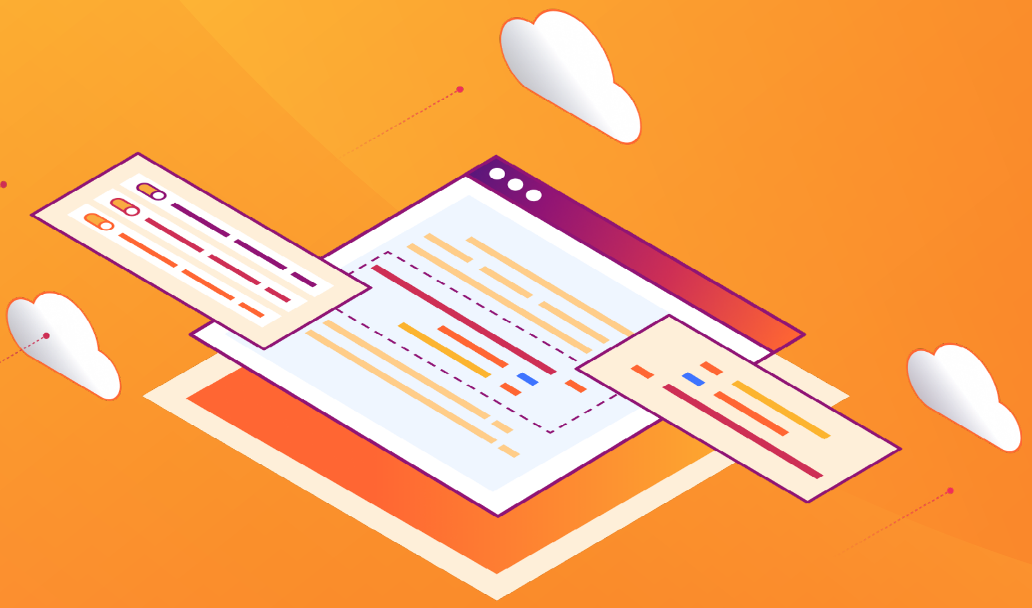
REFERENCE ARCHITECTURE

Evolving to a SASE architecture with Cloudflare



Content

Introduction

Cloudflare One is a Secure Access Service Edge (SASE) platform that protects enterprise applications, users, devices, and networks. By progressively adopting Cloudflare One, organizations can move away from their patchwork of hardware appliances and other point solutions and instead consolidate security and networking capabilities on one unified control plane. Such network and security transformation helps address key challenges modern businesses face, including:

- Securing access for any user to any resource with Zero Trust practices
- Defending against cyber threats, including multi-channel phishing and ransomware attacks
- Protecting data in order to comply with regulations and prevent leaks
- Simplifying connectivity across offices, data centers, and cloud environments

Cloudflare One is built on Cloudflare's connectivity cloud, a unified, intelligent platform of programmable cloud-native services that enable any-to-any connectivity between all networks (enterprise and Internet), cloud environments, applications, and users. It is one of the largest global networks, with data centers spanning more than 300 cities in over 100 countries and interconnection with over 12,500 other networks. It also has a greater presence in core Internet exchanges than many other large technology companies.

As a result, Cloudflare operates within ~50 ms of ~95% of the world's Internet-connected population. And since all Cloudflare services are designed to run across every network location, all traffic is connected, inspected, and filtered close to the source for the best performance and consistent user experience.

This document describes a reference architecture for organizations working towards a SASE architecture, and shows how Cloudflare One enables such security and networking transformation.

Who is this document for and what will you learn?

This reference architecture is designed for IT or security professionals with some responsibility over or familiarity with their organization's existing infrastructure. It is useful to have some experience with technologies important to securing hybrid work, including identity providers (IdPs), user directories, single sign on (SSO), endpoint security or management (EPP, XDR, UEM, MDM), firewalls, routers, and point solutions like packet or content inspection hardware, threat prevention, and data loss prevention technologies.

To build a stronger baseline understanding of Cloudflare, we recommend the following resources:

- What is Cloudflare? | Website (5 minute read) or video (2 minutes)
- Solution Brief: Cloudflare One (3 minute read)
- Whitepaper: Reference Architecture for Internet-Native Transformation (10 minute read)
- Blog: Zero Trust, SASE, and SSE: foundational concepts for your next-generation network (14 minute read)

Those who read this reference architecture will learn:

- How Cloudflare One protects an organization's employees, devices, applications, data, and networks
- How Cloudflare One fits into your existing infrastructure, and how to approach migration to a SASE architecture
- How to plan for deploying Cloudflare One

While this document examines Cloudflare One at a technical level, it does not offer fine detail about every product in the platform. Instead, it looks at how all the services in Cloudflare One enable networking and network security to be consolidated on one architecture. Visit developers.cloudflare.com/reference-architecture/ for further documents specific to a product area or use case.

Disintegration of the traditional network perimeter

Traditionally, most employees worked in an office and connected locally to the company network via Ethernet or WiFi. Most business systems (e.g. file servers, printers, applications) were located on and accessible only from this internal network. Once connected, users would typically have broad access to local resources. A security perimeter was created around the network to protect against outsider threats, most of which came from the public Internet. The majority of business workloads were hosted on-premises and only accessible inside the network, with very little or no company data or applications existing on the Internet.

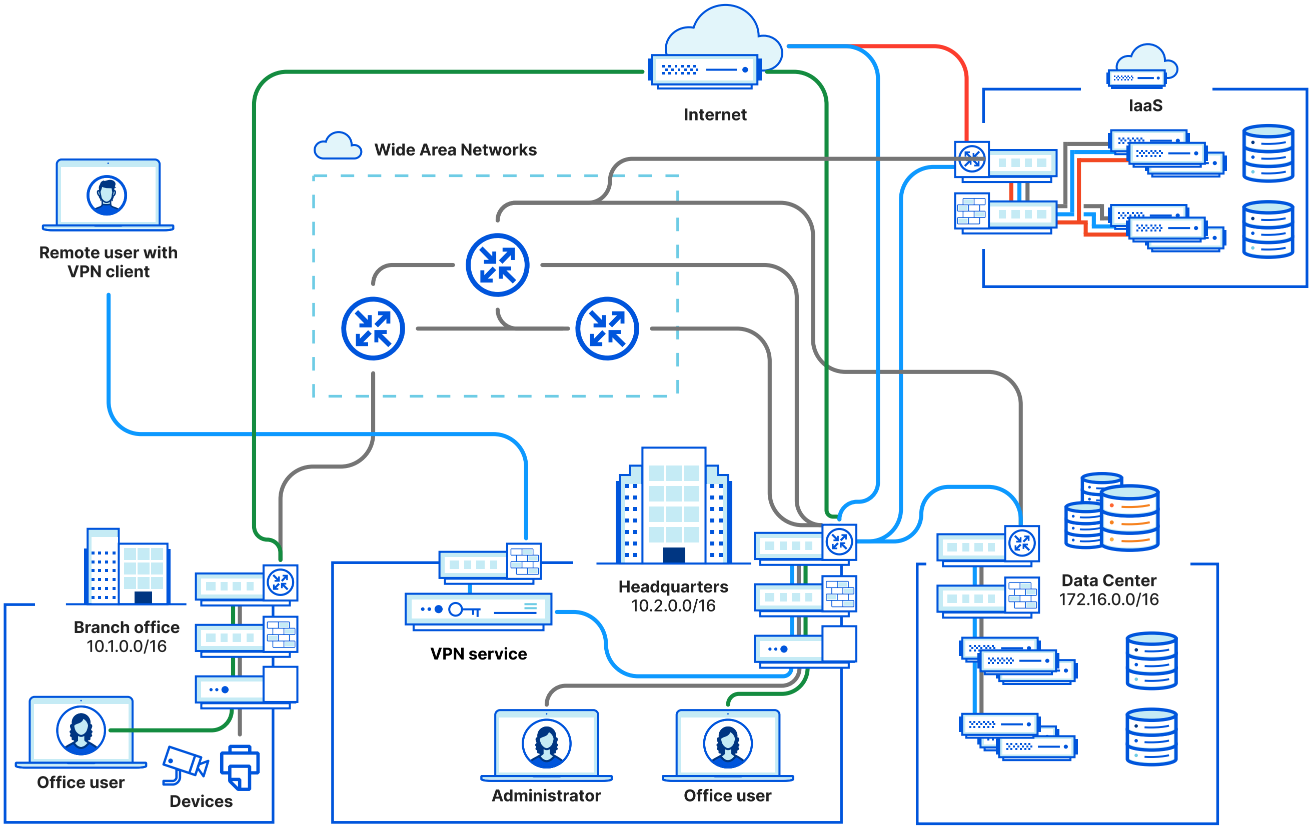
However, three important trends created problems for this "castle and moat" approach to IT security:

1. **Employees became more mobile.** Organizations increasingly embrace remote / hybrid work and support the use of personal (i.e. not company-owned) devices.

2. **Cloud migration accelerated.** Organizations are moving applications, data, and infrastructure from expensive on-premises data centers to public or private cloud environments in order to improve flexibility, scalability, and cost-effectiveness.

3. **Cyber threats evolved.** The above trends expand an organization's attack surface. For example, attack campaigns have become more sophisticated and persistent in exploiting multiple channels to infiltrate organizations, and cybercriminals face lower barriers to entry with the popularity of the 'cybercrime-as-a-service' black market.

Traditional perimeter-based security has struggled to adapt to these changes. In particular, extending the 'moat' outwards has introduced operational complexity for administrators, poor experiences for users, and inconsistency in how security controls are applied across users and applications.



The diagram above shows an example of this adapted perimeter-based approach, in which a mix of firewalls, WAN routers, and VPN concentrators are connected with dedicated WAN on-ramps consisting of MPLS circuits and/or leased lines. The diagram also demonstrates common problem areas. In an effort to centralize policy, organizations sometimes force all employee Internet traffic through their VPN infrastructure, which results in slow browsing and user complaints. Employees then seek workarounds — such as using non-approved devices — which increases their exposure to Internet-borne attacks when they work from home or on public WiFi. In addition, IT teams are unable to respond quickly to changing business needs due to the complexity of their network infrastructure.

Such challenges are driving many organizations to prioritize goals like:

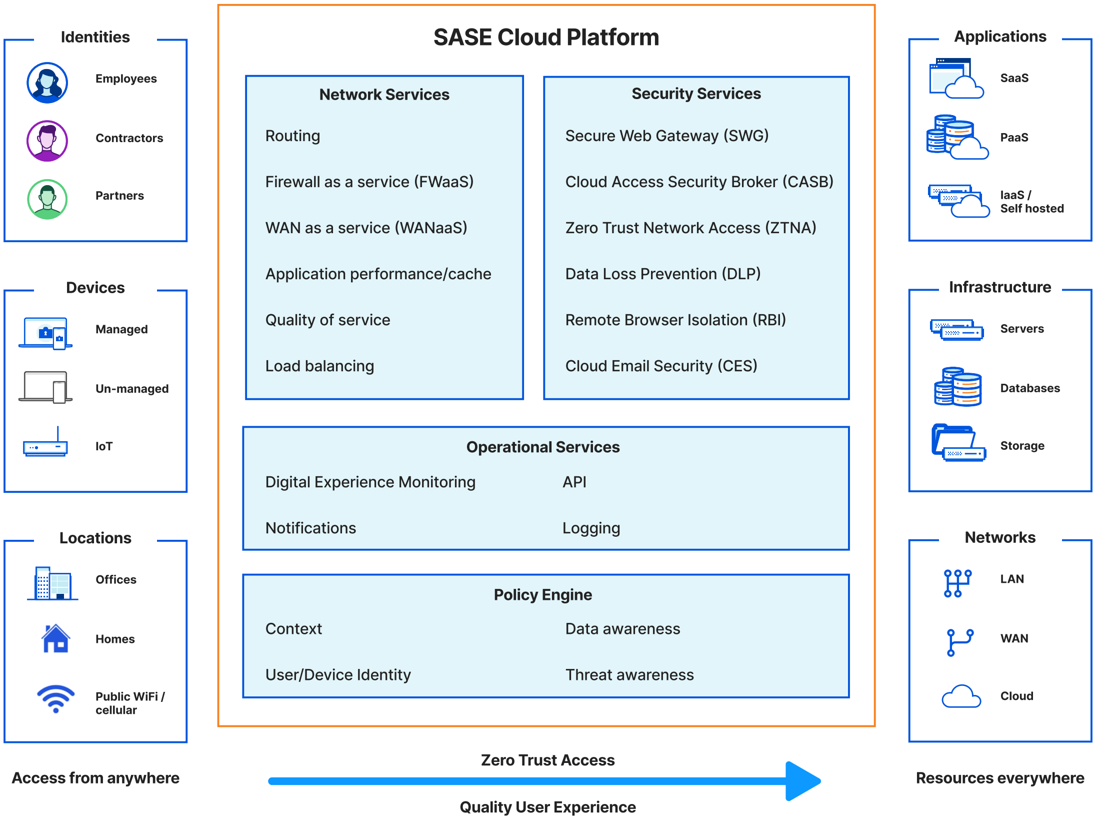
- Accelerating business agility by supporting remote / hybrid work with secure any-to-any access
- Improving productivity by simplifying policy management and by streamlining user experiences
- Reducing cyber risk by protecting users and data from phishing, ransomware, and other threats across all channels
- Consolidating visibility and controls across networking and security
- Reducing costs by replacing expensive appliances and infrastructure (e.g. VPNs, hardware firewalls, and MPLS connections)

Understanding a SASE architecture

In recent years, Secure Access Service Edge, or SASE, has emerged as an aspirational architecture to help achieve these goals. In a SASE architecture, network connectivity and security are unified on a single cloud platform and control plane for consistent visibility, control, and experiences from any user to any application.

SASE platforms consist of networking and security services, all underpinned by supporting operational services and a policy engine:

- Network services forward traffic from a variety of networks into a single global corporate network. These services provide capabilities like firewalling, routing, and load balancing.
- Security services apply to traffic flowing over the network, allowing for filtering of certain types of traffic and control over who can access what.
- Operational services provide platform-wide capabilities like logging, API access, and comprehensive Infrastructure-as-Code support through providers like Terraform.
- A policy engine integrates across all services, allowing admins to define policies which are then applied across all the connected services.



Identities
Employees
Contractors
Partners

Devices
Managed
Un-managed
IoT

Locations
Offices
Homes
Public WiFi / cellular

Access from anywhere

SASE Cloud Platform

Network Services
Routing
Firewall as a service (FWaaS)
WAN as a service (WANaaS)
Application performance/cache
Quality of service
Load balancing

Security Services
Secure Web Gateway (SWG)
Cloud Access Security Broker (CASB)
Zero Trust Network Access (ZTNA)
Data Loss Prevention (DLP)
Remote Browser Isolation (RBI)
Cloud Email Security (CES)

Operational Services
Digital Experience Monitoring
API
Notifications
Logging

Policy Engine
Context
Data awareness
User/Device Identity
Threat awareness

Zero Trust Access
Quality User Experience

Applications
SaaS
PaaS
IaaS / Self hosted

Infrastructure
Servers
Databases
Storage

Networks
LAN
WAN
Cloud

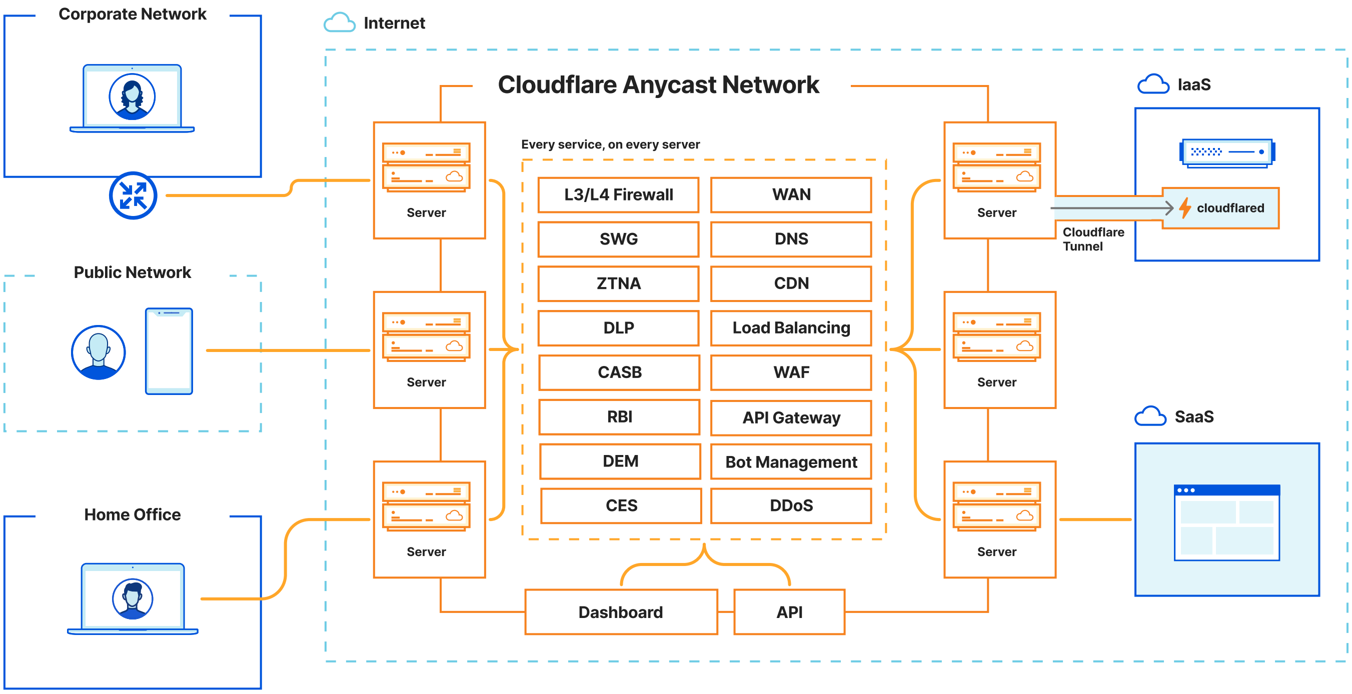
Resources everywhere

Cloudflare One: single-vendor, single-network SASE

Most organizations move towards a SASE architecture progressively rather than all at once, prioritizing key security and connectivity use cases and adopting services like Zero Trust Network Access (ZTNA) or Secure Web Gateway (SWG). Some organizations choose to use SASE services from multiple vendors. For most organizations, however, the aspiration is to consolidate security with a single vendor, in order to achieve simplified management, comprehensive visibility, and consistent experiences.

Cloudflare One is a single-vendor SASE platform where all services are designed to run across all locations. All traffic is inspected closest to its source, which delivers consistent speed and scale everywhere. And thanks to composable and flexible on-ramps, traffic can be easily routed from any source to reach any destination.

Cloudflare's connectivity cloud also offers many other services that improve application performance and security, such as API Gateway, Web Application Firewall, Content Delivery, or DDoS mitigation, all of which can complement an organization's SASE architecture. For example, our Content Delivery Network (CDN) features can be used to improve the performance of a self hosted company intranet. Cloudflare's full range of services are illustrated below.



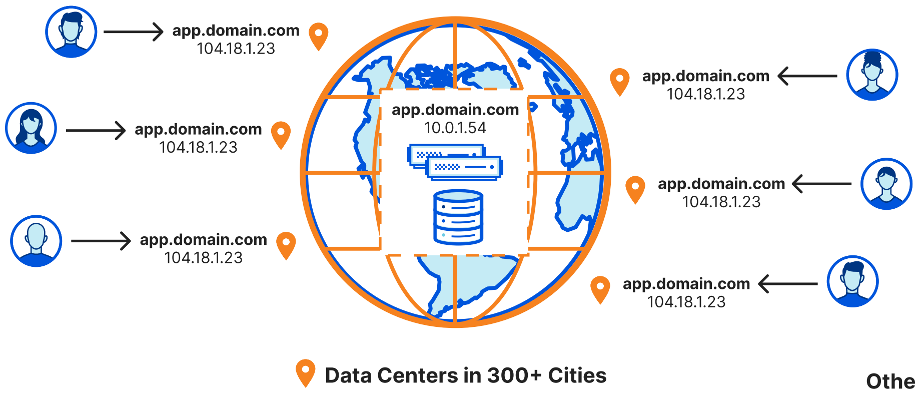
Cloudflare's Anycast Network

Cloudflare's SASE platform benefits from our use of Anycast technology. Anycast allows Cloudflare to announce the IP addresses of our services from every data center worldwide, so traffic is always routed to the Cloudflare data center closest to the source. This means traffic inspection, authentication, and policy enforcement take place close to the end user, leading to consistently high-quality experiences.

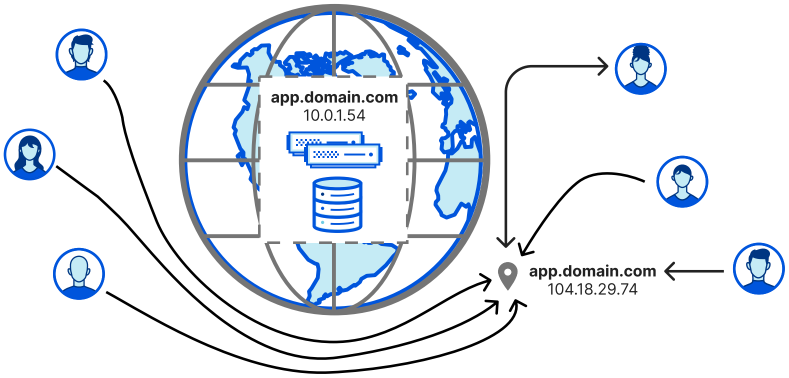
Using Anycast ensures the Cloudflare network is well balanced. If there is a sudden increase in traffic on the network, the load can easily be distributed across multiple data centers – which in turn, helps maintain consistent and reliable connectivity for users. Further, Cloudflare's large network capacity and AI/ML-optimized smart routing also help ensure that performance is constantly optimized.

By contrast, many other SASE providers use Unicast routing in which a single IP address is associated with a single server and/or data center. In many such architectures, a single IP address is then associated with a specific application, which means requests to access that application may have very different network routing experiences depending on how far that traffic needs to travel. For example, performance may be excellent for employees working in the office next to the application's servers, but poor for remote employees or those working overseas. Unicast also complicates scaling traffic loads — that single service location must ramp up resources when load increases, whereas Anycast networks can share traffic across many data centers and geographies.

Cloudflare's Anycast Network

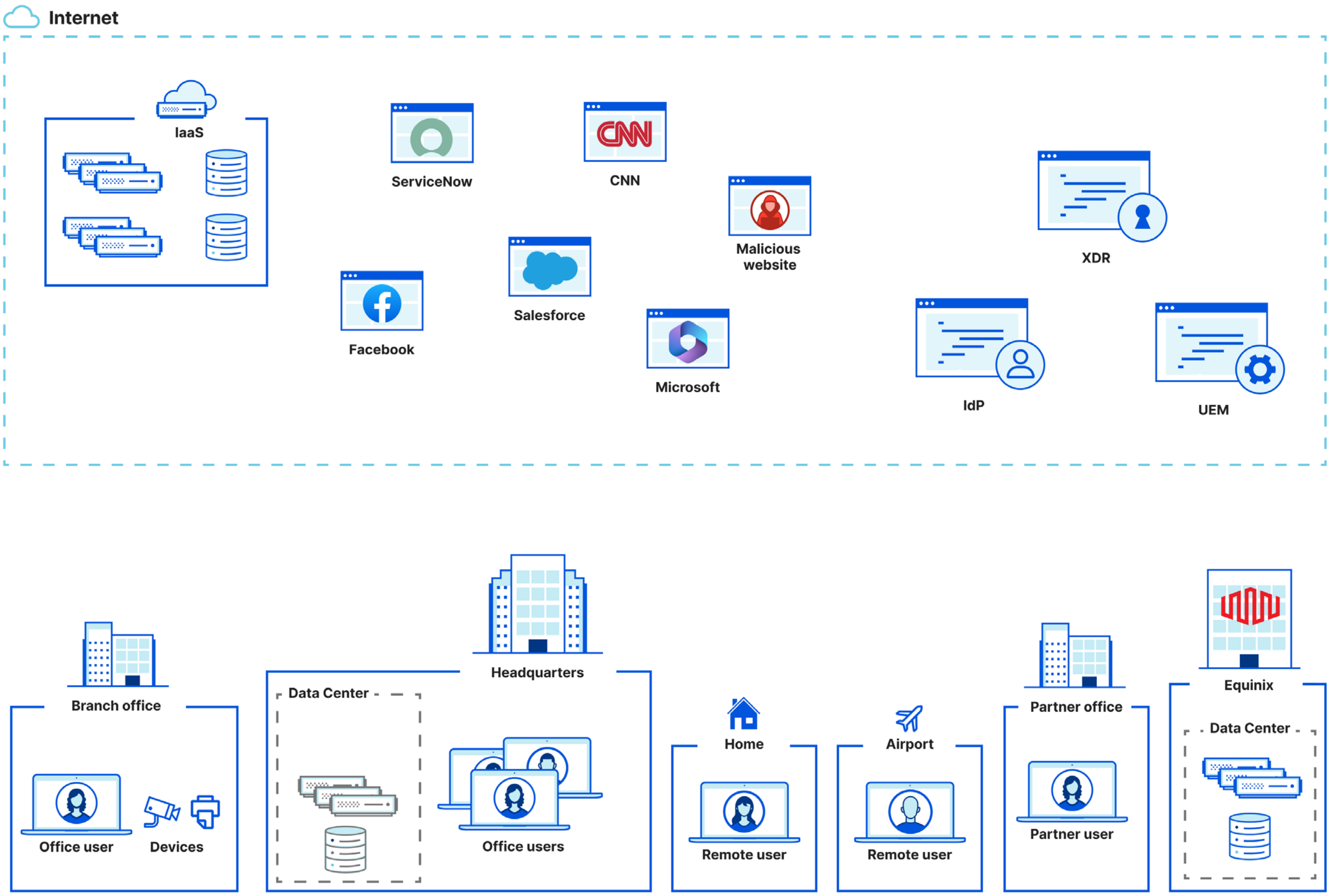


Other Unicast Network



Deploying a SASE architecture with Cloudflare

To understand how SASE fits into an organization's IT infrastructure, see the diagram below, which maps out all the common components of said infrastructure. Subsequent sections of this guide will add to the diagram, showing where each part of Cloudflare's SASE platform fits in.



In the diagram's top half there are a variety of Internet resources (e.g. Facebook), SaaS applications (e.g. ServiceNow), and applications running in an infrastructure-as-a-service (IaaS) platform (e.g. AWS). This example organization has already deployed cloud based identity providers (IdP), unified endpoint management (UEM) and endpoint protection platforms (EPP) as part of a Zero Trust initiative.

In the bottom half are a variety of users, devices, networks, and locations. Users work from a variety of locations: homes, headquarters and branch offices, airports, and others. The devices they use might be managed by the organization or may be personal devices. In addition to the cloud, applications run in a data center in the organization's headquarters and in a data center operators' colo facility (Equinix, in this example).

A SASE architecture will define, secure, and streamline how each user and device will connect to the various resources in the diagram. Over the following sections, this guide will show ways to integrate Cloudflare One into the above infrastructure:

- **Applications and services**: Placing access to private applications and services behind Cloudflare
- **Networks**: Connecting entire networks to Cloudflare
- **Forwarding device traffic**: Facilitating access to Cloudflare-protected resources from any device
- **Verifying users and devices:** Identifying which users access requests come from, and which devices those users have

Connecting Applications

This journey to a SASE architecture starts with an organization needing to provide remote access to non-Internet facing, internal-only web applications and services (e.g. SSH or RDP). Organizations typically deploy VPN appliances to connect users to the company network where the applications are hosted. However, many applications now live in cloud Infrastructure-as-a-Service platforms, where traditional VPN solutions are hard to configure. This often results in poor application and connectivity performance for users.

Tunnels to self-hosted applications

Zero Trust Network Access (ZTNA) is a SASE service that secures access to self-hosted applications and services. ZTNA functionality can be divided broadly into two categories:

1) establishing connectivity between Cloudflare's network and the environments where the applications are running, and 2) setting policies to define how users are able to access these applications. In this section, we first examine the former — how to connect apps to Cloudflare.

Connectivity to self-hosted applications is facilitated through tunnels that are created and maintained by a software connector, *cloudflared*. *Cloudflared* is a lightweight daemon installed in an organizations' infrastructure that creates a tunnel via an outbound connection to Cloudflare's global network. The connector can be installed in a variety of ways:

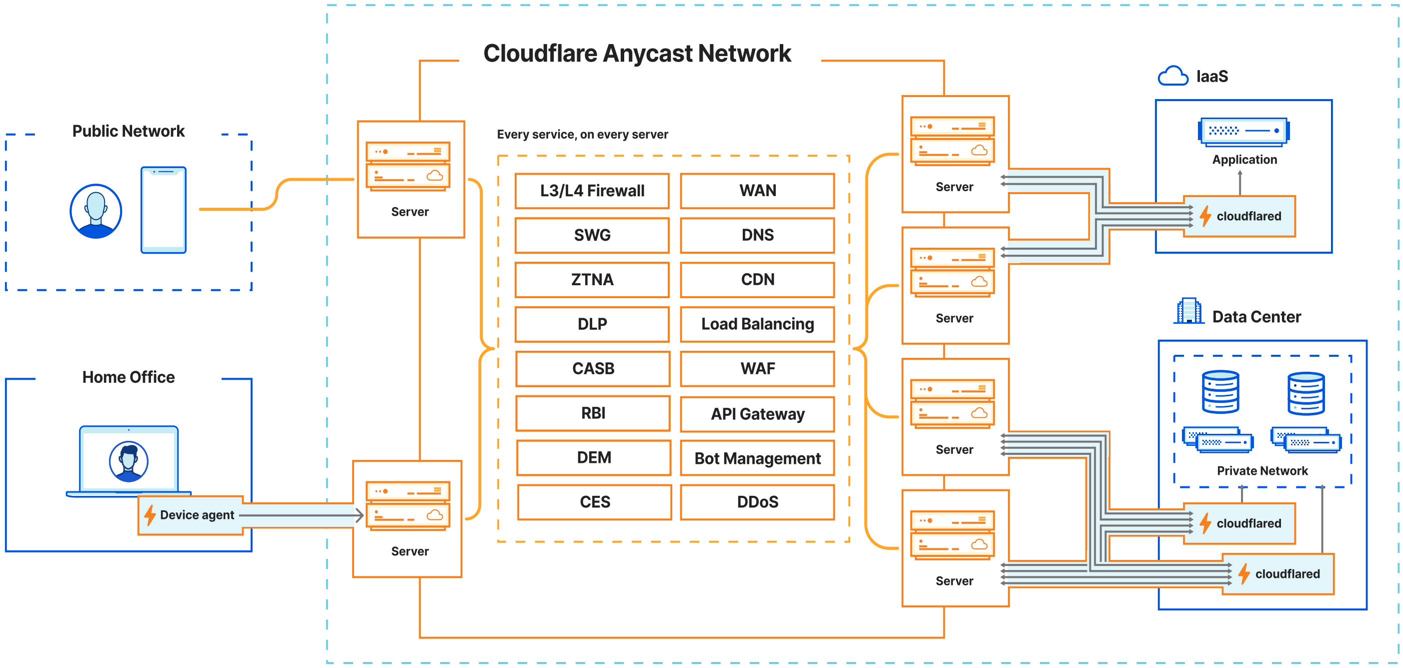
- In the OS installed on the bare metal server
- In the OS that is running in a virtualized environment
- In a container running in a Docker or Kubernetes environment

Cloudflared runs on Windows, Linux, or macOS operating systems and creates an encrypted tunnel using QUIC, a modern protocol that uses UDP (instead of TCP) for fast tunnel performance and modern encryption standards. Generally speaking, there are two approaches for how users can deploy cloudflared in their environment:

1. **On the same server and operating system where the application or service is running**. This is typically in high-risk or compliance deployments where organizations require independent tunnels per application. Cloudflared consumes a small amount of CPU and RAM, so impact to server performance is marginal.
2. **On a dedicated server(s) in the same network where the applications run**. This often takes the form of multiple containers in a Docker or Kubernetes environment.

Cloudflared manages multiple outbound connections back to Cloudflare and usually requires no changes to network firewalls. Those connections are spread across servers in more than one Cloudflare data center for reliability and failover. Traffic destined for a tunnel is forwarded to the connection that is geographically closest to the request, and if a *cloudflared* connection isn't responding, the tunnel will automatically failover to the next available.

For more control over the traffic routed through each tunnel connection, users can integrate with the Cloudflare load balancing service. To ensure reliable local connectivity, organizations should deploy more than one instance of cloudflared across their application infrastructure. For example, with ten front-end web servers running in a Kubernetes cluster, you might deploy three kubernetes services running *cloudflared* replicas.



Once tunnels have been established, there are two methods for how user traffic is forwarded to your application or service. Each method below is protected by policies managed by the ZTNA service that enforces authentication and access (which will be explored in further depth later in this document).

Public hostname

Each public hostname is specific to an address, protocol, and port associated with a private application, allowing for narrow access to a specific service when there might be multiple applications running on the same host.

For example, organizations can define a public hostname (mywebapp.domain.com) to provide access to a web server running on https://localhost:8080, while ensuring no access to local Kubernetes services.

Key capabilities:

- A hostname is created in a public DNS zone and all requests to that hostname are first routed to the Cloudflare network, inspected against configured security and access policies, before being routed through the tunnel to the secured private resource
- Multiple hostnames can be defined per tunnel, with each hostname mapping to a single application (service address and port)
- Support for HTTP/HTTPS protocols
- Access to resources only requires a browser
- When Cloudflare's device client is deployed on an user device, policies can leverage additional contextual signals (e.g. determining whether the device is managed or running the latest OS) in policy enforcement
- For access to SSH/VNC services, Cloudflare renders an SSH/VNC terminal using webassembly in the browser

Applications exposed this way receive all of the benefits of Cloudflare's leading DNS, CDN, and DDoS services as well as our web application firewall (WAF), API, and bot services, all without exposing application servers directly to the Internet.

Private network

In some cases, users may want to leverage ZTNA policies to provide access to many applications on an entire private network. This allows for greater flexibility over the ways clients connect and how services are exposed. It also enables communication to resources over protocols other than HTTP. In this scenario, users specify the subnet for the private network they wish to be accessible via Cloudflare.

Key capabilities:

- Cloudflared, combined with Cloudflare device agent, provides access to private networks, allowing for any arbitrary L4 TCP, UDP or ICMP connections
- One or many networks can be configured using CIDR notation (e.g. 172.21.0.16/28)
- Access to resources on the private network requires the Cloudflare device agent to be installed on clients, and at least one Cloudflare Tunnel server on the connecting network

For both methods, it is important to note that cloudflared only proxies inbound traffic to a private application or network. It does not become a gateway or 'on-ramp' back to Cloudflare for the network that it proxies inbound connections to. This means that if the web server starts its own connection to another Internet-based API, that connection will not be routed via Cloudflare Tunnel and will instead be routed via the host server's default route and gateway.

This is the desirable outcome in most network topologies, but there are some instances in which network services need to communicate directly with a remotely-connected user, or with services on other segmented networks.

If users require connections that originate from the server or network to be routed through Cloudflare, there are multiple on-ramps through which to achieve this, which will be explained further in the "Connecting Networks" section.

SaaS applications

SaaS applications are inherently always connected to and accessed via the public Internet. As a result, the aforementioned tunnel-and-app-connector approach does not apply. Instead, organizations with a SASE architecture inspect and enforce policies on Internet-bound SaaS traffic via a secure web gateway (SWG), which serves as a cloud-native forward proxy.

The SWG includes policies that examine outbound traffic requests and inbound content responses to determine if the user, device, or network location has access to resources on the Internet. Organizations can use these policies to control access to approved SaaS applications, as well as detect and block the use of unapproved applications (also known as shadow IT).

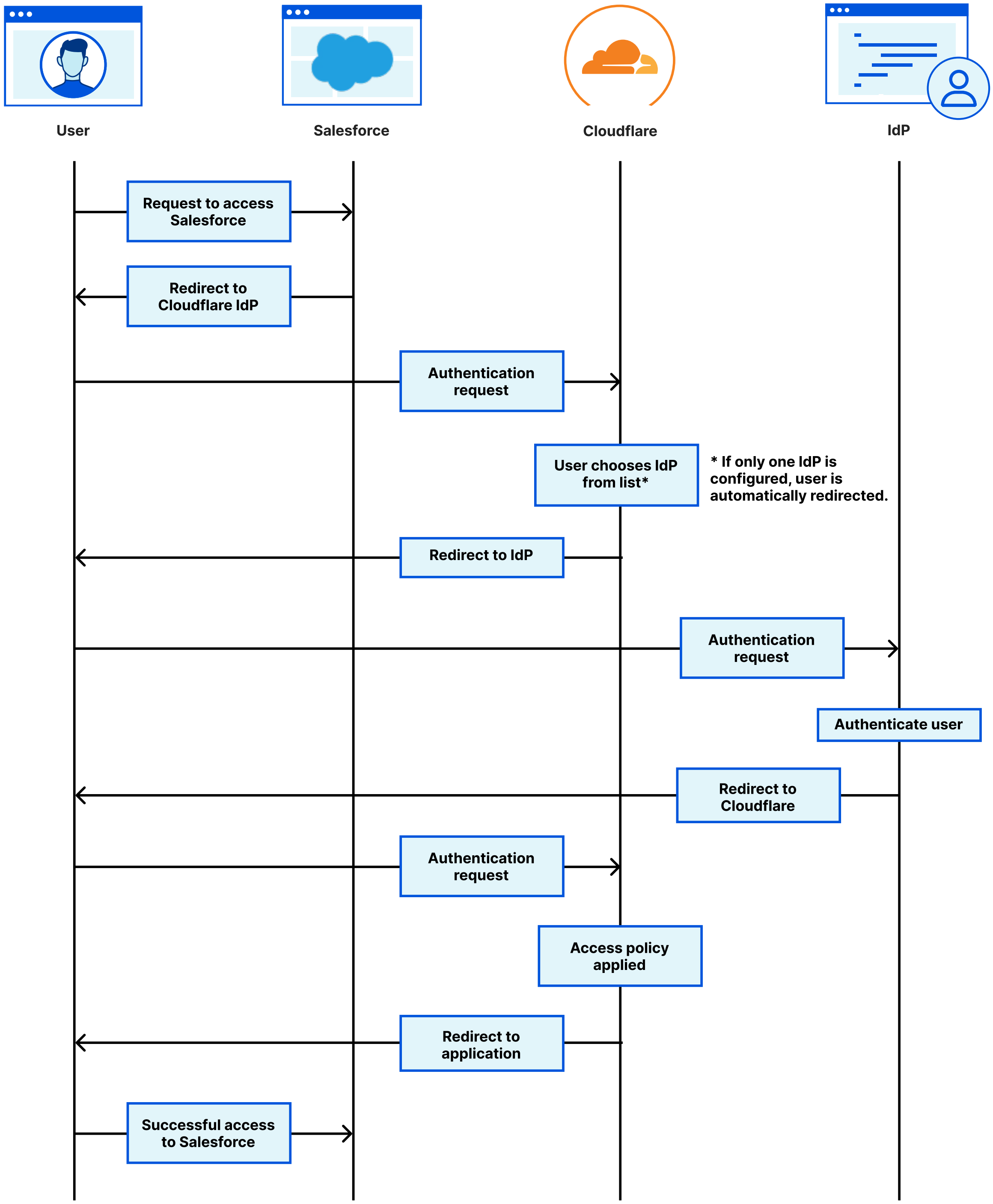
Some SaaS applications allow organizations to configure an IP address allowlist, which limits access to the application based on the source IP address of the request. With Cloudflare, organizations can obtain dedicated egress IP addresses, which can be used as the source address for all traffic leaving their network. When combined with an allowlist in a SaaS application, organizations can ensure that users are only able to access applications if they are first connected to Cloudflare. (More detail on this approach is outlined in a later section about connecting user devices.)

Another method to secure access to SaaS applications is to configure single sign-on (SSO) so that Cloudflare becomes an identity proxy — acting as the identity provider (IDP) — as part of the authentication and authorization process.

Key capabilities:

- Apply consistent access policies across both self-hosted and SaaS applications
- Layer device security posture into the authentication process (e.g. users can ensure that only managed devices, running the latest operating system and passing all endpoint security checks, are able to access SaaS applications)
- Ensure that certain network routes are used for access (e.g. users can require that devices are connected to Cloudflare using the device agent, which allows them to filter traffic to the SaaS application and prevent downloads of protected data)
- Centralize SSO applications to Cloudflare and create one SSO integration from Cloudflare to their IdP — making both infrastructure and access policies SSO-agnostic (e.g. users can allow access to critical applications only when MFA is used, no matter which IdP is used to authenticate)

When Cloudflare acts as the SSO service to an application, user authentication is still handled by an organization's existing identity provider, but is proxied via Cloudflare, where additional access restrictions can be applied. The diagram below is a high-level example of a typical request flow:



The last method of connecting SaaS applications to Cloudflare's SASE architecture is with an API-based cloud access security broker (CASB). The Cloudflare CASB integrates via API to popular SaaS suites — including Google Workspace, Microsoft 365, Salesforce, and more — and continuously scans these applications for misconfigurations, unauthorized user activity, and other security risks.

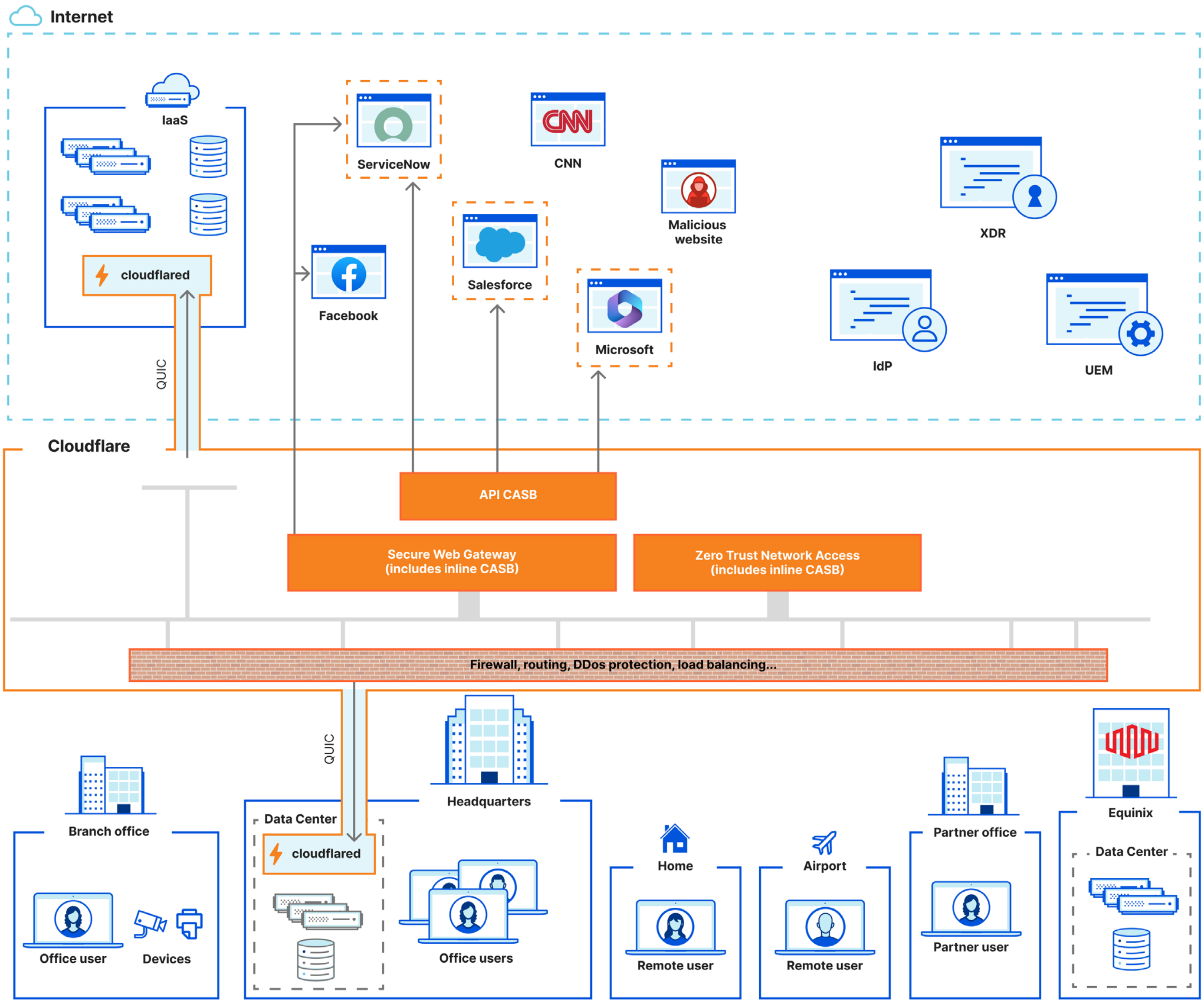
Native integration with the Cloudflare data loss prevention (DLP) service enables CASB to scan for sensitive or regulated data that may be stored in files with incorrect permissions — further risking leaks or unauthorized access. CASB reports findings that alert IT teams to items such as:

- Administrative accounts without adequate MFA
- Company-sensitive data in files stored with public access permissions
- Missing application configurations (e.g. domains missing SPF/DMARC records)

Checkpoint: Connecting applications to Cloudflare

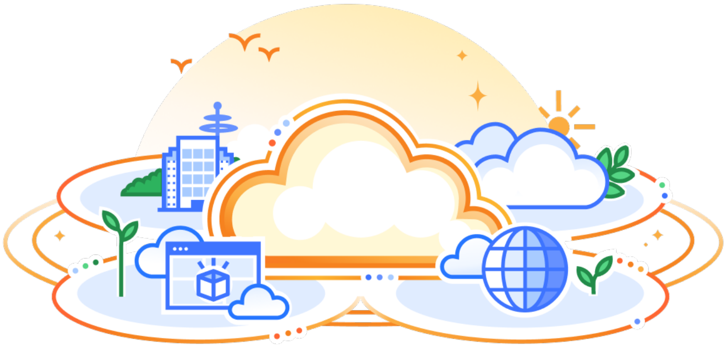
Now, this is what the architecture of a typical organization might look like once they have integrated with Cloudflare services. It is important to note that Cloudflare is designed to secure organizations' existing applications and services in the following ways:

- All self-hosted applications and services are only accessible through Cloudflare and controlled by policies defined by the Cloudflare ZTNA
- SaaS application traffic is filtered and secured via the Cloudflare SWG
- SaaS services are scanned via the Cloudflare CASB to check for configuration and permissions of data at rest



Connecting networks

Once an organization's applications and services have been integrated, it is time to connect Cloudflare to their existing networks. Regional offices, corporate headquarters, retail locations, data centers, and cloud-hosted infrastructure all need to forward traffic to the new corporate SASE network.



When all traffic flows through Cloudflare, SASE services perform the following actions:

- Granting application access
- Filtering general Internet-bound traffic (e.g. blocking access to sites that host malware)
- Isolating web sites to protect users from day-zero or unknown harmful Internet content
- Filtering traffic to identify data defined by DLP policies — then blocking the download/upload of that data to insecure devices or applications
- Providing visibility into the use of non-approved applications and allowing admins to either block or apply policies around their use

There are several approaches for connecting networks to Cloudflare, which can provide further flexibility in how an organization provides access to SASE-protected resources:

1. **Use software agents to create tunnels from host machines back to Cloudflare**. This is typically the method favored by users who own their own servers and applications.
2. **Set up IPsec or GRE tunnels from network routers and firewalls to connect them to the Cloudflare WAN service**. This is the approach that network administrators use when they want to forward traffic to and from entire networks.
3. **Connect a network directly to Cloudflare**. This method works best when an organization's network resides in a supported data center, usually one that is colocated with a Cloudflare data center.

These methods will be explained further in the next sections.

Using software agents

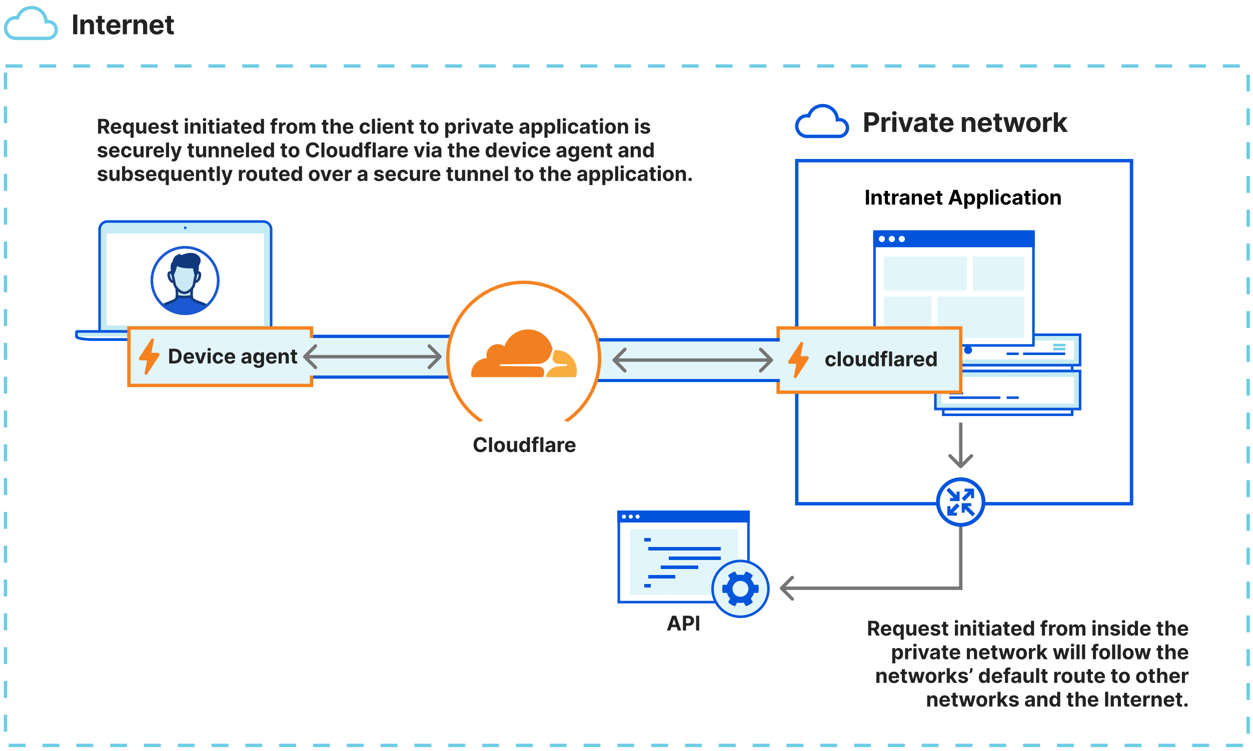
There are two software-based methods of connecting networks to Cloudflare, depending on the type of applications that currently exist on the network.

Client-to-server connectivity

As described in the previous section, *cloudflared* proxies requests to applications and services on private networks. It installs on servers in the private network and creates secure tunnels to Cloudflare over the Internet. These connections are balanced across multiple Cloudflare data centers for reliability and can be made via multiple connectors, which helps increase the capacity of the tunnels.

Using *cloudflared*, Cloudflare Tunnel supports client to server connections over the Tunnel. Any service or application running behind the Tunnel will use the default routing table when initiating outbound connectivity.

This model is appropriate for a majority of scenarios, in which external users need to access resources within a private network that does not require bidirectionally-initiated communication.


Internet
Request initiated from the client to private application is securely tunneled to Cloudflare via the device agent and subsequently routed over a secure tunnel to the application.
Private network
Intranet Application
Device agent
Cloudflare
cloudflared
API
Request initiated from inside the private network will follow the networks' default route to other networks and the Internet.

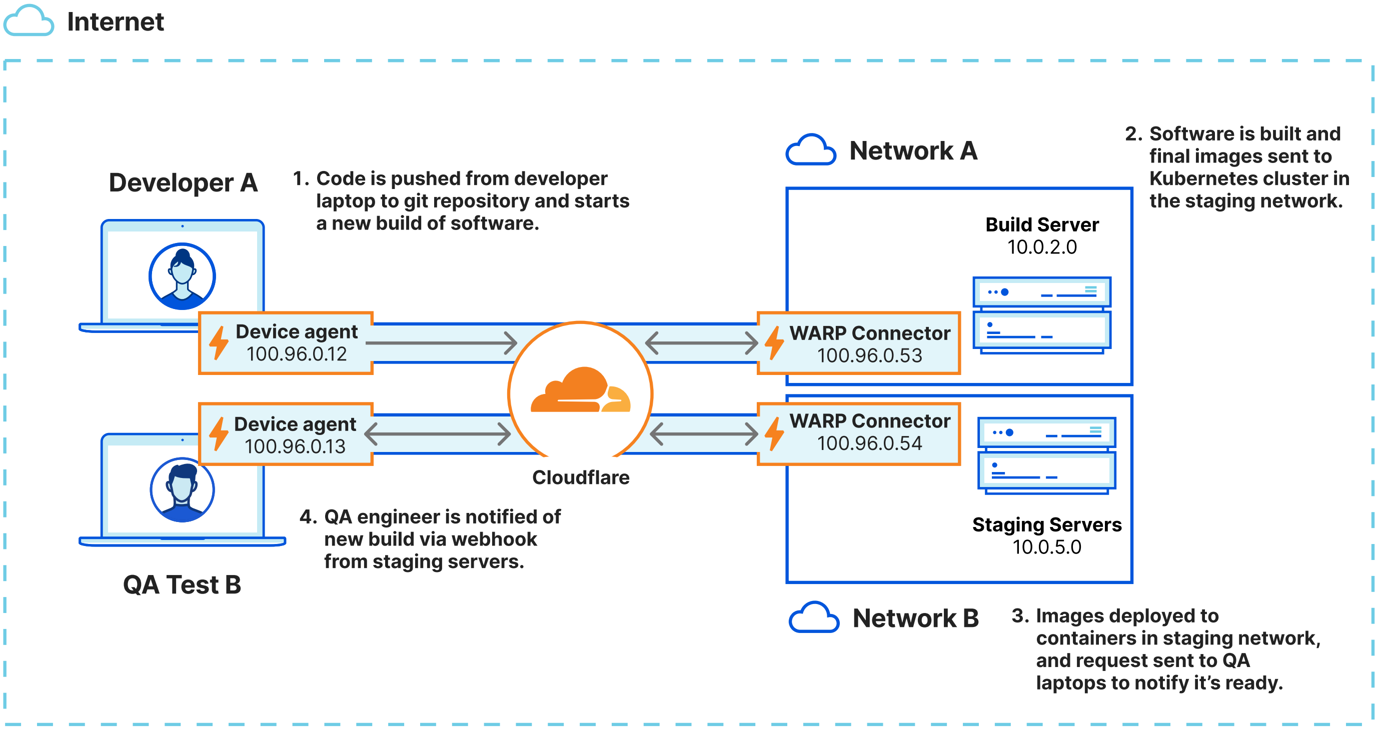
For bidirectional, or meshed connectivity, organizations should use the WARP Connector.

Mesh connectivity

The WARP Connector is a lightweight solution for site-to-site, bidirectional, and mesh networking connectivity that does not require changes to underlying network routing infrastructure. WARP Connector software is installed on a Linux server within an organization’s network, which then becomes a gateway for other local networks that need to on-ramp traffic to Cloudflare.

This provides a lightweight solution to support services such as Microsoft’s System Center Configuration Manager (SCCM), Active Directory server updates, VOIP and SIP traffic, and developer workflows with complex CI/CD pipeline interaction. It can either be run supplementally to cloudflared and Magic WAN, or can be a standalone remote access and site-to-site connector to the Cloudflare network.

The WARP Connector can proxy both user-to-network and network-to-network connectivity, or can be used to establish an overlay network of Carrier Grade NAT (CGNAT) addressed endpoints to provide secure, direct connectivity to established resources using CGNAT IP ranges. This helps address overlapping network IP range challenges, point-solution access problems, or the process of shifting network design without impacting a greater underlying system.



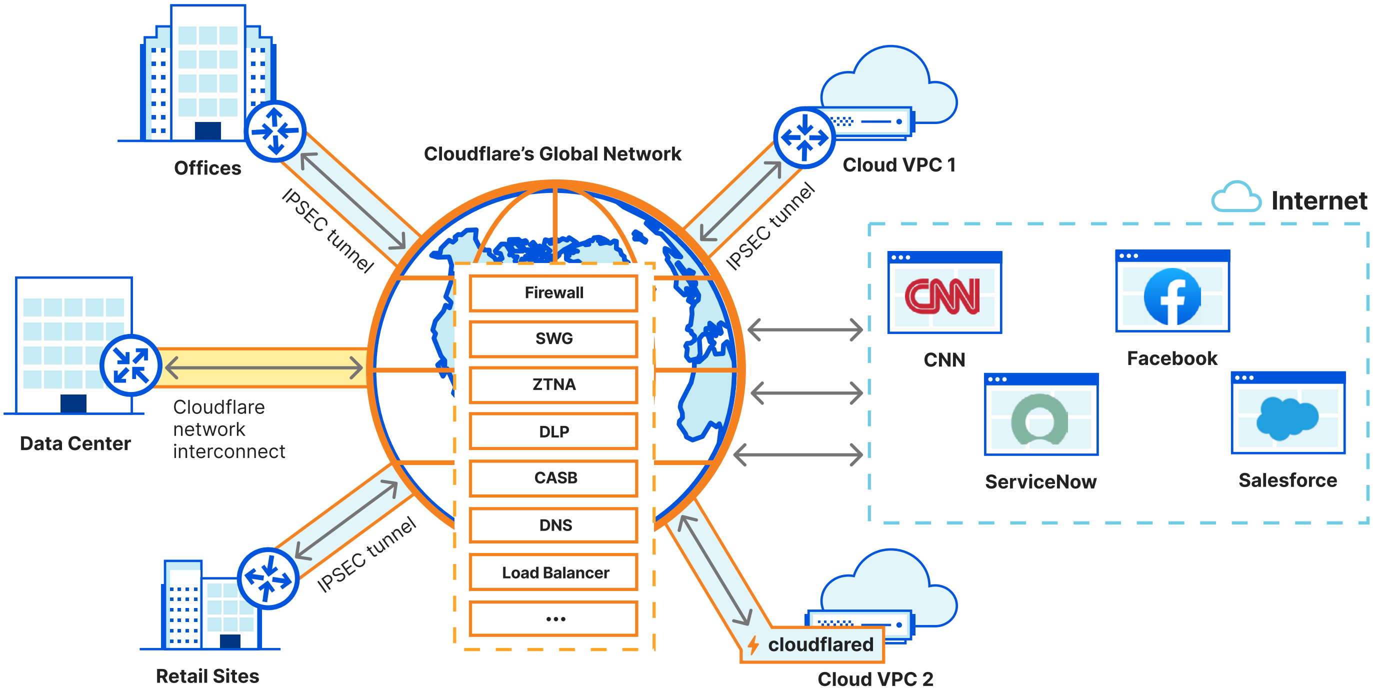
1. Code is pushed from developer laptop to git repository and starts a new build of software.
2. Software is built and final images sent to Kubernetes cluster in the staging network.
3. Images deployed to containers in staging network, and request sent to QA laptops to notify it’s ready.
4. QA engineer is notified of new build via webhook from staging servers.

Cloudflare Tunnel via *cloudflared* is the primary method for connecting users to applications and services on private networks because it is a simpler, more granular and agile solution for many application owners (vs. IP tunnel based connectivity technology, like IPsec and GRE). Cloudflare Tunnel via WARP Connector is the preferred method for mesh or other software-defined networking — most of which require bidirectional connectivity — when organizations do not want to make changes to the underlying network routing or edge infrastructure.

Using network equipment

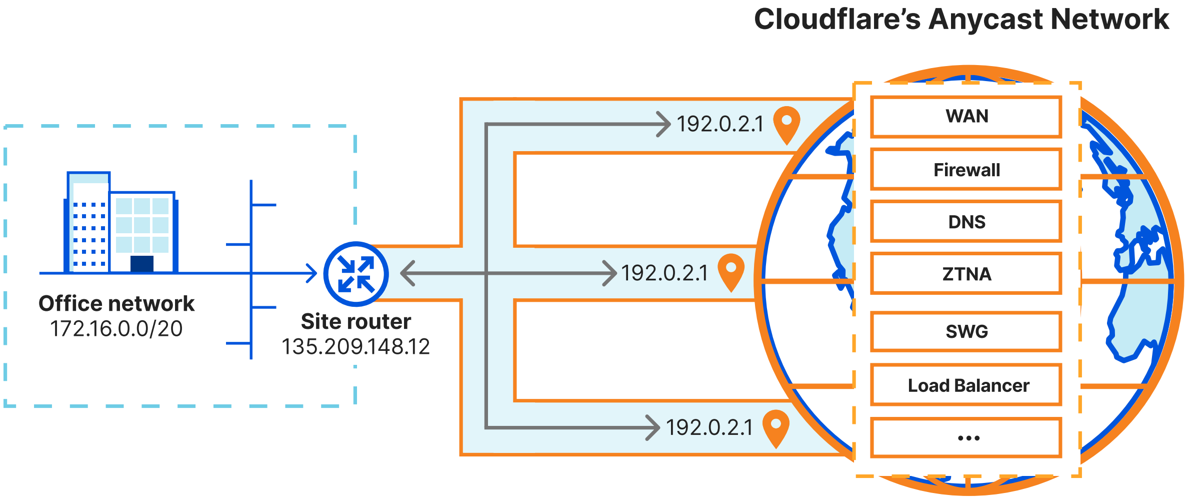
Where it is not optimal or possible to install software agents, networks can also be connected to Cloudflare using existing network equipment, such as routers and network firewalls. To do this, organizations create IPsec or GRE tunnels that connect to Cloudflare's cloud-native Magic WAN service. With Magic WAN, existing network hardware can connect and route traffic from their respective network locations to Cloudflare through a) secure, IPsec-based tunnels over the Internet or, b) across Cloudflare Network Interconnect (CNI) — private, direct connections that link existing network locations to the nearest Cloudflare data center.

Cloudflare's WAN service uses a "light-branch, heavy-cloud" architecture that represents the evolution of software-defined WAN (SD-WAN) connectivity. With Magic WAN, as depicted in the network architecture diagram below, the Cloudflare global network functions as a centrally-managed connectivity hub that securely and efficiently routes traffic between all existing network locations:



As previously described, Cloudflare uses a routing technique called Anycast to globally advertise all of the services and endpoints on the Cloudflare network, including the endpoints for WAN IP tunnels.

With Anycast IPsec or Anycast GRE tunnels, each tunnel configured from an organization's network device (e.g. edge router, firewall appliance, etc.) connects to hundreds of global Cloudflare data centers. Traffic sourced from an organization's network location is sent directly over these tunnels and always routes to the closest active Cloudflare data center. If the closest Cloudflare data center is unavailable, the traffic is automatically rerouted to the next-closest data center.



To further network resiliency, Magic WAN also supports Equal Cost Multi-Path (ECMP) routing between the Cloudflare network and an organization's network location(s). With ECMP, traffic can be load-balanced across multiple Anycast IP tunnels, which helps increase throughput and maximize network reliability. In the event of network path failure of one or more tunnels, traffic can be automatically failed over to the remaining healthy tunnels.

The simplest and easiest way to on-ramp existing network locations to the Magic WAN service is to deploy Cloudflare Magic WAN Connector: a plug-and-play, fully cloud-managed network device that can be deployed in any physical or cloud network. When the WAN Connector is installed into a network, it will automatically establish communication with the Cloudflare network, download and provision relevant configurations, establish resilient IPsec tunnels, and route connected site network traffic to Cloudflare.

The WAN Connector can be deployed as a hardware appliance and will soon be released as a software appliance, making it versatile for a variety of user network environments — on-premises, virtual, or public cloud. Management, configuration, observability, and software updates for WAN Connectors is centrally managed from Cloudflare via either the dashboard or the Cloudflare API. As of 2023, the WAN Connector is currently best-suited for connecting small and medium-sized networks to Cloudflare (e.g. small offices and retail stores).

In situations where deploying the WAN Connector is not feasible or desirable, organizations can securely connect their site networks to Cloudflare by configuring IPsec tunnels from their existing IPsec-capable network devices, including WAN or SD-WAN routers, firewalls, and cloud VPN gateways. Please refer to the Cloudflare documentation for up-to-date examples of validated IPsec devices.

There may also be situations where network-layer encryption is not necessary — for example, when a site's WAN-bound traffic is already encrypted at the application layer (via TLS), or when an IPsec network device offers very limited throughput performance as it encrypts and decrypts IPsec traffic. Under these circumstances, organizations can connect to the Cloudflare network using GRE tunnels.

Organizations may also connect their network locations directly to the Cloudflare network via Network Interconnect (CNI). Currently, Cloudflare supports two types of network interconnect:

1. **Private network interconnect (PNI)**. With PNI, data centers must be co-located with a Cloudflare Interconnection facility, where the link size of the connection is 10GbE or higher.
2. **Virtual private network interconnect (vPNI)**. With vPNI, organizations' data centers do not have to be colocated with a Cloudflare data center. vPNI also works for organizations that are already using services from Cloudflare's Interconnection partners. The connection size of vPNIs depends on the offering by these partners.

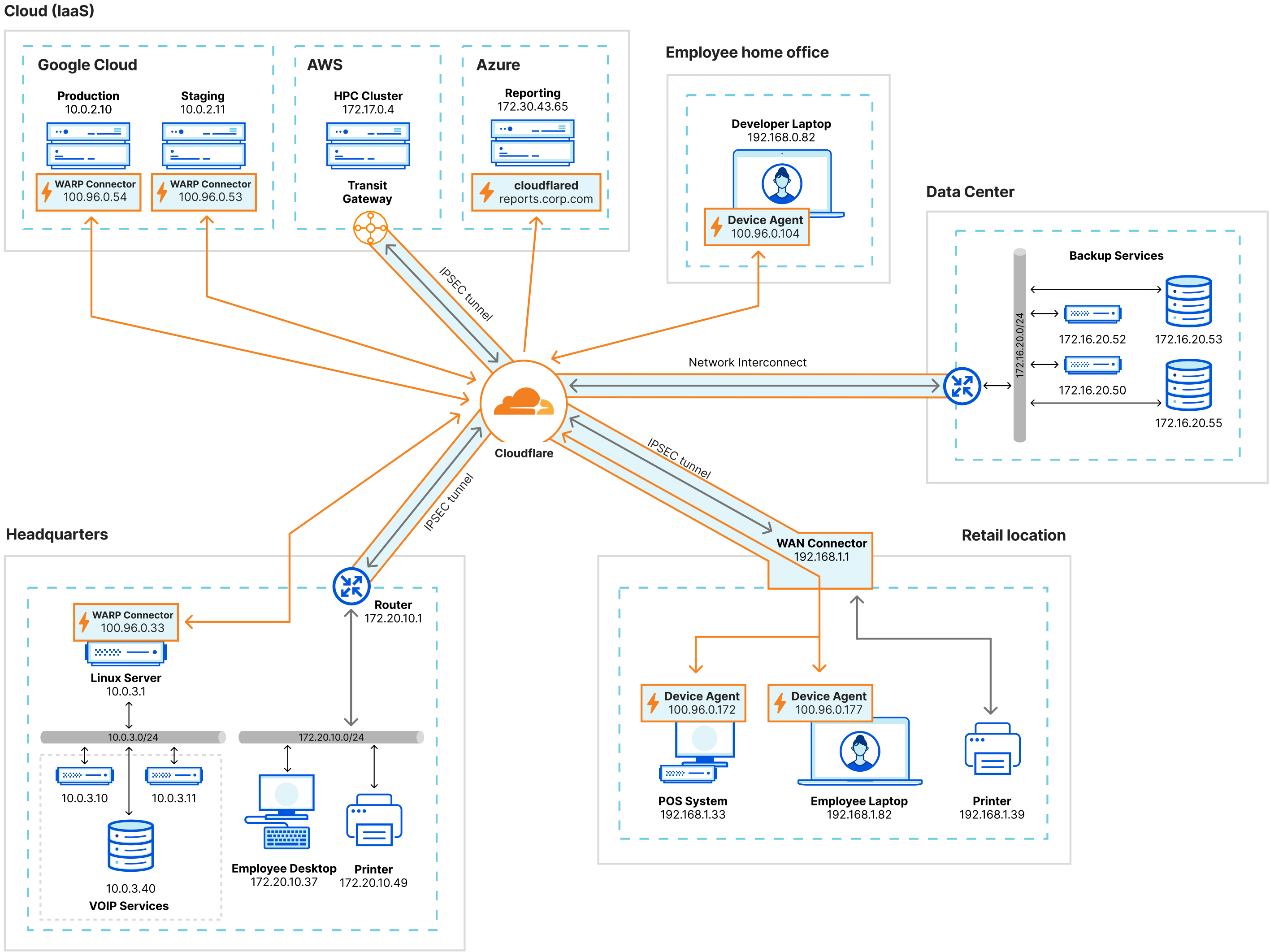
The following table summarizes the different methods of connecting networks to Cloudflare:

Use case	Recommended	Alternative solution
Remote users connecting to applications on private networks in a Zero Trust model (e.g. most VPN replacement scenarios)	**Cloudflare Tunnel** **(with *cloudflared*)**	**Magic WAN** Alternative option if *cloudflared* not suitable for environment
Site-to-site connectivity between branches, headquarters, and data centers	**Magic WAN**	**Cloudflare Tunnel (with WARP Connector)** Alternative option if routingchanges cannot be made at perimeter
Egress traffic from physical sites or cloud environments to cloud security inspection (e.g. most common SWG and branch firewall replacement scenarios)	**Magic WAN**	**N/A**
Service-initiated communication with remote users (e.g. AD or SCCM updates, DevOps workflows, VOIP)	**Cloudflare Tunnel** **(with WARP Connector)**	**Magic WAN** Alternative option if inbound source IP fidelity not required
Mesh networking and peer-to-peer connectivity	**Cloudflare Tunnel** **(with WARP Connector)**	**N/A**

Each of these methods of connecting and routing traffic can be deployed concurrently from any location. The following diagram highlights how different connectivity methods can be used in a single architecture.

Note the following traffic flows:

- All traffic connected via a WARP Connector or device agent can communicate with each other over the mesh network
    - Developers working from home can communicate with the production and staging servers in the cloud
    - The employee in the retail location, as well as the developer at home, can receive VOIP calls on their laptop
- A HPC Cluster in AWS represents a proprietary solution in which no third-party software agents can be installed; as a result, it uses an IPsec connection to Magic WAN
- In the retail location, the Magic WAN Connector routes all traffic to Cloudflare via an IPsec tunnel
    - An employee's laptop running the device agent creates its own secure connection to Cloudflare that is routed over the IPsec tunnel
- The application owner of the reporting system maintains a connection to Cloudflare using cloudflared and doesn't require any networking help to expose their application to employees

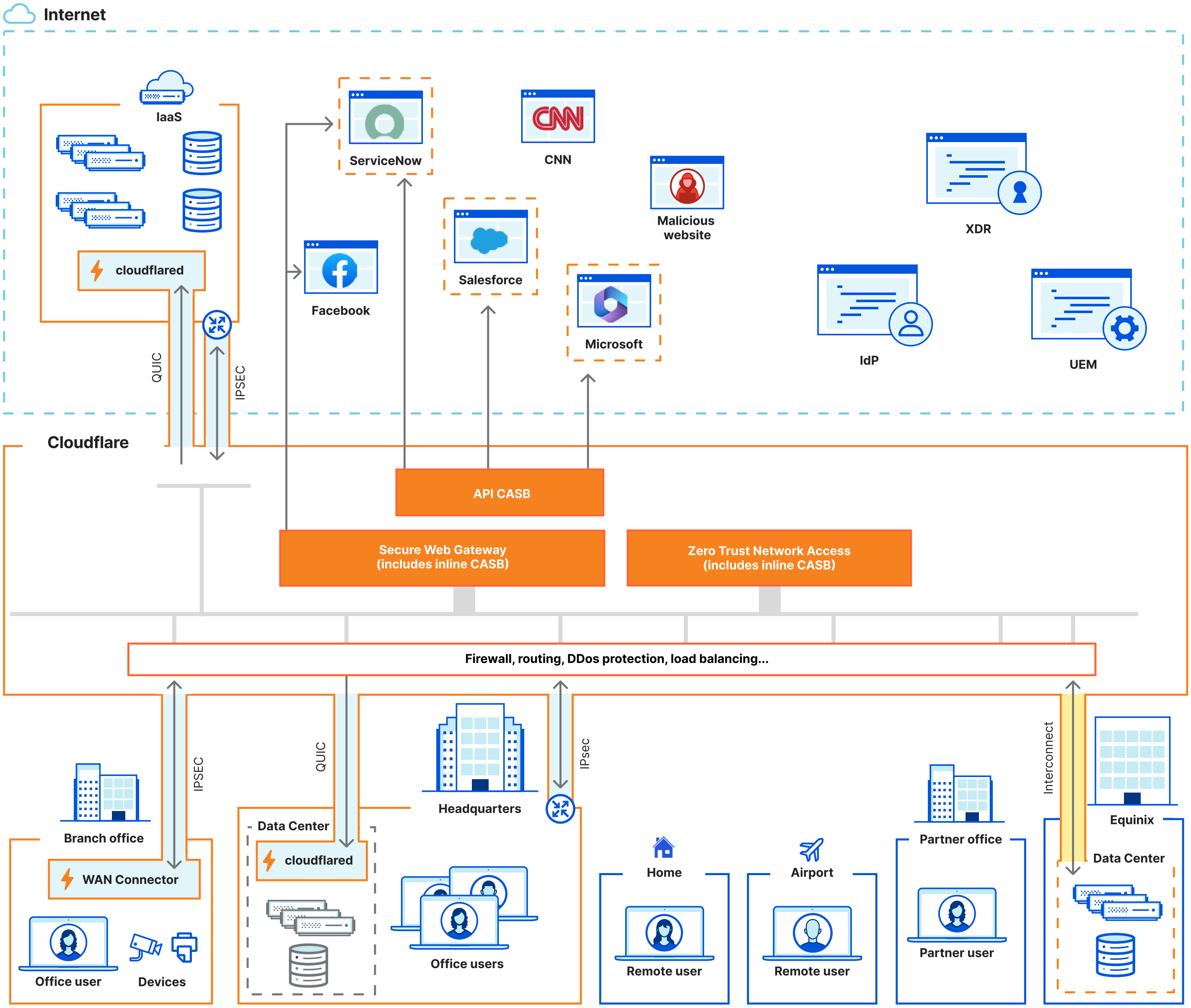


Note: All of the endpoints connected via the WARP connector or device agent are automatically assigned IP addresses from the 100.96.0.0/12 address range, while endpoints connected to Magic WAN retain their assigned RFC1918 private IP addresses. Cloudflared can be deployed in any of the locations by an application owner to provide hostname-based connectivity to the application.

Once the networks, applications, and user devices are connected to Cloudflare — regardless of the connection methods and devices used — all traffic can be inspected, authenticated, and filtered by the Cloudflare SASE services, then securely routed to their intended destinations. Additionally, consistent policies can be applied across all traffic, no matter how it arrives at Cloudflare.

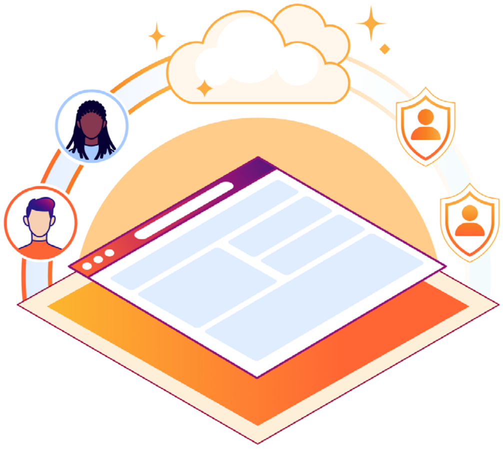
Checkpoint: Connecting networks to Cloudflare

Now this is what a SASE architecture looks like where corporate network traffic from everywhere is forwarded to and processed by Cloudflare. In this architecture, it is possible to make a network connection from any remote location, office location or data center and connect to applications and services living in SaaS infrastructure, cloud-hosted infrastructure or an organization's own on-premise data centers.



Forwarding device traffic

The previous sections explain using ZTNA to secure access to self-hosted applications and using an SW to inspect and filter traffic destined for the Internet When a user is working on a device in any of the company networks that is connected to Cloudflare' connectivity cloud, all that traffic is inspected and policies applied without disrupting the user's workflow. Yet, users are not always (or ever) in the office; they work from home, on the road, or from other public networks. How do you ensure they hav reliable access to your internal applications? How do you ensure their Internet browsing is secure no matter their work location?



There are several approaches to ensure that traffic from a user device which isn't connected to an existing Cloudflare protected network, are also forwarding traffic through Cloudflare and be protected.

- Install an agent on the device
- Modify browser proxy configuration
- Direct the user to a remote browser instance
- Modify DNS configuration

Connecting with a device agent

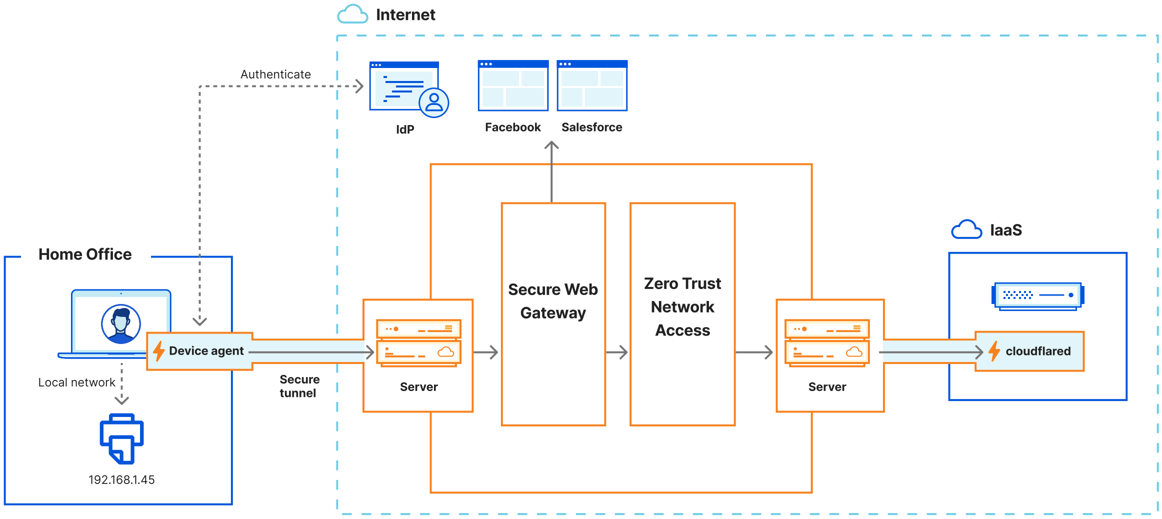
The preferred method of ensuring device traffic is forwarded to Cloudflare is to install the device agent (also referred to as Cloudflare WARP). The agent runs on Windows, macOS, Linux, iOS, and Android/ChromeOS, and creates a secure connection to Cloudflare where all non-local traffic is sent. Because of Cloudflare's use of Anycast networking, the device agent always connects to the nearest Cloudflare server to ensure the best performance for the user. The device agent also collects local machine and network information, which is sent in the request to enrich the policy in Cloudflare.

To allow for flexibility in how different devices and users connect, there are multiple deployment modes:

- A full L4 traffic proxy
- L7 DNS proxy
- L7 HTTP proxy
- The ability to just collect device posture information

For example, organizations might have an office that continues to use an existing DNS filtering service, so they can configure the agent to just proxy network and HTTP traffic.

The agent can also be configured with flexible routing controls that allow for scenarios in which traffic destined for office printers is not sent to the Cloudflare network but, instead, routed to the local network. These split tunnel configurations can be made specific to groups of users, types of device operating system, or networks and by default, traffic destined to all private IPv4 and IPv6 ranges is sent to the device's default gateway. If the application the user is attempting to reach is not in public DNS, you can configure the hostname and domain to be resolved with local DNS services, so that the device agent does not attempt to resolve these using Cloudflare DNS.



The agent is more than just a network proxy; it is able to examine the device's security posture, such as if the operating system is fully up-to-date or if the hard disk is encrypted. Cloudflare's integrations with CrowdStrike, SentinelOne, and other third-party services also provide additional data about the security posture of the device. All of this information is associated with each request and, therefore, available for use in company policies — as explained in the "Unified Management" section.

The agent can be deployed to a device either manually or using existing endpoint management (UEM) technologies. Using the agent, users register and authenticate their device to Cloudflare with the integrated identity providers. Identity information — combined with information about the local device — is then used in your SWG and ZTNA policies (including inline CASB capabilities shared across these Cloudflare services).

Browser proxy configuration

When it is not possible to install software on the device, there are agentless approaches.

One option is to configure the browser to forward HTTP requests to Cloudflare by configuring proxy server details in the browser or OS. Although this can be done manually, it is more common for organizations to automate the configuration of browser proxy settings using Internet-hosted Proxy Auto-Configuration (PAC) files. The browser identifies the PAC file location in several ways:

- MDM software configuring the setting in the browser
- In Windows domains, Group Policy Objects (GPO) can configure the browser's PAC file
- Browsers can use Web Proxy Auto-Discovery (WPAD)

From there, configure a proxy endpoint where the browser will send all HTTP requests to. If using this method, please note that:

- Filtering HTTPS traffic will also require installing and trusting Cloudflare root certificates on the devices.
- A proxy endpoint will only proxy traffic sourced from a set of known IP addresses, such as the pool of public IP addresses used by a site's NAT gateway, that the administrator must specify.

Using remote browser instances

Another option to ensure device traffic is sent to Cloudflare is to use remote browser isolation (RBI). When a remote user attempts to visit a website, the corresponding requests and responses are handled by a headless remote browser running in the Cloudflare network that functions as a "clone" of the user device's local browser. This shields the user's device from potential harmful content and code execution that may be downloaded from the website it visits.

RBI renders the received content in an isolated and secure cloud environment. Instead of executing the web content locally, the user device receives commands for how to 'draw' the final rendered web page over a highly optimized protocol supported by all HTML5-compliant browsers on all operating systems. Because the remote browser runs on Cloudflare's servers, SWG policies are automatically applied to all browser requests.

Ensuring access to sites is protected with RBI does not require any local software installation or reconfiguring the user's browser. Below are several ways to accomplish this:

- Typically, a remote browser session is started as the result of an SWG policy — the user just requests websites without being notified that the content is loading in a remote browser.
- Organizations can also provide users with a link that automatically ensures RBI always processes each request.
- Organizations can also opt to use the ZTNA service to redirect all traffic from self-hosted applications via RBI instances.

All requests via a remote browser pass through the Cloudflare SWG; therefore, policies can enforce certain website access limitations. For instance, browser isolation policies can be established to:

- Disable copy/paste between a remote web page and the user's local machine; this can prevent the employee from pasting proprietary code into third-party chatbots.
- Disable printing of remote web content to prevent contractors from printing confidential information
- Disable file uploads/downloads to ensure sensitive company data is not sent to — or downloaded from — certain websites.
- Disable keyboard input (in combination with other policies) to limit data being exposed, such as someone typing in passwords to a phishing site.

Isolating web applications and applying policies to risky websites helps organizations limit data loss from cyber threats or user error. And, like many Cloudflare One capabilities, RBI can be leveraged across other areas of the SASE architecture. Cloudflare's email security service, for example, can automatically rewrite and isolate suspicious links in emails. This "email link isolation" capability helps protect the user from potential malicious activity such as credential harvesting phishing.

Agentless DNS Filtering

Another option for securing traffic via the Cloudflare network is to configure the device to forward DNS traffic to Cloudflare to be inspected and filtered. First DNS locations are created which allow policies to be applied based on different network locations. They can be determined either by the source IP address for the request or you can use "DNS over TLS" or "DNS over HTTPS".

When using source IP addresses, either the device will need to be told which DNS servers to use, or the local DNS server on the network the device is connected to needs to forward all DNS queries to Cloudflare. For DNS over TLS or HTTPS support, the devices need to be configured and support varies. Our recommendation is to use DNS over HTTPS which has wider operating system support.

All of the above methods result in only the DNS requests — not all traffic — being sent to Cloudflare. SWG DNS policies are then implemented at this level to manage access to corporate network resources.

Summary of SWG capabilities for each traffic forwarding method

The following table summarizes SWG capabilities for the various methods of forwarding traffic to Cloudflare (as of Oct 2023):

	Connection method (for traffic source)				
	IP tunnel or Interconnect (Magic WAN)	**Device Agent (WARP) *[1]**	**Remote Browser**	**Browser proxy**	**DNS proxy**
Types of traffic forwarded	**TCP/UDP**	TCP/UDP	HTTP	HTTP	DNS
Policy types					
DNS	✓	✓	✗	✗	✓
HTTP/S *[2]	✓	✓	✓	✓	N/A
Network (L3/L4 parameter)	✓	✓	✓	✓	✗
Data available in policies					
Identity information	✗	✓	✓	✗	✗ *[3]
Device posture	✗	✓	✗	✗	✗
Capabilities					
Remote browser isolation	✓	✓	✓	✓	N/A
Enforce egress IP	✓	✓	✓	✓	N/A

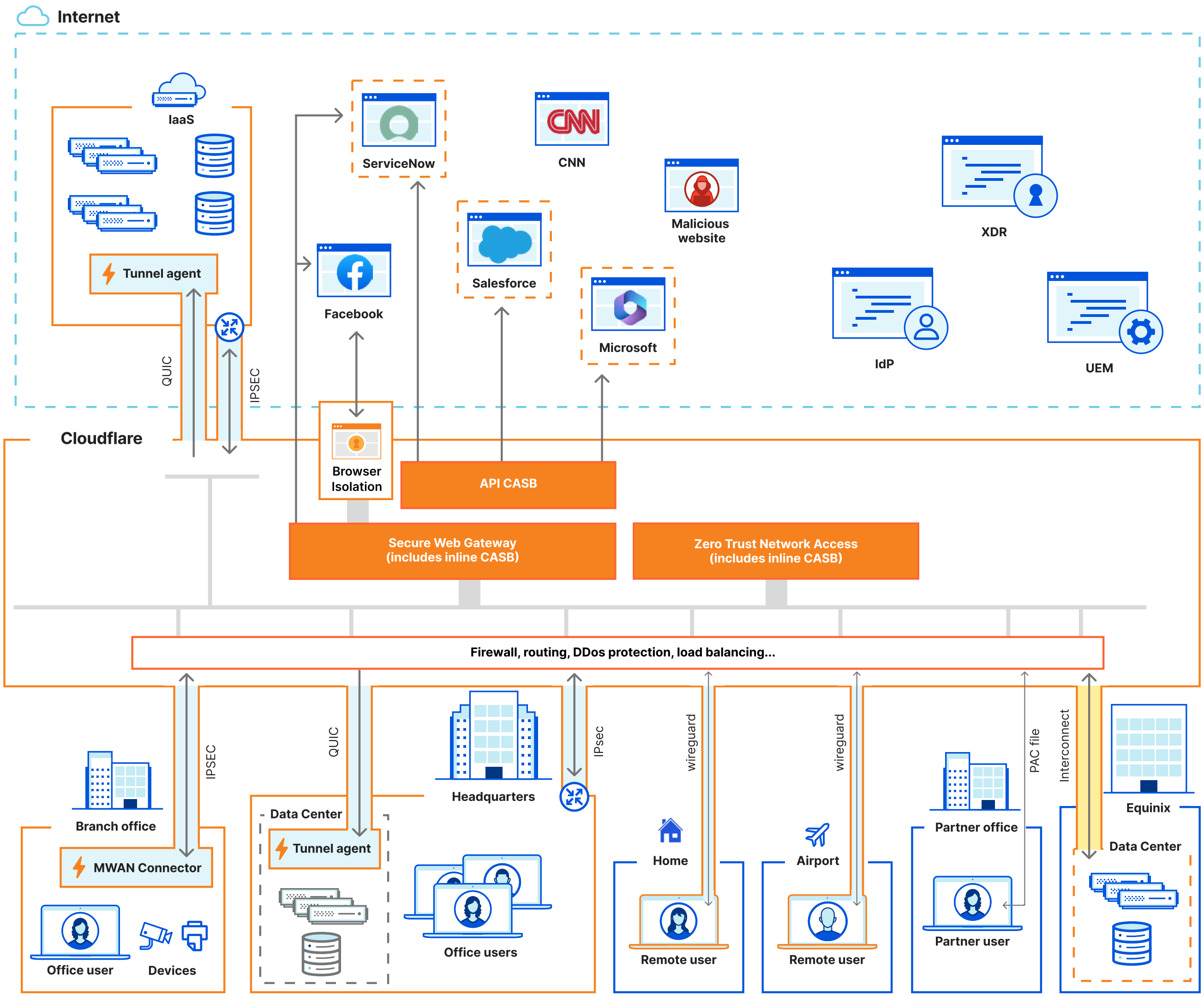
Notes:

1. Running the device agent in DNS over HTTP mode provides user identity information, in addition to the same capabilities as connecting via DNS.
2. To filter HTTPS traffic, the Cloudflare certificate needs to be installed on each device. This can be automated when using the device agent.
3. If configuring DNS over HTTPS, it is possible to inject a service token into the request, which associates the query with an authenticated user.

Checkpoint: Forwarding device traffic to Cloudflare

By connecting entire networks or individual devices, organizations can now route user traffic to Cloudflare for secure access to privately-hosted applications and secure public Internet access.

Once traffic from all user devices is forwarded to the Cloudflare network, it is time for organizations to revisit their high-level SASE architecture:



Verifying users and devices

At this point in implementing SASE architecture, organizations have the ability to route and secure traffic beginning from the point a request is made from a browser on a user's device, all the way through Cloudflare's network to either a company-hosted private application/service or to the public Internet.

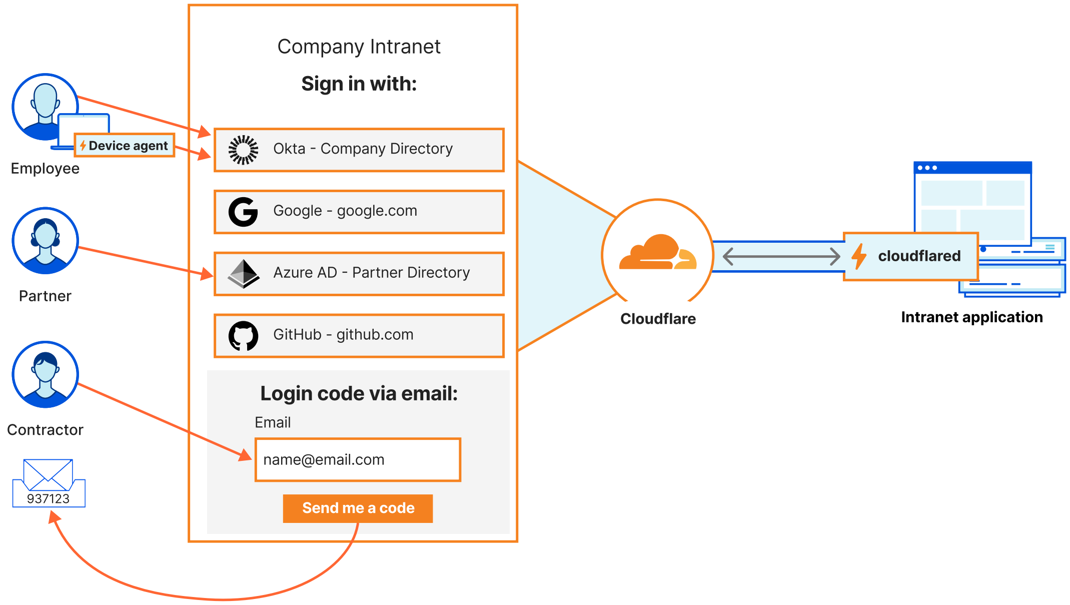
But, before organizations define policies to manage that access, they need to know who is making the request and determine the security posture of the device.

Integrating identity providers

The first step in any access decision is to determine who is making the request – i.e., to authenticate the user.

Cloudflare integrates with identity providers that manage secure access to resources for organizations' employees, contractors, partners, and other users. This includes support for integrations with any SAML - or OpenID Connect (OIDC) - compliant service; Cloudflare One also includes pre-built integrations with Okta, Microsoft Azure AD, Google Workspace, as well as consumer IdPs such as Facebook, GitHub and LinkedIn.

Multiple IdPs can be integrated, allowing organizations to apply policies to a wide range of both internal and external users. When a user attempts to access a Cloudflare secured application or service, they are redirected to authenticate via one of the integrated IdPs. When using the device agent, users must also authenticate to one of their organization's configured IdPs.



Once a user is authenticated, Cloudflare receives that user's information, such as username, group membership, authentication method (password, whether MFA was involved and what type), and other associated attributes (i.e., the user's role, department, or office location). This information from the IdP is then made available to the policy engine.

In addition to user identities, most corporate directories also contain groups to which those identities are members. Cloudflare supports the importing of group information, which is then used as part of the policy. Group membership is a critical part of aggregating single identities so that policies can be more simply written. It is far easier — for example — to set a policy allowing all employees in the sales department to access Salesforce, than to identify each user in the sales organization.

Cloudflare also supports authentication of devices that are not typically associated with a human user – such as an IoT device monitoring weather conditions at a factory. For those secure connections, organizations can generate service tokens or create Mutual TLS (mTLS) certificates that can be deployed to such devices or machine applications.

Trusting devices

Not only does the user identity need to be verified, but the security posture of the user's device needs to be assessed. The device agent is able to provide a range of device information, which Cloudflare uses to build comprehensive security policies.

The following built-in posture checks are available:

- Application check: Checks that a specific application process is running
- File check: Checks for the presence of a file
- Firewall: Checks if a firewall is running
- Disk encryption: Checks if/how many disks are encrypted
- Domain joined: Checks if the device is joined to a Microsoft Active Directory domain
- OS version: Checks what version of the OS is running
- Unique Client ID: When using an MDM too, organizations can assign a verifiable UUID to a mobile, desktop, or laptop device
- Device serial number: Checks to see if the device serial matches a list of company desktop/laptop computers

Cloudflare One can also integrate with any deployed endpoint security solution, such as Microsoft Endpoint Manager, Tanium, Carbon Black, CrowdStrike, SentinelOne, and more. Any data from those products can be passed to Cloudflare for use in access decisions.

All of the above device information, combined with data on the user identity and also the network the device is on, is available in Cloudflare to be used as part of the company policy. For example, organizations could choose to only allow administrators to SSH into servers when all of the following conditions are met: their device is free from threats, running the latest operating system, and joined to the company domain.

Because this information is available for every network request, any time a device posture changes, its ability to connect to an organization's resources is immediately impacted.

Integrating email services

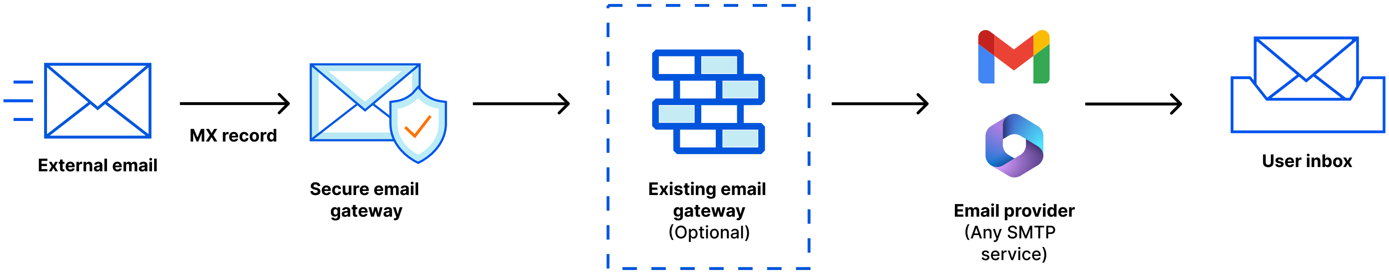
Email — the #1 communication tool for many organizations and the most common channel by which phishing attacks occur — is another important corporate resource that should be secured via a SASE architecture. Phishing is the root cause of upwards of 90% of breaches that lead to financial loss and brand damage.

Cloudflare's email security service scans for signs of malicious content or attachments before they can reach the inbox, and also proactively scans the Internet for attacker infrastructure and attack delivery mechanisms, looking for programmatically-created domains that are used to host content as part of a planned attack. Our service uses all this data to also protect against business and vendor email compromises (BEC / VEC), which are notoriously hard to detect due to their lack of payloads and ability to look like legitimate email traffic.

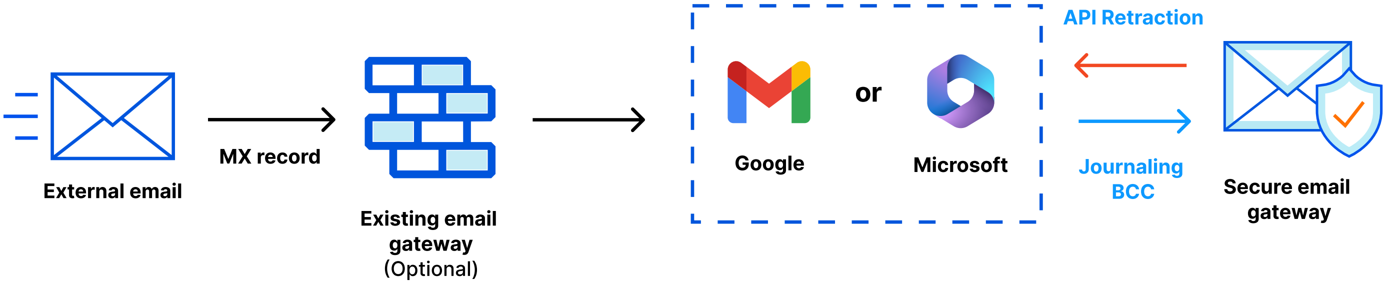
Instead of deploying tunnels to manage and control traffic to email servers, Cloudflare provides two methods of email security setup:

- Inline: Redirect all inbound email traffic through Cloudflare before they reach a user's inbox by modifying MX records
- API: Integrate Cloudflare directly with an email provider such as Microsoft 365 or Gmail

Modifying MX records (inline deployment) forces all inbound email traffic through our cloud email security service where it is scanned, and — if found to be malicious — blocked from reaching a user's inbox. Because the service works at the MX record level, it is possible to use the email security service with any SMTP-compliant email service.

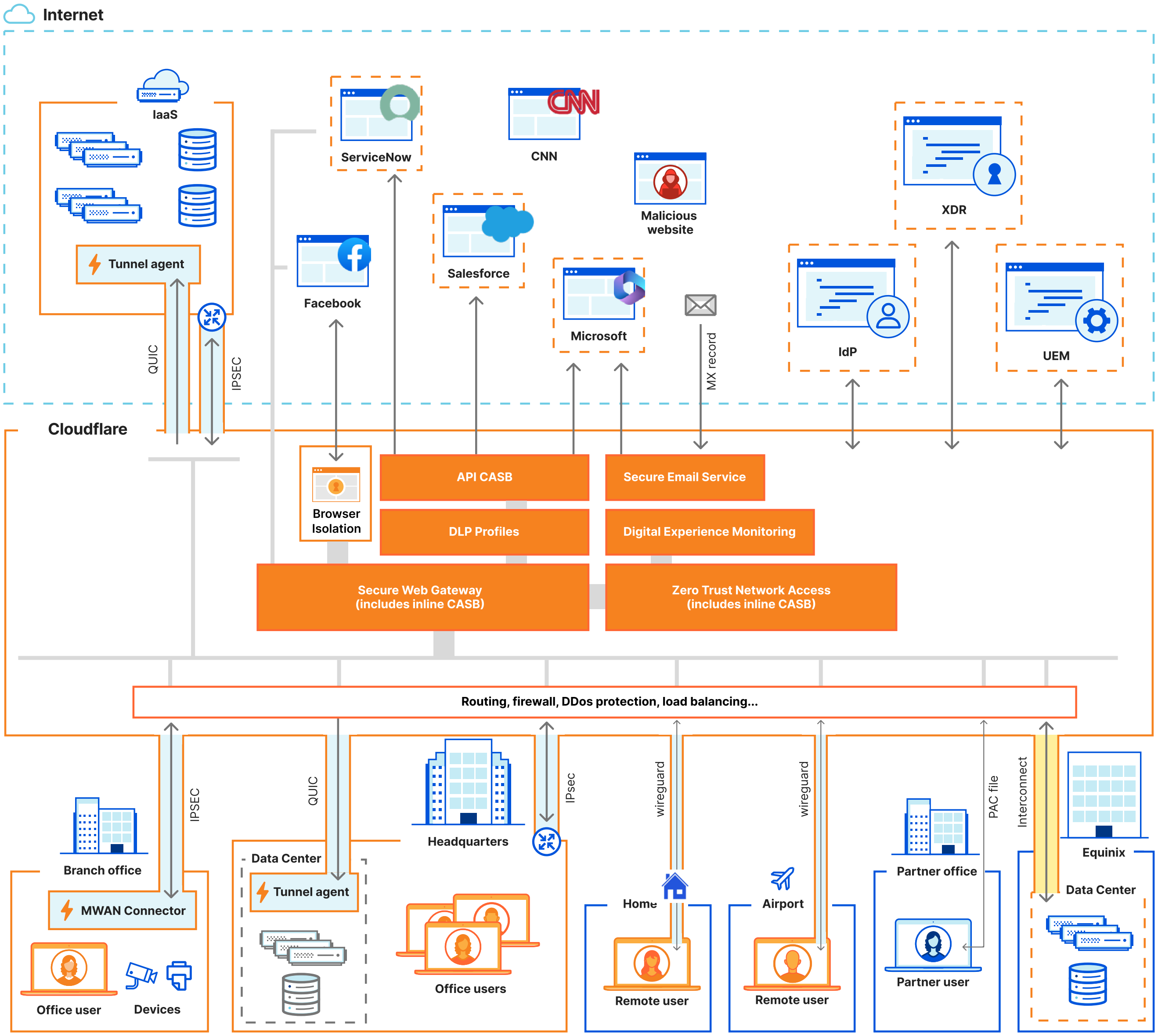

External email
MX record
Secure email gateway
Existing email gateway (Optional)
Email provider (Any SMTP service)
User inbox

Organizations can also opt to integrate email security directly with their email service via APIs. Note that this approach has two drawbacks: there are fewer integrations Cloudflare supports and there is always a small delay between the email being delivered to the service and Cloudflare detecting it via the API.



Checkpoint: A complete SASE architecture with Cloudflare

The steps above provide a complete view of evolving to SASE architecture using Cloudflare One. As the diagram below shows, secure access to all private applications, services, and networks — as well as ensuring the security of users' general Internet access — is now applied to all users in the organization, internal or external.



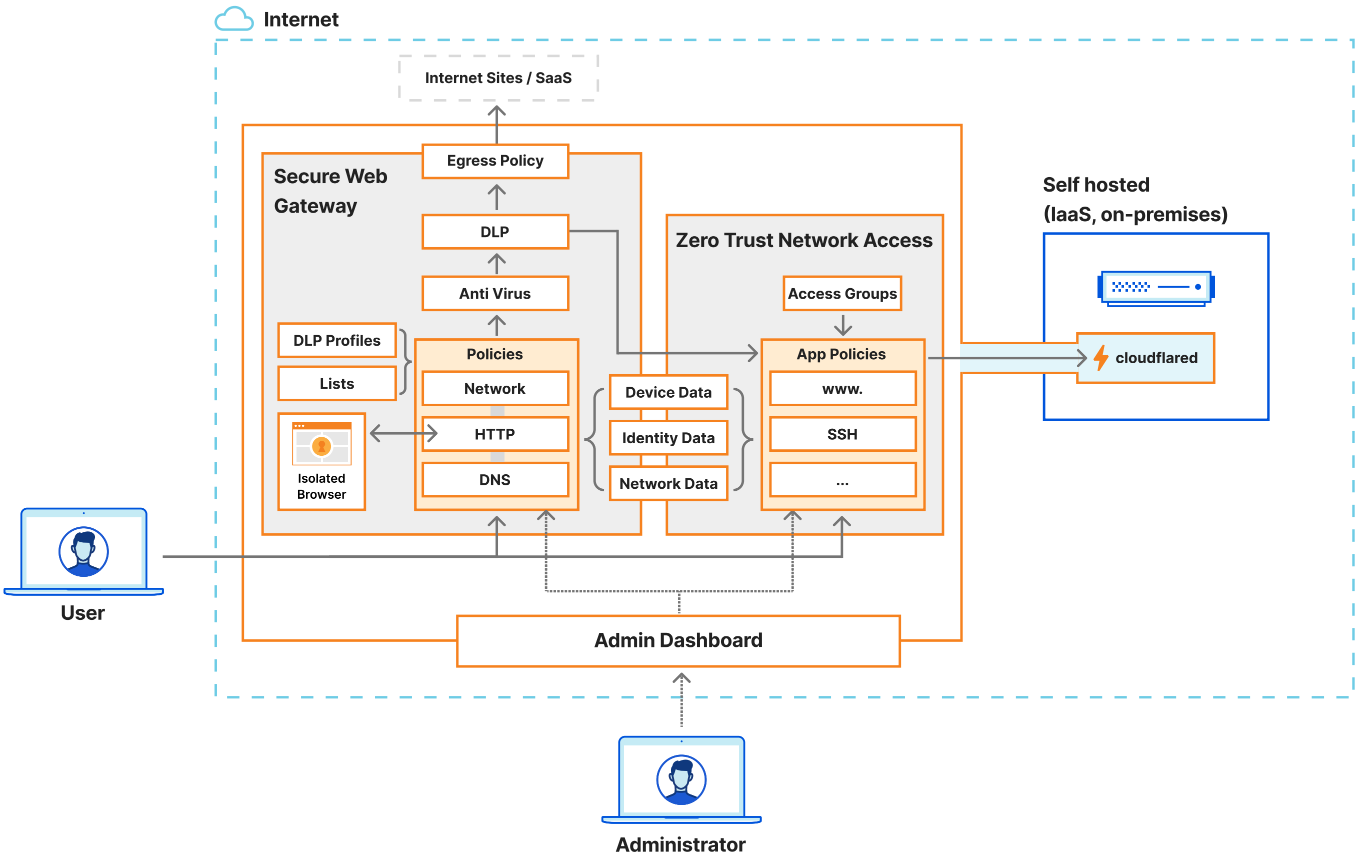
For ease of use, the entire Cloudflare One platform can be configured via API; and with Cloudflare's Terraform provider, organizations can manage the Cloudflare global network using the same tools they use to automate the rest of their infrastructure. This allows IT teams to fully manage their Cloudflare One infrastructure, including all the policies detailed in the next section, using code. There are also (as of Oct 2023) more than 500 GitHub repositories, many of which allow IT teams to use and build tools to manage their Cloudflare deployment.

Unified management

Now that all users, devices, applications, networks, and other components are seamlessly integrated within a SASE architecture, Cloudflare One provides a centralized platform for comprehensive management. Because of the visibility Cloudflare has across the entire IT infrastructure, Cloudflare can aggregate signals from various sources, including devices, users, and networks. These signals can inform the creation of policies that govern access to organization resources.

Before we go into the details of how policies can be written to manage access to applications, services, and networks connected to Cloudflare, it's worth taking a look at the two main enforcement points in Cloudflare's SASE platform that control access: SWG and the ZTNA services. These services are configured through a single administrative dashboard, simplifying policy management across the entire SASE deployment.

The following diagram illustrates the flow of a request through these services, including the application of policies and the source of data for these policies. In the diagram below, the user request can either enter through the SWG or ZTNA depending on the type of service requested. It's also possible to combine both services, such as implementing a SWG HTTP policy that uses DLP service to inspect traffic related to a privately hosted application behind a ZTNA Cloudflare Tunnel. This configuration enables organizations to block downloads of sensitive data from internal applications that organizations have authorized for external access.



In the following sections, we introduce examples of how different policies can be configured to satisfy specific use cases. While these examples are not exhaustive, the goal is to demonstrate common ways Cloudflare One can be configured to address the challenges organizations encounter in its transition to a SASE architecture.

Connecting an IdP to Cloudflare provides the ability to make access decisions based on factors such as group membership, authentication method, or specific user attributes. Cloudflare's device agent also supplies additional signals for policy considerations, such as assessing the operating system or verifying the device's serial number against company-managed devices. However, there are features that allow users to incorporate additional data into deployment for building simple but powerful policies.

Lists

Cloudflare's vast intelligent network continually monitors billions of web assets and categorizes them based on our threat intelligence and general knowledge of Internet content. You can use our free Cloudflare Radar service to examine what categories might be applied to any specific domain. Policies can then include these categories to block known and potential security risks on the public Internet, as well as specific categories of content.

Additionally, Cloudflare's SWG offers the flexibility to create and maintain customized lists of data. These lists can be uploaded via CSV files, manually maintained, or integrated with other processes and applications using the Cloudflare API. A list can contain the following data:

- URLs
- Hostnames
- Serial numbers (MacOS, Windows, Linux)
- Emails
- IP addresses
- Device IDs (iOS, Android)

For example, organizations can maintain a list of IP addresses of all remote office locations, of short term contractors' email addresses, or trusted company domains. These lists can be used in a policy to allow contractors access to a specific application if their traffic is coming from a known office IP address.

DLP profiles and datasets

Cloudflare looks at various aspects of a request, including the source IP, the requested domain, and the identity of the authenticated user initiating the request. Cloudflare also offers a DLP service which has the ability to detect and block requests based on the presence of sensitive content. The service has built in DLP profiles for common data types such as financial information, personally identifiable information (PII), and API keys.

There is even a profile for source code, so users can detect and block the transfer of C++ or Python files. Organizations can create customized DLP profiles and use regular expressions to define the patterns of data they are looking for. For data that is hard to define a pattern for, datasets can be used which match exact data values. These datasets allow for the bulk upload of any data to be matched, such as lists of customer account IDs or sensitive project names. These profiles and data sets can be incorporated into policies to prevent users from downloading large files containing confidential customer data.

To reduce the risk of false positives, internal users have the option to establish a match count on the profile. This means that a specific number of matches within the data are required before profile triggers. This approach prevents scenarios where a random string resembling PII or a credit card number would trigger the profile unnecessarily. By implementing a match count, the policy demands that multiple data elements align with the profile, significantly increasing its accuracy.

Organizations can further increase the accuracy of the DLP profile by enabling context analysis. This feature requires certain proximity keywords to exist within approximately 1000 characters of a match. For example, the string “123-45-6789” will only count as a detection if it is in proximity to keywords such as “ssn”. This contextual requirement bolsters the accuracy of the detection process.

The DLP service seamlessly integrates with both Cloudflare’s SWG and API-driven CASB services. In the case of the API CASB, DLP profiles are selected for scanning each integration with each SaaS application. This customization allows tailored detection criteria based on the type of data you wish to secure within each application.

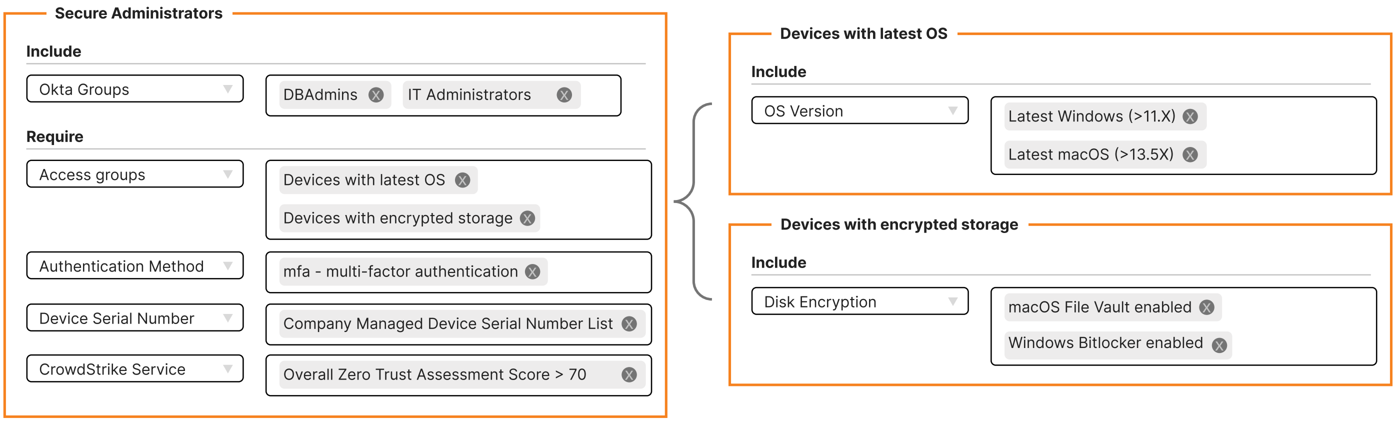
For the SWG service, DLP profiles can be included into any policy to detect the existence of sensitive data in any request passing through the gateway. The most common action associated with this detection is to block the request, providing a robust layer of security.

Access Groups

Access Groups are a powerful tool in the ZTNA service for aggregating users or devices into a unified entity that can be referenced within a policy. Within Cloudflare, multiple pieces of information can be combined into a single Access Group, efficiently reusing data across multiple policies while maintaining it in one centralized location.

Consider an Access Group designed to manage access to critical server infrastructure. The same Access Group can be used in a device agent policy that prevents administrators from disabling their connection to Cloudflare. This approach streamlines policy management and ensures consistency across various policy implementations.

Below is a diagram featuring an Access Group named “Secure Administrators,” which uses a range of attributes to define the characteristics of secure administrators. The diagram shows the addition of two other Access Groups within “Secure Administrators”. The groups include devices running on either the latest Windows or macOS, along with the requirement that the device must have either File Vault or Bitlocker enabled.



Secure Administrators

Include

- Okta Groups: DBAdmins ⓧ, IT Administrators ⓧ

Require

- Access groups: Devices with latest OS ⓧ, Devices with encrypted storage ⓧ
- Authentication Method: mfa - multi-factor authentication ⓧ
- Device Serial Number: Company Managed Device Serial Number List ⓧ
- CrowdStrike Service: Overall Zero Trust Assessment Score > 70 ⓧ

Devices with latest OS

Include

- OS Version: Latest Windows (>11.X) ⓧ, Latest macOS (>13.5X) ⓧ

Devices with encrypted storage

Include

- Disk Encryption: macOS File Vault enabled ⓧ, Windows Bitlocker enabled ⓧ

Consistent with Cloudflare’s overarching flexibility, Access Groups can be created, updated, and applied to policies through Cloudflare API or using Terraform. This allows a seamless integration with existing IT systems and processes, ensuring a cohesive approach to access management.

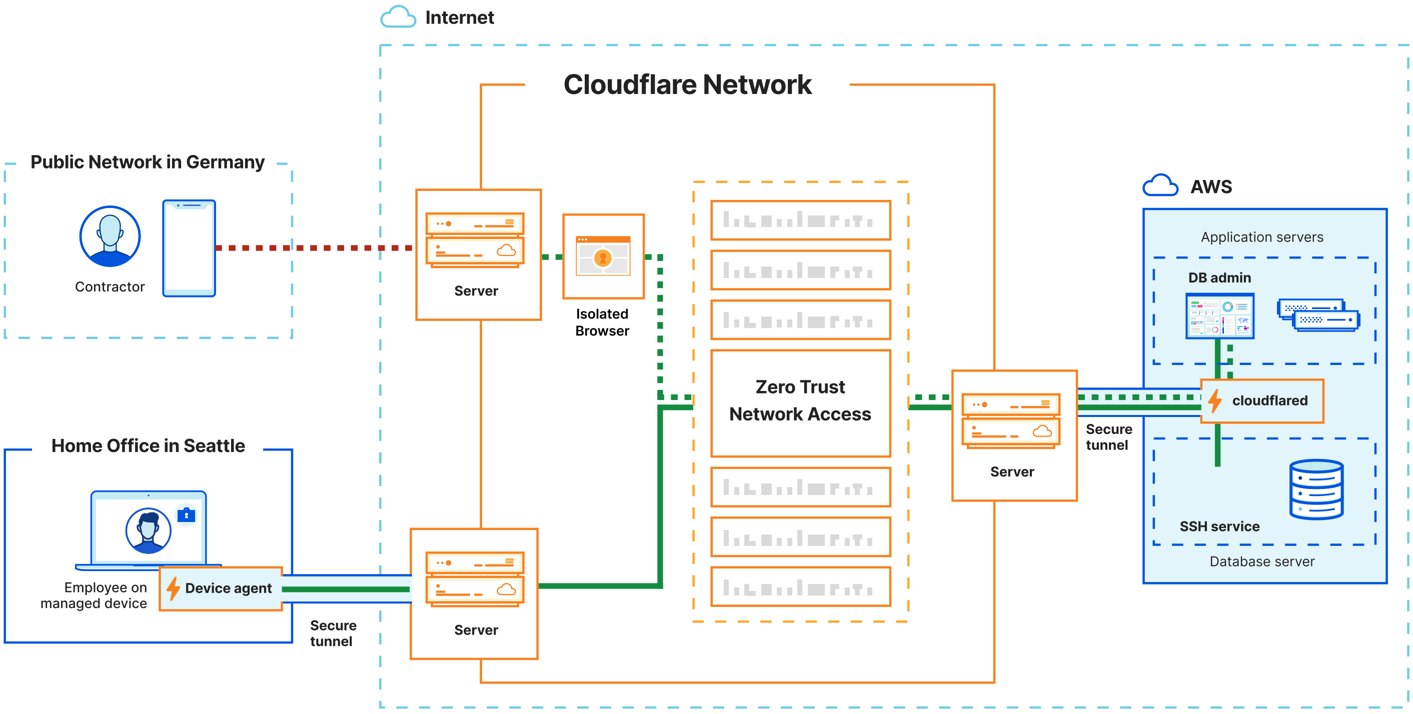
Now that we have a solid understanding of all the components available, let’s zoom in and take a look at some common use cases and how they are configured. Keep in mind that Cloudflare’s policy engines are incredibly powerful and flexible, so these examples are just a glimpse into the capabilities of Cloudflare’s SASE platform.

Example use cases

Secure access to self hosted apps and services

One common driver for moving to a SASE architecture is replacing existing VPN connectivity with a more flexible and secure solution. Cloudflare One SASE architecture enables high performance and secure access to self hosted applications from anywhere in the world. However, the next step entails defining the policies that control access to resources.

In this example, consider two services: a database administration application (pgadmin for example) and an SSH daemon running on the database server. The diagram below illustrates the flow of traffic and highlights the ZTNA service. It's important to note that all other services still retain the ability to inspect the request. For instance, the contractor using her personal cell phone in Germany should only have access to the db admin tool, while the employee on a managed device can access both the db admin tool and SSH into the database server.



The policies that enable access rely on two Access Groups.

- Contractors
    - Users who authenticate through Okta and are part of the Okta group labeled "Contractors"
    - Authentication requires the use of a hardware token
- Database and IT administrators
    - Users who authenticate through Okta and are in the Okta groups "IT administrators" or "Database administrators"
    - Authentication requires the use of a hardware token
    - Users should be on a device with a serial number in the "Managed Devices" list

Both of these groups are then used in two different access policies.

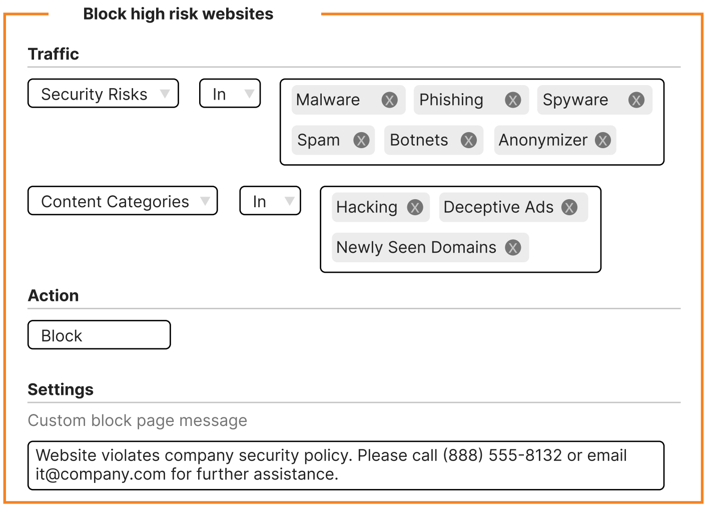
- Database administration tool access
    - Database and IT admins are allowed access
    - Members of the "Contractor" access group are allowed access, but each authenticated session requires the user to complete a justification request
    - The admin tool is rendered in an isolated browser on Cloudflare's Edge network and file downloads are disabled
- Database server SSH access
    - "Database and IT administrators" group is allowed access
    - Their device must pass a Crowdstrike risk score of at least 80
    - Access must come from a device that is running our device agent and is connected to Cloudflare

These policies show that contractors are only allowed access to the database administration tool and do not have SSH access to the server. IT and database administrators can access the SSH service only when their devices are securely connected to Cloudflare via the device agent. Every element of the access groups and policies is evaluated for every login, so an IT administrator using a compromised laptop or a contractor unable to authenticate with a hardware token will be denied access.

Both user groups will connect to Cloudflare through the closest and fastest access point of Cloudflare's globally distributed network, resulting in a high quality experience for all users no matter where they are.

Threat defense for distributed offices and remote workers

Another reason for using a SASE solution is to apply company security policies consistently across all users (whether they are employees or contractors) in the organization, regardless of where they work. The Cloudflare One SASE architecture shows that all user traffic, whether routed directly on the device or through the connected network, will go through Cloudflare. Cloudflare's SWG then handles inspection of this traffic. Depending on the connection method, policies can be applied either to the HTTP or DNS request. For example:



Block high risk websites

Traffic

Security Risks | In | Malware, Phishing, Spyware, Spam, Botnets, Anonymizer

Content Categories | In | Hacking, Deceptive Ads, Newly Seen Domains

Action

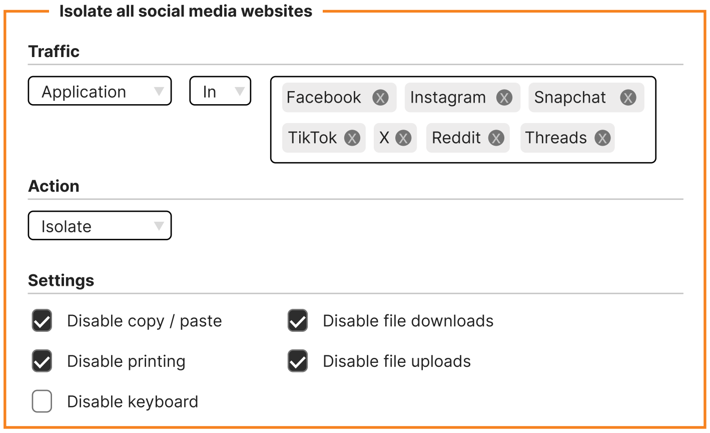
Block

Settings

Custom block page message

Website violates company security policy. Please call (888) 555-8132 or email it@company.com for further assistance.

This can then be applied to secure and protect all users in one simple policy. Cloudflare can write another policy allowing access to social media websites while isolating all sessions in a remote browser hosted on Cloudflare's network.



Isolate all social media websites

Traffic

Application | In | Facebook, Instagram, Snapchat, TikTok, X, Reddit, Threads

Action

Isolate

Settings

☑ Disable copy / paste

☑ Disable printing

☐ Disable keyboard

☑ Disable file downloads

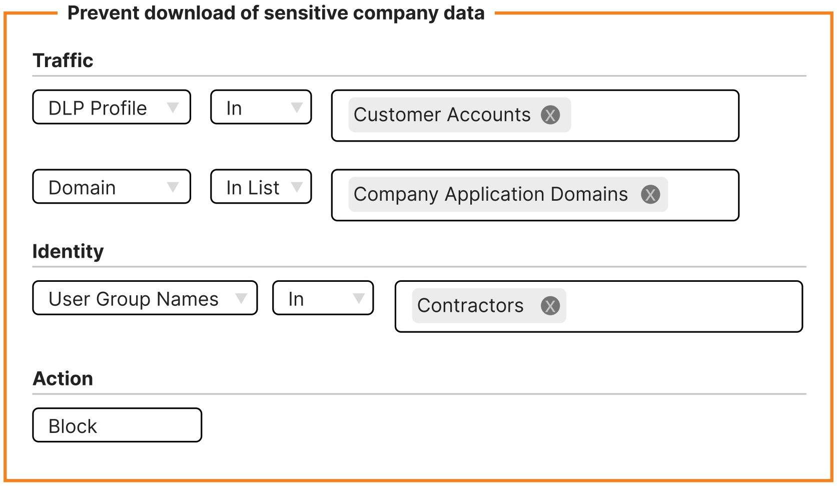
☑ Disable file uploads

With this setup, every request to a social media website ensures the following security measures:

- Any content on the social media website that contains harmful code is prevented from executing on the local device
- External users are restricted from downloading content from the site that could potentially be infected with malware or spyware

Data protection for regulatory compliance

Because Cloudflare One has visibility over every network request, Cloudflare can create policies that apply to the data in the request. This means that the DLP services can be used to detect the download of content from an application and block it for specific user demographics. Let’s look at the following policy.



Prevent download of sensitive company data

Traffic

DLP Profile	In	Customer Accounts
Domain	In List	Company Application Domains

Identity

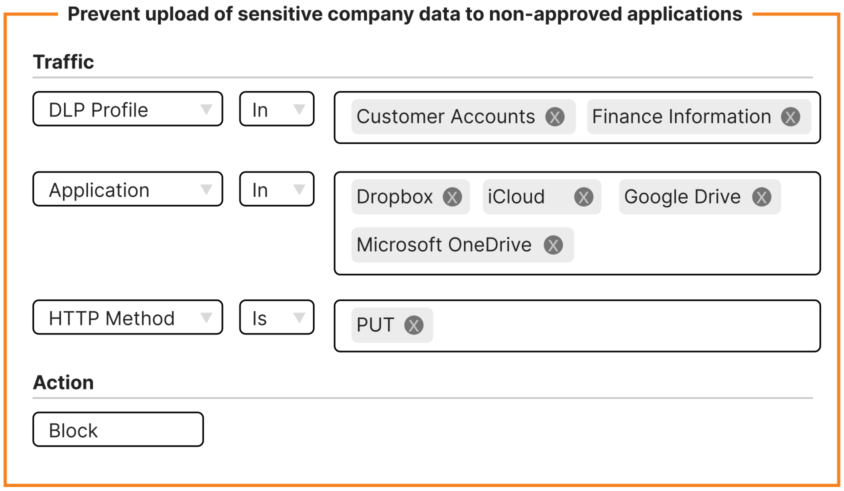
User Group Names	In	Contractors

Action

Block

This policy would prevent contractors from downloading a file containing customer accounts information. Furthermore, Cloudflare can configure an additional policy to block the same download if the user’s device does not meet specific security posture requirements. This ensures the consistent enforcement of a common rule: no sensitive customer data can be downloaded onto a device that does not meet the required security standards.

DLP policies can also be applied in the other direction, ensuring that company sensitive documents are not uploaded to non approved cloud storage or social media.



Prevent upload of sensitive company data to non-approved applications

Traffic

DLP Profile	In	Customer Accounts Finance Information
Application	In	Dropbox iCloud Google Drive Microsoft OneDrive
HTTP Method	Is	PUT

Action

Block

Visibility across the deployment

At this point in the SASE journey, users have re-architectured the IT network and security infrastructure to fully leverage all the capabilities of the Cloudflare One SASE platform. A critical element in long term deployment involves establishing complete visibility into the organization and the ability to diagnose and quickly resolve issues.

For quick analysis, Cloudflare provides built-in dashboards and analytics that offers a daily overview of the deployment's operational status. As traffic flows through Cloudflare, the dashboard will alert internal users to the most frequently used SaaS applications, enabling quick actions if any unauthorized applications are accessed by external users. Moreover, all logging information from all Cloudflare One services is easily accessible and searchable from the administrator's dashboard. This makes it efficient to filter for specific blocked requests, with each log containing useful information such as the user's identity, device information, and the specific rule that triggered the block. This can be very handy in the early stages of deployment where rules can often need tweaking.

However, many organizations rely on existing dedicated tools to manage long term visibility over the performance of their infrastructure. To support this, Cloudflare allows the export of all logging information into such tools. Every aspect of Cloudflare One is logged and can be exported. Cloudflare offers built in integrations for continuous transmission of small data batches to a variety of platforms, including AWS, Google Cloud Storage, SumoLogic, Azure, Splunk, Datadog, and any S3 compatible service. This flexibility allows organizations to selectively choose which fields to control the type and volume of data to incorporate into existing tools.

On top of logs which are related to traffic and policies, Cloudflare also audits management activity. All administrative actions and changes to Cloudflare Tunnels are logged. This allows for change management auditing and, like all other logs, can be exported into other tools as part of a wider change management monitoring solution.

Digital Experience Monitoring

Cloudflare has deep insight into the performance of the Internet and connected networks and devices. This knowledge empowers IT administrators with visibility into minute-by-minute experiences of their end-users, enabling swift resolution of issues that impact productivity.

The Digital Experience Monitoring (DEM) service enables IT to run constant tests against user devices to determine the quality of the connection to company resources. The results of these tests are available on the Cloudflare One dashboard, enabling IT administrators to review and identify root causes when a specific user encounters difficulties accessing an application. These issues could stem from the user's local ISP or a specific underperforming SaaS service provider. This data is invaluable in helping administrators in diagnosing and addressing poor user experiences, leading to faster issue resolution.

The dashboard shows a comprehensive summary of the entire device fleet, displaying real-time and historical connectivity metrics for all organization devices. IT admins can then drill down into specific devices for further analysis.

Summary

Having acquired a comprehensive understanding of Cloudflare's SASE platform, you are now well-equipped to integrate it with existing infrastructure. This system efficiently secures access to applications for both employees and external users, starting from the initial request on the device and extending across every network to the application, regardless of its location. This powerful new model for securing networks, applications, devices, and users is built on the massive Cloudflare network and easily managed through an intuitive management interface.

It's worth noting that many of the capabilities described in this document can be used for free, without any time constraints, for up to 50 users. Simply sign up for an account and head to the Zero Trust section. While this document has provided an overview of the platform as a whole, for those interested in delving deeper into specific areas, we recommend exploring the following resources.

Topic	Content
Cloudflare Tunnels	Understanding Cloudflare Tunnel Open source repository for cloudflared
WAN as a Service	Cloudflare Magic WAN documentation
Secure Web Gateway	How to build Gateway policies
Zero Trust Network Access	How to build Access policies
Remote Browser Isolation	Understanding browser isolation
API-Driven CASB	Scanning SaaS applications
Cloud Email Security	Understanding Cloudflare Email Security
Replacing your VPN	Using Cloudflare to replace your VPN

If you would like to discuss your SASE requirements in greater detail and connect with one of our architects, please visit cloudflare.com/cloudflare-one/ and request a consultation.

References

1. https://www.cloudflare.com/connectivity-cloud/
2. https://www.cloudflare.com/network/
3. https://bgp.he.net/report/exchanges#_participants
4. https://www.cloudflare.com/what-is-cloudflare/
5. https://www.youtube.com/watch?v=XHvmX3FhTwU
6. https://cfl.re/SASE-SSE-platform-brief
7. https://cfl.re/internet-native-transformation-wp
8. https://blog.cloudflare.com/zero-trust-sase-and-sse-foundational-concepts-for-your-next-generation-network/
9. https://developers.cloudflare.com/reference-architecture/
10. https://www.cloudflare.com/learning/access-management/security-service-edge-sse/
11. https://www.cloudflare.com/learning/access-management/what-is-ztna/
12. https://www.cloudflare.com/learning/access-management/what-is-a-secure-web-gateway/
13. https://www.cloudflare.com/cloudflare-one/
14. https://www.cloudflare.com/learning/security/api/what-is-an-api-gateway/
15. https://www.cloudflare.com/learning/ddos/glossary/web-application-firewall-waf/
16. https://www.cloudflare.com/learning/cdn/what-is-a-cdn/
17. https://www.cloudflare.com/learning/ddos/ddos-mitigation/
18. https://www.cloudflare.com/learning/cdn/glossary/anycast-network/
19. https://blog.cloudflare.com/meet-traffic-manager/
20. https://www.cloudflare.com/learning/cloud/what-is-iaas/
21. https://www.cloudflare.com/learning/access-management/what-is-an-identity-provider/
22. https://www.cloudflare.com/learning/security/glossary/what-is-endpoint/
23. https://www.equinix.com/
24. https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/get-started/
25. https://hub.docker.com/r/cloudflare/cloudflared
26. https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/routing-to-tunnel/lb/
27. https://developers.cloudflare.com/cloudflare-one/tutorials/many-cfd-one-tunnel/
28. https://www.cloudflare.com/learning/access-management/what-is-shadow-it/
29. https://developers.cloudflare.com/cloudflare-one/policies/gateway/egress-policies/dedicated-egress-ips/
30. https://www.cloudflare.com/learning/access-management/what-is-a-casb/
31. https://developers.cloudflare.com/cloudflare-one/applications/scan-apps/casb-integrations/
32. https://www.cloudflare.com/learning/access-management/what-is-dlp/
33. https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/private-net/
34. https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/private-net/warp-connector/
35. https://en.wikipedia.org/wiki/Carrier-grade_NAT
36. https://www.cloudflare.com/learning/network-layer/what-is-ipsec/
37. https://www.cloudflare.com/learning/network-layer/what-is-gre-tunneling/
38. https://www.cloudflare.com/network-services/products/magic-wan/
39. https://www.cloudflare.com/network-services/products/network-interconnect/
40. https://blog.cloudflare.com/anycast-ipsec/

41. https://blog.cloudflare.com/magic-wan-connector/

42. https://developers.cloudflare.com/magic-wan/third-party/

43. https://developers.cloudflare.com/magic-wan/get-started/configure-tunnels/

44. https://developers.cloudflare.com/network-interconnect/about/interconnect-types/

45. https://www.peeringdb.com/net/4224

46. https://www.cloudflare.com/network-interconnect-partnerships/

47. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp

48. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/configure-warp/warp-modes/

49. https://www.cloudflare.com/learning/access-management/what-is-dns-filtering/

50. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/configure-warp/route-traffic/split-tunnels/

51. https://datatracker.ietf.org/doc/html/rfc1918

52. https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/private-net/private-hostnames-ips/

53. https://www.cloudflare.com/partners/technology-partners/crowdstrike/endpoint-partners/

54. https://www.cloudflare.com/partners/technology-partners/sentinelone/

55. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/agentless/pac-files/

56. https://datatracker.ietf.org/doc/html/draft-ietf-wrec-wpad-01

57. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/user-side-certificates/

58. https://www.cloudflare.com/learning/access-management/what-is-browser-isolation/

59. https://developers.cloudflare.com/cloudflare-one/policies/browser-isolation/setup/

60. https://www.cloudflare.com/learning/email-security/what-is-email-security/

61. https://developers.cloudflare.com/cloudflare-one/policies/gateway/initial-setup/dns/#connect-dns-locations

62. https://www.cloudflare.com/learning/dns/dns-over-tls/

63. https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/agentless/dns/dns-over-https/#filter-doh-requests-by-user

64. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/generic-saml/

65. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/generic-oidc/

66. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/okta/

67. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/azuread/

68. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/gsuite/

69. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/facebook-login/

70. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/github/

71. https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/linkedin/

72. https://developers.cloudflare.com/cloudflare-one/identity/service-tokens/

73. https://www.cloudflare.com/learning/access-management/what-is-mutual-tls/

74. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/application-check/

75. https://developers.cloudflare.com/cloudflare-one/identity/devices/access-integrations/tanium/

76. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/firewall/

77. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/disk-encryption/

78. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/domain-joined/

79. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/os-version/

80. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/device-uuid/

81. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/corp-device/

82. https://developers.cloudflare.com/cloudflare-one/identity/devices/service-providers/microsoft/

83. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/carbon-black/

84. https://developers.cloudflare.com/cloudflare-one/identity/devices/service-providers/crowdstrike/

85. https://developers.cloudflare.com/cloudflare-one/identity/devices/warp-client-checks/sentinel-one/

86. https://www.cloudflare.com/learning/email-security/business-email-compromise-bec/

87. https://www.cloudflare.com/learning/email-security/what-is-vendor-email-compromise/

88. https://developers.cloudflare.com/email-security/deployment/inline/

89. https://developers.cloudflare.com/email-security/deployment/api/

90. https://www.cloudflare.com/learning/email-security/what-is-smtp/

91. https://developers.cloudflare.com/api/

92. https://registry.terraform.io/providers/cloudflare/cloudflare/latest/docshttps://github.com/cloudflare

93. https://developers.cloudflare.com/cloudflare-one/policies/gateway/domain-categories/

94. https://radar.cloudflare.com/

95. https://developers.cloudflare.com/cloudflare-one/policies/gateway/lists/

96. https://www.pgadmin.org/

97. https://dash.cloudflare.com/sign-up

98. https://one.dash.cloudflare.com/

99. https://github.com/cloudflare/cloudflared

100. https://developers.cloudflare.com/magic-wan/

101. https://developers.cloudflare.com/cloudflare-one/policies/gateway/

102. https://developers.cloudflare.com/cloudflare-one/policies/browser-isolation/

103. https://developers.cloudflare.com/cloudflare-one/applications/scan-apps/

104. https://developers.cloudflare.com/email-security/

105. https://developers.cloudflare.com/learning-paths/replace-vpn/

CLOUDFLARE

1 888 99 FLARE | enterprise@cloudflare.com | Cloudflare.com