ambiente do GET, avalia o preflight, produz relatórios, as-built e os sites de avaliação.	Empacotada nas imagens dos pipelines.
GitLab CI/CD Catalog e GitLab Pages	Recursos nativos da plataforma GitLab	Catálogo de componentes de CI/CD e publicação de sites estáticos.	Distribuição versionada dos componentes Pointer; publicação dos sites de avaliação e deste catálogo.

7. Arquiteturas suportadas

As implantações seguem as **arquiteturas de referência GitLab**, dimensionadas por carga (requisições por segundo) e número de usuários, conforme a documentação oficial. O estado de cada variante (Disponível ou Sob consulta) é informado na tabela.

Implantação da plataforma GitLab em servidores Linux (PS-IMP-01)

• **Disponível** Ofertado nas condições deste documento.

◦ **Sob consulta** Escopo e condições definidos em proposta específica, com execução piloto.

Arquitetura	Carga de referência	Componentes	Estado
1k · nó único	até 20 RPS / 1.000 usuários	1 VM com todos os componentes (8 vCPU, 16 GB); object storage opcional	• Disponível
2k · multi-nó	até 40 RPS / 2.000 usuários	8 VMs: balanceador externo (HAProxy do GET ou do cliente), PostgreSQL, Redis, Gitaly, Sidekiq, 2 GitLab Rails e monitoramento; object storage obrigatório; sem alta disponibilidade	• Disponível
3k · alta disponibilidade	até 60 RPS / 3.000 usuários	27 VMs com Gitaly Cluster (23 com Gitaly Sharded): balanceadores externo e interno (HAProxy), 3 Consul, 3 PostgreSQL com Patroni, 3 PgBouncer, 3 Redis com Sentinel, 3 Gitaly, 3 Praefect e 1 PostgreSQL do Praefect, 2 Sidekiq, 3 GitLab Rails e monitoramento; object storage obrigatório	• Disponível
5k · alta disponibilidade	até 100 RPS / 5.000 usuários	27 VMs com Gitaly Cluster (23 com Gitaly Sharded): mesmos papéis do 3k, com nós maiores de PostgreSQL, Gitaly e GitLab Rails	• Disponível
10k · alta disponibilidade	até 200 RPS / 10.000 usuários	32 VMs com Gitaly Cluster (28 com Gitaly Sharded): papéis do 3k com Redis separado em cache e persistente (3 + 3 VMs) e 4 Sidekiq	◦ Sob consulta
25k · alta disponibilidade	até 500 RPS / 25.000 usuários	34 VMs com Gitaly Cluster (30 com Gitaly Sharded): papéis do 10k com 5 GitLab Rails e nós maiores	◦ Sob consulta
50k · alta disponibilidade	até 1.000 RPS / 50.000 usuários	41 VMs com Gitaly Cluster (37 com Gitaly Sharded): papéis do 10k com 12 GitLab Rails e nós maiores	◦ Sob consulta

- Carga de referência conforme a documentação GitLab: RPS é a métrica principal de dimensionamento; para cada 1.000 usuários, a GitLab testa 20 RPS de API, 2 RPS web, 2 RPS de Git pull e 0,4 RPS de Git push. A coluna mostra o RPS de API e a equivalência em usuários.
- Quantidade de VMs e especificação por papel conforme as arquiteturas de referência GitLab (documentação oficial consultada em 29/09/2026). O detalhamento por papel está no documento de pré-requisitos PR-IMP-01.
- As mesmas arquiteturas valem para VMs existentes e para VMs criadas pelo pipeline na AWS ou no Azure. Na infraestrutura criada pelo pipeline, a quantidade de VMs é a da arquitetura, e o tipo de cada papel é o menor tipo

da lista do pipeline que atende vCPU e memória da arquitetura (AWS: famílias c7i, m7i e r7i; Azure: Fsv2, Dasv5 e Easv5) ou o tipo definido na configuração.

- Gitaly Sharded usa as mesmas especificações de Gitaly e dispensa os 3 nós Praefect e o PostgreSQL do Praefect. Segundo a documentação GitLab, o Gitaly Cluster (Praefect) oferece tolerância a falhas com complexidade adicional de instalação e gestão.
- Os balanceadores externo e interno são uma VM HAProxy cada, instalada pelo GET; o GET não implementa balanceador em alta disponibilidade em servidores próprios. O balanceador do cliente pode substituir o HAProxy externo, com TLS em passthrough até os nós GitLab Rails (2 ou mais nós); TLS terminado no balanceador do cliente e balanceador interno do cliente são atendidos sob consulta.
- Arquiteturas em estado *Sob consulta*, escopo e condições definidos em proposta específica, com execução piloto.

Implantação da plataforma GitLab em Kubernetes (PS-IMP-02)

• **Disponível** Ofertado nas condições deste documento.

◦ **Sob consulta** Escopo e condições definidos em proposta específica, com execução piloto.

Arquitetura	Carga de referência	Componentes	Estado
Cloud Native S	até 100 RPS; carga leve, não adequada a monorepos em uso ativo	No cluster: Webservice 6 a 9 pods, Sidekiq 8 a 12 pods, Gitaly 3 pods (um nó dedicado por pod), serviços de suporte 12 vCPU / 48 GB. Externos: PostgreSQL 8 vCPU / 32 GB, Redis cache e Redis persistente 2 vCPU / 8 GB cada, object storage	• Disponível
Cloud Native M	até 200 RPS; carga moderada, monorepos de uso leve	No cluster: Webservice 14 a 21 pods, Sidekiq 16 a 24 pods, Gitaly 3 pods (15 vCPU / 62 GB cada), suporte 12 vCPU / 48 GB. Externos: PostgreSQL 16 vCPU / 64 GB, Redis cache e persistente 2 vCPU / 8 GB cada, object storage	• Disponível
Cloud Native L	até 500 RPS; carga alta, monorepos de uso moderado	No cluster: Webservice 28 a 42 pods, Sidekiq 32 a 48 pods, Gitaly 3 pods (31 vCPU / 126 GB cada), suporte 12 vCPU / 48 GB. Externos: PostgreSQL 32 vCPU / 128 GB, Redis cache e persistente 2 vCPU / 16 GB cada, object storage	• Disponível
Cloud Native XL	até 1.000 RPS; carga intensa, monorepos de uso intenso	No cluster: Webservice 56 a 84 pods, Sidekiq 64 a 96 pods, Gitaly 3 pods (63 vCPU / 254 GB cada), suporte 24 vCPU / 96 GB. Externos: PostgreSQL 64 vCPU / 256 GB, Redis cache e persistente 2 vCPU / 16 GB cada, object storage	• Disponível
Cloud Native Hybrid 2k	até 40 RPS / 2.000 usuários	No cluster: Webservice, Sidekiq e suporte (12 + 3,6 + 4 vCPU solicitados). Em VMs: 1 Gitaly; PostgreSQL e Redis em 1 VM cada ou em serviços externos	• Disponível
Cloud Native Hybrid 3k	até 60 RPS / 3.000 usuários	No cluster: Webservice, Sidekiq e suporte (16 + 7,2 + 4 vCPU). Em VMs: 3 Consul, 3 PostgreSQL, 3 PgBouncer, balanceador interno, 3 Redis com Sentinel, 3 Gitaly, 3 Praefect e 1 PostgreSQL do Praefect (20 VMs com Gitaly Cluster)	• Disponível
Cloud Native Hybrid 5k	até 100 RPS / 5.000 usuários	No cluster: 36 + 7,2 + 4 vCPU. Em VMs: mesmos papéis do Hybrid 3k, com PostgreSQL e Gitaly maiores (20 VMs com Gitaly Cluster)	• Disponível

Arquitetura	Carga de referência	Componentes	Estado
Cloud Native Hybrid 10k	até 200 RPS / 10.000 usuários	No cluster: 80 + 12,6 + 8 vCPU. Em VMs: papéis do Hybrid 3k com Redis separado em cache e persistente (23 VMs com Gitaly Cluster)	Sob consulta
Cloud Native Hybrid 25k	até 500 RPS / 25.000 usuários	No cluster: 140 + 12,6 + 8 vCPU. Em VMs: papéis do Hybrid 10k com nós maiores (23 VMs com Gitaly Cluster)	Sob consulta
Cloud Native Hybrid 50k	até 1.000 RPS / 50.000 usuários	No cluster: 308 + 12,6 + 8 vCPU. Em VMs: papéis do Hybrid 10k com nós maiores (23 VMs com Gitaly Cluster)	Sob consulta

- Carga de referência conforme a documentação GitLab. Cloud Native é dimensionada só por RPS; a equivalência em usuários vale para Linux package e Cloud Native Hybrid (para cada 1.000 usuários, a GitLab testa 20 RPS de API, 2 RPS web, 2 RPS de Git pull e 0,4 RPS de Git push).
- Cloud Native: segundo a documentação GitLab, o Gitaly no Kubernetes funciona somente no modo sharded, e cada pod de Gitaly é ponto único de falha para os repositórios que atende; o Gitaly Cluster (Praefect) no Kubernetes está em beta e não faz parte da arquitetura de referência.
- PostgreSQL e Redis dentro do Kubernetes não são suportados pelas arquiteturas de referência GitLab; o pipeline exige PostgreSQL e Redis externos na Cloud Native e aceita VMs ou serviços externos no Hybrid.
- Os totais de vCPU e memória dos grupos de nós cobrem só os componentes GitLab; os processos de sistema do Kubernetes exigem recursos adicionais (documentação GitLab). Detalhamento por arquitetura no documento de pré-requisitos PR-IMP-02.
- A arquitetura de referência 1k não tem variante Cloud Native Hybrid.
- Cloud Native Hybrid 2k: PostgreSQL, Redis e Gitaly em nó único, sem alta disponibilidade. A atualização de versão sem parada exige backends em alta disponibilidade (Patroni e Consul, 3k ou mais) ou PostgreSQL externo; no Hybrid 2k com PostgreSQL em VM e na Cloud Native, a atualização usa janela de manutenção.
- As arquiteturas Cloud Native Hybrid valem para cluster e VMs existentes e para EKS e VMs criados pelo pipeline na AWS; no EKS criado pelo pipeline, a quantidade de nós de cada pool é calculada pelos totais da arquitetura ou definida na configuração.
- Cloud Native S, M, L e XL usam os mesmos componentes e papéis, com outra quantidade e tamanho de nós e de réplicas.
- Arquiteturas em estado *Sob consulta* (Cloud Native Hybrid 10k a 50k): escopo e condições definidos em proposta específica, com execução piloto.

8. Origens e destino de migração

O destino das migrações é o **GitLab self-managed** do cliente, inclusive um GitLab implantado pela própria Pointer. Cada origem usa o método oficial correspondente, disparado pelo pipeline Pointer, que também executa as etapas posteriores à importação, a verificação por contagens e o rollback da onda.

Oferta	Origem	Destino	Método	Estado
PS-MIG-01	GitLab.com	GitLab self-managed	Direct Transfer orquestrado pelo GitLab Congregate	• Disponível
PS-MIG-01	GitLab self-managed	GitLab self-managed	Direct Transfer orquestrado pelo GitLab Congregate	• Disponível

Oferta	Origem	Destino	Método	Estado
PS-MIG-01	GitLab self-managed	GitLab self-managed	Exportação e importação por arquivo orquestradas pelo GitLab Congregate (alternativa ao Direct Transfer)	• Disponível
PS-MIG-02	GitHub.com	GitLab self-managed	Importador GitHub da instância de destino, disparado pelo GitLab Congregate	• Disponível
PS-MIG-02	GitHub Enterprise Server	GitLab self-managed	Importador GitHub da instância de destino, pela API, disparado pelo GitLab Congregate	○ Sob consulta
PS-MIG-03	Azure DevOps Services	GitLab self-managed	Exportação montada pelo GitLab Congregate e importação por arquivo na instância de destino	• Disponível
PS-MIG-03	Azure DevOps Server	GitLab self-managed	Exportação montada pelo GitLab Congregate e importação por arquivo na instância de destino	○ Sob consulta
PS-MIG-04	Bitbucket Cloud (bitbucket.org)	GitLab self-managed	Importador Bitbucket Cloud da instância de destino, disparado pelo GitLab Congregate	• Disponível

A matriz de itens migrados de cada origem (nativo, tratado pelo pipeline, ajuste manual, não migra) está no documento de pré-requisitos da oferta.

9. Avaliações: maturidade, Customer Success Plan e levantamento da plataforma

A avaliação combina a percepção das equipes, coletada por formulários, com métricas medidas por API na plataforma atual. Os resultados ficam em um site exclusivo do cliente, com acesso protegido por senha, e em PDFs oficiais.

Avaliação de maturidade DevSecOps e Customer Success Plan (PS-AVA-01)

A avaliação mede a maturidade DevSecOps do cliente em 66 capacidades da plataforma GitLab, organizadas em 9 domínios e respondidas por 5 personas em formulários web, e monta o Customer Success Plan a partir das respostas dos stakeholders e do que é levantado com o cliente. Os resultados ficam em um site exclusivo do cliente no GitLab Pages, com acesso protegido por senha, e em PDFs oficiais. Uma rodada inicial e uma rodada final permitem comparar a maturidade no início e no fim do ciclo.

Levantamento da plataforma (PS-AVA-02)

O levantamento lê por API, somente leitura, a plataforma atual do cliente (GitLab, GitHub, Bitbucket, Azure DevOps ou AWS CodeCommit) e produz o inventário, 38 métricas de base, as ondas sugeridas e a estimativa de esforço da migração para uma instância GitLab de destino.

Coletas repetidas formam uma série histórica e o comparativo das métricas entre as fases inicial e final. Os resultados ficam no mesmo site cifrado da avaliação e em um PDF oficial.

Painel executivo de métricas DORA (PS-DOR-01)

Painel publicado no GitLab Pages com as quatro métricas DORA da instância GitLab do cliente — frequência de deploy, lead time de mudanças, tempo de restauração do serviço e taxa de falha de mudanças — nos escopos que o cliente escolhe: instância inteira, grupos específicos ou projetos específicos. Mostra a situação dos últimos 30 dias com a variação sobre os 30 dias anteriores, a série mensal, a evolução entre coletas e a origem de cada número. Diretoria e alta gestão acessam pelo navegador com uma senha, sem usuário no GitLab.

10. Capacitação

Trilhas de capacitação por perfil, em formato remoto ao vivo, presencial *in company* ou híbrido, com turmas fechadas sob demanda. Detalhes na *data sheet* de Capacitação.

Capacitação na plataforma GitLab (PS-CAP-01)

Catálogo de capacitação com 28 módulos combináveis em 7 trilhas por perfil (Fundamentos, Desenvolvimento, CI/CD e Plataforma, Segurança e Compliance, Administração e Operação Self-Managed, IA no ciclo de desenvolvimento e Gestão e Liderança), com conteúdo conferido na documentação da versão 19.4 da plataforma GitLab. Cada aluno recebe projetos de exercício próprios, corrigidos automaticamente pela API da plataforma GitLab, no grupo da Pointer no GitLab.com ou na instância GitLab do cliente. Presença, avaliação pré e pós e pesquisa de satisfação são registradas em formulários do site da turma, e o engajamento entrega relatório da turma, certificados de conclusão e declarações de participação em PDF com código verificável. As turmas não são treinamentos oficiais da GitLab Inc. e não incluem prova nem certificação oficial.

Onboarding do cliente na plataforma GitLab (PS-ONB-01)

Página de boas-vindas à plataforma GitLab contratada, reutilizável por cliente e publicada no GitLab Pages: a assinatura e a instância, para que serve o Customers Portal e o Support Portal, como abrir e acompanhar um chamado na GitLab, os SLAs oficiais de primeira resposta do Priority Support, o que o suporte da GitLab cobre e não cobre e como a Pointer apoia antes, durante e depois de um chamado. Os fatos oficiais vêm da documentação e do portal de suporte da GitLab, com fonte e data de consulta. Em engajamento com avaliação, a página integra o site da avaliação e traz uma seção personalizada, cifrada, com o Customer Success Plan, a maturidade e o levantamento da plataforma.

Operação assistida (PS-OPS-01)

Acompanhamento da plataforma GitLab entregue pelo pipeline de implantação da Pointer, no período acordado: ajustes de configuração por merge request no repositório do engajamento, execução das rotinas de dia 2 do pipeline (backup sob demanda, verificação avulsa, atualização de versão — sem parada na alta disponibilidade Linux e no Cloud Native Hybrid com backends em alta disponibilidade —, diagnóstico, reexecução das migrations do banco e as-built), manutenção da frota de runners do cliente, dos agentes GitLab para Kubernetes e da configuração do GitLab Geo, apoio a upgrades e análise de incidentes da plataforma. Os chamados do cliente são abertos e acompanhados na ferramenta de chamados da Pointer, com prazo de primeira resposta por severidade em horas úteis. Período, horário de atendimento, canais e prazos são definidos na proposta.

11. Sob consulta e roadmap

Atendidos sob consulta: variantes e origens atendidas com escopo e condições definidos em proposta específica, com execução piloto. **Roadmap:** itens que não fazem parte das ofertas atuais.

Atendidos sob consulta

Item	Família	Situação
Implantação Linux 10k a 50k (alta disponibilidade)	Implantação	Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Cloud Native Hybrid 10k a 50k	Implantação	Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Criação de infraestrutura no Google Cloud (VMs e GKE)	Implantação	O pipeline cria infraestrutura na AWS (VMs e EKS) e no Azure (VMs). No Google Cloud, VMs ou cluster GKE já existentes entram como infraestrutura existente; a criação pelo pipeline é atendida sob consulta, com escopo e condições definidos em proposta específica e execução piloto.
Cloud Native Hybrid com AKS criado pelo pipeline	Implantação	No Azure, o módulo Terraform do GitLab Environment Toolkit cria somente VMs; um cluster AKS existente entra como cluster existente. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Cloud Native (S a XL) com cluster criado pelo pipeline	Implantação	A criação de cluster pelo pipeline cobre o Cloud Native Hybrid na AWS (EKS); na Cloud Native, o cluster EKS existente entra como cluster existente. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
GitLab Operator em OpenShift	Implantação	A implantação pelo GitLab Operator está disponível em Kubernetes; em OpenShift, o pipeline aceita a configuração com alerta (sem Git por SSH via routes, limitação do Operator). Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
GitLab Geo em Kubernetes (Cloud Native Hybrid e Cloud Native)	Implantação	O GitLab Geo está disponível com dois sites Linux package. No Kubernetes, a validação da configuração recusa o Geo nesta versão. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Balanceador do cliente com TLS terminado no balanceador e balanceador interno do cliente	Implantação	O balanceador externo do cliente em TLS passthrough está disponível na implantação Linux. TLS terminado no balanceador do cliente (nós GitLab Rails em HTTP) e balanceador interno do cliente no lugar do HAProxy interno são suportados pela configuração do pipeline e atendidos sob consulta, com escopo e condições definidos em proposta específica e execução piloto.
Busca exata de código (Zoekt)	Implantação	O pipeline configura a busca avançada (Advanced Search) e os escopos da busca global; a busca exata de código com Zoekt não é configurada pelo pipeline. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.

Item	Família	Situação
Frota de runners com executor shell ou docker-autoscaler no Azure	Implantação	A frota de runners do cliente está disponível em VMs com executor Docker, em docker-autoscaler na AWS e em Kubernetes. Executor shell (jobs direto no host, sem isolamento; o pipeline emite alerta) e docker-autoscaler no Azure (VM Scale Set) são suportados pela configuração e atendidos sob consulta, com escopo e condições definidos em proposta específica e execução piloto.
Grafana gerenciado pela Pointer	Implantação	Grafana OSS em versão fixa com o Prometheus da instância como fonte de dados, HTTPS no host próprio do Grafana e login pela instância GitLab (aplicação OAuth, papéis por grupo): no nó de monitoramento da implantação Linux (2k ou mais) ou no kube-prometheus-stack do Kubernetes. Na arquitetura 1k não há nó de monitoramento. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
GitHub Enterprise Server como origem de migração	Migração	Mesmo importador GitHub da plataforma GitLab usado com o GitHub.com (Enterprise Server só pela API). Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Azure DevOps Server como origem de migração	Migração	O fluxo do pipeline atende o Azure DevOps Services. Para Azure DevOps Server/TFS, o runbook do GitLab Congregate prevê uma aplicação própria para listar os usuários. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Levantamento: origem Bitbucket Server/Data Center	Avaliação	Coletor implementado (instância, projeto e repositório; token HTTP de acesso). Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Levantamento: GitHub Enterprise Server e Azure DevOps Server	Avaliação	Mesmos coletores usados com GitHub.com e Azure DevOps Services. Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Capacitação: exercícios em ambiente preparado (GitLab Runner, agente para Kubernetes, DAST, compliance framework, administração self-managed e GitLab Duo Agent Platform)	Capacitação	Os módulos e as trilhas estão disponíveis; estes exercícios exigem ambiente preparado por aluno ou por turma (VM, cluster, aplicação de teste, instância self-managed, GitLab Credits). Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.

Roadmap

Item	Família	Situação
Instalação sem acesso à internet (air-gapped)	Implantação	O runner do engajamento precisa de saída HTTPS para gitlab.com, registry.gitlab.com e packages.gitlab.com, diretamente ou por proxy.
Bitbucket Server/Data Center como origem de migração	Migração	Configuração e preflight prontos no pipeline (importador Bitbucket Server da plataforma GitLab, disparado pelo GitLab Congregate). Sob consulta: escopo e condições definidos em proposta específica, com execução piloto.

Item	Família	Situação
AWS CodeCommit como origem de migração	Migração	O pipeline recusa a origem CodeCommit enquanto a versão fixada do GitLab Congregate for a 8.6.0. Alternativa documentada pela GitLab, fora do pipeline: importação do repositório por URL com as credenciais HTTPS Git, que traz o repositório Git sem issues nem merge requests.
GitLab.com e GitLab Dedicated como destino da migração	Migração	O pipeline trata destino GitLab.com (grupo pai obrigatório; sem token de administrador, não há criação de usuários). O catálogo oferta destino GitLab self-managed; destino GitLab.com ou GitLab Dedicated é atendido sob consulta: escopo e condições definidos em proposta específica, com execução piloto.
Autoria das contribuições na migração do Bitbucket Cloud	Migração	Na migração do Bitbucket Cloud (branches, pull requests como merge requests e labels), a autoria fica na conta da importação. O importador da plataforma GitLab não tem mapeamento posterior de usuários (placeholders) para Bitbucket Cloud: atribui a autoria ao usuário do destino com a identidade Bitbucket vinculada.
Capacitação: criação de usuários dos alunos na instância dos exercícios	Capacitação	Fora do pipeline nesta versão: os exercícios usam usuários já existentes na instância, informados na lista de alunos.

12. Documentos complementares

Cada oferta tem uma *data sheet* e um documento de pré-requisitos. O estudo de custos de infraestrutura estima o custo mensal das arquiteturas em Azure, AWS e Google Cloud, com preços públicos e premissas declaradas.

Código	Documento	Tipo
PS-00	Catálogo de Serviços	Catálogo
PS-IMP-01	Implantação da plataforma GitLab em servidores Linux	Data sheet
PR-IMP-01	Pré-requisitos: Implantação da plataforma GitLab em servidores Linux	Pré-requisitos
PS-IMP-02	Implantação da plataforma GitLab em Kubernetes	Data sheet
PR-IMP-02	Pré-requisitos: Implantação da plataforma GitLab em Kubernetes	Pré-requisitos
PS-MIG-01	Migração GitLab para GitLab self-managed	Data sheet
PR-MIG-01	Pré-requisitos: Migração GitLab para GitLab self-managed	Pré-requisitos
PS-MIG-02	Migração GitHub para GitLab self-managed	Data sheet
PR-MIG-02	Pré-requisitos: Migração GitHub para GitLab self-managed	Pré-requisitos
PS-MIG-03	Migração Azure DevOps para GitLab self-managed	Data sheet
PR-MIG-03	Pré-requisitos: Migração Azure DevOps para GitLab self-managed	Pré-requisitos
PS-MIG-04	Migração Bitbucket Cloud para GitLab self-managed	Data sheet
PR-MIG-04	Pré-requisitos: Migração Bitbucket Cloud para GitLab self-managed	Pré-requisitos
PS-AVA-01	Avaliação de maturidade DevSecOps e Customer Success Plan	Data sheet

Código	Documento	Tipo
PR-AVA-01	Pré-requisitos: Avaliação de maturidade DevSecOps e Customer Success Plan	Pré-requisitos
PS-AVA-02	Levantamento da plataforma	Data sheet
PR-AVA-02	Pré-requisitos: Levantamento da plataforma	Pré-requisitos
PS-DOR-01	Painel executivo de métricas DORA	Data sheet
PR-DOR-01	Pré-requisitos: Painel executivo de métricas DORA	Pré-requisitos
PS-CAP-01	Capacitação na plataforma GitLab	Data sheet
PR-CAP-01	Pré-requisitos: Capacitação na plataforma GitLab	Pré-requisitos
PS-ONB-01	Onboarding do cliente na plataforma GitLab	Data sheet
PR-ONB-01	Pré-requisitos: Onboarding do cliente na plataforma GitLab	Pré-requisitos
PS-OPS-01	Operação assistida	Data sheet
PS-CUS-01	Estudo de custos de infraestrutura	Estudo

Versão web interativa (arquiteturas, pipelines, matriz de migração e calculadora de custos):
<https://pointerdevsecops.com.br>

13. Próximos passos

Para iniciar, a Pointer agenda uma reunião de discovery e, quando contratado, o Levantamento da Plataforma, que mede por API o ambiente atual e subsidia a proposta com dados. A proposta técnico-comercial é elaborada a partir desse levantamento.

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

PS-00 · versão 1.2 · 1 de outubro de 2026. GITLAB is a trademark of GitLab Inc. in the United States and other countries and regions. A Pointer é parceira GitLab Select.