ZSCALER AND SUSE ENTERPRISE LINUX PODMAN DEPLOYMENT GUIDE

MARCH 2024, VERSION 1.0

BUSINESS DEVELOPMENT GUIDE

Contents

Terms and Acronyms

The following table defines acronyms used in this deployment guide. When applicable, a Request for Change (RFC) is included in the Definition column for your reference.

Acronym	Definition
CA	Central Authority (Zscaler)
CSV	Comma-Separated Values
DLP	Data Loss Prevention
DNS	Domain Name Service
DPD	Dead Peer Detection (RFC 3706)
GRE	Generic Routing Encapsulation (RFC2890)
ICMP	Internet Control Message Protocol
IdP	Identity Provider
IKE	Internet Key Exchange (RFC2409)
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security (RFC2411)
PFS	Perfect Forward Secrecy
PSK	Pre-Shared Key
SaaS	Software as a Service
SLES	SUSE Linux Enterprise Server (SUSE)
SSL	Secure Socket Layer (RFC6101)
TLS	Transport Layer Security
VDI	Virtual Desktop Infrastructure
XFF	X-Forwarded-For (RFC7239)
ZPC	Zscaler Posture Control (Zscaler)
ZDX	Zscaler Digital Experience (Zscaler)
ZIA	Zscaler Internet Access (Zscaler)
ZPA	Zscaler Private Access (Zscaler)

Trademark Notice

About This Document

The following sections describe the organizations and requirements of this deployment guide.

Zscaler Overview

Zscaler (NASDAQ: **ZS**) enables the world's leading organizations to securely transform their networks and applications for a mobile and cloud-first world. Its flagship Zscaler Internet Access (ZIA) and Zscaler Private Access (ZPA) services create fast, secure connections between users and applications, regardless of device, location, or network. Zscaler delivers its services 100% in the cloud and offers the simplicity, enhanced security, and improved user experience that traditional appliances or hybrid solutions can't match. Used in more than 185 countries, Zscaler operates a massive, global cloud security platform that protects thousands of enterprises and government agencies from cyberattacks and data loss. To learn more, see **Zscaler's website**.

SUSE Overview

SUSE is a global leader in innovative, reliable, and secure enterprise open source solutions, including SUSE Linux Enterprise, Rancher, and NeuVector. More than 60% of the Fortune 500 rely on SUSE to power their mission-critical workloads, enabling them to innovate everywhere—from the data center to the cloud, to the edge and beyond. SUSE puts the "open" back in open source, collaborating with partners and communities to give customers the agility to tackle innovation challenges today and the freedom to evolve their strategy and solutions tomorrow. To learn more, refer to **SUSE's website**.

Podman Overview

Podman is the default container management and orchestration tool on SUSE Linux Enterprise. In addition to providing a drop-in replacement for Docker Open Source Engine, Podman offers several advantages, including the ability to run containers in rootless mode. This allows regular users to deploy containers without elevated privileges. In other words, rootless mode means that you can deploy a container without becoming root or using sudo.

Audience

This guide is for network administrators, endpoint and IT administrators, and security analysts responsible for deploying, monitoring, and managing enterprise security systems. For additional product and company resources, see:

- **Zscaler Resources**
- **Podman Resources**
- **Appendix A: Requesting Zscaler Support**

Software Versions

This document was authored using the latest version of Zscaler software.

Request for Comments

- **For prospects and customers**: Zscaler values reader opinions and experiences. Contact **partner-doc-support@zscaler.com** to offer feedback or corrections for this guide.
- **For Zscaler employees**: Contact **z-bd-sa@zscaler.com** to reach the team that validated and authored the integrations in this document.

Zscaler and SUSE Introduction

Overviews of the Zscaler and SUSE applications are described in this section.

If you are using this guide to implement a solution at a government agency, some of the content might be different for your deployment. Efforts are made throughout the guide to note where government agencies might need different parameters or input. If you have questions, contact your Zscaler Account team.

ZIA Overview

ZIA is a secure internet and web gateway delivered as a service from the cloud. Think of ZIA as a secure internet on-ramp—just make Zscaler your next hop to the internet via one of the following methods:

- Setting up a tunnel (GRE or IPSec) to the closest Zscaler data center (for offices).
- Forwarding traffic via our lightweight Zscaler Client Connector or PAC file (for mobile employees).

No matter where users connect—a coffee shop in Milan, a hotel in Hong Kong, or a VDI instance in South Korea—they get identical protection. ZIA sits between your users and the internet and inspects every transaction inline across multiple security techniques (even within SSL).

You get full protection from web and internet threats. The Zscaler cloud platform supports Cloud Firewall, IPS, Sandboxing, DLP, and Isolation, allowing you to start with the services you need now and activate others as your needs grow.

ZPA Overview

ZPA is a cloud service that provides secure remote access to internal applications running on a cloud or data center using a Zero Trust framework. With ZPA, applications are never exposed to the internet, making them completely invisible to unauthorized users. The service enables the applications to connect to users via inside-out connectivity rather than extending the network to them.

ZPA provides a simple, secure, and effective way to access internal applications. Access is based on policies created by the IT administrator within the ZPA Admin Portal and hosted within the Zscaler cloud. On each user device, software called Zscaler Client Connector is installed. Zscaler Client Connector ensures the user's device posture and extends a secure microtunnel out to the Zscaler cloud when a user attempts to access an internal application.

Zscaler Resources

The following table contains links to Zscaler resources based on general topic areas.

Name	Definition
ZIA Help Portal	Help articles for ZIA.
ZPA Help Portal	Help articles for ZPA.
Zscaler Tools	Troubleshooting, security and analytics, and browser extensions that help Zscaler determine your security needs.
Zscaler Training and Certification	Training designed to help you maximize Zscaler products.
Submit a Zscaler Support Ticket	Zscaler Support portal for submitting requests and issues.

The following table contains links to Zscaler resources for government agencies.

Name	Definition
ZIA Help Portal	Help articles for ZIA.
ZPA Help Portal	Help articles for ZPA.
Zscaler Tools	Troubleshooting, security and analytics, and browser extensions that help Zscaler determine your security needs.
Zscaler Training and Certification	Training designed to help you maximize Zscaler products.
Submit a Zscaler Support Ticket	Zscaler Support portal for submitting requests and issues.

Podman for SUSE Enterprise Linux Overview

Podman is the default container management and orchestration tool on SUSE Linux Enterprise. In addition to providing a drop-in replacement for Docker Open Source Engine, Podman offers several advantages, including the ability to run containers in rootless mode. This allows regular users to deploy containers without elevated privileges. In other words, rootless mode means that you can deploy a container without becoming root or using sudo.

Podman Resources

The following table contains links to Podman support resources.

Name	Definition
Podman	SUSE Rootless Containers with Podman

Prerequisites

An environment variable named ZPA_PROVISION_KEY is required to run this image. You can retrieve the provisioning key from the **ZPA Admin Portal**. To learn more, see **About App Connector Provisioning Keys** and **Application Connector Deployment Prerequisites** (government agencies, see **About App Connector Provisioning Keys** and **Application Connector Deployment Prerequisites**).

Zscaler SUSE Podman is not currently supported for Kubernetes.

Only one container per host is allowed.

Deploying Podman Image on SUSE Enterprise Linux

To deploy a Podman image on SUSE Enterprise Linux:

1. Install Podman by entering the following command at a command line in SLES:

   ```
   sudo zypper in podman
   ```

   The Podman package is downloaded. The following is shown in the command line.

   ```
   Have a lot of fun...
   azureuser@PODMAN1:~> sudo zypper in podman
   Refreshing service ‘Basesystem_Module_x86_64’.
   Refreshing service ‘Containers_Module_x86_64’.
   Refreshing service ‘Desktop_Applications_Module_x86_64’.
   Refreshing service ‘Development_Tools_Module_x86_64’.
   Refreshing service ‘Public_Cloud_Module_x86_64’.
   Refreshing service ‘Python_3_Module_x86_64’.
   Refreshing service ‘SUSE_Linux_Enterprise_Server_x86_64’.
   Refreshing service ‘Server_Applications_Module_x86_64’.
   Refreshing service ‘Web_and_Scripting_Module_x86_64’.
   Building repository ‘SLE-Module-Basesystem15-SP5-Pool’ cache .............[done]
   Retrieving repository ‘SLE-Module-Basesystem15-SP5-Updates’ metadata .....[done]
   Building repository ‘SLE-Module-Basesystem15-SP5-Updates’ cache ..........[done]
   Building repository ‘SLE-Module-Containers15-SP5-Pool’ cache .............[done]
   Retrieving repository ‘SLE-Module-Containers15-SP5-Updates’ metadata .....[done]
   Building repository ‘SLE-Module-Containers15-SP5-Updates’ cache ..........[done]
   Building repository ‘SLE-Module-Desktop-Applications15-SP5-Pool’ cache ...[done]
   Retrieving repository ‘SLE-Module-Desktop-Applications15-SP5-Updates’ meta[done]
   Building repository ‘SLE-Module-Desktop-Applications15-SP5-Updates’ cache [done]
   Building repository ‘SLE-Module-DevTools15-SP5-Pool’ cache ...............[done]
   Retrieving repository ‘SLE-Module-DevTools15-SP5-Updates’ metadata .......[done]
   Building repository ‘SLE-Module-DevTools15-SP5-Updates’ cache ............[done]
   Building repository ‘SLE-Module-Public-Cloud15-SP5-Pool’ cache ...........[done]
   Retrieving repository ‘SLE-Module-Public-Cloud15-SP5-Updates’ metadata ...[done]
   Building repository ‘SLE-Module-Public-Cloud15-SP5-Updates’ cache ........[done]
   Building repository ‘SLE-Module-Python3-15-SP5-Pool’ cache ...............[done]
   Retrieving repository ‘SLE-Module-Python3-15-SP5-Updates’ metadata .......[done]
   Building repository ‘SLE-Module-Python3-15-SP5-Updates’ cache ............[done]
   Building repository ‘SLE-Product-SLES15-SP5-Pool’ cache ..................[done]
   Retrieving repository ‘SLE-Product-SLES15-SP5-Updates’ metadata ..........[done]
   ```

```
Building repository 'SLE-Product-SLES15-SP5-Updates' cache ...............[done]
Building repository 'SLE-Module-Server-Applications15-SP5-Pool' cache ....[done]
Retrieving repository 'SLE-Module-Server-Applications15-SP5-Updates' metad[done]
Building repository 'SLE-Module-Server-Applications15-SP5-Updates' cache .[done]
Building repository 'SLE-Module-Web-Scripting15-SP5-Pool' cache ..........[done]
Retrieving repository 'SLE-Module-Web-Scripting15-SP5-Updates' metadata ..[done]
Building repository 'SLE-Module-Web-Scripting15-SP5-Updates' cache .......[done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following recommended package was automatically selected: apparmor-abstractions
The following 11 NEW packages are going to be installed:
  aardvark-dns apparmor-abstractions conmon fuse-overlayfs libcontainers-common
  libcontainers-sles-mounts libfuse3-3 libslirp0 netavark podman slirp4netns

11 new packages to install.
Overall download size: 18.6 MiB. Already cached: 0 B. After the operation,
additional 64.9 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): y
```

2. Select `Y` at the prompt `Continue? [y/n/v/...? shows all options] (y):`
3. The Podman packages is installed. The following is shown in the command line.

```
Continue? [y/n/v/...? shows all options] (y): y
Retrieving: libfuse3-3-3.10.5-150400.1.7.x86_64
(SLE-Module-Basesystem15-SP5-Pool)
                                                          (1/11),  75.2 KiB
Retrieving: libfuse3-3-3.10.5-150400.1.7.x86_64.rpm ......................[done]
Retrieving: libslirp0-4.7.0+44-150500.2.1.x86_64
(SLE-Module-Basesystem15-SP5-Pool)
                                                          (2/11),  73.5 KiB
Retrieving: libslirp0-4.7.0+44-150500.2.1.x86_64.rpm ........[done (63.7 KiB/s)]
Retrieving: apparmor-abstractions-3.0.4-150500.11.9.1.noarch
(SLE-Module-Basesystem15-SP5-Updates)
                                                          (3/11),  90.9 KiB
Retrieving: apparmor-abstractions-3.0.4-150500.11.9.1.noarch.rpm .........[done]
Retrieving: libcontainers-sles-mounts-20230214-150500.4.6.1.noarch
(SLE-Module-Basesystem15-SP5-Updates)
                                                          (4/11), 353.7 KiB
Retrieving: libcontainers-sles-mounts-20230214-150500.4.6.1.n[done (31.7 KiB/s)]
Retrieving: libcontainers-common-20230214-150500.4.6.1.noarch
(SLE-Module-Basesystem15-SP5-Updates)
                                                           (5/11), 452.8 KiB
Retrieving: libcontainers-common-20230214-150500.4.6.1.noarch[done (63.7 KiB/s)]
Retrieving: aardvark-dns-1.10.0-150500.3.3.1.x86_64
(SLE-Module-Containers15-SP5-Updates)
                                                           (6/11), 784.3 KiB
Retrieving: aardvark-dns-1.10.0-150500.3.3.1.x86_64.rpm ....[done (415.7 KiB/s)]
Retrieving: conmon-2.1.10-150500.9.9.1.x86_64
(SLE-Module-Containers15-SP5-Updates)
                                                           (7/11),  51.2 KiB
```

```
Retrieving: conmon-2.1.10-150500.9.9.1.x86_64.rpm ........................[done]
Retrieving: netavark-1.10.2-150500.3.3.3.x86_64
(SLE-Module-Containers15-SP5-Updates)

                                                           (8/11),   3.1 MiB
Retrieving: netavark-1.10.2-150500.3.3.3.x86_64.rpm ......................[done]
Retrieving: fuse-overlayfs-1.1.2-3.9.1.x86_64 (SLE-Module-Containers15-SP5-Pool)
                                                           (9/11),  57.8 KiB
Retrieving: fuse-overlayfs-1.1.2-3.9.1.x86_64.rpm ........................[done]
Retrieving: slirp4netns-1.2.0-150500.1.1.x86_64
(SLE-Module-Containers15-SP5-Pool)
                                                          (10/11),  49.5 KiB
Retrieving: slirp4netns-1.2.0-150500.1.1.x86_64.rpm .........[done (49.5 KiB/s)]
Retrieving: podman-4.8.3-150500.3.6.1.x86_64
(SLE-Module-Containers15-SP5-Updates)
                                                          (11/11),  13.5 MiB
Retrieving: podman-4.8.3-150500.3.6.1.x86_64.rpm ............[done (13.0 MiB/s)]

Checking for file conflicts: ............................................[done]
( 1/11) Installing: libfuse3-3-3.10.5-150400.1.7.x86_64 ..................[done]
( 2/11) Installing: libslirp0-4.7.0+44-150500.2.1.x86_64 .................[done]
( 3/11) Installing: apparmor-abstractions-3.0.4-150500.11.9.1.noarch .....[done]
( 4/11) Installing: libcontainers-sles-mounts-20230214-150500.4.6.1.noarch[done]
( 5/11) Installing: libcontainers-common-20230214-150500.4.6.1.noarch ....[done]
( 6/11) Installing: aardvark-dns-1.10.0-150500.3.3.1.x86_64 ..............[done]
( 7/11) Installing: conmon-2.1.10-150500.9.9.1.x86_64 ....................[done]
( 8/11) Installing: netavark-1.10.2-150500.3.3.3.x86_64 ..................[done]
( 9/11) Installing: fuse-overlayfs-1.1.2-3.9.1.x86_64 ....................[done]
(10/11) Installing: slirp4netns-1.2.0-150500.1.1.x86_64 ..................[done]
(11/11) Installing: podman-4.8.3-150500.3.6.1.x86_64 .....................[done]
Running post-transaction scripts .........................................[done]
```

4. Create the Application Connector Container by entering the following command in the SLES command line:

```
sudo podman run \
  --name <Container Name> \
  -d \
  --cap-add cap_net_admin \
  --cap-add cap_net_bind_service \
  --cap-add cap_net_raw \
  --cap-add cap_sys_nice \
  --cap-add cap_sys_time \
 --cap-add cap_sys_resource \
  --restart always \
  --network host \
  -e ZPA_PROVISION_KEY=”3|api.private.zscaler.com|..” \
  docker.io/zscaler/zpa-connector:latest.amd64
```

5. For <Container Name>, enter a name for the container.

6. For "`3|api.private.zscaler.com|..`" enter the ZPA_PROVISION_KEY outlined in [Prerequisites](#).

   The following is shown in the command line.

   ```
   Trying to pull docker.io/zscaler/zpa-connector:latest.amd64...
   Getting image source signatures
   Copying blob 2d473b07cdd5 done   |
   Copying blob daadf42533ea done   |
   Copying blob 8fd946d71440 done   |
   Copying config a81d3abaf5 done   |
   Writing manifest to image destination
   a8ff00260fe1ab90038a0fc68778ca187ec78e0908259d1c78875babd157be3d
   ```

7. Verify the Application Connector Deployment by entering the following command in the command line in SLES,

   ```
   sudo podman ps
   ```

   The following is shown in the command line.

   ```
   CONTAINER ID  IMAGE                                         COMMAND     CREATED
   STATUS         PORTS       NAMES

   a8ff00260fe1  docker.io/zscaler/zpa-connector:latest.amd64             13 minutes ago
   Up 13 minutes              sappodman2
   ```

Managing Deployed Application Connector on SUSE Enterprise Linux Podman

To manage the application connector:

1. To upgrade the Application Connector Manager software, enter the following command in the command line in SLES:

   ```
     sudo podman pull docker.io/zscaler/zpa-connector:latest.amd64
   ```

2. To start and stop the Application Connector, enter the following command in the command line in SLES:

   ```
   sudo podman stop <container_name>

   sudo podman start <container_name>
   ```

3. To peview the Application Connector log, enter the following command in the command line in SLES:

   ```
   sudo podman logs -f <container_name>
   ```

Troubleshooting Application Connector on SUSE Enterprise Linux Podman

The following guidance provides troubleshooting information and guidelines about App Connectors:

- To learn more about App Connectors, see **About App Connectors** (government agencies, see **About App Connectors**).
- To configure App Connectors, see **Configuring App Connectors** (government agencies, see **Configuring App Connectors**).
- The guidance in this document is specific to Podman commands. For overall troubleshooting guidance, see **Troubleshooting App Connectors** (government agencies, see **Troubleshooting App Connectors**).

When troubleshooting App Connectors, consider the following.

App Connector Not Connected to the Cloud

To learn more, see **Not Connected to the Cloud** (government agencies, see **Not Connected to the Cloud**).

If the App Connector was previously working and now shows an error about not being connected to the cloud, you might see an error similar to the following:

```
Apr 02 08:03:21 zpa-connector zpa-connector-child[1737]: [1;39mReceived event from
[13.170.52.111]:38864;broker2.sha1.prod.zpath.net:[56.109.210.189]:2010: BEV_EVENT_
CONNECTED, sock=63 using ECDHE-RSA-AES128-GCM-SHA256[0m

Apr 02 08:03:22 zpa-connector zpa-connector-child[1737]: [1;39mReceived event from
[13.170.52.111]:38864;broker2.sha1.prod.zpath.net:[56.109.210.189]:2010: BEV_EVENT_
READING BEV_EVENT_ERROR, sock=63[0m

Apr 02 08:03:22 zpa-connector zpa-connector-child[1737]: [1;39m*Connector not yet con-
nected to cloud, please check connectivity, if this persists*[0m
```

You are likely to see this error if an App Connector is accidentally deleted from the ZPA Admin Portal. ZPA immediately recognizes that it should not communicate with the App Connector, and it breaks the active connection. There is no way to recover the App Connector.

To resolve this issue:

1. Remove the App Connector on the deployed platform. If you attempt to delete the App Connector and the App Connector still reports an ID number, then the App Connector was not entirely deleted.
2. Restart the process for configuring an App Connector and **deploying** (government agencies, see **deploying**) on a platform.
3. Restart the App Connector in Podman by entering the following command in the command line in SLES:

   ```
   sudo podman restart <container_name>
   ```

DNS Failure, But App Connector Successful

To learn more, see DNS Failure (government agencies, see DNS Failure).

If you have a DNS failure and the ZPA App Connector "root" process is successful, you might see an error similar to the following:

```
The domain (e.g., api.private.com) in the echo statement will depend on what ZPA cloud
you are on.
```

To learn more, see What Is My Cloud Name for ZPA?.

```
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: Checking Enrollment
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: Resolve read certificate-
failed
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: No valid certificate.
Attempting to enroll
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: Enroll: Connecting to api.
private.zscaler.com via co2br.prod.zpath.net.
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: DNS resolution failed for
api.private.zscaler.com
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: Could not create HTTP client
May 25 14:32:53 ussy-nwopapp03 zpa-connector-child[2040]: Certificate enrollment
failed.
May 25 14:32:55 ussy-nwopapp03 zpa-connector[3971]: zscaler-update: zpa-connec-
tor-child exited too fast, was running for 2 seconds
May 25 14:32:55 ussy-nwopapp03 zpa-connector[3971]: zscaler-update: zpa-connec-
tor-child failed too many times in a row- reverting to default software
```

1. At a command line in SLES, enter the following command:

   ```
   sudo podman inspect <container_name>
   ```

2. Inspect if the enable_dns is true on the network. This is not required if you are on the host network.

Get tcpdump

To learn more, see Cloud Connect TCP Dump (government agencies, see Cloud Connect TCP Dump).

Get the tcpdump and see the delay between the packet sent and the TCP RST received. If the delay is too small (e.g., 1 or 2 milliseconds), then most likely the firewall in your organization's premises is sending the RST.

1. At the command line in SLES, enter the following command:

   ```
   sudo podman exec -it <container_name> bash
   ```

2. At the #, enter:

   ```
   tcpdump -w tcpdump.out
   ```

Verify the Frequency of Error Occurrence

To learn more, see **Cloud Connect Error Frequency** (government agencies, see **Cloud Connect Error Frequency**).

This error sometimes occurs intermittently, and indicates a networking issue either in your organization's environment or in the internet path from the App Connector to the ZPA Public Service Edge. In those cases, do the following:

1. At a command line in SLES, enter the following command:

   ```
   sudo podman logs <container_name>  2>&1 | grep fohh_connection_connected
   ```

The following messages are printed every minute and inform you if the connection is always connected.

```
254]:443;4 uptime 1H:42M:5S, RTT (tcp|app): 68678|47784 us

    notice:Broker config override connection, state fohh_connection_connect-
ed, [10.1.0.5]:32840;broker20-1.dfw1.prod.zpath.net:[165.225.37.244]:443;4 uptime
1H:42M:1S, RTT (tcp|app): 70787|56675 us

    notice:Log(event_log) connection, state fohh_connection_connected,
[10.1.0.5]:60574;broker20-2.dfw1.prod.zpath.net:[165.225.37.245]:443;4 uptime
1H:42M:4S, RTT (tcp|app): 69486|48869 us, tx_b 592872, rx_b 148641

    notice:Log(inspection_log) connection, state fohh_connection_connect-
ed, [10.1.0.5]:37606;broker22-2.dfw1.prod.zpath.net:[165.225.37.253]:443;4 uptime
1H:42M:4S, RTT (tcp|app): 69118|48175 us, tx_b 135463, rx_b 135368

    notice:Log(adp_log) connection, state fohh_connection_connected,
[10.1.0.5]:44022;broker13-3.dfw1.prod.zpath.net:[165.225.217.238]:443;4 uptime
1H:42M:4S, RTT (tcp|app): 75832|54487 us, tx_b 135435, rx_b 135362

    notice:Log(stats_log) connection, state fohh_connection_connected,
[10.1.0.5]:35576;broker19-2.dfw1.prod.zpath.net:[165.225.37.241]:443;4 uptime
1H:41M:58S, RTT (tcp|app): 70084|51655 us

    notice:Broker control connection, state fohh_connection_connected,
[10.1.0.5]:47582;broker16-1.dfw1.prod.zpath.net:[165.225.217.248]:443;4 uptime
1H:41M:59S, RTT (tcp|app): 67918|47770 us

    notice:Broker config connection, state fohh_connection_connected,
[10.1.0.5]:43824;broker12.dfw1.prod.zpath.net:[165.225.33.
```

App Connector Upgrade Failure

To learn more, see Upgrade Failure (government agencies, see Upgrade Failure).

There are several different reasons for App Connector upgrade failures:

- Upgrade is in a failed state for more than 24 hours (government agencies, see failed state for more than 24 hours).
- Image cannot download since there is no disk space left (government agencies, see no disk space left).
- Image cannot download due to inconsistent connection between the App Connector and co2br (App Connector to Public Service Edge endpoint) (government agencies, see inconsistent connection).
- The Provisioning Key was deleted in the ZPA Admin Portal (government agencies, see Provisioning Key was deleted).

If none of these reasons are causing the upgrade failures, complete the following steps to try and resolve the upgrade failure:

1. Restart the App Connector. To learn more, see Start, Stop, or Restart an App Connector (government agencies, see Start, Stop, or Restart an App Connector).
2. At a command line in SLES, enter the following commands to stop or start the Application Connector.

   ```
   sudo podman stop | start | restart <container_name>
   ```

App Connector Pause State

To learn more, see Pause State (government agencies, see Pause State).

An App Connector can be in a paused state for one of the following reasons:

- `#define PAUSE_REASON_CERT_EXPIRY: Certificate expiry`
- `#define PAUSE_REASON_UPGRADE: Software upgrade`
- `#define PAUSE_REASON_ADMIN_PROBE_PROCESS_RESTART: Session process restart`
- `#define PAUSE_REASON_ADMIN_PROBE_SYSTEM_RESTART: Session system restart`

An App Connector provides the following log when it enters the pause state:

```
ASSISTANT_LOG(AL_INFO, "Connector(ID = %"PRId64") (Name = %s) entering PAUSE mode
because of - %s", global_assistant.gid, assistant_state_get_configured_name(),
assistant_state_get_pause_reason());
```

1. Check the `journalctl` logs for the reason the App Connector is in a pause state by using the following command:

   ```
   sudo podman logs <container_name>  2>&1 | grep "entering PAUSE mode because of"
   ```

Appendix A: Requesting Zscaler Support

If you need Zscaler Support to provision certain services or to help troubleshoot configuration and service issues, it is available 24/7/365.

To contact Zscaler Support:

1. Go to **Administration** > **Settings** > **Company Profile**.

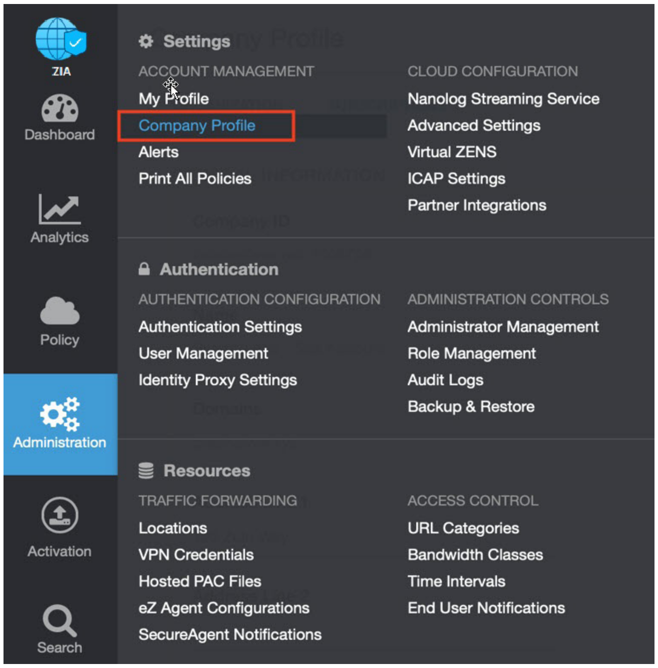
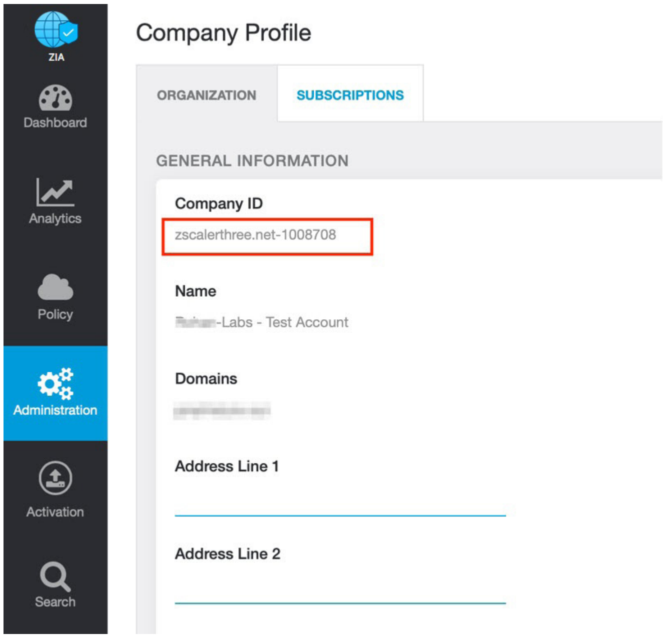


Figure 1. Collecting details to open support case with Zscaler TAC

2. Copy your **Company ID**.



Company ID
zscalerthree.net-1008708

Name
[redacted]-Labs - Test Account

Domains
[redacted]

Address Line 1

Address Line 2

Figure 2. Company ID

3. With your company ID information, you can open a support ticket. Go to **Dashboard** > **Support** > **Submit a Ticket**.

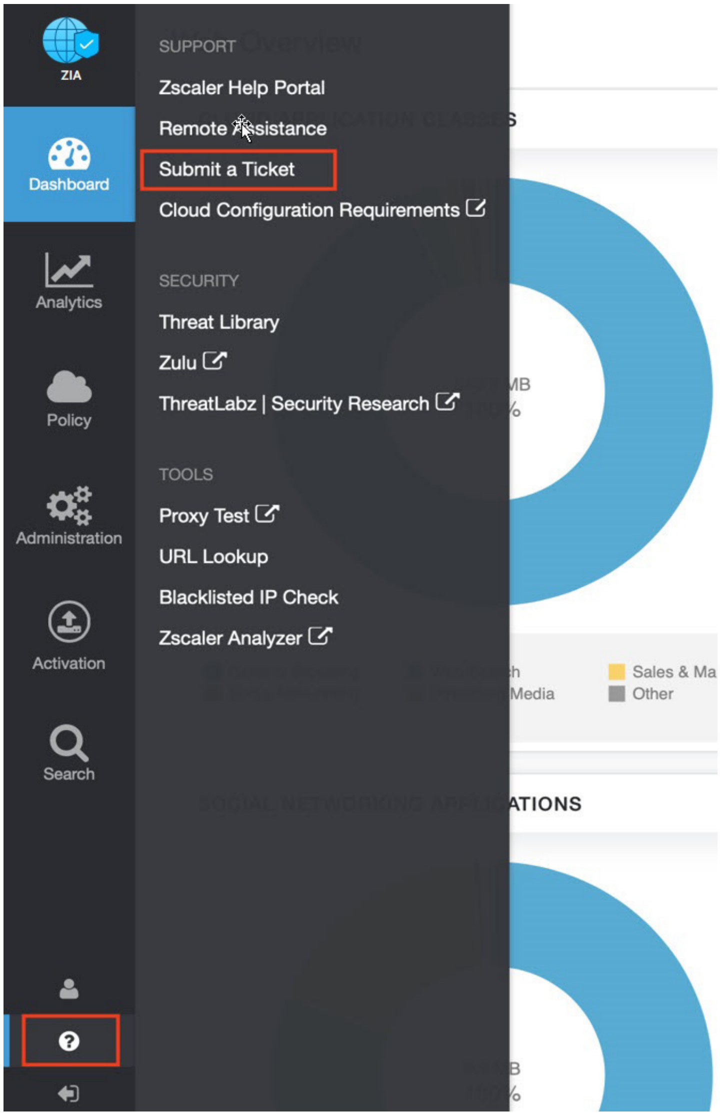


Figure 3. Submit a ticket