Infrastructure Design Document

Table of Contents

AWS Transfer Family with Lambda Custom Identity Provider Solution

Document Version: 1.0
Date: November 12, 2025
Classification: Internal - Confidential
Project: SFTP Infrastructure Modernization

1. Introduction

1.1 Purpose

This Infrastructure Design Document provides a comprehensive technical specification for the AWS Transfer Family with Lambda Custom Identity Provider solution. The document details the architecture, components, configurations, and implementation approach required to eliminate the single Active Directory group limitation currently impacting SFTP user management.

The primary objectives of this document are to:

- Define the complete technical architecture and component specifications
- Document design decisions, rationale, and alternatives considered
- Specify infrastructure requirements, configurations, and dependencies
- Provide detailed deployment architecture and service flow diagrams
- Establish security, permission, and compliance requirements
- Identify risks, assumptions, constraints, and mitigation strategies
- Serve as the authoritative reference for implementation and operations teams

This document supports the Lambda Custom Identity Provider migration proposal and provides the technical foundation for successful implementation.

1.2 Scope

This design document encompasses the following elements:

In Scope:

- AWS Transfer Family SFTP server configuration with Lambda custom identity provider
- Lambda function design for authentication processing against Windows EC2-based Active Directory
- DynamoDB schema design for user and identity provider configuration storage
- Dual storage backend architecture supporting both Amazon S3 and Amazon EFS
- VPC networking design for Lambda-to-Active Directory connectivity
- IAM role and policy specifications for all components
- Security architecture including encryption, access controls, and audit logging
- Monitoring and alerting framework using Amazon CloudWatch

- Integration with existing Windows EC2-based Microsoft Active Directory infrastructure

Out of Scope:

- Windows Active Directory infrastructure modifications or upgrades
- SFTP client configuration or end-user device management
- Data migration from existing Transfer Family server (covered in migration plan)
- Business continuity and disaster recovery procedures (covered in operational runbook)
- End-user training and change management activities
- Cost optimization strategies beyond initial architecture design

1.3 Intended Audience

This document is designed for the following stakeholders:

Primary Audience:

- **Solutions Architects**: Complete architecture understanding for implementation planning
- **Cloud Developers**: Detailed Lambda function specifications and DynamoDB schema design
- **DevOps Engineers**: Infrastructure deployment, configuration management, and automation
- **System Administrators**: Windows AD integration, user management, and operational procedures
- **Security Engineers**: Security architecture review, compliance validation, and audit requirements

Secondary Audience:

- **Project Managers**: Technical scope understanding for timeline and resource planning
- **Business Stakeholders**: High-level architecture overview and business impact assessment
- **Quality Assurance**: Testing requirements and validation criteria
- **Operations Teams**: Post-deployment support and maintenance requirements

Technical Prerequisite Knowledge:
Readers should have familiarity with AWS services (Lambda, DynamoDB, Transfer Family), LDAP/Active Directory concepts, IAM security model, and VPC networking fundamentals.

2. Application and Systems

2.1 Overview

The AWS Transfer Family Lambda Custom Identity Provider solution is a serverless SFTP infrastructure that enables secure file transfer capabilities while eliminating the limitations of native Active Directory integration. The solution processes unlimited Active Directory group memberships through custom authentication logic, provides flexible per-user storage backend assignment, and maintains seamless integration with existing Windows EC2-based Microsoft Active Directory infrastructure.

2.1.1 Business Context

Current State Challenges:

- Single AD group restriction prevents users with multiple organizational roles from accessing SFTP resources
- 100 Active Directory group server limit constrains organizational growth and flexibility
- Single storage backend (S3 only) cannot accommodate diverse use case requirements
- Manual workarounds for complex permission scenarios create administrative overhead
- Limited scalability for future organizational expansion

Solution Benefits:

- **Unlimited Group Memberships**: Custom Lambda logic processes all user AD groups without restrictions
- **Flexible Access Control**: Dynamic permission aggregation from multiple group memberships
- **Dual Storage Support**: Per-user assignment of S3 (object storage) or EFS (POSIX file system)
- **Scalability**: Serverless architecture automatically scales with user growth
- **Cost Efficiency**: Pay-per-use model with no infrastructure management overhead
- **Operational Simplicity**: AWS managed services eliminate patching and maintenance burden

2.1.2 System Capabilities

The solution provides the following core capabilities:

Authentication and Authorization:

- LDAP-based authentication against existing Windows EC2 Active Directory
- Multi-group membership processing with unlimited group support
- Dynamic session policy generation based on aggregated group permissions
- Per-user IAM role and resource access configuration

Storage Management:

- Dual backend support: Amazon S3 for object storage, Amazon EFS for POSIX file systems
- Per-user storage backend assignment based on use case requirements
- Dynamic home directory mapping with logical directory structures
- Support for 10+ TB S3 and 5+ TB EFS storage capacity

Security and Compliance:

- Encryption in transit (TLS/SSL for SFTP, HTTPS for AWS APIs, LDAPS for AD)
- Encryption at rest (S3 SSE-KMS, EFS encryption, DynamoDB encryption)
- Comprehensive audit logging (CloudWatch Logs, CloudTrail for API calls)
- IAM-based access control with principle of least privilege

- VPC isolation for Lambda-to-AD connectivity

Monitoring and Operations:

- Real-time authentication metrics and success/failure tracking
- Performance monitoring for Lambda execution and response times
- Storage utilization tracking and capacity planning
- Automated alerting for authentication failures and performance degradation

2.2 Environment

2.2.1 Deployment Architecture

The solution is deployed across multiple AWS services in a highly available, serverless architecture:

[^158]

Architecture Layers:

Client Layer:

- SFTP clients (FileZilla, WinSCP, command-line sftp) connect to Transfer Family endpoint
- Supported protocols: SFTP (SSH File Transfer Protocol) over TLS
- Authentication methods: Username/password via Lambda custom IdP

Edge Layer:

- AWS Transfer Family server acts as managed SFTP endpoint
- VPC endpoint configuration for private network access (optional)
- Public endpoint for internet-accessible SFTP service
- Lambda custom identity provider configuration for authentication

Application Layer:

- Lambda function (Python 3.9) processes authentication requests
- VPC integration for secure connectivity to Windows Active Directory
- LDAP client libraries for Windows AD communication
- Custom business logic for multi-group processing and permission aggregation

Data Layer:

- DynamoDB tables for user and identity provider configuration storage
- Amazon S3 buckets for object storage use cases
- Amazon EFS file systems for POSIX-compliant file storage
- CloudWatch Logs for centralized logging and audit trails

Management Layer:

- Amazon CloudWatch for metrics, logging, and alerting
- AWS CloudTrail for API activity tracking and compliance
- AWS IAM for identity and access management
- AWS Systems Manager for configuration management (optional)

2.2.2 Resource Details

AWS Transfer Family Server:

- **Type**: Managed SFTP server with Lambda custom identity provider
- **Protocol**: SFTP (SSH File Transfer Protocol)
- **Endpoint**: VPC endpoint or public endpoint with Elastic IP
- **Identity Provider**: AWS Lambda function ARN
- **Logging**: CloudWatch Logs integration for session and file transfer logging
- **Capacity**: Supports 500+ concurrent users with automatic scaling

Lambda Function - Custom Identity Provider:

- **Runtime**: Python 3.9
- **Memory**: 512 MB (adjustable based on performance requirements)
- **Timeout**: 15 minutes (maximum for authentication processing)
- **VPC Configuration**: Private subnets in 2 Availability Zones
- **Environment Variables**:
  - `AD_SERVER`: Windows AD domain controller hostname/IP
  - `AD_SEARCH_BASE`: LDAP search base DN (e.g., DC=corp,DC=com)
  - `AD_SECRET_ARN`: AWS Secrets Manager ARN for AD service account credentials
  - `USER_TABLE_NAME`: DynamoDB user configuration table name
  - `IDP_TABLE_NAME`: DynamoDB identity provider configuration table name
  - `LOG_LEVEL`: Logging verbosity (INFO, DEBUG, WARNING, ERROR)
- **Dependencies**: boto3 (AWS SDK), python-ldap (LDAP client), cryptography (credential encryption)

DynamoDB Tables:

User Configuration Table:

- **Table Name**: `transfer-family-users`
- **Primary Key**: `username` (String)
- **Attributes**:
  - `identity_provider_key`: Reference to IdP configuration
  - `storage_backend`: "S3" or "EFS"

    - `role_arn`: IAM role ARN for user's storage access
    - `home_directory_type`: "PATH" or "LOGICAL"
    - `home_directory_details`: JSON configuration for directory mappings
    - `posix_profile`: UID, GID for EFS users (optional)
    - `session_policy`: Additional IAM policy restrictions (optional)
- **Billing Mode**: On-demand (pay per request)
- **Encryption**: AWS-managed encryption key
- **Point-in-Time Recovery**: Enabled
- **Capacity Estimate**: 500 user records, ~5 KB per record = 2.5 MB total

Identity Provider Configuration Table:

- **Table Name**: `transfer-family-idp-config`
- **Primary Key**: `identity_provider_key` (String)
- **Attributes**:
    - `provider_type`: "LDAP" (Windows AD)
    - `ldap_host`: AD domain controller hostname/IP
    - `ldap_port`: 389 (LDAP) or 636 (LDAPS)
    - `bind_dn`: Service account distinguished name
    - `search_base`: LDAP search base DN
    - `user_search_filter`: LDAP filter for user lookup
    - `group_attribute`: AD attribute for group membership
    - `ip_allowlist`: Optional IP restriction list
- **Billing Mode**: Provisioned (5 RCU / 5 WCU - low traffic)
- **Encryption**: AWS-managed encryption key
- **Capacity Estimate**: 5 IdP configurations, ~10 KB per record = 50 KB total

VPC Configuration:

- **VPC**: Existing or new VPC with CIDR block (e.g., 10.0.0.0/16)
- **Subnets**:
    - Private subnets in 2 Availability Zones for Lambda functions
    - Subnet CIDR examples: 10.0.1.0/24 (AZ-1), 10.0.2.0/24 (AZ-2)
- **NAT Gateway**: One NAT Gateway per AZ for Lambda internet access
- **Route Tables**: Private subnet routes to NAT Gateway for 0.0.0.0/0
- **VPC Endpoints** (optional cost optimization):
    - DynamoDB VPC endpoint (Gateway endpoint, no charge)
    - S3 VPC endpoint (Gateway endpoint, no charge)

Security Groups:

Lambda Security Group:

- **Outbound Rules**:
  - LDAP: Port 389 to Windows AD security group (unencrypted)
  - LDAPS: Port 636 to Windows AD security group (encrypted, recommended)
  - HTTPS: Port 443 to 0.0.0.0/0 (AWS API calls via NAT Gateway)
- **Inbound Rules**: None (Lambda initiates all connections)

Transfer Family Security Group:

- **Inbound Rules**:
  - SFTP: Port 22 from authorized IP ranges (e.g., corporate network, VPN)
  - Source IP examples: 203.0.113.0/24 (corporate), 198.51.100.0/24 (VPN)
- **Outbound Rules**:
  - HTTPS: Port 443 to Lambda function (custom IdP invocation)

Windows AD Security Group (existing, modification required):

- **Inbound Rules** (add):
  - LDAP: Port 389 from Lambda security group
  - LDAPS: Port 636 from Lambda security group (recommended)

Storage Backends:

Amazon S3:

- **Bucket Naming**: `<company>-sftp-<environment>-<region>` (e.g., acme-sftp-prod-us-east-1)
- **Storage Class**: S3 Standard (transition to Intelligent-Tiering or Glacier for archival)
- **Versioning**: Enabled for data protection and recovery
- **Encryption**: SSE-KMS with customer-managed key for enhanced security
- **Bucket Policy**: IAM-based access control via Transfer Family user roles
- **Lifecycle Policies**: Configure based on data retention requirements
- **Capacity**: 10 TB initial allocation, expandable on demand

Amazon EFS:

- **Performance Mode**: General Purpose (low latency, suitable for SFTP workloads)
- **Throughput Mode**: Bursting (scales with file system size)
- **Encryption**: Enabled for encryption at rest with AWS-managed key
- **Mount Targets**: One per Availability Zone in VPC private subnets
- **Access Points**: Configured per user or user group with POSIX permissions

- **Backup**: AWS Backup with daily snapshots and 30-day retention
- **Capacity**: 5 TB initial allocation, automatically scales

IAM Roles and Policies:

Lambda Execution Role: `TransferFamily-Lambda-ExecutionRole`

- Managed Policies: `AWSLambdaVPCAccessExecutionRole`
- Custom Policy: DynamoDB read access, Secrets Manager read access, CloudWatch Logs write

Transfer Family Server Role: `TransferFamily-Server-InvokeLambdaRole`

- Custom Policy: Lambda InvokeFunction permission for custom IdP function

Transfer Family User Role - S3: `TransferFamily-User-S3AccessRole`

- Custom Policy: S3 bucket access scoped to user's home directory path, KMS decrypt permissions

Transfer Family User Role - EFS: `TransferFamily-User-EFSAccessRole`

- Custom Policy: EFS ClientMount, ClientRead, ClientWrite on specific access points, KMS decrypt

Monitoring and Logging:

CloudWatch Log Groups:

- `/aws/lambda/transfer-custom-idp`: Lambda function execution logs, 30-day retention
- `/aws/transfer/<server-id>`: Transfer Family SFTP session logs, 90-day retention
- Log Insights queries for authentication analysis and troubleshooting

CloudWatch Metrics:

- Lambda: Invocations, Duration, Errors, Throttles, ConcurrentExecutions
- Transfer Family: BytesIn, BytesOut, FilesIn, FilesOut, ActiveUsers
- DynamoDB: ConsumedReadCapacityUnits, ConsumedWriteCapacityUnits, UserErrors
- Custom Metrics: Authentication success rate, group processing time, storage backend distribution

CloudWatch Alarms:

- Lambda error rate > 1% for 5 minutes
- Authentication failure rate > 5% for 10 minutes
- Lambda duration > 10 seconds (P99) for 5 minutes
- Transfer Family server unavailable for 2 minutes

3. Specifications

3.1 Technical Specifications

3.1.1 Lambda Function Specifications

Function Configuration:

```
# Lambda Function: transfer-custom-idp<a></a>
Runtime: python3.9
Handler: lambda_function.lambda_handler
Memory: 512 MB
Timeout: 900 seconds (15 minutes)
VPC Configuration: Enabled
Subnets: [subnet-xxxxx (AZ-1), subnet-yyyyy (AZ-2)]
Security Groups: [sg-lambda-transfer-idp]
Environment Variables:
  AD_SERVER: dc.corp.example.com
  AD_SEARCH_BASE: DC=corp,DC=example,DC=com
  AD_SECRET_ARN: arn:aws:secretsmanager:region:account:secret:ad-service-account
  USER_TABLE_NAME: transfer-family-users
  IDP_TABLE_NAME: transfer-family-idp-config
  LOG_LEVEL: INFO
```

Lambda Function Response Schema:

```
{
  "Role": "arn:aws:iam::123456789012:role/TransferFamily-User-S3AccessRole",
  "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[...]}",
  "HomeDirectoryType": "LOGICAL",
  "HomeDirectoryDetails": [
    {
      "Entry": "/",
      "Target": "/bucket-name/home/${transfer:UserName}"
    },
    {
      "Entry": "/shared",
      "Target": "/bucket-name/shared"
    }
  ],
  "PosixProfile": {
    "Uid": 1001,
    "Gid": 1001,
    "SecondaryGids": [2001, 2002]
  },
  "PublicKeys": []
}
```

3.1.2 DynamoDB Schema Specifications

User Configuration Table Schema:

```
{
  "username": "alice.smith",
  "identity_provider_key": "windows-ad-primary",
  "storage_backend": "S3",
  "role_arn": "arn:aws:iam::123456789012:role/TransferFamily-User-S3AccessRole",
  "home_directory_type": "LOGICAL",
  "home_directory_details": [
    {
      "Entry": "/",
      "Target": "/acme-sftp-prod/users/${transfer:UserName}"
    },
    {
      "Entry": "/finance",
      "Target": "/acme-sftp-prod/departments/finance"
    },
    {
      "Entry": "/projects",
      "Target": "/acme-sftp-prod/projects/projecta"
    }
  ],
  "session_policy": {
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["s3:ListBucket"],
        "Resource": "arn:aws:s3:::acme-sftp-prod",
        "Condition": {
          "StringLike": {
            "s3:prefix": ["users/${transfer:UserName}/*", "departments/finance/*"]
          }
        }
      }
    ]
  }
}
```

Identity Provider Configuration Schema:

```
{
  "identity_provider_key": "windows-ad-primary",
  "provider_type": "LDAP",
  "ldap_host": "dc01.corp.example.com",
  "ldap_port": 636,
  "use_ssl": true,
  "bind_dn": "CN=svc-transfer,OU=ServiceAccounts,DC=corp,DC=example,DC=com",
  "search_base": "DC=corp,DC=example,DC=com",
  "user_search_filter": "(&amp;(objectClass=user)(sAMAccountName={username}))",
  "group_attribute": "memberOf",
  "group_search_filter": "(&amp;(objectClass=group)(member={user_dn}))",
```

```
  "ip_allowlist": ["203.0.113.0/24", "198.51.100.0/24"],
  "trace_enabled": true
}
```

3.1.3 Network Specifications

VPC CIDR Block: 10.0.0.0/16

Subnet Allocation:

- Private Subnet AZ-1: 10.0.1.0/24 (Lambda functions)
- Private Subnet AZ-2: 10.0.2.0/24 (Lambda functions)
- Public Subnet AZ-1: 10.0.101.0/24 (NAT Gateway)
- Public Subnet AZ-2: 10.0.102.0/24 (NAT Gateway - optional HA)

Route Tables:

- Private RT: 0.0.0.0/0 → NAT Gateway, 10.0.0.0/16 → local
- Public RT: 0.0.0.0/0 → Internet Gateway, 10.0.0.0/16 → local

DNS Configuration:

- Enable DNS hostnames: Yes
- Enable DNS resolution: Yes
- DHCP Options Set: Custom with Windows AD DNS servers for domain resolution

3.2 Requirements

3.2.1 Functional Requirements

FR-001: Multi-Group Authentication

- Lambda function MUST process unlimited Active Directory group memberships per user
- Group membership processing MUST include nested groups (recursive search)
- Authentication response time MUST be less than 3 seconds for 95th percentile

FR-002: Dual Storage Backend Support

- Solution MUST support both Amazon S3 and Amazon EFS storage backends
- Per-user storage backend assignment MUST be configurable in DynamoDB
- Storage backend selection MUST be dynamic based on user configuration

FR-003: Session Policy Generation

- Lambda MUST generate appropriate IAM session policies based on group aggregation
- Home directory mappings MUST support LOGICAL type for flexible path management
- POSIX profile MUST be provided for EFS users with appropriate UID/GID

FR-004: LDAP Integration

- Lambda MUST authenticate users against Windows EC2-based Active Directory
- LDAPS (LDAP over SSL) MUST be supported for encrypted communication
- Service account credentials MUST be securely stored in AWS Secrets Manager

FR-005: Audit and Compliance

- All authentication attempts MUST be logged to CloudWatch Logs
- Successful and failed authentication events MUST be tracked with metrics
- SFTP file transfer activities MUST be logged for compliance requirements

3.2.2 Non-Functional Requirements

NFR-001: Performance

- Lambda function cold start time MUST NOT exceed 5 seconds
- DynamoDB read operations MUST complete within 10 milliseconds
- LDAP authentication queries MUST complete within 2 seconds

NFR-002: Scalability

- Solution MUST support 500 concurrent SFTP user sessions
- Lambda concurrent executions MUST be monitored and scaled automatically
- DynamoDB tables MUST use on-demand billing for automatic scaling

NFR-003: Availability

- Solution MUST achieve 99.9% availability (AWS Transfer Family SLA)
- Multi-AZ deployment MUST be used for all critical components
- Automatic failover MUST occur without manual intervention

NFR-004: Security

- All data in transit MUST be encrypted (TLS/SSL, LDAPS)
- All data at rest MUST be encrypted (KMS-managed keys)
- IAM roles MUST follow principle of least privilege
- Security groups MUST restrict access to minimum required ports

NFR-005: Maintainability

- Infrastructure MUST be deployed using Infrastructure as Code (IaC)
- Configuration changes MUST NOT require Lambda function redeployment
- Monitoring and alerting MUST provide proactive issue detection

3.2.3 Compliance Requirements

Data Encryption:

- FIPS 140-2 compliant encryption for data at rest (AWS KMS)
- TLS 1.2 or higher for data in transit

Audit Logging:

- Centralized logging for all authentication and file transfer activities
- Log retention of 90 days for SFTP sessions, 30 days for Lambda execution
- CloudTrail enabled for all API activity with 1-year retention

Access Control:

- Multi-factor authentication for administrative access (AWS Console/CLI)
- Role-based access control (RBAC) for AWS resource management
- Separation of duties between development and production environments

3.3 Permission Mapping

[^156]

3.3.1 IAM Role Definitions

Lambda Execution Role Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:region:account:log-group:/aws/lambda/transfer-custom-idp:
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:Query",
        "dynamodb:Scan"
      ],
      "Resource": [
        "arn:aws:dynamodb:region:account:table/transfer-family-users",
        "arn:aws:dynamodb:region:account:table/transfer-family-idp-config"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:region:account:secret:ad-service-account-*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

Transfer Family User Role - S3 Access Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListingOfUserFolder",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::acme-sftp-prod",
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "users/${transfer:UserName}/*",
            "users/${transfer:UserName}"
          ]
        }
      }
    },
    {
      "Sid": "AllowUserFileAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": "arn:aws:s3:::acme-sftp-prod/users/${transfer:UserName}/*"
    },
    {
      "Sid": "AllowKMSDecrypt",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
```

```
      ],
      "Resource": "arn:aws:kms:region:account:key/key-id"
    }
  ]
}
```

3.4 IdP Configuration

3.4.1 Lambda Custom IdP Setup

Transfer Family Server Configuration:

```
aws transfer create-server \
  --endpoint-type PUBLIC \
  --protocols SFTP \
  --identity-provider-type AWS_LAMBDA \
  --identity-provider-details "Function=arn:aws:lambda:region:account:function:transfer-c
  --logging-role "arn:aws:iam::account:role/TransferFamily-CloudWatchLoggingRole" \
  --security-policy-name "TransferSecurityPolicy-2022-03"
```

DynamoDB User Configuration Example:

```
# User with multiple AD groups: Finance, ProjectA, Contractors<a></a>
import boto3

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('transfer-family-users')

user_config = {
    'username': 'bob.jones',
    'identity_provider_key': 'windows-ad-primary',
    'storage_backend': 'S3',
    'role_arn': 'arn:aws:iam::account:role/TransferFamily-User-S3AccessRole',
    'home_directory_type': 'LOGICAL',
    'home_directory_details': [
        {'Entry': '/', 'Target': '/acme-sftp-prod/users/bob.jones'},
        {'Entry': '/finance', 'Target': '/acme-sftp-prod/departments/finance'},
        {'Entry': '/projects', 'Target': '/acme-sftp-prod/projects/projecta'},
        {'Entry': '/contractors', 'Target': '/acme-sftp-prod/groups/contractors'}
    ]
}

table.put_item(Item=user_config)
```

3.4.2 Windows AD Integration

LDAP Service Account Requirements:

- Account Name: `svc-transfer-idp`
- Permissions: Read access to user objects and group memberships

- Password Policy: Non-expiring password (or automated rotation via Secrets Manager)
- Security: Account should not have interactive logon rights

LDAP Connection String:

```
ldaps://dc01.corp.example.com:636
Base DN: DC=corp,DC=example,DC=com
User Search Filter: (&amp;(objectClass=user)(sAMAccountName={username}))
Group Search Filter: (&amp;(objectClass=group)(member={user_dn}))
```

Secrets Manager Configuration:

```
{
  "username": "svc-transfer-idp@corp.example.com",
  "password": "&lt;secure-password&gt;",
  "bind_dn": "CN=svc-transfer-idp,OU=ServiceAccounts,DC=corp,DC=example,DC=com"
}
```

3.5 Service Flow and Setup

[^159]

3.5.1 Authentication Flow Sequence

Step-by-Step Process:

1. **User Connection Initiation**
   - User opens SFTP client (e.g., FileZilla, WinSCP)
   - Enters SFTP endpoint: `sftp.example.com` or `<server-id>.server.transfer.region.amazonaws.com`
   - Provides username and password credentials
2. **Transfer Family Request Processing**
   - AWS Transfer Family receives SFTP connection request
   - Extracts username, password, and source IP address
   - Invokes Lambda custom identity provider function with event payload
3. **Lambda IdP Invocation**
   - Lambda function receives event with authentication request
   - Event payload contains: `username`, `password`, `serverId`, `sourceIp`, `protocol`
4. **IdP Configuration Retrieval**
   - Lambda queries DynamoDB IdP configuration table
   - Retrieves LDAP connection parameters for Windows AD
   - Loads service account credentials from Secrets Manager
5. **User Configuration Lookup**

- Lambda queries DynamoDB user configuration table using username as key
- Retrieves user-specific settings: storage backend, role ARN, home directory mappings
- If user not found, returns authentication failure

6. **Windows AD Authentication**
   - Lambda establishes LDAPS connection to Windows AD domain controller
   - Binds with service account credentials
   - Searches for user DN using `sAMAccountName={username}`
   - Attempts bind with user's provided credentials
   - If authentication fails, returns error response to Transfer Family
7. **Group Membership Processing**
   - Lambda performs LDAP search for all user group memberships
   - Queries `memberOf` attribute for direct group memberships
   - Optionally performs recursive search for nested groups (unlimited depth)
   - Builds complete list of group DNs and group names
8. **Permission Aggregation**
   - Lambda processes all group memberships to determine combined permissions
   - Maps groups to home directory paths (e.g., Finance group → /finance folder)
   - Constructs `HomeDirectoryDetails` array with all accessible paths
   - Applies any additional session policy restrictions
9. **Storage Backend Determination**
   - Lambda checks user configuration for storage backend type (S3 or EFS)
   - For S3: Prepares LOGICAL home directory mappings with S3 bucket paths
   - For EFS: Prepares PATH home directory with EFS file system ID and POSIX profile
10. **Session Configuration Response**
    - Lambda constructs response object with all session parameters
    - Includes: `Role`, `Policy`, `HomeDirectoryType`, `HomeDirectoryDetails`, `PosixProfile`
    - Returns response to AWS Transfer Family server
11. **IAM Role Assumption**
    - Transfer Family assumes the IAM role specified in Lambda response
    - Applies session policy for additional access restrictions
    - Validates role has necessary permissions for storage backend access
12. **Access Granted**
    - User SFTP session established with authenticated credentials
    - User can perform file operations (list, get, put, delete) within authorized directories
    - All file transfer activities logged to CloudWatch Logs

3.5.2 Setup Procedure

Phase 1: Prerequisites (Week 1)

1. Create VPC with private subnets and NAT Gateway
2. Configure security groups for Lambda and Transfer Family
3. Create service account in Windows AD with LDAP read permissions
4. Store AD service account credentials in AWS Secrets Manager

Phase 2: DynamoDB Setup (Week 1)

1. Create `transfer-family-users` table with username as primary key
2. Create `transfer-family-idp-config` table with identity_provider_key as primary key
3. Enable point-in-time recovery and encryption for both tables
4. Populate IdP configuration with Windows AD connection parameters

Phase 3: Lambda Function Deployment (Week 2)

1. Package Lambda function code with dependencies (boto3, python-ldap)
2. Create Lambda execution IAM role with required permissions
3. Deploy Lambda function in VPC with appropriate security group
4. Configure environment variables for AD connectivity and table names
5. Test Lambda function with sample authentication requests

Phase 4: Transfer Family Configuration (Week 2)

1. Create Transfer Family server with Lambda custom IdP
2. Configure server endpoint type (PUBLIC or VPC)
3. Associate CloudWatch logging role for session logs
4. Create IAM roles for user storage access (S3 and EFS)
5. Configure S3 buckets and EFS file systems for storage backends

Phase 5: User Configuration (Week 3)

1. Populate DynamoDB user table with user records and home directory mappings
2. Test authentication for pilot users with single and multiple group memberships
3. Validate storage access for both S3 and EFS users
4. Configure CloudWatch alarms for authentication monitoring

Phase 6: Production Cutover (Week 4)

1. Update DNS records to point to new Transfer Family endpoint
2. Migrate remaining users from legacy SFTP server
3. Decommission old Transfer Family server after validation
4. Establish ongoing monitoring and operational procedures

4. Design Decisions, Risks and Assumptions

4.1 Key Architecture Decisions

[^157]

4.1.1 Decision: Lambda Custom Identity Provider

Selected Approach: AWS Lambda-based custom identity provider for authentication processing

Rationale:

- **Serverless Architecture**: Eliminates infrastructure management, automatic scaling, pay-per-use pricing
- **Flexibility**: Custom Python code enables unlimited group processing and complex business logic
- **AWS Integration**: Native integration with DynamoDB, Secrets Manager, and CloudWatch
- **Proven Pattern**: AWS-documented solution with available SAM templates and best practices

Alternatives Considered:

- API Gateway Custom IdP: Additional complexity with no significant benefit for internal use case
- Native AD Integration: Cannot overcome single group limitation (the problem we're solving)
- EC2-Based SFTP Cluster: Higher operational complexity, cost, and maintenance overhead

Implications:

- Development effort required for Lambda function implementation
- VPC networking required for Windows AD connectivity
- Dependency on Lambda service limits (concurrent executions, timeout)
- Cold start latency considerations for infrequent authentications

4.1.2 Decision: Dual Storage Backend (S3 + EFS)

Selected Approach: Support both Amazon S3 and Amazon EFS with per-user assignment

Rationale:

- **Use Case Diversity**: S3 optimal for data lake/archival, EFS optimal for application data
- **POSIX Compliance**: EFS provides full POSIX file system semantics not available with S3
- **Cost Optimization**: S3 more cost-effective for infrequently accessed data with lifecycle policies
- **Future Flexibility**: Enables gradual migration between storage types based on requirements

Alternatives Considered:

- S3 Only: Cannot meet POSIX requirements for application integration use cases
- EFS Only: Higher cost for archival/backup data, no integration with S3-native AWS services

- S3 Gateway (s3fs on EC2): Adds complexity, performance limitations, not serverless

Implications:

- Two IAM roles required (one for S3 access, one for EFS access)
- Separate configuration logic in Lambda for storage backend determination
- Different home directory types: LOGICAL for S3, PATH for EFS
- POSIX profile required for EFS users (UID/GID management)

4.1.3 Decision: DynamoDB for Configuration Storage

Selected Approach: Two DynamoDB tables for user and IdP configuration

Rationale:

- **Serverless**: No database infrastructure to manage, automatic scaling
- **Low Latency**: Single-digit millisecond read latency for user lookups
- **High Availability**: Multi-AZ replication built-in with 99.99% SLA
- **Cost-Effective**: On-demand billing for unpredictable user authentication patterns

Alternatives Considered:

- RDS Database: Overkill for simple key-value lookups, higher cost, requires management
- S3 with JSON files: Higher latency, no indexing, consistency challenges
- Parameter Store/Secrets Manager: Not designed for user configuration data, limited query capabilities

Implications:

- Simple schema design limited to key-value pairs and JSON documents
- No complex queries or joins (LDAP provides group relationships)
- Eventual consistency for reads (acceptable for user configuration)
- DynamoDB Streams can enable automated user provisioning workflows (future enhancement)

4.1.4 Decision: VPC Integration for Lambda

Selected Approach: Deploy Lambda function in VPC with private subnets

Rationale:

- **Security**: Direct LDAP connectivity to Windows AD without exposing AD to internet
- **Network Isolation**: Lambda traffic contained within corporate network boundaries
- **Compliance**: Meets security requirements for sensitive authentication traffic
- **AD Integration**: Enables LDAP ports 389/636 connectivity to domain controllers

Alternatives Considered:

- Lambda without VPC: Cannot connect to private Windows AD infrastructure
- VPN Connection: Adds complexity, latency, single point of failure
- Direct Connect: Overkill for single Lambda function use case

Implications:

- NAT Gateway required for Lambda internet access (AWS API calls)
- Cold start latency increased by ~1-2 seconds for VPC ENI attachment
- Security group management for Lambda and AD connectivity
- VPC CIDR planning to avoid conflicts with existing networks

4.2 Assumptions

4.2.1 Infrastructure Assumptions

A-001: Windows AD Availability

- Windows EC2-based Active Directory is stable and highly available
- Domain controllers are accessible via LDAP on ports 389/636
- Service account with appropriate permissions can be created
- **Validation**: Perform LDAP connectivity test from Lambda subnet to AD during Phase 1

A-002: Network Connectivity

- VPC can be configured with private subnets and NAT Gateway
- Security groups can be modified to allow LDAP traffic from Lambda
- DNS resolution for AD domain names is functional within VPC
- **Validation**: Network connectivity tests during infrastructure setup phase

A-003: AWS Service Limits

- AWS Transfer Family server limit (50 per account) is sufficient
- Lambda concurrent execution limit (1000 per region) accommodates authentication load
- DynamoDB on-demand throughput limits (40,000 RCU) handle peak authentication requests
- **Validation**: Request service limit increases if needed during planning phase

4.2.2 Operational Assumptions

A-004: User Authentication Patterns

- Users authenticate once per SFTP session (not per file operation)
- Authentication frequency averages 10,000 per month baseline
- Peak authentication load does not exceed 100 concurrent per minute
- **Validation**: Monitor authentication metrics during first 30 days of production

A-005: User Configuration Management

- User records in DynamoDB will be managed via automated provisioning scripts
- User configuration changes are infrequent (weekly or monthly updates)
- No real-time synchronization required between AD and DynamoDB
- **Validation**: Establish user provisioning procedures during implementation

A-006: Storage Capacity

- Initial S3 storage requirement: 10 TB, growing at 2 TB per year
- Initial EFS storage requirement: 5 TB, growing at 1 TB per year
- Storage growth rates remain predictable for capacity planning
- **Validation**: Implement storage utilization monitoring and quarterly capacity reviews

4.2.3 Security Assumptions

A-007: Encryption Requirements

- TLS 1.2 or higher is acceptable for data in transit
- AWS KMS-managed keys meet encryption at rest requirements
- LDAPS (LDAP over SSL) is acceptable for AD authentication
- **Validation**: Security team review and approval of encryption standards

A-008: Access Control

- IAM role-based access control is sufficient for authorization
- No additional MFA required for SFTP user authentication (AD password only)
- IP allowlist restrictions managed in DynamoDB IdP configuration
- **Validation**: Security architecture review and penetration testing

A-009: Audit and Compliance

- CloudWatch Logs retention of 90 days meets audit requirements
- CloudTrail logging of API calls meets compliance standards
- No additional SIEM integration required initially
- **Validation**: Compliance team review of logging and retention policies

4.3 Risks

4.3.1 Technical Risks

R-001: Lambda Cold Start Latency

- **Risk**: Initial Lambda invocation (cold start) may take 3-5 seconds
- **Impact**: User authentication delay during first connection attempt

- **Probability**: Medium (occurs after periods of inactivity)
- **Severity**: Low (subsequent authentications are fast)
- **Mitigation**:
  - Configure provisioned concurrency for consistent performance during business hours
  - Implement CloudWatch scheduled event to warm Lambda function every 5 minutes
  - Set user expectation that first authentication may take longer
- **Contingency**: If unacceptable, provision 1 concurrent execution ($15/month)

R-002: LDAP Connectivity Failures

- **Risk**: Network issues or AD unavailability prevent LDAP authentication
- **Impact**: All SFTP users unable to authenticate during outage
- **Probability**: Low (assumes stable AD infrastructure)
- **Severity**: High (complete service disruption)
- **Mitigation**:
  - Configure Lambda with AD domain controllers in multiple AZs
  - Implement LDAP connection pooling and retry logic
  - Monitor LDAP connectivity with CloudWatch alarms
  - Establish automated alerting and escalation procedures
- **Contingency**: Manual failover to secondary AD domain controller if primary fails

R-003: Lambda Function Errors

- **Risk**: Bugs in Lambda code cause authentication failures
- **Impact**: Users unable to authenticate, potential data access issues
- **Probability**: Medium (custom code introduces defect potential)
- **Severity**: High (service disruption)
- **Mitigation**:
  - Comprehensive unit and integration testing before deployment
  - Lambda versioning with ability to rollback to previous version
  - Implement detailed error logging and tracing with AWS X-Ray
  - Conduct code reviews and static code analysis
- **Contingency**: Immediate rollback to previous Lambda version if issues detected

R-004: DynamoDB Performance Degradation

- **Risk**: High authentication load causes DynamoDB throttling
- **Impact**: Slow authentication response times or failures
- **Probability**: Low (on-demand mode auto-scales)
- **Severity**: Medium (temporary degradation, not complete outage)

- **Mitigation**:
  - Use on-demand billing mode for automatic scaling
  - Implement DynamoDB caching in Lambda for frequently accessed users
  - Monitor DynamoDB metrics and set throttling alarms
- **Contingency**: Increase provisioned capacity if on-demand throttling occurs

4.3.2 Security Risks

R-005: Service Account Credential Exposure

- **Risk**: AD service account credentials compromised via Secrets Manager
- **Impact**: Unauthorized access to AD infrastructure, potential privilege escalation
- **Probability**: Very Low (Secrets Manager encryption and access controls)
- **Severity**: Critical (AD security breach)
- **Mitigation**:
  - Store credentials in Secrets Manager with KMS encryption
  - Restrict IAM access to secret using least privilege principle
  - Enable Secrets Manager rotation policy (90 days)
  - Use AD service account with read-only permissions (no write access)
  - Monitor Secrets Manager access via CloudTrail
- **Contingency**: Immediate service account password change and secret rotation if breach suspected

R-006: Unauthorized Storage Access

- **Risk**: Session policy misconfiguration allows unauthorized file access
- **Impact**: Users accessing files outside their authorized directories
- **Probability**: Medium (configuration complexity)
- **Severity**: High (data breach, compliance violation)
- **Mitigation**:
  - Implement IAM policy review and validation during user configuration
  - Use ${transfer:UserName} variable for dynamic path restrictions
  - Enable S3 bucket versioning and CloudTrail logging for audit trail
  - Conduct regular access reviews and security audits
- **Contingency**: Immediately revoke access and investigate if unauthorized access detected

R-007: SFTP Protocol Vulnerabilities

- **Risk**: SSH/SFTP protocol vulnerabilities exploited by attackers
- **Impact**: Unauthorized access to SFTP server, potential data exfiltration

- **Probability**: Low (AWS manages SFTP server patching)
- **Severity**: Critical (complete compromise)
- **Mitigation**:
  - Use AWS Transfer Family security policy: TransferSecurityPolicy-2022-03 (latest)
  - Restrict SFTP access to authorized IP ranges via security groups
  - Enable CloudWatch logging for all SFTP sessions
  - Implement automated security monitoring and alerting
- **Contingency**: Isolate compromised server, rotate all credentials, conduct forensic investigation

4.3.3 Operational Risks

R-008: Insufficient Operations Team Expertise

- **Risk**: Operations team lacks experience with Lambda, DynamoDB, and LDAP integration
- **Impact**: Slow incident response, misconfiguration, operational errors
- **Probability**: Medium (new technology stack for team)
- **Severity**: Medium (operational inefficiency, potential downtime)
- **Mitigation**:
  - Provide comprehensive training during implementation phase
  - Document detailed operational procedures and runbooks
  - Establish mentorship with AWS Solutions Architect
  - Implement automated monitoring to reduce manual intervention
- **Contingency**: Engage AWS Professional Services for operational support if needed

R-009: User Provisioning Errors

- **Risk**: Incorrect user configuration in DynamoDB causes authentication failures
- **Impact**: Individual users unable to access SFTP, support tickets
- **Probability**: Medium (manual configuration errors)
- **Severity**: Low (limited to individual users)
- **Mitigation**:
  - Implement automated user provisioning scripts with validation
  - Establish user configuration review process before production
  - Create user provisioning templates for common scenarios
  - Implement DynamoDB validation Lambda trigger for configuration errors
- **Contingency**: Quick user configuration correction via DynamoDB console

R-010: Migration Disruption

- **Risk**: Migration from current Transfer Family server causes user disruption

- **Impact**: Temporary SFTP access loss during cutover
- **Probability**: Medium (inherent in any migration)
- **Severity**: Medium (business disruption)
- **Mitigation**:
  - Phased migration approach with pilot users first
  - Maintain parallel systems during migration with quick rollback capability
  - Schedule migration during low-usage maintenance window
  - Provide clear user communication and change management
- **Contingency**: Rollback to previous Transfer Family server if critical issues arise

4.4 Constraints

4.4.1 Technical Constraints

C-001: Lambda Execution Time Limit

- Maximum Lambda execution time: 15 minutes
- LDAP authentication and group processing must complete within timeout
- Complex group hierarchies may approach timeout limit
- **Impact**: Very deep nested groups (>100 levels) may cause timeout

C-002: DynamoDB Item Size Limit

- Maximum item size: 400 KB
- User configuration with extensive home directory mappings constrained by limit
- IdP configuration must fit within single item
- **Impact**: Users with hundreds of directory mappings may require optimization

C-003: Transfer Family Service Limits

- Maximum 50 Transfer Family servers per AWS account
- Maximum 100 users per server (not applicable with Lambda custom IdP)
- Maximum file size: 5 GB per file (API), unlimited via SFTP client
- **Impact**: Multi-tenant architectures may require careful planning

C-004: VPC ENI Limits

- Lambda VPC integration creates Elastic Network Interfaces (ENIs)
- ENI creation takes 30-90 seconds during cold start
- VPC subnet IP address capacity must accommodate Lambda ENIs
- **Impact**: Cold start latency increased by ENI attachment time

4.4.2 Operational Constraints

C-005: Windows AD Dependency

- Solution requires existing Windows AD availability for authentication
- AD domain controller outages prevent all SFTP authentication
- AD group changes not automatically synchronized to DynamoDB
- **Impact**: AD availability is critical path for SFTP service

C-006: Manual User Configuration

- User records must be manually populated in DynamoDB
- No automatic synchronization between AD and DynamoDB user configuration
- User provisioning requires custom automation or manual process
- **Impact**: User onboarding time and administrative overhead

C-007: Storage Backend Migration

- Migrating user from S3 to EFS (or vice versa) requires data copy
- No built-in mechanism for transparent storage migration
- User's existing files must be manually moved between backends
- **Impact**: Storage backend changes are operationally complex

4.4.3 Security Constraints

C-008: LDAP Protocol Limitations

- LDAP authentication transmits credentials (even over SSL)
- No support for Kerberos or certificate-based authentication
- Service account password must be stored in Secrets Manager
- **Impact**: Cannot implement passwordless authentication methods

C-009: IAM Role Limitation

- Single IAM role assigned per user session
- Cannot dynamically change role during active session
- Role permissions fixed at authentication time
- **Impact**: Permission changes require user re-authentication

C-010: Storage Encryption

- S3 and EFS encryption cannot be enabled retroactively
- Existing unencrypted data requires migration to encrypted storage
- KMS keys must be available in same region as storage
- **Impact**: Encryption must be enabled during initial setup

4.5 Dependencies

4.5.1 Internal Dependencies

D-001: Windows EC2 Active Directory

- **Description**: Existing Windows AD infrastructure for user authentication
- **Criticality**: Critical (authentication dependency)
- **Owner**: Windows Infrastructure Team
- **Requirements**:
  - Stable, highly available AD domain controllers
  - Service account with read permissions for LDAP queries
  - Network connectivity from Lambda VPC to AD on ports 389/636

D-002: VPC and Networking Infrastructure

- **Description**: AWS VPC with private subnets, NAT Gateway, security groups
- **Criticality**: Critical (Lambda VPC integration dependency)
- **Owner**: Network Engineering Team
- **Requirements**:
  - VPC with sufficient IP address space for Lambda ENIs
  - NAT Gateway for Lambda internet access
  - Security group rules permitting LDAP traffic

D-003: S3 Buckets and EFS File Systems

- **Description**: Storage backends for SFTP user file storage
- **Criticality**: Critical (data storage dependency)
- **Owner**: Storage Team
- **Requirements**:
  - S3 bucket with appropriate lifecycle policies and encryption
  - EFS file system with performance mode and backup configuration
  - IAM policies for Transfer Family role access

4.5.2 External Dependencies

D-004: AWS Transfer Family Service

- **Description**: Managed SFTP service provided by AWS
- **Criticality**: Critical (core service)
- **Provider**: Amazon Web Services
- **SLA**: 99.9% availability

- **Impact of Outage**: Complete SFTP service unavailable

D-005: AWS Lambda Service

- **Description**: Serverless compute for custom identity provider
- **Criticality**: Critical (authentication processing)
- **Provider**: Amazon Web Services
- **SLA**: 99.95% availability
- **Impact of Outage**: Authentication failures, no new SFTP sessions

D-006: AWS DynamoDB Service

- **Description**: User and IdP configuration storage
- **Criticality**: High (user lookup dependency)
- **Provider**: Amazon Web Services
- **SLA**: 99.99% availability (on-demand)
- **Impact of Outage**: Authentication failures due to configuration unavailability

D-007: AWS Secrets Manager

- **Description**: Secure storage for AD service account credentials
- **Criticality**: High (credential storage)
- **Provider**: Amazon Web Services
- **SLA**: 99.95% availability
- **Impact of Outage**: Cannot retrieve AD credentials, authentication failures

4.5.3 Deployment Dependencies

D-008: AWS SAM CLI

- **Description**: Serverless Application Model CLI for Lambda deployment
- **Criticality**: Medium (deployment tooling)
- **Version**: 1.50.0 or higher
- **Installation**: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html

D-009: Python 3.9 Runtime

- **Description**: Python runtime for Lambda function development
- **Criticality**: High (Lambda runtime)
- **Version**: Python 3.9
- **Dependencies**: boto3 (AWS SDK), python-ldap (LDAP client), cryptography

D-010: Infrastructure as Code (Terraform/CloudFormation)

- **Description**: IaC tools for automated infrastructure deployment

- **Criticality**: Medium (deployment automation)
- **Tools**: AWS CloudFormation, Terraform, or AWS CDK
- **Purpose**: Repeatable, version-controlled infrastructure deployment

5. Appendix

5.1 Glossary

AWS Transfer Family: Fully managed service providing SFTP, FTPS, and FTP access to Amazon S3 or EFS

Lambda Custom Identity Provider: Authentication mechanism using AWS Lambda function for custom logic

LDAP: Lightweight Directory Access Protocol for accessing directory services like Active Directory

LDAPS: LDAP over SSL/TLS for encrypted communication with directory services

DynamoDB: AWS fully managed NoSQL database service with single-digit millisecond latency

IAM: AWS Identity and Access Management for controlling access to AWS resources

VPC: Amazon Virtual Private Cloud for isolated network environment

EFS: Amazon Elastic File System providing scalable POSIX-compliant file storage

S3: Amazon Simple Storage Service for object storage

CloudWatch: AWS monitoring and observability service for logs, metrics, and alarms

NAT Gateway: Network Address Translation service enabling private subnet internet access

KMS: AWS Key Management Service for creating and managing encryption keys

POSIX: Portable Operating System Interface standard for file system operations

5.2 References

AWS Documentation:

1. AWS Transfer Family User Guide: https://docs.aws.amazon.com/transfer/latest/userguide/
2. Lambda Custom Identity Provider: https://docs.aws.amazon.com/transfer/latest/userguide/custom-identity-provider-users.html
3. AWS Lambda VPC Networking: https://docs.aws.amazon.com/lambda/latest/dg/configuration-vpc.html
4. DynamoDB Developer Guide: https://docs.aws.amazon.com/dynamodb/latest/developerguide/

AWS Blog Posts:

1. "Simplify Active Directory authentication with a custom identity provider for AWS Transfer Family" (August 2024)
2. "Enable password authentication for AWS Transfer Family using AWS Secrets Manager" (November 2020)

Implementation Resources:

1. AWS SAM Template Repository: https://github.com/aws-samples/aws-transfer-family-portal
2. Python LDAP Documentation: https://www.python-ldap.org/
3. AWS Transfer Family Workshop: https://catalog.workshops.aws/transfer-family

5.3 Revision History

Version	Date	Author	Changes
1.0	November 12, 2025	Infrastructure Team	Initial infrastructure design document

5.4 Document Approval

Role	Name	Signature	Date
Solutions Architect			
Security Architect			
Cloud Engineering Manager			
IT Director			

End of Infrastructure Design Document

⁂