VisibleThread

VT Docs On-Prem Deployment

March 2024

VisibleThread Deployment Overview
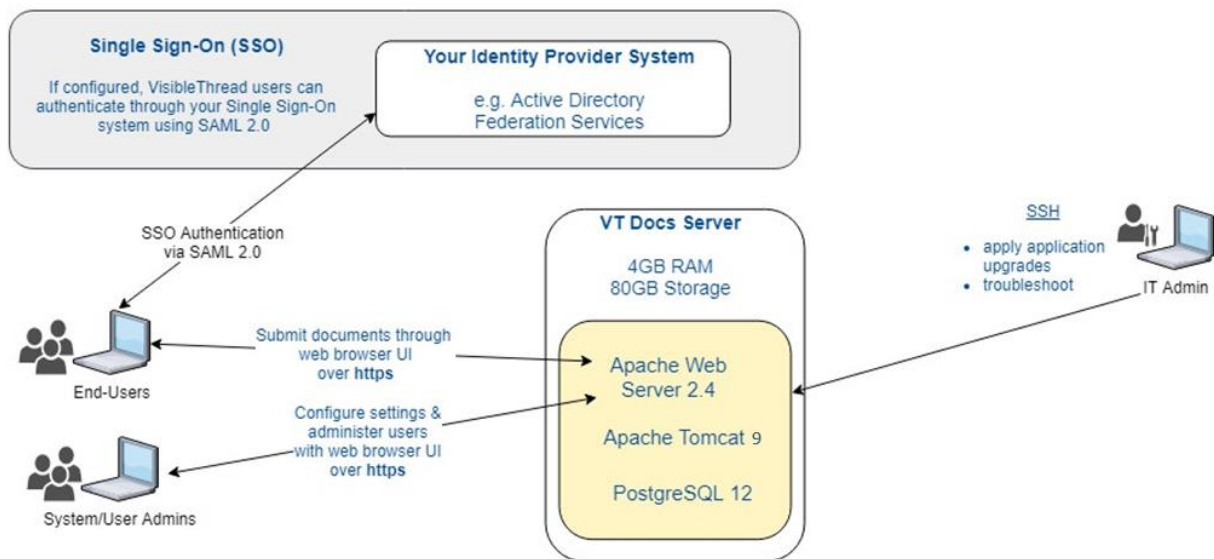
DEPLOYMENT OVERVIEW

A VT Docs deployment involves a single server:

- **VT Docs Server** hosts a web browser-based application.

The diagram below illustrates the server deployed on-prem (inside your organization) including user and system interactions.

VT Docs deployment architecture
This diagram describes the VT Docs Server and end-user/admin interaction & usage.

We provide multiple ways to deploy VT Docs software on-prem:

- A **Red Hat package** (a visiblethread-xxx-.rpm file) that you install into your Red Hat Operating System.
- A pre-packaged **Virtual Appliance** (a Virtual Machine (VM) / .ova file) that you import into your virtualisation infrastructure. The VM contains a pre-installed Ubuntu Operating System plus VisibleThread components.
- **Windows** executable file that you install on your Windows OS.

An on-prem VisibleThread deployment has the following properties:

- Runs inside your organization's network/firewall & is fully under your control. No external access required.
- OS security/patching fully under your control.
- Initial install and application upgrades done by your IT team.
- Low IT management overhead – application upgrades are made available twice a year; your IT team can install the upgrades directly.
- Email/phone/screen-sharing support is available from VisibleThread staff.
- You can optionally setup a test/staging server to try out upgrades before rolling out to production.
- All end-user access is secured over https using a web browser.
- There is an option to encrypt data at rest.
- With a Virtual Appliance deployment, you can use virtualisation backup/restore.

MINIMUM REQUIREMENTS

Below are the minimum requirements for VT Docs:

Red hat Package Install AWS/AZURE	**Virtual Appliance Install**	**Windows Install**
Red Hat Enterprise Linux RHEL 8 Red Hat Enterprise Linux RHEL 9 Amazon Linux 2	VMWare vSphere VMWare vCenter	Windows 10 Windows 11 Windows Server 2016 Windows Server 2019 Windows Server 2022
Minimum Spec	**Minimum Spec**	**Minimum Spec**
4GB RAM 2 vCPU Cores 80GB Disk	4GB RAM 2 vCPU Cores 80GB Disk	4GB RAM allocated for VT Docs above the recommended amount for your windows install 2 vCPU Cores 80GB Disk

INSTALLED COMPONENTS

The following table lists the components/dependencies installed with the application:

VT Docs
Apache Tomcat 9
Apache web server 2.4 (optional for windows Installs)
IIS (Optional for Windows ONLY)
PostgreSQL 12 +
PostgreSQL 14 (Windows ONLY)

NETWORKING

VT Docs is a web application. End-users and system admins access the application via their web browser over https.

IT administrators access the VT Docs server console via ssh on port 22.

All communication with VT Docs uses these ports:

- Port 443 is used for https communication via the web browser.
- Port 22 is used for ssh and secure ftp access.

VT Docs does not require access to any external location.

A self-signed SSL certificate is supplied pre-configured with VT Docs. This allows the application to run "out of the box". You should install your own SSL certificate during deployment.

With a Virtual Appliance deployment, the VT Docs Server is configured to use DHCP to pick up an IP address automatically on startup. It is recommended you allocate a designated static IP address to the server on initial deployment.

USER MANAGEMENT

VT Docs provides a web application to manage users and application licenses. User accounts are created and managed locally on your VT Docs installation using the web browser-based application.

Each user must be created with a username (email address) and a password. Passwords are salted, securely hashed and stored locally in the VT Docs database.

Users can be given 'Admin' privileges which allows them add/remove users from their sandbox. This provides the flexibility to give certain user control over their workspace, but limit access to wider administration functions.

Single Sign-On (SSO) with SAML 2.0 is now available with VT Docs version 3.2 and above. See our configuration help guide for more details.

AUDITING

We support review & audit of user activity including user login, add/remove documents, generate reports. All user activity is tracked in our application database and is available for review.

BUSINESS CONTINUITY & FAILOVER

VT Docs is deployed as a single instance server. All user and configuration data is stored in the PostgreSQL database. It is good practice to maintain a Disaster Recovery environment to allow for business continuity in the event of any issues occurring in the main VT Docs server.

There is no single prescribed 'best approach' to implementing disaster recovery for VT Docs. In all cases this is something your IT and Infrastructure teams will be best placed to solve. Our support team can provide any assistance required to ensure any solution is compatible with the operation of the application.

Disaster Recovery Key Requirements

The key requirements for a DR solution are:

- A second standby VT Docs application server
- Redundant storage for the VT Docs database, which can be connected to the standby VT Docs server

There are many ways to achieve this depending on how you have chosen to deploy VT Docs. For example, if you have deployed in AWS, you can use the high availability features in RDS to achieve this. Alternatively, you may choose to deploy the VT Docs database on your own network on dedicated infrastructure and use storage

replication to provide redundancy for the VT Docs database.

A simple approach to a Disaster Recovery solution is outlined below:
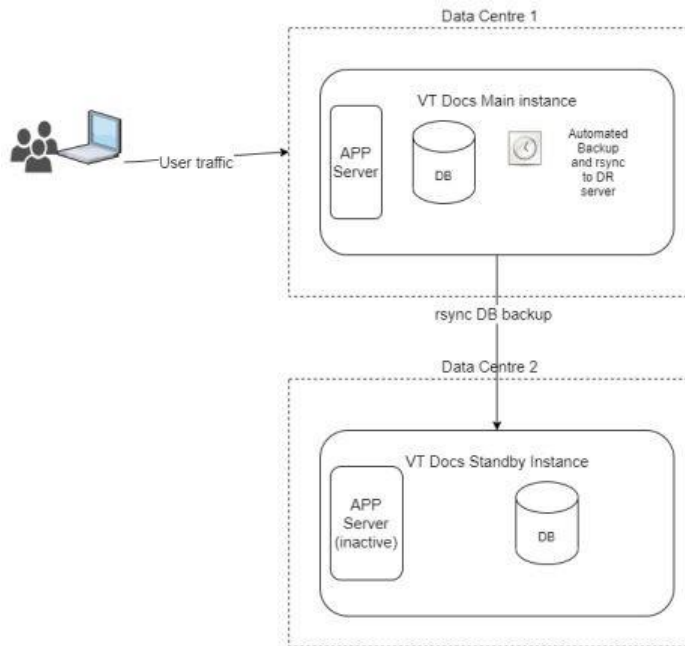
The key points of this solution are:

- An automated job runs nightly on the main VT Docs instance. This job takes a backup of the PostgreSQL database and then transfers the backup to the standby instance using rsync.
- On the standby instance, a VT Docs app server is installed but is inactive.

When we need to failover to the standby instance we must:

- Restore the latest backup on the Standby instance.
- Activate the VT Docs server on the Standby instance.
- Route user traffic to the standby instance. This can be accomplished using either DNS failover, a load balancer or other routing mechanisms.