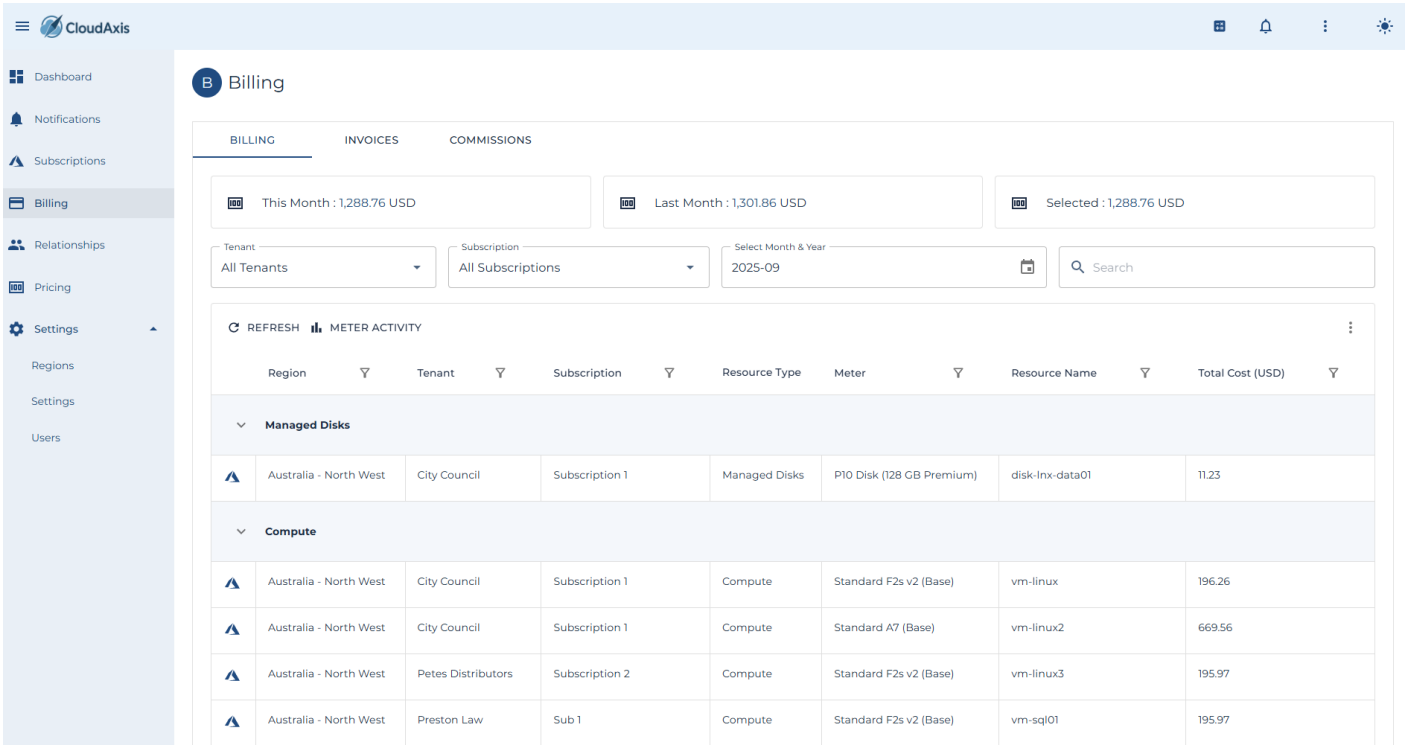
Introduction

Overview

The **CMS for Azure Stack Hub is** a management and billing platform for Azure Stack Hub that connects technical service delivery with commercial operations. It is designed for service providers, enterprises, and partners who need to operate Azure Stack Hub as a structured, multi-tenant cloud service.



At its core, the CMS provides a layered model that reflects how services are delivered and monetized:

- **Platform Administrators** operate and configure the system. They publish pricing, manage global settings, and oversee usage processing across regions.
- **Distributors** represent the top commercial tier beneath the platform. They manage multiple partners and have visibility into aggregated usage, billing, and commissions.
- **Partners** act as the customer-facing organizations. They onboard tenants, manage subscriptions, and handle tenant-level billing and support.
- **Tenants** are the consuming organizations. They manage their own subscriptions and users, while billing and usage flow upward to the partner and distributor.
- **Subscriptions** are the unit of consumption within a tenant. They tie workloads to plans and quotas, generate usage records, and form the basis for billing and invoicing.

This hierarchy ensures that both technical and commercial responsibilities are aligned. Usage flows upward from subscriptions to tenants, partners, and distributors, while pricing, commissions, and governance flow downward from the platform.

Key capabilities include:

- **Unified Multi-Tier Management** – Administrators, distributors, partners, and tenants are modeled consistently in the system.
- **Automated Billing** – Usage data is collected, rated, and reconciled into invoices at the subscription and tenant level.
- **Flexible Pricing and Discounts** – Flat and tiered rates, credits, and overrides can be applied at multiple scopes.
- **Commission Frameworks** – Built-in logic for calculating and attributing commissions to partners and distributors.
- **API-Driven Operations** – All functions are exposed through a standards-based REST API for automation and integration.
- **Branding and Delegation** – Each level of the hierarchy can be branded appropriately, with role-based access control ensuring least-privilege delegation.

The CMS provides the foundation to operate Azure Stack Hub as a commercial service, supporting both external service providers and internal enterprise IT. By unifying consumption data, pricing models, billing, and commissions under a single platform, it enables organizations to deliver cloud services with transparency, accountability, and commercial accuracy.

System Requirements

The CMS is delivered as a .NET-based application backed by a MySQL database. It is packaged for deployment in containerized or virtualized environments and can be operated either alongside Azure Stack Hub infrastructure or on external platforms.

Software Requirements

- **Database**: MySQL 8.0 or later
- **Application Runtime**: .NET 8
- **Frontend**: Blazor Server (runs as part of the API service or in a separate container)
- **Optional Components**: Docker or Kubernetes for container orchestration

Hardware Requirements

- **Production Deployment**
  - Minimum 4 vCPUs
  - 16 GB RAM
  - 100 GB storage (expandable based on retention of usage and billing data)
- **Development or Evaluation Deployment**
  - Minimum 2 vCPUs
  - 8 GB RAM
  - 50 GB storage

Network Requirements

- Outbound HTTPS connectivity to Azure Stack Hub administrative endpoints
- Inbound HTTPS (TCP 443) for API and portal access
- Optional VPN or ExpressRoute for hybrid or private connectivity scenarios

Identity Requirements

- Integration with Microsoft Entra ID or another supported identity provider
- Federated authentication supported through token validation endpoints
- Role-based access control (RBAC) enforced throughout the portal and API

High Availability and Recovery

The CMS can be deployed in a high-availability configuration using container clustering or database replication. Backup and recovery procedures should be established to protect the MySQL database and configuration state. For detailed operational guidance, see *Backups & DR* in the Operations chapter.

BookStack Publishing

BookStack Publishing

Overview

This page confirms that the CMS documentation agent can publish markdown content to BookStack.

Purpose

The BookStack publishing process reads markdown files from the Documentation folder and creates or updates matching BookStack pages.

Known Gaps

> Manual Review Required: Replace this test page with the final publishing workflow documentation.