



ON-PREMISE PREPARATION

Faction Four Systems Inc.

Applies to Smart Squad 2.9 · on-premise deployments · Revised October 2026

What to put in place before Smart Squad is installed in your own environment — server roles, the service account, SQL Server, DNS and firewall. Work through it with your Windows, database and network teams; everything here is their work, not ours, and it needs to be done before an install date is useful.

CONTENTS

System requirements	2
IIS configuration	2
Service account	3
SQL Server preparation	4
DNS configuration	6
Firewall configuration	7
Verification and troubleshooting	9
High availability	7
Questions	10



SYSTEM REQUIREMENTS

COMPONENT	MINIMUM
Operating system	Windows Server 2019, 2022 or 2025
CPU	4 or more cores at 2.4 GHz or higher
Memory	8 GB RAM minimum; 32 GB recommended
Disk	100 GB available; SSD recommended
Database	Microsoft SQL Server 2019 or newer, Standard or Enterprise
Web server	IIS 10.0 or newer
Framework	.NET Framework 4.8 or newer, and .NET Core 7.0 or newer

IIS CONFIGURATION

Install the IIS components

In Server Manager, choose **Add Roles and Features** , **Role-based or feature-based installation** , select the server, then **Web Server (IIS)** and add the features when prompted. Ensure these components are selected:

- **Common HTTP features** — default document, directory browsing, HTTP errors, static content, HTTP redirection
- **Health and diagnostics** — HTTP logging, request monitor
- **Performance** — static and dynamic content compression
- **Security** — request filtering, basic authentication
- **Application development** — .NET Extensibility 4.8, ASP.NET 4.8, **WebSocket Protocol**

.NET Framework

Install .NET Framework 4.8 (or the latest 4.x) from Microsoft, running the installer with administrative privileges. Restart if prompted.

WebSockets

Confirm **WebSocket Protocol** is installed under **Web Server (IIS) > Web Server > Application Development** . Then in IIS Manager, select the server, open **WebSocket Protocol** and confirm **Enable WebSocket Protocol** is checked.

Do not skip this. Smart Squad holds long-lived WebSocket connections for messaging, alerts and live unit locations. Without the protocol enabled the application installs and signs in normally, and then real-time features simply never work.

WCF

Under **Features > .NET Framework 4.8 Features > WCF Services**, install **HTTP Activation, TCP Activation** and **Named Pipe Activation**.

Then in `services.msc`, set each of these to start **Automatic**:

- Net.Tcp Listener Adapter
- Net.Pipe Listener Adapter
- Net.Http Listener Adapter
- Windows Process Activation Service

Application pool

1 Create a dedicated pool

In IIS Manager, under **Application Pools**, choose **Add Application Pool**. Name it `SmartSquadAppPool`, set the .NET CLR version to **.NET CLR v4.0.30319** and leave the managed pipeline mode on **Integrated**.

Smart Squad requires .NET Framework 4.7, but the pool is set to .NET 4 — that setting covers every 4.x version.

2 Set the pool identity

Advanced Settings > Process Model > Identity, choose **Custom account** and enter the service account (`DOMAIN\svc_smartsquad`) and its password.

3 Adjust the pool settings

- Enable 32-bit applications: **False**
- Idle time-out (minutes): **0** — disables idle-based recycling
- Regular time interval (minutes): **1740** — recycles every 29 hours
- Private memory limit: adjust to the server's memory; the default is usually fine

4 Assign the applications to it

For each Smart Squad site, **Manage Application > Advanced Settings** and set the application pool to `SmartSquadAppPool`.

SSL certificate

Obtain a **wildcard certificate** for `*.smartsquad.AGENCY.ca` from a trusted CA, covering every Smart Squad DNS name across all environments.

Import it in IIS Manager under **Server Certificates > Import**, then add or edit the HTTPS binding on each Smart Squad site to use it.

SERVICE ACCOUNT

Smart Squad needs a dedicated domain service account.

1 Create the account in Active Directory

In the OU where service accounts live, create a user — first name *Smart Squad*, last name *Service*, logon name `svc_smartsquad` by convention — with a password meeting your policy. Consider setting **User cannot change password** and **Password never expires**.

2 Grant "log on as a service"

In `secpol.msc`, under **Local Policies > User Rights Assignment**, add the account to **Log on as a service**.

Across several servers, do this through Group Policy instead: a GPO linked to the OU holding the Smart Squad servers, at **Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.

3 Add it to IIS_IUSRS

In `compmgmt.msc`, under **Local Users and Groups > Groups**, add the account to **IIS_IUSRS**.

4 Windows Server 2025 only — set the service identities

In `services.msc`, for each Smart Squad service, open **Properties > Log On**, choose **This account**, enter the service account and password, then restart the service.

SQL SERVER PREPARATION

Requirements

- SQL Server 2019 or newer, Standard or Enterprise
- Authentication mode set to **SQL Server and Windows Authentication**
- SQL Server Agent running
- Full-text search components installed

- FILESTREAM enabled

Create the database container

In SSMS, create a new database named `SmartSquad` :

- Data file — initial size 1 GB, autogrowth 512 MB
- Log file — initial size 512 MB, autogrowth 256 MB

Create the logins

LOGIN	TYPE	NOTES
<code>DOMAIN\svc_smartsquad</code>	Windows	Search for the account created above.
<code>SmartSquadProcessor</code>	SQL	Complex password per your policy. Uncheck Enforce password policy for a service account.
<code>SmartSquadAdmin</code>	SQL	Server roles <code>public</code> and <code>securityadmin</code> , for user management within the Smart Squad database.

Stop at creating the logins. Server roles beyond those noted, user mapping and database permissions are all handled by the Smart Squad installation scripts. Setting them by hand beforehand tends to produce conflicts the installer then has to unpick.

Configure FILESTREAM

1 Enable it at instance level

SSMS › instance **Properties › Advanced** › set **FILESTREAM Access Level** to **Full access enabled**.

2 Add the filegroup and container

On the `SmartSquad` database, **Properties › Filegroups** › **Add Filegroup** named `SmartSquad_FS` with the FILESTREAM option checked. Then **Add File** with logical name `SmartSquad_FS_Data` on a dedicated NTFS volume, for example `E:\MSSQL\FILESTREAM\SmartSquad`.

3 Grant permissions on the directory

The SQL Server service account needs Full Control on the FILESTREAM directory, and the Smart Squad service account needs appropriate access to it.

Give the FILESTREAM volume its own monitoring. Attachment binaries are stored on the filesystem through FILESTREAM, so when that volume fills, attachment upload breaks while the rest of the application continues to look perfectly healthy.

Configure full-text search

1. Confirm it is installed: right-click the server node, **Facets**, the **Surface Area Configuration** facet, and check **FullTextInstalled** is **True**. If not, add the component through SQL Server setup.
2. On the database, under **Storage > Full Text Catalogs**, create a new catalog named **SmartSquadFTC** and set it as the default.
3. In **services.msc**, confirm **SQL Server Full-text Filter Daemon Launcher** is set to Automatic and running.

Maintenance plan

- **Backups** — full daily, differential every 6 hours, transaction log every 30 minutes. Adjust to your own recovery objectives.
- **Index maintenance** — a weekly plan for reorganise and rebuild, plus statistics updates.
- **Integrity** — weekly **DBCC CHECKDB**, with SQL Server Agent alerts configured for integrity issues.

DNS CONFIGURATION

Smart Squad needs separate DNS entries for the service and the web client in each environment.

The names below are examples only — use whatever convention your organisation prefers, as long as each environment and component has its own entry. Replace **AGENCY** with your own domain.

ENVIRONMENT	COMPONENT	EXAMPLE RECORD
Development / test	Service	<code>test.smartsquad.AGENCY.ca</code>
Development / test	Web client	<code>client-test.smartsquad.AGENCY.ca</code>
UAT	Service	<code>uat.smartsquad.AGENCY.ca</code>
UAT	Web client	<code>client-uat.smartsquad.AGENCY.ca</code>
Production	Service	<code>smartsquad.AGENCY.ca</code>
Production	Web client	<code>client.smartsquad.AGENCY.ca</code> , or the same entry as the service

Each is an A or CNAME record pointing at the relevant server, or at a load balancer or proxy in a high-availability setup. If you use CNAMEs, confirm the target hostnames resolve correctly.

Verify and consider

- **Verify propagation** — run `nslookup` against each record from a client machine and confirm it resolves to the right address.
- **The wildcard certificate** (`*.smartsquad.AGENCY.ca`) must cover every name here and be bound in IIS for all Smart Squad sites.
- **TTL** — keep it short during initial deployment so corrections take effect quickly, and extend it once the system is stable.

FIREWALL CONFIGURATION

Windows firewall

On the Smart Squad server, create inbound rules for HTTP (TCP 80) and HTTPS (TCP 443), plus any additional ports your implementation requires.

Outbound access from the Smart Squad server

PURPOSE	ENDPOINTS
Map services	<code>*.mapbox.com</code> , <code>*.hereapi.com</code>
Communication services	<code>*.pubnub.com</code> , <code>*.pndsn.com</code> , <code>*.pubnub.net</code> , <code>*.pubnub.io</code> , <code>*.twilio.com</code> , <code>*.api.flagsmith.com</code>
Virtual interview and calling	<code>*.dyte.io</code> on 2.9.0 and earlier; Cloudflare RealtimeKit domains from 2.9.1
Authentication services	<code>api.yubico.com</code> , <code>*.evidence.com</code> , <code>graph.microsoft.com</code> , <code>login.microsoftonline.com</code>
Support and monitoring	<code>*.freshdesk.com</code> , <code>*.rollbar.com</code>
Cloud storage and services	<code>blob.core.windows.net</code> , <code>file.core.windows.net</code> , <code>cognitive.microsoft.com</code> , <code>*.sendgrid.net</code> , <code>*.azurewebsites.net</code> , <code>*.tts.speech.microsoft.com</code> , <code>*.smartsquad.online</code>
Weather services	<code>*.openweathermap.org</code>
AI services	<code>*.openai.azure.com</code> for all AI functionality; <code>api.anthropic.com</code> for the Smart Glasses feature
Internal systems	The Smart Squad database server (SQL Server, typically TCP 1433), your RMS, your CAD, and any other internal system that integrates with Smart Squad

A block on any one of these produces a single-feature failure while the rest of the application works normally — a blocked Mapbox gives a blank map, a blocked PubNub gives no real-time messages, blocked AI endpoints stall drafting. When one feature misbehaves after a firewall change, match the feature to its endpoint before anything else.

Inbound access to the Smart Squad server

- **PubNub** — `*.pubnub.com` , `*.pndsn.com` , `*.pubnub.net` , `*.pubnub.io`
- **Client networks** that need to reach the web interface over HTTP and HTTPS

DMZ, if applicable

- Configure appropriate rules between the DMZ and the internal network
- Ensure SQL Server communication is properly secured
- Consider a web application firewall for additional protection

VERIFICATION AND TROUBLESHOOTING

Once the preparation steps are complete:

1. Restart the Smart Squad application pool in IIS Manager
2. Restart any associated Windows services
3. Open the Smart Squad web application in a browser
4. Check the application event logs for authentication or permission errors
5. Confirm Smart Squad can reach the organisational resources it needs

Common issues

SYMPTOM	WHAT TO CHECK
Access denied errors	The service account's permissions on network resources; file system permissions on the Smart Squad application directories; the account's database access.
Application pool will not start	The service account password is correct and not expired; "log on as a service" is assigned; the event log for a specific error.
Windows services fail to start	The service account's permissions; the password stored in the service configuration matches the actual password; no Group Policy is restricting the account.

HIGH AVAILABILITY

For mission-critical deployments, Smart Squad supports high-availability configurations.

Database

- **Always On availability groups** — database-level HA with automatic failover
- **Failover cluster instances** — instance-level HA with shared storage
- **Database mirroring** — older, still supported, database-level
- **Log shipping** — useful for disaster recovery and for minimising downtime during planned maintenance

Choose based on your recovery time and recovery point objectives, your existing SQL Server infrastructure, licensing, and your team's experience.

Web tier

- **Network Load Balancing** across multiple Smart Squad web servers
- **Application Request Routing** for more complex routing
- **Third-party load balancers** that support session affinity

In any load-balanced configuration: keep the servers' configuration identical, configure session persistence, implement shared content storage where file uploads are used, and synchronise the IIS configuration across servers.

Also consider

- **Shared storage** — a resilient solution for uploads and shared content
- **Certificate management** — certificates deployed consistently across all servers
- **Monitoring** — comprehensive enough to detect failures before users report them
- **Backups** — taken regularly and *tested for recoverability*

The right approach varies with your infrastructure, policies and requirements — work it through with your infrastructure team.

QUESTIONS

Anything in this guide, or the specifics of your environment — raise it with Faction Four support before your install date rather than on it.