Data perimeter for Amazon Bedrock: Securing generative AI workloads

AWS Prescriptive Guidance



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Prescriptive Guidance: Data perimeter for Amazon Bedrock: Securing generative AI workloads

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Services or capabilities described in AWS documentation might vary by Region. To see the differences applicable to the AWS European Sovereign Cloud Region, see the AWS European Sovereign Cloud User Guide.

Table of Contents

Introduction **1**
Overview 1
Data perimeter objectives for Amazon Bedrock 2
Business Outcome 6
Architecture overview **7**
Implementation guide **9**
Core Implementation 9
Prerequisites 9
AWS environment requirements: 9
Replace placeholder values 10
Getting started 11
Identity perimeter 12
Organizational boundary enforcement 13
Training data protection 15
Knowledge base data isolation 18
Sensitive log data protection 20
AWS KMS key policies 22
AWS Lambda resource policies 31
Amazon VPC endpoint policies 32
Cross account sharing 33
Monitoring violations 35
Best practices and recommendations 36
Resource perimeter 36
Model access controls 38
Service principals 40
Cross-service integration controls 43
Resource tagging strategy 45
Identity tagging controls 47
Network resource access controls 50
Monitoring violations 53
Best practices and recommendations 54
Network perimeter 55
VPC endpoint configuration 56
Network segmentation for AI workloads 57

Regional boundary enforcement 60
Network access control lists (NACLs) 62
Knowledge base network controls 63
VPC endpoint network controls 65
Monitoring network access patterns 67
Best practices and recommendations 68
Conclusion 69
Resources 72
AWS documentation 72
Operational best practices 72
AWS blogs and articles 72
Related APG guides 72
AWS samples and tools 73
Community resources 73
Document history 74

Data perimeter for Amazon Bedrock: Securing generative AI workloads

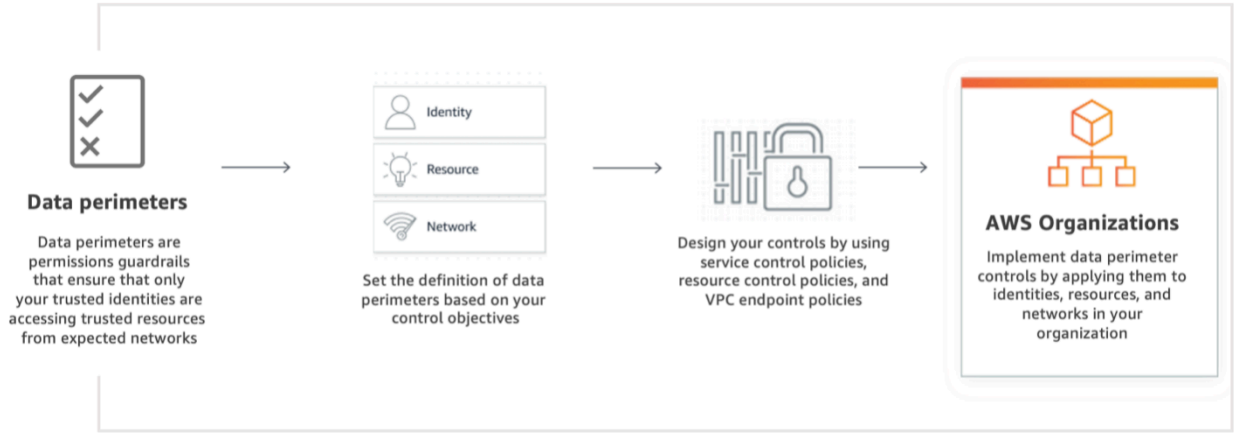
Prashanth Sivarajan and Manigandan Shri, Amazon Web Services

April 2026 (document history)

Data Perimeter on AWS is a security strategy and set of controls designed to help ensure that only trusted identities can access trusted resources from expected networks within your AWS environment. It acts as a robust, coarse-grained boundary around your data, complementing fine-grained access controls and helping remediate unauthorized data access and exfiltration.

This article provides a comprehensive guide to address the AI-specific requirements by establishing perimeter controls around foundation models, custom models, knowledge bases, and the data that flows between them. It provides a structured implementation approach using Amazon Bedrock-specific policies and monitoring to ensure your AI applications operate within defined organizational boundaries.

This guidance is intended for Cloud architects and security engineers building a secure-by-default AI solution with Amazon Bedrock on AWS.



Overview

Amazon Bedrock workloads have unique characteristics that benefit from specialized data perimeter approaches. When you invoke foundation models, your prompts may contain business data, custom models store training datasets, and knowledge bases aggregate organizational

documents. These AI-specific assets have different access patterns and data flows compared to traditional AWS resources.

Amazon Bedrock workloads often involve distributed architectures spanning multiple accounts for model sharing, automated pipelines for data processing, and real-time inference endpoints.

This distributed nature creates additional considerations for implementing effective perimeter controls.

A data perimeter for Amazon Bedrock addresses these challenges through three AI-focused objectives:

Data perimeter objectives for Amazon Bedrock

Objective	Definition	Applied on	Using	Implementation Examples
Identity Perimeter	Only trusted identities can access my resources	Resources	Resource-based policy	• Organizational boundary enforcement • Training data Amazon S3 bucket protection • Amazon CloudWatch logs access control • AWS KMS key policies for encryption • AWS Lambda function invocation restricted to Amazon

			Bedrock Agents
Only trusted identities are allowed from my network	Network	• Amazon VPC endpoint policy	• Amazon VPC endpoint policy restricts access to protected resources only for trusted entities. • Knowledge base data isolation (Amazon OpenSearch/ Amazon S3)

Resource Perimeter	My identities can access only trusted resources	Identities	Service Control Policies (SCP)	• Model-specific access controls (foundation and custom models) • IAM Path-based environment separation • Service principal restrictions with confused deputy protection • Cross-service integration controls (AWS Lambda, Amazon DynamoDB, Amazon S3) • Tag-based resource access (ABAC)
	Only trusted resources can be accessed from my network	Network	• Amazon VPC endpoint policy	• Amazon VPC endpoint policy restricts access to protected resources only for trusted entities.

Network Perimeter	My identities can access resources only from expected networks	Identities	Service control policies (SCP)	• Amazon VPC endpoint enforcement for Amazon Bedrock API calls • Deny public internet access to models
	My resources can only be accessed from expected networks	Resources	Resource-based policy	• Knowledge base network controls (Amazon OpenSearch domain and Amazon S3vector store VPC endpoint enforcement) • Amazon VPC endpoint network access controls for Amazon Bedrock resources

These preventive guardrails are critical for Amazon Bedrock deployments where sensitive prompts, training datasets, and model artifacts require protection from unauthorized access and data exfiltration.

For example, a healthcare organization using Amazon Bedrock to analyze patient records needs to ensure:

- Their custom medical models are only accessible by authorized clinical applications
- Their AI systems never send data to external models
- All patient prompts travel through HIPAA-compliant network paths

Business Outcome

Implementing these controls provides clear visibility into data access patterns, streamlines compliance reporting, and establishes consistent security policies across AI workloads. This creates a foundation for scaling AI initiatives while maintaining operational control and meeting regulatory requirements. The approach enables teams to innovate confidently while maintaining security baseline.

Architecture overview

In a security context, a perimeter provides a clear boundary of trust and ownership. For AWS environments, the perimeter is typically represented as a collection of resources in a group of AWS accounts managed by AWS Organizations. This architecture establishes preventive controls that work together to protect your AI workloads from unauthorized access and data exfiltration.

Identity layer architecture - This layer controls which principals can perform AI operations. Unlike traditional applications where users directly access resources, Amazon Bedrock workloads often involve service-to-service communication, cross-account model sharing, and automated AI pipelines.

The identity perimeter must account for human users invoking models through applications, automated systems processing data through AI workflows, and cross-account scenarios where different AWS accounts share custom models or knowledge bases.

Resource layer architecture - AI workloads interact with numerous AWS resources beyond Amazon Bedrock itself. Custom model training requires Amazon S3 buckets for datasets, knowledge bases use Amazon OpenSearch or vector databases, and model artifacts are stored across multiple services.

The resource perimeter ensures your AI identities only access trusted resources within your organization while preventing external resources from accessing your AI data.

Network layer architecture - Amazon Bedrock API calls carry prompts and responses that benefit from private connectivity. The network perimeter establishes private connectivity through VPC endpoints and ensures AI traffic flows through your designated network paths. This is particularly important for real-time model invocations where you want to maintain consistent network routing and security posture.

Amazon Bedrock-specific policy implementation: The AI data perimeter can be applied using three primary types of AWS security features.

- **Service control policies for AI resources** - Restrict access to Amazon Bedrock models, knowledge bases, and custom model endpoints using `bedrock:*` actions combined with `aws:ResourceOrgID` conditions to prevent unauthorized model access

- **Resource control policies for AI data** - Protect Amazon S3 buckets containing training datasets, Amazon OpenSearch clusters powering knowledge bases, and model artifact storage using `aws:PrincipalOrgID` to ensure only trusted AI applications can access your data
- **VPC endpoint policies for AI traffic** - Secure Amazon Bedrock API calls carrying sensitive prompts and responses through private connectivity, with policies that validate both the calling application and target model are within your trust boundary

These policies address Amazon Bedrock's unique service integrations, such as when the service automatically accesses your Amazon S3 buckets for model training or when knowledge bases query your Amazon OpenSearch clusters during inference.

Implementation guide

This guide is organized into focused sections that build upon each other to create comprehensive data perimeter protection for your Amazon Bedrock workloads.

Core Implementation

1. **Identity perimeter** - Establish who can access your AI resources
2. **Resource perimeter** - Control who has access to the resource and what actions they can perform on it
3. **Network perimeter** - Ensure traffic flows through expected network paths

Each section provides detailed implementation guidance with code examples, best practices, and specific considerations for AI workloads. Start with the identity perimeter as it provides the broadest protection, then layer on resource and network controls based on your specific requirements.

Prerequisites

Before implementing the controls in this guide, ensure you have:

Required knowledge:

- Understanding of AWS Identity and Access Management (IAM) policies and service control policies (SCPs)
- Familiarity with AWS Organizations and organizational units
- Basic knowledge of Amazon Bedrock services and features
- Understanding of VPC networking concepts

AWS environment requirements:

- AWS Organizations enabled with multiple accounts (recommended)
- Appropriate IAM permissions to invoke and configure Amazon Bedrock features.

- Appropriate IAMpermissions to create and modify SCPs, resource policies, and VPC configurations
- AWS CloudTrail logging enabled for audit and monitoring

Replace placeholder values

Throughout this guide, policy examples use placeholder values that you must replace with your actual AWS resource identifiers:

- `o-1234567890` - Replace with your AWS organization ID (find using: `aws organizations describe-organization`)
- `123456789012` - Replace with your AWS account ID(s)
- `us-east-1` - Replace with your deployment region(s)
- `bedrock-training-data-bucket` - Replace with your actual Amazon S3 bucket names
- `vpc-12345678` - Replace with your VPC IDs
- `subnet-ai-workloads-1a` - Replace with your subnet IDs
- `sg-bedrock-ai-workloads` - Replace with your security group IDs

Regional considerations:

Most policy examples in this guide reference us-east-1 as the AWS region. When implementing these policies in your environment:

- Adapt region references to match your deployment regions
- Consider multi-region deployments and ensure policies cover all regions where you operate Amazon Bedrock workloads
- Be aware that someAmazon Bedrock features and models may have different regional availability
- For data residency requirements, explicitly restrict operations to approved regions using `aws:RequestedRegion` condition keys

Getting started

Your Amazon Bedrock data perimeter implementation follows six foundational building blocks. Each step below provides the essential framework, with detailed implementation guidance, code examples, and best practices covered in the subsequent sections:

1. **Assess your AI environment** - Inventory your foundation models, custom models, knowledge bases, and AI applications. Map data flows from training datasets through inference endpoints to understand your current architecture
2. **Control model access** - Implement service control policies (SCPs) with Amazon Bedrock-specific actions (bedrock:InvokeModel, `bedrock:CreateModelCustomizationJob`) and `aws:ResourceOrgID` conditions to ensure access to your custom models and knowledge bases follows organizational policies
3. **Protect AI data sources** - Deploy resource based policies on resources such as Amazon S3 buckets that contain training data, Amazon OpenSearch Service clusters powering knowledge bases, and other AI data stores using `aws:PrincipalOrgID` to ensure only trusted AI applications can access your datasets
4. **Establish private AI connectivity** - Configure VPC endpoints for Amazon Bedrock with policies that route prompts and model responses through your private networks, particularly important for real-time inference workloads
5. **Monitor AI operations** - Set up AWS CloudTrail logging for Amazon Bedrock API calls and create alerts for model invocations, cross-account model access, and network access patterns that fall outside your defined policies
6. **Test with AI workloads** - Validate your controls support legitimate model training pipelines, knowledge base updates, and inference applications while enforcing your organizational policies

The following sections expand on each building block with step-by-step implementation instructions, policy templates, and monitoring configurations to help you establish a comprehensive data perimeter for your AI workloads.

Note - Implementation considerations

- Apply SCP policies at the root organizational unit to ensure coverage across all accounts
- Test the policies in a non-production environment first to identify any legitimate cross-organization access needs

- Document any exceptions and implement them through explicit allow policies rather than modifying the organizational boundary

Identity perimeter

The objective for identity perimeter is to ensure that only trusted identities can access your resources.

Identity perimeter for Amazon Bedrock requires more nuanced controls than traditional applications because AI workloads involve multiple types of principals with varying access patterns. You must secure human users invoking models through applications, service principals performing automated AI operations, and cross-account scenarios where external entities need limited access to specific models.

This diagram is a high level illustration of an identity perimeter.

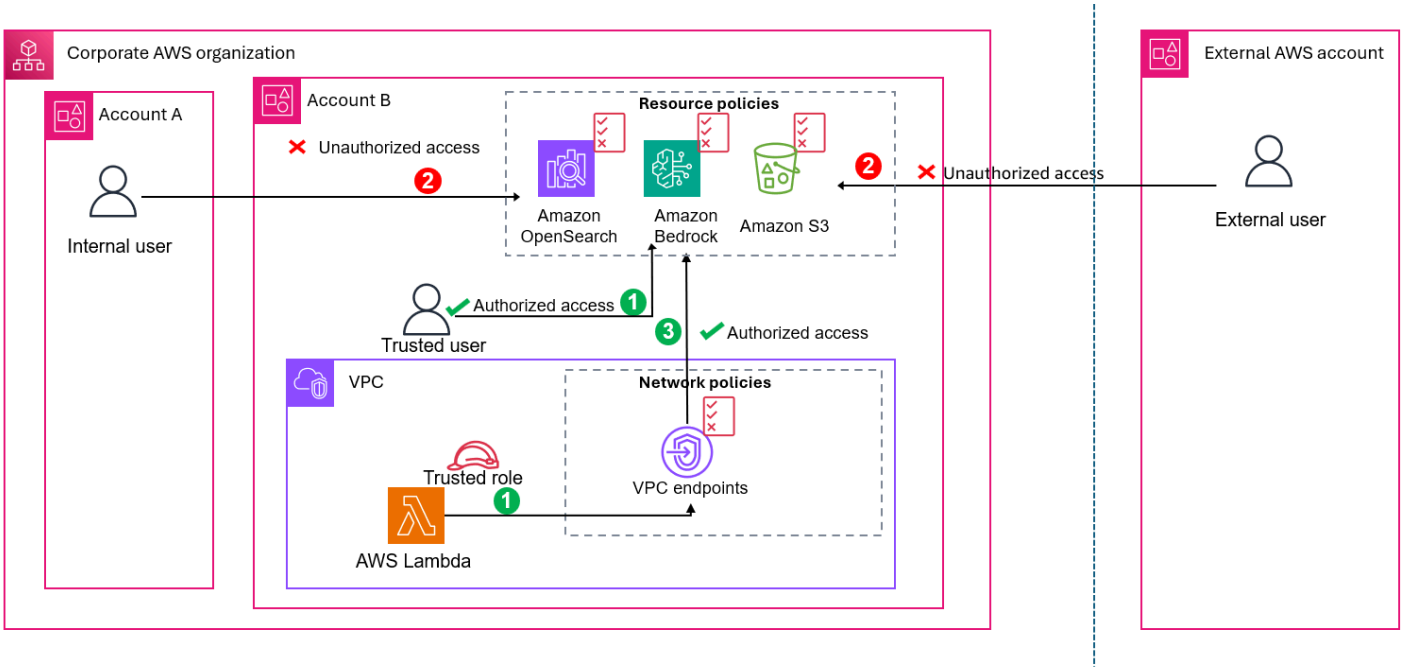


Figure 3.1 High level illustration of identity perimeter

1. Resource policies allow only trusted identities from your AWS organization to accessAmazon Bedrock resources

2. Resource policies deny access from external AWS accounts and untrusted principals

3. VPC endpoint policies allow access only to organizational identities

This section covers the essential components of implementing identity perimeter controls for Amazon Bedrock:

- Organizational boundary - Implementing organization-wide restrictions using `aws:PrincipalOrgID`
- Training data protection - Securing Amazon S3 buckets and datasets used for model training
- Knowledge base data isolation - Protecting data sources and vector stores with resource policies
- Sensitive log data protection - Securing Amazon CloudWatch logs containing AI interactions
- AWS KMS key policies - Implementing encryption-based access controls for AI resources
- AWS Lambda resource policies - Controlling service access to Lambda functions in AI workflows
- VPC endpoint policies - Managing identity access through private network endpoints
- Cross-account sharing - Securing multi-account scenarios while maintaining proper identity boundaries
- Monitoring violations - Detecting and responding to identity perimeter breaches
- Best practices - Identity perimeter implementation recommendations and guidelines

Each subsection provides practical implementation guidance, code examples, and best practices specific to Amazon Bedrock's unique security requirements.

Note - These identity perimeter examples provide foundational patterns but may not cover all organizational structures or authentication scenarios. Adapt the policies to your specific identity providers, compliance requirements, and multi-account architecture. Always test identity controls in non-production environments and consult IAM documentation for the latest policy syntax and condition keys.

Organizational boundary enforcement

Control objective

Identity perimeter *- Only trusted identities can access my resources*

Start by implementing organization-wide restrictions using the `aws:PrincipalOrgID` condition key to prevent any principal outside your AWS organization from accessing your Amazon Bedrock

resources, even if they somehow obtain valid credentials. Apply the following policy as a service control policy (SCP) at the organization level to ensure consistent enforcement across all accounts.

> **Important**
>
> Replace `o-1234567890` with your actual AWS organization ID in all policy examples below. Find your organization ID by running: `aws organizations describe-organization`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictToOrganization",
      "Effect": "Deny",
      "Action": "bedrock:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    }
  ]
}
```

Policy explanation:

- **RestrictToOrganization** - Denies all Amazon Bedrock operations from principals outside your AWS organization, creating a foundational security boundary that prevents external access even with valid credentials.

The null condition is crucial because it prevents bypass attempts where the `aws:PrincipalOrgID` key might be absent. This policy should be applied as a service control policy (SCP) at the organization level to ensure consistent enforcement across all accounts.

Training data protection

Control objective

***Identity perimeter**: Only trusted identities can access my resources*

Custom model training requires uploading datasets to Amazon S3. Only trusted identities can access training data resources by applying the following policy to Amazon S3 buckets containing training data.

Important – Replace placeholder values:

- `o-1234567890` with your AWS organization ID
- `123456789012` with your AWS account IDs
- `bedrock-training-data-bucket` with your actual bucket names

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyExternalAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::bedrock-training-data-bucket/*",
        "arn:aws:s3:::bedrock-training-data-bucket"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    },
    {
      "Sid": "PreventDataExfiltration",
      "Effect": "Deny",
```

```
      "Principal": "*",
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete"
      ],
      "Resource": "arn:aws:s3:::bedrock-training-data-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        }
      }
    }
  ]
}
```

Policy explanation:

- **DenyExternalAccess** - Blocks all Amazon S3 operations from principals outside your organization
- **PreventDataExfiltration** - Specifically denies replication operations that could be used to copy training data to external buckets

Enhanced training data protection:

For highly sensitive training datasets, require encryption and restrict access to specific roles:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyUnauthorizedRoles",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::sensitive-training-data/*",
      "Condition": {
        "ForAllValues:StringNotEquals": {
          "aws:PrincipalArn": [
            "arn:aws:iam::123456789012:role/BedrockTrainingRole",
            "arn:aws:iam::123456789012:role/DataScientistRole"
```

```
          ]
        }
      }
    },
    {
      "Sid": "EnforceSecureTransport",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::sensitive-training-data",
        "arn:aws:s3:::sensitive-training-data/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      }
    },
    {
      "Sid": "DenyUnencryptedUploads",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::sensitive-training-data/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "aws:kms"
        }
      }
    }
  ]
}
```

Policy explanation:

- **DenyUnauthorizedRoles** – Explicitly denies access to sensitive training data for any principal that is not one of the designated training and data science roles
- **EnforceSecureTransport** – Denies all Amazon S3 actions when secure transport (HTTPS) is not used
- **DenyUnencryptedUploads** – Enforces AWS KMS encryption for all uploads

Key protections:

- Organizational boundary prevents external access
- Replication blocked to prevent data exfiltration
- AWS KMS encryption required for uploads
- HTTPS transport enforced
- Access limited to specific IAM roles

Knowledge base data isolation

Control objective

Identity perimeter *– Only trusted identities can access my resources*

Knowledge bases often contain the most sensitive organizational data. Only trusted identities can access knowledge base resources by establishing controls that restrict access to authorized services only.

For OpenSearch-backed knowledge bases, apply the following domain access policy to the OpenSearch domain resource:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/BedrockKnowledgeBaseRole"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:123456789012:domain/bedrock-knowledge-base/*"
    }
  ]
}
```

Policy explanation:

- OpenSearch **domain access**: Restricts OpenSearch access to the Amazon Bedrock Knowledge Base role only

Amazon S3 vector store access control

For Amazon S3-backed vector stores, control access through bucket policies that enforce organizational boundaries:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBedrockKnowledgeBaseAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/BedrockKnowledgeBaseRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bedrock-vector-store-bucket",
        "arn:aws:s3:::bedrock-vector-store-bucket/*"
      ]
    },
    {
      "Sid": "DenyExternalAccountAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::bedrock-vector-store-bucket",
        "arn:aws:s3:::bedrock-vector-store-bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "123456789012"
        }
      }
    }
  ]
}
```

Policy explanation:

- **Authorized service access** – Restricts Amazon S3bucket access to the designated Amazon Bedrock Knowledge Base role only
- **Cross-account boundary** – Denies access from external AWS accounts

> **Note**
>
> – For network-based access controls (VPC endpoint enforcement), see the Network Perimeter section.

Sensitive log data protection

Control objective

Identity Perimeter – Only trusted identities can access my resources

AI workloads generate logs that may contain sensitive prompts, responses, or model parameters. Only trusted identities can access log resources by applying the following resource policy to Amazon CloudWatch log groups containing Amazon Bedrock logs:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOrganizationRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "logs:DescribeLogStreams",
        "logs:GetLogEvents",
        "logs:FilterLogEvents"
      ],
      "Resource": "arn:aws:logs:*:123456789012:log-group:/aws/bedrock/*",
      "Condition": {
        "StringEquals": {
```

```
          "aws:PrincipalOrgID": "o-1234567890"
        }
      }
    }
  ]
}
```

Policy explanation:

- **AllowOrganizationRead** - Only organization members can read Amazon Bedrock logs

AWS KMS key policy for CloudWatch Logs encryption

Encrypt Amazon Bedrockmodel invocation logs and restrict decryption to organization members only:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnableRootAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchLogsAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "logs.us-east-1.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
```

```
      "Condition": {
        "ArnEquals": {
          "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:us-
east-1:123456789012:log-group:/aws/bedrock/*"
        }
      }
    },
    {
      "Sid": "OrganizationOnlyDecrypt",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "kms:Decrypt",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        }
      }
    ]
}
```

Policy explanation:

- **EnableRootAccess** – Grants full AWS KMS permissions to the account root for key administration and management
- **CloudWatchLogsAccess** – Allows CloudWatch Logs service to encrypt and decrypt log data, with encryption context ensuring the key is only used for Amazon Bedrock log groups
- **OrganizationOnlyDecrypt** – Restricts decryption access to principals within your AWS organization, preventing external entities from reading encrypted log data

AWS KMS key policies

Control objective

***Identity perimeter** – Only trusted identities can access my resources*

AWS KMS key policies ensure only trusted identities can access encrypted resources. These policies complement other controls and act as a critical enforcement point for encrypted resources.

Knowledge Base encryption architecture

Amazon Bedrock Knowledge Bases encrypts data at multiple layers:

Amazon S3 **documents** – Documents uploaded to Amazon S3 buckets (used by Knowledge Base data sources) can be encrypted with customer-managed AWS KMS keys

Transient processing – During data ingestion, Amazon Bedrock encrypts temporary data with a customer-managed key you specify

Vector storage – Amazon OpenSearch Serverless collections orAmazon S3 Vectors can use customer-managed keys

Retrieval sessions – `RetrieveAndGenerate` API calls can encrypt session context with a customer-managed key

Knowledge Base Amazon S3 documents encryption key

Apply the followingAWS KMS key policy to keys used for Amazon S3 buckets containing documents for Knowledge Base data sources:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "KMSAdminAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/KMSAdminRole"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "BedrockKnowledgeBaseAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
      "Action": [
```

```
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012",
          "kms:ViaService": "s3.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "DocumentUploadRoleAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/KnowledgeBaseDocumentUploadRole",
          "arn:aws:iam::123456789012:role/DataIngestionRole"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyExternalAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    }
  ]
}
```

Policy explanation:

KMSAdminAccess – Grants full AWS KMS permissions to a dedicated administrative role rather than the account root. This follows the principle of least privilege by restricting key management to specific roles.

BedrockKnowledgeBaseAccess – Allows Amazon Bedrock service to decryptAmazon S3 objects for Knowledge Base operations. The condition restricts access to your account and ensures the request comes through Amazon S3, preventing direct key usage.

DocumentUploadRoleAccess – Grants specific IAM roles permission to encrypt documents when uploading to Amazon S3. Limited to encryption operations only, preventing these roles from accessing existing encrypted data.

DenyExternalAccess – Explicitly denies all access from principals outside your AWS organization. This creates an organizational boundary that prevents external access even if other conditions are met.

Knowledge Base Data store transient data encryption key

Apply the following AWS KMS key policy to keys used for transient data storage during ingestion and retrieval sessions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "KMSAdminAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/KMSAdminRole"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "BedrockOperationsAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
```

```
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:CreateGrant"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    },
    {
      "Sid": "DenyExternalAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    }
  ]
}
```

Policy explanation:

KMSAdminAccess – Provides administrative access to the AWS KMS key for the designated admin role, enabling key management operations while maintaining separation from operational access.

BedrockOperationsAccess – Grants Amazon Bedrock comprehensive key operations needed for transient data processing during ingestion and retrieval. Includes `CreateGrant` permission for delegating temporary access to other services.

DenyExternalAccess – Enforces organizational boundary by denying all operations from principals outside your AWS organization, regardless of other permissions.

Knowledge Base vector store encryption key

Apply the following AWS KMS key policy to keys used for OpenSearch Serverless collections or S3 Vectors:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "KMSAdminAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/KMSAdminRole"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "OpenSearchServerlessAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "aoss.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    },
    {
      "Sid": "S3VectorsAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
```

```
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    },
    {
      "Sid": "DenyExternalAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    }
  ]
}
```

Policy explanation

KMSAdminAccess – Provides full key management capabilities to the designated administrative role, ensuring centralized key governance while avoiding account root access.

OpenSearchServerlessAccess – Grants OpenSearch Serverless the encryption operations needed for vector storage. Scoped to your account to prevent cross-account access through the service.

S3VectorsAccess – Allows Amazon S3 service to perform encryption operations when using Amazon S3 as vector storage backend. Account-scoped to prevent unauthorized cross-account vector access.

DenyExternalAccess – Creates an organizational perimeter by explicitly denying all access from outside your AWS organization, providing defense-in-depth security.

Model customization encryption

Amazon Bedrock supports model customization through fine-tuning, continued pre-training, and distillation. These operations require access to training datasets stored in Amazon S3 buckets. Unlike Knowledge Base operations that only read documents, model customization jobs require additional AWS KMS permissions including `CreateGrant` to enable Amazon Bedrock to perform long-running training operations.

Training data encryption key

Apply the following AWS KMS key policy to keys used for Amazon S3 buckets containing model training datasets and fine-tuning data:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "KMSAdminAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/KMSAdminRole"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "BedrockTrainingAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```
          "aws:SourceAccount": "123456789012",
          "kms:ViaService": "s3.us-east-1.amazonaws.com"
        }
      }
    },
    {
      "Sid": "TrainingRoleAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/BedrockTrainingRole",
          "arn:aws:iam::123456789012:role/DataScientistRole"
        ]
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyExternalAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        },
        "Null": {
          "aws:PrincipalOrgID": "false"
        }
      }
    }
  ]
}
```

Policy explanation

- **BedrockTrainingAccess** – Allows Amazon Bedrock to decrypt training data and create grants for long-running training jobs (via Amazon S3).

- **TrainingRoleAccess** – Allows training and data science roles to encrypt/decrypt training datasets.
- **DenyExternalAccess** – Blocks all access from principals outside the organization.

AWS Lambda resource policies

Control objective

Identity perimeter *– Only trusted identities can access my resources*

Control which services can invoke AWS Lambda functions by applying the following resource policy to AWS Lambda functions used with Amazon Bedrock Agents:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBedrockAgentInvoke",
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:bedrock-agent-*",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:bedrock:us-east-1:123456789012:agent/*"
        }
      }
    }
  ]
}
```

Policy explanation:

AllowBedrockAgentInvoke – Allows only Amazon Bedrock Agents service to invoke AWS Lambda functions, with additional conditions to prevent confused deputy attacks. The source account

condition ensures the request originates from your account, while the source ARN condition validates the request comes from a legitimateAmazon Bedrock Agent.

Amazon VPC endpoint policies

Control objective

Identity perimeter *– Only trusted identities are allowed from my network*

Control which identities can access Amazon Bedrock through VPC endpoints by attaching the following policy to the Amazon Bedrock VPC endpoint.

Important – Replace `o-1234567890` with your AWS organization ID.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOrganizationAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream",
        "bedrock:CreateKnowledgeBase",
        "bedrock:GetKnowledgeBase"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "o-1234567890"
        }
      }
    }
  ]
}
```

Policy explanation:

AllowOrganizationAccess – Permits Amazon Bedrock API calls only from identities within your AWSorganization, creating an identity boundary at the VPC endpoint level.

Note

– For network-based VPC endpoint controls, see the Network Perimeter section.

Cross account sharing

Control objective

Identity perimeter *– Only trusted identities can access my resources*

When sharing custom models across accounts, implement explicit trust relationships with time-bound access and specific use case restrictions by applying the following resource-based policy to the custom model. For example, a parent company might share a specialized legal document analysis model with subsidiaries, but only for specific document types and with audit logging enabled.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SubsidiaryModelAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::555666777888:role/LegalDocumentProcessor"
      },
      "Action": "bedrock:InvokeModel",
      "Resource": "arn:aws:bedrock:*:123456789012:custom-model/legal-analysis-model",
      "Condition": {
        "StringEquals": {
          "bedrock:ModelId": "legal-analysis-model"
        },
        "StringLike": {
          "aws:userid": "*:LegalTeam*"
        },
        "DateLessThan": {
          "aws:CurrentTime": "2026-06-30T23:59:59Z"
        }
      }
    }
  ]
```

```
}
```

Policy explanation:

- **SubsidiaryModelAccess** – Grants time-bound access to a custom legal analysis model for a specific subsidiary role, with additional validation that the user session belongs to the legal team.

Multi-subsidiary sharing pattern

For organizations with multiple subsidiaries requiring access to shared models, apply the following resource-based policy to the shared custom model:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MultiSubsidiaryAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::555666777888:role/SubsidiaryAIRole",
          "arn:aws:iam::999888777666:role/SubsidiaryAIRole",
          "arn:aws:iam::111222333444:role/SubsidiaryAIRole"
        ]
      },
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:GetCustomModel"
      ],
      "Resource": "arn:aws:bedrock:*:123456789012:custom-model/shared-analytics-model",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": ["us-east-1", "eu-west-1"]
        },
        "StringLike": {
          "aws:PrincipalTag/BusinessUnit": ["Finance", "Operations", "Legal"]
        },
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
```

```
    }
  ]
}
```

Policy explanation:

- **MultiSubsidiaryAccess** – Allows multiple subsidiary accounts to access a shared analytics model with regional restrictions, business unit validation, and mandatory HTTPS transport for secure cross-account model sharing

Monitoring violations

Control objective

Identity perimeter *– Only trusted identities can access my resources*

Set up Amazon CloudWatch alarms and EventBridge rules to detect identity perimeter violations by creating an EventBridge rule that monitors AWS CloudTrail events for failed Amazon Bedrock API calls:

```
# Create EventBridge rule to detect unauthorized Bedrock access attempts
aws events put-rule \
  --name BedrockUnauthorizedAccess \
  --description "Alert on unauthorized Bedrock access attempts" \
  --event-pattern '{
    "source": ["aws.bedrock"],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail": {
      "eventName": ["InvokeModel", "InvokeModelWithResponseStream"],
      "errorCode": ["AccessDenied", "UnauthorizedOperation"]
    }
  }'

# Configure SNS target for security alerts
aws events put-targets \
  --rule BedrockUnauthorizedAccess \
  --targets '[{"Id":"1","Arn":"arn:aws:sns:us-east-1:123456789012:security-alerts"}]'
```

Event pattern (for reference):

```
{
```

```
  "source": ["aws.bedrock"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": {
    "eventName": ["InvokeModel", "InvokeModelWithResponseStream"],
    "errorCode": ["AccessDenied", "UnauthorizedOperation"]
  }
}
```

Policy explanation:

- **Event pattern** – Detects failed Amazon Bedrock API calls that indicate potential identity perimeter violations, triggering alerts when unauthorized access attempts occur.

Best practices and recommendations

1. **Principle of least privilege** – Grant only the minimum permissions required for each use case.
2. **Regular access reviews** – Conduct quarterly reviews of cross-account access and model permissions.
3. **Automated policy testing** – Implement automated tests to verify policy effectiveness.
4. **Documentation** – Maintain clear documentation of all identity perimeter exceptions and their business justifications.
5. **Gradual rollout** – Implement identity controls incrementally, starting with the most sensitive workloads.

Resource perimeter

The resource perimeter for Amazon Bedrock extends beyond the service itself to encompass all resources that store, process, or transmit AI-related data. This includes Amazon S3 buckets containing training datasets, OpenSearchclusters powering knowledge bases, and even CloudWatch logs that might contain sensitive prompts or responses.

This diagram is a high level illustration of a resource perimeter.

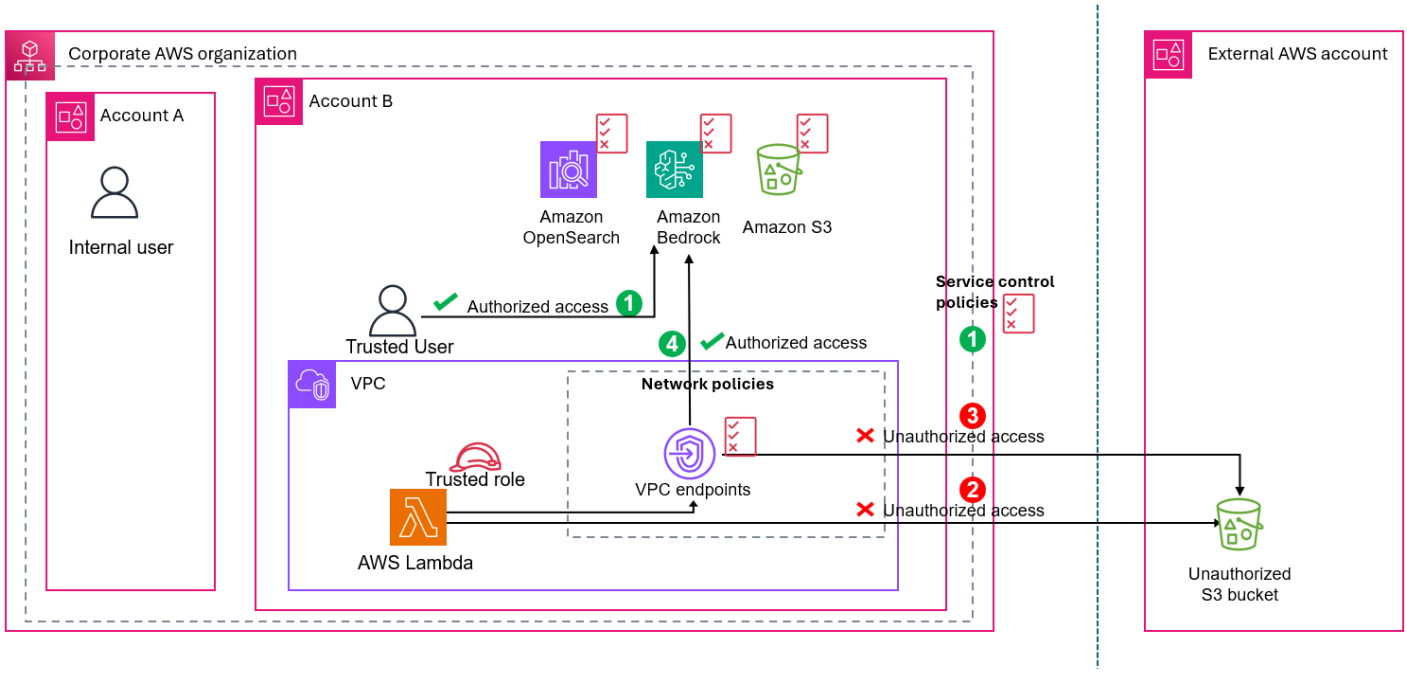


Figure 3.2 High level illustration of resource perimeter

1. Service control policies (SCPs) allow only approved Amazon Bedrock models in approved regions.

2. SCP denies all access to untrusted external Resources

3. Amazon VPC endpoint policies deny access to external resources outside organizational boundaries

4. Amazon VPC endpoint policy allows access only to approved resources with with required tags

This section covers the comprehensive resource protection strategies for Amazon Bedrock:

- **Model access controls** - Implementing fine-grained AWS IAM policies for different Amazon Bedrock models
- **Service orincipals** - Managing automated access patterns and service-to-service authentication
- **Cross-service integration controls** - Managing resource access across integrated AWS services
- **Resource tagging strategy** - Implementing consistent tagging for resource-based access controls
- **Identity tagging controls** - IAM policies for attribute-based resource access control
- **Network resource access controls** - Network-based resource access restrictions
- **Monitoring violations** - Detecting and responding to resource perimeter breaches
- **Best practices** - Resource perimeter implementation recommendations and guidelines

Each subsection provides detailed implementation guidance and security controls specific to protecting AI-related resources.

Note – These resource protection examples demonstrate common patterns but may not address all data types, integration scenarios, or regulatory requirements in your environment. Conduct a thorough data classification and risk assessment for your specific AI workloads. Consult AWS service documentation for the latest encryption options, policy syntax, and integration patterns with your existing security controls.

Model access controls

Control objective

Resource perimeter *– My identities can access only trusted resources*

Different foundation models and custom models require different access patterns. My identities can access only trusted resources by implementing granular policies that restrict access to specific models based on the principal's role and intended use case:

Model-specific access policy (IAM **policy):**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowApprovedModelsInApprovedRegions",
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream",
        "bedrock:ListFoundationModels"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": ["us-east-1", "us-west-2"],
          "bedrock:ModelId": [
            "anthropic.claude-3-sonnet-20240229-v1:0",
            "customer-service-model"
          ]
        }
      }
```

```
    },
    {
      "Sid": "ExplicitDenyOtherRegions",
      "Effect": "Deny",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": ["us-east-1", "us-west-2"]
        }
      }
    }
  ]
}
```

Policy explanation:

- **AllowApprovedModelsInApprovedRegions** – Grants access to specific Amazon Bedrock models while restricting operations to approved regions
- **ExplicitDenyOtherRegions** – Explicitly denies model invocations in regions outside the approved list

Path-based access patterns (IAM policy)

Implement environment separation through role paths by attaching the following IAM policy to roles that need path-based model access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ProductionBedrockAccess",
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": [
```

```
        "arn:aws:bedrock:*::foundation-model/anthropic.claude-3-sonnet-20240229-v1:0"
      ],
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/production/*"
        }
      }
    },
    {
      "Sid": "DevelopmentBedrockAccess",
      "Effect": "Allow",
      "Action": "bedrock:InvokeModel",
      "Resource": [
        "arn:aws:bedrock:*::foundation-model/anthropic.claude-3-haiku-20240307-v1:0"
      ],
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/development/*"
        }
      }
    }
  ]
}
```

Policy explanation:

- **ProductionBedrockAccess** – Grants production roles access to the Claude Sonnet model with streaming capabilities for high-performance production workloads
- **DevelopmentBedrockAccess** – Restricts development roles to the Claude Haiku model without streaming, providing cost-effective access for testing and development

Create roles with paths like `/production/AppName` or `/development/AppName`. Paths are immutable and visible in CloudTrail, making them ideal for environment separation.

Service principals

Control Objective

Resource perimeter *– My identities can access only trusted resources*

Amazon Bedrockoften operates through service principals when integrated with other AWS services. Restrict these service principals to only the necessary actions and resources, and protect against confused deputy attacks using `aws:SourceAccount` and `aws:SourceArn` conditions.

Bedrock Agents service role:

Apply this trust policy to the IAM service role used by Amazon Bedrock Agents:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "bedrock.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:bedrock:*:123456789012:agent/*"
        }
      }
    }
  ]
}
```

Policy explanation:

- **Trust policy** – Allows Amazon Bedrock service to assume this role only when the request originates from your account and from a legitimate Amazon Bedrock agent, preventing confused deputy attacks
- **Confused deputy protection** – The `aws:SourceAccount` and `aws:SourceArn` conditions in the trust policy prevent confused deputy attacks where a malicious actor tricks the service into assuming your role. Always include both conditions when creating service roles.

Apply this permissions policy to the Amazon Bedrock agents service role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BedrockAgentModelAccess",
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": [
        "arn:aws:bedrock:*::foundation-model/anthropic.claude-3-sonnet-20240229-v1:0"
      ]
    }
  ]
}
```

Policy explanation:

- **BedrockAgentModelAccess** – Grants the Amazon Bedrock agent service role access to invoke the Claude Sonnet model with streaming capabilities for agent operations

AWS Lambda integration with Amazon Bedrock

Apply these trust policies to Lambda execution roles that need to invoke Bedrock models:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:lambda:*:123456789012:function:bedrock-*"
        }
```

```
      }
    }
  ]
}
```

Policy explanation:

- **Trust policy** – Allows Lambda to assume this role only for functions with the "bedrock-" prefix in your account, preventing unauthorized Lambda functions from accessing Amazon Bedrockresources

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LambdaBedrockAccess",
      "Effect": "Allow",
      "Action": "bedrock:InvokeModel",
      "Resource": "arn:aws:bedrock:*::foundation-model/amazon.titan-text-express-v1"
    }
  ]
}
```

Policy explanation:

- **LambdaBedrockAccess** – Grants Lambda functions permission to invoke only the Titan Text Express model, implementing least privilege by restricting access to a single, cost-effective model for Lambda integrations
- **Confused deputy protection** – The `aws:SourceAccount` and `aws:SourceArn` conditions in the trust policy prevent confused deputy attacks where a malicious actor tricks the service into assuming your role. Always include both conditions when creating service roles.

Cross-service integration controls

Control objective

***Resource perimeter**: My identities can access only trusted resources*

Amazon Bedrock integrates with numerous AWS services, each requiring specific resource perimeter controls. To enforce least privilege, explicitly allow only the needed actions on specific resources by applying this IAM policy to Lambda execution roles used with Amazon Bedrock agents.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOnlyTrustedResources",
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:Query",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-3-
sonnet-20240229-v1:0",
        "arn:aws:dynamodb:us-east-1:123456789012:table/bedrock-agent-tableName",
        "arn:aws:s3:::bedrock-agent-data/prod/*"
      ]
    },
    {
      "Sid": "DenyExternalOrganizationAccess",
      "Effect": "Deny",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "StringNotEquals": {
          "s3:ResourceOrgID": "o-1234567890"
        },
        "Null": {
          "s3:ResourceOrgID": "false"
        }
      }
```

```
    },
    {
      "Sid": "DenyUnencryptedUploads",
      "Effect": "Deny",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bedrock-agent-data-*/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "aws:kms"
        }
      }
    }
  ]
}
```

Policy explanation:

AllowOnlyTrustedResources – Grants Lambda functions access to specific Amazon Bedrock models, Amazon DynamoDB tables, and Amazon S3 buckets using resource-based restrictions. This statement implements least privilege by limiting access to only resources with the bedrock-agent-* prefix, preventing access to unrelated resources in the account.

DenyExternalOrganizationAccess – Prevents data exfiltration by explicitly denying Amazon S3 access to any bucket outside your AWS organization. The condition checks for the organization ID and denies access when the bucket belongs to a different organization or has no organization ID set.

DenyUnencryptedUploads – Enforces encryption-in-transit by denying any Amazon S3 `PutObject` operation that doesn't use AWS KMS encryption. This ensures all data written to Amazon S3 by Lambda functions is encrypted with customer-managed keys.

Resource tagging strategy

Control objective

***Resource perimeter** – My identities can access only trusted resources*

Implement a comprehensive tagging strategy to support resource perimeter controls. This example shows a recommended tag structure:

```
{
  "RequiredTags": {
```

```
    "DataClassification": ["Public", "Internal", "Confidential", "Restricted"],
    "Purpose": ["Training", "Inference", "KnowledgeBase", "Monitoring"],
    "Owner": ["TeamName"],
    "Environment": ["Development", "Staging", "Production"],
    "CostCenter": ["BusinessUnit"]
  }
}
```

Tag-based access control example (S3 bucket policy):

Control access to sensitive data based on resource and principal tags by applying this bucket policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictedDataAccess",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::bedrock-data-bucket/*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/DataClassification": "Restricted"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/DataClearance": "Restricted"
        }
      }
    }
  ]
}
```

This policy uses pure ABAC (attribute-based access control) - any principal tagged with `DataClearance=Restricted` can access objects tagged `DataClassification=Restricted`, without hardcoding specific role names.

> **Note**
>
> – For identity-based tagging policies that control which resources identities can access, see Identity tagging controls in this section.

Key characteristics of resource-based tagging policies

S3 Bucket Policy (Resource-based):

- **Attached to** – Amazon S3 bucket (controls who can access the bucket)
- **Scope** – Only applies to that specific Amazon S3 bucket
- **Principal** – Uses principal: "*" to apply to all callers
- **Enforcement** – Bucket-level protection - even if IAM allows access, bucket policy can deny
- **Use case** – Protecting specific data repositories from unauthorized access
- **Perimeter** – Identity Perimeter (controls which identities can access resources)

Identity tagging controls

Control objective

Resource perimeter *– My identities can access only trusted resources*

Control which resources can be accessed by identities based on principal and resource tags using IAM policies that implement attribute-based access control (ABAC).

Tag-based resource access control

Grant data clearance levels to specific roles by attaching this IAM policy to roles that need restricted data access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRestrictedDataAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/DataClassification": "Restricted",
          "aws:PrincipalTag/DataClearance": "Restricted"
```

```
        }
      }
    },
    {
      "Sid": "DenyRestrictedDataWithoutClearance",
      "Effect": "Deny",
      "Action": "s3:GetObject",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "s3:ExistingObjectTag/DataClassification": "Restricted"
        },
        "StringNotEquals": {
          "aws:PrincipalTag/DataClearance": "Restricted"
        }
      }
    }
  ]
}
```

Amazon Bedrock model access by classification

Control access to different Amazon Bedrock models based on data classification tags:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowModelAccessByClassification",
      "Effect": "Allow",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/DataClearance": "Restricted"
        },
        "ForAnyValue:StringLike": {
          "bedrock:modelId": [
            "anthropic.claude-3-sonnet*",
            "amazon.titan-text-express*"
```

```
          ]
        }
      }
    },
    {
      "Sid": "DenyHighValueModelsWithoutClearance",
      "Effect": "Deny",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalTag/DataClearance": "Restricted"
        },
        "ForAnyValue:StringLike": {
          "bedrock:modelId": [
            "anthropic.claude-3-opus*"
          ]
        }
      }
    }
  ]
}
```

Policy explanation:

IAM **policy (identity-based) characteristics:**

- **Attached to** – IAM roles/users (controls what the role can do)
- **Scope** – Applies to all resources the role tries to access
- **Principal** – No principal field - applies to whoever assumes the role
- **Enforcement** – Role-level permissions - defines what tagged principals can access
- **Use case** – Granting data clearance levels to specific roles
- **Perimeter** – Resource perimeter (controls which resources identities can access)

Tag-based access patterns:

1. **Principal tags** – `aws:PrincipalTag/DataClearance` defines the identity's clearance level

2. **Resource tags** – `s3:ExistingObjectTag/DataClassification` defines the resource's sensitivity
3. **Model restrictions** – `bedrock:modelId` controls access to specific AI models
4. **ABAC enforcement** – Both principal and resource tags must match for access

Combined effect with resource-based policies:

1. IAM policy grants permissions based on principal tags (resource perimeter)
2. Amazon S3 bucket policy enforces resource-level restrictions (identity perimeter)
3. Both must allow access for the operation to succeed

Network resource access controls

Control objective

Resource perimeter *– Only trusted resources can be accessed from my network*

Prevent your network from accessing resources outside your trusted organizational boundary by implementing Amazon VPC endpoint policies that restrict access to resources within your AWSorganization. This ensures that even if identities are compromised, network traffic can only reach approved resources.

Amazon Bedrock **VPC endpoint policy:**

Apply this policy to your Amazon Bedrock VPC endpoint to ensure only organizational resources can be accessed:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowApprovedModelsInTrustedOrg",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
```

```
      "Resource": [
        "arn:aws:bedrock:*::foundation-model/anthropic.claude-3-sonnet-20240229-v1:0",
        "arn:aws:bedrock:*::foundation-model/amazon.titan-text-express-v1"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgID": "o-example123456"
        }
      }
    },
    {
      "Sid": "AllowModelListing",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "bedrock:GetFoundationModel",
        "bedrock:ListFoundationModels"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgID": "o-example123456"
        }
      }
    }
  ]
}
```

Policy explanation:

- **AllowApprovedModelsInTrustedOrg** – Permits access only to specific approved models (Claude Sonnet, Titan Text Express) within your organization, blocking unauthorized foundation models
- **AllowModelListing** – Allows listing and getting model information for organizational resources while maintaining the organizational boundary

Amazon S3 VPC endpoint policy for AI data:

Apply this policy to your Amazon S3 VPC endpoint to restrict access to training data and knowledge base documents:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggedBedrockBuckets",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::*",
        "arn:aws:s3:::*/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgID": "o-example123456"
        },
        "ForAnyValue:StringEquals": {
          "aws:ResourceTag/Purpose": [
            "BedrockTraining",
            "BedrockKnowledgeBase"
          ]
        }
      }
    }
  ]
}
```

Policy explanation:

- **AllowTaggedBedrockBuckets** – Restricts access to Amazon S3 buckets within your organization that are tagged with `Purpose=BedrockTraining` or `Purpose=BedrockKnowledgeBase`, preventing access to unrelated Amazon S3 resources

OpenSearch VPC endpoint policy

Apply this policy to your OpenSearch VPC endpoint to restrict knowledge base vector storage access:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowOpenSearchInTrustedOrg",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPost",
        "es:ESHttpPut",
        "es:ESHttpDelete"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceOrgID": "o-example123456"
        }
      }
    }
  ]
}
```

Policy explanation:

- **AllowOpenSearchInTrustedOrg** – Restricts OpenSearch access through VPC endpoints to clusters within your organization, preventing knowledge base queries from reaching external vector databases

Important – Replace `o-example123456` with your actual AWS organization ID. Find your organization ID using: `aws organizations describe-organization --query 'Organization.Id'`

Monitoring violations

Control objective

Resource perimeter *– My identities can access only trusted resources*

Continuous monitoring detects unauthorized access attempts and policy violations. Use CloudWatch alarms to alert on access denials that indicate attempts to bypass organizational boundaries or access restricted resources.

Monitor unauthorized API calls:

```
{
  "CloudWatchAlarms": [
    {
      "AlarmName": "UnauthorizedS3Access",
      "MetricName": "UnauthorizedAPICallsCount",
      "Threshold": 1,
      "ComparisonOperator": "GreaterThanOrEqualToThreshold",
      "AlarmActions": [
        "arn:aws:sns:us-east-1:123456789012:security-alerts"
      ]
    }
  ]
}
```

Policy explanation:

- **UnauthorizedS3Access alarm** – Triggers immediate alerts when unauthorized API calls are detected, enabling rapid response to potential resource perimeter violations or data exfiltration attempts

Key events to monitor:

- *Access denied* errors with `PrincipalOrgID` or `ResourceOrgID` conditions (organizational boundary violations)
- AWS KMS decrypt failures (encryption perimeter violations)
- Amazon S3 `PutObject/GetObject` denials (data exfiltration attempts)
- Resource policy changes (potential weakening of perimeter controls)

Best practices and recommendations

1. **Defense in depth**: Layer multiple controls (AWS KMS key policies, IAM, resource policies, encryption, tagging).
2. **Encryption context validation**: Use encryption context in AWS KMS key policies to ensure keys are only used for intended purposes.
3. **Regular audits**: Conduct monthly reviews of resource access patterns and policy effectiveness.

4. **Automated compliance**: Use AWS Config Rules to continuously monitor resource configurations.
5. **Incident response**: Establish procedures for responding to resource perimeter violations.
6. **Documentation**: Maintain detailed documentation of all resource classifications and access requirements.

Network perimeter

The network perimeter for Amazon Bedrock addresses the unique challenges of AI workloads where prompts and responses contain sensitive data that must be protected in transit. Unlike traditional API calls with predictable payloads, AI interactions involve variable-length content that might inadvertently contain confidential information.

This diagram is a high level illustration of a network perimeter.

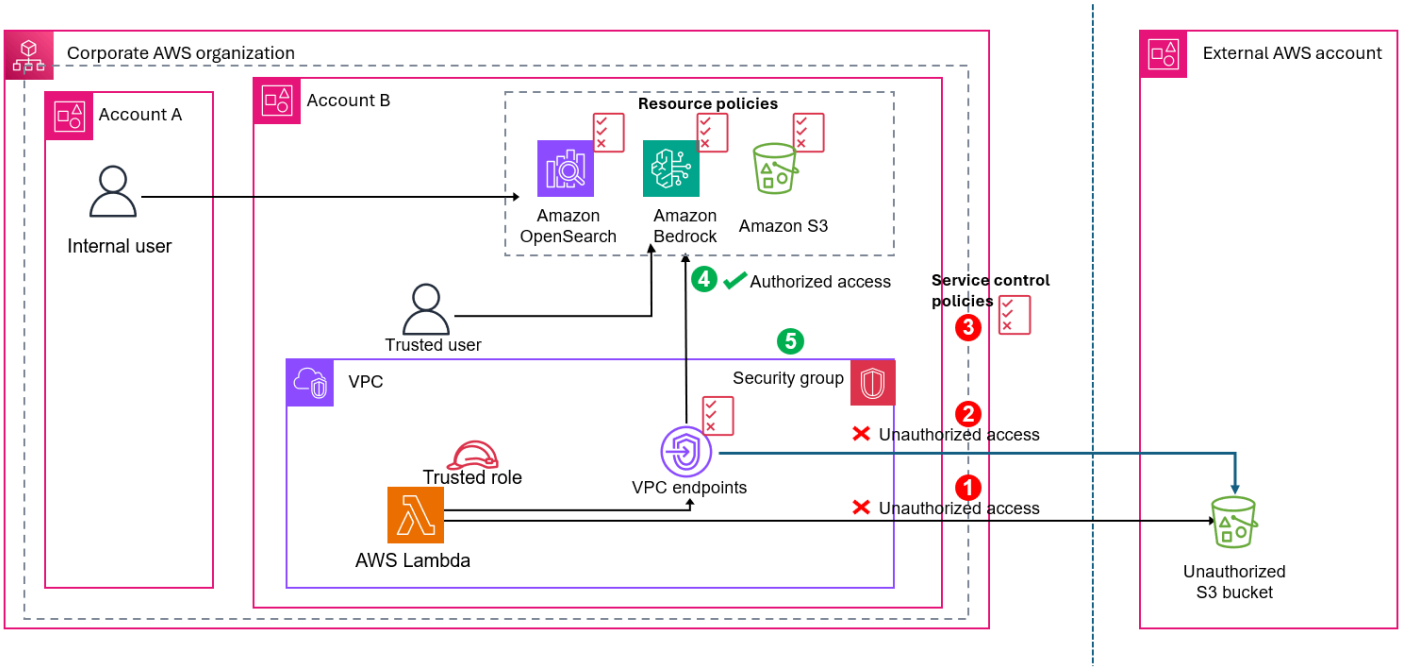


Figure 3.3 High level illustration of a network perimeter

1. SCP denies Amazon Bedrock, Amazon S3, OpenSearch and other operations not originating from VPC endpoint

2. Using Amazon VPC endpoints denies access to external untrusted resources

3. SCP denies Amazon Bedrock operations outside approved regions

4. Resource policies allow access only from known Amazon VPC endpoints

5. AWS security groups isolate AI workloads in dedicated subnets with HTTPS-only access to Amazon VPC endpoints

This section covers the essential network security controls for Amazon Bedrock:

- VPC endpoint enforcement - Implementing SCP controls to enforce private connectivity
- Network segmentation for AI workloads - Isolating AI processing from other network traffic
- Regional boundary enforcement - Implementing data residency and geographic restrictions
- Monitoring network access patterns - Detecting anomalous network behavior in AI workloads
- Network access control lists - Subnet-level network filtering for AI workloads
- Knowledge base network controls - Network access controls for knowledge base resources
- VPC endpoint network controls - Network-based Amazon VPC endpoint access restrictions
- Best practices - Network perimeter implementation recommendations and guidelines

Each subsection provides practical implementation guidance for securing AI workloads at the network level.

Note – These network security configurations provide baseline patterns but may not suit all infrastructure architectures or compliance requirements. Adapt the examples to your specific network topology, traffic patterns, and data residency obligations. Validate that network controls don't disrupt legitimate AI operations and consult Amazon VPC and networking documentation for the latest features and best practices.

VPC endpoint configuration

Control objective

Network perimeter *– My identities can access resources only from expected networks*

Enforce that all Amazon Bedrock operations must originate from Amazon VPC endpoints by applying this service control policy (SCP) at the organization or OU level:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "DenyBedrockPublicAccess",
      "Effect": "Deny",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:SourceVpce": "true"
        }
      }
    }
  ]
}
```

Policy explanation:

DenyBedrockPublicAccess – Blocks Amazon Bedrock API calls that don't originate from an Amazon VPC endpoint, ensuring all AI workload traffic flows through private network paths.

Note – SCPs do not apply to the management account. Test enforcement using member accounts.

Network segmentation for AI workloads

Control objective

***Network perimeter** – My identities can access resources only from expected networks*

Implement network segmentation that isolates AI processing from other workloads. Create dedicated subnets for Amazon Bedrock-related compute resources and configure AWS security groups that only allow necessary traffic flows.

Dedicated AI subnet configuration:

This Amazon VPC configuration defines isolated subnets for AI workloads. Applied during Amazon VPC setup or subnet creation.

```
{
  "VPCConfiguration": {
    "VpcId": "vpc-12345678",
    "Subnets": [
      {
```

```
        "SubnetId": "subnet-ai-workloads-1a",
        "AvailabilityZone": "us-east-1a",
        "CidrBlock": "10.0.10.0/24",
        "Tags": [
          {
            "Key": "Purpose",
            "Value": "BedrockAIWorkloads"
          }
        ]
      },
      {
        "SubnetId": "subnet-ai-workloads-1b",
        "AvailabilityZone": "us-east-1b",
        "CidrBlock": "10.0.11.0/24",
        "Tags": [
          {
            "Key": "Purpose",
            "Value": "BedrockAIWorkloads"
          }
        ]
      }
    ]
  }
}
```

Policy explanation:

- **VPC Configuration** – Creates dedicated subnets for AI workloads with specific CIDR blocks and purpose tags, enabling network isolation and targeted security controls for Amazon Bedrock-related resources

Security group rules for AI workloads

These security group rules control inbound/outbound traffic for AI workload resources. Applied to Amazon EC2 instances, AWS Lambda functions, or other compute resources accessing Amazon Bedrock.

Note – Replace `sg-bedrock-vpc-endpoints` with the actual security group ID attached to your Amazon Bedrock VPC endpoints.

```
{
```

```
  "SecurityGroupRules": [
    {
      "GroupId": "sg-bedrock-ai-workloads",
      "IpPermissions": [],
      "IpPermissionsEgress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 443,
          "ToPort": 443,
          "UserIdGroupPairs": [
            {
              "GroupId": "sg-bedrock-vpc-endpoints",
              "Description": "HTTPS to Bedrock VPC endpoints"
            }
          ]
        }
      ]
    }
  ]
}
```

Policy explanation:

- **SecurityGroupRules** – Restricts AI workload resources to communicate only with Amazon Bedrock VPC endpoints over HTTPS, preventing unauthorized network access and ensuring all Amazon Bedrock traffic flows through private endpoints

VPC endpoint security group configuration

The VPC endpoints for Amazon Bedrock must allow inbound traffic from the AI workload security group. Add this rule to the security group attached to your Amazon BedrockVPC endpoints:

```
{
  "IpPermissions": [
    {
      "IpProtocol": "tcp",
      "FromPort": 443,
      "ToPort": 443,
      "UserIdGroupPairs": [
        {
          "GroupId": "sg-bedrock-ai-workloads",
          "Description": "HTTPS from AI workload security group"
```

```
        }
      ]
    }
  ]
}
```

Policy explanation:

- **VPC Endpoint Security Group** – Allows inbound HTTPS traffic only from the AI workload security group, creating a secure communication channel between AI resources and Amazon Bedrock VPC endpoints

Regional boundary enforcement

Control objective

Network perimeter *– My identities can access resources only from expected networks*

AI workloads often have data residency requirements that mandate processing within specific geographic regions. Implement controls that prevent cross-region data flows and ensure model invocations occur only in approved regions by applying this service control policy (SCP) at the organization or OU level to enforce regional boundaries for all Amazon Bedrock operations:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictToApprovedRegions",
      "Effect": "Deny",
      "Action": "bedrock:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": [
            "us-east-1",
            "us-west-2",
            "eu-west-1"
          ]
        }
      }
    },
```

```
    {
      "Sid": "DataResidencyCompliance",
      "Effect": "Deny",
      "Action": [
        "bedrock:InvokeModel",
        "bedrock:CreateCustomModel"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/DataResidency": "EU"
        },
        "StringNotEquals": {
          "aws:RequestedRegion": [
            "eu-west-1",
            "eu-central-1"
          ]
        }
      }
    }
  ]
}
```

Policy explanation:

- **RestrictToApprovedRegions** – Blocks all Amazon Bedrock operations outside approved regions (us-east-1, us-west-2, eu-west-1).
- **DataResidencyCompliance** – Ensures EU data residency compliance by restricting EU-tagged operations to EU regions only.

Cross-region replication restrictions (Amazon S3 bucket policy):

Prevent cross-region replication of training data by attaching this policy to Amazon S3 buckets containing Amazon Bedrock training data:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PreventCrossRegionReplication",
      "Effect": "Deny",
```

```
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete"
      ],
      "Resource": "arn:aws:s3:::bedrock-*/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-repl-context": "SAME_REGION"
        }
      }
    }
  ]
}
```

Policy explanation:

- **PreventCrossRegionReplication** – Blocks Amazon S3 cross-region replication for Amazon Bedrock data buckets to maintain data residency.

Network access control lists (NACLs)

Control objective

Network perimeter *– My identities can access resources only from expected networks*

Implement network ACLs as an additional layer of defense for AI workload subnets. NACLs provide subnet-level traffic filtering that complements AWS security group rules.

NACL configuration for AI workload subnet:

This NACL is applied to the AI workload subnet to restrict traffic at the subnet boundary. It allows only HTTPS traffic to Amazon VPC endpoints and blocks all external internet access.

```
{
  "NetworkAclEntries": [
    {
      "RuleNumber": 100,
      "Protocol": "6",
      "RuleAction": "allow",
      "Egress": true,
      "PortRange": {
```

```
        "From": 443,
        "To": 443
      },
      "CidrBlock": "10.0.0.0/16"
    },
    {
      "RuleNumber": 110,
      "Protocol": "6",
      "RuleAction": "allow",
      "Egress": true,
      "PortRange": {
        "From": 1024,
        "To": 65535
      },
      "CidrBlock": "10.0.0.0/16"
    }
  ]
}
```

Policy explanation:

- **Rule 100** – Allows outbound HTTPS (port 443) to VPC CIDR for Amazon Bedrock API calls through VPC endpoints.
- **Rule 110** – Allows outbound ephemeral ports (1024-65535) to VPC CIDR for return traffic from Amazon Bedrock responses.
- **Implicit deny** – All other traffic is denied by default NACL rule (32767).

Knowledge base network controls

Control objective

Network perimeter *– My resources can only be accessed from expected networks*

Knowledge base resources must enforce network-based access controls to ensure data access only occurs through approved network paths.

OpenSearch domain network enforcement

Apply the following resource-based policy to OpenSearch domains to enforce Amazon VPCendpoint access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/BedrockKnowledgeBaseRole"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:123456789012:domain/bedrock-knowledge-base/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceVpce": "vpce-1234567890abcdef0"
        }
      }
    }
  ]
}
```

Amazon S3 vector store network enforcement

Apply this resource-based policy to Amazon S3 buckets containing vector store data:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowVPCEndpointAccess",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/BedrockKnowledgeBaseRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bedrock-vector-store-bucket",
        "arn:aws:s3:::bedrock-vector-store-bucket/*"
      ],
      "Condition": {
```

```
        "StringEquals": {
          "aws:SourceVpce": "vpce-1234567890abcdef0"
        }
      }
    }
  ]
}
```

Policy explanation:

- **VPC endpoint enforcement** – Requires all knowledge base access through the specified Amazon VPC endpoint
- **Network boundary** – Blocks direct internet access to knowledge base resources
- **Private network access** – Ensures sensitive AI data flows only through trusted network paths

Important – Replace `vpce-1234567890abcdef0` with your actual Amazon VPC endpoint ID for Amazon S3 and OpenSearch services.

VPC endpoint network controls

Control objective

Network perimeter *– My resources can only be accessed from expected networks*

Enforce network-based access controls at Amazon VPC endpoints to ensure Amazon Bedrock resources are accessed only through specific network paths.

VPC endpoint network policy

Apply this policy to Amazon Bedrock VPC endpoints to enforce specific endpoint access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSpecificVPCEndpoints",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
```

```
        "bedrock:InvokeModel",
        "bedrock:InvokeModelWithResponseStream",
        "bedrock:CreateKnowledgeBase",
        "bedrock:GetKnowledgeBase"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:SourceVpce": [
            "vpce-1234567890abcdef0",
            "vpce-0987654321fedcba0"
          ]
        }
      }
    }
  ]
}
```

Policy explanation:

- **VPC endpoint restriction** – Allows access only through specified Amazon VPC endpoints.
- **Network boundary** – Blocks access from unauthorized network paths.
- **Multi-endpoint support** – Supports multiple Amazon VPCendpoints for high availability.

Important – Replace `vpce-1234567890abcdef0` with your actual AmazonAmazon VPC IDs. To find your Amazon VPC endpoint IDs, use: `aws ec2 describe-vpc-endpoints --filters "Name=service-name,Values=com.amazonaws.*.bedrock"`

Combined identity and network controls

For comprehensive protection, combine this network policy with identity controls:

1. **Identity control** – Use `aws:PrincipalOrgID` to restrict identities (Identity perimeter).
2. **Network control** – Use `aws:SourceVpce` to restrict network paths (Network perimeter).
3. **Defense in depth** – Both conditions must be satisfied for access.

Monitoring network access patterns

Control objective

Network perimeter *– My identities can access resources only from expected networks / My resources can only be accessed from expected networks*

AI workloads can generate unexpected network traffic patterns due to their dynamic nature. Implement comprehensive monitoring that detects anomalous network behavior, such as unusual data volumes, unexpected destination endpoints, or traffic patterns that suggest data exfiltration attempts.

VPC Flow Logs configuration:

Configure Amazon VPC Flow Logs for the Amazon VPC containing your AI workloads using this configuration:

```
{
  "FlowLogConfiguration": {
    "ResourceType": "VPC",
    "ResourceId": "vpc-12345678",
    "TrafficType": "ALL",
    "LogDestinationType": "cloud-watch-logs",
    "LogGroupName": "/aws/vpc/bedrock-flowlogs",
    "DeliverLogsPermissionArn": "arn:aws:iam::123456789012:role/flowlogsRole",
    "Tags": [
      {
        "Key": "Purpose",
        "Value": "BedrockNetworkMonitoring"
      }
    ]
  }
}
```

- **VPC Flow Logs** – Captures all network traffic metadata for the Amazon VPC containing AI workloads, enabling detection of anomalous traffic patterns and potential data exfiltration attempts

CloudWatch Insights queries for anomaly detection:

These queries analyze Amazon VPC Flow Logs to detect unusual network behavior. The first query identifies sources generating high traffic volumes to Amazon Bedrock endpoints, which could indicate data exfiltration or misconfigured applications. The second query detects connections from AI workload subnets to external (non-private) IP addresses, which violates the network perimeter.

```
-- Detect unusual traffic volumes to Bedrock endpoints
fields @timestamp, srcaddr, dstaddr, bytes
| filter dstaddr like /bedrock/
| stats sum(bytes) as total_bytes by srcaddr
| sort total_bytes desc
| limit 20

-- Identify connections to unexpected external endpoints
fields @timestamp, srcaddr, dstaddr, dstport
| filter srcaddr like /10\.0\.1[01]\./
| filter not (dstaddr like /10\./ or dstaddr like /172\.16\./ or dstaddr
 like /192\.168\./)
| stats count() as connection_count by dstaddr, dstport
| sort connection_count desc
```

Query explanation:

- **Traffic volume query** – Identifies sources generating high traffic to Amazon Bedrock endpoints, helping detect potential data exfiltration or misconfigured applications
- **External connection query** – Detects AI workload subnets connecting to non-private IP addresses, indicating potential network perimeter violations

Best practices and recommendations

1. **Zero trust networking** – Verify every network connection, even within your Amazon VPC
2. **Micro-segmentation** – Isolate AI workloads in dedicated network segments.
3. **Continuous monitoring** – Monitor all network traffic for anomalies and policy violations.
4. **Regular testing** – Conduct network penetration testing specific to AI workloads.
5. **Documentation** – Maintain network diagrams and security control documentation.

Conclusion

Implementing AWS data perimeter controls for Amazon Bedrock requires a comprehensive approach that addresses the unique security challenges of generative AI workloads.

The three-layered defense of identity, resource, and network perimeters provides robust protection against data exfiltration, unauthorized access, and compliance violations while maintaining the operational flexibility required for AI innovation.

The key to successful implementation lies in understanding that AI workloads differ fundamentally from traditional applications in their data flows, access patterns, and risk profiles. By implementing the controls outlined in this guide, organizations can apply minimum required security guardrails for an AI solution built on Amazon Bedrock.

Start with organizational boundary controls using SCPs, then implement resource-specific protections for your most sensitive AI data. Layer on network controls and comprehensive monitoring to create defense in depth.

Regular testing and validation ensure your controls remain effective as your AI workloads evolve and new threats emerge.

Remember that data perimeter implementation is an iterative process. Begin with the most critical controls for your highest-risk AI workloads, then expand coverage as you gain experience and confidence with the framework.

The investment in comprehensive data perimeter controls pays dividends in reduced security risk, improved compliance posture, and increased stakeholder confidence in your AI initiatives.

Beyond data perimeter – operational considerations

While this guide focuses on establishing security boundaries through identity, resource, and network perimeters, successful Amazon Bedrock deployments require additional operational practices that complement these foundational controls:

Model lifecycle management

- **Version control** – Pin to specific model versions in production environments to ensure consistent behavior and security posture.

- **Upgrade procedures** – Establish testing and approval workflows before adopting new model versions.
- **Deprecation planning** – Monitor AWS announcements for model deprecations and plan migrations accordingly.
- **Custom model governance** – Document training data lineage, implement approval gates for model deployment, and establish retirement procedures.

Logging and monitoring strategy

- **Prompt logging** – Enable model invocation logging with appropriate encryption and retention policies based on data sensitivity.
- **Performance monitoring** – Establish baseline metrics for model behavior and monitor for drift or unexpected changes.
- **Cost tracking** – Implement CloudWatch alarms for usage spikes and maintain budget alerts for Amazon Bedrock spending.
- **Audit trails** – Leverage CloudTrail for comprehensive audit logging of all Amazon Bedrock API calls..

Knowledge base governance

- **Data classification** – Implement classification schemes for all documents in knowledge bases.
- **Refresh schedules** – Establish procedures for updating knowledge base content and removing stale data.
- **Access patterns** – Monitor query patterns for unusual data access that might indicate security issues.
- **Data retention** – Document and enforce retention policies for knowledge base content.

Agent security practices

- **Action group review** – Require security review for all new agent action groups and Lambda integrations.
- **Input validation** – Implement robust validation in all agent action group Lambda functions..
- **Capability documentation** – Maintain clear documentation of each agent's capabilities and business justification.

- **Invocation monitoring** – Monitor agent invocations for unusual patterns or potential abuse.

Guardrails implementation

- **Content filtering** – Configure Amazon Bedrock guardrails based on use case sensitivity and compliance requirements.
- **Adversarial testing** – Test guardrails with adversarial prompts before production deployment.
- **Threshold tuning** – Monitor guardrail intervention rates and adjust thresholds based on operational experience.
- **Bypass procedures** – Document legitimate use cases that may require guardrail adjustments.

Incident response planning

- **AI-specific incidents** – Define response procedures for prompt injection attacks, data leakage, and model abuse.
- **Automated response** – Implement automated responses for repeated policy violations.
- **Runbook maintenance** – Maintain detailed runbooks for common AI security incidents.
- **Access revocation** – Document procedures for emergency model access revocation.

These operational practices, combined with the data perimeter controls outlined in this guide, provide a comprehensive security and operational framework for your Amazon Bedrock deployments.

For detailed guidance on implementing these operational practices, refer to the **Resources** section, which includes links to AWS documentation, best practices guides, and operational tools.

Resources

AWS documentation

- AWS Data Perimeter - Official AWS data perimeter strategy and guidance
- Amazon Bedrock User Guide - Complete Amazon Bedrock service documentation
- AWS IAM Best Practices - IAM security recommendations
- VPC Endpoints for AWS Services - Private connectivity configuration
- AWS Organizations SCPs - Service control policy reference
- AWS CloudTrail Logging - Audit logging and monitoring
- IAM Access Analyzer - Automated policy analysis

Operational best practices

- AWS Well-Architected Framework - Machine Learning Lens - ML operational best practices and design principles
- Amazon Bedrock Guardrails - Content filtering and safety controls for responsible AI
- AWS Cost Management - Cost optimization and monitoring tools for Amazon Bedrock workloads
- Amazon Bedrock Model Invocation Logging - Logging configuration for prompts and responses

AWS blogs and articles

- Establishing a Data Perimeter on AWS - Foundational concepts and implementation patterns
- Amazon Bedrock Security Best Practices - Amazon Bedrock-specific security guidance
- Building a Secure AI/ML Environment - ML security blog posts

Related APG guides

- AWS Security Reference Architecture - Comprehensive security architecture patterns
- Multi-Account Security Strategy - Organizational security patterns

AWS samples and tools

- AWS Data Perimeter Helper - Tools for analyzing and implementing data perimeter controls
- AWS Security Hub - Centralized security findings and compliance checks
- AWS Config Rules for Data Perimeter - Automated compliance monitoring

Community resources

- AWS Security Blog - Latest security announcements and guidance
- AWS re:Post Security Community - Community Q&A and discussions
- AWS Workshops - Hands-on security workshops

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Initial publication	—	April 29, 2026