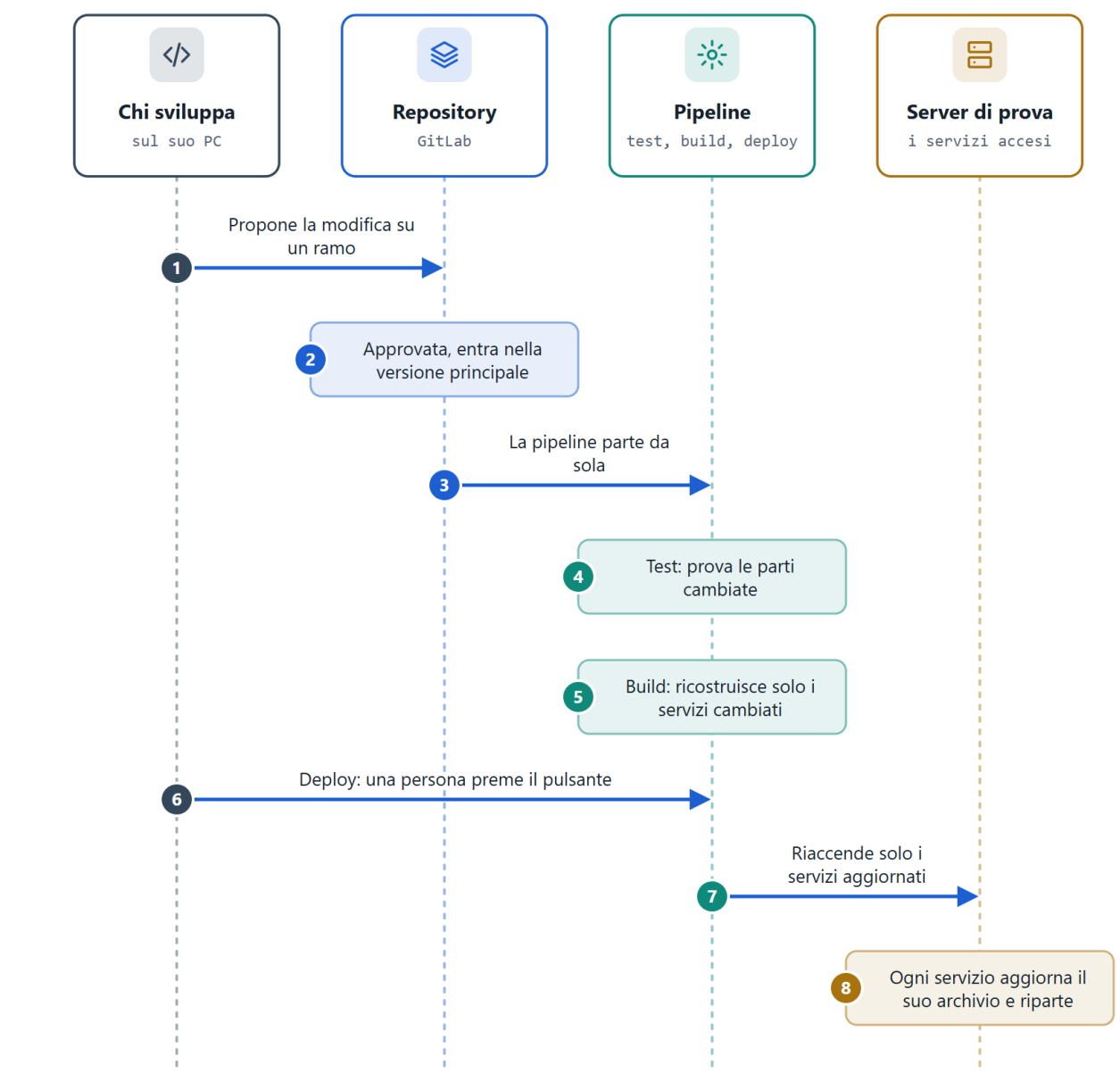
JM1.1 - Dal codice al server

In una frase. Una modifica approvata arriva sul server di prova con un clic, e si ricostruiscono solo i servizi che sono cambiati davvero.

ADSWnext non è un programma solo: è fatto di più servizi, ognuno con il suo compito. Aggiornarli a mano, uno per uno, vorrebbe dire ricordarsi ogni volta l'ordine, i comandi e le impostazioni. Basta una dimenticanza per lasciare il server a metà.

Per questo il lavoro lo fa la **pipeline**: una sequenza di passi automatici che parte ogni volta che una modifica entra nella versione principale del codice. Prova le parti cambiate, prepara i servizi nuovi e poi si ferma ad aspettare. L'ultimo passo, mettere i servizi nuovi al posto dei vecchi, parte solo quando una persona preme un pulsante.



Dal PC di chi sviluppa al server di prova: la pipeline prova, ricostruisce e aspetta il clic

> **Cosa ci guadagni.**
> • **Un aggiornamento è un clic.** Nessuno si collega al server per digitare comandi. La pipeline fa sempre gli stessi passi, nello stesso ordine, e lascia scritto cosa ha fatto.
> • **Si ricostruisce solo ciò che è cambiato.** Se la modifica tocca solo le pagine, si rifà solo l'app web. Una modifica alla sola documentazione non ricostruisce niente.
> • **Chi non è toccato non si ferma.** I servizi rimasti uguali restano accesi. Chi sta lavorando nelle maschere storiche non perde la sessione per un aggiornamento che riguarda le stampe.

Oggi l'installazione che la pipeline tiene aggiornata è una: il **server di prova**, dove si guarda il lavoro prima di portarlo altrove. Una seconda installazione accanto a quella e la distribuzione ai

clienti sono ancora da fare: c'è un piano approvato, non ancora realizzato.

Questo libro racconta cosa gira sul server di prova, come arriva un aggiornamento, come si controlla che sia andato bene e cosa si può fare quando non va.

> **Per chi sviluppa.**
> • La pipeline è il file `.gitlab-ci.yml`, in radice. Gira solo sulla versione principale (`main`), su un runner installato sul server di prova: costruisce ed esegue tutto lì, con Docker.
> • Lo stack è `devops/deploy/docker-compose.deploy.yml`; la spiegazione completa è `devops/deploy/README.md`.
> • Il piano per i clienti è `docs/devops/PIANO-DISTRIBUZIONE-CLIENTI.md`: immagini da un registro, un file di installazione, un comando unico. Non è ancora iniziato.

Revisione #2
Creato 17 novembre 2026 07:59:38 da Giuseppe Anelli
Aggiornato 17 novembre 2026 09:38:54 da Genesys Software