پیشنهاد مدیریتی

# پلتفرم متمرکز مدیریت، کنترل و توزیع امن Container Image و Artifact

Nexus Repository + Portal + PostgreSQL + Monitoring

این سند برای تصمیم‌گیری مدیریتی درباره رسمی‌سازی و عملیاتی‌سازی یک پلتفرم داخلی مدیریت Artifact تهیه شده است؛ پلتفرمی که فراتر از یک Mirror Registry ساده عمل می‌کند و کنترل متمرکز، انتشار کنترل‌شده، دسترسی امن، ثبت رویدادها و مسیر توسعه DevSecOps را در یک معماری Dockerized فراهم می‌سازد.

### دامنه تاییدشده

Nexus، Portal، PostgreSQL، Security Center، Artifact Publishing و Monitoring.

### تصمیم موردنیاز

تایید رسمی‌سازی پلتفرم به‌عنوان سرویس زیرساختی مدیریت‌شده.

### ارزش اصلی

کاهش وابستگی عملیاتی به رجیستری‌های بیرونی و ایجاد نقطه کنترل داخلی برای Pull و Push.

## درخواست تصمیم

تصویب ادامه تثبیت، بهره‌برداری و توسعه کنترل‌شده پلتفرم به‌عنوان مرجع داخلی توزیع Image و Artifact، همراه با الزام به تکمیل سیاست‌های امنیتی، Backup/Restore دوره‌ای، TLS و اتصال تدریجی به CI/CD.

# مساله‌ای که پروژه حل می‌کند

در محیط‌های توسعه، عملیات و شبکه‌های محدود، اتکا به رجیستری‌ها و مخازن بیرونی تنها یک موضوع سرعت نیست؛ این وابستگی می‌تواند منبع شکست استقرار، نبود مالکیت روی نسخه‌ها، ضعف در کنترل دسترسی و کاهش قابلیت ردیابی باشد.

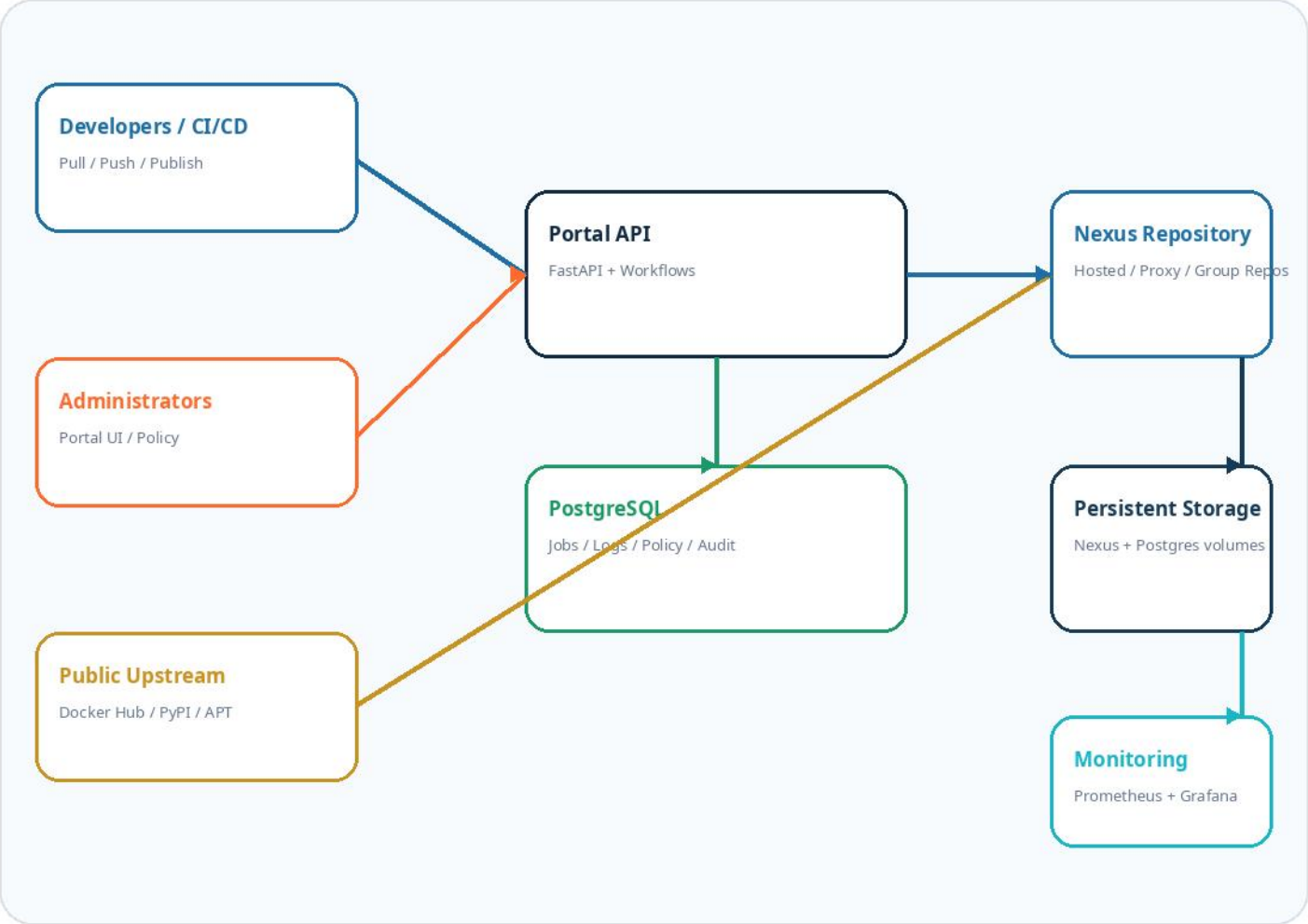
| Management Risk | Operational Impact | Current Problem |
|---|---|---|
| وابستگی به سرویس‌های خارج از کنترل سازمان | کندی، خطا یا توقف Pull در زمان استقرار | وابستگی تکراری به Docker Hub و مخازن عمومی |
| ضعف مالکیت و دشواری بازبینی نسخه‌ها | پراکنده شدن Image های داخلی و نسخه‌های عملیاتی | نبود نقطه متمرکز برای Push داخلی |
| ریسک امنیتی و نبود پاسخگویی | امکان استفاده افراد یا سیستم‌های غیرمجاز از Mirror | Anonymous Access یا دسترسی بدون سیاست مشخص |
| کاهش قابلیت Audit و تصمیم‌گیری فنی | دشواری پیگیری خطا، Digest، Checksum و خروجی کارها | نبود تاریخچه منسجم برای انتشار Artifact |
| هزینه عملیاتی و ریسک پر شدن دیسک | مصرف Storage و دشواری Cleanup | رشد کنترل‌نشده داده و Cache |

### جمع‌بندی مدیریتی

یک Mirror Registry ساده برای افزایش سرعت Pull مفید است، اما برای مدیریت مالکیت، انتشار داخلی، کنترل کاربر و گروه، محدودسازی IP، Audit Log، Backup/Restore و تصمیم‌گیری عملیاتی کافی نیست. نیاز واقعی، یک پلتفرم داخلی با سیاست، فرآیند و قابلیت مدیریت مرکزی است.

# معماری پلتفرم داخلی Artifact

پروژه فعلی یک استقرار Dockerized چندسرویسی ارائه می‌کند که Portal را به Docker Engine، Nexus REST API، ابزارهای دریافت Package و PostgreSQL متصل می‌کند. داده‌های عملیاتی Portal در PostgreSQL نگهداری می‌شود و Nexus مسئول نگهداری Repository ها و Artifact ها است.




### جریان Pull/Proxy

در صورت استفاده از Proxy Repository، درخواست‌ها از Nexus عبور کرده و در Cache داخلی ذخیره می‌شوند.

### جریان Push/Hosted

Image ها و Package های داخلی از Portal یا Docker Client به Hosted Repository منتقل می‌شوند.

### مرز امنیتی

Security Center نقش‌ها، کاربران، IP/CIDR و Anonymous Access را به‌صورت متمرکز مدیریت می‌کند.

# مقایسه مستقیم: Mirror ساده در برابر پلتفرم داخلی مدیریت‌شده

Mirror Registry برای Cache کردن Pull ها ارزشمند است، اما هدف این پروژه ایجاد حاکمیت عملیاتی و امنیتی روی توزیع Artifact است. جدول زیر فقط قابلیت‌هایی را به‌عنوان Implemented نشان می‌دهد که در پروژه فعلی تایید شده‌اند.

| معیار | Basic Mirror Registry | Nexus Internal Platform |
|---|---|---|
| **هدف اصلی** | Pull acceleration | Governance + controlled distribution |
| **Private Push / Hosted** | Limited / manual | **Implemented: docker-hosted, PyPI, APT, raw** |
| **Proxy / Cache** | Core capability | Available via Nexus proxy repos |
| **Repository Group** | Often absent | **Implemented / configurable in Nexus** |
| **Authentication** | Basic or optional | **Implemented: Portal auth + Nexus users** |
| **RBAC** | Limited | **Implemented: groups, roles, permissions** |
| **Anonymous lockdown** | Not central | **Implemented via Security Center sync** |
| **Trusted IP/CIDR** | Usually external | **Implemented: preview/apply firewall policy** |
| **Publishing workflow** | Not provided | **Implemented: Docker, archive, Python, Debian jobs** |
| **Audit / logs** | Minimal | **Implemented: audit log, job logs, history** |
| **Backup / Restore** | External task | Scripts provided for Nexus/PostgreSQL/config |
| **Monitoring** | External task | Prometheus/Grafana stack configured |
| **Vulnerability scanning** | Not inherent | Planned/recommended; Grype flag exists |
| **High Availability** | Not inherent | Future phase / not confirmed |

**نتیجه: Mirror ساده یک قابلیت فنی برای Cache است؛ پلتفرم پیشنهادی یک سرویس زیرساختی برای مالکیت، سیاست‌گذاری، انتشار، ردیابی و کنترل دسترسی Artifact ها است.**

# ارزش پیشنهادی برای سازمان

مزیت پروژه در ترکیب چند قابلیت عملیاتی است: Repository Manager، Portal مدیریتی، کنترل دسترسی، انتشار Artifact، ذخیره پایدار، گزارش و نظارت. این ترکیب ریسک‌های عملیاتی را کاهش می‌دهد و یک مسیر قابل توسعه برای DevSecOps ایجاد می‌کند.

### امنیت و حاکمیت

مدیریت کاربر/گروه، RBAC، محدودسازی Anonymous Access، Trusted IP/CIDR، Audit Log و امکان توسعه سیاست‌های امنیتی.

### ارزش عملیاتی

Pull های تکراری سریع‌تر و قابل اتکاتر، کاهش وابستگی به Upstream، منبع استاندارد برای استقرار، و نگهداری داخلی Image ها و Package ها.

### ارزش مدیریتی

مالکیت سرویس مشخص، شفافیت وضعیت، مسیر Roadmap قابل اندازه‌گیری، کاهش عدم قطعیت عملیاتی و امکان کنترل رشد Artifact ها.

### توانمندسازی DevOps

آمادگی اتصال به CI/CD، تفکیک Repository ها، انتشار کنترل‌شده Docker Image، Python Package و Debian Package از Portal.

## قابلیت‌های تاییدشده کلیدی

**Dockerized Stack**
Nexus, Portal, PostgreSQL, Traefik, Prometheus, Grafana

**Portal State**
Jobs, logs, publishing history, security policy, audit records

**Artifact Publishing**
Docker by name, Docker archive, Python fetch, Debian fetch

**Security Center**
Groups, users, permissions, IP rules, Nexus sync, firewall preview/apply

**Operations**
Health checks, persistent volumes, logging limits, backup/restore scripts

# وضعیت فعلی و واقعیت‌های اجرایی

## Current Status

| | |
|---|---|
| Completed | معماری Monitoring، Portal، PostgreSQL، Nexus، Docker Compose و Health Checkها در مخزن تعریف شده‌اند. |
| Completed | Artifact Publishing و Security Center در Portal پیاده‌سازی شده و APIهای مرتبط وجود دارند. |
| Completed | Audit Log، Backup/Restore scripts، گزارش‌ها و Job Logs در ساختار پروژه وجود دارند. |
| Pending validation | آزمون دوره‌ای Restore، ظرفیت‌سنجی Storage و تست بار برای محیط تولید باید formalize شود. |
| Future phase | TLS، HA سازمانی، Vulnerability Scanning اجباری و Signing/Policy enforcement باید در فازهای بعدی تکمیل شوند. |

## ریسک‌های کلیدی و کاهش ریسک

| | |
|---|---|
| Single-node dependency | تعریف Backup/Restore آزمون‌شده، برنامه DR و بررسی HA در فاز بعد. |
| Storage growth | Retention Policy و Cleanup Policy برای Cache و Hosted Repository ها. |
| Credential protection | استفاده از Secrets، PORTAL_ADMIN_PASSWORD_HASH امن و حذف رمزهای plaintext. |
| TLS and certificates | فعال‌سازی HTTPS، Secure Cookies و گواهی معتبر برای محیط تولید. |
| Vulnerability gap | فعال‌سازی Grype یا ابزار سازمانی و Block policy برای Critical/High در فاز امنیتی. |
| Firewall lockout risk | الزام افزودن IP مدیر و CI/CD قبل از Apply IP Firewall و استفاده از Preview. |

## Roadmap پیشنهادی

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| **Stabilization** | **Governance** | **CI/CD** | **Security** | **Scale / DR** |
| سخت‌سازی تنظیمات، تست End-to-End، Restore test | نهایی‌سازی RBAC، Repository ownership، Retention | اتصال Pipeline ها و استاندارد Push/Pull | Vulnerability Scanning، Signing، Policy enforcement | HA یا DR، Capacity planning، مانیتورینگ SLA داخلی |

# شاخص‌های موفقیت و تصمیم مدیریتی

برای تبدیل پروژه به سرویس زیرساختی مدیریت‌شده، موفقیت باید با شاخص‌های قابل اندازه‌گیری و نه صرفا نصب نرم‌افزار سنجیده شود. مقادیر زیر به‌عنوان KPI پیشنهادی مطرح می‌شوند و نیازمند تعیین Baseline در آغاز بهره‌برداری هستند.

**Cache Hit Ratio**
نسبت درخواست‌های پاسخ‌داده‌شده از Cache داخلی

**Internal Pull Share**
درصد Pull هایی که از رجیستری داخلی سرو می‌شوند

**Unauthorized Attempts**
تعداد تلاش‌های دسترسی غیرمجاز و روند آن

**Backup Success**
نرخ موفقیت Backup و Restore test دوره‌ای

**CI/CD Adoption**
تعداد Pipeline های متصل به Registry داخلی

**Managed Repositories**
تعداد Repository های مدیریت‌شده و دارای مالک

**Failed External Pulls**
کاهش خطاهای استقرار ناشی از Upstream

**Retention Compliance**
میزان رعایت Retention/Cleanup Policy

## Requested Decision

**تصویب رسمی‌سازی، تثبیت و عملیاتی‌سازی Nexus Internal Registry Platform به‌عنوان سرویس زیرساختی مدیریت‌شده برای کنترل، توزیع و حاکمیت Artifact های داخلی.**

## اقدامات فوری پیشنهادی

- تایید مالک سرویس، دامنه استفاده اولیه و سیاست دسترسی تیم‌ها.
- اجرای تست کامل Backup/Restore و مستندسازی Runbook عملیاتی.
- فعال‌سازی HTTPS/TLS و hardening رمزها و Cookieها برای تولید.
- تعریف Retention/Cleanup Policy و ظرفیت‌سنجی Storage.
- اتصال مرحله‌ای CI/CD و فعال‌سازی کنترل‌های امنیتی فاز بعد.

**جمع‌بندی: انتخاب یک Mirror ساده مشکل سرعت Pull را تا حدی کاهش می‌دهد؛ اما این پروژه پایه یک پلتفرم حاکمیتی و عملیاتی است که می‌تواند منبع استاندارد، امن و قابل ردیابی برای Image و Artifact های سازمان باشد.**