StrongDM Deployment: Best Practices & Considerations

strongdm

Table of Contents

About StrongDM

StrongDM is an identity-based privileged access manager (PAM) that provides auditability of infrastructure operations while simultaneously eliminating the need for end users to have direct access to credentials.

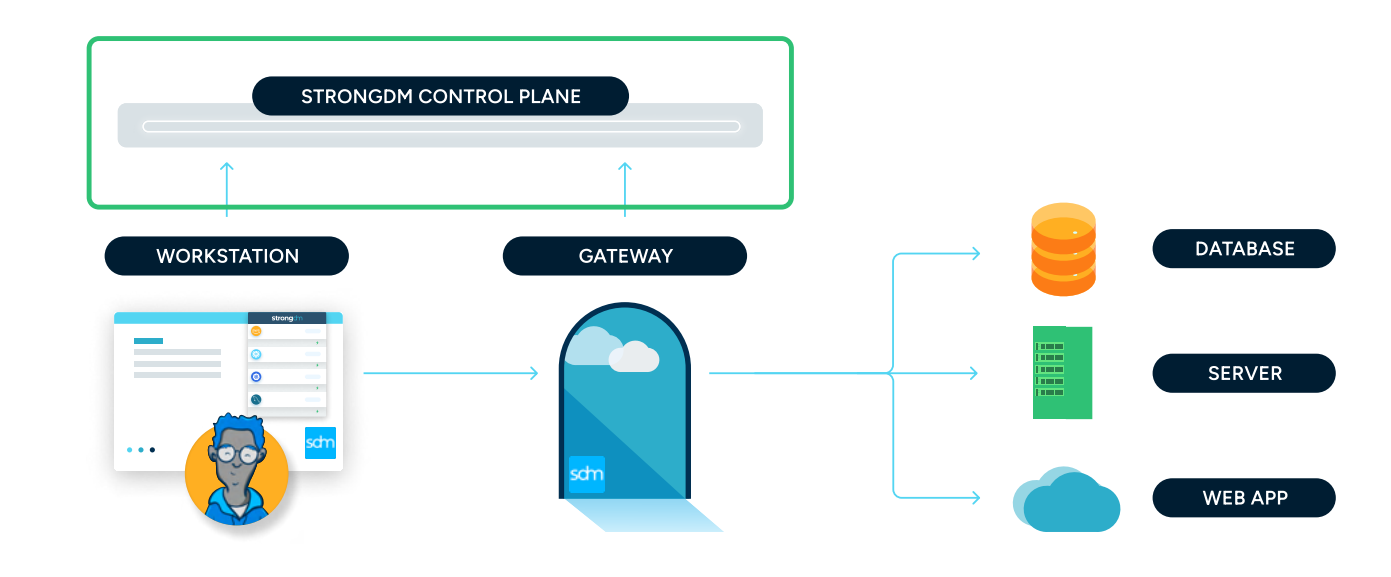
StrongDM provides a modern cloud-scale Dynamic Access Management (DAM) solution as companies focusing on the cloud shift away from legacy PAM solutions in addition to legacy VPN and bastion infrastructure access patterns.

Document Purpose

This document is intended to provide guidance around the architecture and deployment of StrongDM components in various environments. It is meant to inform StrongDM Administrators of common best practices, design and architecture considerations, and the associated configuration options.

Basic StrongDM Components and Prerequisites

StrongDM is a proxy that manages and audits access to databases, servers, clusters, and web applications. The StrongDM network consists of a local desktop client, gateway proxy node, and configuration layer.



Local Desktop Client

The local client tunnels requests from the user's workstation to the gateway, through a single TLS 1.2-secured TCP connection. StrongDM supports Mac, Windows, and Linux client systems.

To authenticate, users log in to the local client; that call can be optionally redirected to your identity provider or SSO.

The local client consists of graphical (Mac and Windows) and command-line interfaces (Mac, Windows, and Linux).

Proxy Node

Proxy Nodes — either Gateways or Relays — are the entry point to your network. They can be deployed with a DNS entry, sit privately on the corporate network, and/or behind a VPN.

In the case of a flat network, it is the node that talks to the target resources (i.e. systems, clusters, DBs, etc.). In more complex architectures with network segmentation, nodes may also talk to other nodes. For example, if internal subnets disallow ingress, relays create a reverse tunnel to form connections to gateways. All data routes through your network of clients, nodes, and resources.

Proxy Nodes decrypt credentials on behalf of end users, and deconstruct requests for the purposes of auditing.

Control Plane

The Control Plane — app.strongdm.com, the server-side/SAAS part of the architecture — manages the configuration of each specific StrongDM customer organization. Users are assigned to roles, and roles are collections of permissions across servers, databases, clusters, and web apps. Configuration is pushed down to the end user's local client, and updated in real-time.

Deployment Best Practices and Considerations

User Onboarding, Authentication, and Authorization

StrongDM offers a role-based access control (RBAC) system that brokers connections to resources either via static access rules, dynamic access rules determined via resource attributes and/or tags in the spirit of attribute-based access controls (ABAC), or temporary access grants to facilitate the model of Zero Standing Permissions.

Best Practices

- Use your existing SSO IDP solution for authentication into StrongDM.
- Use SCIM provisioning to onboard and offboard users.
- Use SCIM provisioning to authorize resource access for users by mapping StrongDM Roles with your IDP's User Groups.
- Configure break glass local administrators in the event of an IDP outage.
- Incorporate temporary access flows where possible in an effort towards Zero Trust.

SSO Authentication

StrongDM recommends using your existing single-sign on provider for authentication. This provides the benefit of one less set of static credentials, a login experience the user is already familiar with, and allows for additional controls such as MFA, device posture enforcement, or any other pre-authentication IDP hooks an organization has already invested in.

StrongDM integrates with a variety of IDPs listed here for authentication.

SCIM Provisioning

SCIM Provisioning allows users to be onboarded and authorized for StrongDM roles through automation with your existing IDP or entitlement provider. This means that IAM Administrators can add and remove users from groups the same way they do in their current workflows to streamline onboarding/offboarding and reduce the risk of onboarding and offboarding errors.

SCIM Provisioning bridges StrongDM Roles and an organization's entitlement groups.

Roles and Access Rules

StrongDM recommends managing user entitlements to SDM Roles via SCIM Provisioning, and to use Dynamic Access Rules to automate Role definitions, especially for ephemeral environments.

Roles
Roles are defined by access rules, either static or dynamic, and are assigned to users to authorize users for resource access.

Static Access Rules
Static Access Rules are the method by which you can assign access to specific resource(s) to a Role explicitly.

Dynamic Access Rules
Dynamic Access Rules provide the tool set to dynamically assign resource access to members of the Role. Each Dynamic Access Rule is made up of two properties:

- **Resource type.** You can choose a specific type of resource, such as a MySQL DBs or EKS Clusters, or you can choose All resource types.
- **Resource tags.** Tags are key-value pairs assigned to resources. An Access Rule may include up to 20 tags.

A Dynamic Access Rule will grant access to all resources that meet all of the criteria specified in its properties. For example, specifying one database type and two tags will grant access only to resources that are of that database type and have both of those tags.

Temporary Access Flows

Time-based temporary access grants can be assigned within StrongDM via the AdminUI, SDKs, or CLI. It is recommended to use temporary access grants wherever possible, to reduce the number of standing privileges.

For example, an organization that is in the early stages of moving towards a Zero Trust model may want to start their journey by requiring all environments classified as sensitive or production data to be accessed solely through temporary access grants whereas development and sandbox environments are granted access via standing roles.

Grant temporary access via AdminUI
Temporary access allows users to gain access to certain resources for a limited amount of time.

More information on granting temporary access via the AdminUI is here:
www.strongdm.com/docs/admin/users/#grant-temporary-acc

Workflows
Customer orgs with the Enterprise bundle also have access to Workflows, a feature which enables you to automate how access requests are submitted, reviewed, and approved (or denied). With access workflows, you can create subsets of resources and allow users with particular roles to request access to them. When requests are made, the pre-selected approvers for that workflow are notified and may then accept or deny the request.

Slack integration with Workflows is available, enabling requests and review/approval from the app.

More information on Workflows can be found here: www.strongdm.com/docs/admin/workflows/

Custom integrations with SDKs
Mature organizations incorporate their existing ITSM infrastructure and ticketing flows with StrongDM SDKs to automate the request, approval, and assignment of temporary access grants. This provides access to resources only as needed to reduce risk exposure, in addition to requiring justification and approval for more robust and granular audit trails.

An example of integration with PagerDuty is here: www.strongdm.com/docs/admin/deployment/scenarios/pagerduty/

Proxy Node Architecture

Best Practices

- Deploy nodes in pairs for redundancy
- Offset maintenance windows for node pairs for continuous uptime
- Use metrics endpoint to monitor performance for scaling considerations
- Deploy gateways geographically close to users
- Deploy relays close to resources

Proxy Node Packaging

Platforms
StrongDM Gateways and Relays are available as a linux binary, and thus are compatible with many target deployment types. Sample deployment walkthroughs for specific target deployment types can be found below:

- AWS EC2
- AWS ECS Fargate
- Azure VM
- Docker
- Kubernetes
- Linux
- Hashicorp Nomad

Recommended Minimum Specifications for Nodes

- 2 CPUs
- 4GB Memory
- **Disk Space:** Minimal, or commensurate with the expected logging volume when recording session logs locally on Gateways and Relays

New Releases and Package Updates

The latest version of Gateways and Relays should always be used to ensure compatibility across StrongDM clients, datasources, and peering Gateways and Relays.

StrongDM includes an auto-update system that is enabled by default to ensure nodes are always current. As long as the StrongDM Gateway or Relay is run by a user with the appropriate permissions and network access, it will automatically update itself. It is recommended to stagger maintenance windows for Gateways and Relays to ensure high availability during updates.

In cases where using the StrongDM auto-update system is not compatible with deployment policies, it is possible to freeze versions so that new versions can be explicitly deployed.

Additional detailed information regarding Gateway and Relay automatic update logic and maintenance window considerations can be found here.

Network Configuration Requirements

StrongDM Gateways have been specifically hardened to be exposed for ingress for StrongDM clients to connect. StrongDM will only accept connections from a StrongDM client that has first authenticated and been authorized via the StrongDM Control Plane for access.

StrongDM Gateways, Relays, and Clients have specific network routability requirements for each that are detailed here.

High Availability

Gateways are inherently intelligent enough to peer with each other and, when necessary, fail-over between themselves. With this in mind, it is best practice to deploy them in pairs for redundancy.

Scaling Gateways and Relays can be done vertically by increasing a Gateway's specifications such as CPU and/or memory when appropriate. Gateways are primarily CPU-bound in performance, and as such, it is recommended to consider scaling up Gateways with consistent saturation of CPU greater than 70%.

Alternatively, Gateways and Relays can also be scaled horizontally by adding additional Gateways and Relays in parallel, for example, via an auto-scaling group that triggers on usage thresholds in existing nodes.

Autoscaling

StrongDM Nodes can be placed in auto-scaling groups that deploy new instances based on CPU or other metrics spikes during unanticipated high traffic scenarios. However, as StrongDM nodes have built in fail-over and routing capabilities, StrongDM Nodes should NOT be placed behind a load balancer. If a load balancer is required, such as in a cluster configuration, there must be a 1:1 relationship between port and container.

Reference scripts for auto-scaling deployments can be found here.

Performance Monitoring

Gateways and Relays provide various metrics on performance that can be monitored, and used for expansion and sizing considerations. More information around monitoring can be found here.

Latency Considerations

In a connection chain of SDM Client → Gateway → Relay → Resource, the SDM client is expected to be the most latent link in the chain. With this in mind, StrongDM recommends deploying Gateways geographically close to users.

Relays, being the last link in the connection chain to a resource, should be placed as close to resources as possible.

In cases where users are geographically distributed, it may be advantageous to also distribute multiple sets of Gateways.

For example, an organization has offices in San Francisco and New York with resources scattered across the same geographic regions. Gateways should be deployed in each office network, with relays located in the resources' networks to ensure optimal performance.

Routing

Mesh Routing

The default routing mechanism in a StrongDM network automatically calculates all available routes from Proxy Nodes to Resources. This peer-to-peer routing is a mesh network. This approach works well for small networks without significant complexity in the form of a large number of gateways or relays and resources. However, with increasing complexity in a network, there is a computational cost (and increased time) for scanning for and calculating routes.

Explicit Routing

Peering Groups implement explicit routing in a StrongDM network. This feature enables an organization's nodes (gateways and relays) and resources to be segmented into explicitly declared groups called peering groups. Administrators can specify which nodes and resources are attached to peering groups and which peering groups peer with other peering groups.

When peering groups are used, the resource connection process is faster than usual because the flow of user traffic from the client to nodes to target resources is predetermined by the network administrator. When a user's client authenticates to StrongDM and requests to connect to a particular resource, that request is routed based on a calculation to the peering group in which the resource and node(s) are attached. Only the node(s) in the peering group check the user's permission level, role(s), and access grants and initiate a connection to the target resource.

In contrast, when the network operates in legacy mode, the path between the client and the target is fluid and unpredictable. Each node attempts to insert itself into the path to serve a resource, even when it may not be physically possible.

For organizations that have a high number of nodes and resources, using peering groups can significantly improve how client traffic is routed to nodes and resources.

At this time, it is optional to use peering groups. If peering groups are not enforced, the network operates in legacy mode.

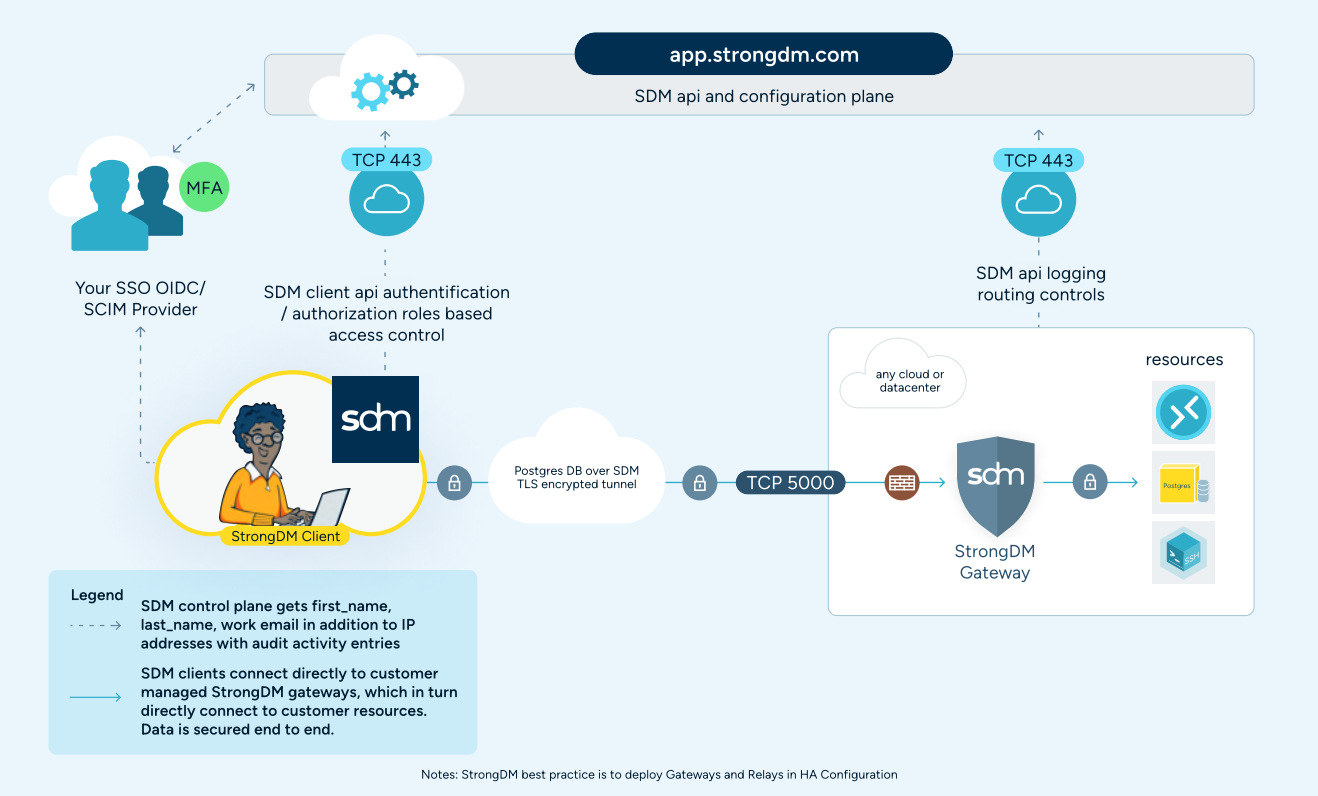
Proxy Node Reference Architectures

Introduction

StrongDM Proxy Node Deployment architecture for any given organization is dependent on their unique resource configurations and network topologies.

These reference architectures are intended to provide examples of various use cases, applicable architecture models, and help inform users on the array of solutions available.

Flat network – Gateways Only

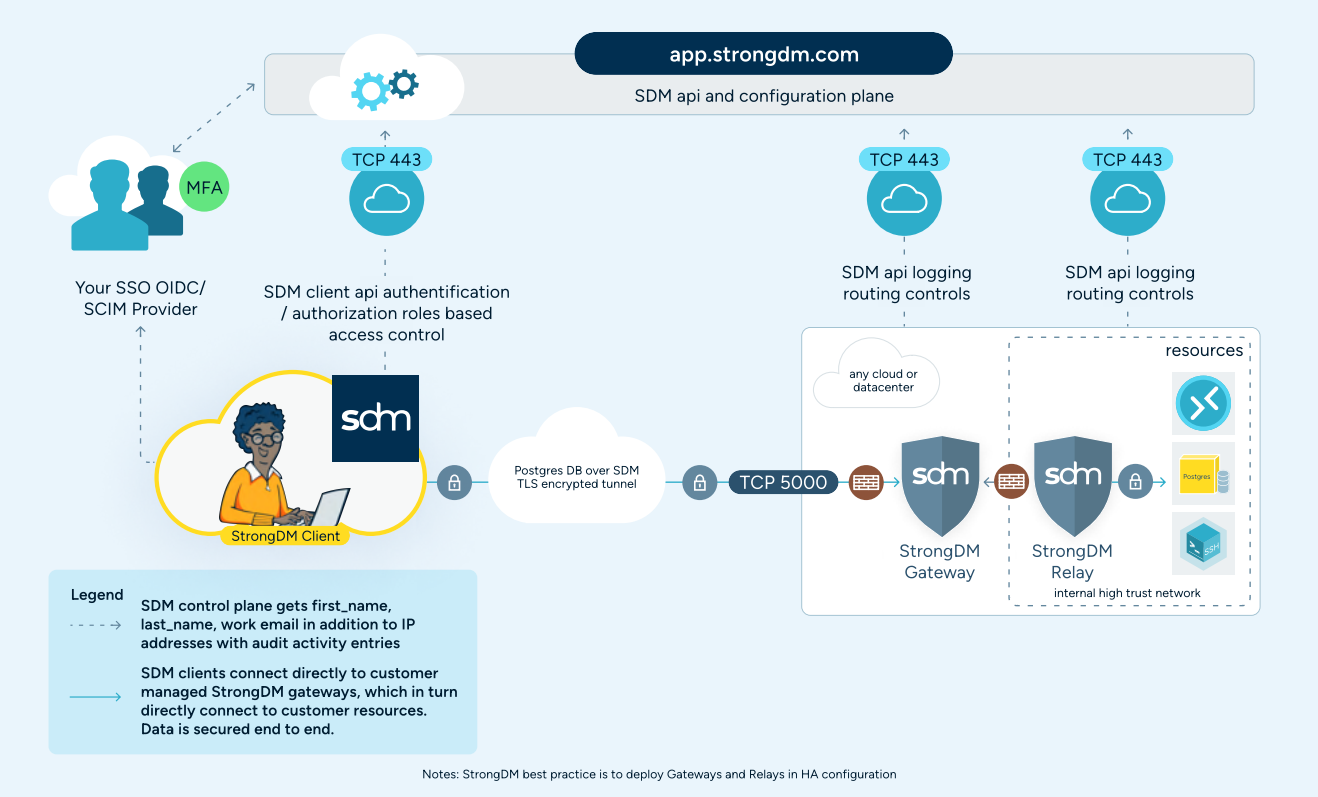


In simple, flat environments, deploying a pair of Gateways may be all that is required to start accessing resources. For a Gateway-only setup to function, resources must have ingress open to the Gateway, and the Gateway must have egress open to the end-resources.

Example use cases:

- AWS Console or AWS CLI proxying
- Servers or Databases that have lower sensitivity or serve public content, but still require private access for administration

Resources in isolated, no-ingress subnets - Gateways/Relays

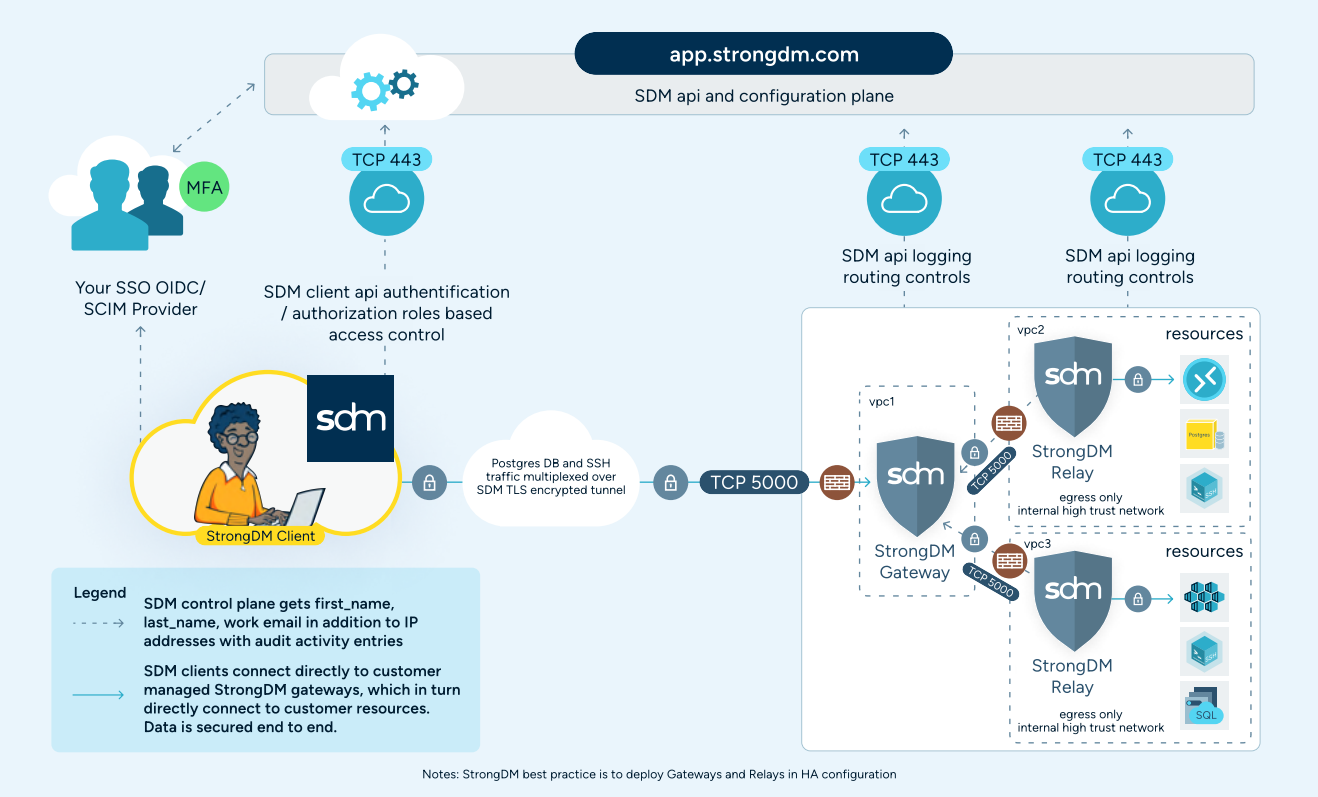


In many organizations, sensitive resources are isolated to subnets that do not permit any type of network ingress. To cover these cases, StrongDM Relays can be used in conjunction with Gateways. StrongDM Relays do not require ingress on Port 5000 as they form a reverse tunnel to Gateways via egress-only network paths. This provides the benefit of isolating resources via traditional network-based methods, while still permitting resource access to authorized consumers via StrongDM.

Example use cases:

- Sensitive Internal website resources
  - Data Science Query Tools (Snowsight, Redash)
  - CI/CD Admin Panels
  - Internally Hosted Repositories
  - Performance Metric Dashboards
- Sensitive Databases with strict network compliance requirements
- SSH / RDP Servers with strict network compliance requirements

Multi-VPC Hub & Spoke - Centralized Gateways serve multiple Relays

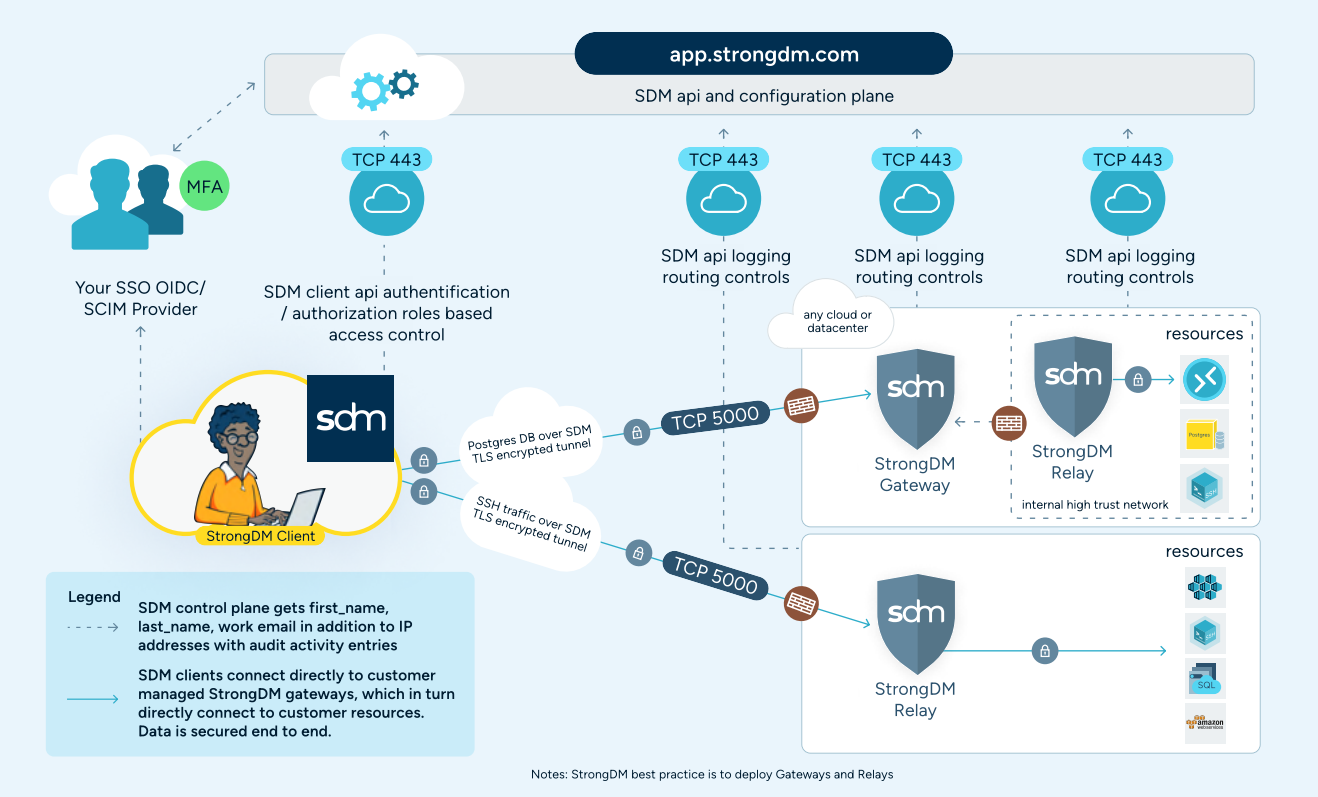


Expanding on the previous example, many organizations distribute resources into multiple, segregated VPCs that do not permit ingress. This pattern acts as a 'hub and spoke' model by using a centralized set of Gateways that serve multiple relays in isolated no-ingress subnets.

Example use cases:

- Segregation of different line of business' applications
- Isolating PHI/PII to separate VPCs that have greater compliance requirements than others

Multi-Environment

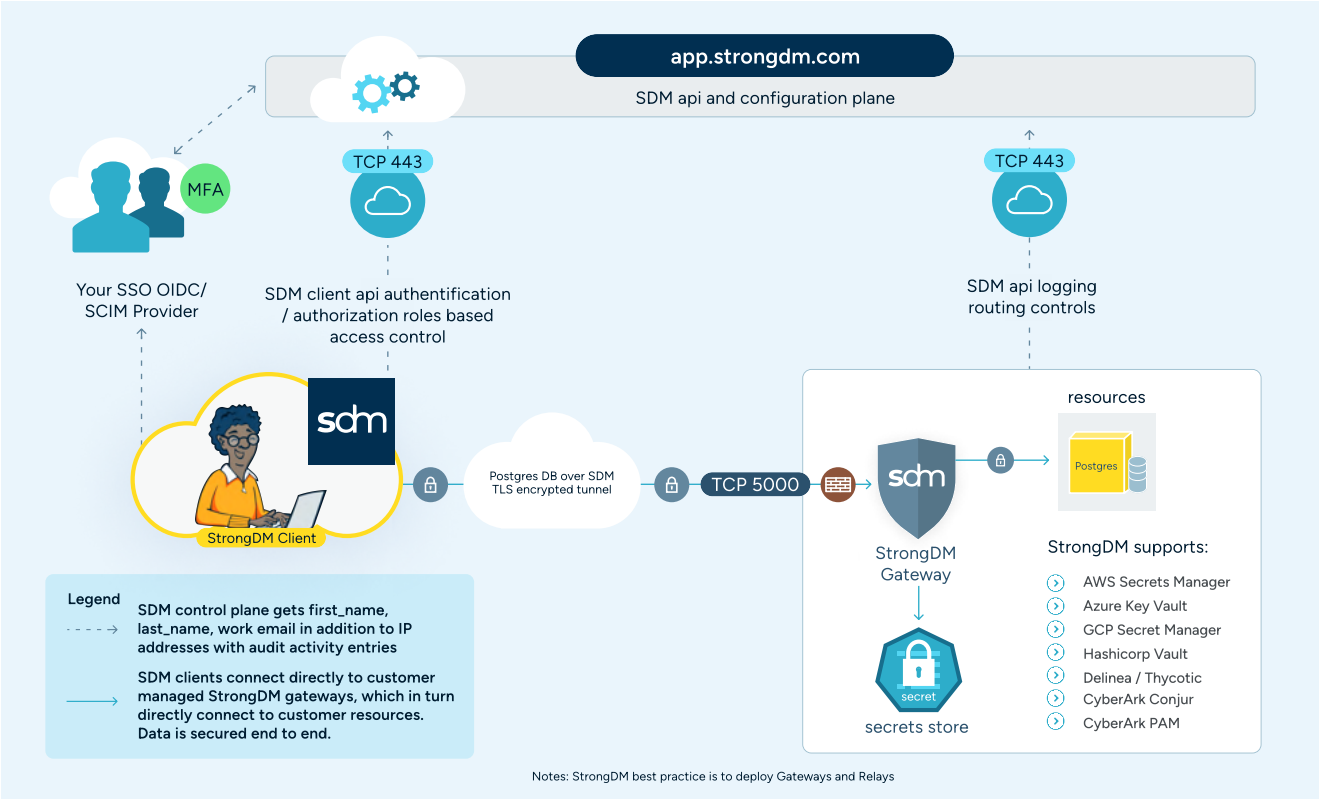


In scenarios where it is not desirable to use a hub and spoke model, it is also possible to utilize separate gateways and relays for their own environments. This model may be employed when an organization uses different cloud providers or on-premises data centers completely independent of each other, or they are located geographically far enough where latency considerations require closer Gateway entry points for clients.

Example use cases:

- On-premises vs Cloud-hosted resources
- Geographical separation that requires latency considerations and closer Gateways/Relays in proximity to users/resources
- Independent teams/application owners that require greater control of the management of their proxy nodes specific to their use cases
- Different sensitivity levels/controls between major environments

Secret Store-backed Resource Credentials



Considering different risk profiles for different resources, StrongDM Admins can choose where actual resource credentials are stored as part of a resource onboarding:

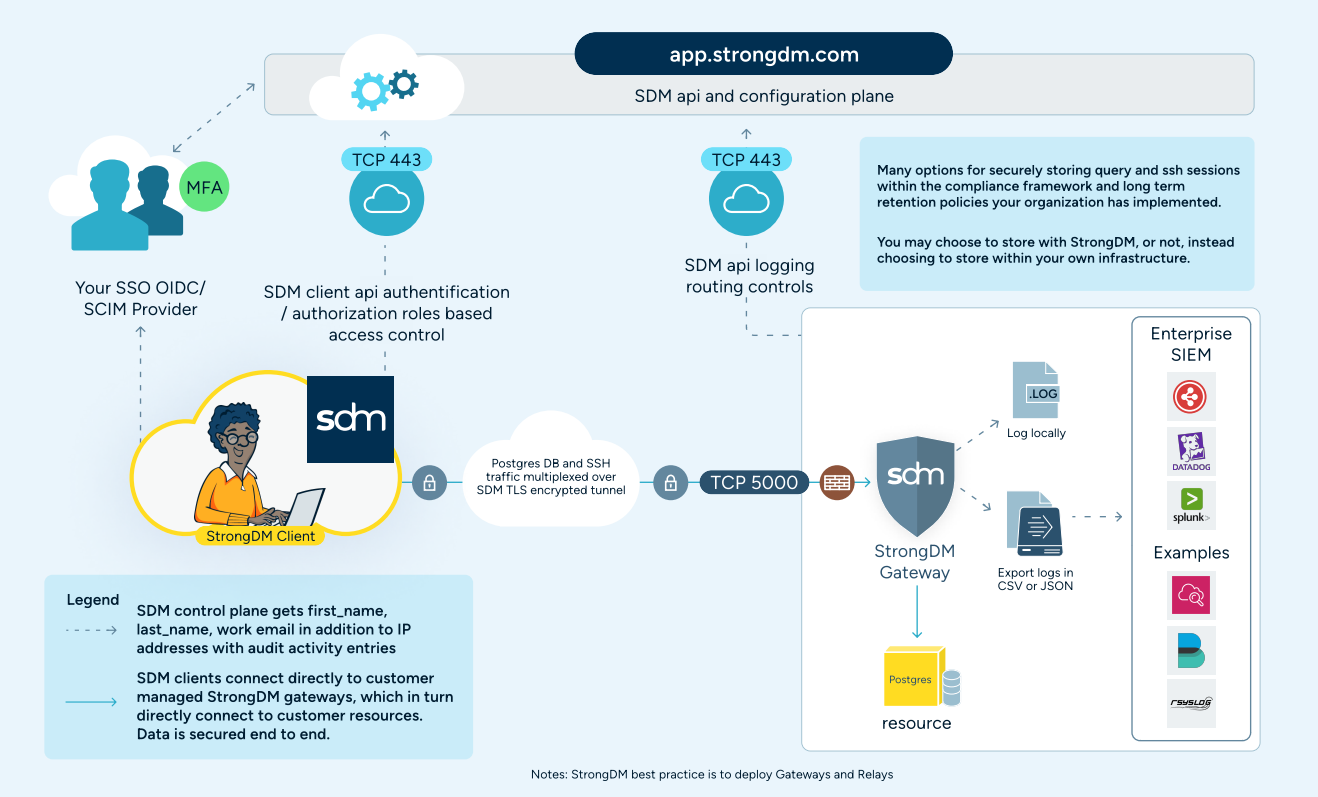
1. The StrongDM Control Plane can be used as a secret store for resource credentials
2. Alternatively, an external secret store can be used for scenarios where an organization would like to keep the credentials within their own network boundaries and independent/out-of-reach from StrongDM-managed infrastructure. Examples of secrets stores supported:

    - AWS Secrets Manager
    - Hashicorp Vault
    - Azure Keyvault
    - CyberArk Conjur
    - CyberArk PAM
    - Delinea

Example use cases:

- A healthcare organization stores non-sensitive resource credentials with StrongDM, but chooses to use AWS Secrets Manager for RDS instances that contain PHI/PII both to reduce attack surface and take advantage of AWS-native credential rotation
- A customer stores all resource secrets in Hashicorp Vault
- A multi-cloud organization uses their respective cloud provider secrets store to keep all credentials in their environment

Session Logging to SIEM and/or Archive Locations



Organizations have the option to store their logs in the following configurations:

1. Stored in StrongDM-managed infrastructure with a retention of 13 months
    - These logs can be encrypted by StrongDM, or alternatively encrypted via a user-held encryption key that StrongDM would not have access to
2. Log solely locally to Gateways/Relays where logs can be shipped to a customer-held archive location and/or SIEM
3. Both stored with StrongDM and logged locally to Gateways/Relays

Local logging format options are configurable, and detailed here.

Example use cases:

- When session logs are considered confidential, and must only be stored in a customer-held location, so an organization chooses to log locally only, and opts out of storage with StrongDM
- An organization takes advantage of StrongDM's included 13 months by storing with StrongDM, but also logs locally to send a subset of logs to their SIEM to trigger workflows based on activity
- For redundancy, an organization logs both to StrongDM and locally, but encrypts all logs to satisfy compliance requirements specific to their business vertical

strongdm

StrongDM provides a dynamic access platform that gives every business secure, dynamic access controls that people love to use. Trusted by the Fortune 500 to fast-growing businesses like SoFi, Chime, Yext, and Better, StrongDM gives businesses the control and visibility they need at the speed they want, with one platform that works for every environment. Connect with us on LinkedIn, Twitter, Facebook, YouTube or head to www.strongdm.com to learn more.