



SFTP Gateway for the Enterprise

White Paper

Version 1.0
September 1, 2024

Prepared By:

Thorn Technologies
9175 Guilford Rd, Suite 212
Columbia, MD 21046
info@thorntech.com
www.thorntech.com

Table of Contents

Introduction.....	2
SFTP Gateway Overview.....	3
Features.....	3
Use Cases.....	4
SFTP Gateway Design & Architecture.....	6
SFTP Gateway Operation.....	6
SFTP Gateway Architecture.....	8
SFTP Gateway High Availability.....	14
HA Architecture.....	14
Cloud Native Services.....	15
VPC.....	15
Network Load Balancer.....	17
AutoScaling Group.....	18
EC2 instances.....	20
PostgreSQL on RDS.....	21
S3.....	22
CloudWatch & Enterprise Observability Tools.....	22
Managed Hosting of SFTP Gateway.....	24
Additional Information.....	25
About Thorn Technologies.....	26

Introduction

The SFTP protocol has long been the standard for file transfer. This remains true today, in the public cloud era of serverless computing. Why is that? When you transfer files with other organizations, you have no control over their tech stacks or security posture. Organizations must fall back on SFTP as the least common denominator compatible with and acceptable to everyone.

Since SFTP is here to stay, many organizations face the challenge of integrating SFTP with the cloud. How do you get files from local servers into cloud storage services like Amazon S3? How do you deploy SFTP servers in a highly available (HA) configuration and keep data synchronized? The SFTP protocol was designed before these modern cloud concepts existed.

When two organizations share data, it is often a recurring, automated process. The nature of the SFTP protocol makes it easy to schedule jobs that run weekly, daily, or hourly to transfer data securely over the internet from one SFTP server to another SFTP server. But what happens when one of these organizations moves their applications to the cloud? Transferring data from one SFTP server to another organization's cloud storage location is not easy to do. Furthermore, modifying every scheduled job or automated business process to transfer data to cloud storage instead of via SFTP is not a viable option.

SFTP Gateway® is a product developed by Thorn Technologies to solve these problems. It is an SFTP server sold as a Virtual Machine (VM) image that stores data in Cloud Storage services like Amazon S3, Azure Blob, and Google Cloud Storage. SFTP users can read and write files to these services in real-time using an SFTP client like FileZilla or OpenSSH.

SFTP Gateway can be deployed in an HA configuration. SFTP Gateway VM instances are ephemeral and do not maintain state on local volumes, allowing VM instances to scale up or scale down in response to load or demand. User configuration and files are consistent regardless of which server instance a user is connected to.

This document provides a comprehensive overview of SFTP Gateway, a powerful tool that enables seamless integration between SFTP (Secure File Transfer Protocol) and cloud storage platforms such as Amazon S3 (Simple Storage Service), Microsoft Azure Blob Storage, and Google Cloud Storage (GCS). It provides an overview of the product architecture, features, and use cases. This document concludes with an in-depth discussion of configuring SFTP Gateway for High Availability (HA) for reliable deployments across the enterprise.

SFTP Gateway Overview

SFTP Gateway is a lightweight, secure, and affordable alternative to legacy Managed File Transfer (MFT) solutions. It was designed to allow organizations to maintain existing automated data transfer processes while leveraging the durability, reliability, and capacity of cloud storage services.

SFTP Gateway is secure because it is deployed to your private cloud environment, allowing you to maintain complete control of your data and keep it secure within your organization. Unlike similar SaaS solutions that have experienced [costly data breaches](#), SFTP Gateway does not expose your data to a third-party processor. Thorn Technologies will work with you to deploy SFTP Gateway in your organization in a manner that complies with your corporate security and data governance requirements.

SFTP Gateway's configuration and management are easier than those of traditional OpenSSH SFTP servers or legacy MFT applications. Administrators access an intuitive, web-based admin dashboard. Deployments are automated using CloudFormation, ARM, GDP, or Terraform templates.



Features

Features of SFTP Gateway include:

- Secure SFTP file transfers to cloud storage (AWS S3, Azure Blob, Google Cloud)
- Simple web-based admin dashboard
- Support for third-party identity services (OIDC, Active Directory/LDAP)
- SSH key-based authentication
- Password-based authentication with configurable complexity
- Brute force attack protection
- Simplified user and key management tools
- Multi-cloud and cross-cloud compatibility
- Load balancing for High Availability (HA)
- Logging and monitoring for compliance and auditing
- REST API for automating and integrating with business workflows
- User-based IP address filtering
- Simplified configuration of encryption algorithms and ciphers
- Automated deployment via cloud templates
- Simple backup and recovery options

Sign In

Username

Password

SIGN IN



Encryption Algorithms

Supported Encryption Algorithms. To customize encryption algorithm options, switch to Editing mode. These algorithms will be enabled for future SFTP sessions.

[Viewing ▾](#) [Reset to Default](#)

HMACs

PARANOID

- ☒ hmac-sha2-512
- ☒ hmac-sha2-512-96
- ☒ hmac-sha2-512-etm@openssh.com

STRONG

- ☒ hmac-sha2-256
- ☒ hmac-sha2-256-96
- ☒ hmac-sha2-256-etm@openssh.com

WEAK

- ☐ hmac-sha1
- ☐ hmac-sha1-96
- ☐ hmac-sha1-etm@openssh.com

Public Keys

PARANOID

- ☒ ssh-ed25519
- ☒ ssh-ed25519-cert-v01@openssh.com
- ☒ ssh-ed448

STRONG

- ☒ ecdsa-sha2-nistp256
- ☒ ecdsa-sha2-nistp256-cert-v01@openssh.com
- ☒ ecdsa-sha2-nistp384
- ☒ ecdsa-sha2-nistp384-cert-v01@openssh.com
- ☒ ecdsa-sha2-nistp521
- ☒ ecdsa-sha2-nistp521-cert-v01@openssh.com

- ☒ rsa-sha2-256
- ☒ rsa-sha2-512

WEAK

- ☐ ssh-dss
- ☐ ssh-rsa
- ☐ ssh-rsa-cert-v01@openssh.com

SSH2 Ciphers

PARANOID

- ☒ aes128-gcm@openssh.com
- ☒ aes256-gcm@openssh.com
- ☒ chacha20-poly1305@openssh.com

STRONG

- ☒ aes128-ctr
- ☒ aes192-ctr
- ☒ aes256-ctr

WEAK

- ☐ 3des-ctr

Key Exchanges

PARANOID

- ☒ curve25519-sha256
- ☒ curve25519-sha256@libssh.org
- ☒ diffie-hellman-group15-sha512

- ☒ diffie-hellman-group16-sha512
- ☒ diffie-hellman-group17-sha512
- ☒ diffie-hellman-group18-sha512

STRONG

- ☒ diffie-hellman-group-exchange-sha256
- ☒ diffie-hellman-group14-sha256
- ☒ ecdh-sha2-nistp256
- ☒ ecdh-sha2-nistp384
- ☒ ecdh-sha2-nistp521
- ☒ rsa2048-sha256

WEAK

- ☐ diffie-hellman-group14-sha1

Use Cases

SFTP Gateway has been proven in several common use cases, including:

- **Cloud Migration:** Organizations looking to migrate data from on-premises storage systems to the cloud can use SFTP Gateway to simplify the process. By leveraging the SFTP protocol, organizations can seamlessly transfer files to and from cloud storage platforms without disrupting their existing workflows.
- **Remote Access to Cloud Storage:** Businesses that require remote access to their cloud storage resources can utilize SFTP Gateway to provide secure file transfer capabilities to authorized users. This allows employees and partners to access files stored in the cloud from any location, improving collaboration and productivity.
- **Data Backup and Recovery:** SFTP Gateway can serve as a reliable backup solution for on-premises or cloud-based data. By regularly transferring files to cloud storage, businesses can ensure that their data is protected in the event of a disaster or hardware failure.

- **Analytics, AI & Machine Learning:** SFTP Gateway is highly beneficial for Analytics, Business Intelligence (BI), and AI because it streamlines the secure transfer of data from multiple external sources into cloud storage (AWS, Azure, Google Cloud). By automating and managing these file transfers, it enables organizations to aggregate data in one place. Once the data is stored, it can be loaded into ETL (Extract, Transform, Load) processes, analyzed for business insights, or used to train large language models (LLMs) for AI applications.

Thousands of customers around the world in dozens of industries are using SFTP Gateway for their specific business needs, such as:

- Call-center audio and video file archiving
- Banks and financial institutions replacing expensive MFT solutions, like IBM Sterling File Gateway
- Healthcare companies transferring patient records in compliance with HIPAA regulations
- Global manufacturing companies complying with GDPR and Data Localization laws
- Law firms integrating case management and e-discovery tools
- IoT devices and Appliances that transmit sensor data via SFTP
- Integrating E-commerce ERP systems like SAP and Workday with Salesforce CRM
- Merger and Acquisition companies onboarding data from acquired organizations
- Customer support and ticketing systems
- First responder and EMS data for incident response analysis
- Digital marketing and advertising engagement data

SFTP Gateway Design & Architecture

SFTP Gateway can be deployed to AWS, Azure, or Google Cloud environments. It stores data in cloud storage services like AWS S3, Azure Blob Storage, and Google Cloud Storage (GCS). SFTP Gateway is “Cross-Cloud,” meaning it can run on one cloud but read and write data to one or more cloud storage locations on other clouds.

SFTP Gateway has abstracted the operation of writing to cloud storage, so from the user’s perspective, it is identical to using any standard SFTP service. The user has no knowledge that the data is backed by cloud storage.

From the cloud architect’s perspective, SFTP Gateway has abstracted the integration of the different cloud storage services. The product works exactly the same regardless of whether data is written to AWS, Azure, or GCP.

Unlike traditional SFTP servers, whose capacity is limited to the size of the physical hard drives, SFTP Gateway provides virtually limitless storage because data is written directly to cloud storage services.

It is important to note that all cloud services provide the same fundamental services - Computing and Durable Object Storage. For readability, we mainly refer to AWS and S3 for the remainder of this document. The concepts are identical regardless of your chosen cloud service provider.

SFTP Gateway Operation

With SFTP Gateway, an SFTP user account can be configured with key-based authentication or a complex password. SFTP users can then use these credentials to connect via an SFTP client like OpenSSH or FileZilla.

Once connected, the SFTP user can perform normal SFTP actions like uploading, downloading, listing, and deleting files. These file operations take place on S3, instead of a traditional Linux file system. SFTP Gateway makes this possible by translating the SFTP protocol to the AWS S3 API.

Each SFTP user is “chrooted” into a home directory, which means they cannot navigate anywhere beyond their home directory and its subfolders. By default, SFTP user home directories are configured as siblings, which prevents users from accessing each others’ data. This is a common pattern in SFTP to achieve data separation.

The default folder structure looks like this:

```
/ → Root of the file system maps to an S3 bucket
/users/
/users/sftpuser1/
/users/sftpuser2/
```

The root of the filesystem points to an S3 bucket. Subfolders point to folder paths within that S3 bucket. The default configuration of SFTP Gateway is for all of the SFTP users to inherit this root S3 bucket.

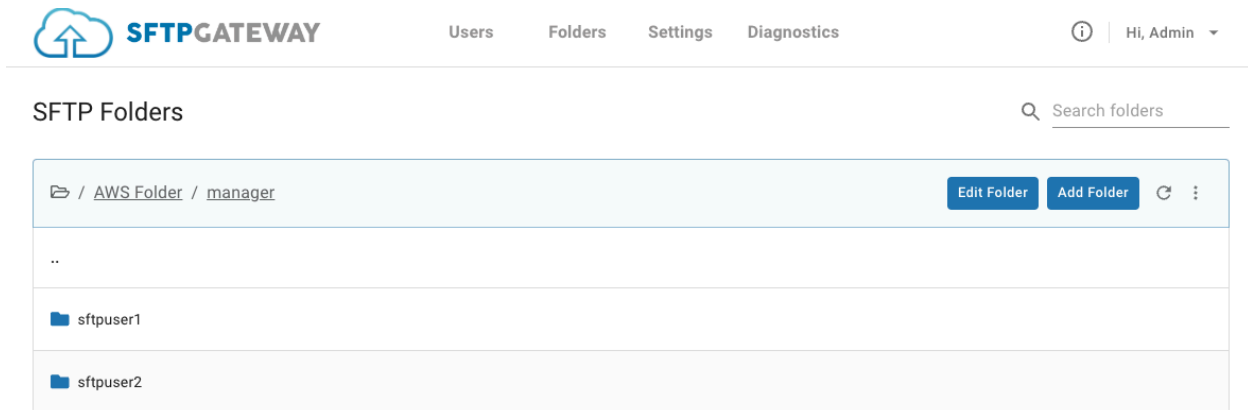
SFTP Gateway also lets you point any folder to any Cloud Storage bucket and path. So in the above example, you could give each SFTP user their own bucket:

```
/ → Root of the file system maps to a "default" S3 bucket
/users/
/users/sftpuser1/ → Maps to Bucket1 on AWS
/users/sftpuser2/ → Maps to Bucket2 on GCP
```

You can also point an SFTP user's home directory to any folder. This opens up the possibility of sharing scenarios, which is not an easy thing to do for traditional SFTP servers or MFT applications. In the example below, when users "sftpuser1" and "sftpuser2" connect with an SFTP client, they will be locked to their home directory and will not be able to access each other's files. However, when the SFTP user "manager" connects she will see two subdirectories in her root directory - "sftpuser1" and "sftpuser2" allowing her to access and share files with both users.

```
/
/users/ ← Home directory for "manager",
/users/sftpuser1/ ← Home directory for sftpuser1
/users/sftpuser2/ ← Home directory for sftpuser2
```

SFTP Gateway comes with a web admin portal for managing SFTP users and settings. You can perform all the actions mentioned above, including creating SFTP users, assigning credentials, and mapping home directories to cloud storage locations.



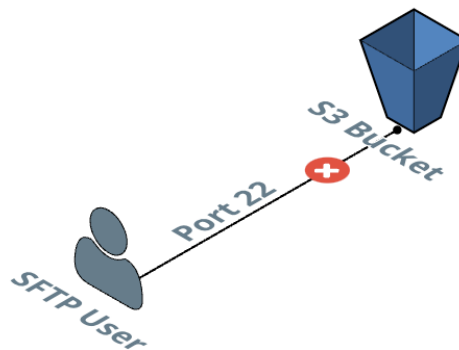
SFTP Gateway Architecture

In cloud computing, object storage (S3) is preferred because of its durability, low cost, and its ability to integrate with other cloud services via events. File ingestion into S3 is often the starting point for business processes in the cloud.

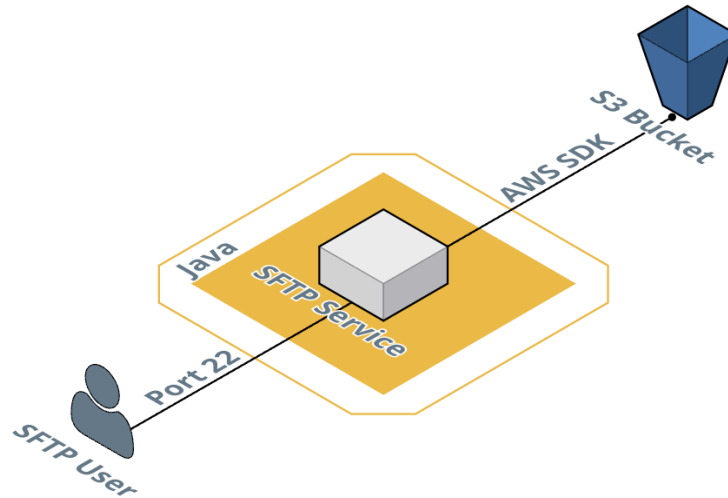
Receiving files from external users poses a challenge in cloud computing. Traditionally, S3 access requires AWS IAM permissions. And IAM access is not typically granted to those outside your organization.

A common approach is to use a service like SFTP, which has its own self-contained user store. Also, SFTP is ubiquitous, so it is ideal when working with external organizations over which you have no control.

Unfortunately, S3 does not have an SFTP interface, and cloud-native services, like AWS Transfer, are complicated and difficult to maintain. The SFTP protocol expects a POSIX-style file system using block storage. But S3 uses object storage, which can only be accessed via a REST API. So, there is often an impasse between SFTP clients trying to send files and somehow getting these files onto S3.

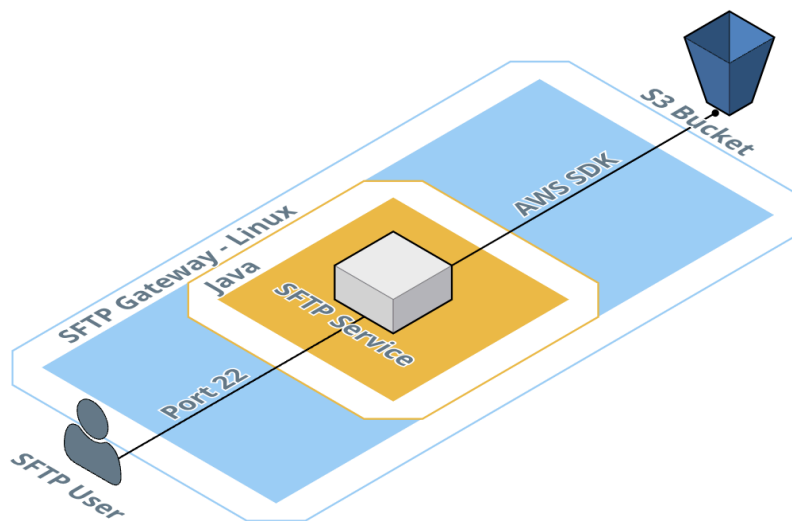


This is where SFTP Gateway comes in. SFTP Gateway provides an SFTP service where users can read and write live files directly to S3 via an SFTP client like FileZilla. In other words, the files that they see are live on S3.



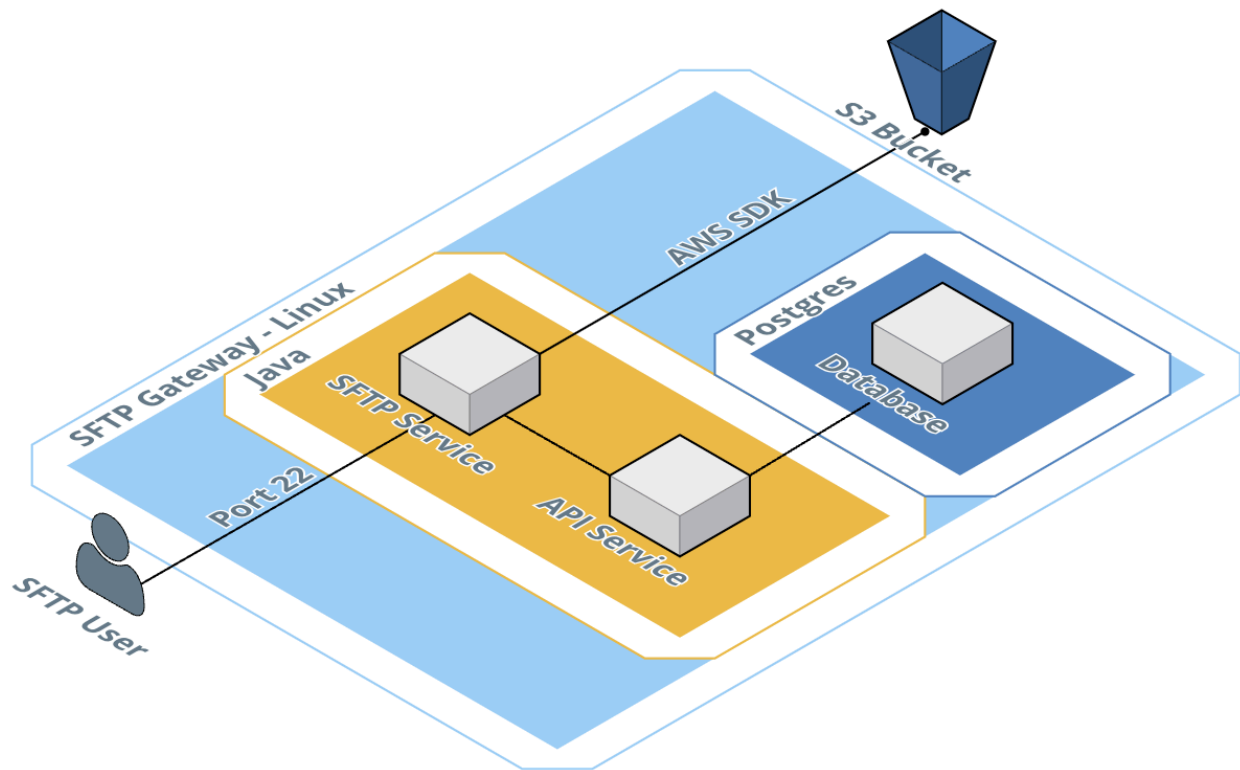
SFTP Gateway accomplishes this by translating SFTP commands to AWS S3 API calls. For example, an SFTP list operation ("ls") would be translated to the corresponding API call ("aws s3 ls"). API responses are normalized so that the SFTP client software receives output in a format defined by the SFTP protocol.

SFTP Gateway is written as a custom Java application. This is necessary because native SFTP services like OpenSSH are tightly coupled to the block storage file system, and cannot be customized to work with the AWS SDK.



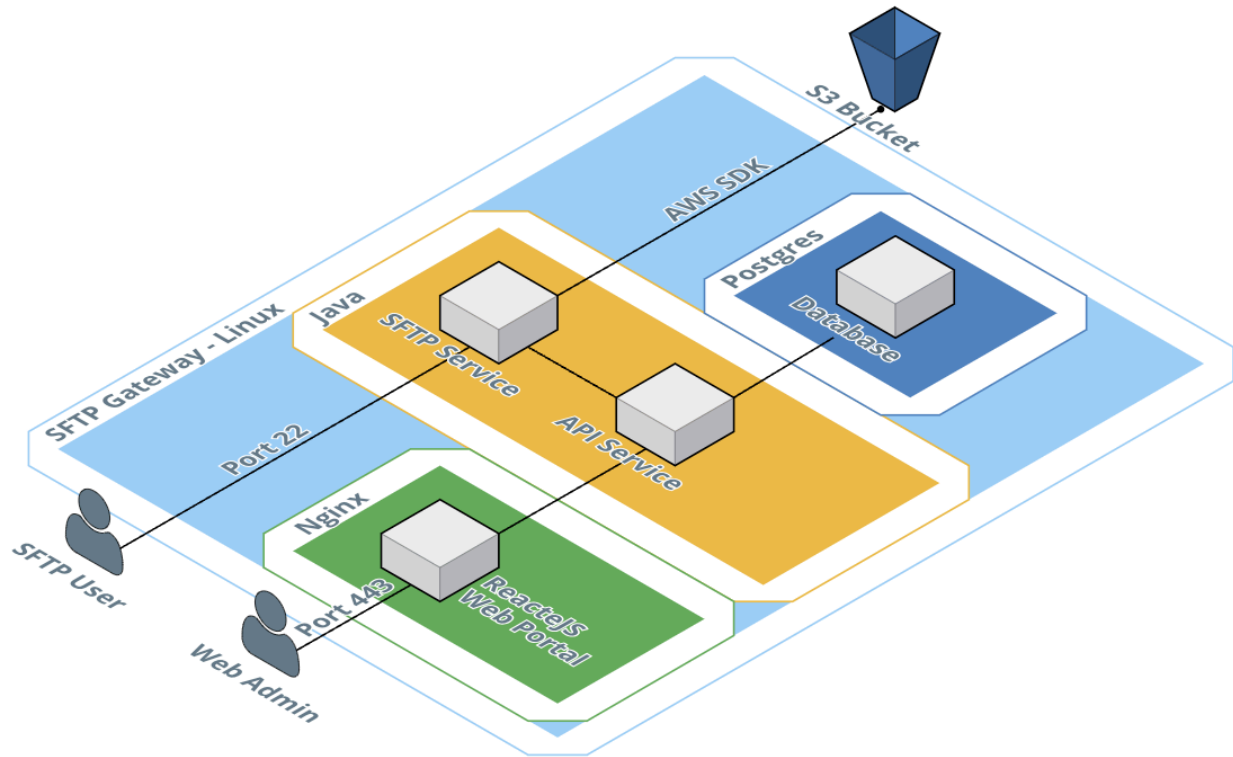
As a Java application, SFTP Gateway runs as a Java service, serving the SFTP protocol over TCP 22. This Java service runs within a Linux VM. The operating system of choice on AWS is Amazon Linux 2023. For Azure and GCP, we use Ubuntu 22.04 LTS.

In addition to the SFTP service, the SFTP Gateway Java application also provides a REST API service. The REST API lets you manage SFTP users, and configure other settings like mapping Home Directories to S3 locations.



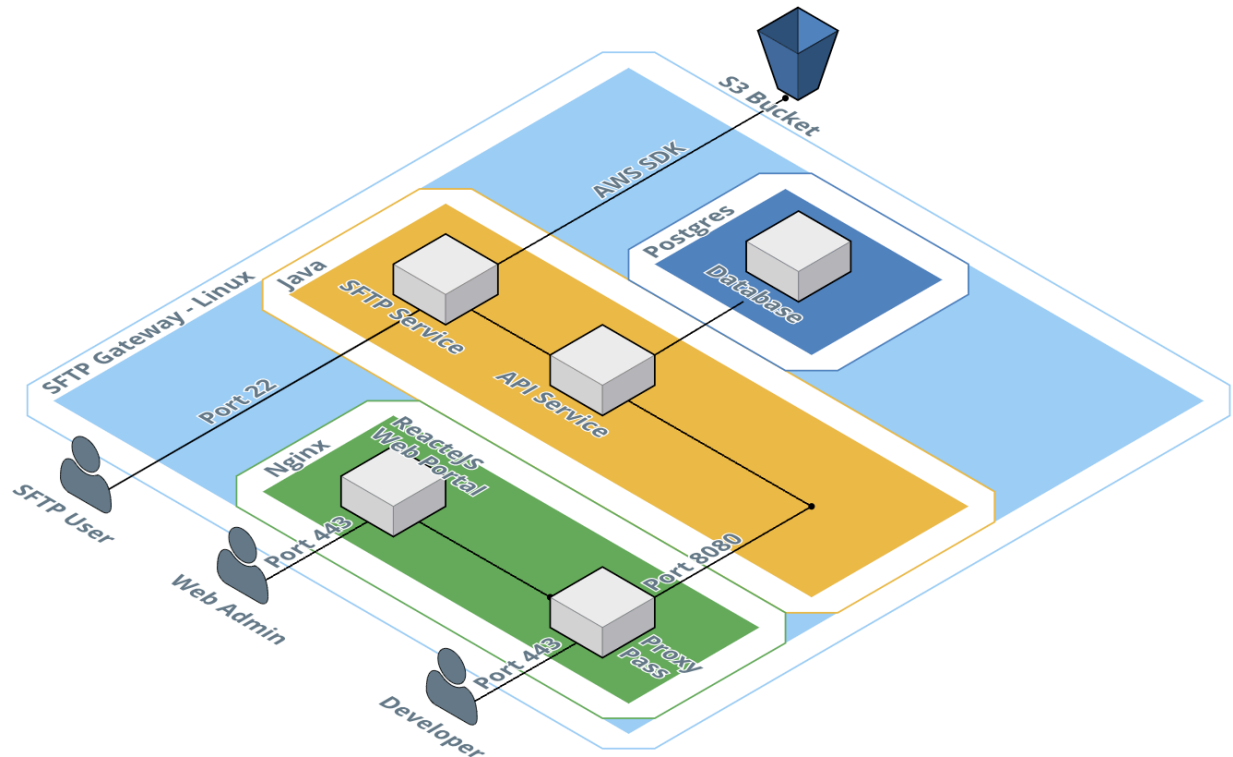
SFTP Gateway users and settings (meta data) are stored in a PostgreSQL database. For single instance deployments, this database is local to the Linux VM. For high availability, the PostgreSQL database lives on a managed database service like RDS that is external to the VMs.

Non-technical administrators can create and configure SFTP users via a user-friendly web admin portal. It is designed to be intuitive, making it easy to perform tasks like generating SSH key pairs at the click of a button. Other features include mapping to S3 locations, creating sharing scenarios, and configuring security.



The web admin portal is a ReactJS web application that is hosted by a pre-configured Nginx web server bundled with SFTP Gateway. Administrators use their web browser to access the web admin portal via HTTPS.

More sophisticated customers may want to fully automate SFTP user management. For example, instead of having an administrator manually click around the web admin portal, a custom script can be triggered by an event to provision a new SFTP user using an authentication token to securely access the REST API.



Our SFTP Gateway web admin portal itself is a consumer of our SFTP Gateway API. The web admin portal is a convenient tool to manage SFTP users via the API, but without requiring any programming skills. This means that anything that is done via the web admin portal can also be achieved via automation.

Security is the highest priority for SFTP Gateway. The product has been independently tested and audited by a leading cyber security firm. A letter of attestation is available upon request to show the results of their security audit and penetration testing.

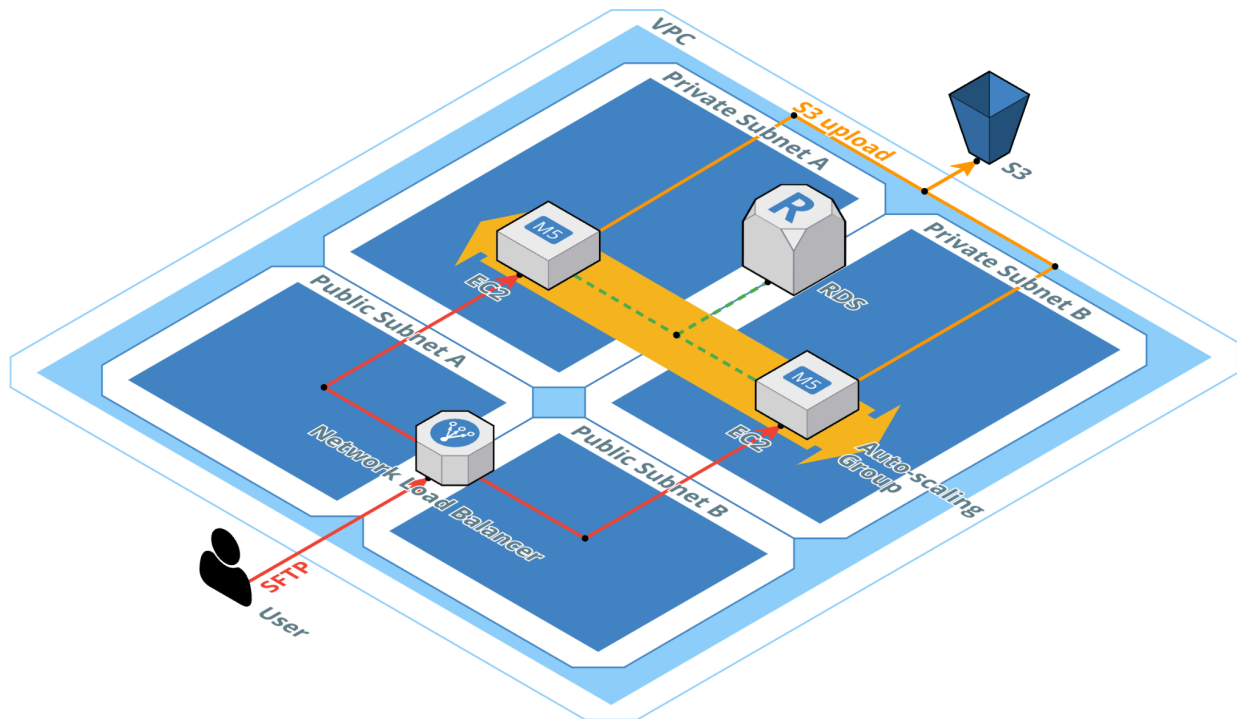
SFTP Gateway High Availability (HA)

SFTP Gateway can be deployed in a Highly Available (HA) configuration to support Enterprise workloads. To accomplish this, application components are decoupled from a single VM instance and leverage cloud native services, such as managed database, load balancing, and log streaming services. Since data is never written to local VM disk volumes, SFTP Gateway instance nodes are ephemeral. Multiple instances can start and stop at any time without impacting the availability of the overall application. The result is a fault-tolerant enterprise application, with virtually limitless capacity, that is deployed across multiple availability zones and can automatically scale to handle any volume.

In the sections that follow, we dive deeper into the individual components of a High Availability deployment. Again, it is important to note that the examples below primarily refer to services offered by Amazon Web Services (AWS), but the same principles apply to all of the major cloud service providers.

HA Architecture

Below is a diagram of SFTP Gateway configured in an HA setup, using Amazon Web Services as an example.



In an HA configuration, multiple instances of SFTP Gateway span different Availability Zones (AZs). This provides business continuity in case of a local data center outage.

The SFTP user connects to an endpoint provided by a Network Load Balancer (NLB). The NLB forwards network traffic on port 22 to one of the EC2 instances.

As an SFTP user authenticates, their credentials are checked against the user store located on a PostgreSQL RDS database. EC2 instances are ephemeral, so user metadata and properties are stored in a database service that is independent from the SFTP Gateway instance nodes.

After data is routed to an instance node, SFTP files are streamed directly to S3 by the Java service running on the instance.

Cloud Native Services

The following sections cover the details of each AWS service used as the file travels from the SFTP user all the way to the S3 bucket.

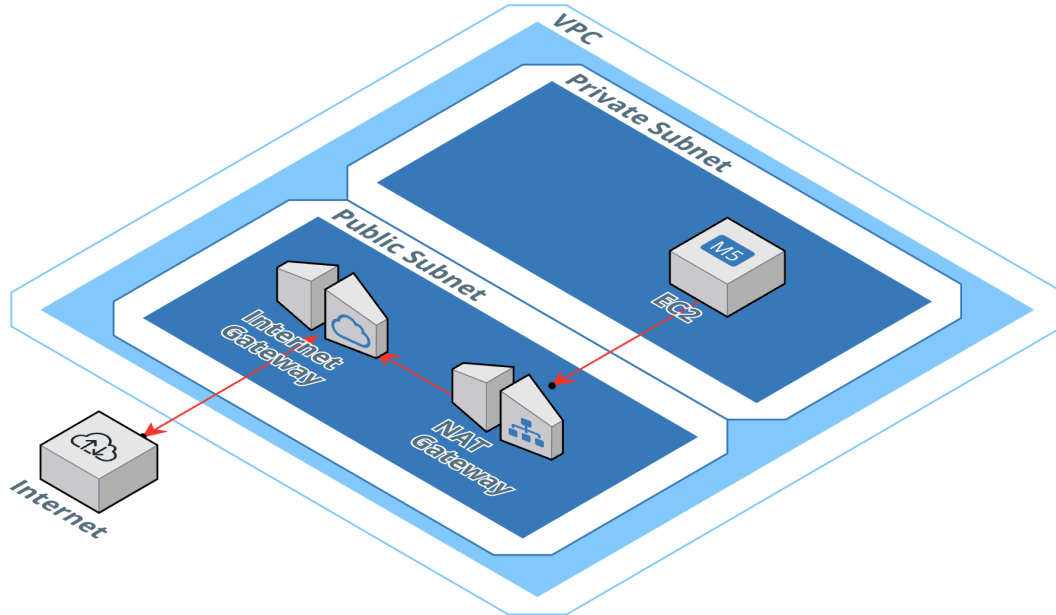
Virtual Private Cloud (VPC)

SFTP Gateway is deployed into a network environment, or Virtual Private Cloud (VPC). Different network configurations are possible, with the simplest being a single instance in a default public subnet.

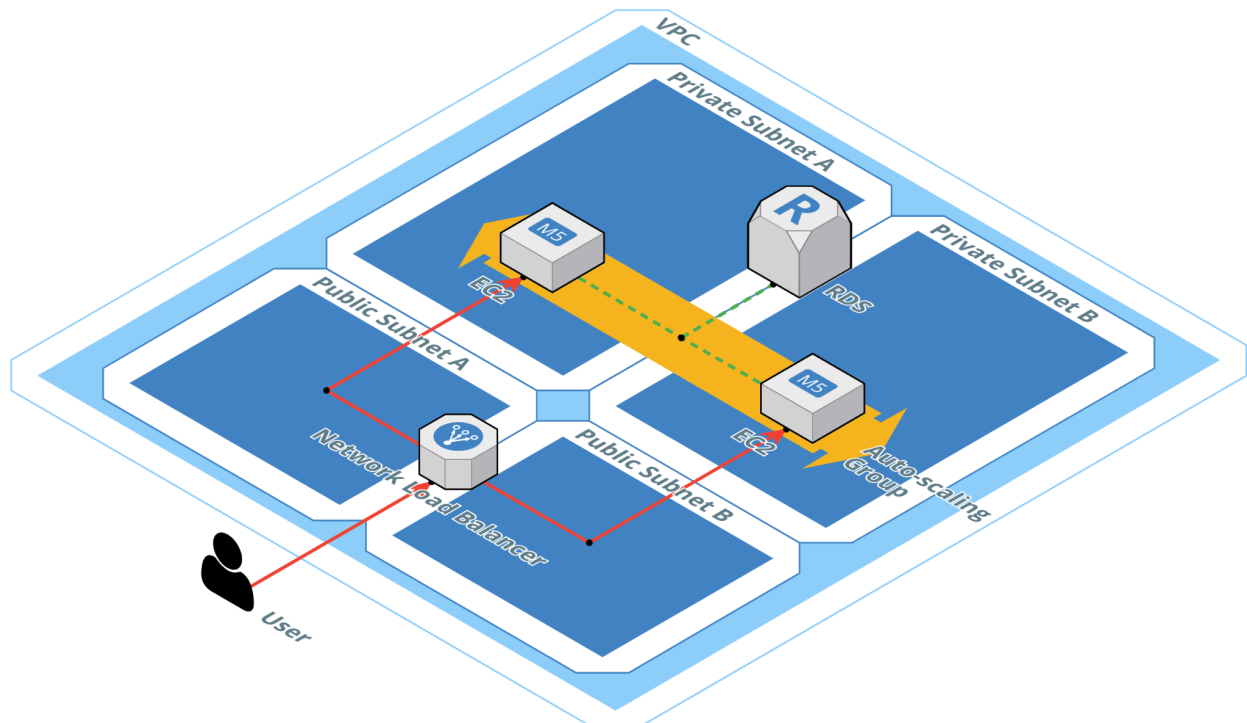
Public and private subnets are defined as follows:

A subnet is **public** if its public IPs are reachable via the internet. This is accomplished using an Internet Gateway (IGW), which provides not only outbound internet connections but also inbound – through the use of a static NAT.

A **private** subnet does not have an IGW and is, therefore, not internet-reachable. To make outbound internet connections, the private subnet relies on a NAT Gateway (NAT GW). The NAT GW lives in a public subnet and passes outbound internet requests to the IGW.



SFTP users need to reach part of your infrastructure over the Internet, so you need a public subnet. All other resources should be protected from Internet exposure by placing them in a private subnet. To achieve high availability, both subnets need to be duplicated across two AZs. This results in needing at least four subnets—two public and two private.



This diagram shows the SFTP Gateway HA configuration deployed into four subnets.

The public subnets contain an NLB, which has an endpoint that SFTP users connect to. The private subnets contain the RDS database, which contains user data, such as public keys.

EC2 instances can be deployed in either the public or private subnets. Deploying them in a public subnet is more convenient for development or testing since you can directly access each instance. For production deployments, however, private subnets are a more secure option because they prevent people from bypassing the NLB and accessing the EC2 instances directly.

In the diagram above, the EC2 instances are deployed in a private subnet, which is the recommended approach.

Network Load Balancer (NLB)

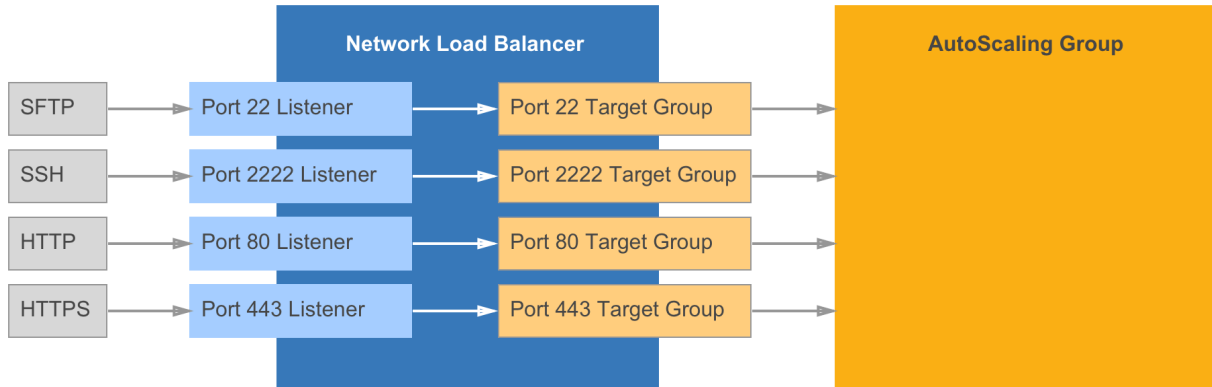
When SFTP users connect, they enter a hostname such as `sftp.example.com`. Typically, this is a DNS record that resolves to the public IP of an SFTP server.

In an HA configuration, the hostname resolves to a DNS endpoint on the NLB. For example, `sftp.example.com` would be a CNAME that points to an NLB endpoint `netwo-1234abcd.elb.us-east-1.amazonaws.com`.

The NLB has Listeners that accept inbound traffic on specific ports. SFTP Gateway uses the following ports:

- **22 (SFTP):** The SFTP Gateway application provides an SFTP service on the default SFTP port. Shell access to the server is not available on this port.
- **2222 (SSH):** The OpenSSH service has been moved to a separate port, making it easier to restrict shell access to authorized administrators and approved IP addresses. This port is available for server maintenance and custom configuration.
- **443 (HTTPS):** Nginx serves a web admin portal that manages SFTP users and settings.
- **80 (HTTP):** Nginx shows an informational page explaining how to create an SSL certificate. This port is also used for LetsEncrypt domain verification.

Each Listener forwards traffic to a Target Group. A Target Group represents a single port for a logical group of EC2 instances. So SFTP traffic would flow from Listener -> Target Group -> Port 22 on one of the EC2 instances.



For SFTP Gateway, Target Groups link to an AutoScaling Group (ASG), which dynamically scales EC2 instances across AZs. The next section covers ASGs.

One important distinction worth noting is that we are not using an Application Load Balancer (ALB). An ALB can inspect HTTP requests, and route traffic accordingly. Since SFTP Gateway uses the SFTP protocol rather than HTTP to transfer files, it must rely on an NLB.

AutoScaling Group

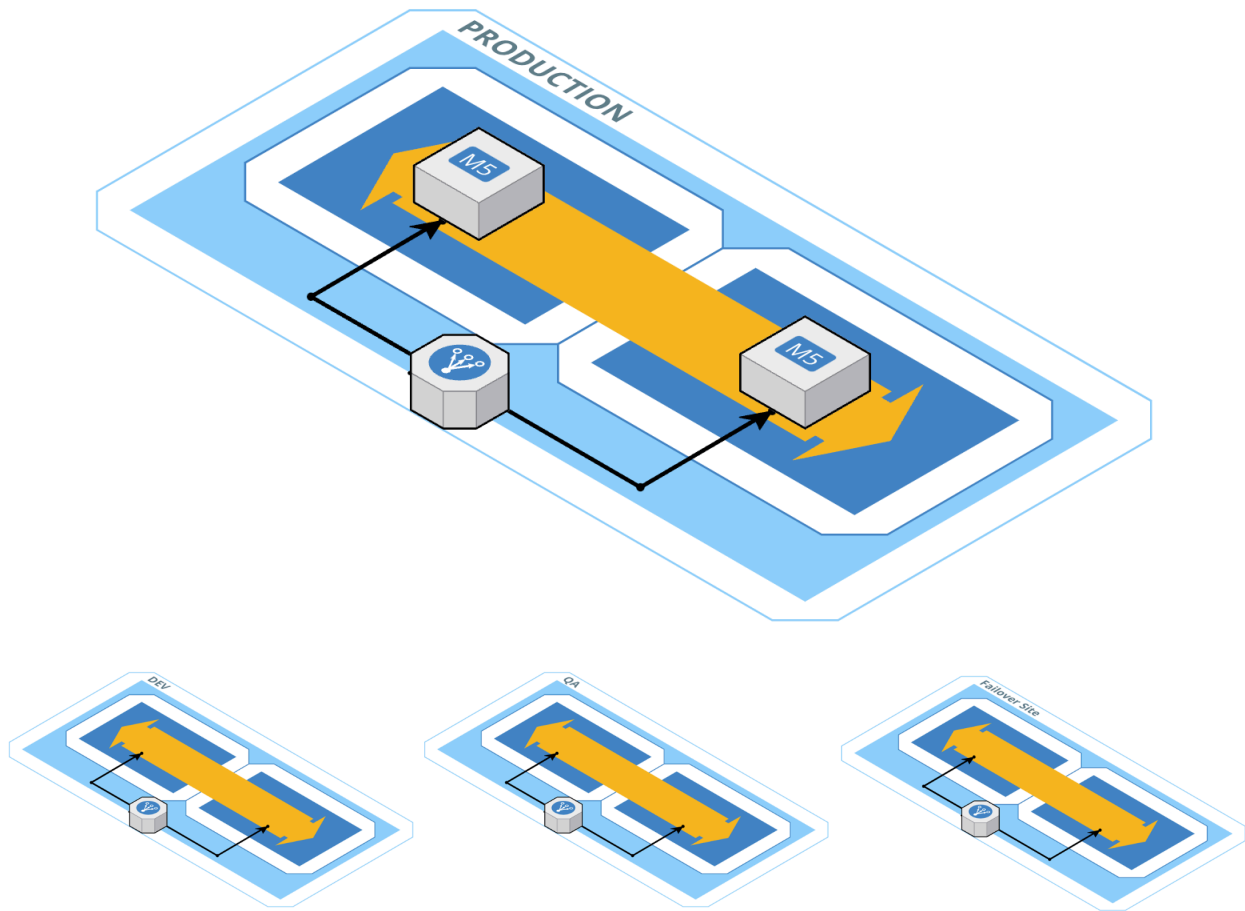
An AutoScaling Group (ASG) deploys EC2 instances across multiple AZs within the same region. You can set a Desired Capacity, which scales the total number of EC2 instances to a specific value. When possible, the ASG spreads out the EC2 instances to cover each AZ.

In the typical HA configuration, the default desired capacity is set to 2. One EC2 instance is deployed in each AZ, which meets the minimum requirements for HA. For greater throughput, multiple EC2 instances can be deployed in each AZ.

There are other situations where it makes sense to set the Desired Capacity to a different value:

Zero instances: You have an environment that is not actively being used, so instances should be spun down – however, you need to be fully operational at a moment's notice. For example, you can have two instances running in Production. But you can spin other environments down to zero instances:

- Dev
- QA
- Disaster recovery or failover site



One instance: You have a DNS service that sends traffic to multiple regions or multiple cloud providers. HA is achieved by having multiple stacks of SFTP Gateway spread across different regions. In this case, each stack might use only one instance.

Four instances: When updating the Launch Configuration to point to a new version of SFTP Gateway, you can temporarily increase the desired capacity from 2 to 4. The incoming instances will reflect the new version. When you decrease the desired capacity back to 2, the old instances are removed. This process is similar to the ASG Instance Refresh feature.

Four or more instances: You can scale out the SFTP Gateway to handle heavy workloads. The NLB distributes traffic across multiple EC2 instances, so each one needs to handle only a fraction of the total workload.

EC2 Instances

SFTP Gateway is an Amazon Machine Image (AMI) that can be spun up as an EC2 instance. The image is Linux-based, running on Amazon Linux 2023.

SFTP Gateway runs a custom Java application that provides an SFTP service on the default SFTP port, TCP 22. Instead of writing files to the local filesystem, files are written directly to S3. This gives SFTP users the ability to read and write live files to S3 via an SFTP client like FileZilla.

Traditionally, OpenSSH has been the standard SFTP service, as it ships with most Linux distributions. OpenSSH depends on the local Linux disk for the SFTP filesystem. This doesn't work well in an HA configuration, because the file state will be inconsistent depending on which server node the SFTP user connects to.

To keep the Linux filesystem state consistent between EC2 servers, one approach is to use an EFS mount. This is a step in the right direction and can work for some scenarios. But EFS can be cost prohibitive for large data sets. S3 is a better long-term storage option. The challenge, then, is to somehow move files from EFS to S3. This solution has limitations – namely, you can't have live read-write, because synchronous actions can't be performed on an asynchronous event-based system.

Another drawback to using EFS is that it is a network disk by nature. SFTP has heavy disk I/O since it involves file transfer. Normally, this is not a problem because disk I/O on locally attached disks is fast. But for a network mount like EFS, you run into situations where the filesystem itself becomes unresponsive. By default, EFS uses burst credits and disk I/O gets throttled when those burst credits are consumed over sustained use. So for heavy workloads, you end up having to pay significant costs for provisioned throughput in order to keep the system stable.

Yet another issue with OpenSSH is that it uses Linux files (/etc/passwd) as the SFTP user store. Local users work well in a single server environment. But for HA, there isn't a good way to sync local Linux users between servers. While you can use an LDAP database to store users, this comes with its own set of challenges. LDAP databases are disk I/O intensive, and cannot be stored on EFS. So you end up having to perform LDAP sync operations. Should anything go wrong with LDAP, troubleshooting can be quite difficult.

SFTP Gateway was designed from the ground up to overcome these issues. Based on customer feedback, here are some of the requirements that needed to go into the HA solution:

- SFTP users must be able to read and write live files to S3
- Files cannot ever be stored on the local Linux filesystem, since local disks can be destroyed at any moment, leading to inconsistent file state
- SFTP user objects and settings must be stored in a separate database service that persists even if all EC2 instances are shut down

Our approach was to implement the SFTP protocol in a custom Java application. This way, we could write SFTP files directly to S3 using the AWS SDK. SFTP uploads can be streamed

directly to S3 from memory without ever touching the filesystem. SFTP users and settings are written to a PostgreSQL database hosted on RDS. Overall, there is very little remaining in terms of local state on each EC2 instance.

PostgreSQL on RDS

The SFTP users and settings are stored in a PostgreSQL database.

For a single instance deployment of SFTP Gateway, the PostgreSQL database is local to the EC2 instance.

For HA, the PostgreSQL database is hosted in an RDS database. EC2 instances are considered to be ephemeral since they can be destroyed or replaced at any time. In order to persist SFTP user objects and configuration settings, this information needs to reside in an external database layer.

In an HA environment, the PostgreSQL RDS database is the source of truth for SFTP user and configuration information. Each EC2 instance is configured via CloudFormation to connect to RDS, so even if all EC2 instances are deleted, the users and settings will still be there for future EC2 instances.

Instead of using a proprietary database for user management, SFTP Gateway can also be configured to use any OIDC-compliant third-party Identity provider, such as Okta, Ping, AWS Cognito, or Azure Active Directory.

S3

SFTP Gateway uses S3 instead of a traditional filesystem for SFTP user Home Directories. This section discusses some of the challenges we faced when implementing this and the design choices we made.

An SFTP service typically saves files to a Linux filesystem, which is Block storage. S3 is a type of Object storage, so an SFTP solution has to reconcile the differences. For example:

- Block storage uses POSIX permissions, Object storage uses ACLs
- Object storage does not have real folders
- Object storage does not allow the moving of files, so you have to copy and delete them instead

Perhaps the most challenging feature in Block storage to duplicate is the ability to mount. SFTP customers sometimes need to configure sharing scenarios. For example, an internal SFTP user needs to read and write to the Home Directories of all external SFTP users. Normally, you can use the mount command in Linux to give two users read/write access to the same location. But

there is no concept of mounting in S3 (we have found open-source tools, such as fuse or s3fs to be unreliable for production environments).

The approach we took was to create a virtual filesystem. This filesystem is stored in the PostgreSQL database, so it's available to every server node. Here are some of the characteristics of this filesystem:

- The root of the filesystem points to a default S3 bucket, which acts as the S3 filesystem.
- Any folder (or subfolder) created will be saved in the PostgreSQL database, and a corresponding folder will be created in S3.
- Folders can be mapped to a different S3 or other Cloud Storage provider location.
- SFTP users can point their Home Directory to any virtual folder.
- SFTP permissions can be assigned at the folder level.

With SFTP Gateway, SFTP users map to a specific S3 location. By default, each SFTP user maps to a sibling folder based on their username, so they cannot traverse to another user's Home Directory. But mappings can be customized to fit scenarios such as one S3 bucket per user, or sharing scenarios like having one internal user accessing the files of several external users.

CloudWatch & Enterprise Observability Tools

SFTP solutions often require audit logging to track the actions of SFTP users.

In an HA scenario, the logs for a server node only include output for that server. In order to track and troubleshoot issues, one would have to log into multiple servers and collate the logs.

SFTP Gateway produces audit log output to record SFTP user actions. Although these audit logs are local to the server, they are streamed to CloudWatch via an agent. This way, logs from each server node are consolidated into a single CloudWatch log group for monitoring and analysis.

Enterprise Observability tools, such as Splunk, can easily be configured to ingest SFTP Gateway application and audit logs and correlate data across multiple instance nodes to identify patterns and anomalies. These tools can:

- Monitor for security incidents by detecting unusual behaviors or access attempts
- Generate real-time dashboards and reports for system health and compliance
- Set up alerts for specific events (failed login attempts, incomplete uploads, etc.)
- Perform detailed forensic analysis by searching and filtering logs
- Aid in regulatory compliance by archiving and analyzing log data for audits

This enhances security, visibility, and operational efficiency across instances.

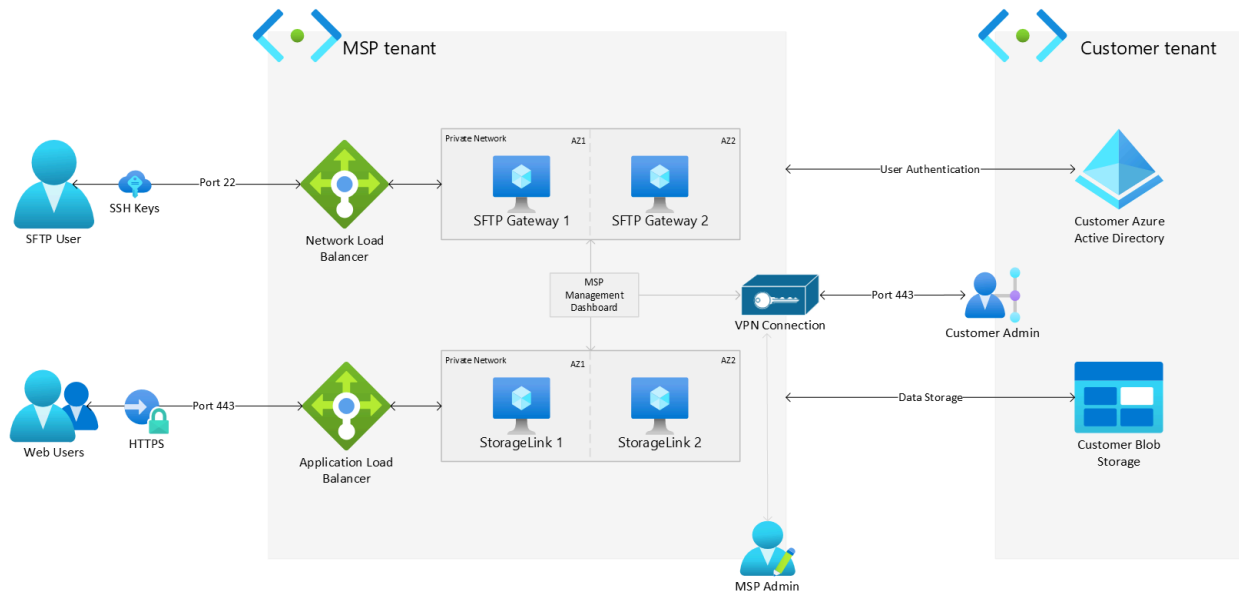
Managed Hosting of SFTP Gateway

SFTP Gateway is secure because it is deployed to your private cloud environment. This allows you to maintain complete control of your data and keep it secure within your organization. However, not every organization has the capacity or expertise to manage enterprise applications in the cloud. In these cases, we have partnered with several Managed Service Providers (MSP) to operate High Availability (HA) stacks of SFTP Gateway on behalf of our customers in the cloud provider of their choice.

Our MSP partners provide the following additional value-added services:

- Continuous monitoring using tools like SolarWinds N-central monitoring agent
- Active Threat detection using AI tools like SentinalOne Complete
- Ransomware and Exploit protection
- 24/7 support and help desk services like Arctic Wolf
- Application patching and updates
- Quarterly manual maintenance
 - System health inspection and audits
 - Security policy review and enforcement
 - Compliance checks

The diagram below shows an Enterprise SFTP Gateway HA deployment by an MSP partner in a Microsoft Azure environment. In this scenario, the MSP is responsible for maintaining load-balanced VM instance nodes and providing network security and monitoring in its own Azure Tenant. However, the SFTP Gateway application is configured to access the customer's Active Directory for user authentication and Azure Blob storage account for storing SFTP files in its own Azure Tenant.



Additional Information

For more information about SFTP Gateway, please refer to the following resources.

SFTP Gateway Knowledge Base - <https://help.thorntech.com/sftp-gateway/>

SFTP Gateway on AWS - <https://aws.amazon.com/marketplace/pp/prodview-atz46phvop3ve>

SFTP Gateway on Azure - <https://azuremarketplace.microsoft.com/en-us/marketplace/apps/thorntechologiesllc.sftpgateway>

SFTP Gateway on GCP - <https://console.cloud.google.com/marketplace/product/thorn-technologies-public/sftp-gateway>

Thorn Technologies YouTube Channel - <https://www.youtube.com/@ThornTechnologies>

G2 Reviews for SFTP Gateway - <https://www.g2.com/products/sftp-gateway/reviews>

Thorn Technologies Website - <https://thorntech.com/>

About Thorn Technologies

Thorn Technologies is disrupting the Managed File Transfer (MFT) industry by providing leaner, reliable, and cost-effective alternatives designed specifically for the cloud. Because Thorn Technology's products are deployed in a user's private cloud environment, Thorn Technologies doesn't touch customers' data or their infrastructure.

Based in the Washington-Baltimore region, Thorn Technologies specializes in developing cloud file transfer software products for organizations around the globe. Clients include Fortune 500 companies spanning multiple industries, such as marketing, data analytics, fintech, retail, manufacturing, healthcare, data storage, and cloud security.

Thorn Technologies products, SFTP Gateway and StorageLink are trusted by organizations of all sizes to seamlessly integrate with systems like Workday, Salesforce, and SAP while effectively replacing legacy solutions such as IBM Sterling File Gateway.

For more information, contact Thorn Technologies at info@thorntech.com