Cloud-Hosted Helpdesk Ticketing System (osTicket) Implementation Lab

Author: Darnell Maiden | **Certifications:** CompTIA Security+, Google IT Support Professional Certificate, Google Cybersecurity Professional Certificate | **Target Environment:** DigitalOcean Ubuntu 24.04 LTS, Apache2, MariaDB, PHP |
PORTFOLIO WEBSITE: https://www.saintmaiden.site/

EXECUTIVE SUMMARY:

PROBLEM:	SOLUTION:	IMPACT:
Organizations and IT support teams frequently struggle with unorganized support emails, lack of ticket tracking, absence of service level agreements (SLAs), and failure to maintain an audit trail for hardware provisioning requests and user incidents.	**Deployed a fully functional, cloud-hosted open-source helpdesk ticketing system (osTicket) on an Ubuntu 24.04 LTS Linux Droplet utilizing a hardened LAMP (Linux, Apache, MariaDB, PHP) stack, least-privilege database user isolation, and Apache virtual host routing.**	**Established a centralized enterprise ticketing portal capable of handling user incident tracking, automated department routing, and secure agent/client workflows while eliminating local hosting bottlenecks.**

TECHNICAL TOPOLOGY/SPECIFICATIONS:

COMPONENT/ROLE	SYSTEM/SPACE NAME	NETWORK MODE & ACCESS	SERVICES & ROLES
Cloud Server	**helpdesk-lab-01**	**DigitalOcean Droplet (Public IPv4)**	**Ubuntu 24.04 LTS, UFW Firewall**
Web Tier	**Apache2**	**HTTP (80)/HTTPS (443)**	**Virtual Host Routing,** `mod_rewrite`, `.htaccess` **Support**
Database Tier	**MariaDB Server**	**Local Loopback (localhost)**	**Relational Data Store** (`osticket_db`), **Least-Privilege Access**
Application Runtime	**PHP 8.x + Extensions**	**CLI/Web Server Module**	`php-mysql`, `php-imap`, `php-gd`, `php-intl`, `php-apcu`, `php-mbstring`

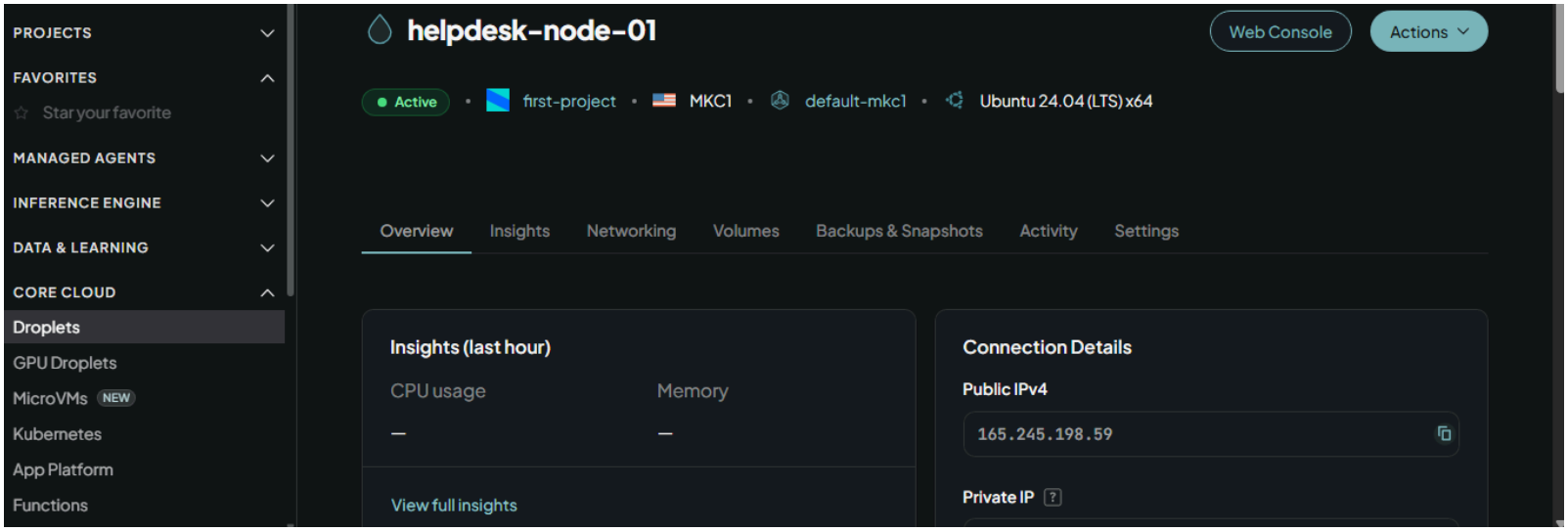
PHASE 1: DROPLET PROVISIONING & NETWORK SECURITY HARDENING

ACTION: **Provisioned an Ubuntu 24.04 LTS cloud instance on DigitalOcean, executed system updates, and configured the Uncomplicated Firewall (UFW) to enforce strict network-layer least privilege.**

BASH: **apt update && apt upgrade -y ufw allow OpenSSH ufw allow 80/tcp ufw allow 443/tcp ufw --force enable**

TROUBLESHOOTING ISSUE: **Default cloud images often leave unnecessary ports exposed or lack baseline package updates. Mitigated by updating system repositories and explicitly whitelisting only essential inbound ports.**

WHY?: **Applying UFW rules blocks all unsolicited inbound traffic by default, preventing external attack vectors while allowing authorized administrative and web traffic (SSH, HTTP, HTTPS).**




```
root@helpdesk-node-01:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp (OpenSSH)           ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22/tcp (OpenSSH (v6))      ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)

root@helpdesk-node-01:~#
```

PHASE 2: LAMP STACK & REQUIRED PHP EXTENSIONS DEPLOYMENT

ACTION: **Installed Apache2, MariaDB server, and the complete prerequisite suite of PHP modules required for osTicket's email parsing, internationalization, and caching functions.**

BASH: (**apt install apache2 mariadb-server php libapache2-mod-php php-mysql php-imap php-gd php-intl php-apcu php-mbstring php-xml php-cli php-json -y**)

TROUBLESHOOTING ISSUE: **Standard web server packages omit advanced libraries needed for email piping (`php-imap`) or multi-byte text rendering (`php-mbstring`, `php-intl`), which would cause runtime errors during ticket creation. Resolved by bundling all mandatory extensions during initial installation.**

WHY?:

- `php-imap` **enables automated conversion of incoming emails into support tickets.**
- `php-apcu` **provides in-memory object and opcode caching to reduce database overhead and CPU utilization.**
- `php-gd` **handles image processing for agent avatars and attachments.**

```
root@helpdesk-node-01:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-10-28 16:00:15 UTC; 17min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 32763 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 32766 (apache2)
      Tasks: 11 (limit: 2315)
     Memory: 46.3M (peak: 47.3M)
        CPU: 419ms
     CGroup: /system.slice/apache2.service
             ├─32766 /usr/sbin/apache2 -k start
             └─32768 /usr/sbin/apache2 -k start
```

```
root@helpdesk-node-01:~# systemctl status mariadb
● mariadb.service - MariaDB 10.11.14 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-10-28 15:53:29 UTC; 24min ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 30508 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 15284)
     Memory: 95.7M (peak: 95.9M)
        CPU: 653ms
     CGroup: /system.slice/mariadb.service
             └─30508 /usr/sbin/mariadbd
```

PHASE 3: SECURE RELATIONAL DATABASE SETUP

ACTION: **Hardened the MariaDB database server via** `mysql_secure_installation`**, created a dedicated database (**`osticket_db`**), and provisioned a local least-privilege database user (**`osticket_user@'localhost'`**) with** `utf8mb4` **encoding.**

BASH: **mysql_secure_installation mysql -u root -p**

SQL: **CREATE DATABASE osticket_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci; CREATE USER 'osticket_user'@'localhost' IDENTIFIED BY 'StrongPassword123!'; GRANT ALL PRIVILEGES ON osticket_db.* TO 'osticket_user'@'localhost'; FLUSH PRIVILEGES; EXIT;**

TROUBLESHOOTING ISSUE: **Default database installations contain guest accounts and blank root credentials vulnerable to automated scanners. Mitigated by setting a strong root password, removing test databases, and isolating application permissions.**

WHY?: **Enforces database-level isolation. If the web application is ever compromised, the attacker's blast radius is strictly restricted to** `osticket_db` **without administrative server or cross-database access.** `utf8mb4` **ensures proper support for modern emojis and international character sets.**

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE osticket_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> CREATE USER 'osticket_user'@'localhost' IDENTIFIED BY 'StrongPassword123!';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON osticket_db.* TO 'osticket_user'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> EXIT;
Bye
```

PHASE 4: APPLICATION DEPLOYMENT & VIRTUAL HOST ROUTING

ACTION: **Downloaded and extracted the osTicket release archive into Apache's web root (`/var/www/html/osticket`), assigned proper ownership to the web user (`www-data`), and configured a dedicated Apache Virtual Host file with `mod_rewrite` enabled.**

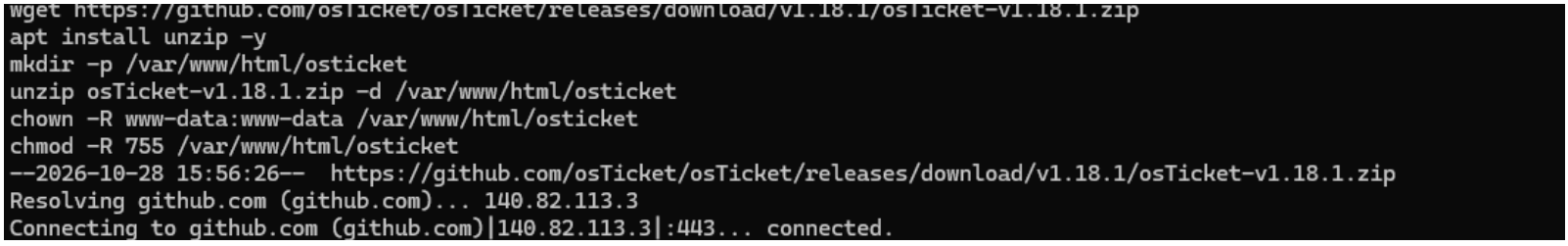
BASH: **cd /tmp wget https://github.com/osTicket/osTicket/releases/download/v1.18.1/osTicket-v1.18.1.zip apt install unzip -y mkdir -p /var/www/html/osticket unzip osTicket-v1.18.1.zip -d /var/www/html/osticket**

chown -R www-data:www-data /var/www/html/osticket

chmod -R 755 /var/www/html/osticket

TROUBLESHOOTING ISSUE: **Default web roots expose parent server folders and fail to process clean URLs. Resolved by pointing the virtual host `DocumentRoot` explicitly to the application upload directory and enabling `AllowOverride All`.**

WHY?: **Virtual host isolation separates application configurations from global web server defaults, while `.htaccess` rewriting rules allow clean, readable routing for support agents and end-users.**





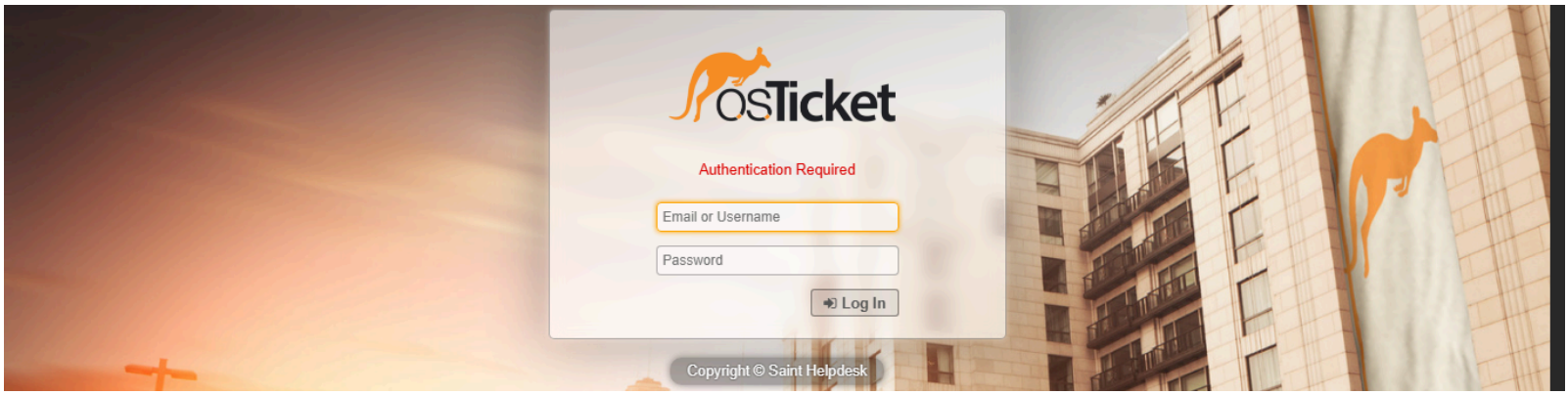
PHASE 5: POST-INSTALLATION HARDENING & SECURITY LOCKDOWN

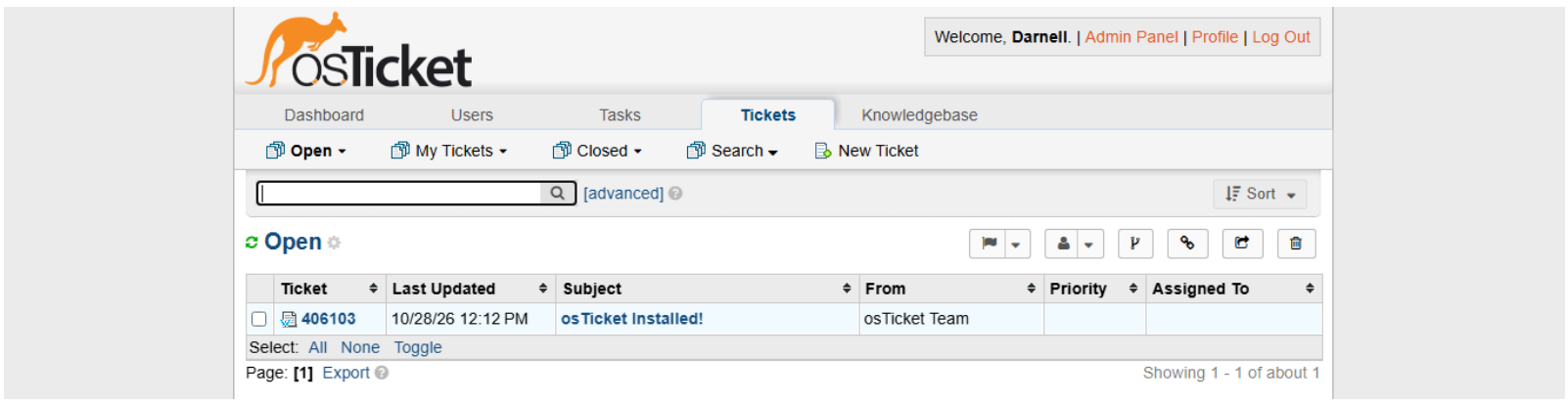
ACTION: **Initialized the configuration template (`ost-config.php`), completed the web-based installation wizard, and subsequently locked file permissions down while purging the public setup directory.**

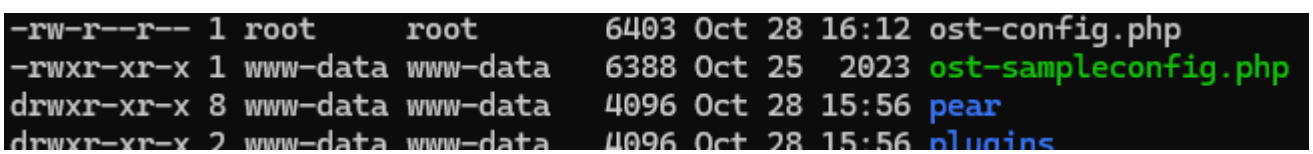
BASH: **cd /var/www/html/osticket/upload/include/ cp ost-sampleconfig.php ost-config.php chmod 666 ost-config.php chmod 644 /var/www/html/osticket/upload/include/ost-config.php rm -rf /var/www/html/osticket/upload/setup**

TROUBLESHOOTING ISSUE: **Leaving configuration files writable allows local web scripts to alter core code, while leaving the `/setup` folder active allows malicious actors to re-run the installer and overwrite admin credentials. Resolved by restoring read-only permissions (`644`) and deleting the setup directory.**

WHY?: **Closes critical post-installation vulnerabilities, ensuring unauthorized local or remote users cannot execute remote code injection or seize administrative control of the helpdesk system.**







TECHNICAL SUMMARY:

SECURITY & NETWORKING DOMAIN	TECH & CONTROLS CONFIGURED
CLOUD INFRASTURCURE	DigitalOcean Ubuntu 24.04 LTS Droplet
NETWORK SECURITY	UFW Firewall (OpenSSH, HTTP, HTTPS enforcement)
WEB SERVER ARCHITECTURE	Apache2, Virtual Hosts, `mod_rewrite`, `AllowOverride All`
DATABASE SECURITY	MariaDB, `mysql_secure_installation`, `utf8mb4`, Least-Privilege User Isolation
APPLICATION HARDENING	File permission restriction (`644`), Setup directory removal

RESULT & BUSINESS IMPACT:

- **ENTERPRISE TICKETING DEPLOYMENT :** Successfully deployed a fully functional, cloud-accessible IT service management portal matching production helpdesk architectures.
- **DEFENSIVE SECURITY POSTURE :** Applied least-privilege access across both network firewall rules and database user permissions, mitigating unauthorized lateral movement.
- **PRODUCTION-READY DOCUMENTATION :** Established a repeatable, documented standard operating procedure (SOP) for deploying and hardening open-scale web applications on Linux cloud environments.

Author: Darnell Maiden | **Certifications:** CompTIA Security+, Google IT Support Professional Certificate, Google Cybersecurity Professional Certificate | **Target Environment:** DigitalOcean Ubuntu 24.04 LTS, Apache2, MariaDB, PHP | **PORTFOLIO WEBSITE:** https://www.saintmaiden.site/