Privates IT-Praxisprojekt

Linux-Server mit Docker

Gemeinsame Serverbasis für Cloud, Ticketsystem, Kommunikation und Webanwendungen mit getrennten Diensten und Datensicherungen.

Stand 09.11.2026 | Server und Container geprüft

Ziel und Nutzen

Eine eigene Serverumgebung bündelt mehrere praktische IT-Projekte. Das Ziel ist ein nachvollziehbarer Betrieb mit getrennten Anwendungen, passenden Zugriffsrechten und einer überprüfbaren Bereitstellung. Die Umgebung dient als privates Lern- und Praxisprojekt im Bereich Systemintegration.

Technischer Aufbau

Der Linux-Server betreibt Apache als Webserver und vorgeschalteten HTTPS-Proxy. Docker Compose verwaltet unter anderem OTRS, Mattermost, Nextcloud und TeamSpeak in eigenen Stacks. Anwendungen mit eigener Datenbank erhalten getrennte Datenbankdienste und persistente Ablagen. Weitere Webanwendungen laufen als systemd-Dienste.

Linux und systemd	Betriebssystem, Dienstverwaltung und zeitgesteuerte Aufgaben
Docker Compose	Anwendungen, Datenbanken, Volumes und Ressourcenlimits
Apache und Let's Encrypt	Domainzuordnung, HTTPS und Weiterleitung zu Anwendungen

Umsetzung

1. Vor jeder Bereitstellung werden bestehende Dienste, belegte Ports und verfügbare Ressourcen geprüft.
2. Die Anwendungen erhalten eigene Projektkonfigurationen. Daten liegen in persistenten Volumes oder gesonderten Anwendungsablagen.
3. Webdienste werden über ihre Domain angesprochen. Für die administrativen Schnittstellen wird der tatsächlich notwendige Zugriffsweg festgelegt.
4. Vor Änderungen entstehen Sicherungen. Anschließend werden Dienststatus, öffentliche Endpunkte und die betroffenen Bedienabläufe geprüft.

Linux-Server mit Docker

Prüfung und Ergebnis

Am 9. November 2026 wurden die laufenden Container von OTRS, Mattermost, Nextcloud, TeamSpeak und Lumos geprüft. Die öffentlichen Webendpunkte von Portfolio, Ticketsystem, Chat, Cloud und weiteren Anwendungen antworteten erfolgreich. Ein HTTP-Erfolg bestätigt die Erreichbarkeit eines Endpunkts. Die jeweiligen Dokumentationen beschreiben zusätzliche fachliche Prüfungen.

Betrieb und Wartung

Für den Betrieb gehören Ressourcenverbrauch, Speicherplatz, Zertifikatserneuerung und Hintergrundaufgaben zur regelmäßigen Kontrolle. OTRS und Mattermost verfügen über aktive Sicherungstimer. Die Sicherung umfasst je nach Anwendung Datenbank, Dateidaten und Konfiguration. Ein Wiederherstellungstest prüft einen gesicherten Stand in einer getrennten Umgebung.

Grenzen und nächste Schritte

Die Dienste teilen sich die Ressourcen eines einzelnen Servers. Eine Kopie der Daten auf diesem Server schützt allein noch nicht vor dessen vollständigem Ausfall. Externe Sicherungen, eine festgelegte Aufbewahrung und zusätzliche Überwachung sind mögliche nächste Ausbauschritte.

Praktische Kenntnisse

Linux-Dienstverwaltung, Containerbetrieb, Domainzuordnung, TLS, Protokollanalyse, Datensicherung und strukturierte Fehlerbehebung.

Verknüpfte Projekte

OTRS-Ticketsystem · Mattermost-Kommunikationsplattform · Nextcloud - PaynsCloud · TeamSpeak-Sprachserver · Mailserver und OTRS-Mailverarbeitung

Projektseite und technische Quellen

Diese Dokumentation online
https://payns.info/dokumentation/linux-docker.html

Docker Compose
https://docs.docker.com/compose/

Apache Reverse Proxy
https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

Die Umsetzung und die Aufbereitung der Dokumentation wurden KI-gestützt unterstützt. Die Beschreibung stützt sich auf die vorhandene Projektkonfiguration, Betriebsnachweise und die genannten Prüfungen.