

Field Enablement Playbook

Demoing Simplified Container Orchestration & Deployment with GitLab

At-a-Glance

Asset Type: Quick-Start Playbook + Hands-On Lab

Target Audience: Solutions Architects, Account Executives, Customer Success Managers

Platform: Highspot / Cornerstone

Time to Complete: 20–25 minutes

Learning Objective

After completing this playbook, you can run a confident 10–15 minute live demo showing how GitLab simplifies container deployment and orchestration compared to custom service adapters and blueprints.

Why This Matters to Customers

- Enterprise teams often manage Docker Swarm with custom adapters, manual blueprint configurations, network/volume connections, and certificate handling.
- This creates complexity, maintenance overhead, and slower releases.
- GitLab unifies everything with native CI/CD, Container Registry, Auto DevOps, and built-in security scanning.
- The result is faster, more secure, and repeatable deployments with less custom work.

Buyer Persona

DevOps leads or platform engineers who currently manage Docker Swarm clusters with custom adapters or complex deployment topologies and want to reduce tool sprawl.

Key Value Messages

- Replace manual adapter uploads and complex topologies with declarative pipelines.
- Shift security left with automatic scanning during build and deploy.
- Achieve scalable orchestration with fewer tools and better visibility.

Hands-On Demo Script

Step 1: Set the Stage (1 minute)

“Today I’ll show how GitLab simplifies what used to require custom service adapters and blueprints.”

Step 2: Project & Image Setup (3–4 minutes)

1. Create or open a GitLab project.
2. Add a basic Dockerfile and .gitlab-ci.yml that builds and pushes the image to GitLab’s Container Registry.
3. Trigger the pipeline to build the container image.

Step 3: Build, Scan & Push (4 minutes)

1. Show the pipeline running with automatic **Container Scanning** and SAST.
2. Demonstrate the image appearing in the Container Registry with secure authentication (using CI_JOB_TOKEN or deploy tokens).
3. Highlight how this replaces manual credential setup in traditional adapters.

Step 4: Deployment & Orchestration (5 minutes)

1. Show a deployment job that pushes to Docker Swarm (for self-managed GitLab) or Kubernetes.
2. Highlight analogous concepts: replicas/scaling, networks, volumes, and secrets — now defined declaratively in the pipeline.
3. Run the deployment and verify the service is live.

Step 5: Validation & Business Value Close (2 minutes)

- Verify logs, service status, and rollback options.
- Close with: “This repeatable pipeline replaces drag-and-drop blueprints and manual credential setup.”

Objection Handlers

- “We already use Docker Swarm heavily” → “GitLab supports Swarm mode while adding native scanning, registry, and pipeline automation — most teams see reduced maintenance and faster releases.”

Quick Reference Card

- 5-step demo checklist
- Sample .gitlab-ci.yml snippet for container build + deploy

- Key talking points
- Links to GitLab Container Registry and Container Scanning docs

Measurement & Reinforcement

- Track playbook views and demo usage in Highspot.
- Post-demo survey: “How confident are you running a container deployment conversation?” (target 4+/5).
- Reinforcement: Pair with a short recorded demo video and include in role-based learning paths.

Created by: Constanze Kratel

Date: March 2026