



# Ivanti Tunnel 4.6.0 - 4.17.0 for Android Guide

April 2025

# Contents

---

<b>Ivanti Tunnel 4.6.0 - 4.17.0 for Android Guide</b> .....	<b>1</b>
<b>Revision history</b> .....	<b>3</b>
<b>New features summary</b> .....	<b>5</b>
Related information from previous releases .....	5
<b>About Ivanti Tunnel</b> .....	<b>7</b>
Overview .....	7
Ivanti Tunnel for Android support on UEM .....	7
Ivanti Tunnel configuration .....	8
Tunnel creation with Ivanti Tunnel for Android .....	8
App traffic allowed through Ivanti Tunnel VPN (Android native and Android Enterprise) .....	9
Send and receive IP packets with Ivanti Tunnel .....	11
Support for anti-phishing with Mobile Threat Defense .....	12
Controlling VPN traffic .....	16
<b>Setting up Ivanti Tunnel for Android native</b> .....	<b>19</b>
Before you configure Ivanti Tunnel for Android native (Ivanti EPMM and Ivanti Neurons for MDM) ..	19
Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM) .....	21
Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM) .....	23
<b>Setting up Ivanti Tunnel for Samsung Knox in Ivanti EPMM</b> .....	<b>27</b>
Before you configure Ivanti Tunnel for Samsung Knox .....	27
Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM) .....	29
Configuring an IP_ANY AppTunnel service on a Standalone Sentry .....	32
Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM) .....	33
Distributing Ivanti Tunnel through Apps@Work .....	35
Configuring app VPN in the Samsung Knox container .....	36
Configuring VPN chaining .....	38
<b>Setting up Ivanti Tunnel for Android Enterprise</b> .....	<b>41</b>
Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM) ..	41
Configuration tasks overview for Android Enterprise (Ivanti EPMM) .....	44
Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM) .....	47
<b>Ivanti Tunnel Configuration Fields and Custom Data</b> .....	<b>55</b>
Ivanti Tunnel for Android native configuration field description .....	55
Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace ..	59
Ivanti Tunnel configuration field description for Android Enterprise .....	67
Example showing the Sentry certificate in the certificate chain .....	78
<b>Ivanti Tunnel for Android device user experience</b> .....	<b>81</b>
Ivanti Tunnel installation on devices .....	81
Troubleshooting .....	84
Anti-phishing enabled user experience .....	90

---

# Revision history

**TABLE 1.** REVISION HISTORY

<b>Date</b>	<b>Revision</b>
April 30, 2025	Updated for Ivanti Tunnel 4.17.0.
March 18, 2025	Updated for Ivanti Tunnel 4.16.1.
January 15, 2025	Updated for Ivanti Tunnel 4.16.0.
October 7, 2024	Updated for Ivanti Tunnel 4.15.0.
July 4, 2024	Updated for Ivanti Tunnel 4.14.0.
April 23, 2024	Updated for Ivanti Tunnel 4.13.0.
October 12, 2023	Updated for Ivanti Tunnel 4.11.0.
November 03, 2022	Updated for Ivanti Tunnel 4.8.0.
July 18, 2022	Updated for Ivanti Tunnel 4.7.0.
April 06, 2022	Updated for Ivanti Tunnel 4.6.0 - 4.6.2.



# New features summary

These are cumulative release notes. If a release does not appear on this section, then there were no associated new features and enhancements.

## Ivanti Tunnel 4.17.0 - New features summary

- **Support for FQDN-based Split Tunnel:** Added the **FqdnAllowedList** functionality, which enables FQDN-based traffic management by allowing administrators to specify FQDN names instead of IP addresses. It enhances flexibility and efficiency by eliminating the need to manage multiple IP entries.  
For more information, see "[Ivanti Tunnel configuration field description for Android Enterprise](#)" on [page 67](#).
- **Android Version 9 – Minimum Support:** Ivanti Tunnel 4.17.0 now supports devices running Android 9 and above.

## Related information from previous releases

If a release does not appear in this section, then there were no associated new features and enhancements.

- [Ivanti Tunnel 4.16.1 - New features summary](#)
- [Ivanti Tunnel 4.16.0 - New features summary](#)
- [Ivanti Tunnel 4.15.0 - New features summary](#)
- [Ivanti Tunnel 4.14.0 - New features summary](#)
- [Ivanti Tunnel 4.13.0 - New features summary](#)
- [Ivanti Tunnel 4.11.0 - New features summary](#)
- [Ivanti Tunnel 4.8.0 - New features summary](#)

- [Ivanti Tunnel 4.7.0 - New features summary](#)
- [Ivanti Tunnel 4.6.2 - New features summary](#)
- [Ivanti Tunnel 4.6.0 - New features summary](#)

# About Ivanti Tunnel

The following provide an overview of Ivanti Tunnel for Android devices:

- ["Overview" below](#)
- ["Ivanti Tunnel for Android support on UEM " below](#)
- ["Ivanti Tunnel configuration" on the next page](#)
- ["Tunnel creation with Ivanti Tunnel for Android" on the next page](#)
- ["App traffic allowed through Ivanti Tunnel VPN \(Android native and Android Enterprise\)" on page 9](#)
- ["Send and receive IP packets with Ivanti Tunnel" on page 11](#)
- ["Support for anti-phishing with Mobile Threat Defense" on page 12](#)

## Overview


Ivanti Tunnel enables VPN capability on Android (native, with no containerization), Android Enterprise (containerized, previously called Android for Work) and Samsung Knox Workspace (containerized) devices.

Ivanti Tunnel interacts with the Unified Endpoint Management (UEM) platform, Standalone Sentry, and Access to allow apps and browsers on Android, Android Enterprise, and Samsung Knox devices to securely access enterprise resources from outside the enterprise network. The enterprise resource can be on premise or in the cloud. It provides the following UEM platforms: Ivanti EPMM and Ivanti Neurons for MDM.

## Ivanti Tunnel for Android support on UEM

The following table describes Ivanti Tunnel for Android support on the Unified Endpoint Management (UEM) platform.

**TABLE 2.** UEM SUPPORT

	Ivanti EPMM	Ivanti Neurons for MDM
Android native	Supported	Supported
Android enterprise	Supported	Supported <hr/>  Only work profile and work managed device modes are supported. <hr/>
Samsung Knox	Supported	Not supported

## Ivanti Tunnel configuration

Configurations for Ivanti Tunnel are created in a Unified Endpoint Management (UEM) platform, which are Ivanti EPMM and Ivanti Neurons for MDM. Ivanti Tunnel receives the configuration from the UEM client. The client for Ivanti EPMM is Mobile@Work, and the client for Ivanti Neurons for MDM is Go.

If you configure Ivanti Tunnel for Android native and Android Enterprise on the same Ivanti EPMM, ensure that the configurations are applied to the correct labels.

## Tunnel creation with Ivanti Tunnel for Android

The following describes how a tunnel session with Ivanti Tunnel for Android is created:

1. Tunnel validates the configuration syntactically.
2. Tunnel establishes a TCP connection with Standalone Sentry on port 443.
3. Ivanti Tunnel and Standalone Sentry mutually authenticate each other using TLS 1.2 using client identity certificates.  
The Android TLS stack is used for this purpose.
4. Standalone Sentry's certificate presented in the TLS handshake is compared with the Standalone Sentry certificate in the Ivanti Tunnel configuration. This step occurs if certificate pinning is enabled.



5. Tunnel initiates the AppTunnel protocol handshake:
  - a. POST with device ID, user ID, and service ID are sent to Standalone Sentry.
  - b. Standalone Sentry validates the parameters. For example, Standalone Sentry checks if the user or device is blocked.
  - c. Standalone Sentry provides additional configuration parameters: interface IP and DNS server IP.
  - d. The TCP connection is switched to the Tunnel protocol.
6. A VPN session is created using Android API VpnService.Builder.
  - a. VPN specific configuration is set in the VPN session based on the Ivanti Tunnel configuration created in UEM.
  - b. Android creates a TUN interface and the VPN icon is set in the system bar. The VPN icon indicates that the tunnel is established and available. The VPN icon (looks like a key for Android native and Android Enterprise, and like a lock for Samsung Knox) in the status bar indicates that the Tunnel session is available. It does not indicate if traffic from an app currently being used is going through the tunnel. The behavior is similar to that of the Wi-Fi icon.



Device users may also see the Tunnel notifications icon, which looks like the Ivanti Tunnel logo. The Tunnel notifications icon does not indicate that Ivanti Tunnel VPN is on. It only indicates that there are notifications from Ivanti Tunnel.

---

Traffic from an app is automatically tunneled through Ivanti Tunnel irrespective of when an app is installed. The app may have been installed before Ivanti Tunnel was initiated or after Tunnel was initiated.

## **App traffic allowed through Ivanti Tunnel VPN (Android native and Android Enterprise)**

When a Tunnel VPN session is created, the Ivanti Tunnel configuration is provided to the Android operating system. The Ivanti Tunnel configuration includes information such as allowed and disallowed apps, routes, and domain name servers. Android enforces access to Ivanti Tunnel, based on the provided configuration. The apps that use Ivanti Tunnel is determined by the allowed and disallowed configuration. You configure either an allowed list or a disallowed list.

- Allowed: Only the apps that are on the allowed list (whitelist) have access to Ivanti Tunnel. Traffic from all other apps is not allowed to go through Ivanti Tunnel and goes through the device network.
- Disallowed: All apps have access to Ivanti Tunnel, except the ones on the disallowed list (blacklist). Traffic from the disallowed list goes through the device network.

Ensure that you have configured either an allowed app list or a disallowed app list is not configured, Ivanti recommends adding at least the following to a disallowed list to avoid OS traffic going through Ivanti Tunnel VPN:

- Mobile@Work if your UEM is Ivanti EPMM (com.mobileiron)
- MobileIron Go if your UEM is Ivanti Neurons for MDM (com.mobileiron.anyware.android)
- Android play store (com.android.vending)
- Google Play Service (com.google.android.gms)
- Carrier Service (com.google.android.ims)
- (For Samsung devices) Samsung Experience Service (com.samsung.android.mobileservice)"

In addition, the following also determine how an app uses Ivanti Tunnel:

- ["Tunnel routes and Ivanti Tunnel for Android" below](#)
- ["DNS servers and Ivanti Tunnel for Android" on the next page](#)
- ["Always-on Tunnel VPN and Ivanti Tunnel for Android" on the next page](#)
- ["Connection recovery for Ivanti Tunnel for Android" on the next page](#)

## Tunnel routes and Ivanti Tunnel for Android

During the creation of the VPN session, configured routes are set to the TUN interface. If the administrator did not configure any routes in Ivanti Tunnel configuration, Tunnel uses 0.0.0.0/0. The configured routes are used in the following ways:

- Only traffic from apps that can use Ivanti Tunnel goes through the configured routes.
- You cannot configure a different set of routes for different allowed apps.

- Traffic from non Android Enterprise apps or to disallowed Android Enterprise apps does not go through the routes configured for Ivanti Tunnel.

## DNS servers and Ivanti Tunnel for Android

DNS requests coming from allowed apps are resolved by the domain name servers (DNS) configured for the VPN during the VPN creation session. These servers are different from the DNS for the original Wi-Fi or cellular connection.

In addition, the Ivanti Tunnel SplitDomain feature allows you to use two different domain name servers to resolve DNS requests, based on the requested domain. The two domain name servers typically are the DNS configured for the device network and the DNS configured for VPN.

## Always-on Tunnel VPN and Ivanti Tunnel for Android

On Android 5 and 6 devices, always-on is an Ivanti implementation. The feature is enabled by default. You can configure by using the key `appRunningCheckIntervalSec`, which configures the check interval.

On Android Enterprise devices running Android N (7.0) and through the most recently released version as supported by Ivanti, Google provides the always-on feature. You can configure the Google implementation of always-on VPN in the Android Enterprise (Android for Work) configuration in Ivanti EPMM and in the Always-on configuration in Ivanti Neurons for MDM.

## Connection recovery for Ivanti Tunnel for Android

If a connection fails, Ivanti Tunnel tries to reconnect periodically, by default. Ivanti Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals. Ivanti Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Ivanti Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Ivanti Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Ivanti Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

You can configure connection recovery using the following keys: `quickRetryMaxAttempts`, `quickRetryIntervalSec`, `slowRetryIntervalSec`.

## Send and receive IP packets with Ivanti Tunnel

The following describes how IP packets are sent and received between the app attempting to connect to a backend resource and Standalone Sentry:

1. The Android app posts the IP packets to the TUN interface.
2. The Tunnel plugin/service receives the IP packets from the TUN interface.
3. The packets are sent as payload of the TCP connection to Standalone Sentry.
4. Standalone Sentry sends the IP packets to the end destination.
5. Standalone Sentry receives IP packets from the end destination and sends the packets over the TCP connection to the Ivanti Tunnel plugin and posts it to the TUN interface.
6. The app gets the payload through the TUN interface.

TCP and UDP are supported. IPv4 is supported.



Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.

---

## Ivanti Tunnel connectivity probe

Ivanti Tunnel sends probes with VPN traffic, after a specified period of idle time, to check if the Tunnel connection to the VPN server is open. If Ivanti Tunnel does not receive a response for at least one of the probe packets, Tunnel closes the current connection and initiates a new connection to the VPN server. The following key-value pairs are available to allow administrators to customize the settings: AtpProbeldleSec, AtpProbeIntervalSec, AtpProbeCount. For information about the key-value pairs, see "[Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace](#)" on page 59.

## Support for anti-phishing with Mobile Threat Defense

Ivanti Tunnel VPN supports anti-phishing with a Mobile Threat Defense (MTD) setup that is enabled for anti-phishing.

You must have the following set up:

- On the Threat Management Console, an Anti-phishing policy.
- On a Unified Endpoint Management (UEM) platform, an anti-phishing policy with the **Use VPN to analyze malicious URLs** option selected.
- On the UEM, add Ivanti Tunnel for distribution to devices.

- Anti-phishing support is provided with Ivanti Tunnel for Android native and Android Enterprise. Samsung Knox is not supported.
- Anti-phishing checks are not applied to Sentry and Access traffic.
- Enable the Phishing Local Classifier and Remote classifier flags in the group on the Mobile Threat Management Console. Otherwise, the detection of phishing URLs is inconsistent across different types of Android devices.

For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the Mobile Threat Defense Solution Guide for Ivanti Neurons for MDM or Ivanti Threat Defense Solution Guide for Ivanti EPMM.

Ivanti Neurons for MDM deployments — See the following:

- ["Adding and configuring the Ivanti Tunnel app for Android native \(Ivanti Neurons for MDM\)" on page 24](#)
- ["Adding and configuring the Ivanti Tunnel app for Android Enterprise\(Ivanti Neurons for MDM\)" on page 48](#)

Ivanti EPMM deployments — See the following:

- ["Distributing through the app storefront \(Ivanti EPMM\)" on page 22](#)
- ["Adding and configuring the Ivanti Tunnel for Android Enterprise \(Ivanti EPMM\)" on page 44](#)

## Ivanti Tunnel for Android native

No additional Tunnel configurations are required for supporting anti-phishing. However, make sure to add the Ivanti Tunnel app to UEM and distribute the app to devices.

A default Ivanti Tunnel VPN configuration that supports anti-phishing is automatically available in the UEM. On Ivanti EPMM, the default Ivanti Tunnel configuration is available when you upgrade to Ivanti EPMM 11.0.0.0. If the option **Use VPN to analyze malicious URLs** is enabled in the anti-phishing policy on UEM, the default Ivanti Tunnel VPN configuration is automatically distributed to the devices to which the anti-phishing policy is distributed.

If you already have Ivanti Tunnel for Android native deployment using a custom Tunnel VPN configuration, the configuration is automatically updated to consume the MTD license and keys.

After initially deploying Ivanti Tunnel VPN to use for anti-phishing with Mobile Threat Defense, if you also want to deploy Ivanti Tunnel VPN to use with Access or Sentry, create a custom Tunnel configuration as described in one of the following sections. Use the link appropriate to your UEM:

- ["Configuration tasks overview for Ivanti Tunnel for Android native \(Ivanti Neurons for MDM\)" on page 23](#)
- ["Configuration tasks overview for Ivanti Tunnel for Android native \(Ivanti EPMM\)" on page 21](#)

The custom configuration automatically consumes the Threat Defense license and key and is also automatically distributed to devices applied to the Threat Defense label. The custom Tunnel configuration is replaces the default Tunnel configuration on devices.

## Ivanti Tunnel for Android Enterprise

For Tunnel for Android Enterprise deployments, in addition to the anti-phishing policy configurations on the Threat Management Console and the UEM, do one of the following:

- For a new Tunnel for Android Enterprise deployment, select the **Use Tunnel for Anti-phishing only** option when configuring Tunnel for Android Enterprise on the UEM.
- If you already have a Tunnel for Android Enterprise deployment, in the **Configuration Choices** section of the Ivanti Tunnel configuration, add a new configuration and select **Use Tunnel for Anti-phishing only**.

See one of the following for information about configuring and distribution Tunnel for Android Enterprise. Use the link appropriate to your UEM:

- ["Configuration tasks overview for Android Enterprise \(Ivanti Neurons for MDM\)" on page 47](#)
- ["Configuration tasks overview for Android Enterprise \(Ivanti EPMM\)" on page 44](#)

## Ivanti Tunnel deployment options for anti-phishing with MTD

You can deploy Tunnel VPN for anti-phishing with Mobile Threat Defense (MTD) only or in addition to Tunnel VPN with Sentry and Access. Any combination of the following Tunnel VPN deployments are supported:

- Data traffic to enterprise resources: This requires a Sentry deployment.
- Authentication traffic to enterprise cloud resources: This requires a Access deployment.

- Analyze phishing URLs: This requires an MTD deployment.

**i** Anti-phishing checks are not applied to Sentry and Access traffic.

## Ivanti Tunnel VPN and MTD deployment anti-phishing blocking behavior

The following table provides some Ivanti Tunnel VPN and MTD deployment scenarios and the corresponding anti-phishing blocking behavior.

**TABLE 3.** IVANTI TUNNEL VPN AND MTD DEPLOYMENT SCENARIOS AND ANTI-PHISHING BLOCKING BEHAVIOR

Ivanti Tunnel VPN + MTD deployment scenario	Anti-phishing blocking behavior
<ul style="list-style-type: none"> <li>• Route is not configured in Tunnel configuration. The default route is 0.0.0.0/0.</li> </ul>	<ul style="list-style-type: none"> <li>• All traffic is routed to Sentry.</li> <li>• Phishing URLs are not blocked.</li> </ul>
<ul style="list-style-type: none"> <li>• Route is configured in Tunnel configuration. For example, 10.0.0.0/8.</li> </ul>	<ul style="list-style-type: none"> <li>• Traffic in the range 10.0.0.0/8 is routed to Sentry.</li> <li>• Phishing URLs are not blocked for traffic going to Sentry.</li> <li>• Phishing URLs for traffic that is not in the range 10.0.0.0/8 is blocked.</li> </ul>
<p>On Android Enterprise in Work Profile mode, in Tunnel configuration,</p> <ul style="list-style-type: none"> <li>• Route is not configured. The default route is 0.0.0.0/0.</li> <li>• A web browser is configured in the allowed app list. For example, Chrome.</li> <li>• Another web browser is available in the container, but is not configured in the allowed or disallowed app list. For example, Firefox.</li> </ul>	<ul style="list-style-type: none"> <li>• Phishing URLs from Chrome are not blocked.</li> <li>• Phishing URLs from Firefox are blocked.</li> </ul>

## Controlling VPN traffic

Ivanti Tunnel VPN on Android native and Android Enterprise devices is always on. App traffic is allowed or disallowed based on the allowed (whitelist) or disallowed (blacklist) list, and the routes the administrator sets up in the Ivanti Tunnel VPN configuration.

The following table compares the behavior between Ivanti Tunnel for Android versus Ivanti Tunnel for iOS.



**TABLE 4.** COMPARISON BETWEEN TUNNEL FOR ANDROID AND IOS

Function	Behavior on Android	Behavior on iOS
Activating Ivanti Tunnel	<p>When Ivanti Tunnel is first launched on Android native devices, device users must accept the Ivanti Tunnel VPN connection and allow access to the Tunnel certificate.</p> <p>This is not applicable to Android Enterprise and Samsung KNOX devices.</p>	<p>If the Ivanti Tunnel VPN profile is installed on your device, the Ivanti Tunnel VPN connection is automatically turned on when you tap a supported managed app and the app attempts to connect to a backend resource.</p> <p>In rare cases, if the VPN connection is not turned on, you can manually turn on VPN in the Ivanti Tunnel app. Your IT administrator will tell you if you need to turn on VPN in the Tunnel app.</p>
Automatic Tunnel triggering	<p>By default, Ivanti Tunnel VPN is always on for Android native and Android Enterprise. User action is not required after the initial activation.</p> <p>If the user disables Tunnel, Tunnel is not triggered automatically. Users must re-enable Tunnel.</p> <p>In the Knox container, on-demand VPN is triggered by managed apps.</p>	<p>Managed apps or Safari domains can automatically trigger a Tunnel VPN session.</p>
Allowing app traffic	<p>Admin must create an allowed list or create an exclusion list to allow or block app traffic.</p>	<p>Admin must make apps managed and assign them Tunnel to enable traffic through Ivanti Tunnel.</p>
Domain name triggers	<p>Ivanti Tunnel VPN is always on. There is no triggering of VPN on Android devices.</p>	<p>Safari can trigger Tunnel using domain names.</p>
Per-app allow/block list	<p>No per-app information is sent to Standalone Sentry. Sentry cannot enforce allow/block lists at a per-app level.</p>	<p>Ivanti Tunnel sends per-app information to Sentry. Sentry can enforce blocking at a per-app level.</p>

**TABLE 4.** COMPARISON BETWEEN TUNNEL FOR ANDROID AND IOS (CONT.)

<b>Function</b>	<b>Behavior on Android</b>	<b>Behavior on iOS</b>
Notifications	Ivanti Tunnel can provide notifications to users for various events (connect/disconnect, allow/block).	When the device is out of compliance, per-app Ivanti Tunnel VPN cannot provide notifications to the user if traffic is blocked.
UDP support	Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.	Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.
ICMP support	ICMP is not supported.	ICMP is not supported.
IPv6	IPv6 is not supported.	IPv6 is not supported.

# Setting up Ivanti Tunnel for Android native

The following address the setup required for app VPN using Ivanti Tunnel for Android native in Ivanti EPMM and Ivanti Neurons for MDM:

- ["Before you configure Ivanti Tunnel for Android native \(Ivanti EPMM and Ivanti Neurons for MDM\)" below](#)
- ["Configuration tasks overview for Ivanti Tunnel for Android native \(Ivanti EPMM\)" on page 21](#)
- ["Configuration tasks overview for Ivanti Tunnel for Android native \(Ivanti Neurons for MDM\)" on page 23](#)

## Before you configure Ivanti Tunnel for Android native (Ivanti EPMM and Ivanti Neurons for MDM)

Before you configure Ivanti Tunnel for Android native, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- ["Required components for Ivanti Tunnel for Android native" below](#)
- ["Requirements for Ivanti Tunnel for Android native" on the next page](#)
- ["Recommendations for Tunnel for Android native" on the next page](#)
- ["Limitations for Ivanti Tunnel for Android native" on page 21](#)

## Required components for Ivanti Tunnel for Android native

The following components are required for Ivanti Tunnel deployment on Android native devices:

- Standalone Sentry with AppTunnel enabled or Access.
- Unified Endpoint Management (UEM) platform: Ivanti EPMM or Ivanti Neurons for MDM.

- Client for Android:
  - Ivanti EPMM: Mobile@Work
  - Ivanti Neurons for MDM: Go

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

## Requirements for Ivanti Tunnel for Android native

The following are requirements for deploying Ivanti Tunnel for Android native:

- If your deployment uses Standalone Sentry:
  - You must have installed Standalone Sentry. See the *Standalone Sentry Installation Guide*.
  - To allow Android 7 devices to use Ivanti Tunnel, Standalone Sentry must use a publicly trusted CA certificate.
  - Standalone Sentry must be set up for AppTunnel using identity certificates for device authentication.
  - For information about setting up a Standalone Sentry for AppTunnel, see *Sentry Guide for Ivanti EPMM* and *Sentry Guide for Ivanti Neurons for MDM*.
- If your deployment uses Access, ensure that Access is set up. See the *Access Guide* for information on how to set up Access.
- Ensure that the appropriate ports are open. See the *Ivanti Tunnel for Android Release Notes*

## Recommendations for Tunnel for Android native

The following are recommendations for deploying Ivanti Tunnel for Android native:

- Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by Ivanti does not accept self-signed certificates.
- If access to the ActiveSync server is going through Standalone Sentry, configure Tunnel so that email clients are excluded from being routed through Tunnel.

## Limitations for Ivanti Tunnel for Android native

The following are limitations of Ivanti Tunnel for Android native:

- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the apps using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and add more Standalone Sentry servers a
- Server authentication through Standalone Sentry with Kerberos is not supported.
- Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.

## Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM)

The following configuration tasks are required to set up Ivanti Tunnel. These configuration tasks are performed in the Ivanti EPMM Admin Portal.

1. ["Creating Ivanti Tunnel VPN configuration \(Ivanti EPMM\)" below.](#)
2. ["Distributing through the app storefront \(Ivanti EPMM\)" on the next page.](#)

### Before you begin

- If you are configuring app VPN, you must have created an IP\_ANY AppTunnel service in Standalone Sentry. For information on setting up an IP\_ANYTunnel service see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti EPMM.
- Ensure that you have created a certificates enrollment setting in Ivanti EPMM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.

## Creating Ivanti Tunnel VPN configuration (Ivanti EPMM)

Create Ivanti Tunnel (Android) VPN configuration in the Ivanti EPMM Admin Portal.

**Procedure**

1. In the Ivanti EPMM Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > VPN**.
3. Enter a name and description for the VPN settings.
4. For **Connection Type**, select **Tunnel (Android)**.
5. Add the necessary configurations and click **Save**.
6. Apply the appropriate label to the app to distribute it to Android devices.

**Next steps**

If you are distributing the app through the app storefront, go to "[Distributing through the app storefront \(Ivanti EPMM\)](#)" below.

**Related topics**

- "[Ivanti Tunnel for Android native configuration field description](#)" on page 55.
- "[Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace](#)" on page 59.

## Distributing through the app storefront (Ivanti EPMM)

Ivanti Tunnel can be added to the app storefront for distribution.

**Procedure**

1. In the Ivanti EPMM Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **Google Play**.
4. Enter "Tunnel" for **Application Name**, and click **Search**.
5. Select the line for Tunnel app.
6. Click **Next**.
7. Select "5.0" for **Min. OS Version**.

8. Click **Next**.
9. Select **Silently Install**.
10. Click **Finish**.
11. Apply the appropriate label to the app to distribute it to Android devices.

**Related topics**

For more information on adding and editing Android apps to the app catalog, see "Managing Mobile apps for Android" in the *Apps@Work Guide*.

## Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM)

The following configuration tasks are required to set up Tunnel. These configuration tasks are performed in Ivanti Neurons for MDM.

1. ["Creating Ivanti Tunnel VPN configuration for Android native \(Ivanti Neurons for MDM\)" on the next page](#)
2. ["Adding and configuring the Ivanti Tunnel app for Android native \(Ivanti Neurons for MDM\)" on the next page](#)

**Before you begin**

- If you are configuring app VPN, ensure the following:
  - You have created a Tunnel service for Android in Standalone Sentry. For information on setting up Standalone Sentry with a Tunnel service, see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti Neurons for MDM.
  - Standalone Sentry is set up to use identity certificates for device authentication.
  - You have created an Identity Certificate configuration in Ivanti Neurons for MDM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access, ensure that you have setup Access. For information about setting up Access see the *Access Guide*. As part of the Access setup, you will have created a Tunnel service.

## Creating Ivanti Tunnel VPN configuration for Android native (Ivanti Neurons for MDM)

Create a Tunnel VPN configuration in **Configurations**.

### Procedure

1. In Ivanti Neurons for MDM, go to **Configurations** > **+Add**.
2. Search for Ivanti Tunnel.
3. Click the **Tunnel** configuration.  
The **Create Tunnel Configuration** page displays.
4. Enter a name for the configuration and click **Android**.  
The configuration fields for Tunnel VPN for Android are displayed.
5. Add the necessary configurations and click **Next**.
6. Choose a distribution option for the configuration and click **Done**.  
The configuration is distributed to the subset of the devices to which the app is distributed. Select the same distribution option that you selected for the Tunnel for Android app.

### Next steps

["Adding and configuring the Ivanti Tunnel app for Android native \(Ivanti Neurons for MDM\)"](#) below.

### Related topics

- For a description of the configuration fields, see ["Ivanti Tunnel for Android native configuration field description"](#) on page 55.
- For a description of the key-value pairs, see ["Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace"](#) on page 59.

## Adding and configuring the Ivanti Tunnel app for Android native (Ivanti Neurons for MDM)

Upload the Tunnel app to Ivanti Neurons for MDM from Google Play and configure it to make it available to Android devices. You can download the app from Google Play.



**Procedure**

1. In the Ivanti Neurons for MDM portal, go to **Apps >App Catalog**.
2. Click **+Add** next to **App Catalog**.
3. Select **Google Play** from the catalog pulldown menu.
4. Use the search to locate the Tunnel app in the Google Play store.
5. Select the Ivanti Tunnel app and click **Next**.  
A description and screen shots of the app are displayed.
6. Make changes, as needed, and click **Next**.
7. Select an app delegation option, and click **Next**.
8. Select a distribution option and click Next.  
The configuration will be distributed to the devices in the group you selected.
9. Click **Install Application configuration settings** to configure the install options.
  - a. Edit the **Name** and **Description** of the settings if necessary.
  - b. **Install on Device**: Enable Install on devices, if you want to require that the app is installed on devices.
  - c. **Silently install on Samsung KNOX and Zebra devices**: This option is not applicable to Android native apps.
  - d. **Do not show app in end user App Catalog**: Select if you do not want the app displayed in the app catalog on users' devices.
10. Click **Next**.
11. Click **Promotion distribution configuration** settings and select a promotion option.  
The promotion option determines how the app appears in the app catalog on the device.
12. Click **Next** and then click **Done**.

**Related topics**

See the *Ivanti Neurons for MDM Guide* or help for more information on adding apps to the Ivanti Neurons for MDM app catalog.



# Setting up Ivanti Tunnel for Samsung Knox in Ivanti EPMM

The following address the setup required for app VPN using Ivanti Tunnel for Samsung Knox Workspace in Ivanti EPMM:

- ["Before you configure Ivanti Tunnel for Samsung Knox" below](#)
- ["Configuration overview for Ivanti Tunnel for the Samsung Knox container \(Ivanti EPMM\)" on page 29](#)
- ["Configuring an IP\\_ANY AppTunnel service on a Standalone Sentry" on page 32](#)
- ["Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace \(Ivanti EPMM\)" on page 33](#)
- ["Distributing Ivanti Tunnel through Apps@Work" on page 35](#)
- ["Configuring app VPN in the Samsung Knox container" on page 36](#)
- ["Configuring VPN chaining" on page 38](#)

## Before you configure Ivanti Tunnel for Samsung Knox

Before you configure Tunnel, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- ["Required components for Ivanti Tunnel for Samsung Knox" on the next page](#)
- ["Requirements for Ivanti Tunnel for Samsung Knox" on the next page](#)
- ["Recommendations for Ivanti Tunnel for Samsung Knox" on page 29](#)
- ["Limitations for Ivanti Tunnel for Samsung Knox" on page 29](#)

## Required components for Ivanti Tunnel for Samsung Knox

The following components are required for deploying Ivanti Tunnel for Samsung Knox:

- Standalone Sentry with AppTunnel enabled.
- Ivanti EPMM with the following:
  - Enabled for Samsung Knox. Ensure that the Samsung general policy is configured with the license for Samsung Knox.
  - Users have Samsung Knox-capable device.
- Ivanti Tunnel for Android.
- Android client: Mobile@Work.



Ivanti Tunnel and Mobile@Work for Android are available from the Google Play store.

---

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

## Requirements for Ivanti Tunnel for Samsung Knox

The following are required for deploying Ivanti Tunnel for Samsung Knox:

- Set up Ivanti EPMM for Samsung Knox. For more information, see the “Samsung Knox support” section in the *Ivanti EPMM Device Management Guide for Android*.
- Install Standalone Sentry. See the *Standalone Sentry Installation Guide*.
- Set up Standalone Sentry for AppTunnel using identity certificates for device authentication. For information about setting up a Standalone Sentry for AppTunnel, see the “Working with Standalone Sentry for AppTunnel” section in the *Sentry Guide for Ivanti EPMM*.
- Add the apps that will use the Ivanti Tunnel VPN to the app catalog on Ivanti EPMM and to the Samsung Knox container.  
For information about adding apps to the Ivanti EPMM app catalog see the “Adding Google Play apps for Android” and “Apps on Samsung Knox devices” sections in the *Ivanti EPMM Apps@Work Guide*.

## Recommendations for Ivanti Tunnel for Samsung Knox

Android 7 devices do not accept self-signed certificates. Therefore, Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate.

## Limitations for Ivanti Tunnel for Samsung Knox

The following are limitations of Ivanti Tunnel for Samsung Knox:

- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the applications using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and deploy additional Sentry servers as needed for horizontal scaling.
- The Certificate Enrollment created for Standalone Sentry setup for AppTunnel must use RSA key length 2048 due to a Knox limitation.
- Routes configured in the Knox VPN configuration in Ivanti EPMM are ignored by Samsung Knox Workspace. Route lists are not supported in the Knox Workspace. All traffic from an app that uses Ivanti Tunnel VPN goes over Tunnel.
- Server authentication through Standalone Sentry with Kerberos is not supported.
- Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.


## Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)

Configuration for Ivanti Tunnel VPN is done in the Ivanti EPMM Admin Portal. Do the following to setup Ivanti Tunnel in the Samsung Knox container:

1. ["Configuring an IP\\_ANY AppTunnel service on a Standalone Sentry" on page 32.](#)
2. ["Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace \(Ivanti EPMM\)" on page 33.](#)
3. ["Distributing Ivanti Tunnel through Apps@Work" on page 35.](#)
4. ["Configuring app VPN in the Samsung Knox container" on page 36.](#)

The VPN configuration for Ivanti Tunnel is done in two separate configurations in the Ivanti EPMM Admin Portal: the VPN configuration for **Tunnel (Samsung Knox Workspace)** and the **Samsung KNOX Container** configuration. The Ivanti Tunnel for Samsung Knox workspace VPN configuration sets the DNS and app behavior. The Samsung Knox container configuration determines whether the VPN configuration is applied per-app individually or to all apps in the container (per-container).

The VPN configuration also determines whether the connection is always-on or on-demand. With always-on VPN, Ivanti Tunnel is started when the Samsung Knox Workspace starts, and the connection stays on. Traffic from an app in the Knox Workspace can go through the Ivanti Tunnel VPN. With on-demand VPN, a Tunnel VPN connection is started when an app that uses Tunnel is launched, and the connection stays on till the last app that can use the Tunnel VPN is killed.

 Ivanti Tunnel must be available in the Samsung Knox Workspace. Sometimes an app can be available in the Knox container as well as outside the container. Only the app in the Knox container can use Tunnel.

The following table describes Tunnel behavior depending on the combination of whether VPN is on-demand or always-on and if the VPN configuration is applied per-app or per-container.

**TABLE 5.** TUNNEL BEHAVIOR

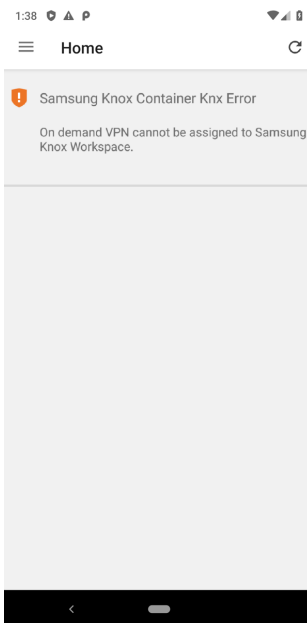
	On-demand	Always-on
Per-container	<p>Not a valid configuration for Samsung Knox.</p> <p>This combination is configurable in Ivanti EPMM, however Tunnel will not work. See <a href="#">"Error messages for a per-container and on-demand Tunnel VPN setup" on the next page.</a></p>	<p>Tunnel starts when the Samsung Knox Workspace container starts.</p> <p>All apps in the container can use Tunnel VPN.</p>
Per-app	<p>Tunnel starts when an app that can use Tunnel is launched.</p> <p>Tunnel stops when there are no apps running that can use Tunnel VPN.</p> <p>The per-app list, which is the list of apps that can use Tunnel VPN, is set in the Knox container configuration.</p>	<p>Tunnel starts when the Samsung Knox Workspace container starts.</p> <p>Only traffic from apps that are configured to use Tunnel are allowed through Tunnel.</p>

## Error messages for a per-container and on-demand Tunnel VPN setup

A per-container and on-demand combination VPN configuration is not supported. However, you can configure per-container and on-demand VPN in the Ivanti EPMM Admin Portal. After the device syncs with Ivanti EPMM, error messages are seen in Mobile@Work on the device and in the device profile in the Admin Portal.

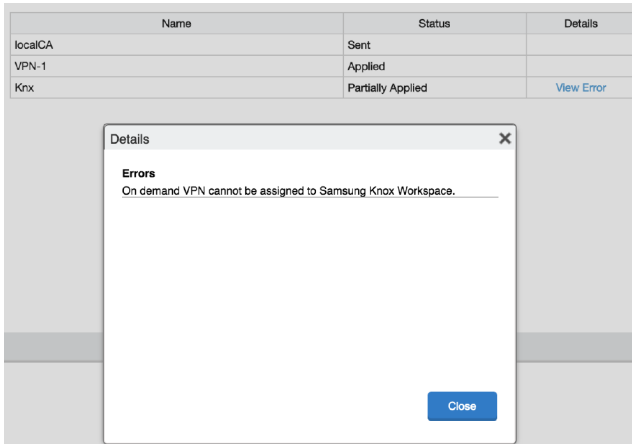
Mobile@Work displays the error as seen in the following figure.

FIGURE 1. DEVICE CONFIGURATION STATUS ERROR



In the Admin Portal, in **Devices & Users > Devices**, the **Configurations** tab for a device displays a link to **View Error** for the Samsung Knox container configuration.

FIGURE 2. VIEW CONFIGURATION ERROR IN THE ADMIN PORTAL



## Configuring an IP\_ANY AppTunnel service on a Standalone Sentry

Configure an IP\_ANY AppTunnel service on a Standalone Sentry enabled for AppTunnel. Ivanti Tunnel creates the tunnel through which traffic is tunneled to the backend resource.



If you already have an IP\_ANY AppTunnel service configured on a Standalone Sentry enabled for AppTunnel, you can skip this section.

### Procedure

1. In the Ivanti EPMM Admin Portal, go to **Settings > Sentry**.
2. Click **Edit** to open the Standalone Sentry settings.
3. In the **AppTunnel Configuration** section under **Services**, click the plus icon to add the following service:
  - **Service name:** <IP\_ANY>
  - **Server Auth:** Pass Through
  - All other fields: default
4. Click **Save**.



### Next steps

Go to "[Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace \(Ivanti EPMM\)](#)" below.

### Related topics

For information about setting up an AppTunnel service in Standalone Sentry, see the "Working with Standalone Sentry for AppTunnel" section in the *Sentry Guide for Ivanti EPMM*.

## Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM)

The Ivanti Tunnel (Samsung Knox Workspace) VPN configuration determines, DNS, and app behavior.

### Before you begin

- Enable Standalone Sentry for AppTunnel.
- Set up Standalone Sentry to use identity certificates for device authentication.
- Create a certificates enrollment setting in Ivanti EPMM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.

### Procedure

1. In the Ivanti EPMM Admin Portal, go to **Policies & Configs > Configurations**.
2. Click **Add New > VPN**.
3. Enter a name and description for the VPN settings.

4. Configure the following:

Item	Description
Connection Type	Select <b>Tunnel (Samsung Knox Workspace)</b>
Sentry	Select the Standalone Sentry on which you have enabled AppTunnel.
Identity Certificate	Select the Certificate Enrollment setting you created for Sentry setup for AppTunnel.
VPN Chaining	Disable: Default. Inner: Select to enable VPN chaining.
VPN on Demand	Select the check box to enable VPN on demand. If unchecked, the VPN connection is always on.

5. **Routes List** is not supported in the Samsung Knox Workspace. Routes configured here will be ignored.
6. In **DNS Resolver IPs**, configure the list of DNS for Tunnel.  
Each entry is separated by ';'. IPv4 only.  
The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination.
7. In **DNS Search Domain List**, enter a list of search domains for DNS resolver separated by a semicolon (;).
8. In Custom Data, add key-value pairs to configure the app.
9. Click **Finish**.
10. Apply the appropriate label to the app to distribute it to Samsung Knox devices.

### Next steps

To distribute the app through the app storefront, go to ["Distributing Ivanti Tunnel through Apps@Work" on the next page](#).

### Related topics

- See ["Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace" on page 59](#) for a description of the custom data key-value pairs.
- See ["Configuring VPN chaining" on page 38](#) for information about how to configure VPN Chaining.

- See also, "[Configuration overview for Ivanti Tunnel for the Samsung Knox container \(Ivanti EPMM\)](#)" on page 29.

## Distributing Ivanti Tunnel through Apps@Work

Adding Ivanti Tunnel to the app storefront allows you to determine which Samsung Knox devices will get the app.

### Procedure

1. In the Ivanti EPMM Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **Google Play**.  
The app wizard appears.
4. Type "Tunnel" for **Application Name**, and click **Search**.
5. Select the listing for Ivanti Tunnel.
6. Click **Next**.
7. Select "5.0" for **Min. OS Version**.
8. Click **Next**.
9. Select **Silently Install**.
10. Click **Finish**.
11. Apply the appropriate label to the app to distribute it to Samsung Knox devices.

### Next steps

Add the Tunnel app to the Samsung Knox container and to configure VPN for the apps that will use Ivanti Tunnel. See "[Configuring app VPN in the Samsung Knox container](#)" on the next page.

### Related topics

- For more information on adding and editing Google Play apps to the app catalog, see "Managing Mobile apps for Android" in the *Apps@Work Guide*.

## Configuring app VPN in the Samsung Knox container

Update the Samsung Knox container configuration:

- Add Tunnel to the Samsung Knox container configuration so that the app is available in the container on Samsung Knox devices.
- Configure the apps in the container to use Tunnel VPN.

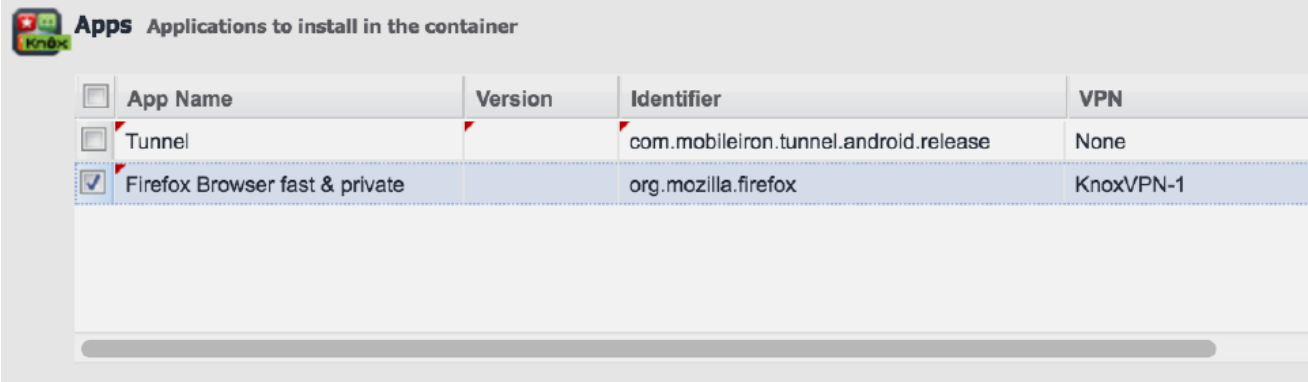
The Samsung Knox container configuration determines which VPN configuration is used and whether the VPN configuration is applied per app or per container.

Assigning different VPN configurations to apps is not supported. Example: Assigning VPN1 to App1 and VPN2 to App2 is not supported. Only one VPN configuration is supported in the Samsung Knox container. Two separate VPN configurations are allowed only for VPN chaining.

### Procedure

1. In the Ivanti EPMM Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the Samsung Knox container configuration and click **Edit**.

FIGURE 1. APPS CONFIGURATION



<input type="checkbox"/>	App Name	Version	Identifier	VPN
<input type="checkbox"/>	Tunnel		com.mobileiron.tunnel.android.release	None
<input checked="" type="checkbox"/>	Firefox Browser fast & private		org.mozilla.firefox	KnoxVPN-1

3. In the **Apps** section do the following:
  - a. Click **+** to add Ivanti Tunnel.
  - b. For **App Name**, select the Tunnel app from the drop down list.  
All other fields for the app are set to default values. Do not make any changes to the default values.
  - c. Similarly, if needed, add other apps to make the apps available in the Samsung Knox container.
4. Configure the apps to use Ivanti Tunnel VPN. Do one of the following:
  - ["Configuring per-app VPN" below.](#)
  - ["Configuring per-container VPN" on the next page.](#)
5. Click **Save**.

#### Related topics

See also, ["Configuration overview for Ivanti Tunnel for the Samsung Knox container \(Ivanti EPMM\)" on page 29.](#)

## Configuring per-app VPN

If you configure per-app VPN, only apps to which the Ivanti Tunnel VPN configuration is applied can use Tunnel VPN.

#### Procedure

1. In the Samsung Knox container configuration, scroll down to the **Apps** section.
2. For apps that will use Ivanti Tunnel VPN, in the **VPN** field, select the Tunnel (Samsung Knox Workspace) VPN configuration from the drop down list.

Only the specified apps can use Tunnel VPN.

Configure **VPN** in the **Apps** section only if a VPN configuration is not specified in the **Apps Settings** section.

3. Click **Save**.

#### Related topics

See also, ["Configuring app VPN in the Samsung Knox container" on the previous page.](#)

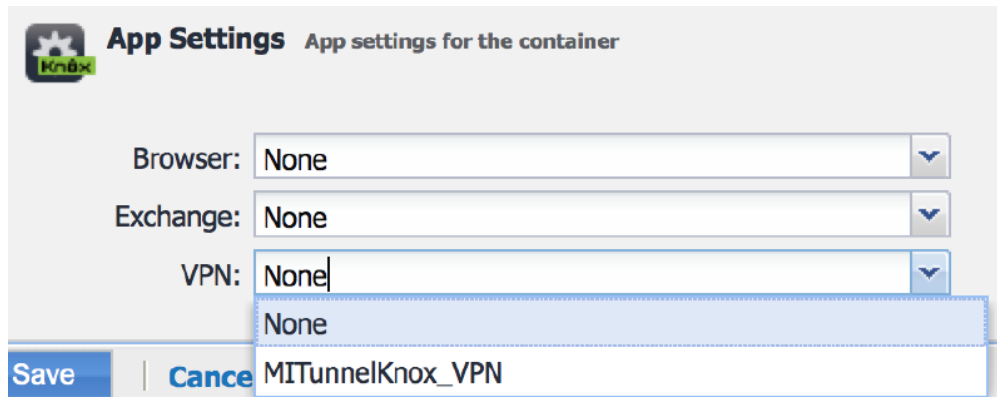
## Configuring per-container VPN

If you configure per-container VPN, all apps in the Knox container can use Ivanti Tunnel VPN.

### Procedure

1. In the Samsung Knox container configuration, scroll down to the **App Settings** section.

FIGURE 2. APPS SETTINGS CONFIGURATION



2. For **VPN** in **App Settings**, select the Ivanti Tunnel (Samsung Knox Workspace) VPN configuration from the drop down list.

The selected VPN configuration is applied to all apps in the Samsung Knox container.

Configure **VPN** in **App Settings** only if a VPN configuration is not specified for any app in the **Apps** section. If you configure **VPN** in the **App Settings**, the **VPN** selection in **Apps** automatically resets to **None**.

3. Click **Save**

### Related topics

See also, "[Configuring app VPN in the Samsung Knox container](#)" on page 36.

## Configuring VPN chaining

VPN chaining is the nesting of a VPN tunnel in another VPN tunnel. VPN chaining provides additional security by hiding the Ivanti Tunnel VPN end destination. With Ivanti Tunnel you can configure VPN chaining with OpenVPN as the outer tunnel and Tunnel as the inner tunnel. VPN chaining can be configured for per-app only.

## Before you begin

- Configure Tunnel for Samsung Knox Workspace as described in "[Configuration overview for Ivanti Tunnel for the Samsung Knox container \(Ivanti EPMM\)](#)" on page 29.
- Configure an OpenVPN VPN setting in the Ivanti EPMM Admin Portal. For more information, see the "Configuring new VPN settings" and the "OpenVPN" sections in the *Ivanti EPMM Device Management Guide* for Android.

Use the OpenVPN setting on Ivanti EPMM only to configure Samsung "OpenVPN net.openvpn.knox.connect" for Samsung Knox devices. The configuration is available only to limited customers as approved by Samsung. Contact Samsung to get the correct OpenVPN package. It is supported only on devices with the Samsung Knox option selected in the VPN setting.

## Procedure

1. In the Ivanti EPMM Admin Portal, go to **Policies & Configs > Configurations**.
2. Select and **Edit** the Ivanti Tunnel VPN configuration for Samsung Knox Workspace.
  - a. In the Ivanti Tunnel VPN configuration for Samsung Knox Workspace, for **VPN Chaining**, select **Inner**.
  - b. Click **Save**.
3. Select and **Edit** the OpenVPN configuration.
  - a. In the OpenVPN configuration, for **VPN Chaining**, select **Outer**.
  - b. Click **Save**.
4. Select and **Edit** the Samsung Knox container configuration.

FIGURE 1. APPS CONFIGURATION



<input type="checkbox"/>	App Name	Version	Identifier	VPN
<input type="checkbox"/>	Firefox Browser fast & private		org.mozilla.firefox	None
<input type="checkbox"/>	Google Chrome: Fast & Secure		com.android.chrome	Android Knox Config
<input type="checkbox"/>	Tunnel		com.mobileiron.tunnel.andro...	Samsung Open VPN
<input type="checkbox"/>	Facebook		com.facebook.katana	None

5. In the **Apps** section of the Samsung Knox container configuration, do the following:
  - a. For VPN for Tunnel, select the OpenVPN configuration with outer VPN chaining (Configured in step 3).
  - b. For apps that will use VPN chaining, select the Ivanti Tunnel VPN configuration with inner VPN chaining (Configured in step 2).
6. Ensure that the configurations are applied to a label that contains the devices for which you want to allow VPN chaining with Ivanti Tunnel.



# Setting up Ivanti Tunnel for Android Enterprise

The following address the setup required for app VPN using Ivanti Tunnel for Android Enterprise in Ivanti EPMM and Ivanti Neurons for MDM:

- ["Before you configure Ivanti Tunnel for Android Enterprise \(Ivanti EPMM and Ivanti Neurons for MDM\)" below](#)
- ["Configuration tasks overview for Android Enterprise \(Ivanti EPMM\)" on page 44](#)
- ["Configuration tasks overview for Android Enterprise \(Ivanti Neurons for MDM\)" on page 47](#)

## Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM)

Before you configure Ivanti Tunnel, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- ["Required components for deploying Ivanti Tunnel for Android Enterprise" below](#)
- ["Requirements for deploying Ivanti Tunnel for Android Enterprise" on the next page](#)
- ["Recommendations for deploying Ivanti Tunnel for Android Enterprise" on page 43](#)
- ["Limitations for Ivanti Tunnel for Android Enterprise" on page 43](#)
- ["Shared-kiosk mode" on page 43](#)

## Required components for deploying Ivanti Tunnel for Android Enterprise

The following components are required for Ivanti Tunnel deployment on Android Enterprise devices:

- Standalone Sentry with AppTunnel enabled or Access
- UEM with the following:
  - UEM enabled for Android Enterprise
  - Users have Android Enterprise-capable device.  
UEM is Ivanti EPMM or Ivanti Neurons for MDM.
- Client for Android Enterprise:
  - Ivanti EPMM: Mobile@Work
  - Ivanti Neurons for MDM: Go



Tunnel for Android Enterprise and Mobile@Work for Android are available from the Google Play store.

---

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

## Requirements for deploying Ivanti Tunnel for Android Enterprise

The following are required for deploying Ivanti Tunnel for Android Enterprise:

- Your Ivanti Neurons for MDM must be set up for Android Enterprise. For more information, see:
  - Ivanti EPMM: *Ivanti EPMM Device Management Guide for Android and Android Enterprise*.
  - Ivanti Neurons for MDM: *Getting Started with Android for Work*.
- If your deployment uses Standalone Sentry:
  - You must have installed Standalone Sentry. See the *Standalone Sentry Installation Guide*.
  - Standalone Sentry must be set up for AppTunnel using Identity certificates for device authentication.

For information about setting up a Standalone Sentry for AppTunnel, see:

*Standalone Sentry Guide for Ivanti EPMM* and *Standalone Sentry Guide for Ivanti Neurons for MDM*.

- If your deployment uses Access, ensure Access is set up.  
See the *Access Guide* for information on how to set up Access.

- Ensure that the appropriate ports are open.  
See the *Ivanti Tunnel for Android Release Notes*.

## Recommendations for deploying Ivanti Tunnel for Android Enterprise

The following are recommendations for deploying Ivanti Tunnel for Android Enterprise:

- Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by Ivanti does not accept self-signed certificates.
- If your deployment includes Android 5 and 6 devices, and if Standalone Sentry uses a self-signed certificate, see *Using a Self-signed certificate with Standalone Sentry and Tunnel* [knowledge base article](#) in the Support and Knowledge Base portal. The configuration sections describe the use of Ivanti EPMM UI. However for Ivanti Neurons for MDM as well, create a certificate setting and upload the Sentry server certificate to Ivanti Neurons for MDM and distribute the certificate setting to devices.
- If access to the ActiveSync server is going through Standalone Sentry, configure Ivanti Tunnel so that email clients are excluded from being routed through Tunnel.

## Limitations for Ivanti Tunnel for Android Enterprise

The following are limitations of Ivanti Tunnel for Android Enterprise:

- Deployments that use a trusted front-end such as Apache/F5 to terminate SSL or the use of backend proxy from Standalone Sentry to upstream applications are not supported. (Cloud only)
- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the apps using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and add more Standalone Sentry servers as needed for horizontal scaling.
- Real-time audio/video apps may not work. UDP functionality and scale will vary depending on the app. Performance of real-time audio and video apps has not been tested.
- Server authentication through Standalone Sentry with Kerberos is not supported.

## Shared-kiosk mode

Ivanti Tunnel for Android supports shared-kiosk mode. Before you deploy Ivanti Tunnel, ensure that shared-kiosk mode is set up and deployed. After the UEM is set up for shared-kiosk mode, follow the configuration tasks for setting up Ivanti Tunnel for Android Enterprise.

For a better user experience, Ivanti recommends updating the UEM client version to Go 72 for Android or Mobile@Work 10.8.0.0 for Android through the most recently released version as supported by Ivanti.

To set up shared-kiosk mode:

- For Ivanti EPMM, see "Android shared-Kiosk mode overview" in the *Ivanti EPMM Device Management Guide for Android and Android Enterprise devices*.
- For Ivanti Neurons for MDM, see "Setting up Kiosk Mode for Android" in the *Ivanti Neurons for MDM Guide*.

## Configuration tasks overview for Android Enterprise (Ivanti EPMM)

The following configuration tasks are required to set up app VPN with Ivanti Tunnel. These configuration tasks are performed in the Ivanti EPMM Admin Portal:

1. ["Adding and configuring the Ivanti Tunnel for Android Enterprise \(Ivanti EPMM\)" below](#)
2. ["Creating an Always-On VPN configuration \(Ivanti EPMM, optional\)" on page 47](#)

## Adding and configuring the Ivanti Tunnel for Android Enterprise (Ivanti EPMM)

Add the Ivanti Tunnel app to Ivanti EPMM from Google Play and configure it as follows to make it available to Android Enterprise devices.

### Before you begin

- If you are configuring app VPN, you must have created an IP\_ANY AppTunnel service in Standalone Sentry. For information on setting up an IP\_ANYTunnel service see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti EPMM.
- If you are configuring Tunnel to support anti-phishing with MTD, you must have an MTD setup enabled for anti-phishing. See ["Support for anti-phishing with Mobile Threat Defense" on page 12](#)

### Procedure

1. In the Ivanti EPMM Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.

3. Click **Google Play**.
4. Enter Tunnel for **Application Name**, and click **Search**.
5. Select the line for Tunnel app.
6. Click **Next**.
7. Select "5.0" for **Min. OS Version**.
8. Click **Next**.
9. Select **Install this app for enterprise**.  
Additional fields are exposed.
10. Select **Silently Install**.
11. Select **Enable Access** only if you have an Access as a service deployment.  
Selecting this option enables authentication traffic through Access. The option is available only if Access as a service is set up with Ivanti EPMM.
12. Scroll down to **Configuration Choices**.
13. Do one of the following:
  - Select **Use Tunnel for Anti-phishing only**, to enable Tunnel VPN to analyze phishing URLs.  
  
Do not select this option if you have any restrictions configured. Selecting the option removes any configured restrictions and hides the **Default Configuration for Tunnel** section. To configure anti-phishing when you have an existing Tunnel deployment, add a new Tunnel configuration and select the option.  
  
OR
  - Expand **Default Configuration for Tunnel** to configure the restrictions for the app.  
  
Select either **Use Tunnel for Anti-phishing only** or configure the restrictions under **Default Configuration for Tunnel**. To deploy Tunnel for MTD and for Sentry or Access, create two separate configurations.
14. Click **Finish**.

### Next steps

Go to "[Creating an Always-On VPN configuration \(Ivanti EPMM, optional\)](#)" on page 47.

### Related topics

- See ["Ivanti Tunnel configuration field description for Android Enterprise" on page 67](#) for a description of the restrictions.
- For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide*.
- For information about adding and configuring an Android Enterprise app, see "App configuration for Android Enterprise apps," in the *Apps@Work Guide*.
- For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the *Threat Defense Solution Guide for Ivanti Neurons for MDM*.
- ["Support for anti-phishing with Mobile Threat Defense" on page 12](#).
- ["Adding multiple Ivanti Tunnel configurations" below](#).

## Adding multiple Ivanti Tunnel configurations

You can create multiple Ivanti Tunnel configurations and assign the configuration to a label. One reason you may need to create multiple Tunnel configurations is when you configure Tunnel to support anti-phishing with an MTD deployment as well as for deployment with Sentry or Access.

When you add Ivanti Tunnel for Android Enterprise to the App Catalog, a default Tunnel configuration for MTD is automatically available. The Ivanti Tunnel MTD configuration is pushed to devices when you select **Use Tunnel for Anti-phishing only**. However, selecting the option removes all other restrictions. Therefore, to also configure Ivanti Tunnel for Sentry or Access add a separate Tunnel configuration.

If you have an existing Ivanti Tunnel configuration to use with Sentry or Access, add a new Tunnel configuration for anti-phishing and vice versa.

### Procedure

1. On Cloud, go to **Apps > App Catalog**.
2. Highlight and click the Ivanti Tunnel app for Android Enterprise.
3. On the **App Configurations** tab, for **Managed Configuration for Android** click + to add a new configuration.
4. Select **Use Tunnel for Anti-phishing only** or configure restrictions in **Managed Configurations**.

5. Select a distribution for the new configuration.
6. Click **Save**.

## Creating an Always-On VPN configuration (Ivanti EPMM, optional)

The Ivanti Tunnel app can be configured for Always-On VPN status for devices using Android 7 through the most recently released version as supported by Ivanti EPMM 9.3.

With Always-On VPN, the VPN connection is always on. Any app in the Android Enterprise container can go through the tunnel.

If a connection fails, Tunnel tries to reconnect periodically. Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals.

Ivanti Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

### Procedure

1. Go to **Policies & Configs > Configurations** and click the **Add New** pull down menu.
2. Select **Android > Android Enterprise** to display the **New Android enterprise Setting** screen.
3. Select the **Always-On VPN** check box to display the **App Identifier** pull down menu. The pulldown menu lists only apps that are configured to be installed as Android Enterprise apps.
4. Select a VPN app to apply the Always-On setting. Click **Save**.



In **Device Details**, the Android Enterprise setting displays as **Partially Applied** with an error message if the selected app is not installed on the device, the app is not a VPN app, or the VPN app does not support Always-on.

---

## Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM)

The following configuration tasks are required to set up Tunnel. These configuration tasks are performed in the Cloud Admin Portal.

1. ["Adding and configuring the Ivanti Tunnel app for Android Enterprise\(Ivanti Neurons for MDM\)" below.](#)
2. ["Creating an Always-On VPN configuration \(Cloud, optional\)" on page 51.](#)

## Adding and configuring the Ivanti Tunnel app for Android Enterprise (Ivanti Neurons for MDM)

Upload the Ivanti Tunnel app to Ivanti Neurons for MDM from Google Play and configure it to make it available to Android Enterprise devices. You can download the app from Google Play.

### Before you begin

- Ensure that you have met the requirements detailed in ["Before you configure Ivanti Tunnel for Android Enterprise \(Ivanti EPMM and Ivanti Neurons for MDM\)" on page 41.](#)
- If you are configuring app VPN,
  - You must have created a Tunnel service for Android in Standalone Sentry. For information on setting up Standalone Sentry with a Tunnel service, see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti Neurons for MDM.
  - Standalone Sentry must be set up to use identity certificates for device authentication.
  - Ensure that you have created a Identity Certificate configuration in Ivanti Neurons for MDM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access, you must have setup Access. For information about setting up Access see the *Access Guide*. As part of the Access setup, you will have created a Tunnel service. If you are configuring Ivanti Tunnel to support anti-phishing with MTD, you must have an MTD setup enabled for anti-phishing. See ["Support for anti-phishing with Mobile Threat Defense" on page 12](#)

### Procedure

1. In the Ivanti Neurons for MDM portal, go to **Apps >App Catalog**.
2. Click **+Add** next to **App Catalog**.
3. Select **Google Play** from the catalog pull-down menu.



4. In the search text box, enter Tunnel to locate the app in the Google Play store.
5. Click on the Tunnel icon in the search results.

A description and screen captures of the app are displayed.

6. Click **Select**.  
Options to add categories and a description are displayed.
7. Make changes as needed and click **Next**.

The App Delegation screen displays noting that AFW is enabled.

8. Click **Next**.
9. Select a distribution option and click **Next**.  
The configuration will be distributed to the devices in the group you selected.

The App Configurations screen displays.

10. Click + for **Managed Configurations for Android** to configure settings for the app.
11. Enter a name and description for the configuration.
12. Select **Blocks the user for uninstalling the app** if you do not want device users to uninstall the app.
13. Do one of the following:

- Select **Use Tunnel for Anti-phishing Only**, to enable Tunnel VPN to analyze phishing URLs.

Do not select this option if you have any restrictions configured. Selecting the option removes any configured restrictions and hides the **Managed Configurations** section. To configure anti-phishing when you have an existing Tunnel deployment, add a new configuration and select the option.

OR

- Expand **Managed Configurations** to configure the restrictions for the app.

Select either **Use Tunnel for Anti-phishing Only** or configure the restrictions under **Managed Configurations**. To deploy Tunnel for MTD and for Sentry or Access, create two separate configurations.

14. Select a distribution option for the configuration and click **Next**.
15. Click **Install Application configuration settings** to configure the install options.
  - a. Edit the **Name** and **Description** of the settings if necessary.
  - b. **Install on Device**: Enable Install on devices, if you want to require that the app is installed on devices.
  - c. **Silently install on Samsung KNOX and Zebra devices**: This option is not applicable to Android Enterprise apps.
  - d. **Do not show app in end user App Catalog**: Select if you do not want the app to display in the app catalog on users' devices.
16. Click **Next**.
17. Click **Promotion distribution configuration** settings and select a promotion option. The promotion option determines how the app appears in the app catalog on the device.
18. Click **Next** and then click **Done**.

### Next steps

Go to "[Creating an Always-On VPN configuration \(Cloud, optional\)](#)" on the next page.

### Related topics

- See "[Ivanti Tunnel configuration field description for Android Enterprise](#)" on page 67 for a description of the restrictions.
- For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide*.
- For information about adding and configuring an Android Enterprise app, see "Adding Google Play Store app for Android Enterprise," in the *Ivanti Neurons for MDM Guide*.
- For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the *Mobile Threat Defense Solution Guide for Cloud*.
- "[Support for anti-phishing with Mobile Threat Defense](#)" on page 12.
- "[Adding multiple Tunnel configurations](#)" on the next page.

## Adding multiple Tunnel configurations

You can create multiple Tunnel configurations and assign the configuration to a distribution group. One reason for creating multiple Tunnel configurations is when you configure Tunnel to support anti-phishing with an MTD deployment as well as for deployment with Sentry or Access.

Use the procedure described here to create additional Tunnel configurations for anti-phishing or for tunneling to Sentry or Access.

If you have an existing Tunnel configuration to use with Sentry or Access, add a new Tunnel configuration for anti-phishing and vice versa.

### Procedure

1. On the Ivanti EPMM Admin Portal, go to **Apps > App Catalog**.
2. Select the Tunnel app for Android Enterprise and click **Edit**.
3. Scroll down to **Configuration Choices**.
4. Click **Add+** to add a new Tunnel configuration.
5. Select **Use Tunnel for Anti-phishing only** or expand **Configuration for Tunnel** to configure the restrictions for the app.
6. If you configured restrictions for the Tunnel app, apply the new configuration to a label.
7. Click **Save**.

## Creating an Always-On VPN configuration (Cloud, optional)

The Ivanti Tunnel app can be configured for Always-On VPN status for devices using Android 7 through the most recently released version as supported by Ivanti.

With Always-On VPN, the VPN connection is always on. Any app in the Android Enterprise container can go through the tunnel.

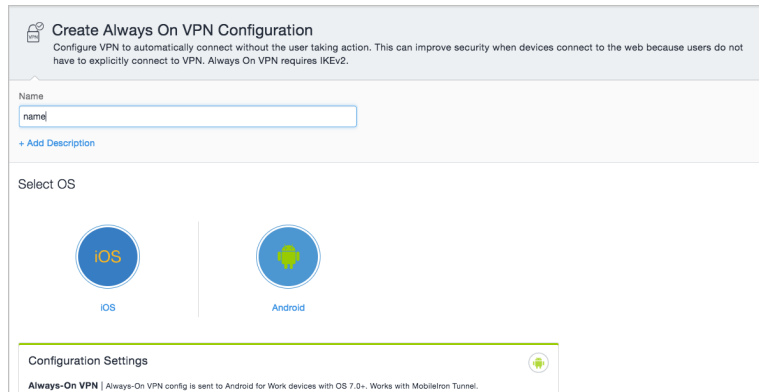
If a connection fails, Tunnel tries to reconnect periodically. Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals.

Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

### Procedure

1. In Cloud, go to **Configuration** and click **+Add**.
2. Click **Always On VPN**.
3. Enter a name for the configuration.
4. Select the **Android** operating system.

FIGURE 1. ALWAYS ON VPN CONFIGURATION



**Create Always On VPN Configuration**  
Configure VPN to automatically connect without the user taking action. This can improve security when devices connect to the web because users do not have to explicitly connect to VPN. Always On VPN requires IKEv2.

Name  
name

+ Add Description

Select OS

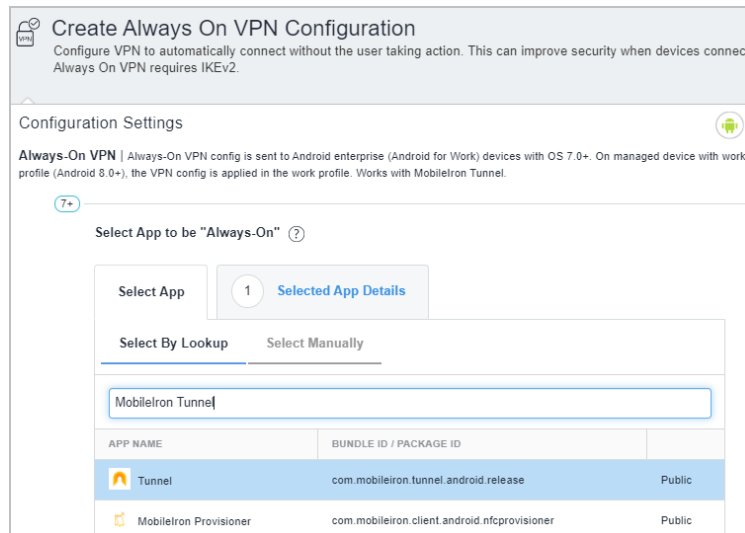
IOS Android

Configuration Settings  
Always-On VPN | Always-On VPN config is sent to Android for Work devices with OS 7.0+. Works with MobileIron Tunnel.

5. In **Configuration Settings**, enter Tunnel in the search box.

6. Select the Tunnel app. Click **Next**.

FIGURE 2. SELECT APP FOR ALWAYS-ON



7. Select a distribution group, and click **Done**.



# Ivanti Tunnel Configuration Fields and Custom Data

The following describe the configuration fields and key-value pairs for configuring Ivanti Tunnel for Android:

- ["Ivanti Tunnel for Android native configuration field description" below](#)
- ["Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace" on page 59](#)
- ["Ivanti Tunnel configuration field description for Android Enterprise" on page 67](#)
- ["Example showing the Sentry certificate in the certificate chain" on page 78](#)

## Ivanti Tunnel for Android native configuration field description


The following table provides field descriptions for the Ivanti Tunnel configuration. There are some variations in field names between Ivanti EPMM and Ivanti Neurons for MDM.

**TABLE 6.** TUNNEL CONFIGURATION FIELD DESCRIPTION

Item	Description
Name	Enter a name for the Tunnel VPN profile.
Description	Enter a description for the profile.
Connection Type (Ivanti EPMM)	Select <b>Tunnel (Android)</b> . Only fields relevant to Tunnel for Android are displayed.
Choose OS to create Tunnel Configuration (Ivanti Neurons for MDM)	Click <b>Android</b> . Fields relevant to Tunnel for Android are displayed.
Enable Access (Ivanti EPMM)	Select to enable authentication traffic through Access. The option is available only if Access as a service is set up with Ivanti. For information about how to set up Access as a service with Ivanti EPMM, see the <i>Access Guide</i> .
Profile selection mode to use for this configuration (Ivanti Neurons for MDM)	Select one of the following: <ul style="list-style-type: none"> <li>• <b>Sentry Profile Only:</b> Select if Tunnel traffic goes only through Standalone Sentry.</li> <li>• <b>Access Profile Only:</b> Select if Tunnel traffic goes to Access. This option is available only if an Access as a service deployment is set up with Cloud.</li> <li>• <b>Sentry + Access Profile:</b> Select if Ivanti Tunnel VPN supports both traffic to Access for authentication to enterprise cloud resources and through Standalone Sentry to on-premise enterprise resources. This option is available only if an Access as a service deployment is set up with Ivanti Neurons for MDM.</li> </ul>
Sentry (Profile)	<b>Ivanti EPMM:</b> Select the Standalone Sentry on which you created the IP_ANY tunnel service. <b>Cloud:</b> Select the Standalone Sentry profile on which you created the Tunnel service for Android. The option is not available if the profile mode is <b>Access Profile Only</b> .
Sentry Service (Ivanti Neurons for MDM)	Select the Tunnel service you created for Android. The option is not available if the profile mode is <b>Access Profile Only</b> .
Identity Certificate (Ivanti EPMM)	Select the Certificate Enrollment setting you created for Sentry setup for AppTunnel.





**TABLE 6.** TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
Client Cert. Alias (Ivanti Neurons for MDM)	Select the Identity Certificate configuration you created for Standalone Sentry setup.  If the profile mode is Access only or Sentry + Access, select the same certificate you select for SCEP Identity.
SCEP Identity (Ivanti Neurons for MDM)	Select the Identity Certificate configuration you created for Ivanti Tunnel.  This field is applicable if the profile mode is Access only or Sentry + Access.
Debug Info Recipient (Ivanti Neurons for MDM)  For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	Enter a valid email address. The device debug logs are sent to the configured email address.  When users tap <b>Email Debug Info</b> , the To field is auto filled with the configured email address.
UI Notification Level (Ivanti Neurons for MDM)  For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure. <ul style="list-style-type: none"> <li>• Never show notifications: Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel.</li> <li>• Error notifications only: Only errors notifications are displayed. This is the default setting if the key-value is configured.</li> <li>• All notifications: Error notifications and connect/disconnect confirmations are displayed.</li> </ul> <hr/> <div style="display: flex; align-items: center;">  <p>There are no notifications to indicate that an app is blocked or allowed.</p> </div> <hr/>
Debug Log (Ivanti Neurons for MDM)  For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	Select the log level. The client app can override the VPN profile.

**TABLE 6.** TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
Tunneled Applications (Ivanti EPMM)	<p>Select one, either <b>Add Allowed Apps</b> or <b>Add Disallowed Apps</b>, to configure the apps that can use Tunnel.</p> <p>If you select an app from the app catalog, the package name is automatically added. Otherwise, enter the app name and the package name. If the list is empty, all apps are allowed through Tunnel VPN.</p>
Add Allowed apps	<p>Use this setting if you want only the listed apps to use Tunnel VPN. Only apps in the App Catalog can be added to the app list. This setting creates a whitelist.</p> <p>For Cloud,</p> <ul style="list-style-type: none"> <li>• enter a semicolon (;) separated list.</li> <li>• if <b>Allowed Apps List</b> is configured, the <b>Disallowed Apps List</b> setting is grayed out and vice versa.</li> </ul>
Add Disallowed apps	<p>Use this setting if you do not want the listed apps to use Ivanti Tunnel VPN. Only apps that are not listed will use Tunnel VPN. This setting creates a blacklist.</p> <p>For Cloud,</p> <ul style="list-style-type: none"> <li>• enter a semicolon (;) separated list.</li> <li>• if <b>Allowed Apps List</b> is configured, the <b>Disallowed Apps List</b> setting is grayed out and vice versa.</li> </ul>

**TABLE 6.** TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
Routes List / Added Routes	<p>Configure the network routes that are allowed through Ivanti Tunnel. Use CIDR format. Each entry in the list is separated by ';'. IPv4 only.</p> <p>This enables split tunneling where only specific traffic can be taken through Tunnel. The routes configured only impact apps that use Ivanti Tunnel.</p> <p>Example: 10.0.0.0/8;101.210.48.9/32</p> <hr/> <p> In an Access deployment, if routes are not configured, then authentication traffic that is federated through Access goes to Access and all data-traffic goes to Sentry.</p> <p> Ivanti recommends configuring a route list so that only traffic destined to on-premise enterprise resources goes through Standalone Sentry and all other data traffic goes directly to the destination.</p> <hr/>
DNS Resolver IP	<p>Configure the list of DNS for Tunnel.</p> <p>Each entry is separated by ';'. IPv4 only.</p> <p>The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination.</p>
Search Domain	<p>Enter a list of search domains for DNS resolver separated by a semicolon (;)</p>
<p><b>Custom Data</b></p> <p>Add key-value pairs to configure the app. See <a href="#">"Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace"</a> below for a description of the restrictions.</p>	

## Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace

The following table provides a description of the custom data key-value pairs.

**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION

Key	Value: Enter	Description
<b>Manage Tunnel timeout</b>		
TcpIdleTmoMs	<i>An integer</i>	<p>The Tunnel TCP session idle timeout, on Standalone Sentry, in milliseconds.</p> <p>Tunnel sends this value to Standalone Sentry during the initial handshake in header X-App-TcpIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour).</p> <p>Frequently, in production environments, there are firewalls and load balancers between the device and Standalone Sentry. Each network element may have a different idle timeout, shorter than the timeout for Standalone Sentry. Ivanti recommends that the value for TcpIdleTmoMs is less than the idle timeout for all the other network elements.</p> <p>As an alternative, consider configuring TCP keep-alive.</p>
<b>VPN connection</b>		
AllowBypass (Android native only)	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	<p><b>true:</b> Allows all apps to bypass this VPN connection. Apps may use methods such as <code>setProcessDefaultNetwork(Network)</code> to send and receive directly over the underlying network or any other network for which they have permissions.</p> <p><b>false:</b> Default, if the key-value pair is not configured. All traffic from apps is forwarded through the VPN interface. Apps cannot bypass the VPN.</p>


**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
SplitDomainsList	<i>List of domain suffixes separated a semicolon (;)</i>	<p>Example: acme.com; google.com</p> <p>DNS requests with domains matching the values are sent to the DNS for the VPN. DNS requests with non-matching domains are sent to the device's DNS.</p> <p>Example: All DNS queries that match *.company.com are handled by the VPN DNS server, but all other queries are handled by the device network DNS i.e. not the VPN DNS server.</p> <p>The DNS handler for the Tunnel plugin decides which DNS request will be sent to which DNS server, based on the configured domains:</p> <ul style="list-style-type: none"> <li>• All sub domains are matched. Example: example.com matches example.com, staf.example.com, and jira.example.com</li> <li>• The configured domain is considered completed with top domains. Anything to the right of the top domain is omitted. Example: example.com does not match example.com.akamai.com</li> <li>• Only complete domains are matched. Example: example.com does not match myexample.com</li> <li>• '*' and '?' are not valid characters for the configuration.</li> </ul> <p>The filtering is done on an IP packet level, therefore, DNS resolver functionality is not provided. The default behavior sends all DNS requests to the DNS for the VPN.</p>


**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
SplitUDPPortList	List of UDP ports separated by a semicolon (;)	List of UDP ports to send through Ivanti Tunnel VPN. All other UDP packets are sent directly to destination. If the key-value pair is not configured, all UDP packets are sent through Ivanti Tunnel VPN.  <b>Example</b> 53;161-162;200-1024
MTU	<i>An integer</i>	Tunnel MTU. The default value if the key-value is not configured is 1400
quickRetryMaxAttempts	<i>An integer</i>	Number of attempts to reconnect to VPN. The default if the key-value pair is not configured is 3.
quickRetryIntervalSec	<i>An integer</i>	Time between attempts to reconnect to VPN in seconds. The default if the key-value pair is not configured is 1.
slowRetryIntervalSec	<i>An integer</i>	Time between attempts to reconnect to VPN in seconds. The default if the key-value pair is not configured is 60.
TcpKeepCount	<i>An integer</i>	The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead. The default value, if the key-value pair is not configured, is 20. The key is part of the Android operating system specifications.
TcpKeepIntervalSec	<i>An integer</i>	The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds. The default value, if the key-value pair is not configured, is 2 seconds. The key is part of the Android operating system specifications.

**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)


Key	Value: Enter	Description
AtpProbeldleSec	<i>An integer</i>	<p>Sets the minimum idle time, in seconds, after which probe packets are sent out with outbound Tunnel traffic. If Tunnel does not receive a response for at least one of the probes sent, the existing connection is dropped and a new connection is established with the server.</p> <p>The minimum idle time is based on the last inbound response received by Tunnel. For example, if the value is 60 seconds, if Tunnel does not receive any inbound traffic for 60 seconds, probe packets are sent with the next outbound Tunnel traffic.</p> <p>Default value if the key-value pair is not configured: 60 seconds</p>
AtpProbeIntervalSec	<i>An integer</i>	<p>Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 1 second</p>
AtpProbeCount	<i>An integer</i>	<p>Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 5</p>
<b>Certificates</b>		
DisablePinning	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	<p><b>false:</b> Default, if the key-value pair is not configured. Certificate pinning is enabled.</p> <p><b>true:</b> Certificate pinning is disabled. Disabling certificate pinning is not recommended for security reasons.</p> <hr/> <p> The Standalone Sentry server certificate is automatically pushed to the device.</p> <hr/>
<b>Troubleshooting</b>		

**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)


Key	Value: Enter	Description
UINotificationLevel	<ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 2</li> </ul>	<p>The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure.</p> <p>Configure one of the following levels of user notifications that the Tunnel app will provide:</p> <ul style="list-style-type: none"> <li>• 0: Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel.</li> <li>• 1: Only errors notifications are displayed. This is the default setting if the key-value is configured.</li> <li>• 2: Error notifications and connect/disconnect confirmations are displayed.</li> </ul> <hr/> <p> There are no notifications to indicate that an app is blocked or allowed.</p>
DebugLog	<ul style="list-style-type: none"> <li>• 0</li> <li>• 6</li> <li>• 4</li> <li>• 3</li> <li>• 2</li> </ul>	<p>Controls the amount of logging. The client app can override the VPN profile.</p> <ul style="list-style-type: none"> <li>• 0: Default setting if the key-value pair is not configured. Minimal level of logs are collected.</li> <li>• 6: ERROR level</li> <li>• 4: INFO level.</li> <li>• 3: DEBUG level</li> <li>• 2: VERBOSE level</li> </ul>



**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
AllowCapture	<ul style="list-style-type: none"> <li>false</li> <li>true</li> </ul>	<p>Allows users to capture traffic in a PCAP file.</p> <p><b>false:</b> Device users are not allowed to trigger inner traffic capture.</p> <p><b>true:</b> Device users are allowed to trigger inner traffic capture and email the PCAP file.</p> <p>The default, if the key-value pair is not configured, is false.</p> <hr/> <p> The PCAP file may contain sensitive information.</p> <hr/>
debugInfoRecipient	<i>Email address</i>	<p>The device debug logs are sent to the configured email address.</p> <p>When users tap <b>Email Debug Info</b>, the To field is auto filled with the value configured for debugInfoRecipient.</p>
EnableUserControl	<ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>	<p><b>true:</b> Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user.</p> <p><b>false:</b> Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user.</p> <p>Default value if the key-value pair is not configured: true</p> <p>The key-value pair is not applicable to Tunnel deployed in the Samsung Knox workspace. By default, device users in the Samsung Knox workspace do not have the option to enable or disable Tunnel VPN.</p>
DefaultMaxNumLogs	<i>An integer</i>	<p>Sets the maximum number of log files.</p> <p>The default if the key-value pair is not configured is 8.</p>
DefaultMaxPcapSize	<i>An integer</i>	<p>Sets the maximum pcap file size in bytes.</p> <p>The default if the key-value pair is not configured is 2097152.</p>
DefaultMaxNumPcaps	<i>An integer</i>	<p>Sets the maximum number of pcap files.</p> <p>The default if the key-value pair is not configured is 10.</p>

**TABLE 7.** TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
AnalyticsEnabled	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	<p><b>true:</b> Enables collection of analytics data for Mixpanel.</p> <p><b>false:</b> Collection of analytics data is disabled.</p> <p>Default value if the key-value pair is not configured: true.</p>
SendDeviceID	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	<p><b>true:</b> Ivanti Tunnel provides the device ID to Access.</p> <p>The device ID is reported on Access in Reports &gt; Errors.</p> <p><b>false:</b> Ivanti Tunnel does not provide the device ID to Access.</p> <p>The key-value pair is useful in identifying devices that encounter connection errors when authenticating through Access.</p> <p>Default value if the key-value pair is not configured: false</p>
<b>Tethering</b>		
ExcludeTethering	<ul style="list-style-type: none"> <li>• true</li> <li>• false</li> </ul>	<p><b>true:</b> Ivanti Tunnel VPN continues to work on the tethered host device without impacting the tethering client connection.</p> <p><b>false:</b> Ivanti Tunnel VPN may impact the tethering client connection.</p> <p>Default value if the key-value pair is not configured: false</p> <p>This key-value pair may be required for Ivanti Tunnel for Android native only.</p> <p>If the KVP is configured to true, ensure that internal IP ranges do not overlap with the IP ranges used by the tethering client. Avoid the following IP ranges:            192.168.42.0/23 (192.168.42.0 ~ 192.168.43.255)            192.168.44.0/22 (192.168.44.0 ~ 192.168.47.255)            192.168.48.0/23 (192.168.48.0 ~ 192.168.49.255)</p> <hr/> <p> Tethering traffic from client devices does not go through the VPN of the host device.</p> <hr/>




## **Ivanti Tunnel configuration field description for Android Enterprise**

The following table provides a description of the configuration fields for Ivanti Tunnel enterprise.

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE

Restriction	Description
Ivanti Tunnel profile mode ( Ivanti Neurons for MDM)	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry.</li> <li>• Access Profile Only: Select if Tunnel traffic goes to Access. This option is available only if an Access as a service deployment is set up with Cloud.</li> <li>• Sentry + Access Profile: Select if Ivanti Tunnel VPN supports both traffic to Access for authentication to enterprise cloud resources and through Standalone Sentry to on-premise enterprise resources. This option is available only if an Access as a service deployment is set up with Ivanti Neurons for MDM.</li> </ul>
Sentry Server	<p>Specify the FQDN for the Sentry server that is configured with the IP_ANY service. Configure <b>Sentry Server</b> if you selected one of the following Ivanti Tunnel profile modes:</p> <ul style="list-style-type: none"> <li>• Sentry Profile Only</li> <li>• Sentry + Access Profile</li> </ul>
AllowedAppList	<p>Optional. Use only if <b>DisallowedAppList</b> is empty. Applies only to apps in the Android Enterprise work profile.</p> <p>Provide a list of apps in the Android Enterprise profile that are allowed to use the Ivanti Tunnel VPN connection by supplying the app package names, separated by ‘;’.</p> <p><b>Example</b></p> <p>Example: com.salesforce.chatter;com.appexample.two</p> <p>If <b>AllowedAppsList</b> has one or more entries, only the apps in the list are allowed to use VPN.</p> <p>This is a whitelist.</p>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
DisallowedAppList	<p>Optional. Use only if <b>AllowedAppList</b> is empty. Applies only to apps in the enterprise work profile.</p> <p>Provide a list of applications in the Android Enterprise profile to be prevented from using Tunnel by supplying the app package names separated by ';':</p> <p>Example: com.salesforce.chatter;com.appexample.two</p> <p>If <b>AllowedAppList</b> is empty, then all apps can use VPN except the apps in the <b>DisallowedAppList</b>.</p> <p>This is a blacklist.</p> <p>Configuration conditions:</p> <ul style="list-style-type: none"> <li>• Both DisallowedList and DNSResolverIP Configured:</li> <li>• DNS resolved by non-DNSResolverIP for DisallowedList apps, independent of SplitDomainsList priority.</li> </ul> <hr/> <p> Anti-phishing URLs not blocked for DisallowedList apps as traffic bypasses tunnel.</p>
AllowBypass	<p>Select to allow all apps to bypass this Ivanti Tunnel VPN.</p>
AddedRoutes	<p>Enter the network routes that are allowed through Ivanti Tunnel. Use CIDR format. Each entry in the list is separated by a semicolon (;). IPv4 only.</p> <p>This enables split tunneling where only specific traffic can be taken through Tunnel. The routes configured only impact apps that use Tunnel.</p> <p>Example: 10.0.0.0/8;101.210.48.9/32</p> <hr/> <p> In an Access deployment, if routes are not configured, then authentication traffic that is federated through Access goes to Access and all data-traffic goes to Sentry.</p> <p> Ivanti recommends configuring a route list so that only traffic destined to on-premise enterprise resources goes through Standalone Sentry and all other data traffic goes directly to the destination.</p>


**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
ExcludedRoutes	<p>This API excludes a network route from the VPN interface. Calling this method overrides previous calls to addRoute(IpPrefix) for the same destination. This functionality leverages the new Android API level 33 to allow exclusion of specified IP routes from VPN traffic, directing them through the device's native network.</p> <p>If multiple routes match the packet destination, route with the longest prefix takes precedence.</p>
DNSResolverIP	<p>Enter the list of DNS for Ivanti Tunnel. Each entry is separated by a semicolon (;). IPv4 only.</p> <p>The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination.</p> <p>Configuration conditions:</p> <ul style="list-style-type: none"> <li>• DNSResolverIP Configured, AllowedAppList Not Configured: DNS resolved by IP in DNSResolverIP.</li> <li>• Both DNSResolverIP and AllowedAppList Configured: <ul style="list-style-type: none"> <li>◦ Browsing with AllowedAppList app: DNS resolved by DNSResolverIP.</li> <li>◦ Browsing with non-AllowedAppList app: DNS resolved by device DNS.</li> </ul> </li> </ul>
SplitUdpPortList	

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
SplitDomainsList	<p>Enter a list of domains suffixes separated by a semicolon (;).</p> <p>Example: mobileiron.com;google.com</p> <p>DNS requests with domains matching the values are sent to the VPN's DNS. DNS requests with non-matching domains are sent to the device's DNS.</p> <p>Example: All DNS queries that match *.company.com are handled by the VPN DNS server, but all other queries are handled by the device network DNS i.e. not the VPN DNS server.</p> <p>The Ivanti Tunnel plugin's DNS handler decides which DNS request will be sent to which DNS server, based on the configured domains:</p> <ul style="list-style-type: none"> <li>• All sub domains are matched. Example: mobileiron.com matches mobileiron.com, taf.mobileiron.com, and jira.mobileiron.com</li> <li>• The configured domain is considered completed with top domains. Anything to the right of the top domain is omitted. Example: mobileiron.com does not match mobileiron.com.akamai.com</li> <li>• Only complete domains are matched. Example: mobileiron.com does not match mymobiliron.com</li> <li>• '*' and '?' are not valid characters for the configuration.</li> </ul> <p>The filtering is done on an IP packet level, therefore DNS resolver functionality is not provided.</p> <p>The default behavior sends all DNS requests to the VPN's DNS Server.</p>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)


Restriction	Description
IsSplitDomainsListisPriority	<p>Domains (DNS IP) are resolved based on this configuration if SplitDomainsList and DNSResolverIP are configured.</p> <p>If 'Yes', SplitDomainsList takes priority and DNS is resolved based on SplitDomainsList configuration.</p> <p>If 'No', DnsResolvedIP takes priority and DNS is resolved based on DnsResolvedIP configuration.</p> <hr/> <p> Default for "IsSplitDomainsListisPriority" is Yes.</p> <hr/> <p>Configuration conditions:</p> <ul style="list-style-type: none"> <li>• If SplitDomainsList Configured: DNS resolved by SplitDomainsList.</li> <li>• If SplitDomainsList Not Configured but DNSResolverIP Configured: DNS resolved by DNSResolverIP.</li> <li>• If Neither Configured: DNS resolved by Sentry DNS.</li> </ul>
SearchDomain	Enter a list of search domains for DNS resolver separated by a semicolon (;).
SentryService (Ivanti Neurons for MDM only)	Name of the IP Tunnel service defined on Sentry.
SentryPort (Ivanti EPMM only)	Sentry Tunnel port. Use port 443, typically.
ClientCertAlias	<p><b>Ivanti EPMM</b></p> <p>This is the certificate alias set up in Ivanti EPMM. The value is <code>\$CERT_ALIAS:&lt;name-of-SCEP&gt;\$</code> where <code>&lt;name-of-SCEP&gt;</code> is the <b>Certificate Enrollment</b> setting configured in Ivanti EPMM UI.</p> <p>Example: <code>\$CERT_ALIAS:scepIdentityCert\$</code> where <code>scepIdentityCert</code> is the name of the SCEP configured in Ivanti EPMM.</p> <p><b>Ivanti Neurons for MDM</b></p> <p>Select the Identity certificate setting you created.</p>




**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
SentryCertificate ( Ivanti EPMM only)	<p>Copy and paste the Sentry certificate from the <b>sentry-server-cert-chain.pem</b> file.</p> <p>This is required if <b>DisablePinning</b> is not selected.</p> <p>For information on how to retrieve the <b>sentry-server-cert-chain.pem</b> file see <a href="#">KB article</a>.</p> <p>For an example of which section of the <b>sentry-server-cert-chain.pem</b> file to copy, see "<a href="#">Example showing the Sentry certificate in the certificate chain</a>" on page 78.</p>
DisablePinning	<p>Disabling certificate pinning is not recommended for security reasons. If selected, the <b>SentryCertificate</b> is not required.</p>
EnableUserControl	<p>Select the check box to enable.</p> <p>Enabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user.</p> <p>Disabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user.</p>
enableOpenssl	<p>Enable this parameter to reduce the reading and writing time to Sentry. It also enhances performance for low-end devices where the performance is limited by device or application speed.</p>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
UINotificationLevel	<p>Choose one of the following levels of user notifications that the Tunnel app will provide:</p> <ul style="list-style-type: none"> <li>• <b>Never show notifications:</b> Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel.</li> <li>• <b>Error notifications only:</b> Only errors notifications are displayed.</li> <li>• <b>All notifications:</b> Error notifications and connect/disconnect confirmations are displayed.</li> </ul> <p>The user will see error notifications or all Ivanti Tunnel related notifications, based on the level of notifications you choose.</p> <hr/> <p> There are no notifications to indicate that an app is blocked or allowed.</p>
DebugLog	<p>Controls the amount of logging. The client app can override the VPN profile.</p> <ul style="list-style-type: none"> <li>• Default setting if the key-value pair is not configured. Minimal level of logs are collected.</li> <li>• ERROR level</li> <li>• INFO level.</li> <li>• DEBUG level</li> <li>• VERBOSE level</li> </ul>
TrafficVerboseLog	<p>Captures traffic logs.</p> <ul style="list-style-type: none"> <li>• Off: Default setting. No logs are collected.</li> <li>• Minimal: Minimal logs are collected.</li> <li>• All: Detailed logs are collected.</li> </ul>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
Allow traffic capture	<p>Allows users to capture traffic in a PCAP file.</p> <hr/> <p> The PCAP file may contain sensitive information.</p> <hr/>
TcpIdleTmoMs	<p>The Ivanti Tunnel TCP session idle timeout, on Standalone Sentry, in milliseconds.</p> <p>Tunnel sends this value to Standalone Sentry during the initial handshake in header X-App-TcpIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour).</p> <p>Frequently, in production environments, there are firewalls and load balancers between the device and Standalone Sentry. Each network element may have a different idle timeout, shorter than the timeout for Standalone Sentry. Ivanti recommends that the value for TcpIdleTmoMs is less than the idle timeout for all the other network elements.</p> <p>As an alternative, consider configuring TCP keep-alive.</p>
UdpIdleTmoMs	
MTU	<p>Enter an integer for Tunnel MTU.</p> <p>The default value is 1400.</p>
DebugInfoRecipient	<p>Provide an email address.</p> <p>The device debug logs are sent to the configured email address.</p> <p>When users tap <b>Email Debug Info</b>, the To field is autofilled with the value configured for debugInfoRecipient.</p>
quickRetryMaxAttempts	<p>Number of attempts to reconnect to VPN.</p> <p>The default is 3.</p>
quickRetryIntervalSec	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default is 1.</p>
slowRetryIntervalSec	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default is 60.</p>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
appRunningCheckIntervalSec	<p>Time between app status checks in seconds.</p> <p>By default this key is enabled with an interval of 60 seconds.</p> <p>To disable this key, enter 0.</p>
TcpKeepIdleSec	<p>Enables or disables TCP keep-alive and specifies the interval between the last data packet sent and the first keep-alive probe in seconds. ACKs are not considered as data.</p> <p>A value of 0 means TCP keep-alive is disabled.</p> <p>The default value, if the key-value pair is not configured, is 0.</p> <p>TCP keep-alive helps detect a dead tunnel connection and prevents most network load balancers and firewalls from idle-out the connection. The Standalone Sentry TcplIdleTmoMs is not impacted by TCP keep-alive.</p> <p>The key is part of the Android operating system specifications.</p>
TcpKeepCount	<p>The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead.</p> <p>The default value, if the key-value pair is not configured, is 20.</p> <p>The key is part of the Android operating system specifications.</p>
TcpKeepIntervalSec	<p>The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds.</p> <p>The default value, if the key-value pair is not configured, is 2 seconds.</p> <p>The key is part of the Android operating system specifications.</p>
AtpProbeldleSec	<p>Sets the minimum idle time, in seconds, after which probe packets are sent out with outbound Tunnel traffic. If Tunnel does not receive a response for at least one of the probes sent, the existing connection is dropped and a new connection is established with the server.</p> <p>The minimum idle time is based on the last inbound response received by Tunnel. For example, if the value is 60 seconds, if Tunnel does not receive any inbound traffic for 60 seconds, probe packets are sent with the next outbound Tunnel traffic.</p> <p>Default value if the key-value pair is not configured: 60 seconds</p>

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
AtpProbeIntervalSec	Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeldleSec. Default value if the key-value pair is not configured: 1 second
AtpProbeCount	Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeldleSec. Default value if the key-value pair is not configured: 5
AtpProbeldleLimit	
InternalDebugOption1	Use only if instructed by Support for troubleshooting purposes.
TunIP	Use only if instructed by Support for troubleshooting purposes.
MaxNumLogs	Specify the maximum number of log files. The default is 8.
MaxNumPcaps	Specify the maximum number for pcap files. The default is 10.
AnalyticsEnabled	Check to enable collection of analytics data for Mixpanel. The box is checked by default.
SaveAfwConfiguration	Enable this configuration only if requested by Support.
AutoBackgroundLaunch	Check to enable the Tunnel app to automatically launch. The app is automatically launched without user interaction when a user tries to connect to a backend resource. For the feature to work, ensure that always-on is also enabled. The feature is available on Android N, O, and P.
AllowPerAppTunnel	For internal use only. Do not use this setting.
ClientCertsNumInChain	The value designates the number of certificates in the certificate chain that are passed to Sentry or Access. By default, only the leaf certificate is used. Ivanti recommends not changing the default setting unless additional certificates need to be passed to Sentry or Access.

**TABLE 8.** CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
SendDeviceID	
slowRetryMaxAttempts	Allows restarting the tunnel session. When <b>slowRetryMaxAttempts</b> are reached, the session stops, and a new connection begins. If the key-value pair is not configured, the default value is 0 seconds, which disables auto restart.
FqdnAllowedList	<p>Allows FQDN-based traffic management by letting admins define traffic rules using domain names rather than IP addresses.</p> <ul style="list-style-type: none"> <li>• Routing behavior based on configuration combinations: <ul style="list-style-type: none"> <li>◦ Only <b>FqdnAllowedList</b> configured: All FQDNs are resolved and routed via Sentry.</li> <li>◦ <b>FqdnAllowedList</b> + <b>AddedRoutes</b>: FQDNs in the <b>FqdnAllowedList</b> and IPs in the <b>AddedRoutes</b> are routed via Sentry.</li> <li>◦ <b>FqdnAllowedList</b> + <b>AddedRoutes</b> + <b>ExcludeRoutes</b>: All traffic goes through Sentry except for traffic to IPs specified in the <b>ExcludeRoutes</b>.</li> <li>◦ <b>FqdnAllowedList</b> + <b>AllowedAppList</b>: Only apps in the <b>AllowedAppList</b> can send traffic from FQDNs in the <b>FqdnAllowedList</b> via Sentry. All other traffic is routed through non-Sentry.</li> </ul> </li> </ul>

## Example showing the Sentry certificate in the certificate chain

The Sentry certificate is in bold. Copy and paste the section in bold for pinning.

```
Certificate(s) for host: app1416.auto.mobileiron.com
Certificate:          C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support,
CN=app1416.auto.mobileiron.com
Serial Number:       3173868363
Signature Algorithm: SHA256withRSA
Issuer:              C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support,
CN=ProxyCA
Validity:            Sat Aug 07 16:22:47 UTC 2021
PEM:
```

```
-----BEGIN CERTIFICATE-----
MIIFKjCCAxKgAwIBAgIFAL0tY0swDQYJKoZIhvcNAQELBQAwwczEQMA4GA1UEAwwH
UHJveHlDQTEQMA4GA1UECwwHU3VwcG9ydDETMDEGALUECgwKTW9iaWxlSXJvbjEw
MBQGA1UEBwwNTW91bnRhaW4gVmlldzETMBEALUECAwKQ2FsaWZvc5pYTELMAkG
A1UEBhMCVVMwHhcNMTYwODA4MTYyMjQ3WhcNMTYwODA3MTYyMjQ3WjCBhZEkMCIG
A1UEAwWbYXBwMTQxNi5hdXRvLm1vYm1sZWlyb24uY29tMRAwDgYDVQQLDAdTdxBw
b3J0MRMwEQYDVQKDApNb2JpbGVJcm9uMRYwFAYDVQQHDA1Nb3VudGFpbWV3
MRMwEQYDVQKDApDyWxpZm9ybmlhMQswCQYDVQKGEwJVUzCCASIDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAIYdxpmUGJy6Z3BJ21MxBs5w3kKVddANQmo1cVJC
InhJLrm4lK3Mazs09/2bF3t+ND8xTkI2YjRiZaz94B2dkrI7fcX0r7tjbStcXUP
yM6+49ipuBxjUKJNs20ZFJdRC0VK8ecbBS1DFOnlIW+fGUEqtWVA/k3nrwoeMfeP
zKg4hHBzB4+B369nzyIkxXxy9gUKfRLEs/kxWAexJB8eopxf6Zdf9W8tUp15h1c
ar6m3TY07pL3KU03U0K7mJXx0lsYqES8DHkTHfj2jYqnEqhxNurwTARYYmuV4iFU
BuztHaKzE00Sco4SMtBqa3FdU0J9EH11tiuzPqPFA4HTxA0CAwEAAoBrzCBrdAJ
BgNVHRMEAjAAMBGA1UdDgQMBApTZW50cnktS2V5MIGJBgNVHSMGgYEWf6F3pHUw
czEQMA4GA1UEAwwHUHJveHlDQTEQMA4GA1UECwwHU3VwcG9ydDETMDEGALUECgwK
TW9iaWxlSXJvbjEwMBQGA1UEBwwNTW91bnRhaW4gVmlldzETMBEALUECAwKQ2Fsa
WZvc5pYTELMAkGALUEBhMCVVOBACs7+swDQYJKoZIhvcNAQELBQADggIBAeV/
sdXPHxUHZSBuKbBpj2h8oXQa4Nlz1FawqNzbdBTktmUgqkcyu2hxilu6Mg1iqKc
Uz6IfLI3zMw4QULhw1aqlaAlqkpQ9x45wySx+BufzpiC00qkC28wJdKnBODM3Jig
CpvJcElvS3jTYuyjgJRNbaM0HGGIrHu4NBGrHljevHawOHTkvr9QmYhHhT2XYnug
FFX5Gic1ot9vGLA+UrZpVGRDg2Kcql5Eb+K99kjekTQ+0x7oFNj1wb8v1ZMpm/b
zzssOIiltccZPVodJ0ksrmBFhH1m1L8VwcE5nqAwMrJ2vump1OIUXLxZHcWYYpYX
nZ1DcvxZqz78AaULQV7UnUCr4Idbu16X3/06LrCVBYq9zTiQwo/ZgWx5NFsXJVH
DpLhr30sBQ4kiorsXULHxmnqA31snp3KDpt2WnJvFw9uCYNd0fg65zuSP/5Dw9Th
8zFv+ksSVVOXFEMJ7jL6j4LfcSB2weE1wdkqG2ze+fC259Gbg9p4PAHpg9UtB2D
0d15saQKhBAwUhpNMLCHVxDi/1zpnbvtRNW3C5G/luKpZ1V6rDeoaDsV/I+S9Szx
LvDsyje6vP0OVMocv9w1Y/iHvs/P2SBh2+Zhfyvt1/5v44cwrDYm1kdWDnsqHMUr
ESUMZjp96tJL1vK6GaM6AraGZSbtvhhjd5rZHZjd
-----END CERTIFICATE-----
```

```
Certificate: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support,
CN=ProxyCA
Serial Number: 11333611
Signature Algorithm: SHA256withRSA
Issuer: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support,
CN=ProxyCA
Validity: Wed Aug 01 16:22:37 UTC 2046
PEM:
```

```
-----BEGIN CERTIFICATE-----
MIIFxGCCA6ygAwIBAgIEAKzV6zANBgkqhkiG9w0BAQsFAADBzMRAdDgYDVQKDDAdQ
cm94eUNBMRAdDgYDVQKDLAdTdxBwb3J0MRMwEQYDVQKDApNb2JpbGVJcm9uMRYw
FAYDVQQHDA1Nb3VudGFpbWV3MRMwEQYDVQKDApDyWxpZm9ybmlhMQswCQYD
```

VQQGEwJVUzAeFw0xNjA4MDgxNjIyMzdaFw00NjA4MDExNjIyMzdaMHMxEDAObgNV  
BAMMB1Byb3h5Q0ExEDAObgNVBAsMB1N1cHBvcnQxEzARBgNVBAoMck1vYmlsZUly  
b24xZjAUBgNVBACMDU1vdW50YWluIFZpZCcxZzARBgNVBAgMCKNhbg1mb3JuaWEx  
CzAUBgNVBAYTA1VTMlIClIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAGeAgkD6  
rKQ5ASn1lV3zvhnTtGjDhQ3dGvtUA1I6S8jwCXbb7Ed4dK2zz50If16z6LDNGPsE  
0q3EdLVXuZeUA0nVkiThkd9hmXLrp911j1TmuibMh6oF34YJAuMzpIMLVuRcQjhK  
9k1b3mZsjSGeFGKWxgKJ+iAwz+Hp30PptUreQYgCF1cPQQMb2U5gGJm6zgN2sBt  
2tflTb6cDnurkriLfeqm+o7ppvmULMof9OJF7Lr8v18ozVArF60kaw0ywKex/1ZP  
exruAEey/6QPWW6LWYuakmEGIQ10R5WCk9p5Wv2+Y+pFTgcWmr0GmeHTFrfljSM  
FJzLzMSbq2gAJBU1tEGZaY449y+yuSgUqxTItOEWDQY/KDcvM9gny90Seo8h+8R  
fn9/B0a//wo4H/fnJX7OZMXI1kPtEg/6roVvsexzHkI1H9FPPrDW0g45Aps1/6g5n  
QxE3wCCiPHN2hmXgI33kREkXSL4pRTlbbdUW1+fkX2BYpxzY1LS6ZCXf+RvNFenH  
iWoogp6gUC1RzDKmwYMSpduWjireOP0MRJ8cckys8Bon+3/i7SPpAj7gfEYRb0BD  
lyx/T0EiufP/wuLQSwdk2sqrwVYjdqrfqFSIAGyI8dkzTZue/tqzPTYMk0xyKCVX  
Wly4v3PDRnm2G83qsC3r6ndK2ct52aInyOimBzcCAwEAAANgMF4wDwYDVR0TAQH/  
BAUwAwEB/zALBgNVHQ8EBAMCAAYwHwYDVR0jBBgwFoAUCqgSnfU4jXXhXPrDVER  
UIzpcKowHQYDVR0OBYYEFAqoEp31OI114Vz66w1RK1CM6XJKMA0GCSqGSIB3DQEB  
CwUAA4ICAQBbfw9G+5U5wHDPX8ZCN8Jy5fqPHXjtDjcXJi4wgq75om2EAp4nWRb8  
WmASZ8pz3ZA6JVM5QG1wS616bU5dxNMnu+snzueDdTDUZXV7aPtkn6QkNJeZqvMG  
UWLR3TYHXQIEViOrsxkh2oW+9j4XB1N9Kb6R8H02S2JDu8tdX4GQUi6xt9gA9IFM  
QakHGPDh0PslHauT2Gz7KUYzRqscqF2N1KQS6Z/VhTm3CEexOZNRhBIZM/1NMs2i  
VMjWUI+d0ouUiagcenPtM26hR9uuCkvwzNSVrhPljN1V1c1juKGo3K9VTbISXkmG  
hnW1B1QdPhpat1uDB49Lb7gZnIMDCcfzRlZhwgqJgFyTOeekJpMqtfJpU8s6Rbc  
B7EY6i3AGFc8dbtEZbZEL8HHKBObL0EUjHeWJtadGKaakT0Rh6Qgc0boDx3mwwBG  
lSI1J8/OqkQz4LYJuBWywYJu+BHCufuKdduqDEfzG83wwv1izRB5kZeWvSuu/+1  
dz4p7yAB7mWC/I1afqR7WRmURWWhVhZMQF3mr8wJZpL2MgbnF/z12cGgytNgw6L  
6/zj412DbSxouc6TrPtUtSK86Z+v4Ryi6waJGh/Fg1QQy8Ro+PMxT/gBvT7v3bwe  
P3NBgqk3ncF8RMsQhWjlCuPWZX0cgL11J/hs2e5+HURxzKInsQjl8Q==  
-----END CERTIFICATE-----



# Ivanti Tunnel for Android device user experience

The following provide some information on the device user experience for Ivanti Tunnel for Android:

- ["Ivanti Tunnel installation on devices" below](#)
- ["Controlling VPN traffic" on page 16](#)
- ["Troubleshooting" on page 84](#)

## Ivanti Tunnel installation on devices

The Ivanti Tunnel app is installed automatically on Android devices if the following conditions are true:

- The app configuration has the **Silently Install** option selected.
- The app is applied to a label that includes the Android devices.

If the **Silently Install** option is not selected, users can choose to install the app from the app catalog on the device.

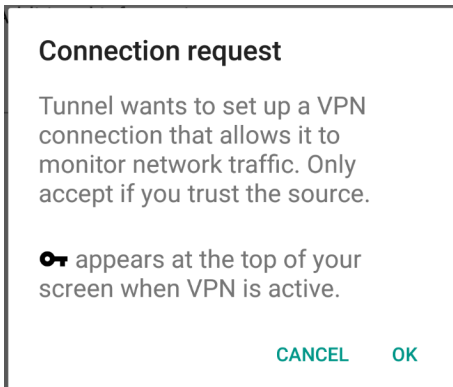
The following section proved some screen captures of what users see when they install Ivanti Tunnel:

- ["Accept Ivanti Tunnel connection \(Android native and enterprise only\)" below](#)
- ["Allow certificate \(Android native and enterprise only\)" on the next page](#)
- ["Ivanti Tunnel VPN connection" on page 83](#)

## Accept Ivanti Tunnel connection (Android native and enterprise only)

The first time that Ivanti Tunnel attempts to set up a VPN connection, device users are prompted to accept the Tunnel VPN connection. Device users must tap **OK** to continue using Ivanti Tunnel.

FIGURE 1. ACCEPT TUNNEL CONNECTION

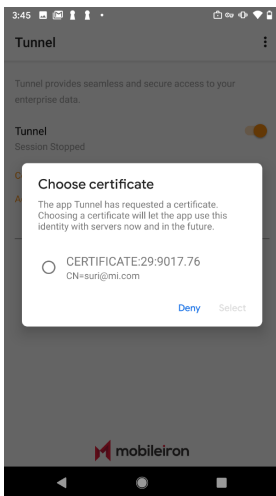


## Allow certificate (Android native and enterprise only)

During the installation, users are prompted to accept a certificate. The certificate is preselected.

Do not change the certificate selection.

FIGURE 2. ALLOW CERTIFICATE



Tap **Select** to install the certificate on the device and continue with the installation. Ivanti Tunnel uses this certificate to authenticate the device to Standalone Sentry.

On Android Enterprise devices, user interaction is not needed to accept the Ivanti Tunnel certificate for the following cases,

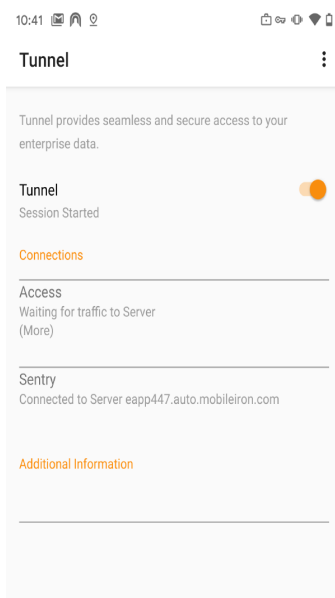
- Always On VPN (in Android Enterprise configuration on the UEM) and AutoBackgroundLaunch (Ivanti Tunnel for Android Enterprise configuration on the UEM) are enabled.
- The client version on the device is Go 72 for Android or Mobile@Work 10.8.0.0 for Android through the most recently released version as supported by Ivanti. In this case, the **Choose Certificate** dialog is not seen. The UEM client silently accepts the certificate. This change improves the user experience when the device is in shared-kiosk mode.

## Ivanti Tunnel VPN connection

The key icon on Android native and Android Enterprise devices, or the lock icon on Samsung Knox devices, indicates that the Ivanti Tunnel VPN configuration has been pushed and verified without any errors, and the VPN session has been established. This does not indicate if Tunnel is connected or not. The location and the icon can vary depending on the device and Android version.

The state of the Tunnel session after it is initiated remains as **Started**. With an Access deployment, the connection status changes from **Connected** to **Waiting** periodically if there is no Access traffic going through Ivanti Tunnel.

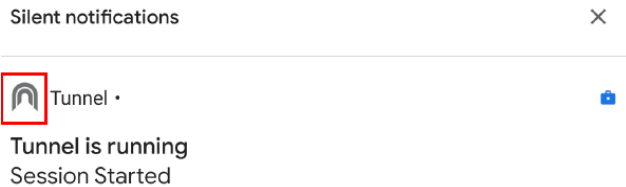
FIGURE 3. TUNNEL VPN CONNECTION STATE AND ICONS



## Ivanti Tunnel notifications icon

If Ivanti Tunnel notifications is enabled, users see the following icon if there are any notifications. On Android O through the latest version as supported by Ivanti, the ability to configure notifications in the Ivanti Tunnel app is not available. On these devices, configure notification in Android Settings.

FIGURE 4. TUNNEL NOTIFICATIONS ICON



The icon is not visible if there are no notifications.

## Troubleshooting

The Ivanti Tunnel app collects device and traffic logs to help with troubleshooting. The Ivanti Tunnel configurations from the UEM are also available to view in the app.

- ["Collecting log and PCAP files" below](#)
- ["Viewing logs" on page 86](#)
- ["Clearing logs" on page 88](#)
- ["Viewing Ivanti Tunnel configuration" on page 88](#)

## Collecting log and PCAP files

You can collect tunnel log and PCAP files to help with diagnostics and troubleshooting.

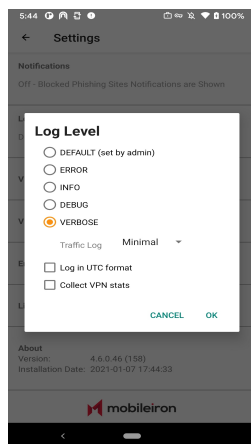
## Procedure

1. In the Ivanti Tunnel configuration (VPN configuration for Android native or app configuration for Android Enterprise) on your UEM (Ivanti EPMM or Ivanti Neurons for MDM), set the following:
  - AllowCapture to **true**
  - UINotificationLevel
  - DebugLog
  - TrafficVerboseLog
  - debugInfoRecipient
2. Force the device to check in.
3. In the Tunnel app go to **Settings** and select a log level.

If you select **Verbose**, you also have the option to select the **Traffic Log** level as **Off**, **Minimal**, or **ALL**. The traffic log level is disabled (**Off**) by default.

Due to the amount of data that is collected when you select Verbose, you will notice a decline in Tunnel performance.

FIGURE 1. TUNNEL LOG LEVEL



4. To collect PCAP files, under **Capture Traffic**, check **Enable**.

5. Tap device **Settings** > **Email Logs**.

The default email client for the device will be opened. The log files and PCAP files will be compressed and attached to an email.

### Related topics

For a description of the custom data, see "[Ivanti Tunnel Configuration Fields and Custom Data](#)" on [page 55](#).

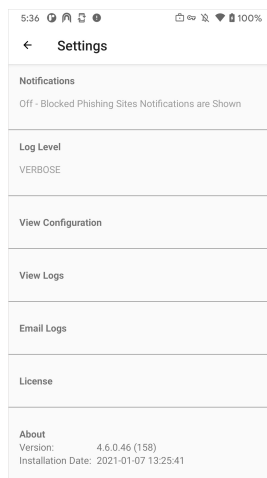
## Viewing logs

You can view logs on the Ivanti Tunnel app.

### Procedure

1. In the Ivanti Tunnel app, tap on the three vertical dots to expand the menu.

FIGURE 2. TUNNEL SETTINGS



## 2. Tap **Settings > Logs**.

FIGURE 3. TUNNEL CONFIGURATION



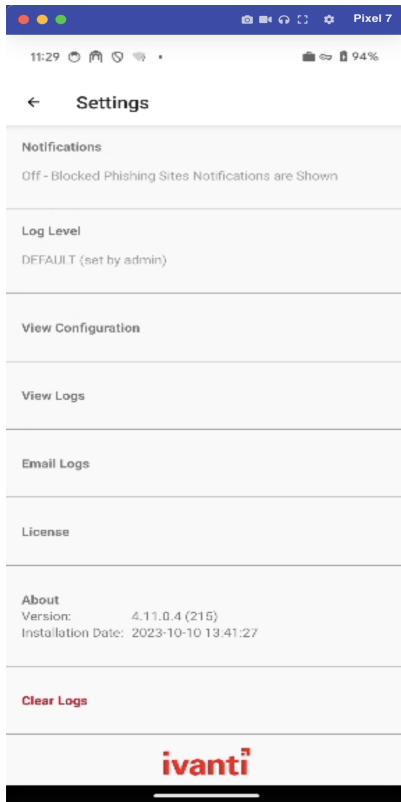
## 3. Tap the vertical three dots on the top right corner to expand the menu.



## 4. Tap an option from the menu to view a filtered set.

## Clearing logs

The **Clear Logs** option is now added to Tunnel for Android apps **Settings** section.



Using this option logs will be cleared for the following files:

- Tunnel Log file
- MTD Log File
- PCAP Log file

## Viewing Ivanti Tunnel configuration

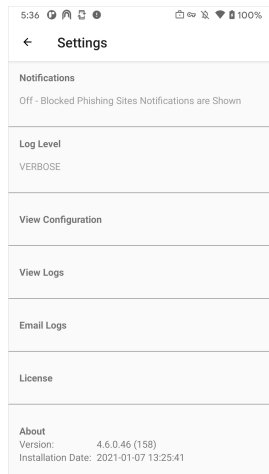
You can view Ivanti Tunnel configuration in the Tunnel app.



## Procedure

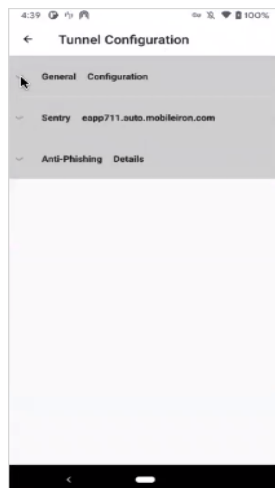
1. In the Ivanti Tunnel app, tap on the three vertical dots to expand the menu.

FIGURE 4. TUNNEL SETTINGS



2. Tap **Settings > View Tunnel Configuration**.

FIGURE 5. TUNNEL CONFIGURATION



The Ivanti Tunnel Configuration information is grouped under **General configuration**, **Access**, **Standalone Sentry**, and **Anti-phishing**. Expand the section to view configuration details.

## Anti-phishing enabled user experience

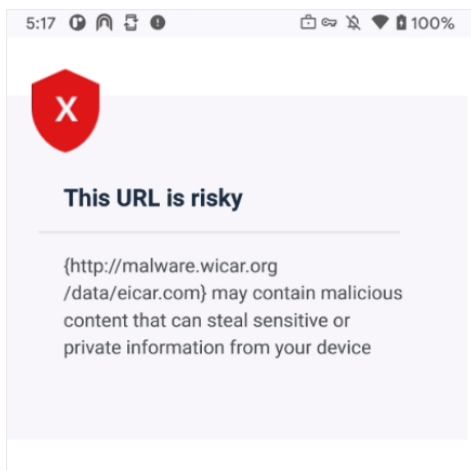
Anti-phishing support with Threat Defense provides an additional layer of protection to users. For more information see ["Support for anti-phishing with Mobile Threat Defense" on page 12.](#)

If the anti-phishing feature is configured, the user experience differs depending on whether the link URL is an HTTP URL or an HTTPS URL.

### HTTP phishing URL

When users tap on an unsafe HTTP link, instead of the link URL, users see a browser page notifying users that they are accessing an unsafe page.

FIGURE 1. HTTP PHISHING URL



### HTTPS phishing URL

When users tap on an unsafe HTTPS link, users see a heads up notification that they are accessing an unsafe page and the link URL is blocked. For subsequent requests, by default, the page continues to be blocked but the heads up notification is not seen.

FIGURE 2. HTTPS PHISHING URL



### Anti-phishing status

You can check whether anti-phishing is enabled or running by launching the Ivanti Tunnel app. If anti-phishing support for Android devices is not configured, the status for **Anti-phishing** displays as **Disabled**. If anti-phishing is configured and running, the status displays as **Running**. In some cases you may see the status as **Not running**. This may be due to a time lag for the configuration to propagate or there may be a misconfiguration.

FIGURE 3. ANTI-PHISING STATUS

