Firewall and Port Requirements for Zenoss® 4.2 Deployments

Applies To

- Zenoss Resource Manager 4.2.X

Tested On

- Zenoss Resource Manager 4.2.3

Summary

This article details the set of network ports and protocols that must be enabled for a Zenoss instance to function properly. The exact requirements for a specific Zenoss installation depend on how its components are distributed and/or replicated from the Zenoss master server, and what classes of devices are being monitored. In most cases, the port numbers used by Zenoss daemons are set by their configuration file(s) in `$ZENHOME/etc`. Some monitoring templates use a configuration property to specify the target port on monitored devices.

In the default, single-server installation, all communication between Zenoss components is through the loopback network (lo) and local ("unix") sockets.

This document only addresses those ports used by a standard Zenoss Service Dynamics Resource Management installation with the Windows Monitoring ZenPack installed. Impact and Analytics port requirements will be documented in separate articles. For ZenPacks not included in the standard installation, consult the *Resource Management Extended Monitoring* guide or the individual ZenPack's documentation.

Browser Connections to the Zenoss Web Server

By default, the Zenoss web server load balancer (nginx) listens on TCP port 8080, but can be configured to accept requests on the default HTTP port (80), the HTTPS port (443), or an arbitrary port.

Zenoss Administrator's / Operator's Browser to Zenoss UI					
Description	**Port**	**Proto**	**Direction**	**Source / Destination**	**Notes**
Web UI	8080*	TCP	OUT	nginx	*Port number is configurable.
Google Maps portlet (Dashboard)	80 (HTTP)	TCP	OUT	callhome.zenoss.com, *Various Google sites*	This portlet can be removed.
Default site portlet (Dashboard)	80 (HTPP)	TCP	OUT	www2.zenoss.com	This portlet can be removed or reconfigured to access another website.

Port Requirements for Prerequisite Software

Zenoss Resource Manager depends on various third-party software packages (see the *Resource Manager Installation* document for a complete list). Some of this software includes daemons that must be accessible either locally, over the network, or both. By default, all of the Zenoss software and its dependencies are initially installed on the Zenoss master server. For increased performance, some of the third-party daemons can be run on a separate dedicated (physical or virtual) server, noted as "off-host" in the table below.

Port Requirements for Prerequisite Software

Daemon	Port	Proto	Dir	Source / Destination	Notes
(*bind*)	53 (DNS)	UDP	OUT	*DNS Server(s)*	Very strongly recommended.
ntpd	123 (NTP)	UDP	IN/OUT	*Zenoss servers, network time servers*	Strongly recommended for Zenoss instances with more than one server. Should be run on all Zenoss servers.
memcached	11211 (memcache)	TCP	IN	zenactiond, zeneventserver, zenhub, zenjobs, zope	Required on the master server and every remote hub.
rabbitmq	5672 (AMQP)	TCP	IN	zenactiond, zencatalogservice, zeneventd, zeneventserver, zenhub, zenjobs, zope	One RabbitMQ instance (or cluster) is required for each Zenoss instance. Can be run off-host.
sshd	22 (SSH)	TCP	IN	*ssh client, Zenoss master server*	Inbound access is required on all Zenoss servers.
(*various*)	22 (SSH)	TCP	OUT	*Zenoss master server*	Outbound SSH access is required on the master server.
zends (mysql)	`/var/lib/zends/zends.sock` 13306	unix TCP	— IN	zenactiond, zencatalogservice, zeneventd, zeneventserver, zenhub, zenjobs, zope	One instance is required. The object database and event database can be located on separated instances. Local connections use the unix socket. Remote connections conventionally go to TCP port 13306. Can be run off-host.

Zenoss Event Server Daemons

The Zenoss daemons listed below are required. They are normally run on the Zenoss master server. As noted below, some of them can be moved to a different dedicated (physical or virtual) server ("off-host"), and others may be run ("replicated") on additional servers but still require an instance running on the master server.

Note that by default, the master server also runs the Hub and Collector daemons detailed in subsequent sections.

Zenoss Event Server Daemons					
Daemon	**Port**	**Proto**	**Dir**	**Source / Destination**	**Notes**
nginx	8080 (HTTP) 8090 (HTTP) 8091 (HTTP) 9081[1] (HTTP)	TCP TCP TCP TCP	IN OUT OUT OUT	(*browser*) zenhub zenrender zope	Can be replicated (with zope) on additional servers to off-load report generation.
zenactiond	25 (SMTP) 5672 (AMQP) 11211 (memcache) 13306[2]	TCP TCP TCP TCP	OUT OUT OUT OUT	(SMTP daemon) rabbitmq memcached zends	
zencatalogservice	8085 13306[2]	TCP TCP	IN OUT	zope, zenjobs, zenhub zends	
zeneventd	5672 (AMQP) 11211 (memcache) 13306	TCP TCP TCP	OUT OUT OUT	rabbitmq memcached zends	Can be moved or replicated off-host.
zeneventserver	8084 5672 (AMQP) 13306[2]	TCP TCP TCP	IN OUT OUT	zope, zenhub rabbitmq zends	Can be moved off-host.
zenjobs	5672 (AMQP) 8085 13306[2]	TCP TCP TCP	OUT OUT OUT	rabbitmq zencatalogservice zends	
zenjserver	8700	TCP	IN	zope	
zentune	5672 (AMQP) 8084 11211 (memcache) 13306[2]	TCP TCP TCP TCP	OUT OUT OUT OUT	rabbitmq zeneventserver memcached zends	
zope (runzope)	9081[1] 25 (SMTP) 5672 (AMQP) 8084 8700 11211 (memcache) 13306[2]	TCP TCP TCP TCP TCP TCP TCP	IN OUT OUT OUT OUT OUT OUT	nginx (*SMTP daemon*) rabbitmq zeneventserver zenjserver memcached zends	Can be replicated (with nginx) on additional servers to off-load report generation.

1. For performance, Zenoss normally runs multiple instances of the Zope daemon (two, by default). Each instance is automatically configured with a unique incoming HTTP port. By default, the initial instance listens on port 9081 and each additional instance uses the port number equal to that of the previous instance plus one (9082, 9083, etc.). See the *zenwebserver* chapter of the *Resource Management Extended Monitoring* guide for information on how to manage the number of concurrent Zope servers.
2. Local connections to ZenDS are through the `/var/lib/zends/zends.sock` unix socket.

Zenoss Hub Daemons

The zenhub daemon must run on every hub.

Zenoss Hub Daemons					
Daemon	**Port**	**Proto**	**Dir**	**Source / Destination**	**Notes**
zenhub	8081[1] 8090 8789[1] 5672 (AMQP) 8084 8085 11211 (memcache) 13306	TCP TCP TCP TCP TCP TCP TCP TCP	IN IN IN OUT OUT OUT OUT OUT	*collector daemons* nginx *collector daemons* rabbitmq zeneventserver zencatalogservice memcached zends	XML-RPC Graph rendering "ZenHub"
zentune	5672 (AMQP) 8084 11211 (memcache) 13306	TCP TCP TCP TCP	OUT OUT OUT OUT	rabbitmq zeneventserver memcached zends	

1. The "ZenHub" and XML-RPC port numbers are specified when the hub is created. They default to the lowest port numbers greater than 8789 and 8081, respectively, which are not being used by an existing hub.

Zenoss Collector Daemons

Note that all collector daemons must be able to connect to their hub's "ZenHub" and XML-RPC ports, which usually vary from hub to hub. Most collector daemons do not need to be run if they are not required for monitoring the devices assigned to the collector.

Daemons marked with a dagger (†) must be run on every collector.

Zenoss Collector Daemons					
Daemon	**Port**	**Proto**	**Dir**	**Source / Destination**	**Notes**
zencommand	22 (SSH) *	TCP	OUT	*monitored devices*	*See note 1.
zeneventlog	135 (EPMAP) *	TCP TCP	OUT IN/OUT	*monitored devices* *monitored devices*	*See notes 2 and 3.
zenjmx	*	TCP	OUT	monitored devices	*See note 4.
zenmailtx	25 (SMTP) 110 (POP3)	TCP TCP	OUT OUT	*outgoing SMTP server* *monitored POP3 service*	
zenmodeler†	*	*	OUT	*monitored devices*	*The port(s) and protocol(s) used for device modeling are determined by which modeler plugins have been enabled for the device(s) being modeled.
zenperfsnmp	161 (SNMP)	UDP	—	*monitored devices*	

Zenoss Collector Daemons

Daemon	Port	Proto	Dir	Source / Destination	Notes
zenping	echo request echo-reply	ICMP ICMP	OUT IN	*monitored devices* *monitored devices*	Devices with a non-empty IP address will periodically be probed by the zenping daemon by default. Set the **zPingMonitorIgnore** configuration property to **true** (checked) to prevent this behavior.
zenprocess	161 (SNMP)	UDP	—	*monitored devices*	
zenrender†	8091	TCP	IN	nginx	
zenrrdcached†	**$ZENHOME/var/rrdcached.sock**	unix	—		
zenstatus	*	TCP	OUT	*monitored devices*	*zenstatus attempts to open a TCP connection to the port defined for each monitored IP Service on each device where the service is monitored.
zensyslog	514 (Syslog)	UDP	IN	*monitored devices*	
zentune	13306	TCP	OUT	*zends*	
zentrap	162 (SNMPTrap)	UDP	IN	*monitored devices*	
zenucsevents	80 (HTTP)	TCP	OUT	*monitored devices*	
zenvcloud	443 (HTTPS)	TCP	OUT	*monitored devices*	
zenvmwareevents	443 (HTTPS) or 80 (HTTP)	TCP	OUT	*monitored devices*	
zenvmwaremodeler	443 (HTTPS) or 80 (HTTP)	TCP	OUT	*monitored devices*	
zenvmwareperf	443 (HTTPS) or 80 (HTTP)	TCP	OUT	*monitored devices*	
zenwebtx	80 (HTTP)	TCP	OUT	*monitored devices*	
zenwin	135 (EPMAP) *	TCP TCP	OUT IN/OUT	*monitored devices* *monitored devices*	*See note 2.
zenwinperf	445 (MS-DS)	TCP	OUT	*monitored devices*	

1. **zencommand**: In addition to running commands on monitored devices, the zencommand daemon is also used to run commands (for example, Nagios plugins) on the collector. Those commands often then connect to the monitored device. See the **Additional Monitoring Port Usage** section below for more information.
2. **zeneventlog**, **zenwin**: These daemons use Windows RPC to communicate with the WMI service on the remote device. By default, Windows RPC allocates a dynamic port, in addition to port 135, in the range of 49152–65535 or 1025–5000 depending on the version of Windows. See the *Service overview and network port requirements for Windows* Microsoft support article for more information.
3. **zeneventlog**: This daemon will attempt to monitor the Windows event logs of any devices where the **zWmiMonitorIgnore** configuration property is set to **False** (unchecked) and the **zWinEventlog** configuration property is set to **True** (checked), which is the default configuration on the **/Server/Windows** device class.
4. **zenjmx**: The zenjmx daemon provides monitoring of remote Java® applications using Java Monitoring Extensions (JMX) using either RMI (Remote Method Invocation) or JMXMP (JMX Messaging Protocol). Device class specific configuration properties are used to define the remote port and authentication credentials.

   Note that the RMI protocol requires a second connection that, by default, goes to a dynamically allocated (essentially random) port number. See the *Java 2 Platform Standard Edition* chapter in the *Resource Manager Extended Monitoring* guide for more information.

Zenoss Collectors by Device Class

This table details the collector daemons invoked by default on devices in each device class. The listed port numbers are for outgoing connections from the collector to the monitored device. See the **Zenoss Collector Daemons** section above for additional information about a specific daemon.

Device classes marked with a dagger (†) are used primarily as containers for sub-classes (or other special purposes). Zenoss recommends that you do not add devices to these classes, but to an appropriate subclass instead.

Zenoss Collector to Monitored Devices by Device Class			
Device Class	**Port**	**Proto**	**Daemon / Notes**
/**Devices**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/**AWS**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/AWS/**EC2**	443 (HTTPS) or 80 (HTTP)	TCP	zencommand
/Devices/**CiscoUCS**	80 (HTTP)*	TCP	zenucsevents *The destination port number is specified when the device is added and is maintained in the **zCiscoUCSManagerPort** configuration property.
/Devices/**Discovered**†	161 (SNMP)	UDP	zenperfsnmp Devices are normally added to this class by the auto-discovery process. Modeling uses the SNMP, SSH, and WMI protocols.
/Devices/**HTTP**	80 (HTTP)*	TCP	zencommand Invokes the **`check_http`** Nagios plugin *The port number can be changed in the HttpMonitor data source of the HttpMonitor monitoring template.
/Devices/**KVM**	161 (SNMP)	UDP	zenperfsnmp
/Devices/**Network**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**BIG-IP**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**Check Point**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/Check Point/**SPLAT**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/**Cisco**†	161 (SNMP)	UDP	zenperfsnmp Some Cisco devices can be configured to send SNMP traps and syslog messages to their Zenoss collector. See **zensyslog** and **zentrap** in the previous section.
/Devices/Network/Cisco/**6500**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/Cisco/6500/**VSS**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/Cisco/**ACE**	161 (SNMP) 80 (HTTP)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/Cisco/**ASA**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/**ASR**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/ASR/**1000**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/ASR/**9000**	161 (SNMP) 23 (TELNET)	UDP TCP	zenperfsnmp zencommand
/Devices/Network/Cisco/**CatOS**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/**Codec**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/**FWSM**	161 (SNMP)	UDP	zenperfsnmp

Zenoss Collector to Monitored Devices by Device Class

Device Class	Port	Proto	Daemon / Notes
/Devices/Network/Cisco/**IDS**	443 (HTTPS)	TCP	zencommand
/Devices/Network/Cisco/**MDS**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/MDS/**9000**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Cisco/**Nexus**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zenmodeler (NETCONF over SSH)
/Devices/Network/Cisco/Nexus/**1000V**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zenmodeler (NETCONF over SSH)
/Devices/Network/Cisco/Nexus/**5000**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zenmodeler (NETCONF over SSH)
/Devices/Network/Cisco/Nexus/**7000**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zenmodeler (NETCONF over SSH)
/Devices/Network/Cisco/**VSG**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zenmodeler (NETCONF over SSH)
/Devices/Network/Cisco/**WLC**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**Juniper**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Juniper/**M10i**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**NetScreen**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**Router**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Router/**Cisco**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Router/**Firewall**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Router/**RSM**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Router/**TerminalServer**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/**Switch**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Switch/**Nortel**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Network/Switch/**Passport**	161 (SNMP)	UDP	zenperfsnmp
/Devices/**Ping**	Echo request Echo reply	ICMP	zenping Devices in this class will only be monitored for up / down status. See **zenping** in the previous section.
/Devices/**Power**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Power/**UPS**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Power/UPS/**APC**	161 (SNMP)	UDP	zenperfsnmp
/Devices/**Printer**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Printer/**InkJet**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Printer/**Laser**	161 (SNMP)	UDP	zenperfsnmp
/Devices/**Server**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/Server/**Cmd**	22 (SSH)	TCP	zencommand This device class is depreciated; use one of the sub-classes of **/Devices/Server/SSH/** instead.
/Devices/Server/**Darwin**	161 (SNMP)	UDP	zenperfsnmp, zenprocess
/Devices/Server/**JBoss**	161 (SNMP) *	UDP TCP	zenperfsnmp, zenprocess zenjmx *The port number used for JMX monitoring is set by the **zJBossJmxManagementPort** configuration property. See **zenjmx** in the previous section.
/Devices/Server/**Linux**	161 (SNMP)	UDP	zenperfsnmp, zenprocess
/Devices/Server/**Remote**	161 (SNMP)	UDP	zenperfsnmp, zenprocess
/Devices/Server/**Scan**	*	TCP	zenstatus
/Devices/Server/**Solaris**	161 (SNMP)	UDP	zenperfsnmp, zenprocess

Zenoss Collector to Monitored Devices by Device Class

Device Class	Port	Proto	Daemon / Notes
/Devices/Server/**SSH**†	—	—	
/Devices/Server/SSH/**AIX**	22 (SSH)	TCP	zencommand
/Devices/Server/SSH/**HP-UX**	22 (SSH)	TCP	zencommand
/Devices/Server/SSH/**Linux**	22 (SSH)	TCP	zencommand
/Devices/Server/SSH/**Solaris**	22 (SSH) 161 (SNMP)	TCP TCP	zencommand zenperfsnmp
/Devices/Server/**Tomcat**	161 (SNMP) *	UDP TCP	zenperfsnmp, zenprocess zenjmx *The port number used for JMX monitoring is set by the **zTomcatJmxManagementPort** configuration property. See **zenjmx** in the previous section.
/Devices/Server/Virtual Machine **Host**†	161 (SNMP)	UDP	zenperfsnmp, zenprocess
/Devices/Server/Virtual Machine Host/**ESX**	161 (SNMP)	UDP	zenperfsnmp, zenprocess
/Devices/Server/Virtual Machine Host/**EsxTop**	161 (SNMP) 443 (HTTPS)	UDP TCP	zenperfsnmp zencommand Invokes the **`resxtop`** command on the collector. resxtop is part of the VMware vSphere CLI. resxtop connects to the monitored device using HTTPS.
/Devices/Server/Virtual Machine Host/**Xen**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/Server/**WebLogic**	161 (SNMP) *	UDP TCP	zenperfsnmp, zenprocess zenjmx *The port number used for WebLogic monitoring is set by the **zWebLogicJmxManagementPort** configuration property. See **zenjmx** in the previous section.
/Devices/Server/**Windows**	161 (SNMP) 135 (EPMAP), *	UDP TCP	zenperfsnmp zeneventlog
/Devices/Server/Windows/**WMI**	135 (EPMAP), * 445 (MS-DS)	TCP TCP	zeneventlog, zenwin zenwinperf
/Devices/Server/Windows/WMI/**Active Directory**	135 (EPMAP), * 445 (MS-DS)	TCP TCP	zeneventlog, zenwin zenwinperf
/Devices/Server/Windows/WMI/**MSExchange**	135 (EPMAP), * 445 (MS-DS)	TCP TCP	zeneventlog, zenwin zenwinperf
/Devices/Server/Windows/WMI/**MSSQLServer**	135 (EPMAP), * 445 (MS-DS)	TCP TCP	zeneventlog, zenwin zenwinperf
/Devices/**Storage**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/Storage/**Brocade**	161 (SNMP)	UDP	zenperfsnmp
/Devices/Storage/**NetApp**	161 (SNMP) 22 (SSH)	UDP TCP	zenperfsnmp zencommand
/Devices/**vCloud**	443 (HTTPS)	TCP	zenvcloud
/Devices/**VMware**	443 (HTTPS) or 80 (HTTP)	TCP	zenvmwaremodeler, zenvmwareevents, zenvmwareperf
/Devices/**Web**†	161 (SNMP)	UDP	zenperfsnmp
/Devices/Web/**SugarCRM**	80 (HTTP)	TCP	zenwebtx

Additional Monitoring Port Usage

Additional monitoring functionality can be added to a device or device class by binding the appropriate monitoring template or adding a data source to an already bound template. This table shows the port numbers used when specific monitoring capabilities are applied. See the *Resource Manager Extended Monitoring* guide for more information.

Additional Monitoring Port Usage

Description	Port	Proto	Direction	Collector Daemon	Notes
Apache HTTP Server™ Monitoring	80 (HTTP)	TCP	OUT	zencommand	Provided by the **Apache** monitoring template.
DNS Monitoring	53 (Domain)	UDP	OUT	zencommand	Provided by the **DigMonitor** and **DnsMonitor** monitoring templates. Invokes the **`check_dig`** or **`check_dns`** Nagios plugin respectively.
FTP Service Monitoring	21 (FTP)	TCP	OUT	zencommand	Provided by the **FtpMonitor** monitoring template. Invokes the **`check_ftp`** Nagios plugin.
IRC Service Monitoring	6667 (IRCD)	TCP	OUT	zencommand	Provided by the **IRCD** monitoring template. Invokes the **`check_ircd`** Nagios plugin.
Jabber® Service Monitoring	5223	TCP	OUT	zencommand	Provided by the **JabberMonitor** monitoring template. Invokes the **`check_jabber`** Nagios plugin.
LDAP Response Time Monitoring	389 (LDAP)	TCP	OUT	zencommand	Provided by the **LDAPServer** monitoring template. Invokes the **`check_ldap`** or **`check_ldaps`** Nagios plugin.
Microsoft Message Queuing (MSMQ) Monitoring	445 (MS-DS)	TCP	OUT	zenwinperf	Provided by the **MSMQQueue** monitoring template.
Microsoft Internet Information Services (IIS) Monitoring	445 (MS-DS)	TCP	OUT	zenwinperf	Provided by the **IIS** monitoring template.
MySQL® Monitoring	3306	TCP	OUT	zencommand	Provided by the **MySQL** monitoring template.
Network News Transport Protocol (NNTP) Monitoring	119 (NNTP) or 563 (NNTPS)	TCP	OUT	zencommand	Provided by the **NNTPMonitor** monitoring template. Invokes the **`check_nntp`** or **`check_nntps`** Nagios plugin.
Network Time Protocol (NTP) Monitoring	123 (NTP)	UDP	OUT	zencommand	Provided by the **NTPMonitor** monitoring template. Invokes the **`check_ntp`** Nagios plugin.
SQL Transactions	*	TCP	OUT	zencommand	Provided by the **SQL** data source type. *The destination port number depends on the SQL server and is specified in the data source properties.
WebSphere® Application Server	80 (HTTP)*	TCP	OUT	zenwebtx	Provided by the **Websphere** monitoring template. *A custom port number can be set in the **Initial URL** data source property.