BeyondTrust

Privileged Identity 7.3
Deployment and Sizing Guide

Table of Contents

Deploy Privileged Identity 3
Plan for Deployment 4
Where Are the Target Systems Located? 5
What Accounts Will I Manage? 5
Should I Employ High Availability? 5
How Will I Deploy the Infrastructure? 6
Firewall Considerations 7
When to Use Zone Processors 8
Zone Processor Use Case Scenarios 10
Deploy Zone Processors 14
Zone Processor Support Files 17
Deployment Strategies 18
Single Server, No High Availability 18
Multi-System, No High Availability 18
Multi-System, Minimum High Availability 19
Multi-System, Full High Availability 20
Sizing Guidelines 21
Database Host Sizing 21
Management Console and Zone Processor Sizing 25
Web App and Service Host Sizing 27

Deploy Privileged Identity

Privileged Identity can be deployed centrally to manage one or more domains, whether trusted or untrusted. It also adds management capabilities to DMZ systems or offline machines across multiple platforms. The goal of management, in the context of Privileged Identity, is to:

- Gain control of privileged credentials and sessions
- Remove permanent administrative access
- Audit when access is granted, and to whom
- Audit what a user does with that access

Privileged Identity can perform thousands of management operations per minute from a single node, making it one of the fastest management platforms available, and ideal for incident response. By placing Privileged Identity at the center of your network, it can integrate with identity and access management products, governance products, security assessment products, and orchestration products, helping to provide automated incident response.

This guide describes some of the key concepts when deploying Privileged Identity, including database, zone processors, high availability, and more.

Plan for Deployment

Privileged Identity requires some basic installation prerequisites:

- Database
- IIS Web Server
- Service Accounts
    - Management console user
    - Deferred processor
    - Web application

There are also some advanced features:

- BeyondTrust Privileged Remote Access's SecureApp
- Session recording
- Zone processing

The database may be Microsoft SQL 2008 R2 or later, though we recommend using the most current version of Microsoft SQL Server. The IIS web server will be hosted on Windows Server 2012 R2 or later. A service account is required for the web application to access the database. The same service account can be used for the deferred processor or zone processors to perform scheduled jobs, though we recommend using a different service account.

Host systems require Microsoft .NET Framework version 4.5.2 or later. The management console host and deferred and zone processor hosts may also need Windows Management Framework v4 or later.

There are no permanent agents deployed with Privileged Identity, so network connectivity is required across a variety of ports depending on what is being managed.

For more information, please see the Privileged Identity Installation Guide at https://www.beyondtrust.com/docs/privileged-identity/install/index.htm.

When planning for a deployment, you need to answer six basic questions:

- What platforms will be managed?
- Where are those platforms physically and logically located?
- For Windows domains or AD-joined Linux/Unix hosts, are there trusts in place between the various domains?
- What accounts will be managed on those platforms, and what accounts will perform the management?
- How much high availability infrastructure will you use during this deployment and for what components?
- How will the infrastructure supporting Privileged Identity be deployed?

IMPORTANT!

Remain aware of evolving business needs as you plan for deployment, and adjust resources accordingly. If you need to move managed systems between management sets, you can do so without fear of losing any password information.

Where Are the Target Systems Located?

Where the targets systems are physically located, relative to the Privileged Identity hosts, can affect the deployment strategy. For example:

- An Amazon Web Services instance must be managed; however, no hosts are allowed direct connectivity to the internet. You must configure the use of a proxy server in order to manage the target AWS instance.
- A Linux machine that resides in a DMZ must be managed. You could deploy a zone processor (and install the cross-platform support library) into the DMZ, stand up another Privileged Identity instance in the DMZ, or open up specific firewall ports.
- If the systems are located across a high-speed WAN link, you might consider deploying a zone processor or managing them directly from the central location.

Logical separation is just as important as physical separation. In the context of Privileged Identity, logical separation refers to trusted versus untrusted systems. Trusted systems, Windows systems in particular, are very easy to manage from a central location with a single trusted account. Untrusted Windows systems can potentially be managed by the central instance of Privileged Identity if managing a single local account password, but when it comes to propagating the password to items like tasks, COM, and others, account impersonation becomes an issue and must be accounted for.

Determine where the systems are located, both physically and logically, to effectively design the infrastructure.

What Accounts Will I Manage?

It is important to consider what accounts will be managed and which accounts will perform the management.

In the Linux/Unix world, more so than the Windows world, there are many options for who will perform a management task and in what context. For example:

A low-powered account will log in and manage its own password. In this case, the account will issue **passwd**. In order to change its own password, it must:

- Have permission to change its own password
- Not be in violation of the minimum password age policy
- Not be in violation of the password history policy
- Not violate password requirements regarding length and complexity

A root account that performs a password change can change any user's password, by executing **passwd userName**. There is nothing else to consider in regards to a root account. In some cases, a root account can set passwords that do not comply with the configured password length; thus, complexity requirements and minimum age and history policies are not even considered.

This concept applies to all password changes across every platform: what account will log in and what account will be changed.

Should I Employ High Availability?

High availability (HA) should be employed whenever possible. All components of Privileged Identity support a highly available configuration. In most cases, an HA deployment is a function of the infrastructure the solution is installed on.

- **Database**: Install the database as a cluster, database availability group (SQL AlwaysOn), or mirror. Potentially replicate the database to an alternate location.
- **Web App**: Configure IIS hosts to be load-balanced using Microsoft load balancing or an external hardware load balancer.

- **Management Console**: Deploy multiple management consoles on multiple servers.
- **Deferred/Zone Processors**: Deploy multiple deferred or zone processors on multiple servers.

Virtualizing these host servers adds additional HA aspects such as virtual machine failover, hot migration, etc.

IMPORTANT!

High Availability is no replacement for a good disaster recovery strategy. Be sure to perform regular database and virtual machine backups, no matter what HA strategy is employed.

How Will I Deploy the Infrastructure?

How you deploy your infrastructure will be impacted by the physical and logical layout of the network. The general guidelines are:

- Choose the best location for your active database. Management consoles, web applications, web services, deferred processors, and zone processors must be able to communicate directly with this database. Typically clustered resources (cluster, AlwaysOn, mirror) are located on the same LAN. Databases may then be replicated to off-site databases.
- Web application and web service hosts should be kept logically close to the database host. The information sent from client to web app and web service or vice versa is relatively small compared to the work sent between the web app and web service host and database. It is best to ensure a fast and reliable connection between the database and web app and web service host, even when web application users are far away or over a slow link.
- Management consoles are typically installed on a central server and accessed via a remote desktop session (RDS). It is best to ensure a fast and reliable connection between the database and the primary management console, even when managers must use RDS over a slow link.
- Zone processors should be deployed to the same network as managed targets. This minimizes the number of firewall configurations required and keeps management traffic close to the managed targets.

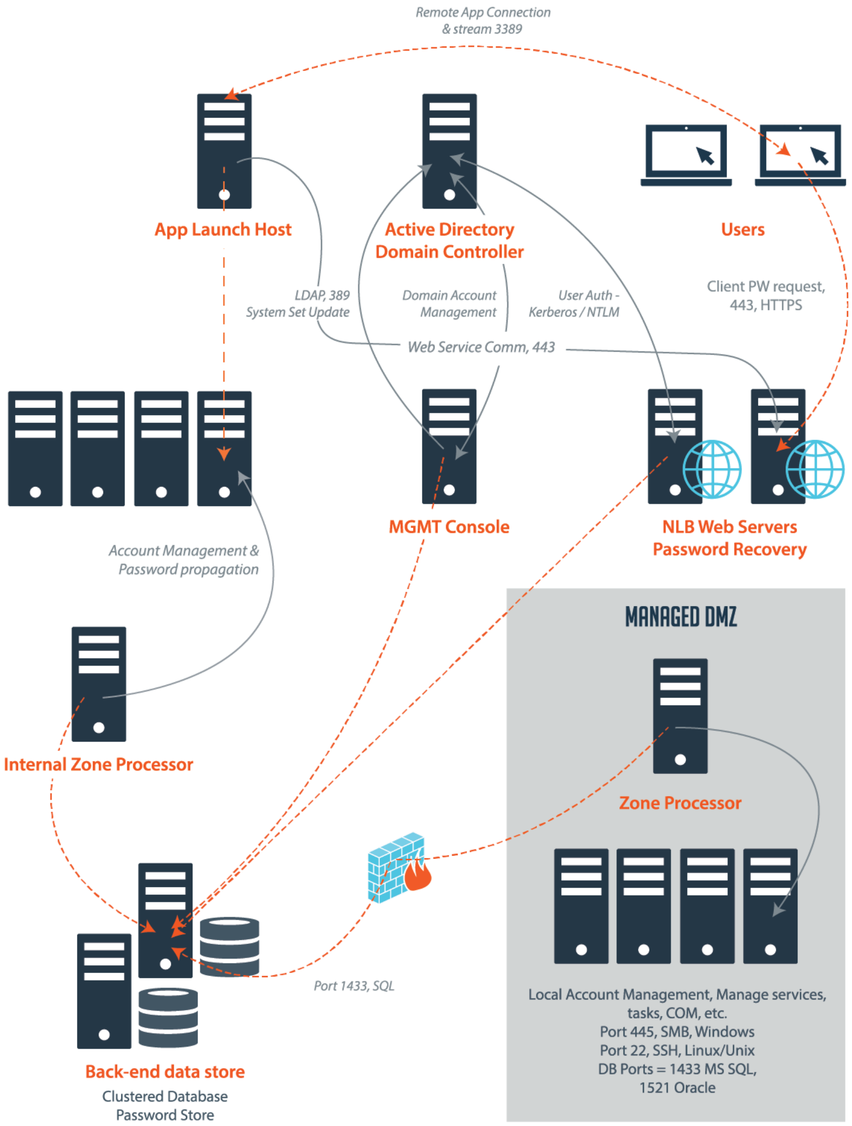
You must choose whether to deploy on physical or virtual systems, on-premises, or in the cloud. Most deployments occur on virtual machines located on-premises, though all of these scenarios, or combinations of these scenarios, are fully supported. There is no inherent difference to deploying on-premises versus in the cloud, and the basic connectivity and infrastructure requirements remain the same.

No component of Privileged Identity will work if the database is offline or inaccessible. Deferred and zone processors need constant connectivity to the database to maintain functionality. If a deferred processor attempts to start when the database is unavailable, it will fail to start and must be manually started. Web applications and web services will attempt a new connection to the database on each activation attempt.

Firewall Considerations

You must take your firewall into consideration when deploying Privileged Identity. There are many components in Privileged Identity that must speak with each other and with other systems. Here are some of the communications that occur:

- In the basic scenario, the management console, zone processors, deferred processors, web application, and web services all require communication with the central database, which listens on port 1433 by default.
- Management occurs from the management console or zone/deferred processors over a variety of ports depending on the management targets.
- Users connect to the web application and web services over HTTPS, port 443.
- If the user clicks an application launch link, they will be directed to a remote desktop session (RDS) using port 3389.
- The RDS server requires web service connectivity over HTTPS, port 443.
- The RDS server creates a connection from itself to the management target with a specific application, which may further operate on a variety of ports.



Port	Direction	Description
22	TCP, outbound, SSH	Used to manage SSH-based devices.
23	TCP, outbound, Telnet	Used to manage non-Windows devices that support Telnet.
25/465/587	TCP, outbound, SMTP	Used to send email. Only required if email notifications will be sent from Privileged Identity.

Port	Direction	Description
80/443	TCP, inbound, HTTP/S	Used to access the web application and web service.
88	TCP/UDP, outbound, Kerberos	Used by the jump server when authenticating with Kerberos.
135 & Ephemeral ports	TCP/UDP, outbound, RPC port mapper service	Used for most Windows COM/DCOM-based operations. The remote DCOM management port and ephemeral ports are typically provided by granting access to **DLLHOST.EXE** in the **%systemroot%\system32** directory. Ephemeral ports vary by target Windows operating systems. • COM/DCOM/MTS • Internet Information Services (IIS) • Scheduled Tasks (iTask interface) • SQL Server Reporting Services action account (SSRS) • SCOM RunAs accounts
161	TCP, outbound, SNMP	Used during system/network discovery operations and device management functions.
389/636	TCP, outbound, LDAP/LDAPS	Used for LDAP-compliant directories such as Active Directory.
443	TCP, outbound, HTTPS	Used for ESXi native management, as well as various cloud service providers and SAML/OAUTH authentication providers.
445	TCP, outbound, SMB	Used for Windows Server.
464	TCP/UDP, outbound, Kerberos	Used by the jump server when authenticating with Kerberos.
514	UDP, outbound, syslog	Used to communicate to logger systems such as ArcSight, QRadar, Splunk, syslog, etc.
623	UDP, outbound, IPMI	Used to manage lights-out devices such as Dell DRAC, HP iLO, etc.
1025	TCP, outbound, Teradata	Used to discover and manage Teradata databases.
1433	TCP, outbound, MS SQL Server	Used to connect product components to the Microsoft SQL Server data store.
1521	TCP, outbound, Oracle	Used to discover and manage Oracle databases.
3306	TCP, outbound, MySQL	Used to discover and manage MySQL databases.
3389	TCP, outbound and inbound, Remote Desktop Protocol (RDP)	Used for remote connections to target servers (automatic sessions) as well as inbound to the application launch server.
Port 5000	TCP, outbound, Sybase	Used to discover and manage Sybase ASE databases.
Port 5432	TCP, outbound, PostgreSQL	Used to discover and manage PostgreSQL databases.
Port 50000	TCP, outbound, DB2	Used to discover IBM DB2 databases.

When to Use Zone Processors

Privileged Identity can perform many different types of work, such as system discovery, account discovery, password rotation and propagation, and more. You can perform this work interactively (from the management console), or you can schedule this work to be performed at a specific time. Scheduling work for Privileged Identity to perform creates a job.

Running scheduled jobs requires a service to be present to run these jobs. The zone processor is employed to handle not only multiple network segments and divisions but also specific job types.

In this section, we'll identify multiple cases, as well as consider design and purchase decisions regarding zone processors.

In terms of code, there is no difference between a zone processor and a deferred processor. However, functionally, the difference is significant:

- A deferred processor handles all job types for all systems in all management sets.
- A zone processor handles specific job types for systems in one or more specific management sets.

The deferred processor does not account for additional zone processor assignments. As such, the deferred processor will run all jobs against all systems in all management sets.

Assign a zone processor to at least one specific management set and at least one specific job type, for example, password rotation. The zone processor will run only that specific job type against that specific set of systems defined in that specific management set. A zone processor will never try to manage anything else.

***Note:** Zone processors are a licensed feature of Privileged Identity.*

Zone Processor Use Case Scenarios

A deferred processor is deployed with a thick client (Privileged Identity Management Console) whereas a zone processor can be deployed independently. A deferred processor and a zone processor are technically the same thing. However, a deferred processor is only limited by job types, and a zone processor can be limited by both management sets and job types. Because management sets define management zones that are often a segmented network, or a network with clear security boundaries, the deferred processor should be assigned only the job type **Reports**, or be disabled. You will need to deploy a zone processor to handle all other job types.

The following scenarios show when you should have zone processors and a deferred processor, as well as their configuration.

Scenario 1: All Access Everywhere #1

You have a well-connected network with high-speed, highly reliable links where:

- There are no well-defined internal security boundaries, or if there are boundaries, they will not be managed by this instance of Privileged Identity.

Recommendation:

One or more deferred processors will suit your needs.

Scenario 2: All Access Everywhere #2

You have a well-connected network with high-speed, highly reliable links where:

- There are no well-defined internal security boundaries, or if there are boundaries, they will not be managed by this instance of Privileged Identity.
- You simply want to improve the job processing throughput of the job scheduling system.

Recommendation:

Either multiple zone processors or multiple zone processors and a deferred processor will be sufficient. No special configuration is required.

> ***Note:*** *In Privileged Identity, a single job processor can handle only one job at a time. This can lead to other jobs getting backed up in the job queue until a job processor becomes available. If there are two processors, then two jobs may run simultaneously. This concept will scale linearly. More processors means more jobs at the same time.*

Scenario 3: All Access Everywhere #3

You have a well-connected network with high-speed, highly reliable links where:

- There are no well-defined internal security boundaries, or if there are boundaries, they will not be managed by this instance of Privileged Identity.
- You want to ensure that password change jobs do not interfere with account elevation jobs.

Note: *If you want to ensure that password change jobs do not interfere with account elevation jobs, you can install an additional zone processor to manage any management set, restricting it to account elevation jobs only.*

Recommendation:

- Deploy one or more zone processors to handle all job types and one or more zone processors to handle only account elevation jobs.

Scenario 4: WAN Links with All Access Everywhere #1

Your organization is setup as follows:

- Is divided into multiple geographical regions separated by WAN links where WAN traffic is NOT a concern.
- Has no well-defined internal security boundaries, or if there are boundaries, they will not be managed by this instance of Privileged Identity.

Recommendation:

Deploy one or more deferred processors to handle all job types and one or more zone processors to target specific management sets.

Scenario 5: WAN Links with All Access Everywhere #2

Your organization is setup as follows:

- Is divided into multiple geographical regions separated by WAN links, where WAN traffic IS a concern.

In this scenario, you are concerned about the amount of traffic that Privileged Identity sends over a WAN link from a central point. Hundreds of simultaneous connections from a single source can be problematic over slow or unreliable long distance links, or where there is a need not to send management traffic over the link.

At this point, you must determine the following:

- The number of regions or offices that require a zone processor
- If management traffic may be sent over a link

Recommendation:

- If you require that no management traffic goes over the WAN link, then each segment, or zone, including the zone where Privileged Identity is actually located, will need its own zone processor. Furthermore, you must have the default zone processor not run discovery, refresh, or management jobs. A management set is required for each zone where there is a zone processor.

Note: *A zone processor can do management set updates only if you do not include or exlude systems when you configure the management set. You can use the zone processor if its only job is reporting.*

- If you prefer a zone processor to handle a job locally on the segment, but it is all right for management traffic to traverse the WAN links, then configure zone processors in each of the zones and let the default deferred processor continue with its default configuration to manage anything anywhere. Note that this is not a system of preference, but a system of availability. If the default deferred processor is available before a zone processor is available, the default deferred processor will run the job over the WAN links, even if the zone processor is scheduled to run the job later.

Scenario 6: A Network with a DMZ

A DMZ (de-militarized zone) is a section of a network where traffic is explicitly cut off from the rest of the network. You can think in terms of the internal network (where you are), the DMZ (where the secured servers are), and the external network (the internet).

When you have a DMZ to manage, you have four choices for Privileged Identity configuration:

- **Not recommended:** Completely open the firewall to allow the Privileged Identity host full access into the DMZ from the internal network.
- **Not recommended:** Allow the Privileged Identity host to establish a VPN (private, on-demand connection) into the DMZ and have full access from the internal network.
- **Recommended:** Stand up a separate instance of Privileged Identity in the DMZ. This is an option, but since it at least doubles the MS Windows and MS SQL licensing requirements, plus the Privileged Identity management requirements, this is not the best choice. Creating a standalone instance of Privileged Identity is not a bad design decision, as it fully separates the infrastructure and exposure if Privileged Identity ever gets compromised internally or externally, but it is an option that rarely gains traction.
- **Preferred:** Install a zone processor in the DMZ to handle the DMZ systems. Allow that zone processor (known host) access through the firewall (one direction, one port, known host) to the Privileged Identity central database (known destination, known port). This is often the most accepted scenario when working with a DMZ, as it is familiar, comfortable, easy to manage, easy to understand, and easy to secure.

Building on the fourth option, as this is the recommended zone processor scenario, you should set up zone processors for each zone. In this scenario, there are only two zones: internal and DMZ.

You may allow the deferred processor to run but should turn off its ability to perform password changes or discoveries. It should be relegated to admin activity reports and management set updates only.

IMPORTANT!

*If the option **Attempt remote connection to targets found** is enabled in the management set criteria, the management set update of a deferred or zone processor located in a DMZ will fail, unless the option **Verify connectivity system as a criteria for inclusion in or exclusion from the set** is also checked.*

You need one zone processor for the internal network and one for the DMZ.

If neither the deferred processor nor the zone processor is configured as described, the following will occur:

- The default deferred processor may attempt to manage DMZ systems. This will fail, as the DMZ does not allow incoming connections from the internal network.
- Failure of the deferred processor to manage a DMZ system will cause the job to fail, causing avoidable alerts to be generated from Privileged Identity, which in turn require human response.

Scenario 7: The Network Has Trust Issues

Note: *This scenario applies to managing Windows workstations and servers only.*

In the Windows world, there is a concept of trust. Trust is what allows an identity from one domain to access a resource in another domain. If there is a trust in place and going in the correct direction (it is possible for A to trust B but for B not to trust A) and Privileged Identity is in on the correct side of that trust, then things are relatively easy. But if there is no trust or Privileged Identity is on the wrong side of that trust, then things are going to be more complex.

If there are trusts in place and Privileged Identity is on the correct side of the trust, then refer to the previously described scenarios.

If there are no trusts in place or Privileged Identity is on the incorrect side of the trust, you will likely need zone processors if you want to manage the whole network through a single instance of Privileged Identity. This will require further investigation. Specifically determine:

- Will you solely be dealing with password changes (domain or local), and not propagating those password changes to scheduled tasks, IIS, COM/DCOM or other remote COM related items?
    - If the answer to this question is “yes, the scope will be limited AND no, we will not propagate to any of those items,” zone processors may not be required. Alternate administrators or cached credentials could be used in this scenario so long as your environment falls under scenario 1, 2, or 3. If your environment falls under scenario 4, 5, or 6, a zone processor will likely be required.
    - If the answer to the question indicates that propagation may be required, then a zone processor will likely be required.
- If zone processors are a requirement, there will be no fewer than two zone processors (assuming only two domains). The actual number of zone processors required will be determined through further discovery of the total number of untrusted domains, DMZs, regions, etc., as defined in the prior scenarios.

Scenario 8: Clustered Services

Note: *This scenario applies to managing Windows servers only.*

Privileged Identity can propagate passwords. This means that once the password is changed for the account in question, Privileged Identity can also update all the other references for the account, such as those used by Windows services.

Windows allows for clustered services. Clustering is a means of ensuring that even if the service in question (such as email, database, or web) goes down on one server, there is a duplicate service on another machine that will continue to provide service to the customer.

Privileged Identity can propagate password changes to Windows clustered services. However, there are some considerations:

- The most important consideration is that starting with Windows Server 2008, the management of clustered services is no longer backward or forward compatible.
- Clustered services running on one version of a Windows Server. For example, Windows Server 2008, cannot be managed by a different Windows Server version, such as Windows Server 2012.
- This is a limitation imposed by Microsoft.

If you manage clustered services, you must consider whether the clustered services are hosted on a version of Windows that is exactly the same as the Windows OS that is running Privileged Identity.

If the operating systems are not the same, then zone processors will be required. This scenario, to guarantee there are no problems during password propagation, requires that all zones, including the zone where Privileged Identity is hosted, will have a zone processor. The default deferred processor, if left enabled, must be configured NOT to perform password management jobs, in order to guarantee that the correct zone processor with the correct OS requirements manages the services. If the default deferred processor performs password management jobs, it will cause a service outage or possible failover/destruction of the cluster.

Note: *The number of zone processor deployments planned impacts the database hardware requirement, as detailed in "Sizing Guidelines" on page 21.*

Deploy Zone Processors

Considering a zone processor license has been purchased and applied, management sets are used to define the lists of systems for which a zone processor will be responsible. Thus, proper planning of management sets is essential to proper deployment of zone processors.

Consider a network with only two segments: internal and DMZ. At a minimum, two management sets will be created, one for each zone. In turn, a zone processor will be deployed and assigned to each specific management set. In this way, when you create a job destined for an internal system, it is run by the internal zone processor. Similarly, when you create a job destined for a server in the DMZ, it is run by the zone processor in the DMZ.

Zone processors require direct connectivity to the database. This communication is unidirectional from a known source to a known destination over a known port. Specifically, the communication is initiating from the zone processor host to the central database over the SQL communications port.

At a Glance

When the zone processor feature is enabled, the **Zone Processors** button will be available in the **Stored Jobs** dialog, available by clicking the **Jobs** button in the management console.

Zone processors can be deployed by pushing the zone processors files and settings from the management console (by clicking **Install**) on the **Zone Processors** dialog, or by using the standalone installer, available in the in the **SupplementalInstallers** folder within the installation directory. The standalone installer must be configured for each zone you are deploying a zone processor to.

When installing a zone processor, prerequisites such as .NET framework requirements, Windows Management Framework requirements, and required database provider requirements are not verified. If the correct database provider is not present when the zone processor attempts to startup, the startup process will fail.

For more information, please see Host System Requirements at https://www.beyondtrust.com/docs/privileged-identity/install/prerequisites/host-system.htm.

Pushing a Zone Processor

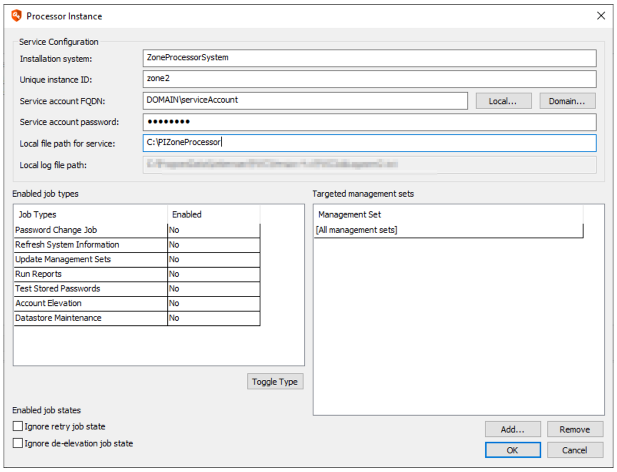
When pushing a zone processor, you need file system and remote registry access to the target host. If either of these is unavailable, the push will fail. When pushing a zone processor, the database configuration is identical to that currently configured for the management console.

1. In the management console, click the **Jobs** button.
2. On the **Stored Jobs** dialog, click **Zone Processors**.

***Note:** if you don't see the **Zone Processors** button, the feature is not enabled. A zone processor is enabled by the purchase and application of a license.*

3. Click **Install**.

4. Supply the following information:
   - **Installation system**: This is the name (simple, IP, or FQDN) for the zone processor host.
   - **Unique instance ID**: This is the instance ID of this zone processor. It must be unique on that system to avoid collisions with other zone processors hosted on the same system.
   - **Service account FQDN**: This is the qualified name of the account that will run the service. It must be an administrator of the target host and be granted **Logon as a Service**. If using integrated authentication to the database, this account must also have proper database access as defined in the Privileged Identity Installation Guide.



Processor Instance

Service Configuration

Installation system: ZoneProcessorSystem

Unique instance ID: zone2

Service account FQDN: DOMAIN\serviceAccount

Service account password: ••••••••

Local file path for service: C:\PIZoneProcessor

Local log file path: [redacted]

Enabled job types

Job Types	Enabled
Password Change Job	No
Refresh System Information	No
Update Management Sets	No
Run Reports	No
Test Stored Passwords	No
Account Elevation	No
Datastore Maintenance	No

Targeted management sets

Management Set
[All management sets]

Enabled job states

□ Ignore retry job state

□ Ignore de-elevation job state

For more information, please see the Privileged Identity Installation Guide at https://www.beyondtrust.com/docs/privileged-identity/install/index.htm.

   - **Local file path for service**: The physical location for the zone processor and its supporting files to be copied to.

***Note:** We do not recommend using the default file path.*

   - **Enabled job types**: Specify the types of jobs this zone processor will be allowed to perform.
   - **Management Sets**: Specify one or more management sets this zone processor will be responsible for managing.
5. Click **OK** to begin the process. The files will be copied and the registry configured, but you will need to start the service as a separate step.

Zone Processors Via Standalone Installer

When a zone processor cannot be automatically pushed, such as when dealing with an untrusted system or DMZ, use the zone processor standalone installer located in the **SupplementalInstallers** directory.

1. Launch **CreateZoneInstaller.exe**.
2. Supply the following information:
   - **Installer Template**: This value will already be configured.
   - **New Installer**: This is the new file that will be created and distributed to the target zone processor hosts.
   - **Job Log Path**: You can change the log file path for jobs if desired.
   - **Service Log Path**: You can change the log file path for the zone processor scheduling service if desired.
   - **Zone ID**: This is the instance ID of this zone processor. It must be unique on the system to avoid collisions with other zone processors hosted on the same system.
   - **Service Account Username**: This is the qualified name of the account that will run the service. It must be an administrator of the target host and be granted **Logon as a Service**. If using integrated authentication to the database, this account must also have proper database access as defined in the Privileged Identity Installation guide.

> **Note:** *If the zone processor is installed to a DMZ or an untrusted endpoint, it must still be an administrator of the target host and must be granted Logon as a Service, but a separate, explicit SQL Security Login account will need to be applied in the DataStore configuration. This explicit SQL Security Login account must have appropriate permissions to the application database.*

> *For more information, please see the Privileged Identity Installation Guide at https://www.beyondtrust.com/docs/privileged-identity/install/index.htm.*

- **Service Account Password**: The password for the service account. Click the **Encrypt** button to encrypt the password inside of the created installer package. If you don't encrypt the password, the password will be kept in clear text in the installer package.
- **Management Set Affinity**: Define one or more management sets to assign to the zone processor. If assigning more than one management set, separate management set names by a semicolon.
- **Job Affinity**: Define the job types this zone processor will run.
- **Database Settings**: If no settings are made, this installer will use the same database settings currently defined in the console, even if they might not work for this specific zone processor. Click the ellipses symbol (**…**) next to **Database Settings** to define custom settings for this zone processor to use when connecting to the database, such as changing the server name to an IP address or changing authentication to an explicit SQL account rather than integrated authentication. After making changes, select the option for **Use Customized DB Settings**.
- **Retry Options**: If no settings are made, the installer will use the same retry settings as currently defined for this management console. Click the ellipses symbol (**…**) to configure a different retry policy for this zone processor.

3. Click **Create**.
4. Copy the new MSI file to the target machine and install it.

Zone Processor Support Files

When deployed, zone processor support files will aid in managing Windows systems for password changes not involving propagation.

You'll find additional helper files in the SupplementalInstallers folder.

If you need any of the following, you must install **IntegrationComponents.msi** on the zone processor host:

- Password propagation
- Help desk ticketing integrations
- Event sink notifications
- Email

Help desk integration support also requires certain file system and registry information to be manually copied from the management console host.

If you need any of the following, you must install **CrossPlatformSupportLibrary.msi** on the zone processor host:

- Connecting to anything with SSH or Telnet
- Connecting to other non-Windows platforms

Deployment Strategies

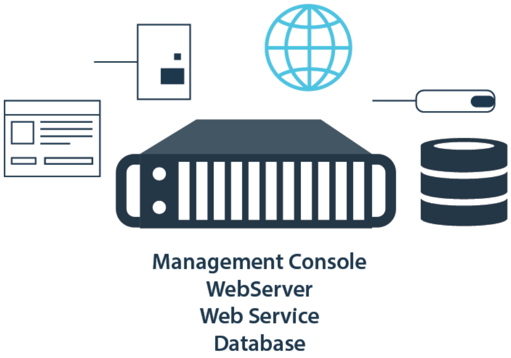
Here are some possible deployment strategies for Privileged Identity. Included are descriptions of bare minimum deployments through enterprise deployments.

Single Server, No High Availability

A bare-minimum installation uses a single server. The single machine hosts the database, management console, and web site service. The same system can also host the application launcher and session recording software. This system may be a virtual system or a physical system. If this is a production system, it requires at least two CPU cores and 4GB or more of RAM. The addition of Application Launcher to this host greatly increases CPU and RAM requirements.

Although not required, we recommend you install the database in its own instance (rather than a shared database instance) both for security and resource availability.

Backup is achieved by backing up the virtual machine or by backing up the database and encryption key. Virtual machine backup is achieved by using a solution appropriate to your virtual host. Database backup is performed by configuring a backup job using SQL Management Studio.

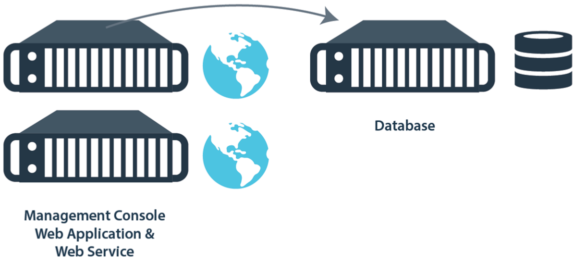


This deployment solution provides minimum scalability and no high availability. This solution is suitable for testing and for small environments, due to memory, storage, and high availability constraints. The single-server solution also poses the greatest security risk as the encryption key and the encrypted data are hosted on the same server.

Multi-System, No High Availability

This minimum-deployment method includes two systems: one for a database and a second for a web app and management console. We recommend that you install Application Launcher on a separate system, if applicable.

The database system hosts the MS SQL database, preferably in its own instance, not shared with other applications. This machine may be a physical system or a virtual system.



The management console, web app, and web service may be hosted by a single virtual system. This system should have at least two CPU cores and 2 GB of RAM. Hard drive space should provide for multiple gigabytes of free space for log file growth, as required by the management console and web application.

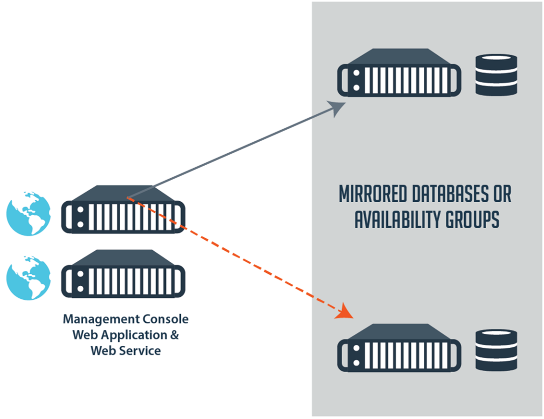
This deployment solution provides scalability to meet most medium to medium-large environments that are generally well connected. Backup is achieved by backing up the virtual machine or by backing up the database. Virtual machine backup is achieved by using a solution appropriate to your virtual host. Database backup is performed by configuring a backup job using SQL Management Studio.

Multi-System, Minimum High Availability

We recommend that medium deployments include no fewer than three systems: at least two servers for database high availability, and one for the management console, web app and web service. We recommend that application launching be deployed on a separate system, if applicable.

The database utilizes database availability groups, mirroring, or database clustering to provide for higher availability.

Availability groups, also known as AlwaysOn, require only two database servers but can leverage more. This deployment method also offers a readable secondary server, which makes it very easy to work with reporting services tied to the non-modifiable copy of the database. Availability groups also allow up to two synchronous replicas and two asynchronous replicas to be simultaneously active. We recommend using availability groups as the high-availability architecture for the Privileged Identity database.



Note: *The AlwaysOn availability groups concept was introduced with MS SQL 2012.*

Database mirroring requires the use of at least two database servers - one as a primary DB and any additional servers for failover. Database mirroring supports automatic and manual failover scenarios. At least two database systems are required for manual failover and three for automatic failover. In a manual failover, the database settings in the management console must be changed by hand, whereas in an automatic failover, the SQL Native Client and the witness server perform the failover.

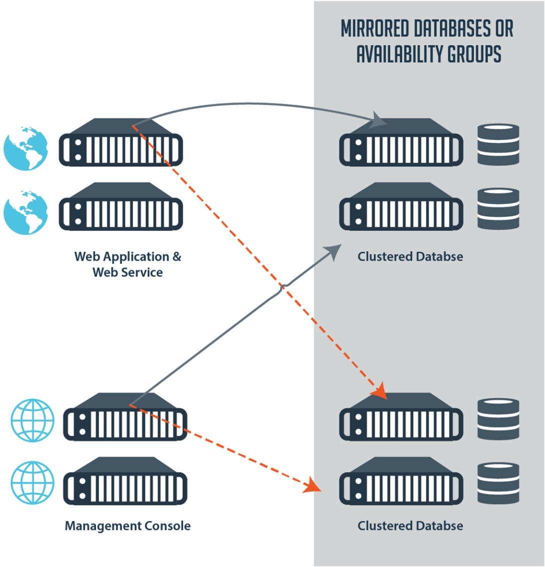
Note: *Mirroring functionality was deprecated in SQL Server 2012.*

Clustering may be used for the database in this type of deployment, but requires the use of shared hard drives and multiple network interfaces for each server.

Multi-System, Full High Availability

Large deployments will contain five or more servers:

- Two or more database servers
- One or two management console servers
- Two or more password retrieval web site servers


MIRRORED DATABASES OR AVAILABILITY GROUPS
Web Application & Web Service
Clustered Databse
Management Console
Clustered Databse

BeyondTrustrecommends installing Application Launcher on a separate system, if applicable. Additional transcoder/media servers should also be placed on separate servers.

While the provided image demonstrates one possible database configuration, there are actually multiple options for configuring the database:

- The two database servers can be configured as an availability group, without configuring the database as a failover cluster. If using an availability group, multiple replicas can be made available.
- The two database servers can be configured as a failover cluster. Additionally, you can mirror the clustered database to a single system or to another cluster of systems, which adds one or two more systems, respectively.

The Privileged Identity components are pointed to the active nodes in either scenario.

The two web servers are configured as a network load-balanced cluster, which may be either software-based or hardware-based. A network load-balanced cluster provides high availability to the data source, as well high-availability access to the data (stored passwords). This provides constant access to stored passwords, even if two or more servers should fail.

To deploy more than one management console in a highly available solution, deploy a secondary console on a separate machine. Loss of the management console results in the loss of the following abilities in a GUI:

- **Configure data store settings**
- **Configure email settings and alerts**
- **Configure Application Launch**
- **Configure event sinks**
- **Configure or change encryption key**
- **Create new password change jobs**
- **Scheduled jobs**: Will not run from the deferred processor, though zone processors that operate independently on secondary systems will continue to function.

In other words, the web app and web service communicate directly to the database, independent of the management console. The converse is also true, so disruption of one does not constitute disruption of the other.

Note: *The presence of a high-availability deployment does not negate the need for regular backups.*

Sizing Guidelines

In this section, we cover sizing for a variety of elements in Privileged Identity, including databases, host systems, hard disks, and more.

Database Host Sizing

The database is the most critical part of the Privileged Identity infrastructure. It contains the data for the entire program and is the only piece of the software which would truly cause a production issue if it were to become unavailable. Database sizing covers not only the physical sizing of the database files but also the physical requirements of the server itself.

Database CPU and RAM Resources

Unless the proper RAM and CPU requirements are met, the database platform will not even turn on. The minimum specifications for a production Privileged Identity database are 2 GB of RAM and two CPU cores, though we recommend at least 4 GB of RAM and 4 CPU cores. This is sufficient for hundreds of managed systems, thousands of stored passwords, and an otherwise basic installation which does not include any zone processors.

Proper sizing involves base lining to determine the usage profile of the database. The database involves large amounts of read and comparatively small amounts of write transactions following the initial population of the database. The amount of data (number of systems, jobs, etc.) as well as the number of zone processors affects the requirements of the database.

The amount of system data there is to read and write has an obvious effect on the need for database resources. However, the frequency at which these items are read and written, and the amount of transactional data the server must service, will affect the threading and queuing resources required during any given transaction. To offset this transactional requirement, an increase in the available thread and queue capability is required. This is in large part achieved by adding more CPU and RAM.

An implementation with zone processors requires more database resources than those without. A zone processor, just like a default deferred processor, queries the database every *N* seconds to check for work to do (*N* is equal to the sleep time value defined for the zone processor and has a default value of 6 seconds).

In other words, a zone processor will check the scheduling status of every job in the database to determine which jobs are past due and will then run the most out-of-date job. This information is calculated by running a query in the database every *N* seconds per zone processor. For example, an implementation with 10 zone processors and 200 jobs will require the database to evaluate all 200 jobs, 10 times every 6 seconds. Then, if any zone processor must run a job, it will read and write to the database, while the other zone processors continue to perform queries every 6 seconds, looking for work.

Each zone processor requires a minimum of an additional 512 MB RAM on the baseline database server. Each zone processor requires 1/5th of a CPU core.

Example 1

Server Infrastructure Requirements without Zone Processors Environment Description:

- System count: 1,000
- Stored managed passwords: 2,000
- Propagation enabled: yes
- Number of base jobs: 20
- Base job frequency: monthly
- Number of re-randomization jobs per month: 200 (password recoveries)

- Total number of password changes per year: (20 base x 12 months) + (200 re-randomizations x 12 months) = 2,640
- History enabled: yes
- Number of zones: 1 (default zone)

Recommended DB: 4 CPU cores, 4 GB RAM

Factors:

- Small number of base jobs = low CPU, low RAM
- Large number of recoveries resulting in re-randomization including propagation = add RAM
- Large number of recoveries, year over year = add CPU cores in time
- Single zone (no zone processors) = low CPU, low RAM.

In example 1, although there are a large number of monthly password recoveries that result in subsequent re-randomizations of the target account, overall CPU and RAM utilization remains low. The need for more CPU and RAM will be affected by the total number of systems, total number of jobs, and additional zone processors.

Example 2

Server Infrastructure Requirements with Zone Processors Environment Description:

- System count: 1,000
- Stored managed passwords: 2,000
- Propagation enabled: yes
- Number of base jobs: 20
- Base job frequency: monthly
- Number of re-randomization jobs per month: 200 (password recoveries)
- Total number of password changes per year: (20 base x 12 months) + (200 re-randomizations x 12 months) = 2,640
- History enabled: yes
- Number of zones: 10 (default zones)

Recommended DB: 4-8 CPU cores, 9-10 GB RAM

Factors:

- Small number of base jobs = low CPU, low RAM
- Large number of recoveries resulting in re-randomization including propagation = add RAM
- Large number of recoveries, year over year = add CPU cores in time
- Multiple zone processors = add CPU, add RAM.
- Baseline = 4 CPUs
- Zone processor requirement = 10 zone processors x 1/5th CPU = +2 cores
- Baseline = 4 GB RAM
- Zone processor requirement = 10 zone processors x 512 MB = +5 GB

Database Storage Sizing

The following variables are used to calculate the size of the database:

- Number of systems
- Number of accounts
- Number of passwords
- Number of groups (system lists)
- Amount of recoveries and other web operations
- Number of delegations
- Number of password jobs
- Stored password history
- Propagation turned on or off
- Job failures
- Job logging

With the exception of the secure file store, Privileged Identity stores only text in the DB fields, as opposed to storing binary data. Following are observed numbers from a Privileged Identity database running SQL 2008 x64 Enterprise Edition.

Initial Sizing

- 4 MB (.92 MB Free) default formatted DB, with one dynamic group and one active directory query

Managed Items

- System added to group (no information) = .0006 MB per system
- System information and account usage = .005 MB per system
- System added to a job = .0006 MB per system
- Stored Password with job settings and first-time discovery information = .045 MB per new password
- Job re-runs, stored passwords = .0017 MB per password history

Example: *Given a system list with 1,000 systems, that change one account on each of those systems via one job, and in which the accounts are changed successfully on the first attempt, the initial database would be roughly:*

Initial size: 4 MB
+ (.0006 MB x 1,000 Windows systems)
+ (.005 MB x 1,000 optional initial discovery)
+ (.0006 MB x 1,000 systems in the job)
+ (.045 MB x 1,000 password change jobs)
Total: *55.2 MB*

***Note:** If the optional discovery (which has no effect on jobs with propagation settings) is not performed, the size of the initial database is set to 55.2MB.*

Password History

Password histories require approximately .0017 MB.

***Example:** 1,000 historical passwords, at 15 characters each, require an additional 1.7 MB. After the initial password change job, there are an additional 11 password change jobs which are added to the password history.*

*1.7 MB x 11 password changes = **18.7 MB***

***Note:** No job info is stored with historic passwords.*

Logging

Logging requirements vary, based on operations performed versus success versus failure rates. Logging consumes approximately **20 MB** of space over the course of a year, assuming no failures are logged. The space consumed by logging grows linearly.

First Year

After one year for 1,000 Windows systems whose passwords change 12 times per year without propagation, the estimated size of the database is:

55.2 MB + 18.7 MB + 20 MB = **93.9MB**

Annual Growth

Assuming no other changes, the database is expected to grow at the following rate:

1,000 systems x .0017 password history x 12 months + 1,000 systems x .0017 new password change jobs + 20 MB logs = **42.1MB per year**

Actual Planned Size

The size of the database will likely be much larger due to the factors mentioned at the beginning of this section. In particular, job logging, web audit information, and account usage information will impact the physical size of the database the most. Once the base measurement is determined using the above math, for actual capacity planning, triple the minimum size of the database:

93.9 MB x 3 = **281.7MB for the first year**

42.1 MB growth x 3 = **126.3MB projected actual growth per year**

Management Console and Zone Processor Sizing

CPU and RAM

The management console and zone processors consume a small amount of RAM and CPU cycles when idle. At startup, the management console and scheduling service consume approximately 15-20 MB of RAM, with CPU utilization near 0%, and resources allocated as they work.

Both components are multi-threaded and can take advantage of as many CPU cores as are available when performing work (e.g. password rotation, discovery, etc.).

> ***Note:*** *Because Privileged Identity is a 32-bit application, no more than 2 GB of RAM will ever be available to each application.*

During an operation, CPU utilization for either process is capped at 95% of available CPU resources, to ensure the system does not become unresponsive. Similarly, threading is throttled when total CPU resources reach 95%. No additional threads are launched until CPU utilization falls below 95%. When threads cannot be launched, total job performance will suffer. This condition may be avoided by the addition of additional CPU cores.

To help maintain RAM resources, plan your server's RAM capacity to include:

- 2 GB RAM for the host OS
- Additional RAM for other applications like anti-virus
- 2 GB RAM for each Privileged Identity component installed on the host

To help maintain CPU resources, plan your server's RAM capacity to include:

- 2 CPU cores for the host OS
- CPU requirements for other applications like anti-virus
- 2 CPU cores for each Privileged Identity application component installed on the host

The minimum recommendation for a host server is 4 CPU cores and 4 GB of RAM. During operations, additional CPU cores will improve overall job processing performance during multi-threaded operations.

The system also has a number of single-threaded operations that occur, where adding more CPU cores will not improve performance. Specifically:

- Console operations, such as changing management sets or opening the stored jobs dialog, where the data for a dialog is read from the database and rendered on-screen. Improving the database and console performance, however, will improve the initial load and ensure a fast CPU, whereas sufficient RAM on the host will improve dialog load time.
- Individual system operations. A job with 500 systems will dispatch 100 threads by default, to manage 100 systems. The remaining 400 will remain pending until there is at least a single available thread. During these operations, there may be 50 operations to perform against the host, such as getting system information, reading service target propagation info, and performing the propagation. These operations will be performed in a single-threaded, linear fashion against any particular host.

Logging

A number of logs are enabled by default for the following items:

- **Management Console Operations**: Logs every operation performed in the management console.
- **Deferred Processor Service Log**: Logs scheduling service activity such as when jobs are launched or finished.

- **Job Logs**: Scheduled jobs create job log files named after each job ID. Each successive job run appends to the job log.

Job log archiving is also enabled by default with a default setting of 1 MB. When a component starts, the size of the file is checked. If it exceeds the archive threshold, the log is moved to the log archival location and compressed, and a new log file is created. With this in mind, an active log can still exceed the 1 MB threshold.

An exact size recommendation for log storage is not possible, though most customers rarely exceed 4 GB in total logging that spans years of activity.

Web App and Service Host Sizing

CPU and RAM

The web application runs via IIS and leverages a COM application for database access and related functionality. These COM applications (represented by **dllhost.exe**) consume a small amount of RAM and CPU cycles when idle, and shut down automatically when not in use (per COM default settings). At startup, the COM application consumes approximately 15-20 MB of RAM, with CPU utilization near 0%, and resources allocated as they work.

> ***Note:*** *Because Privileged Identity is a 32-bit application, no more than 2 GB of RAM will ever be available to each application.*

To help maintain RAM resources, plan your server's RAM capacity to include:

- 2 GB RAM for the host OS
- RAM for other applications like anti-virus
- 2 GB RAM for each Privileged Identity application component installed on the host

To help maintain CPU resources, plan your server's RAM capacity to include:

- 2 CPU cores for the host OS
- CPU requirements for other applications like anti-virus
- 2 CPU cores for each Privileged Identity application component installed on the host

Our minimum recommendation for a host server is 4 CPU cores and 4 GB of RAM.

Logging

One log is enabled by default for the web application or web service:

- **RouletteAppServiceSupport**: Logs web service basic activity and errors.
- **RouletteWebApplication**: Logs web application basic activity and errors.

Job log archiving is also enabled by default with a default setting of 1 MB. When a component starts, the size of the file is checked. If it exceeds the archive threshold, the log is moved to the log archival location and compressed, and a new log file is created. With this in mind, an active log can still exceed the 1 MB threshold.

An exact size recommendation for log storage is not possible, though most customers rarely exceed 4 GB in total logging that spans years of activity.