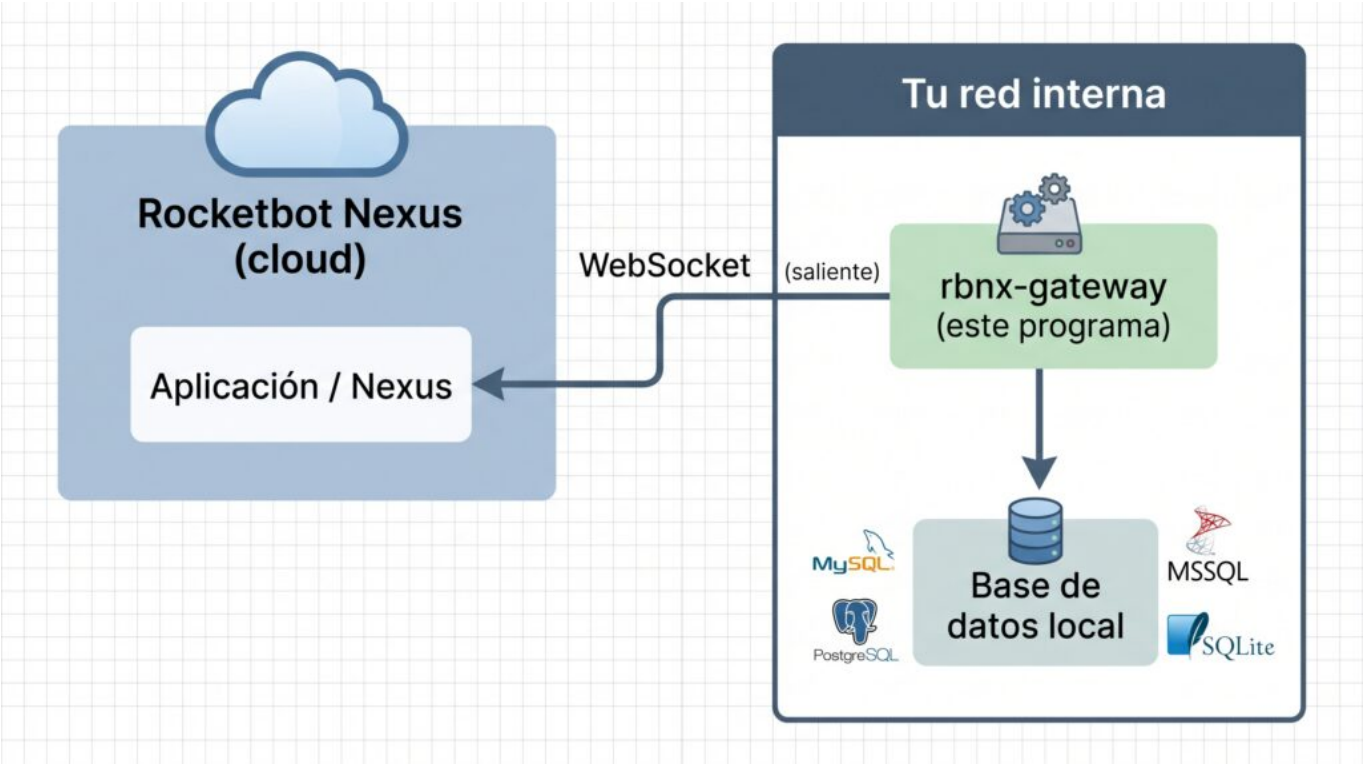
Nexus — Gateway

Manual de Usuario — Rocketbot Nexus Gateway On-Premises

¿Qué es el Gateway On-Premises?

El Gateway On-Premises es un agente ligero que se ejecuta **dentro de tu red corporativa** y permite que Rocketbot Nexus acceda a bases de datos locales sin exponer credenciales ni abrir puertos entrantes en el firewall.



La conexión es **siempre saliente** desde tu red hacia Nexus. Tus credenciales de base de datos **nunca salen** de tu infraestructura.

Requisitos

Plataforma	Requisito mínimo
Windows	Windows 10 / Server 2016 o superior
macOS	macOS 12 (Monterey) o superior
Linux	glibc 2.17+ (CentOS 7, Ubuntu 16.04 o superior)

Para MSSQL: es necesario instalar el driver ODBC de Microsoft por separado (ver sección MSSQL).

Conectividad: el servidor debe poder hacer conexiones HTTPS/WSS salientes al puerto 443 hacia la URL de tu instancia de Nexus.

Instalación

Binario

1. Descarga el binario para tu sistema operativo desde el panel de Nexus **(Configuración → Gateway Tokens → Descargar binario)**
2. Dale permisos de ejecución (macOS/Linux):**chmod +x rbnx-gateway**
3. Verifica que funciona:./**rbnx-gateway --version** # rbnx-gateway 1.0.0

Configuración

1. Obtener un token de gateway

1. Inicia sesión en tu instancia de Rocketbot Nexus.
2. Ve a **Configuración → Gateway On-Premises**.
3. Haz clic en **Nuevo token**, asigna un nombre descriptivo (ej. Servidor Producción).
4. **Copia el token inmediatamente** — solo se muestra una vez. El token tiene el formato: gwt_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

2. Crear el archivo de configuración

Copia la plantilla de ejemplo:

```
cp config.example.yaml config.yaml
```

Edita config.yaml con tus datos:

```
server:
  url: "wss://app.tuempresa.com/gateway"   # URL de tu instancia Nexus
  token: "gwt_xxxxxxxxxxxxxxxxxxxxxxxx"    # Token obtenido en el paso
anterior
  reconnect_interval_seconds: 5           # Segundos entre reconexiones
  max_reconnect_attempts: 0               # 0 = intentos infinitos

database:
  type: "mysql"          # mysql | postgres | mssql | sqlite
  host: "localhost"
```

```
  port: 3306
  database: "mi_base_de_datos"
  user: "usuario"
  password: "contraseña"
  ssl: false

logging:
  level: "INFO"          # DEBUG | INFO | WARNING | ERROR
  file: null             # null = solo consola; ruta de archivo para guardar
logs
```

3. Variables de entorno (alternativa al archivo YAML)

Todos los valores del archivo de configuración pueden sobreescribirse con variables de entorno con el prefijo RBNX_:

Variable	Equivalente en YAML
RBNX_SERVER_URL	server.url
RBNX_SERVER_TOKEN	server.token
RBNX_DATABASE_TYPE	database.type
RBNX_DATABASE_HOST	database.host
RBNX_DATABASE_PORT	database.port
RBNX_DATABASE_NAME	database.database
RBNX_DATABASE_USER	database.user
RBNX_DATABASE_PASSWORD	database.password
RBNX_DATABASE_SSL	database.ssl
RBNX_LOGGING_LEVEL	logging.level
RBNX_LOGGING_FILE	logging.file

Ejemplo:

```
export RBNX_SERVER_TOKEN="gwt_xxx"
export RBNX_DATABASE_PASSWORD="mi_password_seguro"
./rbnx-gateway --config config.yaml
```

Ejecución

Iniciar el gateway

./rbnx-gateway --config /ruta/a/config.yaml

Al iniciar correctamente verás en los logs:

```
2026-03-27 10:00:00 [INFO] rbnx-gateway: Starting Rocketbot Nexus gateway —
db=mysql server=wss://...
2026-03-27 10:00:00 [INFO] rbnx-gateway.mysql: MySQL connected to
```

```
localhost/mi_base
2026-03-27 10:00:01 [INFO] rbnx-gateway.ws: Connecting to wss://... (attempt
1)…
2026-03-27 10:00:01 [INFO] rbnx-gateway.ws: Registered as gateway 550e8400-
e29b-41d4-a716-446655440000
2026-03-27 10:00:01 [INFO] rbnx-gateway.ws: Gateway ready — waiting for
queries
```

El punto verde en el panel de Nexus confirmará que el gateway está conectado.

Detener el gateway

Presiona Ctrl+C o envía SIGTERM al proceso. El gateway cerrará la conexión con la base de datos de forma limpia.

Opciones de línea de comandos

```
rbnx-gateway [opciones]

  --config, -c PATH    Ruta al archivo YAML de configuración
  --log-level LEVEL    Sobreescribe el nivel de log (DEBUG, INFO, WARNING,
ERROR)
  --version, -v        Muestra la versión y sale
  --help, -h           Muestra esta ayuda
```

Configuración por tipo de base de datos

MySQL

```
database:
  type: "mysql"
  host: "192.168.1.50"
  port: 3306              # opcional, defecto: 3306
  database: "ventas"
  user: "nexus_user"
  password: "secret"
  ssl: false
```

Permisos mínimos recomendados:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ventas.* TO 'nexus_user'@'%';
```

PostgreSQL

```
database:
  type: "postgres"
  host: "db.internal"
  port: 5432              # opcional, defecto: 5432
```

```
  database: "erp"
  user: "nexus_user"
  password: "secret"
  ssl: false              # true para forzar SSL
```

Microsoft SQL Server

> **Requisito previo:** instalar el driver ODBC de Microsoft.
>
> - **Windows:** ODBC Driver 17 for SQL Server
> - **macOS:** brew install msodbcsql17
> - **Linux (Ubuntu):** apt-get install msodbcsql17

```
database:
  type: "mssql"
  host: "sqlserver.corp"
  port: 1433
  database: "Inventario"
  user: "sa"
  password: "secret"
  ssl: false
  driver: "ODBC Driver 17 for SQL Server"   # ajusta si tienes otra versión
```

El gateway detecta automáticamente los drivers instalados e indica si falta alguno.

SQLite

```
database:
  type: "sqlite"
  path: "/datos/app.db"   # ruta absoluta al archivo .db
```

No requiere host, user ni password.

Ejecutar como servicio del sistema

Linux (systemd)

Crea el archivo /etc/systemd/system/rbnx-gateway.service:

```
[Unit]
Description=Rocketbot Nexus On-Premises Gateway
```

```
After=network.target

[Service]
Type=simple
User=nexus
ExecStart=/opt/rbnx-gateway/rbnx-gateway --config /opt/rbnx-
gateway/config.yaml
Restart=always
RestartSec=10
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target


sudo systemctl daemon-reload
sudo systemctl enable rbnx-gateway
sudo systemctl start rbnx-gateway
sudo systemctl status rbnx-gateway
```

Windows (Tarea programada)

```
schtasks /Create /SC ONSTART /TN "Nexus Gateway" /TR "C:\nexus\rbnx-
gateway.exe --config C:\nexus\config.yaml" /RU SYSTEM /F


O instala como servicio con NSSM:


nssm install NexusGateway "C:\nexus\rbnx-gateway.exe" "--config
C:\nexus\config.yaml"
nssm start NexusGateway
```

macOS (launchd)

```
Crea ~/Library/LaunchAgents/com.rbnx.gateway.plist:

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.rbnx.gateway</string>
  <key>ProgramArguments</key>
  <array>
    <string>/opt/rbnx-gateway/rbnx-gateway</string>
    <string>--config</string>
    <string>/opt/rbnx-gateway/config.yaml</string>
  </array>
  <key>RunAtLoad</key>
```

```
  <true/>
  <key>KeepAlive</key>
  <true/>
</dict>
</plist>
```

```
launchctl load ~/Library/LaunchAgents/com.rbnx.gateway.plist
```

Integrar un DataSource en Nexus

1. En Nexus, ve a **Data Sources → Nuevo DataSource**.
2. Selecciona el tipo **Gateway On-Premises**.
3. En el campo **Gateway Token**, elige el token correspondiente.
    - El indicador muestra **verde** si el gateway está conectado, **gris** si no.
4. Haz clic en **Probar conexión** para verificar.
5. Guarda el DataSource y úsalo normalmente en tus aplicaciones.

Reconexión automática

El gateway se reconecta automáticamente si la conexión se interrumpe. El intervalo de espera aumenta progresivamente (backoff exponencial) hasta un máximo de 60 segundos:

```
Intento 1: 5 s
Intento 2: 10 s
Intento 3: 20 s
...
Máximo:   60 s
```

Para deshabilitar el límite de intentos: max_reconnect_attempts: 0 (valor por defecto).

Solución de problemas

El gateway muestra “Registration failed: Invalid token”

- Verifica que el token en config.yaml sea correcto y no tenga espacios.
- Comprueba que el token **no esté revocado** en el panel de Nexus.

- Los tokens son sensibles a mayúsculas/minúsculas.

El gateway se conecta pero las consultas fallan

- Activa el modo debug: --log-level DEBUG para ver las consultas SQL generadas.
- Verifica que el usuario de base de datos tenga permisos sobre las tablas consultadas.
- Comprueba que el nombre de la base de datos en la configuración sea exacto.

Error: “No SQL Server ODBC driver found” (MSSQL)

Instala el driver correspondiente a tu sistema operativo (ver sección MSSQL). Luego verifica con:

```
python3 -c "import pyodbc; print(pyodbc.drivers())"
```

El indicador en Nexus nunca se pone verde

- Verifica que el servidor pueda alcanzar la URL de Nexus: curl -I https://tu-nexus.com
- Comprueba que el puerto 443 saliente no esté bloqueado por el firewall.
- Revisa que la URL en server.url use wss:// (no ws://).

Diagnóstico de conexión lenta, desconexiones o tablas sin datos

El gateway registra y reporta métricas para saber **dónde** está el problema:

En el log del gateway (nivel INFO):

- Una línea por consulta: select clientes ok rows=50 size=1.2MB total=420ms (queue=0 db=80 serialize=35)
  - db: tiempo ejecutando la consulta en tu base de datos.
  - queue: tiempo esperando la conexión a la base (otra consulta o el vigilante de tablas la estaban usando).
  - serialize: tiempo preparando el resultado (JSON/base64).
  - Si total es mucho mayor que queue + db + serialize, la diferencia es red o espera en el servidor.

- Cada 30 s una línea health rtt=… ping_loss=… loop_lag_max=…:
  - rtt / ping_loss: latencia y pérdida de la conexión hacia Nexus (medidas con un ping propio).

    - loop_lag_max: si es alto (> 1000 ms) el propio gateway se quedó bloqueado (equipo saturado, consulta muy pesada).

- Al desconectarse: Socket.IO disconnected — reason=<motivo> (por ejemplo ping timeout, transport error, server disconnect).
- Avisos Fingerprint of <tabla> took Ns si el vigilante de cambios tarda demasiado en una tabla (súbele watch_interval_seconds).

En Nexus (Application → Gateway Tokens): punto verde/ámbar, RTT, pérdida de ping, tiempos promedio de consulta, desconexiones de las últimas 24 h con su motivo, y el botón **Speed test** (latencia + Mbps de subida/bajada).

Respuesta demasiado grande: si un resultado supera database.max_response_bytes (18 MB por defecto) la consulta falla con Result too large en vez de provocar una desconexión. Reduce el límite de filas, selecciona menos columnas o evita columnas BLOB. Debe mantenerse por debajo de SOCKET_MAX_HTTP_BUFFER_SIZE del backend (20 MB por defecto).

Opciones nuevas (todas opcionales):

```
server:
  ping_interval_seconds: 15      # 0 = desactivar el ping de medición
  metrics_interval_seconds: 30   # 0 = no enviar métricas a Nexus
database:
  max_response_bytes: 18874368
```

Variables de entorno equivalentes: RBNX_SERVER_PING_INTERVAL, RBNX_SERVER_METRICS_INTERVAL, RBNX_DATABASE_MAX_RESPONSE_BYTES.

Logs verbosos para diagnóstico

```
./rbnx-gateway --config config.yaml --log-level DEBUG 2>&1 | tee gateway-debug.log
```

Seguridad

- **El token es equivalente a una contraseña.** Trátalo con el mismo cuidado.
- Usa variables de entorno (RBNX_SERVER_TOKEN, RBNX_DATABASE_PASSWORD) en lugar de escribir secretos en el archivo YAML en entornos de producción.
- Crea un usuario de base de datos dedicado con **mínimos privilegios** (solo SELECT si tus aplicaciones son de solo lectura).
- Revoca tokens inmediatamente desde el panel de Nexus si un servidor se compromete.

- El gateway no abre ningún puerto entrante — solo hace conexiones salientes al puerto 443.

Referencia rápida

```
# Iniciar
./rbnx-gateway --config config.yaml

# Iniciar con debug
./rbnx-gateway --config config.yaml --log-level DEBUG

# Solo con variables de entorno (sin archivo)
RBNX_SERVER_TOKEN=gwt_xxx RBNX_DATABASE_HOST=localhost \
RBNX_DATABASE_NAME=mydb RBNX_DATABASE_USER=root \
RBNX_DATABASE_PASSWORD=secret ./rbnx-gateway

# Ver versión
./rbnx-gateway --version
```