Backflipt

TransferIQ Deployment Guide

September 2025

Introduction

This page provides detailed instructions on subscribing to the Backflipt TransferIQ solution via the AWS Marketplace and installing it within your native AWS account. To ensure a seamless deployment and optimal performance, please review and complete the mandatory pre-installation requirements outlined in this guide.

TransferIQ Orchestrate Use Cases

Insurance and Financial Industries

Overview

TransferIQ Orchestrate streamlines partner management and secure file routing in highly regulated environments. Specifically, it helps financial institutions and insurance firms:

- Automate partner onboarding and establish file transfer routes.
- Maintain rigorous compliance with data protection regulations.
- Enhance audit readiness with full visibility and logging.

Key Features

- Streamlined Onboarding & Routing Workflow
  - Facilitates collaboration between business teams and MFT administrators—handling partner onboarding, route setup, and updates through a governed, role-based workflow.
- Status Visibility & Request Tracking
  - Offers dashboards with real-time visibility into request progress, reducing manual back-and-forth and improving coordination across teams.
- Robust Audit Trails
  - Every request and change is logged, supporting compliance requirements with precise audit trail coverage.
- Role-Based Access Control & Approvals
  - Ensures that only authorized users can make or approve changes—critical for maintaining separation of duties and security standards.

Summary

TransferIQ Orchestrate delivers a secure, compliant, and efficient solution for managing partner relationships and file transfers in the financial and insurance sectors. It addresses regulatory demands, streamlines processes, and reinforces audit and security requirements through AI-enhanced workflows and visibility.

Transportation Industries

Overview

- The transportation industry must orchestrate secure, efficient data exchanges among carriers, terminals, logistics providers, and customers—often across multimodal networks with varied systems and data formats.
- Regulatory compliance is essential, encompassing TSA, DOT, Customs, and other international and domestic mandates governing security, documentation, and trade.
- Real-time supply chain visibility and operational efficiency are vital to avoid delays, maintain service quality, and meet customer expectations.

Key Features

- Streamlines partner onboarding: Simplifies the process of bringing new carriers, terminals, or logistics providers online, moving from manual to automated workflows.
- Enhances visibility: Offers real-time dashboards and request status tracking for transportation teams, improving coordination and reducing confusion.
- Ensures audit readiness: Automatically logs all requests and changes for complete audit trails, supporting compliance with stringent regulations.
- Accelerates routing: Allows faster configuration of file routes for new shipping lanes or services—key where timing is critical.
- Maintains separation of duties: Keeps operational workflows and administrative controls distinct, reinforcing internal governance.

Summary

In transportation industry, TransferIQ Orchestrate bolsters the exchange of sensitive operational data by:

- Automating complex partner onboarding across multimodal supply chains.
- Offering clear, actionable visibility into transfer workflows.
- Ensuring regulatory-compliant, audited exchanges.
- Keeping operations lean, secure, and responsive.

Healthcare Industry

Overview

- Healthcare organizations must handle highly sensitive patient data across a sprawling network of partners—providers, payers, labs, pharmacies, EHR vendors, telehealth platforms, and more—while complying with privacy regulations like HIPAA, HITECH, and the 21st Century Cures Act.
- Protecting PHI is critical, and breaches are costly.
- Complexity stems from the sheer number of partners, variance in systems, and strict interoperability and compliance mandates.

Key Features

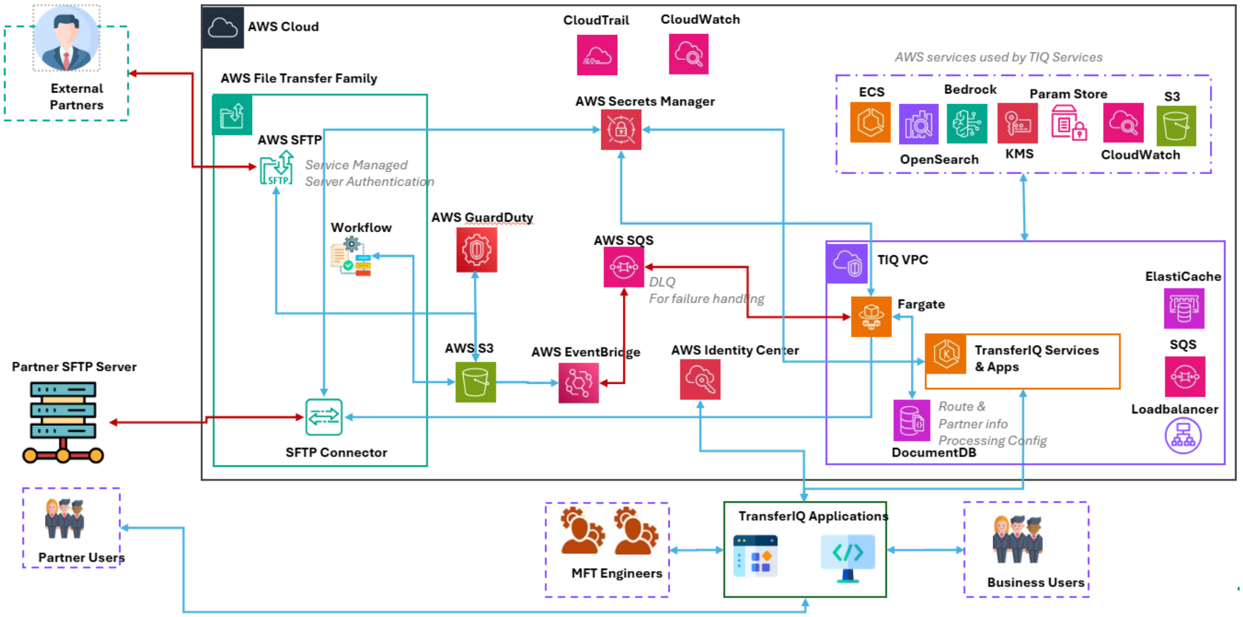
- Streamlined & Secure Partner Onboarding
  - Enables healthcare business users (e.g., from clinical, administrative teams) to submit onboarding and routing requests via intuitive, selfservice forms.
  - Workflows route these through approvals and finally to MFT admins who enact changes—ensuring rapid, visible progress with minimal manual effort.
- Robust Compliance & Audit Trails
  - Every request and action is logged, producing HIPAAcompliant audit trails that support internal review and external regulation.
- Clear Visibility Across Departments
  - Healthcare teams across clinical and administrative functions gain realtime visibility into the status of requests (onboarding, routing), promoting transparency and coordination.
- Support for Multiple MFT Administrators
  - Enables multiple MFT admins—aligned to different departments—to manage tasks without conflict, enhancing both operational flexibility and control.

Summary

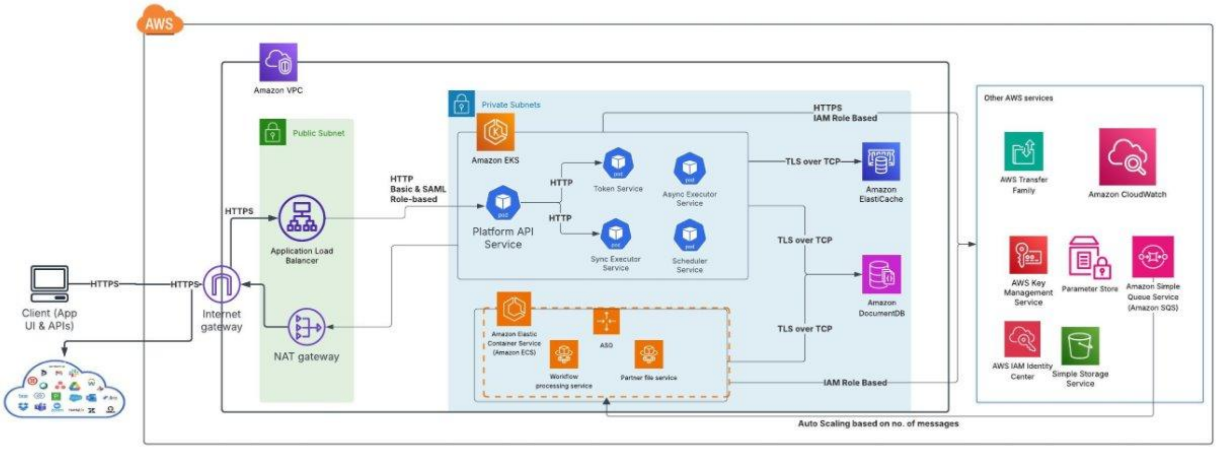
TransferIQ Orchestrate empowers healthcare organizations to onboard and manage partner file transfers both securely and efficiently, without sacrificing compliance or visibility. It streamlines collaboration between clinical and IT teams, embeds strong audit and governance controls, and supports scalable operations across departments.

Architecture Diagrams

Backflipt's TransferIQ Solution Architecture



Network Architecture



Deployment Pre-requisites

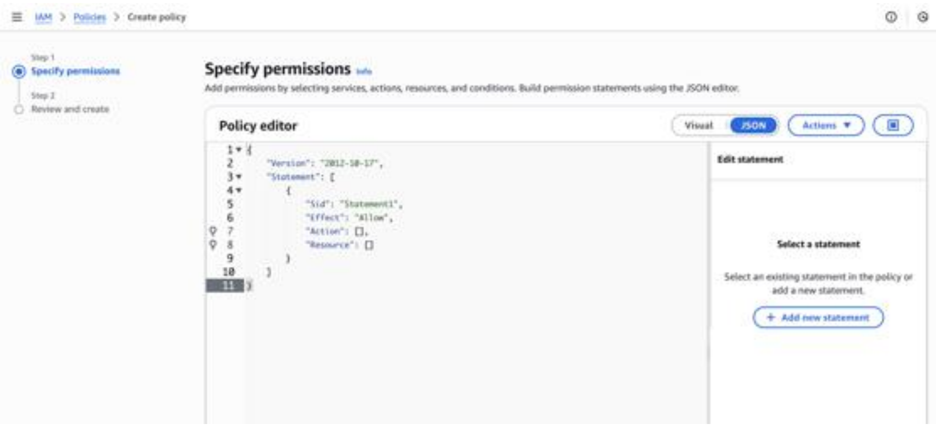
Make sure the User has sufficient permissions for performing actions related to the following services:

- IAM Roles
- IAM Policies
- AWS Certificate Manager (ACM)
- Marketplace
- CloudFormation Stack
- EC2 Load balancers (view only)

Step 1: Create an IAM Role and Policy:

To install the TransferIQ Solution, it is recommended to create an IAM role with enough permissions to create and maintain the resources in the CloudFormation stack, which contains all the resources required for the solution.

1. Navigate to the Policies section under IAM service in AWS Management Console.
2. Click on Create Policy and select JSON.



Specify permissions

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [],
            "Resource": []
        }
    ]
}
```

Select an existing statement in the policy or add a new statement.

3. Paste the following policy.

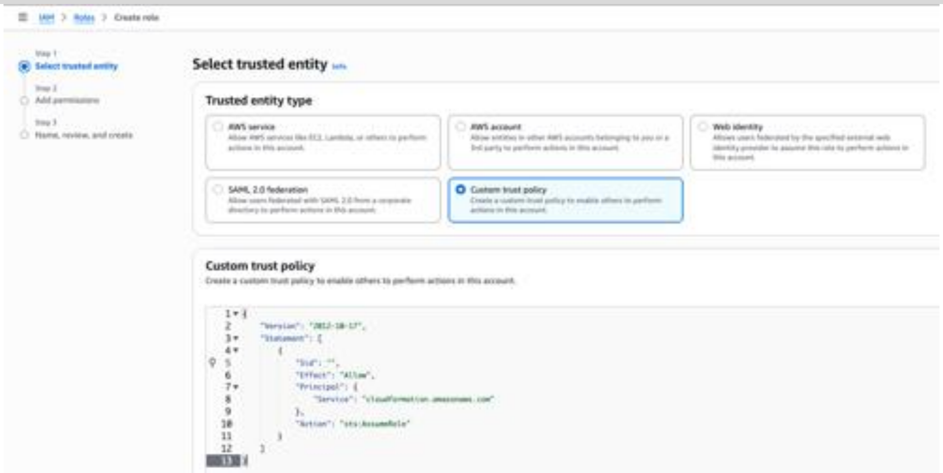
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2NetworkingAndComputeAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateRouteTable",
        "ec2:AttachInternetGateway",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:CreateInternetGateway",
        "ec2:CreateNatGateway",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSubnet",
        "ec2:CreateTags",
        "ec2:CreateVpc",
        "ec2:CreateVpcEndpoint",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:DeleteInternetGateway",
        "ec2:DeleteNatGateway",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSubnet",
        "ec2:DeleteTags",
        "ec2:DeleteVpc",
        "ec2:DeleteVpcEndpoints",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeNatGateways",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:ModifySubnetAttribute",
        "ec2:ModifyVpcAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupIngress",
      "ec2:RevokeSecurityGroupEgress",
      "ec2:RunInstances",
      "ec2:TerminateInstances",
      "ec2:DisassociateRouteTable",
      "ec2:DetachInternetGateway",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DeleteSecurityGroup",
      "ec2:DetachNetworkInterface"
    ],
    "Resource": "*"
  },
  {
    "Sid": "RDSProvisioningAccess",
    "Effect": "Allow",
    "Action": [
      "rds:AddTagsToResource",
      "rds:CreateDBCluster",
      "rds:CreateDBInstance",
      "rds:CreateDBSubnetGroup",
      "rds:DescribeDBClusters",
      "rds:DescribeDBInstances",
      "rds:DescribeDBSubnetGroups",
      "rds:ListTagsForResource",
      "rds:DeleteDBInstance",
      "rds:DeleteDBCluster",
      "rds:DeleteDBSubnetGroup",
      "rds:RemoveTagsFromResource",
      "rds:ModifyDBCluster",
      "rds:ModifyDBInstance"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ElastiCacheProvisioningAccess",
    "Effect": "Allow",
    "Action": [
      "elasticache:AddTagsToResource",
      "elasticache:CreateCacheCluster",
      "elasticache:CreateCacheSubnetGroup",
      "elasticache:DeleteCacheCluster",
      "elasticache:DeleteCacheSubnetGroup",
      "elasticache:DescribeCacheClusters",
      "elasticache:DescribeCacheSubnetGroups",
      "elasticache:ListTagsForResource",
      "elasticache:ModifyCacheCluster",
      "elasticache:RemoveTagsFromResource",
      "elasticache:CreateReplicationGroup",
      "elasticache:DescribeReplicationGroups",
      "elasticache:DeleteReplicationGroup"
    ],
    "Resource": "*"
  },
  {
    "Sid": "S3BucketManagementAccess",
    "Effect": "Allow",
    "Action": [
      "s3:CreateBucket",
      "s3:DeleteBucket",
      "s3:DeleteBucketPolicy",
      "s3:GetBucketPolicy",
      "s3:GetBucketPublicAccessBlock",
      "s3:GetBucketTagging",
      "s3:ListBucket",
      "s3:PutBucketPolicy",
      "s3:PutBucketPublicAccessBlock",
      "s3:PutBucketTagging",
      "s3:TagResource",
      "s3:UntagResource"
    ],
    "Resource": [
      "arn:aws:s3:::*"
    ]
  },
  {
    "Sid": "SQSFullAccess",
    "Effect": "Allow",
    "Action": [
      "sqs:CreateQueue",
      "sqs:DeleteMessage",
      "sqs:DeleteQueue",
      "sqs:GetQueueAttributes",
      "sqs:ListQueues",
      "sqs:PurgeQueue",
      "sqs:SendMessage",
      "sqs:SetQueueAttributes",
      "sqs:TagQueue",
      "sqs:UntagQueue"
    ],
    "Resource": "*"
  },
  {
    "Sid": "IAMRoleAndProfileManagement",
    "Effect": "Allow",
    "Action": [
      "iam:AddRoleToInstanceProfile",
      "iam:AttachRolePolicy",
      "iam:CreateInstanceProfile",
      "iam:CreateRole",
      "iam:DeleteRole",
      "iam:DeleteRolePolicy",
      "iam:GetInstanceProfile",
      "iam:GetRole",
      "iam:ListAttachedRolePolicies",
      "iam:PassRole",
      "iam:PutRolePolicy",
      "iam:UpdateAssumeRolePolicy",
      "iam:CreateServiceLinkedRole",
      "iam:DeleteRole",
      "iam:DeleteRolePolicy",
      "iam:RemoveRoleFromInstanceProfile",
      "iam:DeleteInstanceProfile",
      "iam:ListInstanceProfilesForRole",
      "iam:DetachRolePolicy",
      "iam:ListRolePolicies",
      "iam:ListAttachedRolePolicies",
      "iam:GetRolePolicy",
      "iam:TagUser",
      "iam:TagRole",
      "iam:TagPolicy",
      "iam:UntagUser",
      "iam:UntagRole",
      "iam:UntagPolicy"
    ],
    "Resource": "*"
  },
  {
    "Sid": "EKSClusterAndNodegroupAccess",
    "Effect": "Allow",
    "Action": [
      "eks:AssociateAccessPolicy",
      "eks:CreateAccessEntry",
      "eks:CreateAddon",
      "eks:UpdateAddon",
      "eks:DescribeAddon",
      "eks:CreateCluster",
      "eks:CreateNodegroup",
      "eks:DeleteAccessEntry",
      "eks:DeleteNodegroup",
      "eks:DescribeAccessEntry",
      "eks:DescribeAddon",
      "eks:DescribeCluster",
      "eks:DescribeNodegroup",
      "eks:ListAccessEntries",
      "eks:ListAddons",
      "eks:ListAssociatedAccessPolicies",
      "eks:ListClusters",
      "eks:ListNodegroups",
      "eks:DeleteAddon",
      "eks:DeleteCluster",
      "eks:TagResource",
      "eks:UntagResource"
    ],
    "Resource": "*"
  },
  {
    "Sid": "KMSKeyAndEncryptionAccess",
    "Effect": "Allow",
    "Action": [
      "kms:CreateAlias",
      "kms:CreateKey",
      "kms:Decrypt",
      "kms:DescribeKey",
      "kms:Encrypt",
      "kms:ListAliases",
      "kms:ListGrants",
      "kms:PutKeyPolicy",
      "kms:TagResource",
      "kms:UntagResource",
      "kms:DeleteAlias",
      "kms:ScheduleKeyDeletion",
      "kms:ListResourceTags"
    ],
    "Resource": "*"
  },
  {
    "Sid": "SSMParameterDocumentAssociationAccess",
    "Effect": "Allow",
    "Action": [
      "ssm:AddTagsToResource",
      "ssm:CreateAssociation",
      "ssm:CreateDocument",
      "ssm:DeleteAssociation",
      "ssm:DeleteDocument",
      "ssm:DeleteParameter",
      "ssm:DescribeAssociation",
      "ssm:DescribeDocument",
      "ssm:DescribeParameters",
      "ssm:GetDocument",
      "ssm:GetParameters",
      "ssm:PutParameter",
      "ssm:UpdateAssociation",
      "ssm:UpdateDocument",
      "ssm:RemoveTagsFromResource"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ResourceLifecycleManagement",
    "Effect": "Allow",
    "Action": [
      "rds:ModifyDBSubnetGroup",
      "iam:UntagRole",
      "tag:TagResources",
      "tag:UntagResources",
      "elasticache:AddTagsToResource",
      "elasticache:RemoveTagsFromResource",
      "kms:TagResource",
      "kms:UntagResource",
      "iam:TagRole",
      "rds:RemoveTagsFromResource"
    ],
    "Resource": [
      "*"
    ]
  }
]
}
```

4. Click on next, name it 'TiQ-mft-resources-policy', and create the policy.
5. Follow the same process to create another Policy with the name ‘TiQ-mft-awsservices-policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMRoleAndPolicyManagement",
      "Effect": "Allow",
      "Action": [
        "iam:CreatePolicy",
        "iam:DeletePolicy",
        "iam:GetPolicy",
        "iam:ListRoles",
        "iam:ListPolicies",
        "iam:TagRole",
        "iam:TagPolicy",
        "iam:PutUserPolicy",
        "iam:AttachUserPolicy",
        "iam:ListAttachedUserPolicies",
        "iam:DetachUserPolicy",
        "iam:ListPolicyVersions",
        "iam:GetUserPolicy",
        "iam:GetPolicyVersion",
        "iam:DeleteUserPolicy",
        "iam:UntagPolicy",
        "iam:CreatePolicyVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SecretsManagerAccess",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:UntagResource",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:CreateSecret",
        "secretsmanager:DeleteSecret",
        "secretsmanager:ListSecrets",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3BucketManagement",
      "Effect": "Allow",
      "Action": [
        "s3:PutBucketAcl",
        "s3:PutBucketVersioning",
        "s3:PutEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:PutBucketNotification",
        "s3:PutBucketLogging",
        "s3:TagResource",
        "s3:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "TransferFamilyAccess",
      "Effect": "Allow",
      "Action": [
        "transfer:CreateServer",
        "transfer:DeleteServer",
        "transfer:DescribeServer",
        "transfer:UpdateServer",
        "transfer:ListServers",
        "transfer:CreateConnector",
        "transfer:DeleteConnector",
        "transfer:DescribeConnector",
        "transfer:UpdateConnector",
        "transfer:ListConnectors",
        "transfer:CreateUser",
        "transfer:DeleteUser",
        "transfer:UpdateUser",
        "transfer:DescribeUser",
        "transfer:ListUsers",
        "transfer:TagResource",
        "transfer:UntagResource",
        "transfer:ListTagsForResource",
        "transfer:ImportSshPublicKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "LambdaCreationAccess",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunction",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration",
        "lambda:TagResource",
        "lambda:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EventBridgeAccess",
      "Effect": "Allow",
      "Action": [
        "events:CreateEventBus",
        "events:DeleteEventBus",
        "events:DescribeEventBus",
        "events:ListEventBuses",
        "events:DescribeRule",
        "events:PutRule",
        "events:EnableRule",
        "events:DisableRule",
        "events:DeleteRule",
        "events:ListRules",
        "events:ListTargetsByRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:TagResource",
        "events:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchAccess",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups",
        "logs:DeleteLogGroup",
        "logs:TagLogGroup",
        "logs:ListLogGroupsForQuery",
        "logs:PutRetentionPolicy",
        "logs:TagResource",
        "logs:UntagLogGroup",
        "logs:UntagResource",
        "logs:ListTagsForResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ECSAccess",
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:DeleteCluster",
        "ecs:RegisterTaskDefinition",
        "ecs:DeregisterTaskDefinition",
        "ecs:DescribeClusters",
        "ecs:DescribeServices",
        "ecs:CreateService",
        "ecs:UpdateService",
        "ecs:DeleteService",
        "ecs:ListClusters",
        "ecs:ListServices",
        "ecs:ListTaskDefinitions",
        "ecs:TagResource",
        "ecs:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2NetworkInterface",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteNetworkInterface",
        "ec2:CreateNetworkInterface",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:CreateTags",
        "ec2:DeleteTags"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ELBAccess",
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:CreateListener",
        "elasticloadbalancing:CreateTargetGroup",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:ModifyTargetGroup",
        "elasticloadbalancing:DeleteLoadBalancer",
        "elasticloadbalancing:DeleteListener",
        "elasticloadbalancing:DeleteTargetGroup",
        "elasticloadbalancing:Describe*",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:ModifyLoadBalancerAttributes",
        "elasticloadbalancing:RemoveTags"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ECSScalingPolicies",
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:*"
    ],
    "Resource": "*"
  },
  {
    "Sid": "CloudwatchAlarms",
    "Effect": "Allow",
    "Action": [
      "cloudwatch:*"
    ],
    "Resource": "*"
  },
  {
    "Sid": "GuardDuty",
    "Effect": "Allow",
    "Action": [
      "guardduty:CreateMalwareProtectionPlan",
      "guardduty:TagResource",
      "guardduty:GetMalwareProtectionPlan",
      "guardduty:ListTagsForResource",
      "guardduty:UpdateMalwareProtectionPlan",
      "guardduty:UntagResource",
      "guardduty:DeleteMalwareProtectionPlan"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Sid": "CloudFormationFullAccess",
    "Effect": "Allow",
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation:DescribeStacks",
      "cloudformation:ListStacks",
      "cloudformation:GetTemplateSummary",
      "cloudformation:ValidateTemplate",
      "cloudformation:UpdateStack",
      "cloudformation:DeleteStack",
      "cloudformation:DescribeStackResources",
      "cloudformation:DescribeStackEvents",
      "cloudformation:DescribeStackResource",
      "cloudformation:ListStackResources",
          "cloudformation:ListExports",
          "cloudformation:RollbackStack",
          "cloudformation:CreateChangeSet",
          "cloudformation:TagResource",
          "cloudformation:UntagResource"
        ],
        "Resource": "*"
      }
    ]
}
```

6. Navigate to the Roles section under the IAM service in the AWS Management Console.
7. Click on Create role.
8. Select Custom trust Policy and Use the following Trust Policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



Select trusted entity

Trusted entity type

○ AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

● Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

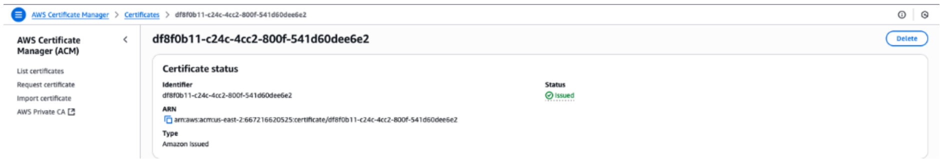
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudformation.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

9. Scroll to the bottom and Click on Next button, attach the previously created policies to it.
    - TiQ-mft-resources-policy
    - TiQ-mft-awsservices-policy
10. Name the role 'TiQ-cfs-assume-role' and create the role.

Step 2: DNS names and ACM certificate:

To secure network traffic for users connecting to the Backflipt TransferIQ application, an SSL certificate associated with the fully qualified domain name (FQDN) should either be imported or created in the AWS Certificate Manager (ACM). This certificate’s Amazon Resource Name (ARN) and the FQDN must then be provided as inputs while installing the solution through the CloudFormation stack.


AWS Certificate Manager > Certificates > df8f0b11-c24c-4cc2-800f-541d60dee6e2
AWS Certificate Manager (ACM)
List certificates
Request certificate
Import certificate
AWS Private CA
Delete

df8f0b11-c24c-4cc2-800f-541d60dee6e2

Certificate status

Identifier
df8f0b11-c24c-4cc2-800f-541d60dee6e2

Status
Issued

ARN
arn:aws:acm:us-east-2:667216620525:certificate/df8f0b11-c24c-4cc2-800f-541d60dee6e2

Type
Amazon Issued

Note: Ensure that both the DNS name and the SSL certificate are valid, as the DNS URL will be used throughout the Post Installation steps.

Technical Requirements

This product is a **containerized** microservices application on Amazon EKS across two Availability Zones with Amazon DocumentDB for persistence and supporting services (SQS, S3, KMS, AWS Transfer Family). The requirements below reflect exactly what is needed to deploy and operate it.

At-a-Glance

Area	Requirement
AWS Account	Active account with billing; access to 2+ AZs in the chosen Region
Supported Regions	us-east-1, us-east-2, ap-south-1
IAM	Ability to create/update IAM roles, policies, and accept AWS Marketplace terms
Networking	One VPC with 2 public and 2 private subnets (one pair per AZ); Internet/NAT gateways; security groups
EKS (Kubernetes)	Versions supported: 1.31, 1.32, 1.33; Managed Node Group: m5.xlarge, min 3 / desired 4 / max 5
Database	Amazon DocumentDB; [Unverified] instance class provided as “cache.t3.medium” (this looks like an ElastiCache class). Please confirm the DocumentDB class (e.g., db.t3.medium, db.r5.large, etc.); backup/retention: none (per your input)
Messaging	Amazon SQS queues created by the stacks (no pre-creation)
Caching	Amazon Elasticache for Redis created by stacks (no pre-creation)
Object Storage	Amazon S3 buckets created/managed by the stacks (no pre-creation)
Key Management	AWS KMS used for encryption (keys/policies managed by the stacks)
Security & scanning	AWS Guardduty created by stacks (no pre-creation)
Transfer	AWS Transfer Family used by the solution; provisioned/configured by the stacks
TLS	ACM certificate for the application FQDN(s) if public ingress is required

Operator Skills	Basic AWS Console and basic computer knowledge
Workstation	Modern web browser for AWS Console access (CLI/tools optional if you follow console steps)

Details

1. *Compute & Orchestration (EKS*

- Cluster: Multi-AZ across two AZs within the selected Region.
- Kubernetes versions: 1.31 / 1.32 / 1.33.
- Nodes: Managed Node Group, instance type m5.xlarge, autoscaling min 3 / desired 4 / max 5.
- Ingress/TLS: Public or private load balancer as provisioned; ACM certificate required for HTTPS endpoints.

2. *Database (Amazon DocumentDB)*

- Engine: Amazon DocumentDB.
- Instances: Clustered across private subnets.
- Instance class: db.t3.medium
- Backups/retention: None.

3. *Messaging (Amazon SQS)*

- Queues: Provisioned by the deployment stacks; includes any DLQs required by the solution.
- No operator pre-creation required.

4. *Object Storage (Amazon S3)*

- Buckets: Provisioned/used by the stacks; no pre-creation required.

5. *Key Management (AWS KMS)*

- Encryption: Keys and policies as provisioned by the stacks for at-rest encryption of applicable services.

6. *AWS Transfer Family*

- Usage: Managed file transfer as part of the solution; created/configured by the stacks.

7. *AWS ElastiCache Redis*

- Engine: Redis 7.1
- Node type: cache.t2.micro
- Usage: To increase the application performance; created/configured by the stacks.

8. *Networking & DNS*

- VPC: One VPC with two public and two private subnets (one pair per AZ).
- Gateways: Internet Gateway and NAT Gateways for egress from private subnets.
- DNS/TLS: Public DNS record(s) mapped to the provisioned load balancer; ACM certificate for TLS.

Skills & Specialized Knowledge

Intended Operator Profile

Operators with basic AWS knowledge and basic computer skills can deploy and use the product by following the step-by-step guide.

Core Skills Required for Deployment

- AWS Console navigation: Select Region, open service consoles, follow CloudFormation stack events, and read stack outputs.
- AWS Marketplace: Subscribe to the product and initiate deployment.
- AWS CloudFormation: Launch a stack, supply parameters, acknowledge capability prompts, and monitor until CREATE_COMPLETE.
- IAM (console-level): Create/choose the deployment role, understand role assumption for installation.
- DNS & TLS (ACM): Request/import a certificate for the application FQDN, and create the DNS record that points to the provisioned Load Balancer.
- VPC & networking (basic): Understand public vs. private subnets and security groups sufficiently to provide/confirm parameters.
- Basic command line: Run the provided curl command(s) and replace placeholders in the JSON payload for post-install configuration.

Service-Specific Familiarity (console-level)

Service	Skill expected	Why it’s needed during deployment
EKS	High-level awareness (no kubectl required)	Confirm ingress endpoint and service health from console after stack creation.
ECS	Console overview of Services/Tasks and Auto Scaling policies	The stack configures ECS to scale workers based on SQS depth; operator should verify service is healthy.
SQS	Find queue ARNs and view approximate message count	Provide/confirm ARNs in post-install steps and verify processing/backlog.
DocumentDB	Provide master credentials; basic connectivity awareness	Supply parameters at launch and confirm cluster availability post-deploy.
S3	Console-level familiarity with buckets/folders	Select or verify bucket paths referenced by the application configuration.
ElastiCache	Console-level verification	Confirm cache cluster/parameter group status (created by the stack).
AWS Transfer Family	Conceptual understanding	Validate endpoint status if used by the workflow.
Secrets Manager	Locate and view (no secret creation during install)	Review application secrets/rotation settings created by the stack.
EventBridge	Console overview of rules/targets	Verify rules created by the stack are enabled and targeting the right services.
KMS	Conceptual understanding of key usage	Acknowledge/enforce encryption settings managed by the stack.

GuardDuty	Console overview (enablement/visibility)	Confirm GuardDuty is active in the account/Region per your security posture.
CloudWatch	View logs, metrics, and basic alarms	Check component health and troubleshoot initial start-up if needed.

Access & Permissions

- Ability to assume an IAM role with permissions to create and manage the resources included in the stack (EKS, ECS, EC2/VPC/ELB, IAM, ACM, S3, SQS, DocumentDB, ElastiCache, Transfer Family, Secrets Manager, EventBridge, KMS, and GuardDuty visibility).

Note: Because the CloudFormation stack creates all required resources, the operator need only have console-level familiarity with the services above to complete deployment and perform first-run checks.

Environment Configuration Requirements

This section defines the baseline environment required to deploy and run the product via the provided AWS CloudFormation stack. All core resources (EKS, ECS workers for SQS-driven jobs, DocumentDB, SQS, S3, ElastiCache, AWS Transfer Family, Secrets Manager, EventBridge, KMS, ALB, and GuardDuty configuration) are created by the stack; no manual pre-provisioning is required beyond DNS/TLS and an IAM role to execute the deployment.

1) AWS Account & Regions

- AWS account with billing enabled and permissions to create resources via CloudFormation.
- Supported Regions: us-east-1, us-east-2, ap-south-1.
- Availability Zones: At least two AZs available in the chosen Region (for multi-AZ deployment).
- Marketplace subscription: Accept the product's AWS Marketplace terms before launching the stack.
- Service quotas: Capacity to create EKS clusters/node groups, load balancers, NAT gateways/ENIs, SQS queues, DocumentDB instances, ElastiCache nodes, and KMS keys as provisioned by the template.

2) Identity, Access & Security Baseline

- Deployment IAM role with permissions to provision the services used by the stack (EKS, ECS/Auto Scaling, EC2/VPC/ALB, IAM, ACM, S3, SQS, DocumentDB, ElastiCache, Transfer Family, Secrets Manager, EventBridge, KMS, CloudWatch, GuardDuty visibility).
- KMS usage: Permission to create and use CMKs for at-rest encryption where applicable (keys are created/attached by the stack).
- GuardDuty: Account/Region visibility enabled to observe findings (stack aligns with GuardDuty; it does not replace your security operations).

3) Networking & DNS

- VPC with two public and two private subnets (one pair per AZ).
- Internet Gateway and NAT Gateway(s) so private subnets have outbound internet egress for image pulls/updates during deployment.
- Security groups permitting least-privilege communication between application components (configured by the stack).
- DNS domain you control (public hosted zone or external DNS) to publish the application FQDN.
- TLS certificate (ACM) in the same Region as the load balancer, validated for the chosen FQDN. You will map the FQDN to the stack-provisioned LB DNS name.

4) Operating System / Workstation

- Operator workstation: Any OS capable of running a modern web browser.
- Browser: Current Chrome/Edge/Firefox/Safari to use the AWS Console.
- Command line (optional): Ability to run curl to post the provided JSON payload for post-install configuration.
- Cluster node OS: Managed by the template (Amazon EKS managed node groups). I cannot verify a mandated AMI/OS image from your inputs.

5) Product Configuration Inputs (Provided at Launch)

- EKS versions supported: 1.31, 1.32, 1.33.
- Node group: m5.xlarge (autoscaling min 3 / desired 4 / max 5).
- DocumentDB: Cluster credentials (master username/password) supplied at launch; backups disabled per your input. *(Instance class will be finalized in your edited section.)*
- SQS: Queues created by the stack (no pre-creation).
- S3: Buckets created/used by the stack (no pre-creation).
- ElastiCache / ECS / Transfer Family / Secrets Manager / EventBridge / KMS: Created and configured by the stack; no manual setup required.
- Application DNS/TLS: Provide ACM certificate ARN and target FQDN to expose the application over HTTPS.

6) Licensing & Third-Party Dependencies

- AWS Marketplace: Subscription acceptance is required before deployment.

7) Environment Readiness Checklist

- Account in us-east-1 / us-east-2 / ap-south-1 with 2+ AZs available
- Marketplace subscription accepted
- Deployment IAM role created with required permissions
- VPC with 2 public + 2 private subnets, IGW and NAT in place
- Public ACM certificate issued in target Region for the application FQDN
- DNS control to create record pointing FQDN → stack LB DNS name
- Workstation with browser (and curl for post-install payload)
- Service quotas sufficient for resources the template will create

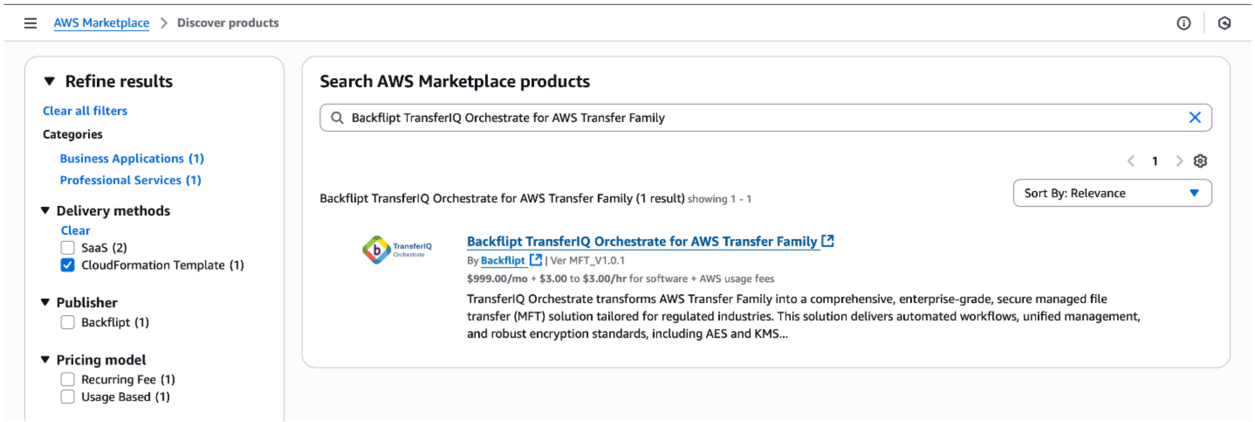
Product Subscription

Step 1: Sign in to the AWS Management Console and Navigate to Marketplace

1. Open the AWS Management Console and sign in with your credentials.
2. In the search bar at the top of the AWS Console, type AWS Marketplace.
3. Select AWS Marketplace from the list of services to open the Marketplace Dashboard.

Step 2: Subscribe to 'Backflipt TransferIQ for AWS Transfer Family'

1. Search for Backflipt TransferIQ for AWS Transfer Family in the Discover products panel.
2. Select the product and click View purchase options, which will redirect to another page.



Backflipt TransferIQ Orchestrate for AWS Transfer Family

Backflipt TransferIQ Orchestrate for AWS Transfer Family (1 result) showing 1 - 1

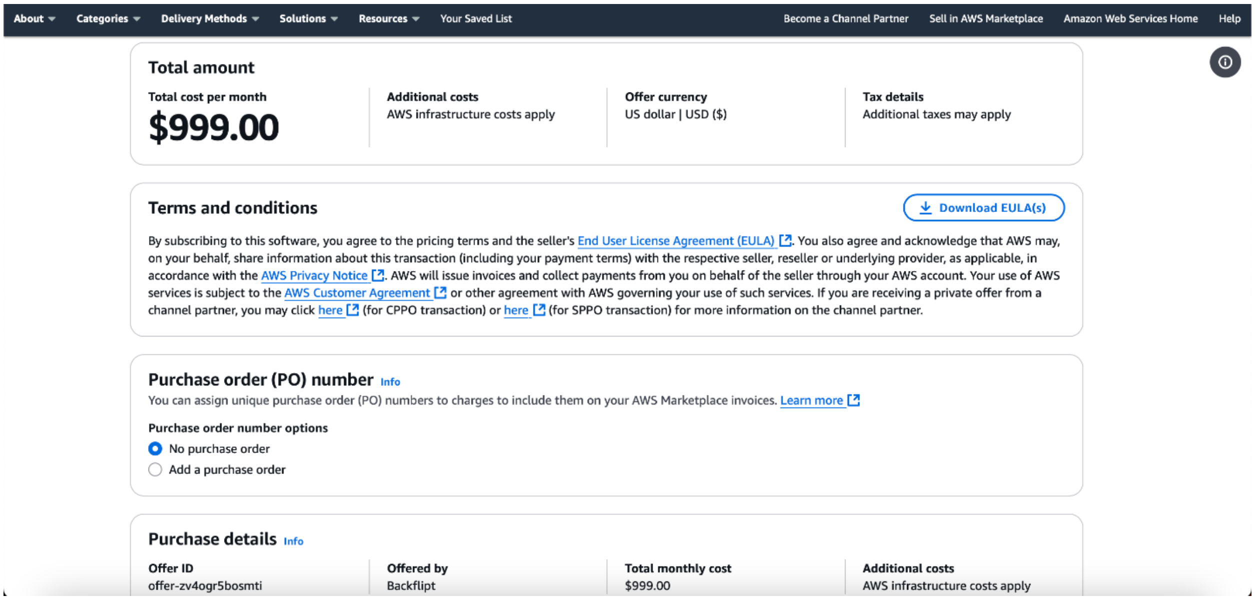
Backflipt TransferIQ Orchestrate for AWS Transfer Family

By Backflipt | Ver MFT_V1.0.1

$999.00/mo + $3.00 to $3.00/hr for software + AWS usage fees

TransferIQ Orchestrate transforms AWS Transfer Family into a comprehensive, enterprise-grade, secure managed file transfer (MFT) solution tailored for regulated industries. This solution delivers automated workflows, unified management, and robust encryption standards, including AES and KMS...

3. Scroll down to the bottom and click on Subscribe. The subscription process will take around 1-2 minutes.



Total amount

Total cost per month	Additional costs	Offer currency	Tax details
$999.00	AWS infrastructure costs apply	US dollar | USD ($)	Additional taxes may apply

Terms and conditions

By subscribing to this software, you agree to the pricing terms and the seller's End User License Agreement (EULA). You also agree and acknowledge that AWS may, on your behalf, share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the AWS Privacy Notice. AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the AWS Customer Agreement or other agreement with AWS governing your use of such services. If you are receiving a private offer from a channel partner, you may click here (for CPPO transaction) or here (for SPPO transaction) for more information on the channel partner.

Purchase order (PO) number Info

You can assign unique purchase order (PO) numbers to charges to include them on your AWS Marketplace invoices. Learn more

Purchase order number options

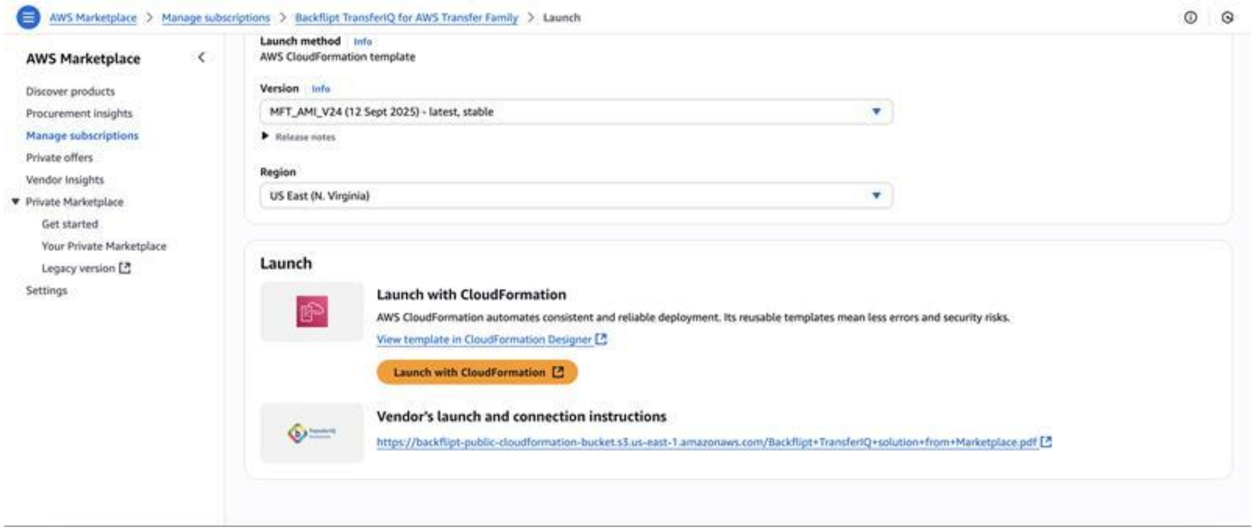
● No purchase order

○ Add a purchase order

Purchase details Info

Offer ID	Offered by	Total monthly cost	Additional costs
offer-zv4ogr5bosmti	Backflipt	$999.00	AWS infrastructure costs apply

4. Go through the Instructions by clicking the link available under Vendor’s launch and connection instructions.



Launch method Info

AWS CloudFormation template

Version Info

MFT_AMI_V24 (12 Sept 2025) - latest, stable

▶ Release notes

Region

US East (N. Virginia)

Launch

Launch with CloudFormation

AWS CloudFormation automates consistent and reliable deployment. Its reusable templates mean less errors and security risks.

View template in CloudFormation Designer

Vendor’s launch and connection instructions

https://backflipt-public-cloudformation-bucket.s3.us-east-1.amazonaws.com/Backflipt+TransferIQ+solution+from+Marketplace.pdf

TransferIQ Solution Deployment

List of AWS Resources

TransferIQ leverages a wide range of AWS services to ensure secure, scalable and highly available operations. When a typical customer deployment is complete, the following resources are provisioned and integrated to support the application's functionality, monitoring and security. Amazon Virtual Private Cloud (VPC)

- Amazon Virtual Private Cloud (VPC)
- Amazon EC2
- Amazon Simple Queue Service (SQS)
- Amazon Simple Storage Service (S3)
- AWS Systems Manager (SSM)
- Amazon CloudWatch
- Amazon GuardDuty
- AWS Identity and Access Management (IAM)
- AWS Transfer Family
- Amazon Elastic Container Service (ECS)
- AWS Secrets Manager
- Amazon EventBridge
- AWS Key Management Service (KMS)
- Amazon ElastiCache
- Amazon Kubernetes Service (EKS)
- Amazon DocumentDB
- IAM Identity Center
- Amazon Route 53
- AWS Certificate Manager (ACM)

Deployment Options Overview

TransferIQ supports flexible deployment on AWS, providing three main options, each with distinct characteristics concerning resiliency, availability, and cost:

1. Multi-AZ Deployment

Description: Core components are distributed across two AZs within the same AWS region.

Use Case: this approach is Recommended for production environments, as it provides automatic file transfer failover mechanisms should one AZ lose service.

Benefits: Enhances fault tolerance, as workloads continue running even if one AZ goes down.

How it works: AWS services such as DocumentDB, EKS pods, AWS Fargate Tasks are enabled with Auto Scaling based on the volume, and Elastic Load Balancers are deployed across 2 availability zones.

Material contains the regions supported

TransferIQ supports deployment in the following AWS regions, enabling customers to choose data residency that best fits their requirements and compliance needs. The selection of regions also supports compliance with local regulations. The regions available for hosting TransferIQ solutions include:

North America

- US East (Ohio)
- US East (N. Virginia)

Asia Pacific (APAC)

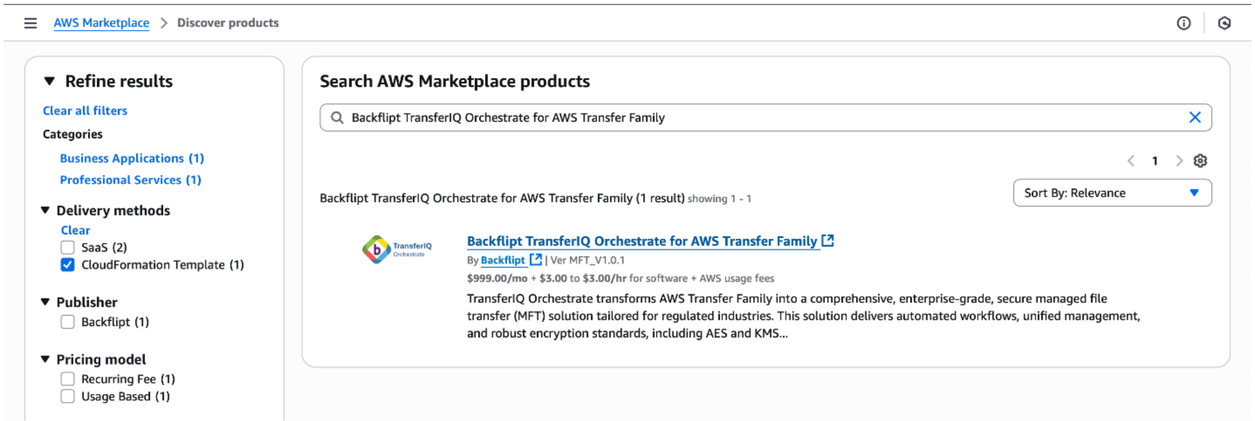
- Asia Pacific (Mumbai)

The actual regions supported may depend on the specifics of your AWS Marketplace listing and product configuration, so customers should consult with your cloud team and their AWS account manager for the most up-to-date and region-specific deployments.

Steps to Deploy Solution from MarketPlace:

Step 1: Launch Product from MarketPlace

1. Navigate to the Manage subscriptions section in AWS Marketplace in AWS Management Console.
2. Under Active subscriptions, search for ‘Backflipt TransferIQ Orchestrate for AWS Transfer Family’.



Backflipt TransferIQ Orchestrate for AWS Transfer Family

Backflipt TransferIQ Orchestrate for AWS Transfer Family (1 result) showing 1 - 1

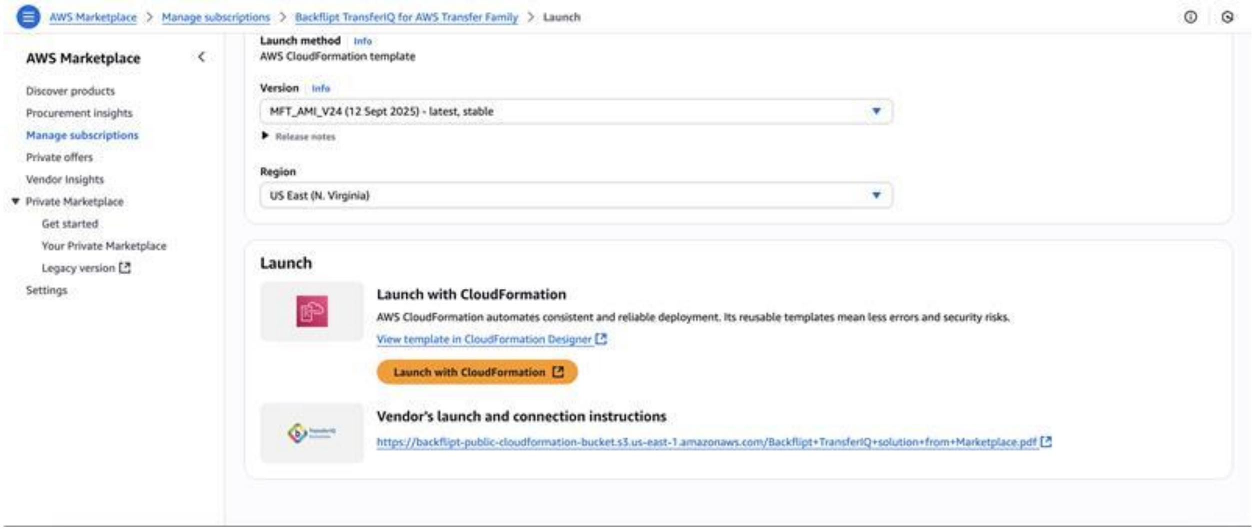
Backflipt TransferIQ Orchestrate for AWS Transfer Family

By Backflipt | Ver MFT_V1.0.1

$999.00/mo + $3.00 to $3.00/hr for software + AWS usage fees

TransferIQ Orchestrate transforms AWS Transfer Family into a comprehensive, enterprise-grade, secure managed file transfer (MFT) solution tailored for regulated industries. This solution delivers automated workflows, unified management, and robust encryption standards, including AES and KMS...

3. Click on the Launch under the Actions column.
4. Select the appropriate region and click on the Launch with CloudFormation button.



Launch method Info

AWS CloudFormation template

Version Info

MFT_AMI_V24 (12 Sept 2025) - latest, stable

Region

US East (N. Virginia)

Launch

Launch with CloudFormation

AWS CloudFormation automates consistent and reliable deployment. Its reusable templates mean less errors and security risks.

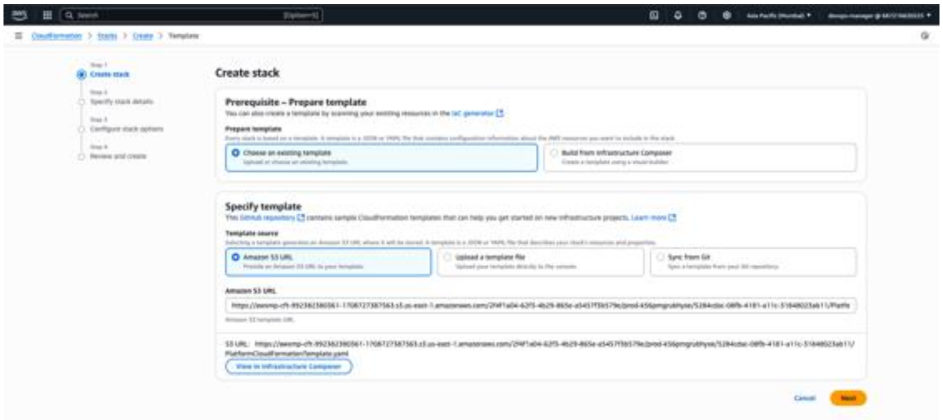
View template in CloudFormation Designer

Vendor's launch and connection instructions

https://backflipt-public-cloudformation-bucket.s3.us-east-1.amazonaws.com/Backflipt+TransferIQ+solution+from+Marketplace.pdf

Step 2: CloudFormation Stack

1. After clicking on Continue to Launch, the User will be redirected to the AWS management Console with the CloudFormation stack loaded in it.
    a. Select Choose an existing template option in Pre-requisites - Prepare Template Section
    b. Select Amazon S3 URL in Specify template section
    c. Click on the Next button at the bottom of the screen.



Create stack

Prerequisite – Prepare template

Prepare template

● Choose an existing template ○ Build from Infrastructure Composer

Specify template

Template source

● Amazon S3 URL ○ Upload a template file ○ Sync from Git

Amazon S3 URL

2. Stack Name: Enter a unique name as TIQ-Prod-Stack-20250815 for the stack
    a. For Example,
        i. Dev environment can have name as TIQ-Dev-Stack-20250815
        ii. QA environment can have name as TIQ-Qa-Stack-20250815
        iii. PROD environment can have name as TIQ-Prod-Stack-20250815
3. Parameters:
    a. The User will be prompted to input values for the VPC configuration, Database Configuration, EKS configuration, and Backflipt TransferIQ configuration.
    b. Fill in the required fields based on the resource configuration specified in your CloudFormation template.
4. Once parameters are filled, click Next to continue.

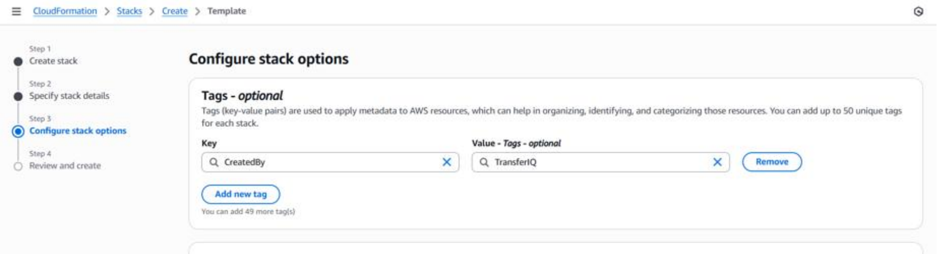
Input parameters are:

Parameters	**Description**	**Values**
Custom Prefix	Custom prefix for all the resources	example: 'Transfer-IQ'
DocumentDB Master Username	Master username for DocumentDB	tiqdbadmin (do not include any special characters or spaces, but you can choose to change if you want)
DocumentDB Master Password	Master password for DocumentDB	<password> (do not include special characters or spaces)
Backflipt TransferIQ Helm chart URL	The URL to the Helm chart ZIP file needed for Backflipt TransferIQ installation	https://backflipt-public-cloudformation-bucket.s3.us-east-1.amazonaws.com/Cloudformation-scripts/helm-charts.zip
Docker Token	Docker authentication token required to pull images from the Backflipt private registry.	dckr_pat_43BFaBxzNC2QMNCpCegA0o2y89g
SSL Certificate ARN	ARN of the AWS ACM SSL certificate used to enable secure HTTPS access to the Backflipt TransferIQ	<arn-of-uploaded-certificate>
TransferIQ URL	URL through which the Backflipt TransferIQ will be accessible after installation	<sub-domain>.<domain-name> example: transferiq.backflipt.com

TransferIQ Application Subdomains	Subdomain(s) through which the Backflipt application components will be accessible	<sub-domain>.<domain-name> example: orchestrate.backflipt.com
Default User Email for Backflipt TransferIQ	Email address for the default Super Admin account used to access the Backflipt TransferIQ	<email-id>
Default Password for Backflipt TransferIQ	Password for the Super Admin user to log in to the Backflipt TransferIQ. Avoid using special characters	<password> (do not include special characters or spaces)

Step 3: Configure Stack Options

1. Options:
   - Choose optional configurations like Tags, Permissions, and Advanced Options.
   - Adding the Tags Key as ‘CreatedBy' and the value as 'TransferIQ’ is **mandatory**



Configure stack options

Tags - *optional*

Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organizing, identifying, and categorizing those resources. You can add up to 50 unique tags for each stack.

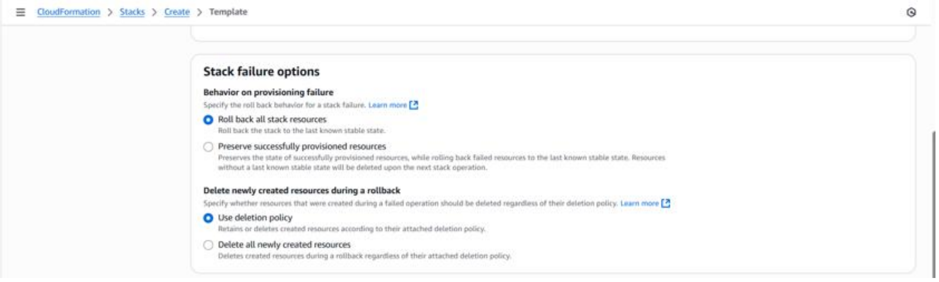
Key: CreatedBy

Value - *Tags - optional*: TransferIQ

You can add 49 more tag(s)

   - Tags: key-value pairs can be added to organize resources within AWS.
   - Permissions: Use the Role created in the Pre-requisites and proceed
     - TiQ-cfs-assume-role
2. Stack Failure Options:

- Select Roll back all stack resources and delete newly created resources during a rollback as Use deletion policy



Stack failure options

Behavior on provisioning failure

Specify the roll back behavior for a stack failure. Learn more

● Roll back all stack resources
Roll back the stack to the last known stable state.

○ Preserve successfully provisioned resources
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

Delete newly created resources during a rollback

Specify whether resources that were created during a failed operation should be deleted regardless of their deletion policy. Learn more

● Use deletion policy
Retains or deletes created resources according to their attached deletion policy.

○ Delete all newly created resources
Deletes created resources during a rollback regardless of their attached deletion policy.

3. Stack Policy:
   - The user can add a stack policy to protect certain resources during updates by specifying it in the Stack Policy section, however, select No stack policy is recommended.
4. Notification Options:
   - The user can specify a new or existing Amazon Simple Notification Service topic where notifications about stack events notifications for stack events are sent. however, this is an optional step.
5. Click Next to continue.

Step 4: Review and Create Stack

1. Review the configuration: Review all the details entered in the previous steps.
2. Acknowledgements:
   - Acknowledge any warnings or messages related to the stack template, such as IAM resource creation or security settings.
3. Review and confirm all the sections, then click on Submit.

Step 5: Monitor Stack Creation

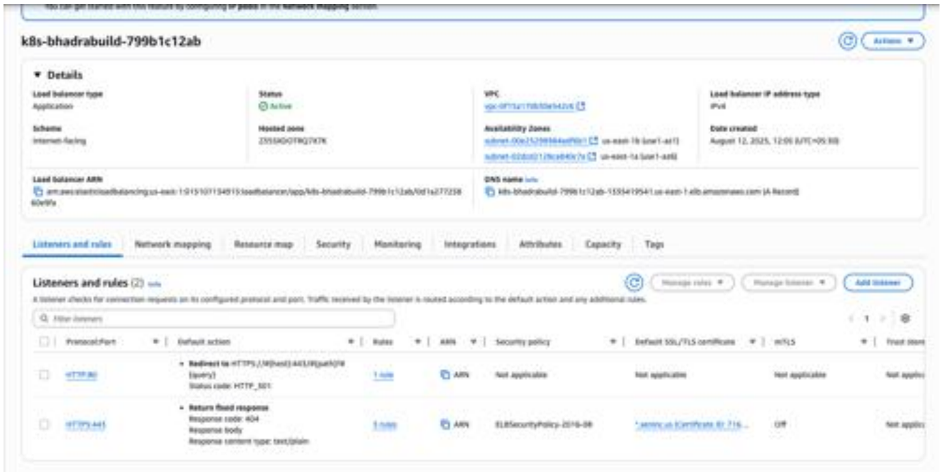
1. After clicking Create stack, CloudFormation will create the resources defined in the template.
2. The User will be redirected to the Stacks Dashboard, where the stack status will appear.
   - The Status will initially show as CREATE_IN_PROGRESS.
3. Once the stack creation is complete, the Status will update to CREATE_COMPLETE.
4. It would take approximately 23-28 minutes for the complete stack installation.

Step 6: View Resources and Outputs

1. After creating the stack, click on its name to see its details.
2. Resources Tab: This will show all the resources created by the CloudFormation stack.

Step 7: Load Balancer DNS mapping

1. Upon successful stack creation, a Load Balancer will be automatically generated in the EC2 Load Balancers section, identifiable by the prefix k8s-bhadra-build.
2. Use the DNS Name provided by this Load Balancer to create a DNS record corresponding to the TransferIQ URL supplied as an input during CloudFormation stack creation, within the DNS management system (e.g., Amazon Route 53). This ensures that users accessing the TransferIQ URL will be correctly redirected to the TransferIQ Application.



k8s-bhadrabuild-799b1c12ab

▼ Details

Load balancer type: Application

Status: Active

VPC: [illegible]

Load balancer IP address type: IPv4

Scheme: Internet-facing

Hosted zone: Z35SXDOTRQ7X7K

Availability Zones: [illegible] us-east-1b (use1-az1)
[illegible] us-east-1a (use1-az6)

Date created: August 12, 2025, 12:05 (UTC+05:30)

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:[illegible]:loadbalancer/app/k8s-bhadrabuild-799b1c12ab/[illegible]

DNS name: k8s-bhadrabuild-799b1c12ab-[illegible].us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (2)

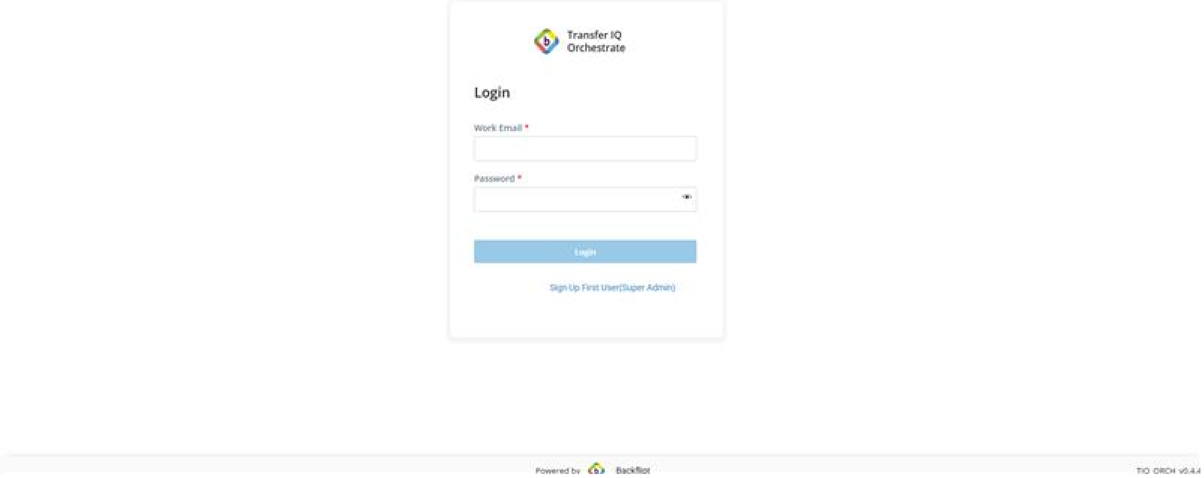
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
□	HTTP:80	Redirect to HTTPS://#{host}:443/#{path}?#{query} Status code: HTTP_301	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not applic
□	HTTPS:443	Return fixed response Response code: 404 Response body Response content type: text/plain	[illegible] rules	ARN	ELBSecurityPolicy-2016-08	[illegible] (Certificate ID: 716...	Off	Not applic

Step 8: Orchestrate App Launch

1. Once DNS mapping and route propagation are complete, open a web browser and navigate to the **Orchestrate URL** you provided during stack creation. The login page

will appear (as shown in the screenshot).



Estimated Time for Completion of Deployment

Estimated Time for Deployment

1. CloudFormation Stack Creation:
    a. The guide mentions that creating a CloudFormation stack typically takes 23 to 28 minutes. This is the time needed to provision the necessary resources and configure the environment.
2. Total Deployment Time:
    a. An exact total time for the entire process is around 1 hour, as the steps leading up to and following the CloudFormation stack creation (e.g., DNS configuration, IAM Identity Center setup, and post-installation tasks) might take additional time depending on the complexity of your environment and configurations.

Key Time Estimates

- CloudFormation Stack Creation: 23–28 minutes
- IAM Identity Center Setup: A few minutes for user and application configuration
- DNS and Load Balancer Mapping: Dependent on DNS propagation time, but configuration steps should take under 10 minutes
- Post-Installation Steps (e.g., importing the Orchestrate app, setting DNS, etc.): Likely another 5–10 minutes depending on the environment

For a complete, uninterrupted setup, a reasonable time estimate would be around 45 mins to 1 hour assuming no major errors or troubleshooting.

Troubleshooting

Recovering from CloudFormation Stack Failures:

- Issue: If the CloudFormation stack fails during creation, it will show a status of "CREATE_FAILED."
- Recovery Steps:
  - Monitor Stack Creation: View the stack events in the AWS Management Console to identify which resource caused the failure.
  - Rollback and Reattempt: If a failure occurs, select "Rollback all stack resources and delete newly created resources" in the rollback options to clean up.
  - Recreate the Stack: After identifying and correcting the issue, such as a configuration error, re-run the CloudFormation stack creation process with updated parameters.

S3 Bucket Name Conflict (Not Unique):

- Issue: If the chosen S3 bucket name is not unique, the deployment will fail during stack creation.
- Recovery Steps:
  - Check for Conflicting Names: Ensure the S3 bucket name is globally unique.
  - Change the Bucket Name: If the bucket name conflict is detected, update the name in the CloudFormation template and reattempt the stack creation.

Elastic IP Limit Reached:

- Issue: AWS limits the number of Elastic IPs that can be allocated to an account. If the account exceeds this limit, provisioning Elastic IPs will fail.
- Recovery Steps:
  - Release Unused Elastic IPs: Go to the EC2 dashboard in AWS and release any unused Elastic IPs.
  - Request Quota Increase: If more Elastic IPs are needed, request an increase through the AWS Service Quotas dashboard.
  - Reattempt Stack Creation: After resolving the Elastic IP issue, reattempt the CloudFormation stack creation or update the load balancer configuration.

DNS Mapping and Load Balancer Issues:

- Issue: Problems with DNS name resolution or load balancer configuration might occur during installation.
- Recovery Steps:
  - Check Load Balancer Status: Ensure the load balancer has been successfully created and that the DNS name matches the TransferIQ URL provided during installation.
  - Recreate DNS Records: If necessary, reconfigure DNS records in Route 53 to ensure the TransferIQ URL points to the correct load balancer.

IAM Role and Policy Issues:

- Issue: Insufficient IAM role permissions can cause failures during stack creation or post-installation steps.
- Recovery Steps:
  - Review IAM Policies: Ensure that the IAM roles and policies (like TiQ-cfs-assume-role) have the correct permissions as specified in the deployment guide.
  - Attach Missing Permissions: If missing permissions are found, attach the required IAM policies and reattempt the failed process.

Post-Installation Failures (Application Launch, Configuration Errors):

- Issue: Post-installation tasks may fail due to incorrect configuration settings.
- Recovery Steps:
  - Re-import Orchestrate App: If the Orchestrate app import fails, retry importing the app after ensuring the correct BSON file is selected.
  - Recheck Environment Variables: Ensure that the environment variables for AWSRegion, tiqBucketName, etc., are correctly configured.
  - Re-run Metadata Settings Payload: Execute the cURL command again with the correct values for your environment to configure the metadata settings.

Troubleshooting CloudWatch Logs:

- Issue: If there are application-specific issues or system errors, CloudWatch logs might provide insights into what went wrong.
- Recovery Steps:
  - Review Logs: Use Amazon CloudWatch to access application logs and identify any errors or exceptions.

        - Adjust Configurations Based on Log Findings: Depending on the log data, adjust configurations (like IAM policies, S3 bucket settings, etc.) and restart the affected components.

Reconfigure DNS and Custom Domain:

- Issue: DNS records or custom domains might be misconfigured, leading to accessibility issues.
- Recovery Steps:
    - Edit Application DNS: If DNS mapping fails, recheck and reconfigure the custom domain in the application settings.
    - Ensure Correct Subdomain Configuration: Make sure the subdomain set during the CloudFormation stack is correctly mapped to the TransferIQ Application URL.

Re-Executing the Metadata Payload via cURL:

- Issue: If the metadata payload for the application configuration is not executed correctly, the app might not function as expected.
- Recovery Steps:
    - Re-execute the Payload Command: Use the provided cURL command to push the correct metadata settings to the TransferIQ application, ensuring the necessary environment-specific values are populated.

IAM Identity Center Setup

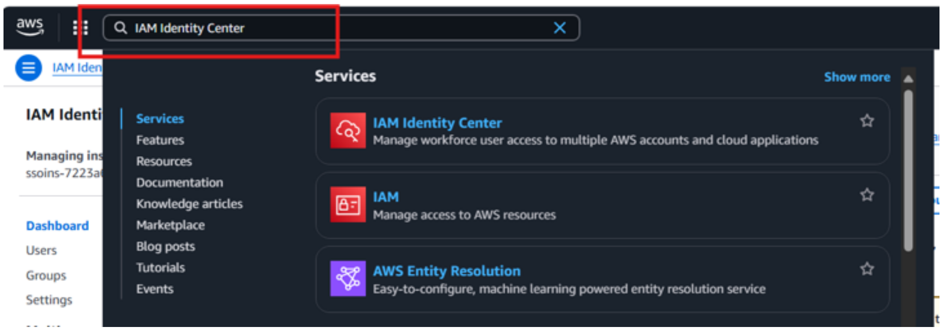
The IAM Identity Center setup is a foundational requirement for enabling secure, role-based, and seamless access to the TransferIQ Orchestrate MFT solution.

By using IAM Identity Center as the central identity store, organizations can:

- Unify User Management: Maintain a single source of truth for user accounts and groups, reducing administrative overhead and avoiding duplication across systems.
- Enable Single Sign-On (SSO): Provide users with a seamless login experience using their existing corporate credentials, without creating and managing separate usernames/passwords for TransferIQ Orchestrate.
- Enhance Security & Compliance: Enforce corporate authentication policies such as MFA (Multi-Factor Authentication), password rotation, and session controls directly from AWS Identity Center.
- Streamline Onboarding & Offboarding: Quickly provision or revoke access when employees join, change roles, or leave the organization—ensuring immediate alignment with compliance requirements.
- Support Centralized Access Control: Assign permissions and group memberships in one place, ensuring consistent enforcement across all integrated AWS and third-party applications.

Step 1: Create User Identities in IAM Identity Center

1. Open IAM Identity Center Services

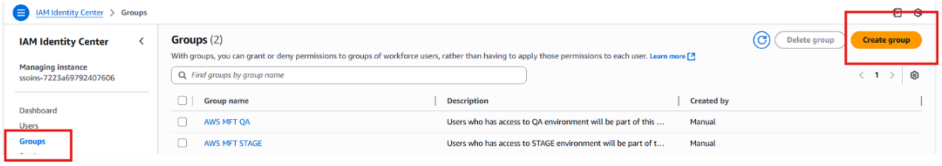


IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications

IAM
Manage access to AWS resources

AWS Entity Resolution
Easy-to-configure, machine learning powered entity resolution service

2. Click on Groups, Click on Create Group

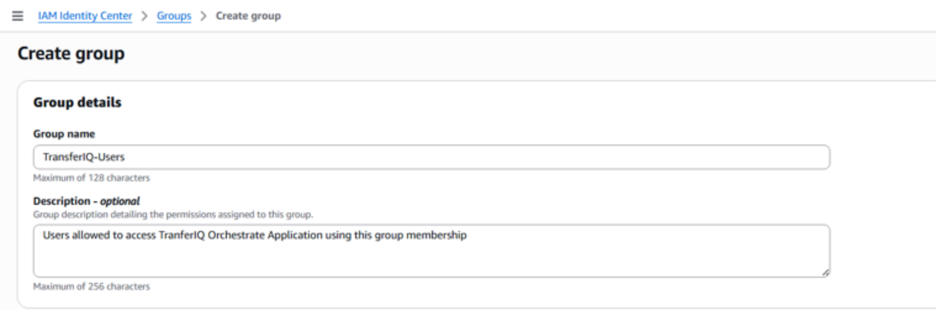


Groups (2)

With groups, you can grant or deny permissions to groups of workforce users, rather than having to apply those permissions to each user. Learn more

Group name	Description	Created by
AWS MFT QA	Users who has access to QA environment will be part of this ...	Manual
AWS MFT STAGE	Users who has access to STAGE environment will be part of t...	Manual

- Add Group details as mentioned in the screenshot, Scroll to the bottom and Click on Create Group



Create group

Group details

Group name: TransferIQ-Users

Maximum of 128 characters

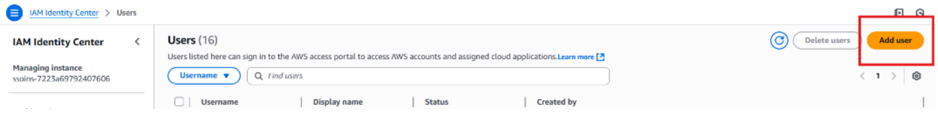
Description - *optional*

Group description detailing the permissions assigned to this group.

Users allowed to access TranferIQ Orchestrate Application using this group membership

Maximum of 256 characters

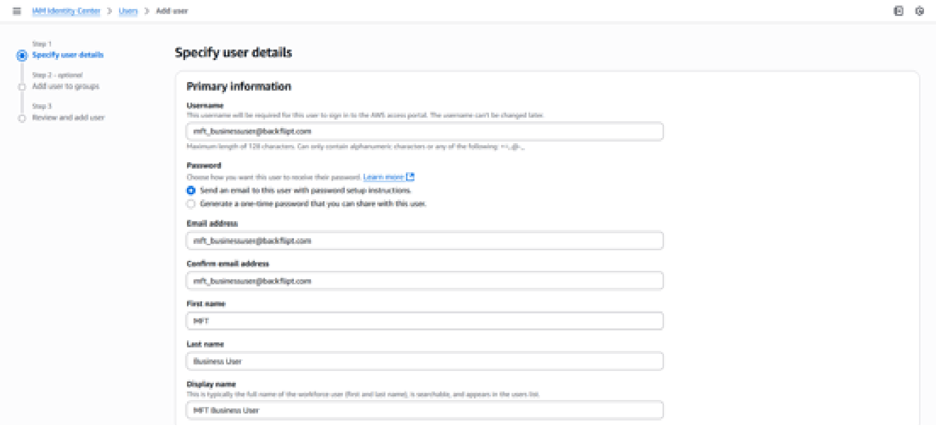
3. Click on Add User, provide user details, and click on Next
    1. Note: Follow this step only to add a new user, however, existing users can also be used for authentication for TranferIQ Orchestrate Application



Users (16)

Users listed here can sign in to the AWS access portal to access AWS accounts and assigned cloud applications. Learn more

Username	Display name	Status	Created by



IAM Identity Center > Users > Add user

Step 1
● Specify user details
Step 2 - optional
○ Add user to groups
Step 3
○ Review and add user

Specify user details

Primary information

Username
This username will be required for this user to sign in to the AWS access portal. The username can't be changed later.
mft_businessuser@backflipt.com
Maximum length of 128 characters. Can only contain alphanumeric characters or any of the following: +=,.@-_

Password
Choose how you want this user to receive their password. Learn more
● Send an email to this user with password setup instructions.
○ Generate a one-time password that you can share with this user.

Email address
mft_businessuser@backflipt.com

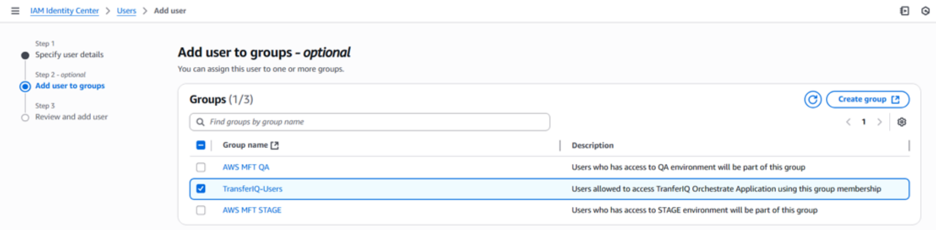
Confirm email address
mft_businessuser@backflipt.com

First name
MFT

Last name
Business User

Display name
This is typically the full name of the workforce user (first and last name), is searchable, and appears in the users list.
MFT Business User

3. Select the Group, and click on Next



IAM Identity Center > Users > Add user

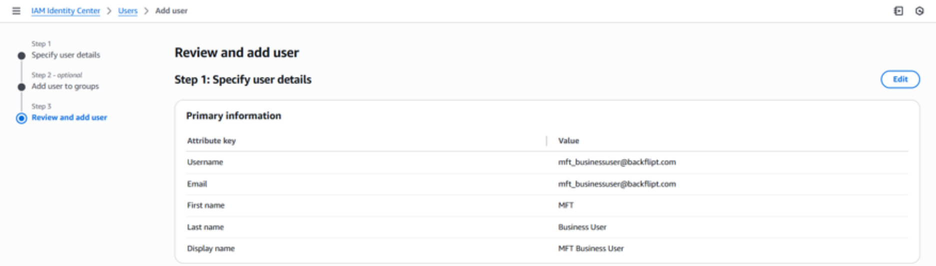
Step 1
● Specify user details
Step 2 - *optional*
● Add user to groups
Step 3
○ Review and add user

Add user to groups - *optional*
You can assign this user to one or more groups.

Groups (1/3)

☐	Group name	Description
☐	AWS MFT QA	Users who has access to QA environment will be part of this group
☑	TransferIQ-Users	Users allowed to access TranferIQ Orchestrate Application using this group membership
☐	AWS MFT STAGE	Users who has access to STAGE environment will be part of this group

4. Click on Add User



IAM Identity Center > Users > Add user

Step 1
● Specify user details
Step 2 - *optional*
● Add user to groups
Step 3
● Review and add user

Review and add user

Step 1: Specify user details

Primary information

Attribute key	Value
Username	mft_businessuser@backflipt.com
Email	mft_businessuser@backflipt.com
First name	MFT
Last name	Business User
Display name	MFT Business User

Step 2: Configure Applications for SSO in IAM Identity Center

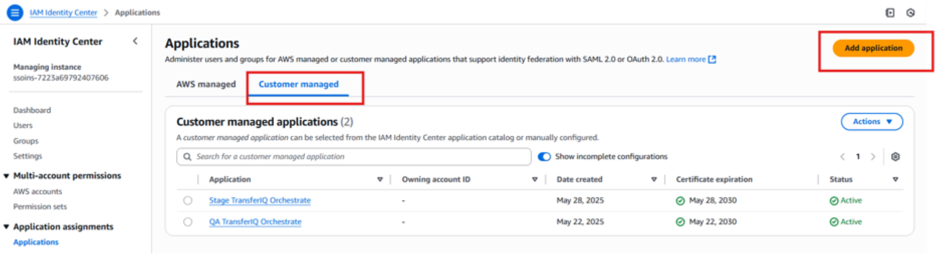
Integration with IAM Identity Center (formerly AWS SSO) is required to enable secure and seamless user access to the TransferIQ Orchestrate application. This setup allows users to authenticate using their corporate identities and facilitates a Single Sign-On (SSO) experience.

By integrating with IAM Identity Center, organizations can:

- Leverage centralized identity management.
- Enforce consistent access control policies.
- Eliminate the need for separate application-specific credentials.
- Streamline user onboarding and offboarding.

This ensures users can access TransferIQ Orchestrate through a unified login experience while adhering to enterprise security and compliance standards.

1. Add a new Customer Managed Application



IAM Identity Center > Applications

IAM Identity Center

Managing instance
ssoins-7223a69792407606

Applications

Administer users and groups for AWS managed or customer managed applications that support identity federation with SAML 2.0 or OAuth 2.0. Learn more

Customer managed applications (2)

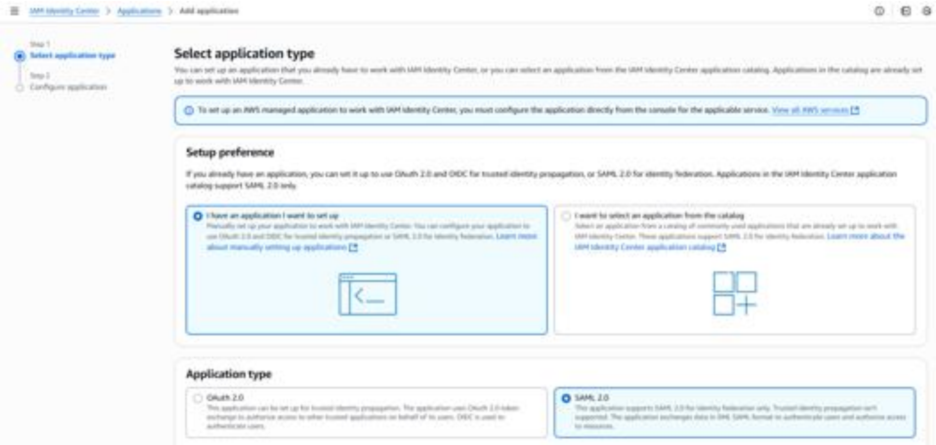
A *customer managed application* can be selected from the IAM Identity Center application catalog or manually configured.

Search for a customer managed application

Show incomplete configurations

	Application	Owning account ID	Date created	Certificate expiration	Status
○	Stage TransferIQ Orchestrate	-	May 28, 2025	✓ May 28, 2030	✓ Active
○	QA TransferIQ Orchestrate	-	May 22, 2025	✓ May 22, 2030	✓ Active

2. Select Application Type, Select ‘I have an application I want to set up’, Select SAML 2.0, Click on Next



IAM Identity Center > Applications > Add application

Select application type

You can set up an application that you already have to work with IAM Identity Center, or you can select an application from the IAM Identity Center application catalog. Applications in the catalog are already set up to work with IAM Identity Center.

To set up an AWS managed application to work with IAM Identity Center, you must configure the application directly from the console for the applicable service. View all AWS services

Setup preference

If you already have an application, you can set it up to use OAuth 2.0 and OIDC for trusted identity propagation, or SAML 2.0 for identity federation. Applications in the IAM Identity Center application catalog support SAML 2.0 only.

● I have an application I want to set up
Manually set up your application to work with IAM Identity Center. You can configure your application to use OAuth 2.0 and OIDC for trusted identity propagation or SAML 2.0 for identity federation. Learn more about manually setting up applications

○ I want to select an application from the catalog
Select an application from a catalog of commonly used applications that are already set up to work with IAM Identity Center. These applications support SAML 2.0 for identity federation. Learn more about the IAM Identity Center application catalog

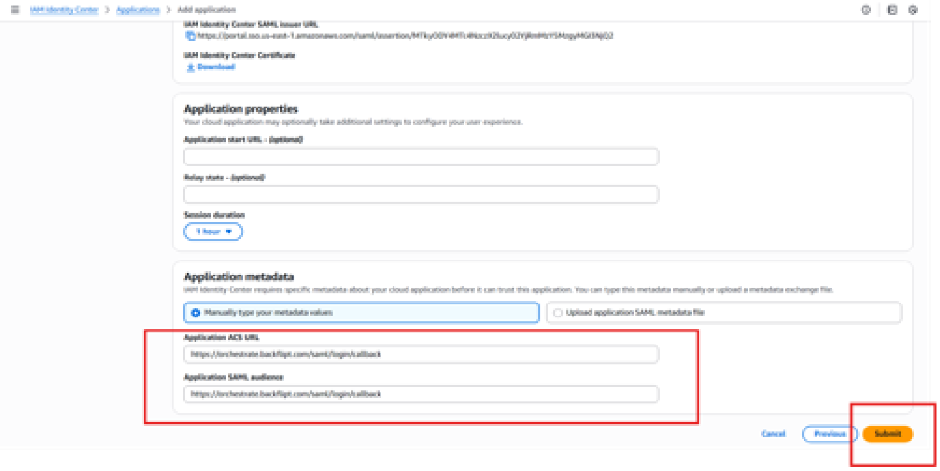
Application type

○ OAuth 2.0
This application can be set up for trusted identity propagation. The application uses OAuth 2.0 token exchange to authorize access to other trusted applications on behalf of its users. OIDC is used to authenticate users.

● SAML 2.0
This application supports SAML 2.0 for identity federation only. Trusted identity propagation isn't supported. The application exchanges data in XML SAML format to authenticate users and authorize access to resources.

3. Scroll down to Application Metadata and select Manually Type your metadata values, provide Application ACS URL, and Application SAML Audience
    a. Format of the URL would be your TransferIQ Application URL from https://backflipt1.atlassian.net/wiki/spaces/Devops/pages/1775599711/Backflipt+s+TiQ+solution+from+Marketplace#Step-2%3A-Cloudformation-Stack /saml/login/callback
        1. For Example: https://orchestrate.backflipt.com/saml/login/callback

4. Click on Submit



IAM Identity Center > Applications > Add application

IAM Identity Center SAML issuer URL
https://portal.sso.us-east-1.amazonaws.com/saml/assertion/[illegible]

IAM Identity Center Certificate
Download

Application properties
Your cloud application may optionally take additional settings to configure your user experience.

Application start URL - *(optional)*

Relay state - *(optional)*

Session duration
1 hour ▾

Application metadata
IAM Identity Center requires specific metadata about your cloud application before it can trust this application. You can type this metadata manually or upload a metadata exchange file.

● Manually type your metadata values ○ Upload application SAML metadata file

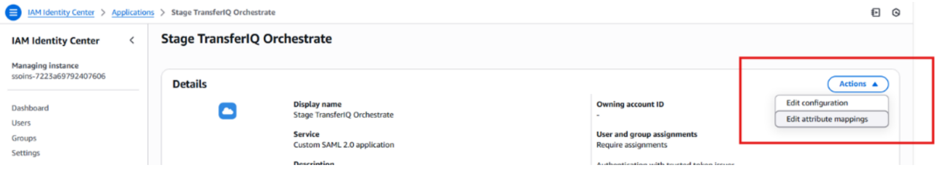
Application ACS URL
https://orchestrate.backflipt.com/saml/login/callback

Application SAML audience
https://orchestrate.backflipt.com/saml/login/callback

Step 3: Edit Attribute Mappings **in IAM Identity Center**

The advantage of adding Attribute Mappings in IAM Identity Center for a custom application like TransferIQ Orchestrate is that it ensures the right user identity data flows into the application during Single Sign-On (SSO). In short, adding attribute mappings allows your custom application to trust and consume identity data directly from IAM Identity Center without maintaining separate user profiles, while enabling automation, consistency, security, and a better SSO experience.

- To Edit Attribute Mappings, Select the Custom Application created in Step 2, Click on Actions, Select Edit attribute mappings



IAM Identity Center > Applications > Stage TransferIQ Orchestrate

Stage TransferIQ Orchestrate

Details

Display name
Stage TransferIQ Orchestrate

Service
Custom SAML 2.0 application

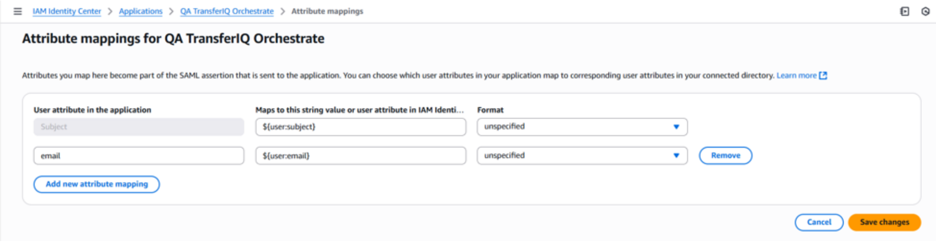
Owning account ID
-

User and group assignments
Require assignments

1. Add First Attribute Mappings as below:
    1. User attribute in the application: Subject
    2. Mapping of this string value or attribute of IAM Identity Center: ${user:subject}
    3. Format: Unspecified
2. Add Second Attribute Mappings as below:

    1. User attribute in the application: email
    2. Mapping of this string value or attribute of IAM Identity Center: ${user:email}

    3. Format: Unspecified
3. Click on Save Changes



IAM Identity Center > Applications > QA TransferIQ Orchestrate > Attribute mappings

Attribute mappings for QA TransferIQ Orchestrate

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. Learn more

User attribute in the application	Maps to this string value or user attribute in IAM Identi...	Format	
Subject	${user:subject}	unspecified	
email	${user:email}	unspecified	Remove

Step 5: Launch the Orchestrate Application Sign-Up page will appear, allowing users to create their accounts and access the application.

Enforcing Least Privilege in Access Management and IAM roles, Policies

Introduction

Least privilege is a fundamental security best practice that ensures identities (users, roles, or services) are granted only the minimum set of permissions required to perform their tasks. By limiting access to exactly what is needed, organizations reduce the risk of accidental changes, misuse, or security breaches.

Each IAM role and policy is defined with its purpose, scope, and justification for the permissions granted. This ensures that only the minimum required access is provided to execute CloudFormation stacks and enable the TransferIQ services.

This section provides guidance on how the IAM roles and policies are configured for Backflipt TransferIQ MFT. It highlights how the solution implements least privilege principles and security controls to balance functionality and security.

Prerequisite IAM Role for Executing CloudFormation Stack

To deploy the TransferIQ MFT application using AWS CloudFormation, a prerequisite IAM role is required. This role is designed in alignment with the principle of least privilege (PoLP), ensuring that it has only the minimum permissions necessary to execute the stack while avoiding overly broad or administrative access.

Least Privilege Considerations

When applying least privilege to CloudFormation, three layers of access are considered in the design of this role:

Permissions for CloudFormation Service

Only the designated execution role is permitted to create, update, or delete CloudFormation stacks. This reduces the risk of unauthorized users making changes outside of controlled deployment processes.

Permissions to Provision Resources

The role grants CloudFormation the ability to provision only the resources required by the TransferIQ MFT application (e.g., networking, compute, storage, and application services). While dynamic provisioning requires "Resource": "*", the scope is still restricted to the specific services necessary for the stack, not unrestricted administrative access.

Permissions for Provisioned Resources

After resources are created, they enforce their own least-privilege runtime permissions (e.g., IAM roles for Lambda, bucket policies for S3, or security groups for EC2). The

CloudFormation execution role is limited to deployment only and does not maintain unnecessary control over provisioned resources.

Scope of the Execution Role

- Enables CloudFormation to dynamically provision infrastructure across required AWS services.
- Grants only the minimum set of actions necessary for stack creation and lifecycle management.
- Ensures separation of duties between deployment permissions (via this role) and runtime permissions (applied directly to provisioned resources).
- Avoids reliance on broad Administrator privileges, providing a secure, controlled alternative.

Important Note

This IAM role is required in the following cases:

- If the client/user has not been granted administrator privileges in the AWS account.
- If the client/user wishes to deploy the application using least privilege access, rather than relying on unrestricted admin rights.

Sample User’s Policy:

```
{                   "Sid": "CloudFormationFullAccess",
       "Effect": "Allow",
       "Action": [
         "cloudformation:CreateStack",
         "cloudformation:DescribeStacks",
         "cloudformation:ListStacks",
         "cloudformation:GetTemplateSummary",
         "cloudformation:ValidateTemplate",
         "cloudformation:UpdateStack",
         "cloudformation:DeleteStack",
         "cloudformation:DescribeStackResources",
         "cloudformation:DescribeStackEvents",
         "cloudformation:DescribeStackResource",
         "cloudformation:ListStackResources",
         "cloudformation:ListExports",
         "cloudformation:RollbackStack",
         "cloudformation:CreateChangeSet"
       ],
       "Resource": "*"}
```

AWS Identity and Access Management (IAM) role and IAM policy

IAM Roles and Policies for Deployment Execution

CloudFormation Execution Role

Purpose: This role allows AWS CloudFormation to create and manage infrastructure resources as defined in the deployment templates.

Role Name: TransferIQ-DeploymentRole

- Purpose:
  This IAM role is assumed by AWS CloudFormation during the execution of the TransferIQ deployment stack.

```
Trusted Entity:
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "cloudformation.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Why Needed: CloudFormation requires this trust relationship to assume the role and provision resources on behalf of the user.

IAM Policies Attached to the Role

CloudFormation Access

- Policy Name: CloudFormationFullAccess
- Description: Grants CloudFormation the ability to create, update, validate, and delete stacks, ensuring that TransferIQ templates can be deployed, updated, and rolled back successfully.

EC2 Networking and Compute Access

- Policy Name: EC2NetworkingAndComputeAccess

- Description: Enables provisioning and management of VPCs, subnets, security groups, and EC2 instances, which are required to provide secure networking and compute environments for TransferIQ.

RDS Provisioning Access

- Policy Name: RDSProvisioningAccess
- Description: Allows creation and management of RDS databases and clusters to support TransferIQ persistence and relational database requirements.

ElastiCache Provisioning Access

- Policy Name: ElastiCacheProvisioningAccess
- Description: Provides permissions to manage cache clusters and replication groups, enabling TransferIQ workloads to use caching for performance optimization.

S3 Bucket Management Access

- Policy Name: S3BucketManagementAccess
- Description: Grants permissions to create and manage S3 buckets and objects, which are required for storing configuration data, logs, and artifacts used by TransferIQ.

SQS Full Access

- Policy Name: SQSFullAccess
- Description: Enables creation and management of SQS queues to support asynchronous processing and reliable message delivery within TransferIQ workflows.

IAM Role and Profile Management

- Policy Name: IAMRoleAndProfileManagement
- Description: Allows creation and management of IAM roles, instance profiles, and inline policies, ensuring that TransferIQ components can securely assume roles with the required access.

EKS Cluster and Nodegroup Access

- Policy Name: EKSClusterAndNodegroupAccess

- Description: Grants permissions to provision and manage EKS clusters, nodegroups, and add-ons, enabling TransferIQ to run containerized workloads in a managed Kubernetes environment.

KMS Key and Encryption Access

- Policy Name: KMSKeyAndEncryptionAccess
- Description: Provides access to create and manage KMS keys and perform encryption/decryption operations, ensuring sensitive TransferIQ data is protected at rest and in transit.

SSM Parameter and Document Access

- Policy Name: SSMParameterDocumentAssociationAccess
- Description: Grants access to store, retrieve, and manage SSM parameters and documents, enabling secure storage of deployment configurations and automation scripts for TransferIQ.

Transfer Family Access

- Policy Name: TransferFamilyAccess
- Description: Provides permissions to create and manage Transfer Family servers, users, and connectors, which form the core of the TransferIQ secure file transfer solution.

Additional Service Access

- Lambda Access: Enables creation and management of Lambda functions used for orchestration within TransferIQ.
- EventBridge Access: Grants permissions to create and manage event rules and targets for workflow automation.
- CloudWatch Logs and Alarms: Provides visibility into operations and system health by enabling logging, monitoring, and alerting.
- ECS Access: Supports provisioning of ECS clusters and services if TransferIQ components are deployed on containerized infrastructure.
- Elastic Load Balancing Access: Grants permissions to create load balancers for distributing traffic and scaling workloads.
- GuardDuty Access: Enables integration with GuardDuty for malware protection and enhanced security monitoring.

IAM Policy is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMRoleAndPolicyManagement",
      "Effect": "Allow",
      "Action": [
        "iam:CreatePolicy",
        "iam:DeletePolicy",
        "iam:GetPolicy",
        "iam:ListRoles",
        "iam:ListPolicies",
        "iam:TagRole",
        "iam:TagPolicy",
        "iam:PutUserPolicy",
        "iam:AttachUserPolicy",
        "iam:ListAttachedUserPolicies",
        "iam:DetachUserPolicy",
        "iam:ListPolicyVersions",
        "iam:GetUserPolicy",
        "iam:GetPolicyVersion",
        "iam:DeleteUserPolicy",
        "iam:UntagPolicy",
        "iam:CreatePolicyVersion"
      ],
      "Resource": "*"
    },
    {
      "Action": [
        "secretsmanager:UntagResource",
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:CreateSecret",
        "secretsmanager:DeleteSecret",
        "secretsmanager:ListSecrets",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "SecretsManagerAccess"
    },
    {
      "Action": [
        "s3:PutBucketAcl",
        "s3:PutBucketVersioning",
        "s3:PutEncryptionConfiguration",
        "s3:GetBucketLocation",
        "s3:PutBucketNotification",
        "s3:PutBucketLogging"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "S3BucketManagement"
    },
    {
      "Action": [
        "transfer:CreateServer",
        "transfer:DeleteServer",
        "transfer:DescribeServer",
        "transfer:UpdateServer",
        "transfer:ListServers",
        "transfer:CreateConnector",
        "transfer:DeleteConnector",
        "transfer:DescribeConnector",
        "transfer:UpdateConnector",
        "transfer:ListConnectors",
        "transfer:CreateUser",
        "transfer:DeleteUser",
        "transfer:UpdateUser",
        "transfer:DescribeUser",
        "transfer:ListUsers",
        "transfer:TagResource",
        "transfer:UntagResource",
        "transfer:ListTagsForResource",
        "transfer:ImportSshPublicKey"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "TransferFamilyAccess"
    },
    {
      "Action": [
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunction",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "LambdaCreationAccess"
    },
    {
      "Action": [
        "events:CreateEventBus",
        "events:DeleteEventBus",
        "events:DescribeEventBus",
        "events:ListEventBuses",
        "events:DescribeRule",
        "events:PutRule",
        "events:EnableRule",
        "events:DisableRule",
        "events:DeleteRule",
        "events:ListRules",
        "events:ListTargetsByRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EventBridgeAccess"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups",
        "logs:DeleteLogGroup",
        "logs:TagLogGroup",
        "logs:ListLogGroupsForQuery",
        "logs:PutRetentionPolicy"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "CloudWatchAccess"
    },
    {
      "Action": [
        "ecs:CreateCluster",
        "ecs:DeleteCluster",
        "ecs:RegisterTaskDefinition",
        "ecs:DeregisterTaskDefinition",
        "ecs:DescribeClusters",
        "ecs:DescribeServices",
        "ecs:CreateService",
        "ecs:UpdateService",
        "ecs:DeleteService",
        "ecs:ListClusters",
        "ecs:ListServices",
        "ecs:ListTaskDefinitions"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "ECSAccess"
    },
    {
      "Action": [
        "ec2:DeleteNetworkInterface",
        "ec2:CreateNetworkInterface",
        "ec2:GetSecurityGroupsForVpc"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2NetworkInterface"
    },
    {
      "Action": [
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:CreateListener",
        "elasticloadbalancing:CreateTargetGroup",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:ModifyTargetGroup",
        "elasticloadbalancing:DeleteLoadBalancer",
        "elasticloadbalancing:DeleteListener",
        "elasticloadbalancing:DeleteTargetGroup",
        "elasticloadbalancing:Describe*",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:ModifyLoadBalancerAttributes"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "ELBAccess"
    },
    {
      "Sid": "CloudFormationFullAccess",
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStacks",
        "cloudformation:GetTemplateSummary",
        "cloudformation:ValidateTemplate",
        "cloudformation:UpdateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:ListStackResources",
        "cloudformation:ListExports",
        "cloudformation:RollbackStack",
        "cloudformation:CreateChangeSet"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2NetworkingAndComputeAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateRouteTable",
        "ec2:AttachInternetGateway",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:CreateInternetGateway",
        "ec2:CreateNatGateway",
        "ec2:CreateRoute",
        "ec2:CreateRouteTable",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSubnet",
        "ec2:CreateTags",
        "ec2:CreateVpc",
        "ec2:CreateVpcEndpoint",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:DeleteInternetGateway",
        "ec2:DeleteNatGateway",
        "ec2:DeleteRoute",
        "ec2:DeleteRouteTable",
        "ec2:DeleteSubnet",
        "ec2:DeleteTags",
        "ec2:DeleteVpc",
        "ec2:DeleteVpcEndpoints",
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeAddresses",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeImages",
        "ec2:DescribeInstances",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeNatGateways",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeVpcEndpointServices",
        "ec2:DescribeVpcs",
        "ec2:ModifySubnetAttribute",
        "ec2:ModifyVpcAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RunInstances",
        "ec2:TerminateInstances",
        "ec2:DisassociateRouteTable",
        "ec2:DetachInternetGateway",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteSecurityGroup"
      ],
      "Resource": "*"
    },
    {
      "Sid": "RDSProvisioningAccess",
      "Effect": "Allow",
      "Action": [
        "rds:AddTagsToResource",
        "rds:CreateDBCluster",
        "rds:CreateDBInstance",
        "rds:CreateDBSubnetGroup",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "rds:DescribeDBSubnetGroups",
        "rds:ListTagsForResource",
        "rds:DeleteDBInstance",
        "rds:DeleteDBCluster",
        "rds:DeleteDBSubnetGroup"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ElastiCacheProvisioningAccess",
      "Effect": "Allow",
      "Action": [
        "elasticache:AddTagsToResource",
        "elasticache:CreateCacheCluster",
        "elasticache:CreateCacheSubnetGroup",
        "elasticache:DeleteCacheCluster",
        "elasticache:DeleteCacheSubnetGroup",
        "elasticache:DescribeCacheClusters",
        "elasticache:DescribeCacheSubnetGroups",
        "elasticache:ListTagsForResource",
        "elasticache:ModifyCacheCluster",
        "elasticache:RemoveTagsFromResource",
        "elasticache:CreateReplicationGroup",
        "elasticache:DescribeReplicationGroups",
        "elasticache:DeleteReplicationGroup"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3BucketManagementAccess",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:DeleteBucketPolicy",
        "s3:GetBucketPolicy",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetBucketTagging",
        "s3:ListBucket",
        "s3:PutBucketPolicy",
        "s3:PutBucketPublicAccessBlock",
        "s3:PutBucketTagging"
      ],
      "Resource": [
        "arn:aws:s3:::*"
      ]
    },
    {
      "Sid": "SQSFullAccess",
      "Effect": "Allow",
      "Action": [
        "sqs:CreateQueue",
        "sqs:DeleteMessage",
        "sqs:DeleteQueue",
        "sqs:GetQueueAttributes",
        "sqs:ListQueues",
        "sqs:PurgeQueue",
        "sqs:SendMessage",
        "sqs:SetQueueAttributes",
        "sqs:TagQueue",
        "sqs:UntagQueue"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IAMRoleAndProfileManagement",
      "Effect": "Allow",
      "Action": [
        "iam:AddRoleToInstanceProfile",
        "iam:AttachRolePolicy",
        "iam:CreateInstanceProfile",
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:DeleteRolePolicy",
        "iam:GetInstanceProfile",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:PassRole",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:CreateServiceLinkedRole",
        "iam:DeleteRole",
        "iam:DeleteRolePolicy",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:DeleteInstanceProfile",
        "iam:ListInstanceProfilesForRole",
        "iam:DetachRolePolicy",
        "iam:ListRolePolicies",
        "iam:ListAttachedRolePolicies",
        "iam:GetRolePolicy"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EKSClusterAndNodegroupAccess",
      "Effect": "Allow",
      "Action": [
        "eks:AssociateAccessPolicy",
        "eks:CreateAccessEntry",
        "eks:CreateAddon",
        "eks:CreateCluster",
        "eks:CreateNodegroup",
        "eks:DeleteAccessEntry",
        "eks:DeleteNodegroup",
        "eks:DescribeAccessEntry",
        "eks:DescribeAddon",
        "eks:DescribeCluster",
        "eks:DescribeNodegroup",
        "eks:ListAccessEntries",
        "eks:ListAddons",
        "eks:ListAssociatedAccessPolicies",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "eks:DeleteAddon",
        "eks:DeleteCluster"
      ],
      "Resource": "*"
    },
    {
      "Sid": "KMSKeyAndEncryptionAccess",
      "Effect": "Allow",
      "Action": [
        "kms:CreateAlias",
        "kms:CreateKey",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:Encrypt",
        "kms:ListAliases",
        "kms:ListGrants",
        "kms:PutKeyPolicy",
        "kms:TagResource",
        "kms:UntagResource",
        "kms:DeleteAlias",
        "kms:ScheduleKeyDeletion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SSMParameterDocumentAssociationAccess",
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource",
        "ssm:CreateAssociation",
        "ssm:CreateDocument",
        "ssm:DeleteAssociation",
        "ssm:DeleteDocument",
        "ssm:DeleteParameter",
        "ssm:DescribeAssociation",
        "ssm:DescribeDocument",
        "ssm:DescribeParameters",
        "ssm:GetDocument",
        "ssm:GetParameter",
        "ssm:PutParameter",
        "ssm:UpdateAssociation",
        "ssm:UpdateDocument"
      ],
      "Resource": "*"
    }
  ]
}
```

IAM Roles and Policies Created by CloudFormation for TransferIQ MFT

S3 Stack Deployment –

In S3 Stack we provisions multiple Amazon S3 buckets for application assets, application versions, execution logs for TransferIQ Application, and MFTBucket for Managed File Transfer (MFT). This stack also sets up IAM roles, policies, and VPC endpoint access to secure data exchange with S3.

The primary IAM-related resources are:

- IAM Role for Lambda Folder Creation
- Inline IAM Policies for S3 and CloudWatch Logs
- S3 Bucket Policies enforcing HTTPS and access control
- VPC Endpoint Policy restricting S3 access

IAM Role: Lambda Folder Creator

Role Name - ${CustomPrefix}-${Environment}-S3-objectCreation-lambda-role

Trust Policy (AssumeRole)

- Principal: lambda.amazonaws.com
- Purpose: Grants AWS Lambda permission to assume this role and execute S3 folder creation logic.

Inline Policies

Policy Name: S3FolderCreatorPolicy

- Permissions Granted: S3 Actions: s3:PutObject

Scope:

- arn:aws:s3:::bft-mft-${CustomPrefix}-${Environment}/
- arn:aws:s3:::bft-mft-${CustomPrefix}-${Environment}/*

Purpose: Allows Lambda to create logical "folders" inside the MFT bucket by placing empty objects with prefix names.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-tiqstg-staging/",
        "arn:aws:s3:::bft-mft-tiqstg-staging/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

S3 Bucket Policies:

These are the Bucket policies created by CloudFormation stack

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyHttp",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::bft-s3-assets-tiqstg-staging/*",
        "arn:aws:s3:::bft-s3-assets-tiqstg-staging"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    }   ]
}
```

AssetsBucket Policy

- Deny Non-HTTPS Access: Explicitly denies all s3:* actions if aws:SecureTransport is false.
- Use Case: Hosting assets where controlled public access is required (e.g., static files).

AppVersionsBucket Policy

- Deny Non-HTTPS Access: Denies s3:* if the request is not over TLS.
- Use Case: Stores application version artifacts securely, with no public access.

ExecutionsBucket Policy

- Deny Non-HTTPS Access: Denies s3:* if not using TLS.
- Use Case: Stores execution logs securely with transport-level encryption enforced.

. VPC Endpoint for S3 -

- Interface VPC Endpoint for Amazon S3. This Security Group Allows inbound TCP traffic on port 443 (HTTPS) from any source (0.0.0.0/0).

VPC Endpoint Policy - Effect: Allow Actions: s3:*

- Resources: All objects and root buckets of Assets, AppVersions, and Executions buckets.
- Condition: aws:ResourceAccount = current AWS Account ID
- Purpose: Ensures only resources within the same AWS account can use the endpoint to access these buckets. Endpoint access is tightly scoped to the three

buckets. Public internet exposure is minimized by forcing private VPC connectivity.

EKS Stack Deployment:

This deployment provisions an Amazon EKS cluster with managed node groups, workstation EC2 for management, and supporting IAM roles and policies. IAM roles and policies ensure secure operation of the EKS control plane, worker nodes, and management EC2 workstation.

IAM Role: EKS Control Plane Role

Role Name - ${CustomPrefix}-${Environment}-EKSClusterRole

Trust Policy

- Principal: eks.amazonaws.com ,Purpose: Allows the Amazon EKS control plane service to assume this role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EKSClusterAssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "eks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ] }
```

AWS Managed Policies

- AmazonEKSClusterPolicy – Grants permissions required by the EKS control plane to manage cluster resources.
- AmazonEKSVPCResourceController – Allows EKS to manage VPC resources like ENIs for worker nodes.
- Inline Policies
- DenyLogGroupCreation – Explicitly denies logs:CreateLogGroup. Prevents EKS from auto-creating log groups, enforcing centralized log group management.

- KMSUsagePolicy – Grants Encrypt, Decrypt, ListGrants, and DescribeKey permissions on a specific KMS key. Enables encryption/decryption of Kubernetes secrets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "logs:CreateLogGroup",
      "Resource": "*",
      "Effect": "Deny"
    }
  ]
}
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ListGrants",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-east-1:account-id:key/238e1f51-4d77-4427-bdba-
8825205f6a53",
      "Effect": "Allow"
    }
  ]
}
```

- Security Considerations - Enforces least privilege by denying unnecessary log creation. KMS access scoped to a single key by ARN.

IAM Role: Workstation EC2 Role

Role Name - bft-Workstation-EC2-Role-${CustomPrefix}-${Environment}

Trust Policy - Principal: ec2.amazonaws.com, EC2 instance profile role for workstation instances.

Attached Managed Policies

- AmazonSSMManagedInstanceCore – Allows EC2 workstation management via AWS Systems Manager.

Inline Policy: EksOidcAndClusterMgmt

Permissions granted:

- OIDC Provider Management iam:CreateOpenIDConnectProvider, DeleteOpenIDConnectProvider, GetOpenIDConnectProvide required for enabling IAM Roles for Service Accounts (IRSA).
- EKS Cluster Operations
  - eks:DescribeCluster, eks:ListClusters, eks:UpdateClusterConfig, eks:UpdateClusterVersion
  - eks:CreateNodegroup, eks:DeleteNodegroup, eks:CreateAddon, eks:UpdateAddon, eks:DeleteAddon
  - Grants ability to manage EKS clusters, nodegroups, and add-ons.
- PassRole into EKS
  - iam:PassRole on EKSClusterRole
  - Allows workstation EC2 to pass the control plane role to EKS when managing the cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateOpenIDConnectProvider",
        "iam:DeleteOpenIDConnectProvider",
      "iam:UpdateOpenIDConnectProviderThumbprint",
      "iam:GetOpenIDConnectProvider",
      "iam:ListOpenIDConnectProviders",
      "iam:TagOpenIDConnectProvider"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "OIDCProviderManagement"
  },
  {
    "Action": [
      "eks:DescribeCluster",
      "eks:ListClusters",
      "eks:UpdateClusterConfig",
      "eks:UpdateClusterVersion",
      "eks:CreateFargateProfile",
      "eks:DeleteFargateProfile",
      "eks:CreateNodegroup",
      "eks:DeleteNodegroup",
      "eks:CreateAddon",
      "eks:UpdateAddon",
      "eks:DeleteAddon",
      "eks:DescribeClusterVersions",
      "eks:TagResource"
    ],
    "Resource": [
      "arn:aws:eks:us-east-1:account-id:cluster/bft-eks-tiqstg-staging"
    ],
    "Effect": "Allow",
    "Sid": "EksClusterOperations"
  },
  {
    "Action": [
      "eks:DescribeClusterVersions"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EKSOIDCoperation"
  },
  {
    "Action": [
      "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::account-id:role/transfer-iq-stage-EKSStack-FNTRYTC1T-
EKSClusterRole-Fstnu004fq8A"
      ],
      "Effect": "Allow",
      "Sid": "PassEksControlPlaneRole"
    }
  ]
}
```

Security Considerations - Scope restricted to specific EKS cluster ARN wherever possible. Enables OIDC provider creation to support IRSA.

IAM Role: Worker Node Instance Role

Role Name - ${CustomPrefix}-${Environment}-EKSNodeInstanceRole

Trust Policy

- Principal: ec2.amazonaws.com
- Purpose: Allows EKS-managed EC2 worker nodes to assume this role.

Attached Managed Policies

- AmazonEKSWorkerNodePolicy – Worker node access to communicate with EKS control plane.
- AmazonEC2ContainerRegistryReadOnly – Grants pull access from Amazon ECR.
- AmazonEKS_CNI_Policy – Required for VPC CNI plugin to configure networking for pods.

Inline Policy: CloudWatch Logs

- Permissions: logs:CreateLogGroup, logs:CreateLogStream, logs:PutLogEvents, logs:DescribeLogGroups, logs:DescribeLogStreams, logs:GetLogEvents, logs:FilterLogEvents.
- Purpose: Allows worker nodes to publish application and system logs to CloudWatch Logs.

Security Considerations

- Policies follow AWS EKS best practices for worker node roles.
- CloudWatch logs access is broad (*) and could be scoped further in future.

Instance Profiles

- EC2EksInstanceProfile – Binds the WorkstationEC2Role to the workstation EC2 instance for cluster administration.

EKS Access Entry

- Grants AmazonEKSClusterAdminPolicy to the WorkstationEC2Role.
- Provides cluster admin access to the EC2 workstation instance.
- Scope: Cluster-wide

MFT Stack Deployment

This CloudFormation template provisions an AWS Managed File Transfer (MFT) platform that integrates AWS Transfer Family (SFTP), Amazon S3, EventBridge, SQS, ECS Tasks, and GuardDuty.

The stack enables secure file transfers between partners and internal systems, with strong security controls, event-driven automation, and scalable processing.

IAM Role: Transfer family User Role

- Role-Name: ${CustomPrefix}_transfer_family_user_role
- Purpose: Role for SFTP partner users to interact with specific S3 folders.
- Trusted Service: transfer.amazonaws.com
- Key Permissions:
    - List and upload files in specific S3 prefixes (pfts, efts, businessUnit, guardDuty, pfsTemporary).
    - Download and manage (delete, tag) files within those folders.
    - Enforces conditions restricting access to only the expected account and prefixes.

Policy name bft-partner-user-role –

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:PutObject",
        "s3:PutObjectTagging",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev/*",
        "arn:aws:s3:::bft-mft-transferiqdev-dev"
      ],
      "Effect": "Allow"
    }
  ]
}
```

IAM Role: Transfer family CloudWatch role

- Role-name: {CustomPrefix}_transfer_family_cloudwatch-role
- Purpose: Allows AWS Transfer Family SFTP server to write logs into CloudWatch.
- Trusted Service: transfer.amazonaws.com
- Key Permissions:
  - Create and describe log streams and log groups.
  - Publish logs (logs:PutLogEvents) to /aws/transfer/*

Policy name transferiqdev-dev_transfer_family_loggroup_policy –

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:CreateLogGroup",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:us-east-1:account-id:log-group:/aws/transfer/*",
      "Effect": "Allow"
    }
  ]
}
```

IAM Role -GuardDutyS3ProtectionRole

- Role-Name: ${CustomPrefix}-GuardDutyS3ProtectionRole
- Purpose: Role for Amazon GuardDuty Malware Protection on S3 bucket.
- Trusted Service: malware-protection-plan.guardduty.amazonaws.com
- Key Permissions:
    - Manage EventBridge rules for malware detection.
    - Configure S3 bucket notifications.
    - Get and put validation objects for security scans.
    - Read objects for malware scans.

- Policy name GuardDutyS3MalwareProtectionPolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "events:ManagedBy": "malware-protection-plan.guardduty.amazonaws.com"
        }
      },
      "Action": [
        "events:PutRule"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:account-id:rule/DO-NOT-DELETE-
AmazonGuardDutyMalwareProtectionS3*"
      ],
      "Effect": "Allow",
      "Sid": "AllowManagedRuleToSendS3EventsToGuardDuty"
    },
    {
      "Condition": {
        "StringEquals": {
          "events:ManagedBy": "malware-protection-plan.guardduty.amazonaws.com"
        }
      },
      "Action": [
        "events:DeleteRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:account-id:rule/DO-NOT-DELETE-
AmazonGuardDutyMalwareProtectionS3*"
      ],
      "Effect": "Allow",
      "Sid": "AllowUpdateTargetAndDeleteManagedRule"
    },
    {
      "Action": [
        "events:DescribeRule",
        "events:ListTargetsByRule"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:account-id:rule/DO-NOT-DELETE-
AmazonGuardDutyMalwareProtectionS3*"
      ],
      "Effect": "Allow",
      "Sid": "AllowGuardDutyToMonitorEventBridgeManagedRule"
    },
    {
      "Action": [
        "s3:PutBucketNotification",
        "s3:GetBucketNotification"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev"
      ],
      "Effect": "Allow",
      "Sid": "AllowEnableS3EventBridgeEvents"
    },
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev/malware-protection-resource-
validation-object"
      ],
      "Effect": "Allow",
      "Sid": "AllowPutValidationObject"
    },
    {
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev"
      ],
      "Effect": "Allow",
      "Sid": "AllowCheckBucketOwnership"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowMalwareScan"
    }
  ]
}
```

IAM Role: sftp-events-role

- Role-Name: {CustomPrefix}-sftp-events-role
- Purpose: EventBridge role for forwarding SFTP server events to SQS.

- Trusted Service: events.amazonaws.com
- Key Permissions:
  - sqs:SendMessage, sqs:ReceiveMessage to bftSftpServerQueue.
- Inline Policy name – AllowSQSSendMessage

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-1:account-id:transferiqdev-
dev_mft_queue",
      "Effect": "Allow"
    }
  ]
}
```

IAM Role: pfts-push-events-role

- Role-Name: {CustomPrefix}-pfts-push-events-role
- Purpose: Handles asynchronous push events from SFTP server.
- Trusted Service: events.amazonaws.com
- Key Permissions:
  - sqs:SendMessage, sqs:ReceiveMessage to bftSftpServerPfsAsyncQueue
- Inline Policy name – AllowSQSSendMessage

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-1:account-id:transferiqdev-
dev_async_file_transfer_queue",
      "Effect": "Allow"
    }
  ]
}
```

IAM Role -pfts-monitoring-events-role

- Role-Name: {CustomPrefix}-pfts-monitoring-events-role
- Purpose: Handles monitoring events for SFTP data flows.
- Trusted Service: events.amazonaws.com
- Key Permissions:
    - sqs:SendMessage, sqs:ReceiveMessage to bftSftpServerPfsMonitoringQueue.
- Inline Policy name - AllowSQSSendMessage

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_monitoring_file_transfer_queue",
      "Effect": "Allow"
    }
  ]
}
```

IAM Role - pfts-pull-events-role

- Role-Name: {CustomPrefix}-pfts-pull-events-role
- Purpose: Handles pull events (when files are retrieved from SFTP).
- Trusted Service: events.amazonaws.com
- Key Permissions:

    - sqs:SendMessage, sqs:ReceiveMessage to bftSftpServerQueue.
- Inline Policy name – AllowSQSSendMessage

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage"
      ],
      "Resource": "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_mft_queue",
      "Effect": "Allow"
    }
  ]
}
```

IAM Role – ECS fargate TaskExecutionRole

- Role-Name: {CutomPrefix}-TaskExecutionRole
- Purpose: Execution role for ECS tasks that power TransferIQ data movement.
- Trusted Service: ecs-tasks.amazonaws.com
- Key Permissions:
    - SQS: Full messaging support (send, receive, delete).
    - S3: Read/write/delete in specific folders of the MFT bucket.
    - Secrets Manager: Retrieve and create secrets for credentials.
    - CloudWatch Logs: Write application logs.
    - AWS Transfer Family: Test connections and trigger file transfers.
- Inline Policy name – CustomTaskExecutionInlinePolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sqs:SendMessage",
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl"
      ],
      "Resource": [
        "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_mft_queue",
        "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_async_file_transfer_queue",
        "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_monitoring_file_transfer_queue",
        "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_file_delta_checker_queue",
        "arn:aws:sqs:us-east-1:account-id:transferiqdev-dev_dead_letter_queue"
      ],
      "Effect": "Allow",
      "Sid": "SQSPermissions"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket",
        "s3:PutObjectTagging",
        "s3:GetObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::bft-mft-transferiqdev-dev",
        "arn:aws:s3:::bft-mft-transferiqdev-dev/*"
      ],
      "Effect": "Allow",
      "Sid": "S3Permissions"
    },
    {
      "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:TagResource",
        "secretsmanager:PutSecretValue",
        "secretsmanager:CreateSecret"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "SecretsManagerPermissions"
      },
      {
        "Action": [
          "logs:CreateLogStream",
          "logs:PutLogEvents",
          "logs:CreateLogGroup"
        ],
        "Resource": [
          "arn:aws:logs:us-east-1:account-id:log-group:/ecs/transferiqdev-dev-clamAv-logGroup:*",
          "arn:aws:logs:us-east-1:account-id:log-group:/ecs/transferiqdev-dev-wps-logGroup:*",
          "arn:aws:logs:us-east-1:account-id:log-group:/ecs/transferiqdev-dev-pfs-logGroup:*"
        ],
        "Effect": "Allow",
        "Sid": "CloudWatchLogsPermissions"
      },
      {
        "Action": [
          "transfer:TestConnection",
          "transfer:StartFileTransfer",
          "transfer:StartDirectoryListing"
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Sid": "TransferFamilyPermissions"
      }
    ]
}
```

IAM Role - CPUScaleOutAlarmRole

- Role-Name: {CustomPrefix}-CPUScaleOutAlarmRole
- Purpose: Role for CloudWatch alarms & Application Auto Scaling for ECS services.
- Trusted Service: cloudwatch.amazonaws.com, application-autoscaling.amazonaws.com

- Key Permissions:
    - Scale ECS services (ecs:UpdateService).
    - Apply autoscaling policies.
- Inline Policy Name- CPUScaleOutAlarmPolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling:SetDesiredCapacity",
        "application-autoscaling:Describe*",
        "ecs:UpdateService",
        "ecs:DescribeServices"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

IRSA Stack Deployment

This CloudFormation stack provisions an IAM Role for Service Account (IRSA) that integrates with an Amazon EKS cluster. It binds a Kubernetes service account to an IAM role using the cluster's OIDC provider, allowing EKS pods to assume the role via sts:AssumeRoleWithWebIdentity.

The role enables TransferIQ workloads running inside the EKS cluster to securely access AWS services such as S3, KMS, IAM, Transfer Family, Secrets Manager, SSM, SSO, and SQS.

The role is scoped for namespace + service account in Kubernetes and implements least-privilege policies restricted with conditions like aws:ResourceTag/CreatedBy=TransferIQ.

IAM Role: ClusterResourcesAccessRole

Role-name: <CustomPrefix>-resources-access-role
Trust Policy (Assume Role Policy)

- Allows federated access from the EKS OIDC provider.
- Restricted to a specific Kubernetes namespace + service account:
- system:serviceaccount:<Namespace>:<ServiceAccountName>

- Uses sts:AssumeRoleWithWebIdentity for secure short-lived credentials.

The IRSA role has two managed inline policies.

IAM Policy: S3KMSAccessPolicy

Purpose

Provides the role with permissions to read, write, delete, and list objects in the following S3 buckets:

- Assets (S3Assets)
- App Versions (S3AppVersions)
- Executions (S3Executions)
- MFT bucket (MFTBucketName)

Also grants KMS encryption/decryption rights for a specific KMS key (KMSKeyArn).

Key Actions

- S3: GetObject, PutObject, DeleteObject, ListBucket, AbortMultipartUpload, GetObjectTagging, PutObjectTagging.
- KMS: Encrypt, Decrypt, ReEncrypt*, GenerateDataKey*, DescribeKey.

Use Case

- Store and retrieve artifacts, configurations, logs, and transfer data.
- Encrypt/decrypt sensitive data in MFT processes.

Policy name - S3KMSAccessPolicy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::bft-s3-assets-transferiqdev-dev",
        "arn:aws:s3:::bft-s3-assets-transferiqdev-dev/*",
        "arn:aws:s3:::bft-s3-app-versions-transferiqdev-dev",
        "arn:aws:s3:::bft-s3-app-versions-transferiqdev-dev/*",
        "arn:aws:s3:::bft-s3-executions-transferiqdev-dev",
        "arn:aws:s3:::bft-s3-executions-transferiqdev-dev/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-east-1:account-id:key/5b3bce41-365e-481e-a0aa-34d1155b3cc3",
      "Effect": "Allow",
      "Sid": "AllowExternalKMS"
    }
  ]
}
```

IAM Policy: SDKIntegrationPermissions Policy

This policy provides a broad set of AWS service permissions that TransferIQ components need.

a. S3 (Extended Operations)

- List all buckets, get objects, and write objects with tagging and ACLs.
- Supports full integration with TransferIQ storage requirements.

b. IAM (Role & Policy Management)

- List and inspect roles/policies.
- Create/tag/attach policies and roles with prefix tiq*.
- Pass roles to AWS services like Transfer Family, Lambda, ECS tasks, SSM, SSO, Secrets Manager.

Ensures TransferIQ can dynamically provision IAM roles/policies needed for workflows.

c. AWS Transfer Family

- Manage users, connectors, agreements, profiles, servers.
- Required for core TransferIQ file transfer workflows.

d. AWS Secrets Manager

- Create and manage secrets, replicate to regions, update values.
- Scoped to secrets tagged or prefixed for TransferIQ.

- Used for credentials and integration endpoints.

e. AWS Systems Manager (SSM)

- Get/put parameters, add/remove tags.
- Supports configuration management.

f. AWS KMS

- Encrypt/decrypt data keys, tag keys.
- Supports secure data handling.

g. AWS SSO & Identity Store

- Create and manage SSO permission sets.
- List users and groups in Identity Store.
- Supports identity integration for managed transfers.

h. AWS STS

- Get caller identity (for validation and introspection).

i. AWS Resource Tagging API

- Tag/untag resources to maintain governance.

j. AWS EventBridge Scheduler

- Create, update, and get schedules with prefix tiq*.
- Supports scheduled transfers.

k. AWS SQS

- Send/receive/delete messages, manage queues.
- Used for asynchronous transfer workflows.

Policy Name – SDKIntegrationPermissions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject",
        "s3:PutObjectTagging",
        "sts:GetCallerIdentity",
        "iam:ListPolicies",
        "iam:ListRoles",
        "iam:CreateRole",
        "iam:ListRolePolicies",
        "iam:PassRole",
        "iam:PutRolePermissionsBoundary",
        "iam:TagRole",
        "iam:GetSAMLProvider",
        "iam:CreatePolicy",
        "iam:TagPolicy",
        "iam:AttachRolePolicy",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "transfer:CreateUser",
        "transfer:UpdateUser",
        "transfer:ImportSshPublicKey",
        "transfer:ListServers",
        "transfer:DescribeServer",
        "transfer:DescribeUser",
        "transfer:TagResource",
        "transfer:CreateConnector",
        "transfer:CreateAgreement",
        "transfer:CreateProfile",
        "transfer:UpdateCertificate",
        "transfer:UpdateConnector",
        "transfer:UpdateProfile",
        "transfer:ListAgreements",
        "transfer:ListCertificates",
        "transfer:ListConnectors",
        "transfer:ListProfiles",
        "transfer:UpdateAgreement",
        "transfer:TestConnection",
        "secretsmanager:CreateSecret",
        "secretsmanager:ReplicateSecretToRegions",
        "secretsmanager:TagResource",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "secretsmanager:GetResourcePolicy",
        "secretsmanager:PutResourcePolicy",
        "ssm:GetParameter",
        "ssm:PutParameter",
        "ssm:AddTagsToResource",
        "kms:GenerateDataKey",
        "kms:Encrypt",
        "kms:Decrypt",
        "sso:ListInstances",
        "sso:CreatePermissionSet",
        "sso:AttachManagedPolicyToPermissionSet",
        "sso:AttachCustomerManagedPolicyReferenceToPermissionSet",
        "sso:PutInlinePolicyToPermissionSet",
        "sso:CreateAccountAssignment",
        "sso:ProvisionPermissionSet",
        "sso:DescribeAccountAssignmentCreationStatus",
        "sso:TagResource",
        "identitystore:ListUsers",
        "identitystore:ListGroups",
        "tag:TagResources",
        "tag:UntagResources",
        "scheduler:CreateSchedule",
        "scheduler:UpdateSchedule",
        "scheduler:GetSchedule",
        "sqs:SendMessage"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

IAM Roles and Policies for ALB Controller

This IAM Role for AWS Application Load Balancer (ALB) Controller into an existing EKS cluster. It leverages IAM Roles for Service Accounts (IRSA) to securely grant Kubernetes workloads only the minimum AWS permissions required, following the principle of least privilege.

The key IAM resource created here is the AlbIRSArole, which provides the ALB controller running in the EKS cluster the ability to manage AWS load balancers and associated networking/security resources.

IAM Role: AlbIRSArole

Role-Name: AlbIRSArole-{Random-characters}

- Type: AWS::IAM::Role

- Purpose: This role is assumed by the Kubernetes service account aws-load-balancer-controller in the kube-system namespace using OIDC federation (via sts:AssumeRoleWithWebIdentity).

- The ALB controller pod in EKS does not have IAM user credentials. Instead, it securely assumes this IAM role to perform AWS API operations necessary to provision and manage Application Load Balancers.

IAM Policy: AlbControllerPolicy

This inline policy grants the ALB controller specific permissions. Each section is designed to enable a set of operations tied to load balancer provisioning, networking, tagging, and security integration.

a. Service-Linked Role Creation

- Action: iam:CreateServiceLinkedRole
- Why: ALB requires a service-linked role (AWSServiceRoleForElasticLoadBalancing) to operate.

b. ELB and EC2 Describe Permissions

- Action: Describes ELB resources, listeners, target groups, VPCs, subnets, security groups, and EC2 networking.
- Why: The controller must query AWS infrastructure to determine where to create and attach load balancers.

c. WAF, Shield, ACM, and IAM Certificates

- Action: wafv2:AssociateWebACL, shield:CreateProtection, acm:DescribeCertificate, iam:GetServerCertificate, etc.
- Why: Enables the controller to integrate with Web Application Firewall (WAF), Shield DDoS protection, and TLS certificates for HTTPS listeners.

d. Security Group Management

- Action: Create, delete, modify, and tag security groups.
- Why: Load balancers require dynamically managed security groups for ingress/egress rules.

e. Target Group & Load Balancer Management

- Action: elasticloadbalancing:CreateTargetGroup, CreateLoadBalancer, ModifyTargetGroup, DeleteLoadBalancer, etc.

- Why: Core permissions that let the controller create ALBs, configure listeners, and manage backend target groups.

f. Tagging Operations

- Action: ec2:CreateTags, elasticloadbalancing:AddTags, RemoveTags
- Why: Kubernetes uses tags (elbv2.k8s.aws/cluster) to track and manage resources created by the ALB controller.

g. Listener, Rule, and Certificate Modifications

- Action: Create/modify/delete listeners, rules, attach/detach SSL certificates.
- Why: Required to support Ingress routing rules and HTTPS termination.

h. Register/Deregister Targets

- Action: elasticloadbalancing:RegisterTargets, DeregisterTargets
- Why: Maps Kubernetes pods/services to ALB target groups.

Policy –

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"
        }
      },
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowCreateServiceLinkedRole"
    },
    {
      "Action": [
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetGroupAttributes",
        "elasticloadbalancing:DescribeTags",
        "elasticloadbalancing:DescribeSSLPolicies",
        "elasticloadbalancing:DescribeRules",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeLoadBalancerAttributes",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeListenerCertificates",
        "ec2:GetCoipPoolUsage",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcPeeringConnections",
        "ec2:DescribeTags",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeInternetGateways",
        "ec2:DescribeInstances",
        "ec2:DescribeCoipPools",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeAddresses",
        "ec2:DescribeAccountAttributes"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowELBDescribeActions"
    },
    {
      "Action": [
        "wafv2:GetWebACLForResource",
        "wafv2:GetWebACL",
        "wafv2:DisassociateWebACL",
        "wafv2:AssociateWebACL",
        "waf-regional:GetWebACLForResource",
        "waf-regional:GetWebACL",
        "waf-regional:DisassociateWebACL",
        "waf-regional:AssociateWebACL",
        "shield:GetSubscriptionState",
        "shield:DescribeProtection",
        "shield:DeleteProtection",
        "shield:CreateProtection",
        "iam:ListServerCertificates",
        "iam:GetServerCertificate",
        "cognito-idp:DescribeUserPoolClient",
        "acm:ListCertificates",
        "acm:DescribeCertificate"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowWAFAndShieldAndACM"
    },
    {
      "Action": [
        "ec2:RevokeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowSecurityGroupChanges"
    },
    {
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateSecurityGroup"
        },
        "Null": {
          "aws:RequestTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Effect": "Allow",
      "Sid": "AllowCreateTagsOnSecurityGroup"
    },
    {
      "Condition": {
        "Null": {
          "aws:RequestTag/elbv2.k8s.aws/cluster": "true",
          "aws:ResourceTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": [
        "ec2:DeleteTags",
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Effect": "Allow",
      "Sid": "AllowModifyTagsOnSecurityGroup"
    },
    {
      "Condition": {
        "Null": {
          "aws:ResourceTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": [
        "ec2:RevokeSecurityGroupIngress",
        "ec2:DeleteSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowSecurityGroupIngressChanges"
    },
    {
      "Condition": {
        "Null": {
          "aws:RequestTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": [
        "elasticloadbalancing:CreateTargetGroup",
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowCreateTargetGroupAndLoadBalancer"
    },
    {
      "Action": [
        "elasticloadbalancing:DeleteRule",
        "elasticloadbalancing:DeleteListener",
        "elasticloadbalancing:CreateRule",
        "elasticloadbalancing:CreateListener",
        "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowRuleAndListenerActions"
    },
    {
      "Condition": {
        "Null": {
          "aws:RequestTag/elbv2.k8s.aws/cluster": "true",
          "aws:ResourceTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": [
        "elasticloadbalancing:RemoveTags",
        "elasticloadbalancing:AddTags"
      ],
      "Resource": [
        "arn:aws:elasticloadbalancing:*:*:targetgroup/*/*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/net/*/*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/app/*/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowTagRemovalAndAdditionOnTargetsAndLoadBalancers"
    },
    {
      "Action": [
        "elasticloadbalancing:RemoveTags",
        "elasticloadbalancing:AddTags"
      ],
      "Resource": [
        "arn:aws:elasticloadbalancing:*:*:listener/net/*/*/*",
        "arn:aws:elasticloadbalancing:*:*:listener/app/*/*/*",
        "arn:aws:elasticloadbalancing:*:*:listener-rule/net/*/*/*",
        "arn:aws:elasticloadbalancing:*:*:listener-rule/app/*/*/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowTagRemovalAndAdditionOnListenersAndRules"
    },
    {
      "Condition": {
        "Null": {
          "aws:ResourceTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": [
        "elasticloadbalancing:SetSubnets",
        "elasticloadbalancing:SetSecurityGroups",
        "elasticloadbalancing:SetIpAddressType",
        "elasticloadbalancing:ModifyTargetGroupAttributes",
        "elasticloadbalancing:ModifyTargetGroup",
        "elasticloadbalancing:ModifyLoadBalancerAttributes",
        "elasticloadbalancing:DeleteTargetGroup",
        "elasticloadbalancing:DeleteLoadBalancer"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowModifyLoadBalancerAttributes"
    },
    {
      "Condition": {
        "StringEquals": {
          "elasticloadbalancing:CreateAction": [
            "CreateTargetGroup",
            "CreateLoadBalancer"
          ]
        },
        "Null": {
          "aws:RequestTag/elbv2.k8s.aws/cluster": "false"
        }
      },
      "Action": "elasticloadbalancing:AddTags",
      "Resource": [
        "arn:aws:elasticloadbalancing:*:*:targetgroup/*/*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/net/*/*",
        "arn:aws:elasticloadbalancing:*:*:loadbalancer/app/*/*"
      ],
      "Effect": "Allow",
      "Sid": "AllowAddTagsOnCreateActions"
    },
    {
      "Action": [
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets"
      ],
      "Resource": "arn:aws:elasticloadbalancing:*:*:targetgroup/*/*",
      "Effect": "Allow",
      "Sid": "AllowRegisterDeregisterTargets"
    },
    {
      "Action": [
        "elasticloadbalancing:SetWebAcl",
        "elasticloadbalancing:RemoveListenerCertificates",
        "elasticloadbalancing:ModifyRule",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:AddListenerCertificates"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "AllowWebAclAndListenerCertsModification"
    }
  ]
}
```

Conclusion

The provided IAM design establishes a comprehensive, end-to-end security and operations model for the TransferIQ platform across CloudFormation-driven provisioning, S3 data domains, EKS orchestration, Transfer Family (SFTP), event-driven workflows (EventBridge/SQS), ECS task execution, GuardDuty Malware Protection, and IRSA-based least-privilege access within Kubernetes. The separation of concerns across roles (deployment, control-plane, workstation/admin, worker nodes, application tasks, and service-specific roles) is clear, and critical controls such as TLS-only S3 access, private VPC access to S3 via endpoints, scoped KMS usage, and fine-grained Transfer Family permissions are implemented.

Alignment with AWS Prescriptive Guidance .

This implementation follows the AWS recommendations for least privilege in CloudFormation-provisioned resources by:

- Identifying resources to be deployed (e.g., Transfer Family, S3, EKS).
- Reviewing service-specific permissions (IAM, S3, Transfer Family, EKS).
- Designing policies that restrict actions to those strictly required for functionality.
- Applying conditions to further constrain access (e.g., by prefix, principal, account ID).

Conclusion

The IAM roles and policies provisioned through CloudFormation have been explicitly designed to align with the Principle of Least Privilege. Each role grants the minimal set of permissions necessary for its intended function, avoids unnecessary wildcards, and applies contextual restrictions.

This approach reduces security exposure, prevents misuse, and ensures compliance with AWS best practices for IAM policy design.

Security

Root Privileges not required for deployment

TranferIQ product uses AWS Identity and Access Management (IAM) roles and policies to securely control all access to AWS resources needed for deployment and daily operation.

- IAM roles specific to AWS Transfer Family (or your MFT solution) provide only the minimum necessary permissions for users, servers, workflows, and logging.
- All user accounts, service-managed users, and workflow users are assigned *least privilege* IAM roles, never relying on root account credentials or privileges.
- Service actions (such as access to Amazon S3, CloudWatch logging, Lambda functions, and workflow execution) are managed through clearly scoped IAM roles, with trust relationships only between authorized services and resources.

Public S3 Buckets & Resource Policies

TransferIQ uses four Amazon S3 buckets. All the four S3 buckets are private with no public access.

Stored Secrets

- TransferIQ deployment requires a Docker token to pull container images from Docker Hub. During the CloudFormation stack creation, the Docker token is provided as a parameter. Once the stack is successfully created, the token is securely stored in AWS Parameter store for future use. This ensures that sensitive credentials are not exposed in plaintext and are managed according to AWS security best practices.
- In TransferIQ, a pair of PGP encryption and decryption are generated and stored securely in AWS Secrets Manager. These keys are used to encrypt and decrypt files during data transfer, ensuring that file contents remain protected and accessible only to authorized processes.
- In TransferIQ, customer SFTP server details such as the SSH private key and username are securely stored in AWS Secrets Manager. This ensures that sensitive connection information remains encrypted, centrally managed and accessible only to authorized services.

Sensitive Data Storage

In TransferIQ, customer SFTP server details such as the SSH private key and username are securely stored in AWS Secrets Manager. This ensures that sensitive connection information remains encrypted, centrally managed and accessible only to authorized services.

Data Encryption Configuration

TransferIQ ensures that sensitive customer information is protected through encryption. Customer passwords stored in AWS DocumentDB are encrypted using AWS Key Management Service (KMS). This integration with KMS provides secure key storage, centralized management and strong encryption.

Health Check

Monitoring Application Health

To monitor the health of the TransferIQ application, a dedicated health check endpoint is available at: https://<domain-name>/health

This endpoint provides real-time status information, including:

- dbStatus – indicates the health of the database connection.
- redisStatus – indicates the health of the Redis cache.
- overallStatus – reflects overall health of the application.

Documentation of Keys in Deployment Guide

- TransferIQ deployment requires a Docker token to pull container images from Docker Hub. During the CloudFormation stack creation, the Docker token is provided as a parameter. Once the stack is successfully created, the token is securely stored in AWS Secrets Manager for future use. This ensures that sensitive credentials are not exposed in plaintext and are managed according to AWS security best practices.
- In TransferIQ, a pair of PGP encryption and decryption are generated and stored securely in AWS Secrets Manager. These keys are used to encrypt and decrypt files during data transfer, ensuring that file contents remain protected and accessible only to authorized processes.
- In TransferIQ, customer SFTP server details such as the SSH private key and username are securely stored in AWS Secrets Manager. This ensures that sensitive connection information remains encrypted, centrally managed and accessible only to authorized services.

Guidelines for Managing AWS Service Limits

This section provides guidance on validating and managing AWS service limits for the Backflipt TransferIQ solution. For a standard deployment, the resources created by the solution are well within default AWS account limits. Quota issues are not expected. However, customers should validate quotas in their accounts to avoid deployment or operational disruptions.

Services Within Default Quotas:

The solution uses several AWS services including VPC, EC2, EKS, ECS, SQS, S3, Systems Manager, CloudWatch, GuardDuty, Secrets Manager, EventBridge, KMS, ElastiCache, DocumentDB, IAM Identity Center, and Route 53.

For these services, the number of resources provisioned by the solution is significantly below the default service quotas. For example, the deployment creates only a few subnets, EC2 instances, queues, and S3 buckets, whereas default quotas allow hundreds or thousands of such resources. Accordingly, quota increases are not expected for these services under normal operation. Customers are advised to confirm quotas in advance as a best practice.

Services with Potential Quota Impact

AWS Identity and Access Management (IAM)

The application creates IAM roles and policies through the application. Although the expected number of IAM resources remains within the default account quotas, there is potential to approach these limits over time. Customers should monitor the number of roles and policies in their account and request a quota increase if role or policy counts approach the account limit.

AWS Transfer Family

The solution provisions one AWS Transfer Family server (SFTP/AS2) and automatically creates connectors through the application. Under typical usage, the number of connectors will remain below the default service quotas. If connector counts increase significantly, a quota increase will be required.

Adjustable vs. Non-Adjustable Quotas

Some quotas can be increased through the Service Quotas console or by submitting an AWS Support case (for example, EC2 vCPU quotas or IAM role counts). Other quotas are fixed and cannot be raised (for example, the limit of 100 S3 buckets per account). In such cases, customers should ensure that the target AWS account has sufficient quota to support the resources required by this solution. For instance, this solution requires four S3 buckets, which is well below the default limit of 100 per account. Customers should select an account where the required resources can be created.

Conclusion

This solution is not expected to exceed any AWS service quotas under standard deployment and operation. Prescriptive steps are provided to validate quotas before deployment, with specific guidance for IAM and AWS Transfer Family. By following this guidance, customers can ensure that service quota limits will not interfere with deployment or ongoing operation.

Emergency Maintenance

Fault Handling, Software Recovery, and Troubleshooting Instructions

Recovering from CloudFormation Stack Failures:

- Issue: If the CloudFormation stack fails during creation, it will show a status of "CREATE_FAILED."
- Recovery Steps:
- Monitor Stack Creation: View the stack events in the AWS Management Console to identify which resource caused the failure.
- Rollback and Reattempt: If a failure occurs, select "Rollback all stack resources and delete newly created resources" in the rollback options to clean up.
- Recreate the Stack: After identifying and correcting the issue, such as a configuration error, re-run the CloudFormation stack creation process with updated parameters.

S3 Bucket Name Conflict (Not Unique):

- Issue: If the chosen S3 bucket name is not unique, the deployment will fail during stack creation.
- Recovery Steps:
- Check for Conflicting Names: Ensure the S3 bucket name is globally unique.
- Change the Bucket Name: If the bucket name conflict is detected, update the name in the CloudFormation template and reattempt the stack creation.

Elastic IP Limit Reached:

- Issue: AWS limits the number of Elastic IPs that can be allocated to an account. If the account exceeds this limit, provisioning Elastic IPs will fail.
- Recovery Steps:
- Release Unused Elastic IPs: Go to the EC2 dashboard in AWS and release any unused Elastic IPs.
- Request Quota Increase: If more Elastic IPs are needed, request an increase through the AWS Service Quotas dashboard.
- Reattempt Stack Creation: After resolving the Elastic IP issue, reattempt the CloudFormation stack creation or update the load balancer configuration.

DNS Mapping and Load Balancer Issues:

- Issue: Problems with DNS name resolution or load balancer configuration might occur during installation.
- Recovery Steps:

- Check Load Balancer Status: Ensure the load balancer has been successfully created and that the DNS name matches the TransferIQ URL provided during installation.
- Recreate DNS Records: If necessary, reconfigure DNS records in Route 53 to ensure the TransferIQ URL points to the correct load balancer.

IAM Role and Policy Issues:

- Issue: Insufficient IAM role permissions can cause failures during stack creation or post-installation steps.
- Recovery Steps:
- Review IAM Policies: Ensure that the IAM roles and policies (like TiQ-cfs-assume-role) have the correct permissions as specified in the deployment guide.
- Attach Missing Permissions: If missing permissions are found, attach the required IAM policies and reattempt the failed process.

Post-Installation Failures (Application Launch, Configuration Errors):

- Issue: Post-installation tasks may fail due to incorrect configuration settings.
- Recovery Steps:
- Re-import Orchestrate App: If the Orchestrate app import fails, retry importing the app after ensuring the correct BSON file is selected.
- Recheck Environment Variables: Ensure that the environment variables for AWSRegion, tiqBucketName, etc., are correctly configured.
- Re-run Metadata Settings Payload: Execute the cURL command again with the correct values for your environment to configure the metadata settings.

Troubleshooting CloudWatch Logs:

- Issue: If there are application-specific issues or system errors, CloudWatch logs might provide insights into what went wrong.
- Recovery Steps:
- Review Logs: Use Amazon CloudWatch to access application logs and identify any errors or exceptions.
- Adjust Configurations Based on Log Findings: Depending on the log data, adjust configurations (like IAM policies, S3 bucket settings, etc.) and restart the affected components.

Reconfigure DNS and Custom Domain:

- Issue: DNS records or custom domains might be misconfigured, leading to accessibility issues.
- Recovery Steps:
- Edit Application DNS: If DNS mapping fails, recheck and reconfigure the custom domain in the application settings.

- Ensure Correct Subdomain Configuration: Make sure the subdomain set during the CloudFormation stack is correctly mapped to the TransferIQ Application URL.

Re-Executing the Metadata Payload via cURL:

- Issue: If the metadata payload for the application configuration is not executed correctly, the app might not function as expected.
- Recovery Steps:
- Re-execute the Payload Command: Use the provided cURL command to push the correct metadata settings to the TransferIQ application, ensuring the necessary environment-specific values are populated.
- These recovery steps are meant to guide you in resolving common issues that might arise during the installation or configuration of the Backflipt TransferIQ solution.

Support

Backflipt Standard Support Model

- Backflipt's Standard Support model includes Phone and Email support.
- Email – Support@backflipt.com
- Phone – 408-890-2032
- Between 7 am PST to 5 pm PST

SLA for Support			
Level of Severity	Description of Severity	Characteristics	Response Time
Level 1 - Critical	Critical Business Impact: Critical issue occurring on production system preventing business operations. A large number of users are prevented from working with no procedural workaround.	1. System hangs or crashes 2. Critical functionality not available 3. Large number of end users blocked from work 4. Impact is escalating quickly	Issues received between 7 AM to 5 PM PST on a business day will be acknowledged in 1hr. Team will embark on resolving these issues as top priority
Level 2 - Major	Significant Business Impact: Major issue occurring on production system severely impacting business. A large number of users are impacted by issue but they are still able to work in a limited capacity.	1. Significant performance degradation 2. Important functionality not available 3. Small number of users blocked from work 4. Impact is escalating	Issues received between 7 AM to 5 PM PST on a business day will be acknowledged in 4 hrs. Team will embark on resolving these issues if there are no other pending Level 1 issues
Level 3 - Medium	Normal Business Impact: Issue causing a partial or	1. Some system functions not available	Issues received between 7 AM to 5 PM PST on a

	non-critical loss of functionality on production system. A small number of users are affected.	2. Minor performance degradation	business day will be acknowledged in one business day. Team will embark on resolving these issues if there are no other pending Level 1, Level 2 issues
		3. Small number of users impacted	
		4. Impact is not escalating	
Level 4 - Low	Minimal Business Impact: Issue occurring on non-production system or question, comment, feature request, documentation issue or other non-impacting issue.	1. Incorrect product behavior without impact	Issues received between 7 AM to 5 PM PST on a business day will be acknowledged in three business day. Team will embark on resolving these issues if there are no other pending Level 1, Level 2, Level 3 issues
		2. Product question or enhancement	