

Administration

- [Branding & Settings](#)
- [Upgrades & Maintenance](#)
- [Troubleshooting](#)

Branding & Settings

The CMS is designed to be white-label and flexible. Platform administrators can adjust branding and system settings to align with the service provider or enterprise identity.

Branding Options

- **Logos:** Upload light and dark mode variants.
- **Colors & Themes:** Adjust portal colors, typography, and layout to match brand guidelines.
- **Naming:** Override display names (e.g., “Tenant” → “Department”) for internal deployments.
- **Custom Domains:** Bind branded domain names (e.g., portal.<org>.cloud).

System Settings

- **Identity:** Configure authentication providers (Entra, Google, local accounts).
- **Currencies:** Define default billing currency and FX handling.
- **Notifications:** Configure SMTP settings for system-generated emails (billing, alerts, password resets).
- **Licensing:** Apply CMS license keys or renewals.

Upgrades & Maintenance

The CMS is delivered as containerized services. Maintenance focuses on applying updates, securing the environment, and ensuring continuous availability.

Upgrades

- **Container Images:** Pull updated images from the registry.
- **Rolling Updates:** Apply updates one container at a time to avoid downtime.
- **Schema Migrations:** Apply database migrations as part of the upgrade process.

Maintenance Tasks

- **Database Maintenance:** Regularly optimize and purge old usage records as per retention policy.
- **Certificate Renewal:** Update TLS certificates before expiry.
- **License Management:** Validate CMS license validity and renew as required.

Recommendations

- Maintain separate test/staging environments for validating upgrades.
- Schedule maintenance windows for major version changes.
- Monitor system health post-upgrade to confirm normal operations.

Troubleshooting

When issues occur, the CMS provides multiple layers of diagnostic tools. Effective troubleshooting combines log analysis, API checks, and data validation.

Common Issues

- **Login Failures:** Often linked to misconfigured identity providers or expired federation tokens.
- **Usage Not Processing:** Check region connectivity and raw usage ingestion jobs.
- **Billing Errors:** Confirm pricing tables are consistent and schema migrations are complete.
- **Commission Mismatches:** Verify commission rates are scoped correctly (partner vs distributor).

Tools & Techniques

- **Logs:** Inspect application logs for API errors and job failures.
- **API Checks:** Use Swagger endpoints to verify health and validate query results.
- **Database Queries:** Run direct MySQL queries to confirm data integrity in billing or usage tables.
- **Health Dashboards:** Review monitoring metrics for system load and resource exhaustion.

Escalation

- Gather logs, failing API calls, and reproduction steps.
- Engage platform vendor support with diagnostic bundles.
- Document resolution in the internal knowledge base for future reference.