

**[ECS] Zero Trust POC**

orange-gradient.png

[**Overview**](#_2dui7zm542cu) **2**

[**POC Summary**](#_tta9xxt83c6c) **4**

[**How to use this guide**](#_bd2skj8qctn9) **4**

[**High Level Solution Overview**](#_uqodn9788y7a) **5**

[Cloudflare Zero Trust](#_wlu2f6pdgcsx) 5

[**Pre-POC Checklist**](#_bk7xo9tbit62) **6**

[**POC Checklist & Success Criteria**](#_5ifl9hhllkom) **8**

[**POC ISSUES**](#_q7gswb9h9v76) **10**

[**APPENDIX: Test Configurations**](#_b27xlcvvailk) **11**

[Basic Account Set ups](#_ugyv987v50en) 11

[Set the Teams Domain](#_tctwd52l7al) 11

[Install Cloudflare Certificate on Devices](#_grw3jp21hvn) 11

[Download WARP on Devices](#_z49v0sqxywul) 11

[Enable Proxy](#_t29gn78vc9g5) 12

[Configure the IdP login methods](#_vuv05phs1khh) 12

[(optionally) Configure One-time PIN to test multi-auth providers <link>](#_hg7stp2bywpe) 13

[(optionally) Configure Access App Launcher](#_6j1innivnejb) 13

[Configure Device Enrollment Policy](#_g3azxd6copli) 14

[Configure Device Posture Attributes <link>](#_3yggbwronnu2) 14

[(optionally) Create Lists to quickly create rules that match and take action against several items at once <link>](#_mozngt8au5v6) 15

[(optionally) Create Groups so that a set of rules can be configured at once and quickly applied across many Access Applications](#_fpsujp4eyr7f) 15

[Create Access Group <link>](#_osohp48wr9bw) 15

[Create Device list](#_m4j43jah6yip) 15

[Add Device Posture Check](#_2x2xr1a3ta4s) 16

[Configure Cloudflare Access](#_9y2vp2sl05d7) 19

[Access: Self-hosted App](#_h8ibhf90wz3x) 19

[Access: SaaS Applications](#_dljpn36usc2u) 20

[Access: Non-HTTP Applications](#_cnhd39lu3v4b) 21

[Access: Private Networks/Apps](#_f40i39i9hmu4) 22

[Configure Cloudflare Tunnel](#_mdmal3nkqqc) 26

[Installing Cloudflared (daemon) on internal servers](#_dsth3a2xim72) 26

[Configuring Tunnel for Applications/Networks](#_84g4wzw8v9nu) 26

[Routing to an application](#_gl0slloupzqw) 27

[HTTPS applications:](#_p70pwca1w1f6) 27

[Tunnel to another instance on another subnet](#_yzf6iju144kb) 27

[Routing to a network](#_1kuryjpp6f8t) 27

[Running Cloudflared as a Service <link>](#_7dwhw6l1z4d7) 28

[Setting up Tunnel for an entire Private Network](#_vq291j1mdz1r) 28

[Running Cloudflared as a Service <link>](#_om6e8cmrldp5) 29

[(optional) Resolve Private IP and Hostnames <link>](#_dum4l4xdxwc7) 30

[(optional) Deploy Cloudflared Replicas](#_jxyo7jpfycdx) 31

[Configure Cloudflare Gateway](#_iwwhuk32uarx) 33

[Configuring a Default Location](#_z0fo6pqf2tkb) 33

[Configuring Gateway Policies](#_y55hct8s8amz) 33

[Configuring DNS Filtering Policies](#_dyd7a0ud89de) 33

[Configuring HTTP Filtering Policies](#_wszg11mshc4e) 34

[Configuring Network Filtering Policies](#_x0kzf5fb2aoe) 35

[Configure Cloudflare Remote Browsing](#_k882j15wp6q0) 38

[View Analytics & Configure Logpush](#_gwwyfxjckyu3) 39

[Analytics](#_7ljfgb9l6hwu) 39

[User Logs](#_3ibkuxct1si7) 39

[Admin Logs](#_tggotw6eyee0) 40

[Access Logs](#_tceicokxbr85) 40

[Gateway Logs](#_behkskj96w20) 40

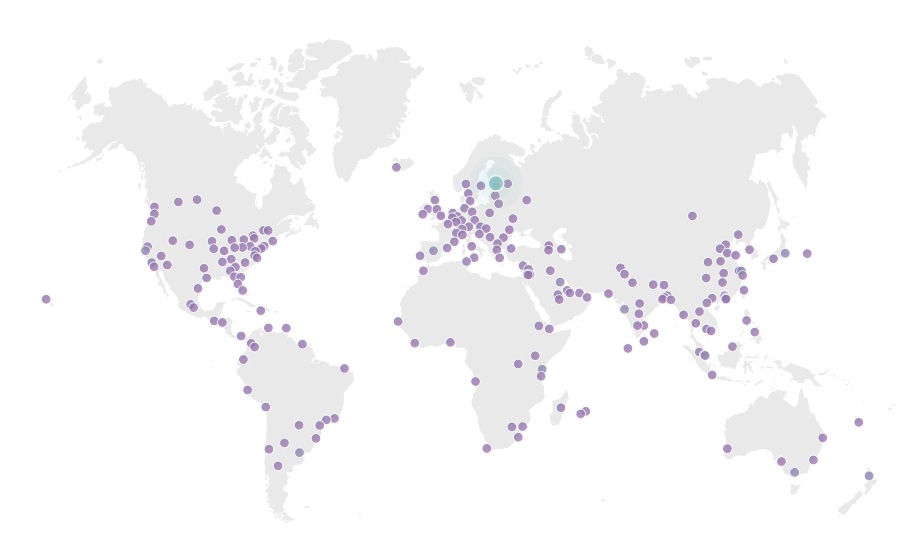
[Configure Logpush](#_u4ljetu58vbr) 41

[**Success Criteria**](#_flvufwxy9rnh) **44**

[**Timeline**](#_17dp8vu) **44**

[Proof of Concept Terms](#_yo9suf1bk6a) 44

# Overview



Cloudflare is on a mission to help build a better Internet. Cloudflare is one of the world’s largest networks. Today, businesses, non-profits, bloggers, and anyone with an Internet presence boast faster, more secure websites and apps thanks to Cloudflare.

Approximately 25 million Internet properties are on Cloudflare, and our network is growing by tens of thousands each day. Cloudflare powers Internet requests for ~17% of the Fortune 1,000 and serves 25 million HTTP requests per second on average.

We provide security and performance for approximately 25 million Internet properties and offer great functionality such as SSL and content distribution to every website on our network. Our services run silently in the background, keeping many of the websites and services you depend on up and running. Cloudflare offers a [free DNS service called 1.1.1.1](https://www.cloudflare.com/learning/dns/what-is-1.1.1.1/) that you can use on any device. Cloudflare’s 1.1.1.1 protects your data from being analysed or used for targeting you with ads.

Above all, we are mission-driven. That’s why we protect organizations working on behalf of the arts, human rights, civil society, or democracy with [Project Galileo](https://www.cloudflare.com/galileo/), giving them Cloudflare’s highest level of protection for free.

# 

# POC Summary

| **POC Start Date** | 25/04/22 |
| --- | --- |
| **POC Finish Date** | 21/02/22 |
| **Cloudflare Product(s)** | Cloudflare Zero Trust |
| **Cloudflare Entitlements** | **Cloudflare Account ID:**  **johndoe@example.com**  **Entitlements Configured:**   * Cloudflare for Teams Ent - 5 user license * Cloudflare Enterprise - 1 zone |
| **POC Scope** | e.g.  **Access**  Customer has programmers (WFH) trying to connect to on-prem and AWS services. They will be accessing http(s) and non-http apps. To provide architectural diagram on apps to connect, where they sit in network and where they want to install Cloudflared connector to connect networks to Cloudflare.  **Gateway & RBI**  Customer wants to test gateway and RBI for internet-bound traffic |
| **Onboarding Decisions** | **Zone:**  e.g. CNAME set up for zerotrust.example.com |
| **Technical Contact (Customer)** |  |
| **Technical Contact (Cloudflare)** | Claire Lim <[clairelim@cloudflare.com](mailto:clairelim@cloudflare.com)> |
| **Escalation Contact (Cloudflare)** |  |

# 

# High Level Solution Overview

## Cloudflare Zero Trust

# 

Cloudflare Zero Trust replaces legacy security perimeters with our global edge, making the Internet faster and safer for teams around the world.

Zero Trust access for all of your applications: [**Cloudflare Access**](https://www.cloudflare.com/en-gb/teams/access/)

* Authenticate users on our global edge network
* Onboard third-party users seamlessly
* Log every event and request

A Secure Web Gateway to protect users and devices: [**Cloudflare Gateway**](https://www.cloudflare.com/teams/gateway/)

* Enforce your company’s Acceptable Use Policy (AUP)
* Block risky sites with custom blocklists and built-in threat intel
* Enhance visibility and protection into SaaS applications

A fast and reliable solution for remote browsing: [**Cloudflare Zero Trust Remote Browsing**](https://www.cloudflare.com/teams/browser-isolation/)

* Execute all browser code in the cloud
* Mitigate the impact of attacks
* Seamless, lightning-fast end user experience

# Pre-POC Checklist

**Note:** we will use this document as a working file during the POC.

Any issues/questions during POC will be documented under [POC Issues](#_q7gswb9h9v76) for future reference and your test cases will be documented here. Should you have any questions, feel free to add comments to highlight the document.

| **Task** | **Status** |
| --- | --- |
| [Open up the Document Outline](https://support.google.com/docs/answer/6367684?hl=en&co=GENIE.Platform%3DDesktop) by the side for easy reference |  |
| Read through [High level solution overview](#_wsczeivde4cs) to understand our Cloudflare offerings |  |
| Define test cases in [POC Checklist](#_5ifl9hhllkom) & [Success Criteria](#_flvufwxy9rnh)   1. **Zero Trust Network Access and Cloudflare Tunnel**  * There are 2 ways to connect to corporate resources via tunnel      1. **Via Public Hostname**      * Clientless method of accessing the applications via public hostname (resolved by public DNS) * Even though the hostname is public, if you add an access app on the public hostname, you will be automatically redirected to the ZTNA login page. Only after authenticating yourself and satisfying all ZTNA policies to access that app, then you will be able to get to the service  1. **Via Private IP/hostname**      * From user to Cloudflare, we can use our Client (WARP) to connect the device to Cloudflare via a Wireguard tunnel which is a new-generation VPN protocol that is a lot more performant to communicate with Cloudflare securely. * Each connection from the user to Cloudflare and from Cloudflare to the resource is protected via a secure tunnel to ensure there’s only 1 point of contact to the origin. * Cloudflare Access can be used to connect to private resources – define what services you want to connect to, where they sit in your network (private IP) and where you are installing Cloudflare daemon (Cloudflared). |  |
| Origin is working without Cloudflare |  |
| Origin firewall is not blocking [Cloudflare IPs](https://www.cloudflare.com/ips/)  cloudflared (our daemon to connect to Cloudflare edge and run tunnels) will connect to [Cloudflare edge port 7844](https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/configuration/ports-and-ips/). |  |
| Set aside a non-production zone for testing |  |
| Sign up for Cloudflare account at dash.cloudflare.com and provide the account that you have signed up with e.g. johndoe.example.com to the Cloudflare team to provide entitlements |  |
| On POC Start date | |
| Once entitlements have been provided (on start of POC), check account entitlements are in place (e.g. Zone is on enterprise plan, ZT ent) |  |
| Add your zone to cloudflare dashboard  If using [CNAME setup](https://support.cloudflare.com/hc/en-us/articles/360020348832-Understanding-a-CNAME-Setup), TXT record has been verified in the registrar. |  |
| [SSL has been configured](https://developers.cloudflare.com/ssl/get-started) for the domains and the Cloudflare SSL configuration is set to the correct mode (minimally Full) |  |

# POC Checklist

The Proof of Concept will be considered a success when the following criteria have been met:

## 

| **What?** | **Test case** | **Who** | **Status** |
| --- | --- | --- | --- |
| 1. Basic Account Set ups | 1. [Set up the Team Domain](#_tctwd52l7al) and [Turn on proxy](#_t29gn78vc9g5) | Partner |  |
| 1. [Configure Identity Providers](#_vuv05phs1khh) and [OTP](#_hg7stp2bywpe) | Customer |  |
| 1. [(optional) Configure App Launcher](#_6j1innivnejb) | Partner |  |
| 1. Prepare End Devices    1. [Install WARP](#_z49v0sqxywul)    2. [Install certificates](#_26p79snei3i2)    3. [Configure Device enrollment policies](#_g3azxd6copli)    4. (optional) [Configure Device posture attributes](#_3yggbwronnu2) | Customer  (a-b)  Partner  (c-d) |  |
| 1. (optional) Create [Lists](#_mozngt8au5v6) and [Groups](#_fpsujp4eyr7f) | Partner |  |
| 2. Setup Cloudflare tunnel | 1. [Create the tunnel and install the daemon](#_mdmal3nkqqc) | Customer |  |
| 1. Create tunnel to route to    1. [Public Hostname](#_gl0slloupzqw) - using option 1 (tunnel to public hostname)    2. [Enter network](#_xx0ixguxsjic) - using option 2 (WARP-to-tunnel)   [Client] – warp – [CF] – cloudflared tunnel – [origin]   * *WARP-to-tunnel connection requires WARP to be enrolled and logged in on the end-user device (see section 3 on Posture check to ensure set ups are completed. After which you should be able to resolve private IP on the client with WARP running)* | Partner |  |
| 3. Application Access Policy and Access Groups | 1. [Create Access groups](#_osohp48wr9bw) for different users | Partner |  |
| 1. Create an Access Application for the app login page    1. For [self-hosted apps](#_h8ibhf90wz3x) with public hostname       1. note when setting up tunnel for an application, you can route the tunnel to a public hostname which allows you to set up access policies on this hostname       2. For [non-http self-hosted apps](#_cnhd39lu3v4b) like SSH, you can also achieve the same via the public hostname    2. For [Private network/apps](#_f40i39i9hmu4) in the private subnet you have routed, you can set up the access application for the destination private IP | Partner |  |
| 1. Create different policy settings for user working from Corporate IP and user working from home using user groups | Partner |  |
| 1. Ensure all users can access the application with the rules configured | Partner |  |
| 1. [Check the logs](#_tceicokxbr85) to see the source IP | Partner |  |
| 4. Configure Gateway | 1. Test Gateway Policies using    1. [Location](#_z0fo6pqf2tkb) - connects end devices from a source IP (Office network) to Cloudflare for DOH    2. [WARP](#_y55hct8s8amz) - connects end devices with WARP Client installed to Cloudflare for DNS, HTTP and Network filtering | Partner |  |
| 5. Set up Remote browser isolation | 1. [Create policy to isolate certain sites](#_k882j15wp6q0) | Partner |  |

# 

# POC ISSUES

| **Date** | **Issues/Discussion** | **Next Steps/Answer** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# APPENDIX: Test Configurations

## Basic Account Set ups

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Set the Teams Domain  * 1. Go to the Teams Dashboard (dash.teams.cloudflare.com)   2. On the onboarding screen, choose the team name. Your teams domain will be <team\_name>.cloudflareaccess.com. You can verify the teams domain under Settings > General > Team Domain  Enable Proxy  * 1. Navigate to Settings > Network and enable the Proxy setting. This will allow you to start routing your HTTP traffic to Gateway.    Install Cloudflare Certificate on Devices  * 1. Download the right certificate for your device and follow the installation instructions [here](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/install-cloudflare-cert)  Download WARP on Devices  * 1. WARP is our client that connects corporate desktop to Cloudflare gateway for advanced filtering and securely & privately connect remote user through closest Cloudflare Data centers. We use boringtun (our own wireguard implementation) to encrypt traffic from device to our edge to ensure no snooping is happening   2. Download WARP on your device [here](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/download-warp)   3. WARP can be deployed across multiple devices via [manual](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/deployment/manual-deployment) deployment or [managed](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/deployment/mdm-deployment) deployment      1. Note: for Managed Deployment, there are some additional parameters we can configure WARP to do are [here](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/deployment/mdm-deployment/parameters)   4. If your organization uses a firewall or other policies to restrict Internet traffic, you may need to [make a few changes](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/deployment/firewall) to allow WARP to connect.   5. Log in to WARP on your device by entering your teams domain.   Turn on WARP    Open **WARP Settings > Account > Login with Teams**    Provide the teams domain name and give access   Configure the IdP login methods  * 1. Go to Teams Dashboard > Settings > Authentication > Login Methods > Add new   2. Select the identity provider to add. You will be prompted for the App ID and Client Secret to authenticate this. To get this ID and client secret, follow the steps on the right. You can follow the set up guides for Active Directory (ADFS) via SAML [here](https://developers.cloudflare.com/cloudflare-one/identity/idp-integration/adfs/).      * 1. Once you are done, the login method has been added and you can test the login method below by running “Test”  **(optionally) Configure One-time PIN to test multi-auth providers** [<link>](https://developers.cloudflare.com/cloudflare-one/identity/one-time-pin)(optionally) Configure Access App Launcher  * 1. Head to Teams Dashboard > Settings > Authentication > App Launcher and Select Manage      * 1. Create a rule for App Launcher      * 1. Save the App Launcher rules  Configure Device Enrollment Policy  * 1. Go over to Teams Dashboard > Devices > Connect a Device   2. Create a device enrollment policy      * 1. Install WARP client on end devices (Macbook and iPhone). The WARP client can be installed from [here](https://developers.cloudflare.com/warp-client/setting-up)   2. Enter your team domain in the WARP client and install the Cloudflare root certificate [here](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp/install-cloudflare-cert)  **Configure Device Posture Attributes** <[link](https://developers.cloudflare.com/cloudflare-one/identity/devices)>  * 1. Under My Team > Devices > Device Posture, create device posture attributes you want to evaluate end user devices against.   2. Select Add to add the device posture attribute and follow the tutorial here to set this up for the specific device posture check.      1. Note: if you don’t have an existing endpoint device posture service, you can simply set this up for `WARP` and `Serial number • Corporate Devices` to test.    **(optionally) Create Lists to quickly create rules that match and take action against several items at once** <[link](https://developers.cloudflare.com/cloudflare-one/policies/lists)>   *e.g. create a list of corporate devices by adding corporate device serial numbers.* **(optionally) Create Groups so that a set of rules can be configured at once and quickly applied across many Access Applications** Create Access Group <[link](https://developers.cloudflare.com/cloudflare-one/identity/users/groups)>  * + 1. Under My Team > Groups, create a group for all Cloudflare employees with emails ending in @cloudflare.com so that only Cloudflare employees can access the app    Create Device list  * + 1. Under My Team > Lists, add a list for the serial number of your corporate devices. You can manually create a list or upload via CSV.      * + 1. Under My Team > Groups, create a group for corporate devices  Add Device Posture Check  * + 1. We have several options for device posture checks. You can access them under My Team > Devices > Device Posture      * + 1. Device posture check set ups are [here](https://developers.cloudflare.com/cloudflare-one/identity/devices)     2. This example will add WARP (Cloudflare’s client) as a device posture attribute     3. Under My Team > Groups, create a group for WARP to require WARP on the end device accessing the application     Then you can configure the Access to use these groups as policies to grant access to an application | |
| **Success/Test Criteria** | |
| * Cloudflare account successfully created and upgraded to Enterprise * Team Domain name set up * Login Methods set up * Download WARP and install it on end devices | |

## 

## 

## Configure Cloudflare Access

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Access: Self-hosted App [<link>](https://developers.cloudflare.com/cloudflare-one/applications/configure-apps/self-hosted-apps)   1. **Create an Application under Access to protect your page**    1. Navigate to Access > Applications > Create Application 2. **Select the self-hosted application to protect your application**      1. **Set up the Application Settings**     1. Give the application a name, define the session duration, application domain and identity providers to use.       1. Note: you can set multiple identity providers e.g. OTP and Azure or have just one IdP. You can also set different authentication methods per application.      * 1. Set the policy that defines rules by which traffic will be compared against. Only traffic matching rules defined will be allowed. You can follow the set up guide [here](https://developers.cloudflare.com/cloudflare-one/policies/zero-trust) and get some common configurations [here](https://developers.cloudflare.com/cloudflare-one/policies/zero-trust/common-configs).     *e.g. Create a rule to allow all employees access based on emails ending on (corp email)*    *e.g. Create require rule for country in PH - only if this criteria is met then the visitor can access the self-hosted app*   * 1. (optionally) Configure any advanced settings required <[link](https://developers.cloudflare.com/cloudflare-one/policies/zero-trust/cors)>  1. **Save to see the app created under the access tab.**     1. Once saved, when a user tries to access the application domain “example.com”, he will be redirected to the Cloudflare Access Login page where he can use the login methods to sign in. Only users who satisfy the policies set up will be allowed to pass through to access the application protected behind Cloudflare.  Access: SaaS Applications <[link](https://developers.cloudflare.com/cloudflare-one/applications/configure-apps/saas-apps)>   1. **Create an Application under Access to protect your SaaS Application**    1. Go to Access > Applications again and create a new SaaS application      1. **Set up the Application Settings**    1. Select the application you want to secure    2. Fill in the entity ID and Assertion Consumer Service (ACS) URL. You can follow the set up guide [here](https://developers.cloudflare.com/cloudflare-one/tutorials) for the SaaS application to secure.    3. Add the policy rules 2. **Save and set up the Cloudflare Access App in your SaaS application.**     1. Once you have saved the app, you will receive an SSO Endpoint, Access Entity ID or Issuer and Public Key.    2. Follow the steps [here](https://developers.cloudflare.com/cloudflare-one/tutorials) to parse this into the SaaS application you are securing.    3. After completing the steps, when users go to the SaaS application domain for your organisation, they will see an option to log in via the Cloudflare Access app. When they select this login method, they will be redirected to the Cloudflare Access Login page. If the user passes the login method and policies set, he will then be able to access the SaaS application  Access: Non-HTTP Applications <[link](https://developers.cloudflare.com/cloudflare-one/applications/non-http)**>**   1. [**Install Cloudflared on your server**](#_dsth3a2xim72) 2. **Create a tunnel for the SSH connection <**[**link**](https://developers.cloudflare.com/cloudflare-one/tutorials/ssh)**>**    1. Create a tunnel    2. Add the tunnel configuration to route to a hostname and connect to port 22    3. Route the tunnel to the hostname    4. Run the tunnel 3. **Create an Application under Access for your internal IP or Hostname**    1. Go to Access > Applications again and create a new Internal IP or Hostname application 4. **Set up the Application Settings**    1. Give the application a name, define the session duration, application domain and identity providers to use. *e.g. ssh.example.com*    2. Set the policy name and rules to set up the app for    3. For some non-HTTP apps like SSH/VNC, we can enable browser rendering under the Set Up step      1. **Save and make any additional configurations required for the app as specified in the tutorials** [**here**](https://developers.cloudflare.com/cloudflare-one/applications/non-http)  Access: Private Networks/Apps  1. **Create an Application under Access for your internal IP or Hostname**    1. Go to Access > Applications again and create a new Internal IP or Hostname application      1. **Set up the Application Settings**    1. Give the application a name, and provide the destination IP    2. Set the policy name and rules.       1. For different use cases, setups and policies required differ. Usually, best practice is to Block the subnet for all users and then specifically allow users who require the access to the subnet/internal application the access.       2. For example, in this case, I routed my tunnel to a private subnet 10.138.0.0/24. Within this subnet, I have many private applications such as 10.138.0.63 (Private app 1) and 10.138.0.129 (Private app 2 (wiki)). I want to provide granular zero trust access i.e. I want all employees to have access to only specific private applications in this subnet and I want all of these connections to run through the same tunnel. The following is how I can set this up          1. Create an application to allow specific users to access this wiki application 10.138.0.63          2. So what I will do is create the policies to allow certain people to access this application, and deny the rest.          3. Similarly, for the Private App 2, I will do the same for my app on 10.138.0.129          4. I will also create another policy for the entire subnet and I will block all users from accessing the full subnet for better security practice. This can be done under Gateway > Policies > Network          5. All of the Rules that I have set up will be shown under the gateway network policies      * + - 1. Now, even though my tunnel routes to the whole subnet, only users that need access to Private Apps will be able to access the applications.     1. There are many ways we can set up these policies e.g. we can enable access to an entire subnet for different user groups such as have 10.138.0.0/24 be open for all HR personnel and 10.148.0.0/24 be open for all Marketing personnel. | |
| **Success/Test Criteria** | |
| * Successful set up of Access for Self-hosted Apps * Successful set up of Access for SaaS Apps * Successful set up of Access for Non-HTTP apps | |

## 

## Configure Cloudflare Tunnel

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Configuring Tunnel for Applications OR Networks [<link>](https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/tunnel-guide/)   1. Under Cloudflare dashboard, go to the Access > Tunnels tab and select Create a tunnel 2. Give the tunnel a name 3. Follow the instructions to install a connector at your origin. This installs Cloudflare’s lightweight daemon (Cloudflared) which allows us to connect your origin to the 2 closest Cloudflare data centers with the fastest connections.     1. Install Cloudflared on an instance that is connected to all services you want to test so you can use 1 installation of Cloudflare daemon to test everything.    2. After installing this, you may be redirected to the dashboard. Select the domain/zone to authorise argo tunnel for and install      1. Now, you can decide how to route the tunnel to a service    Routing an application to a public hostname You can create a tunnel to connect a service from local host or any instance the VM that the daemon is installed in can connect to.  e.g. if the daemon is installed in 10.138.0.39 and it can connect to 10.148.0.129, you can create a tunnel for service http 10.148.0.129:8080 or ssh 10.148.0.129:22 HTTP applications: You can also add additional HTTP host headers and configure connection settings   HTTPS applications: If the application is a HTTPS application and you want to use your self-signed cert, Cloudflared must be instructed not to verify. Hence, you need to add the ingress rules ([noTLSverify](https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/configuration/configuration-file/ingress#notlsverify)) as specified below    You can also configure additional TLS and connection settings     * + 1. Note: The application will default to the Cloudflare settings of the hostname in your account that includes the Cloudflare Tunnel DNS record, including [cache rules](https://support.cloudflare.com/hc/en-us/articles/202775670-Customizing-Cloudflare-s-cache) and [firewall policies](https://developers.cloudflare.com/firewall/). You can change these settings for your hostname in Cloudflare's dashboard.     2. **(Optional) You can also use Load Balancer** with Cloudflared<[Link](https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/routing-to-tunnel/lb)>     3. Once the tunnel is up and running, you can check the subdomain/domain you have routed it to to test if it is working well     4. You then have the option to [configure zero trust network access policies for your application (Self-hosted/SaaS/HTTP/Private app/Subnet)](#_9y2vp2sl05d7) accordingly  Routing an entire network  * + 1. Configure the networks you want to route through the tunnel. These connections will use WARP (on the end device) to route traffic from 10.148.0.0/24 to Cloudflare (instead of resolving locally) and then from Cloudflare to origin, it will use the tunnel established by Cloudflared to reach the origin.     2. This network needs to be reachable by the VM the Cloudflared is being installed in.     3. To route private IP ranges through WARP, go to the dashboard. Under Settings > Network> Split Tunnels,     4. By default, Cloudflare WARP excludes certain IP ranges under the split tunnel entries.        1. If you are in exclude mode, delete the private IP range for <private ip subnet>        2. If you are in include mode, add the private IP range for <private ip subnet>*e.g. in exclude mode, if I want to route the tunnel to 10.0.0.0/8 subnet, I can remove the split tunnel*     5. Under Settings > Network > Firewall > Proxy, ensure HTTP filtering is enabled to allowed Cloudflare to proxy traffic of private IP ranges to tunnels     6. **Users can now connect over this private network by** [**enrolling their devices into the** WARP **agent**](https://developers.cloudflare.com/cloudflare-one/connections/connect-devices/warp) **in the same account**        1. Note: Gateway with WARP mode must be used        2. You can add the [identity provider](#_ugyv987v50en) (Under settings > authentication) and define the device enrollment requirements (Under settings > devices > device enrollment)     7. By routing the entire network, all apps in the subnet 10.148.0.0/24 can be reached via the same tunnel e.g. RDP, SSH, HTTP, HTTPS apps in 10.148.0.0/24.     8. Test if you can access a private instance via the internal IP when WARP is enabled on the end device     9. You can also have the option to [apply zero trust network access policies to the subnet](#_f40i39i9hmu4) by creating an application for the subnet from access, and defining who can access the subnet/instance in the subnet  (optional) Resolve Private IP and Hostnames [<link>](https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/private-net/private-hostnames-ips)  * + - 1. Once you have connected your private network via Cloudflare Tunnel and you can connect to it via WARP, you can set up Cloudflare tunnel to work with your private DNS to resolve internal/private hostnames       2. Ensure UDP proxy is enabled          1. Under Settings > Network > Firewall > Proxy, ensure UDP is enabled          2. Create Local Domain Fallback entry to enable internal/private DNS resolver to resolve specific domains       3. Under Settings > Network > Local Domain Fallback, click manage          1. Create a new entry pointing to the internal DNS resolver.          2. e.g. For private hostname myorg.privatecorp, have an internal resolver at 10.0.0.25 resolve the hostname rather than attempting to resolve this publicly      * + - * 1. Ensure that Split Tunnels are configured to include traffic to private IPs and hostnames in the traffic sent by WARP to Cloudflare <[link](https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/private-net#optional-ensure-that-traffic-can-reach-your-network)>         2. Once you are done, you can run the tunnel again and you will now be able to resolve the private hostname  1. You can also see the tunnel health in the dashboard under Access > Tunnels    (optional) Deploy Cloudflared Replicas [<link>](https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/run-tunnel/deploy-cloudflared-replicas)  *Cloudflare for Teams allows you to deploy many cloudflared instances through the same tunnel. The same tunnel can represent multiple, redundant instances of cloudflared, giving your team the ability to scale instances dynamically. This allows for highly scalable and available tunnels. You can read more about the use cases* [*here*](https://blog.cloudflare.com/highly-available-and-highly-scalable-cloudflare-tunnels/)*.* | |
| **Success/Test Criteria** | |
| * Successful installation of Cloudflared * Successful deployment of Tunnels that are active on the dashboard in teams (Access > Tunnels) | |

## 

## Configure Cloudflare Gateway

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Configuring a Default Location <[Link](https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/locations/configuring-a-location)>  Note: you can use Gateway to filter and log DNS queries from any device in your network without installing client software using locations <[guide](https://developers.cloudflare.com/cloudflare-one/tutorials/secure-dns-network)>    *e.g. Applying DNS filtering policy to all networks other than those in locations* Configuring Gateway Policies <[Link](https://developers.cloudflare.com/cloudflare-one/policies/filtering)> Configuring DNS Filtering Policies  * 1. You can get documentation on setting up DNS policies [here](https://developers.cloudflare.com/cloudflare-one/policies/filtering/dns-policies-builder)   2. All DNS policies are set under teams dashboard > Gateway > Policies > DNS     *e.g. Using DNS filtering to block access to any host under netflix.com*    *e.g. Using DNS filtering to override unreachable DNS queries to your domain* Configuring HTTP Filtering Policies  * 1. You can get documentation on setting up HTTP policies [here](https://developers.cloudflare.com/cloudflare-one/policies/filtering/http-policies)   2. All HTTP policies are set under teams dashboard > Gateway > Policies > HTTP     *e.g. Using HTTP filtering to block file uploads to google drive (*[*guide*](https://developers.cloudflare.com/cloudflare-one/tutorials/block-uploads)*)*    *e.g. Using HTTP Filtering to block malware* Configuring Network Filtering Policies  * 1. You can get documentation on setting up Network policies [here](https://developers.cloudflare.com/cloudflare-one/policies/filtering/network-policies)   2. All Network policies are set under teams dashboard > Gateway > Policies > Network     *e.g. Using Network policy to override requests to a destination network and port to specified network and port*   1. **In enabling the policies, you can create block rules and display a customised block page according to the guide** [**here**](https://developers.cloudflare.com/cloudflare-one/policies/filtering/configuring-block-page) 2. **Test the policy**    1. You can test if the policy is working by doing a dig or nslookup to see if the policy is working as intended.    2. if the block page is enabled, you should see no error in the answer section and 162.159.36.12 and 162.159.46.12 as answers when the domain is successfully blocked | |
| **Success/Test Criteria** | |
| * Successful creation of Gateway Location * Successful creation and testing of Gateway DNS, HTTP and Network Policies | |

## 

## 

## Configure Cloudflare Remote Browsing

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Note: Remote Browsing is set up under Gateway  <[Link](https://developers.cloudflare.com/cloudflare-one/policies/browser-isolation)>   1. **Set up the HTTP Policy**     *e.g. Isolate all security risks*   * 1. prerequisite for remote browsing is to have WARP Client installed on end devices   2. note: under Step 4 Configure Policy Settings, you can define granular browser controls to prevent data from leaving the enterprise  1. **Test the isolated browsing experience** <[link](https://developers.cloudflare.com/cloudflare-one/policies/browser-isolation/configuration)> | |
| **Success/Test Criteria** | |
| * Successful creation and testing of Remote Browsing HTTP policy | |

## 

## View Analytics & Configure Logpush

| **Status** | Not Started |
| --- | --- |
| **Configuration Steps** | |
| Analytics Cloudflare for Teams offers comprehensive and in-depth visibility into your network. We provide 2 types of analytics available in the dashboard under Analytics tab [<link>](https://developers.cloudflare.com/cloudflare-one/analytics)   * Access: [Shadow IT Analytics](https://developers.cloudflare.com/cloudflare-one/analytics/access) shows which SaaS applications your end users are visiting so you to create the appropriate Access or Secure Web Gateway policies   ​​   * Gateway: [Gateway Analytics](https://developers.cloudflare.com/cloudflare-one/analytics/gateway) shows you top allowed and blocked requests across all of your locations.  User Logs Teams also provides a User Log under My Team > Users [<link>](https://developers.cloudflare.com/cloudflare-one/analytics/users) which provides a comprehensive list of users who have authenticated to Teams. For each user that logged in, you can see their name, their email address, and whether they’re actively utilizing a seat in both Access and Gateway. You can also revoke the sessions and users here.   Admin Logs Under Logs > Admin, you can get access to the audit logs in teams to view all the recent changes in configuration.   Access Logs Under Logs > Access, you can view all the logins across the corporate network at an application level [<link>](https://developers.cloudflare.com/cloudflare-one/analytics/logs/audit-logs)   Gateway Logs Under Logs > Gateway, you can view DNS, HTTP and Network activity across the corporate network as users in Gateway locations or running on WARP traverse the internet [<link>](https://developers.cloudflare.com/cloudflare-one/analytics/logs/activity-log)   Configure Logpush We can configure Logpush to selected SIEMs and 3rd party storage. Once exported, your team can analyze and audit the data as needed.<[link](https://developers.cloudflare.com/cloudflare-one/analytics/logs/activity-log#cloudflare-logpush-integration)>  Under Logs > Logpush, select Connect a service    Give the logs a name, select what logs you want to push to the SIEM/3rd party storage and then select the fields you want to push    Connect to your SIEM/3rd party storage provider of choice    In the SIEM/3rd party storage provider dashboard, create a connection and copy the HTTP source address to Cloudflare. We will push a log to you with the ownership token for verification purposes. | |
| **Success/Test Criteria** | |
| * Ability to view and filter analytics and logs * Logs are pushed to selected SIEM | |

# 

# Success Criteria

The Proof of Concept will be considered a success when the following criteria have been met:

| **Criteria** | **Comments** |
| --- | --- |
| Scalability |  |
| Impossible to do lateral movement in Zero Trust Network Access |  |
| Successful integration for TCP ZTNA routing and private web applications |  |
| Performance testing - overall improvement in performance |  |

# Timeline

| **Date** | **Task** | **Status** |
| --- | --- | --- |
| 20/01/22 | POC Kick-off   * Confirm POC requirements and logistics * Start POC |  |
| 27/01/22 | POC Check in |  |
| 31/01/22 | Close POC   * Review POC results and develop implementation schedule |  |

## 

## Proof of Concept Terms

* Customer reviews and agrees to POC scope outlined in this document. Both technical evaluator and Decision makers are required to agree to this by return email prior to the POC commencing
* Variations to the POC must be agreed between the Cloudflare AE and Decision maker and will be added to this document
* The evaluation period will be for a period of 2 weeks. The Solutions Engineer and customer will agree on a POC start date to ensure that resources can be aligned on both sides
* Upon completion of the POC, a POC review will be scheduled to present the findings of the POC. Both technical evaluator and Decision makers will be present for this review