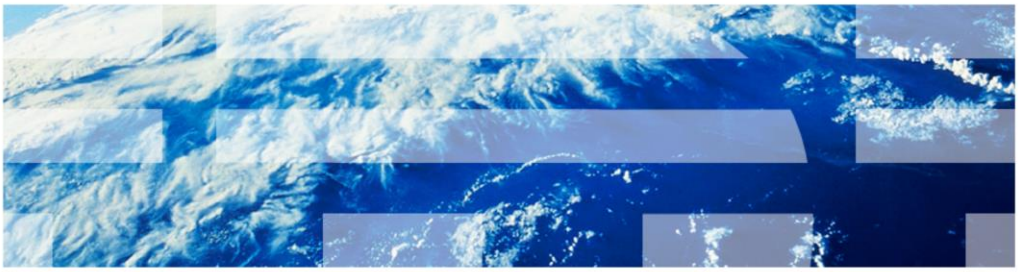
IBM

IBM Tivoli Application Dependency Discovery Manager 7.2.1

Anchors



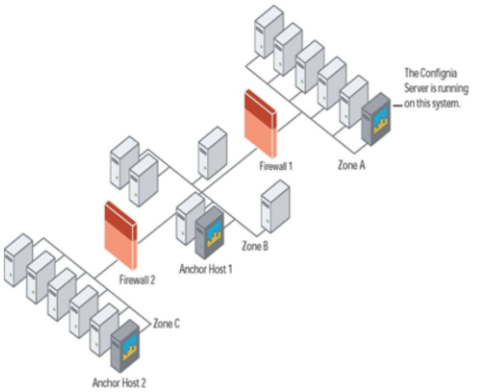
This presentation covers anchors in Tivoli® Application Dependency Discovery Manager version 7.2.1

After you complete this module, you can describe how Tivoli Application Dependency Discovery Manager anchors work and diagnose common errors.

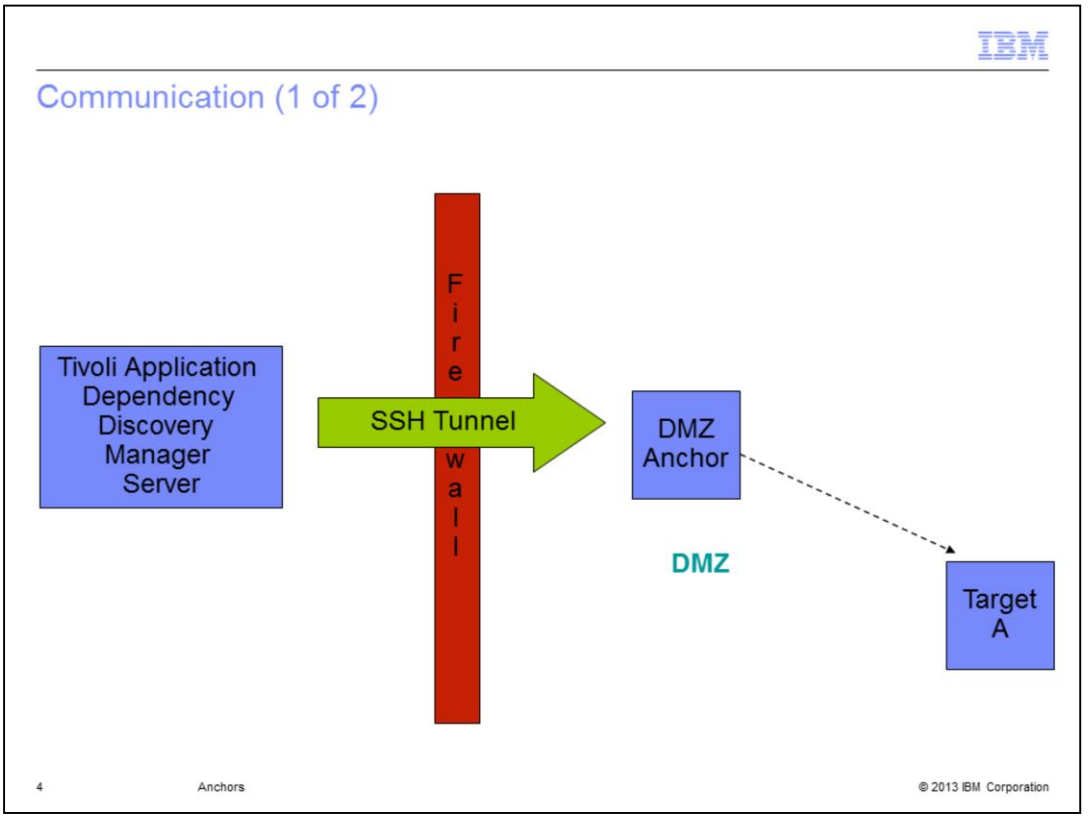
Tivoli Application Dependency Discovery Manager anchors

With anchors, you can run discovery across firewall zones

- Many companies restrict access to areas of their network using firewalls
- Tivoli Application Dependency Discovery Manager can use a server that is located within the firewall zone to perform discoveries on behalf of the primary root server
- If there are Windows servers within the firewall zone, a Windows Gateway is also required

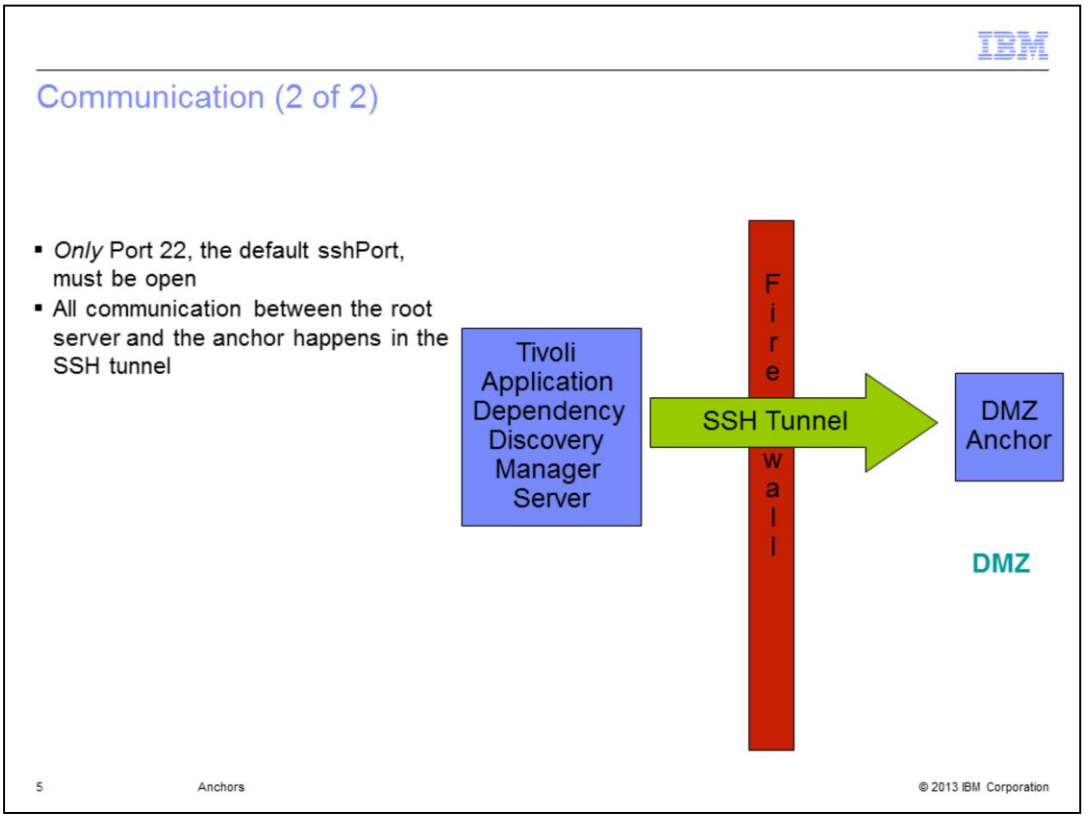


Many companies use firewalls to restrict access to parts of their network, making discovery in those areas difficult. Tivoli Application Dependency Discovery Manager anchors allow for discovery across firewall zones. From an anchor in the firewall zone, Tivoli Application Dependency Discovery Manager can perform discoveries on behalf of the primary discovery management server, called the root server. If Windows servers are within the firewall zone, then you also need a Windows gateway in that zone.

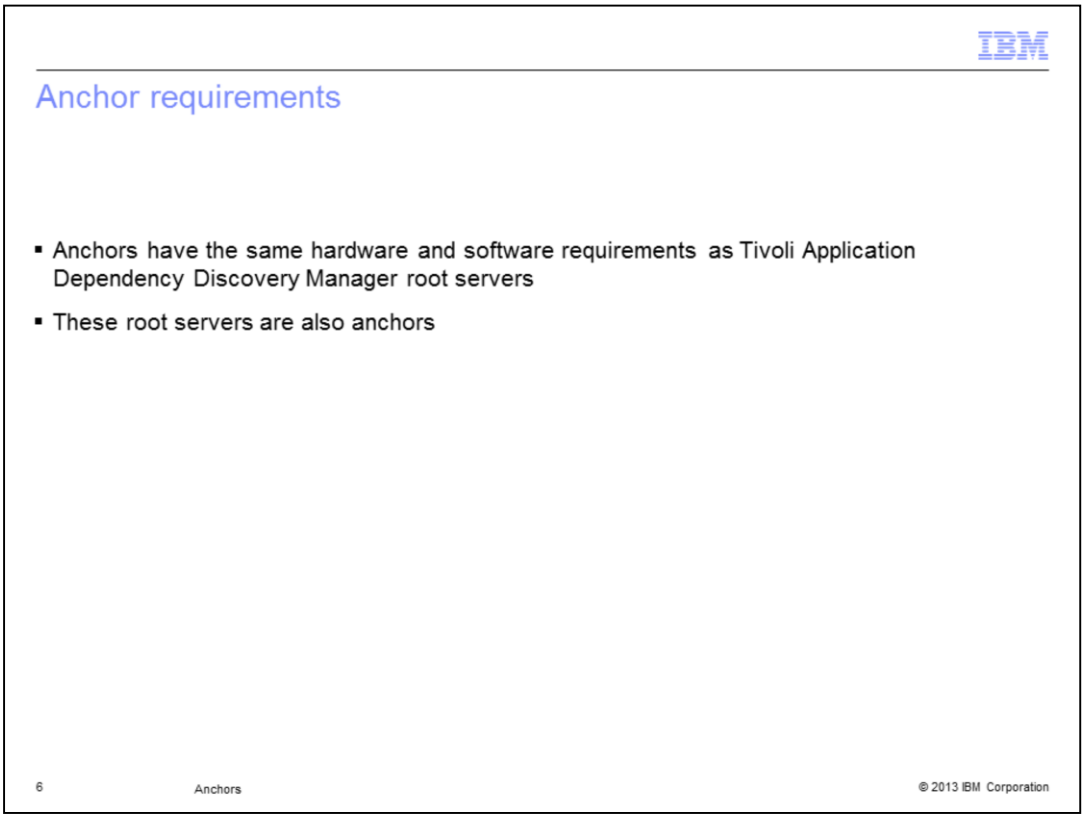


The root server is itself an anchor. It uses sockets to communicate to the remote anchor through an SSH tunnel.

A socket address is the combination of an IP address and a port into a single identity, much like one end of a telephone connection is the combination of a telephone number and a particular extension.

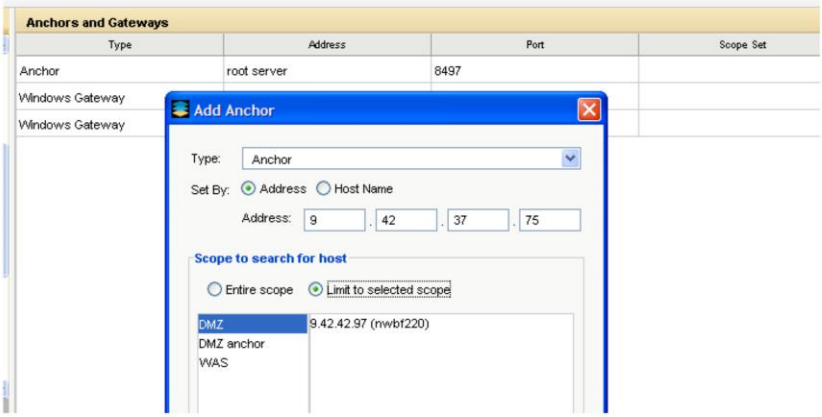


Only Port 22, or whatever has been defined as the ssh port, must be open on the firewall between the root server and the anchor. All communication between the root server and the anchor is done in the SSH tunnel.



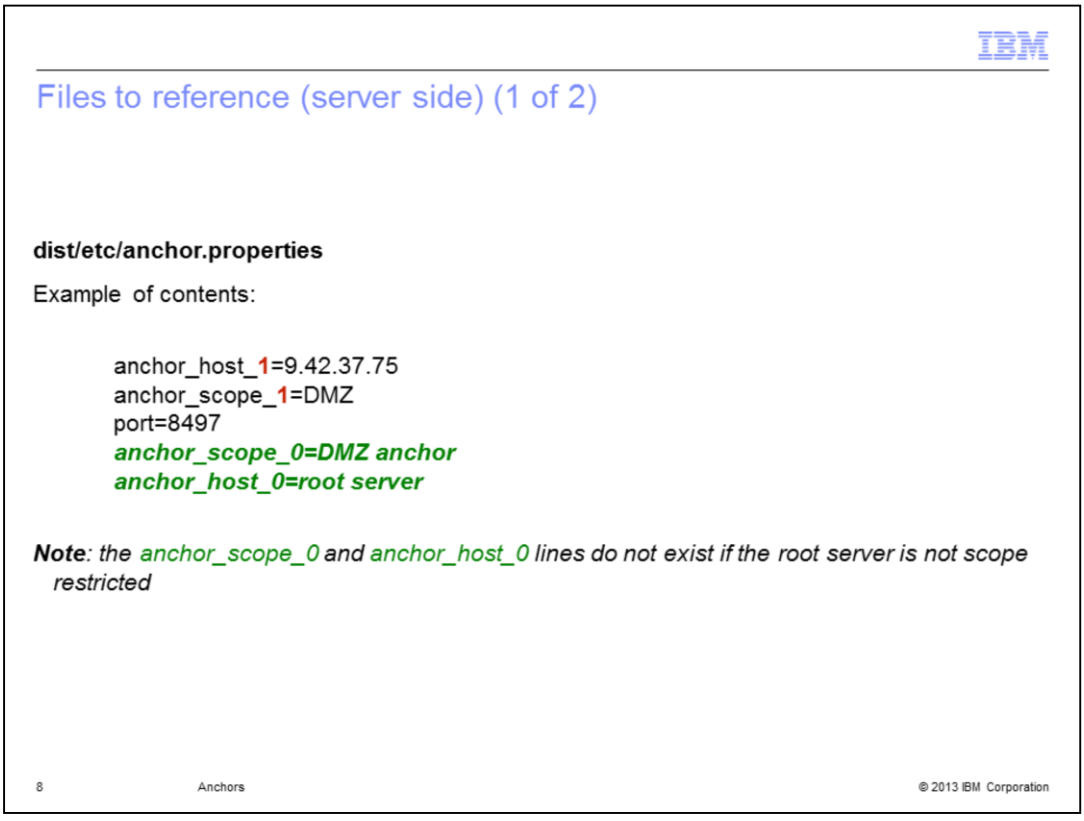
When you remember that the root server is itself an anchor, it is easy to remember that anchors have all the same hardware and software requirements as Tivoli Application Dependency Discovery Manager servers.

Anchors are defined by the Data Management Console

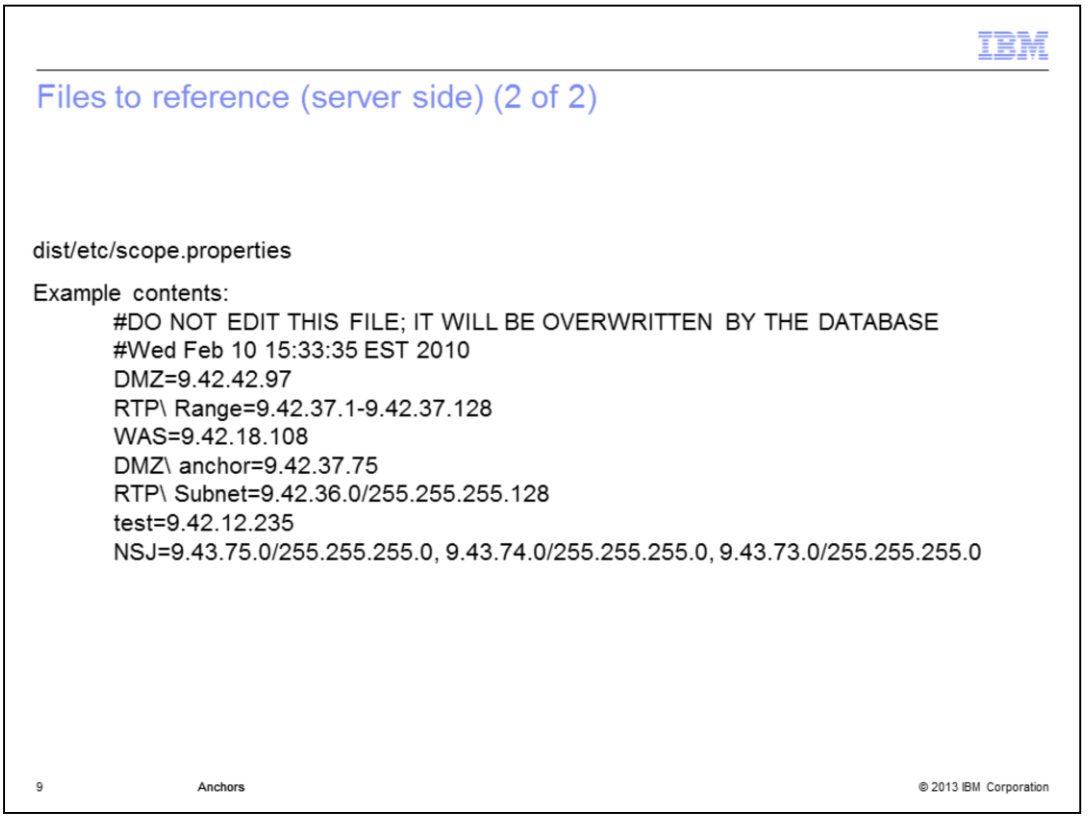


7 Anchors

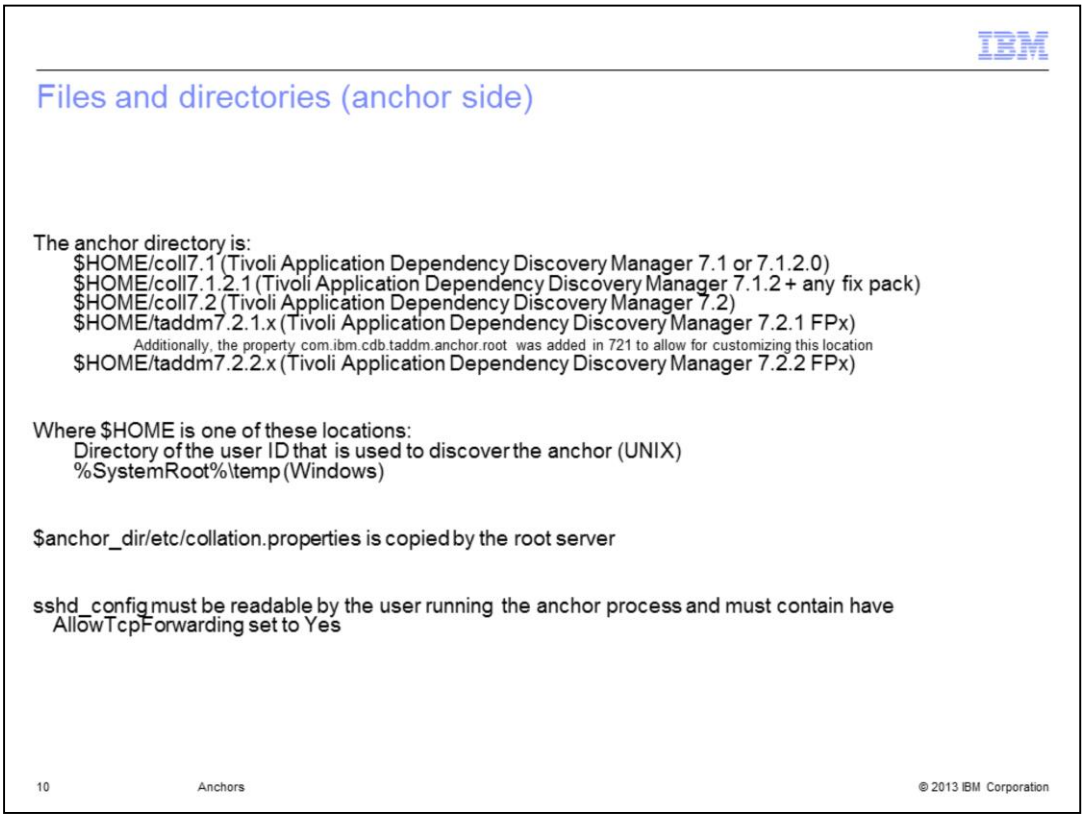
In the data management console, navigate to the Anchors and Gateways panel to define the anchor and, if applicable, its scope restriction.



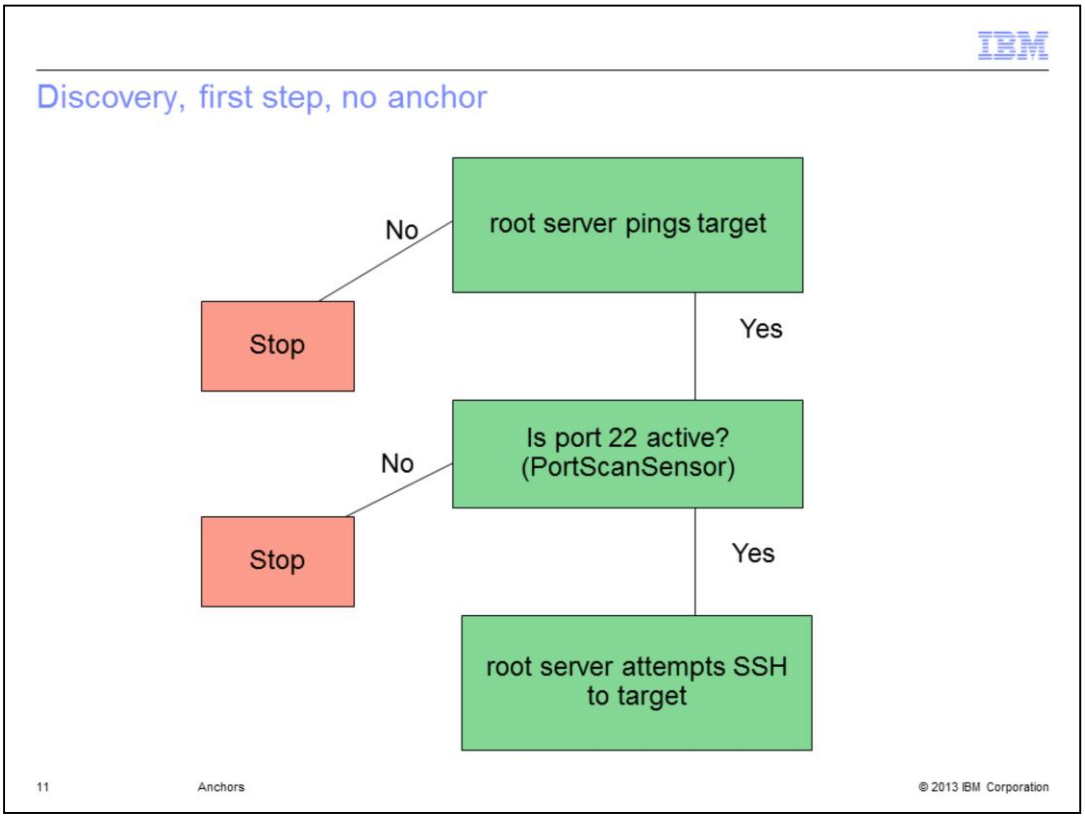
The anchor.properties file, which is located in the **dist/etc** subdirectory under the main Tivoli Application Dependency Discovery Manager directory, contains the defined anchors and their scope restrictions, if those exist. This file is automatically updated when you modify the Anchors and Gateways panel. You do not have to manually edit this file unless you use anchors for Network Address Translation (or NAT) environments. In the case of NAT subnets, you must assign the anchors to a zone using the anchor zone property. See the *Tivoli Application Dependency Discovery Manager User's Guide* for further instructions regarding NAT zones.



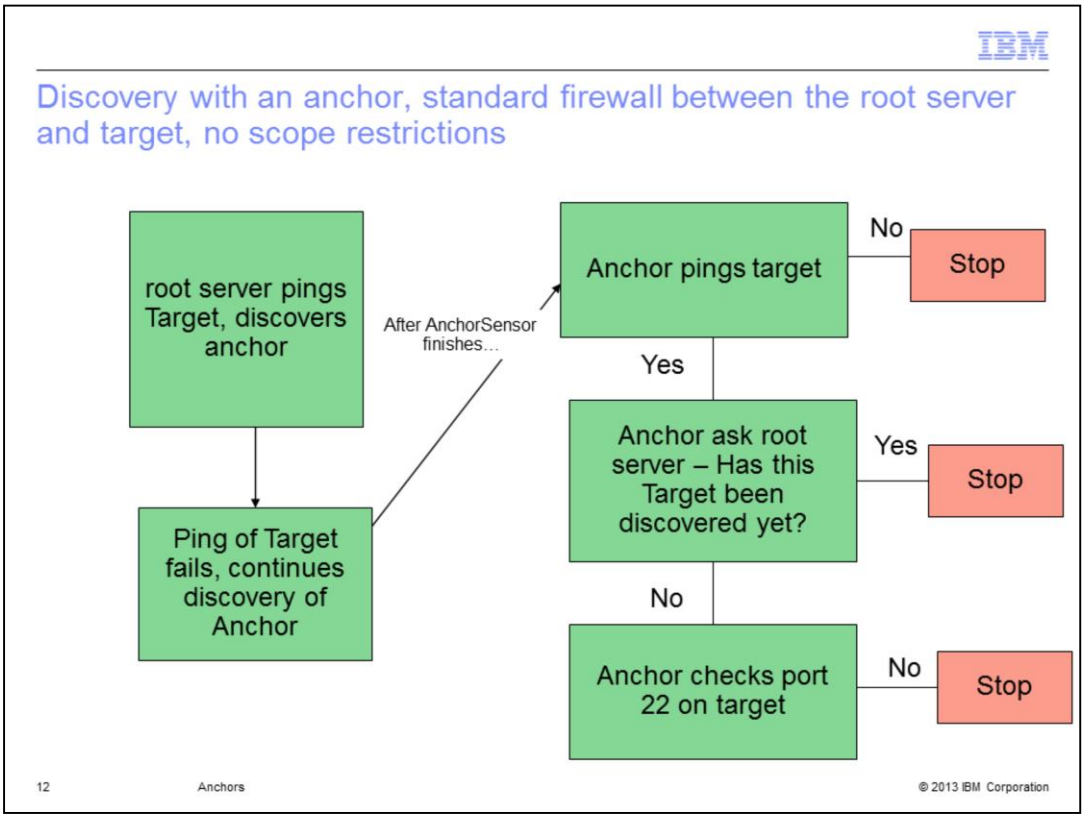
Do not modify the dist/etc/scope.properties file manually, because it updates automatically when you modify scopes on the data management console. It is a useful file, however, for diagnosing all anchor problems. If you report a discovery issue involving anchors to IBM Level 2 Support, provide the scope.properties file with the anchor.properties file.



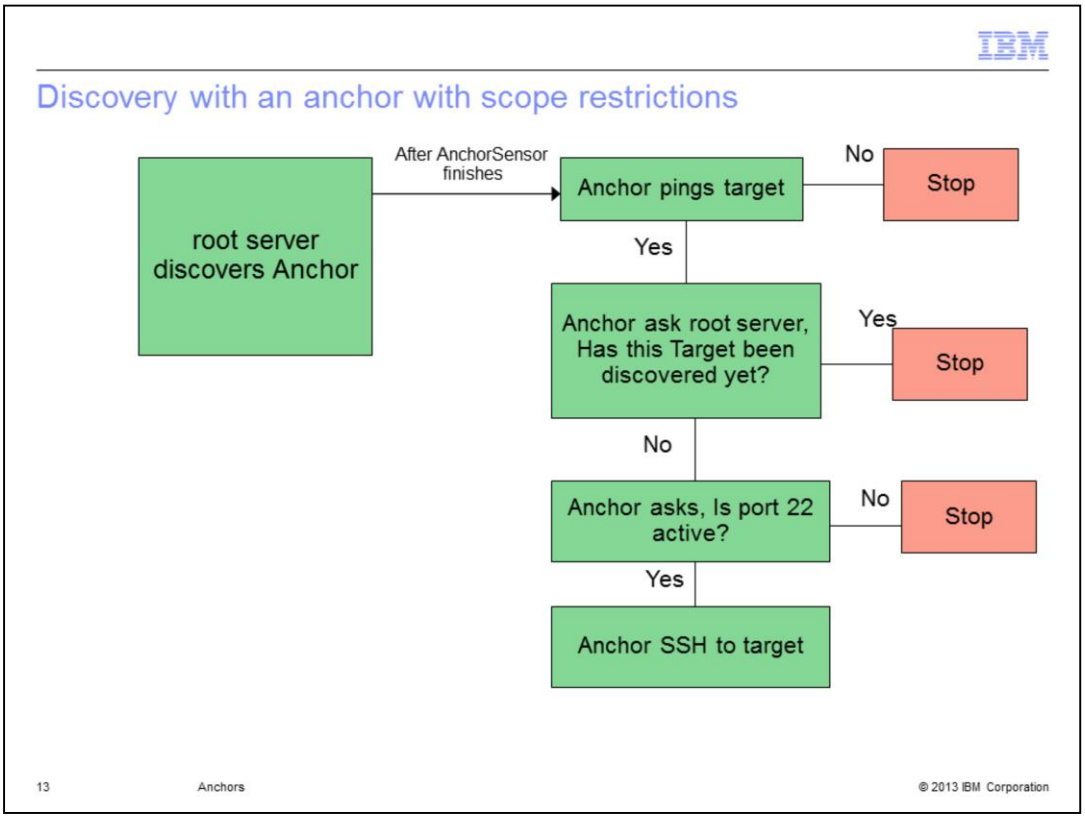
The root server copies all files that the anchor requires to run a discovery as part of the initial anchor deployment. These files are copied to the anchor directory on the anchor. The exact location of the anchor directory depends upon the platform (UNIX versus Windows), the user's home directory, and the version of Tivoli Application Dependency Discovery Manager. The directory structure below the anchor home directory will always be the same.



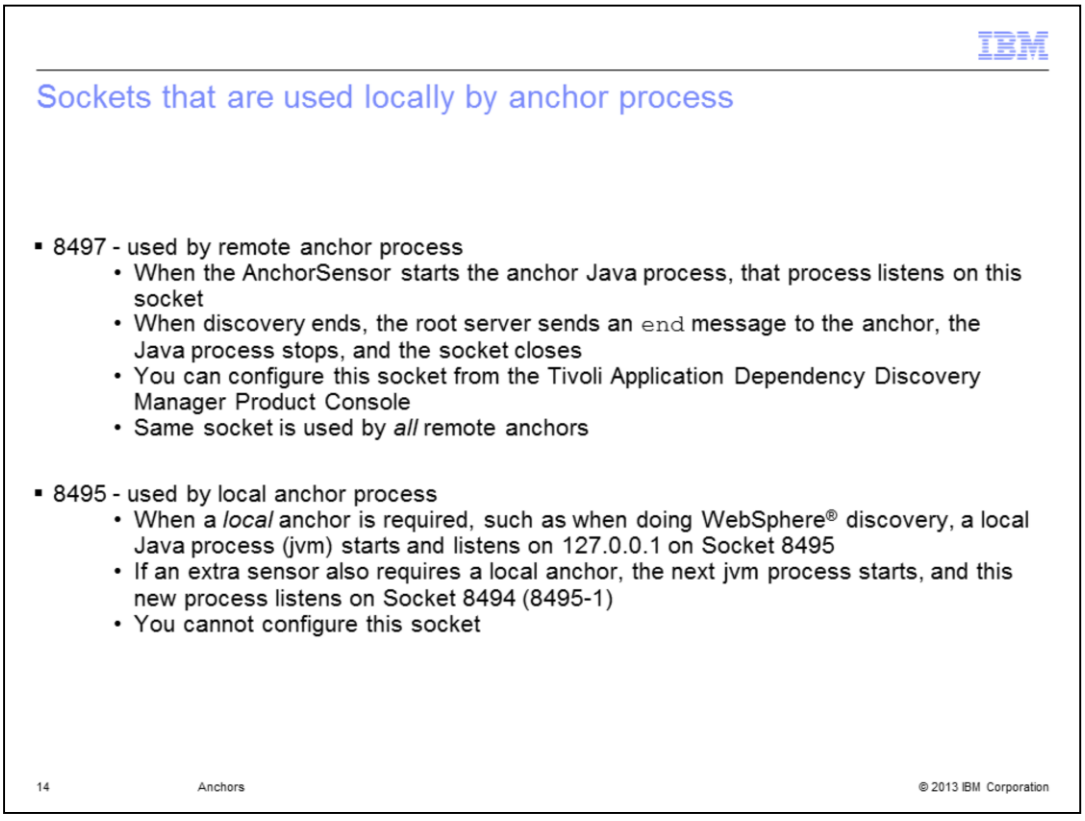
When root server runs a discovery, it first tries to ping the target IP to determine if it is an active IP. If there is no response, discovery proceeds no further. If the IP responds successfully to a ping, the root server then scans the SSH port. If the SSH port is active, the root server then attempts to discover the target. If the SSH port is not active, the root server attempts to discover the target using SNMP, but in this demonstration, for the sake of simplicity, you are not concerned with the SNMP part of discovery.



If an anchor is used for discovery and there are no scope restrictions, the root server still attempts to ping the target. After the anchor sensor completes, the anchor also attempts to ping the target. The anchor attempts further discovery if the target responds *and* the target has not already been discovered. Tivoli Application Dependency Discovery Manager keeps information that indicates whether a target has been discovered internally, but basically if the PortSensor ran successfully on the target from the root server or another anchor, then this anchor does not run discovery.



When the root server is restricted to the anchor, it does not run the ping sensor on the target. Only the anchor runs the ping sensor against the target.



The remote anchor process uses Socket 8497. When the Java process starts on the anchor by the AnchorSensor, that Java process listens on Socket 8497. When discovery ends, three events occur:

The root server sends an end message to the anchor.

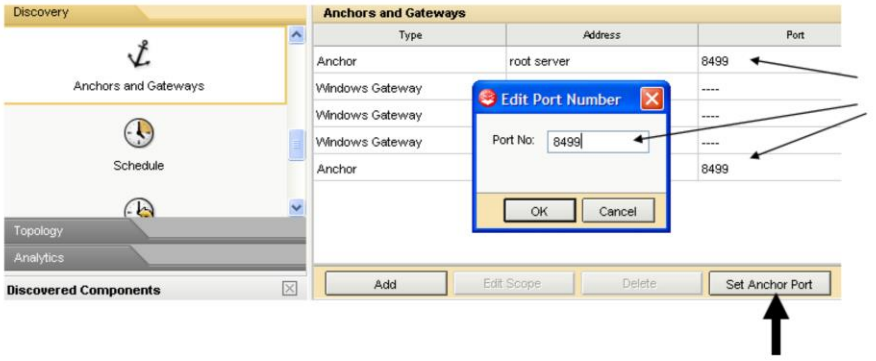
The Java process stops.

The socket closes.

You configure this socket, which all remote anchors use, from the Tivoli Application Dependency Discovery Manager Product Console.

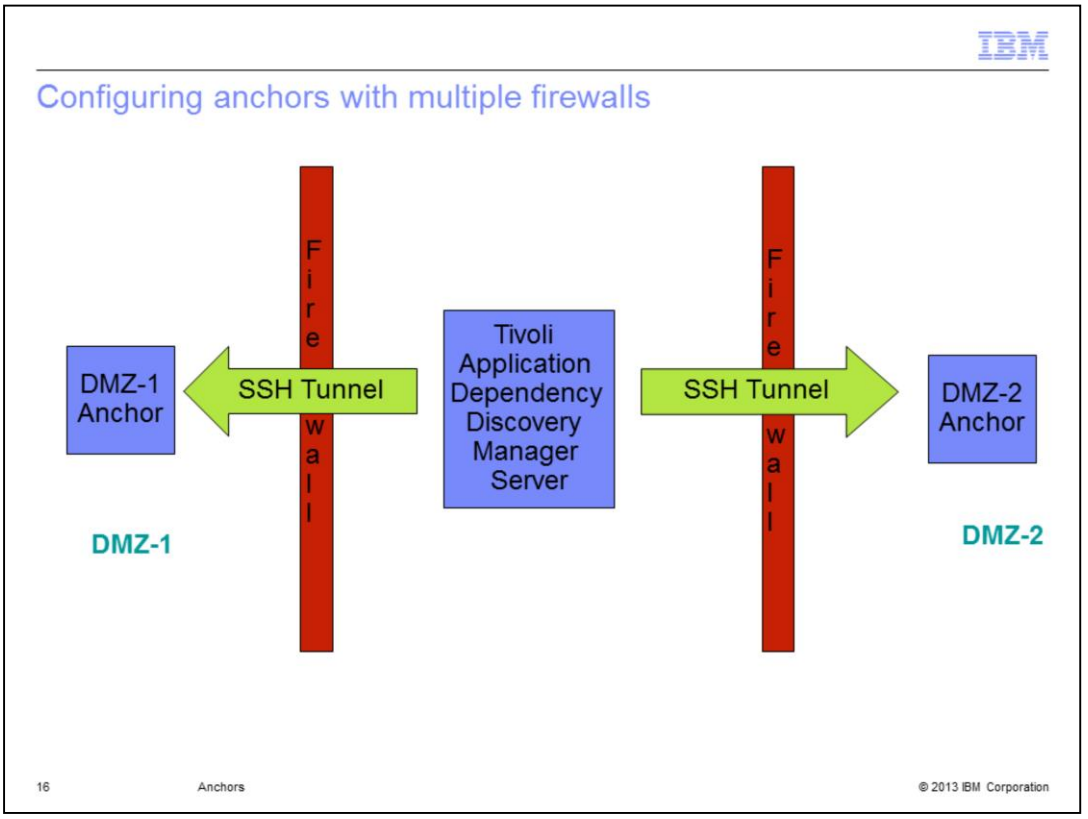
The local anchor process uses Socket 8495. For example, a local anchor is necessary when you run a WebSphere discovery. In such a case, a local Java process starts and listens from 127.0.0.1 on Socket 8495. If an additional sensor also requires a local anchor, the next Java process starts. This new process listens on Socket 8494 (8495 - 1). This socket cannot be configured.

Configuring socket that remote anchors use



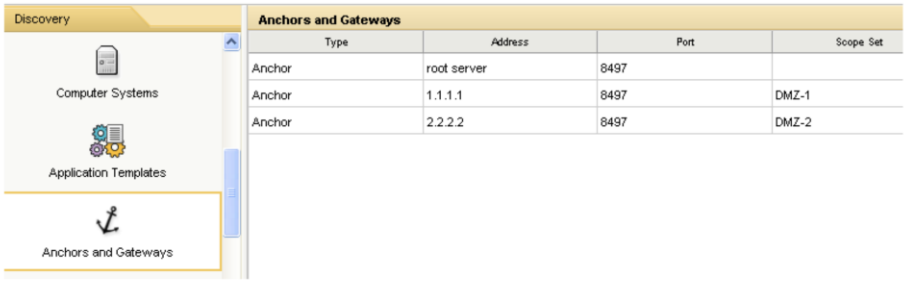
15 Anchors

You configure the socket that remote anchors use from the Tivoli Application Dependency Discovery Manager Product Console Anchors and Gateways panel.



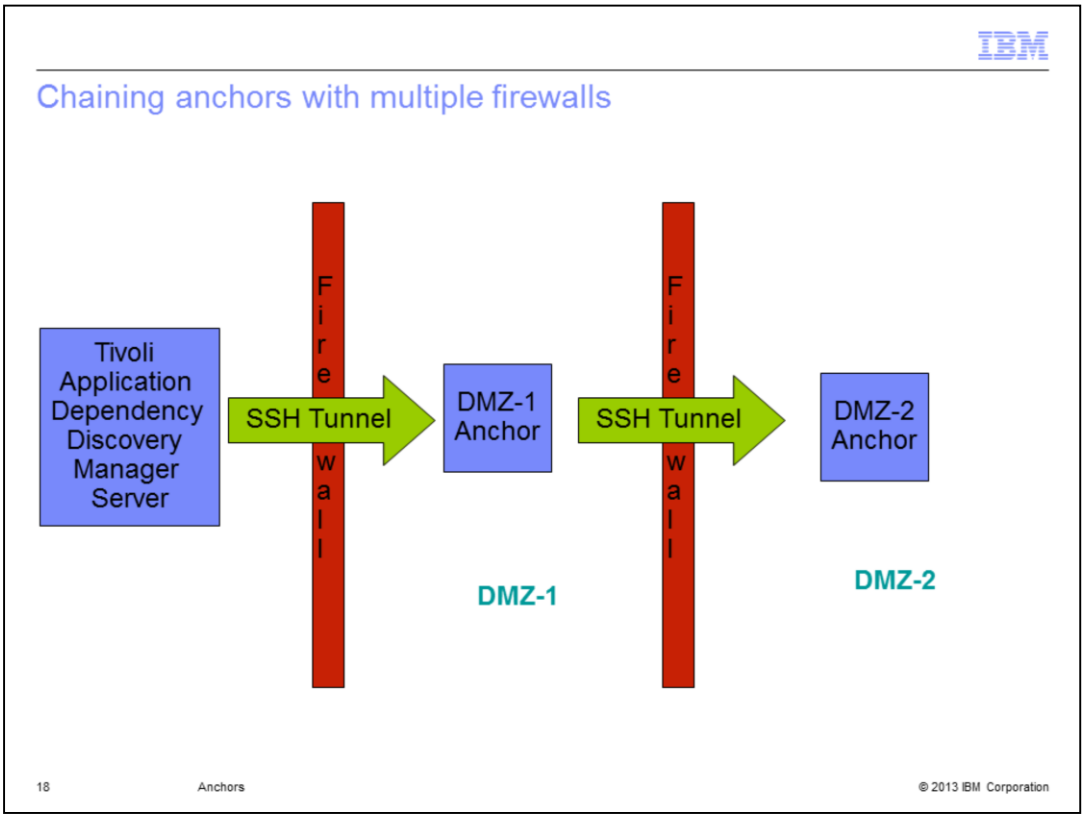
If there is only one firewall between the root server and the target, an anchor is placed beyond each firewall. Each anchor is responsible for discovery within its network zone. Only the SSH port must be open from root server to each anchor.

Multiple anchors



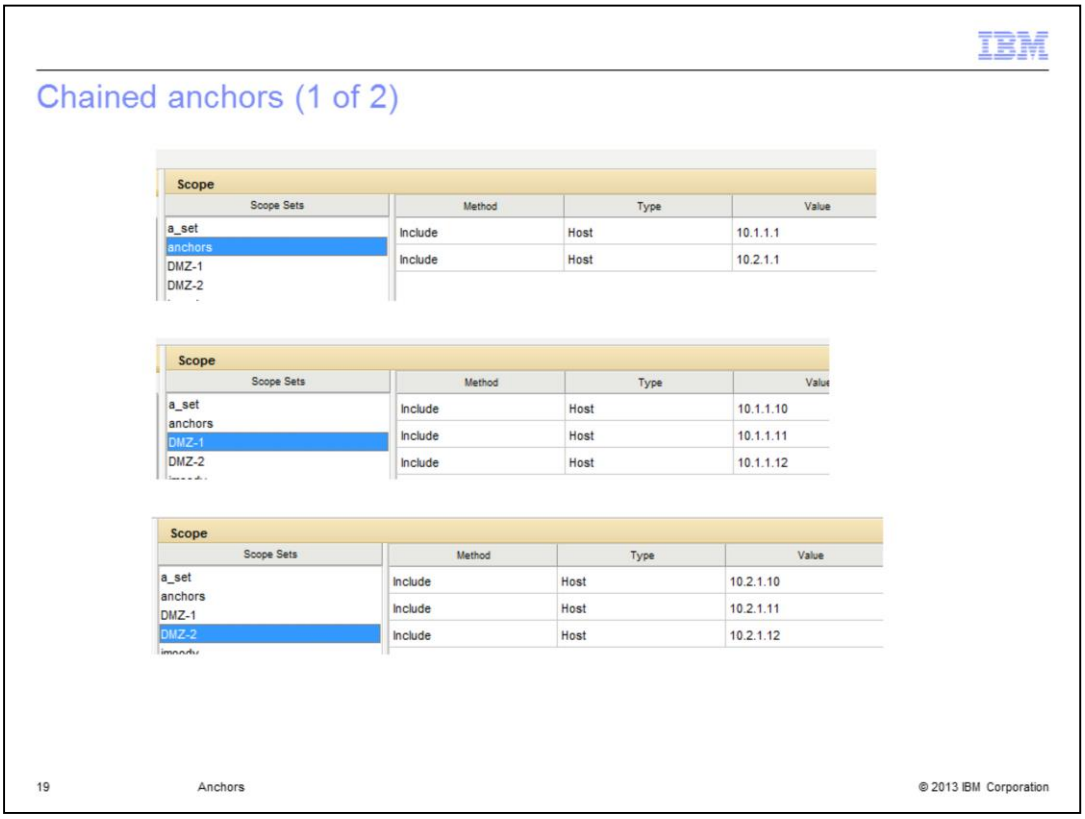
17 Anchors

To define the anchors for the previous example, Anchor IP 1.1.1.1 can run discoveries of all IPs in the scope DMZ-1, and Anchor IP 2.2.2.2 can run discoveries of all IPs in the scope DMZ-2.

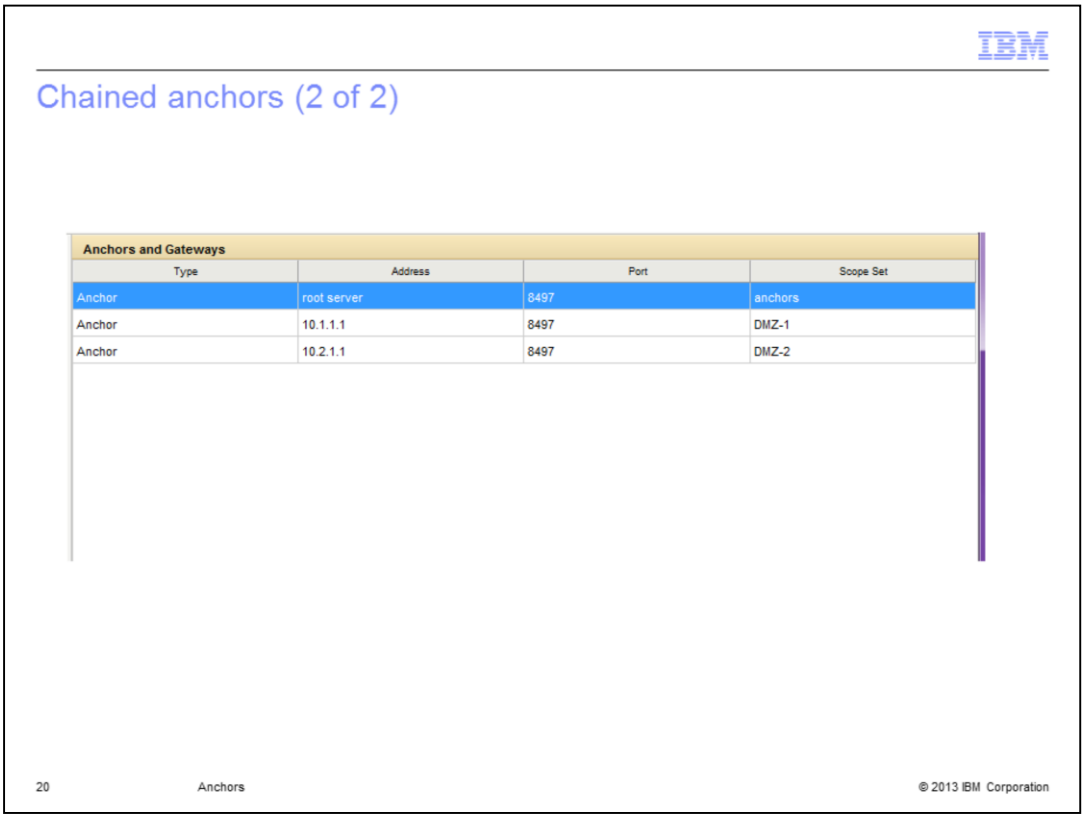


When there are multiple firewalls between the root server and the target, an anchor is placed within each firewalled network zone. Only the SSH port must be open on the firewall between each anchor (including the root server) and the next anchor.

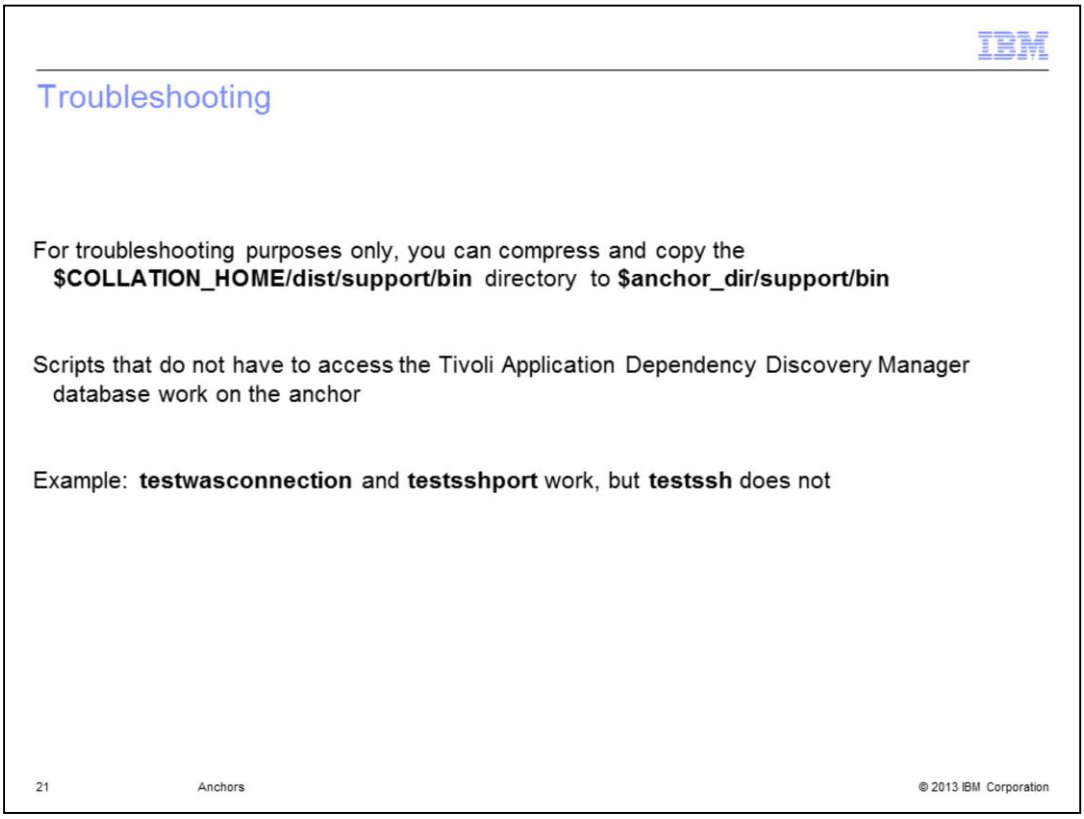
In this example with multiple firewalls, the root server can access only Anchor 1. Anchor 1 in turn accesses Anchor 2. Only the SSH port must be open from the root server to Anchor1 and from Anchor1 to Anchor2.



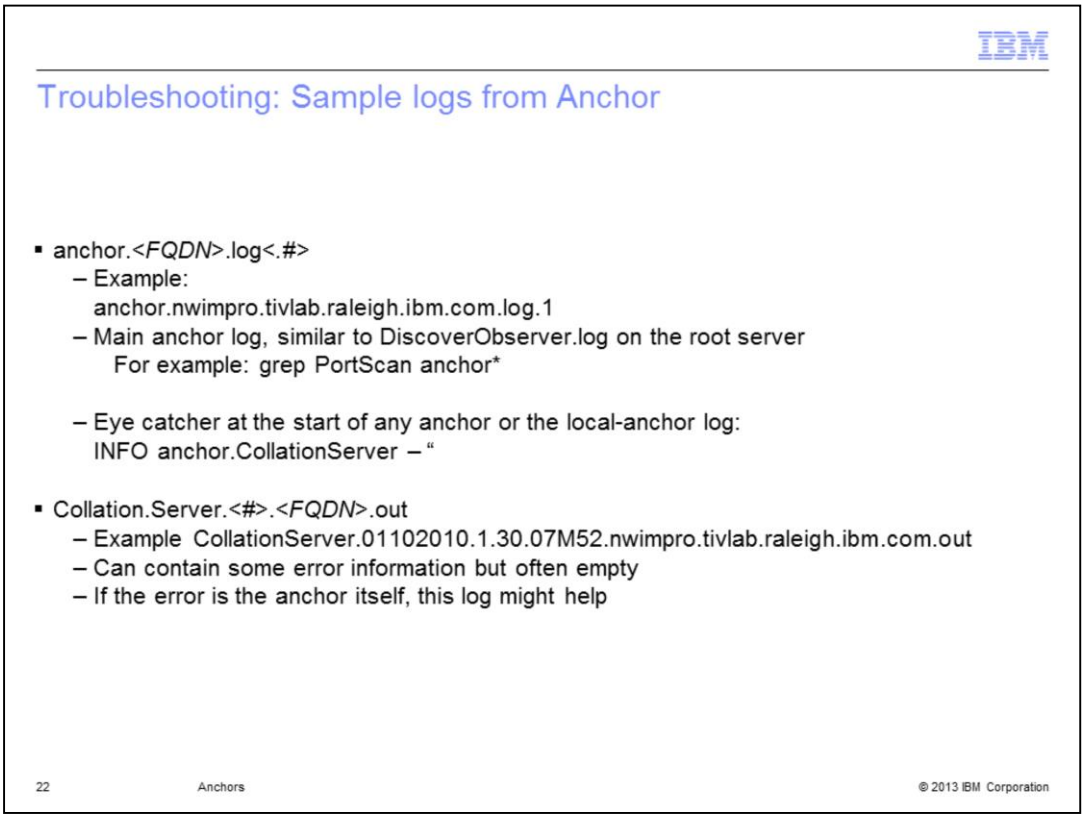
As an example of how to define chained anchors, first define the scopes for the anchors, DMZ-1 and DMZ-2.



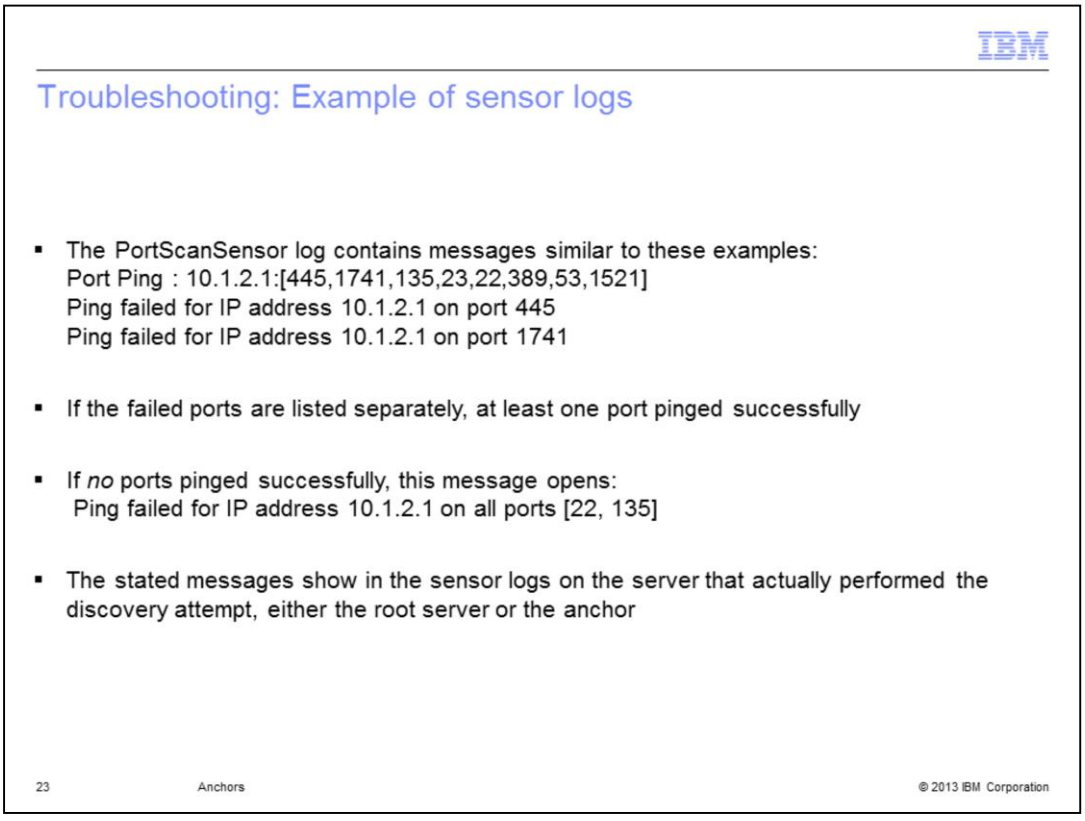
Next, you restrict the scope of each anchor to its appropriate DMZ. In this example, the root server is restricted to discover only the anchors themselves. That setup is not always necessary, but it prevents the root server from attempting to discover any targets.



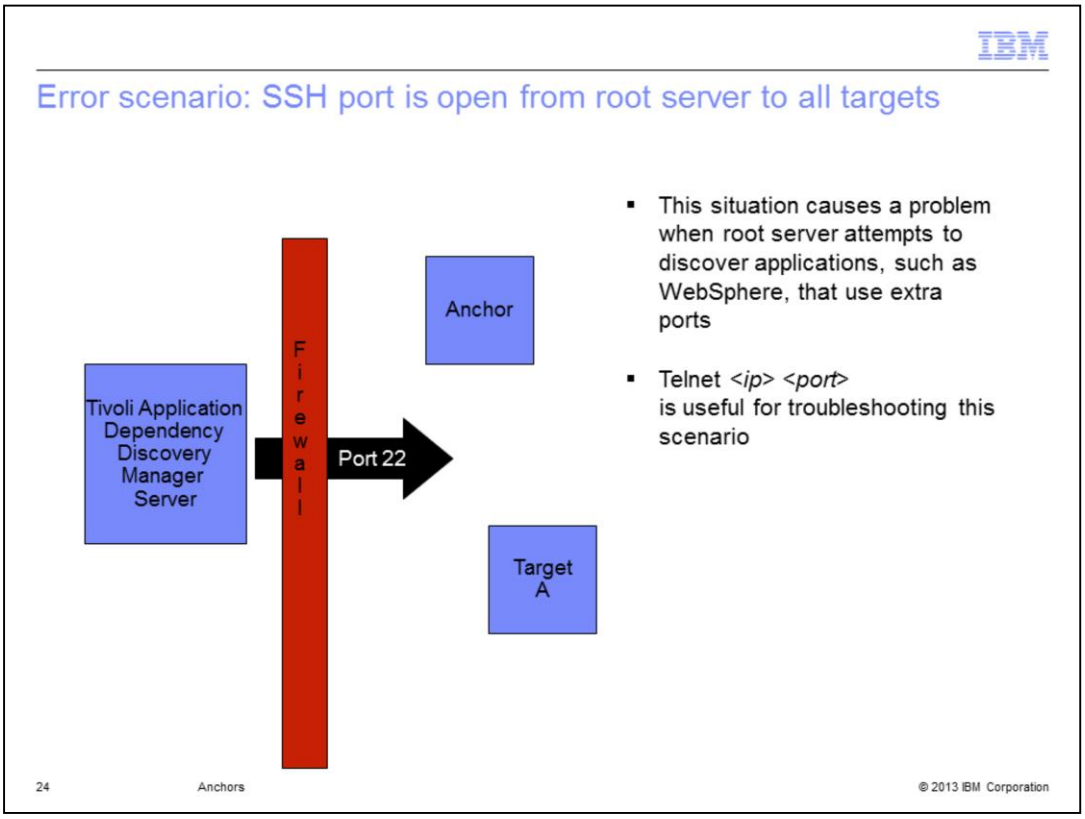
Though the scripts under dist/support/bin directory are not officially supported, you can copy them to the anchor, in attempts to diagnose the problem. Scripts that do not require access to the Tivoli Application Dependency Discovery Manager database work on the anchor.



Each new discovery in main log on the anchor starts with the string “INFO anchor.CollationServer –.” The Collation.Server files on the anchor are often empty, but sometimes you can use them to diagnose a communication problem on the anchor itself.

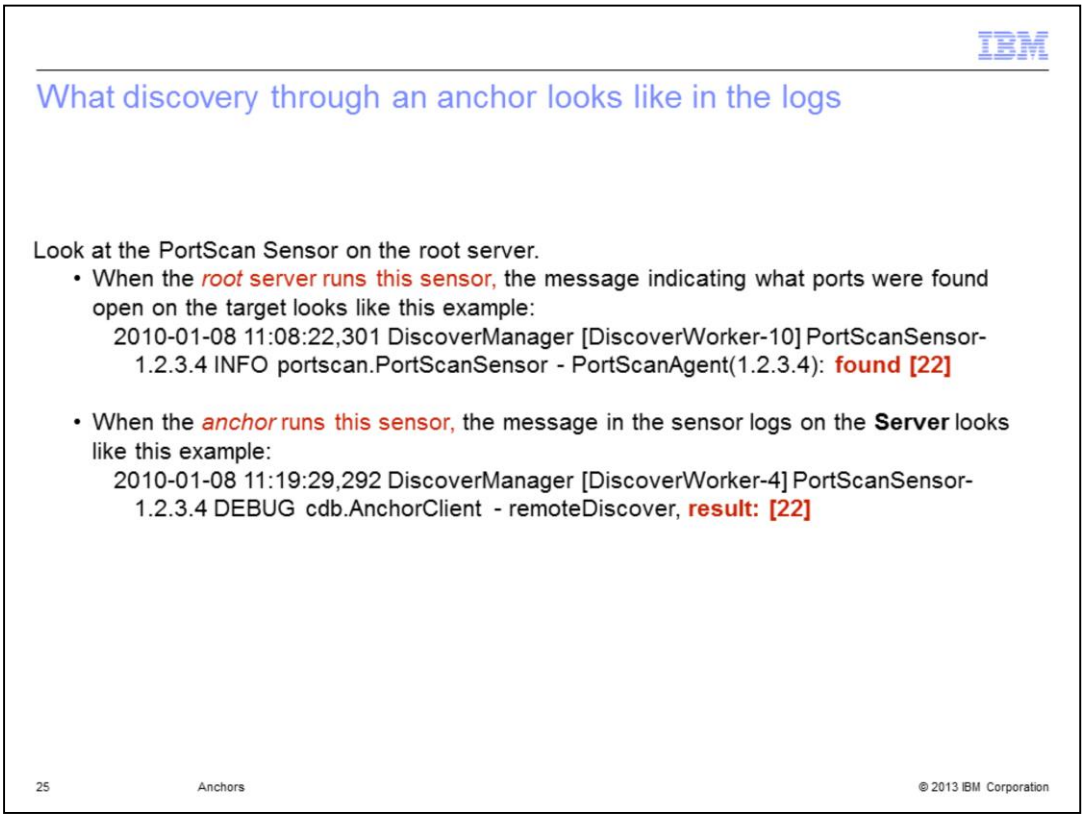


Examination of the port scan sensor logs can be helpful in determining which server performed the discovery. They can show failures or, more importantly, unexpected success, from the wrong root server or anchor, which can explain discovery results.

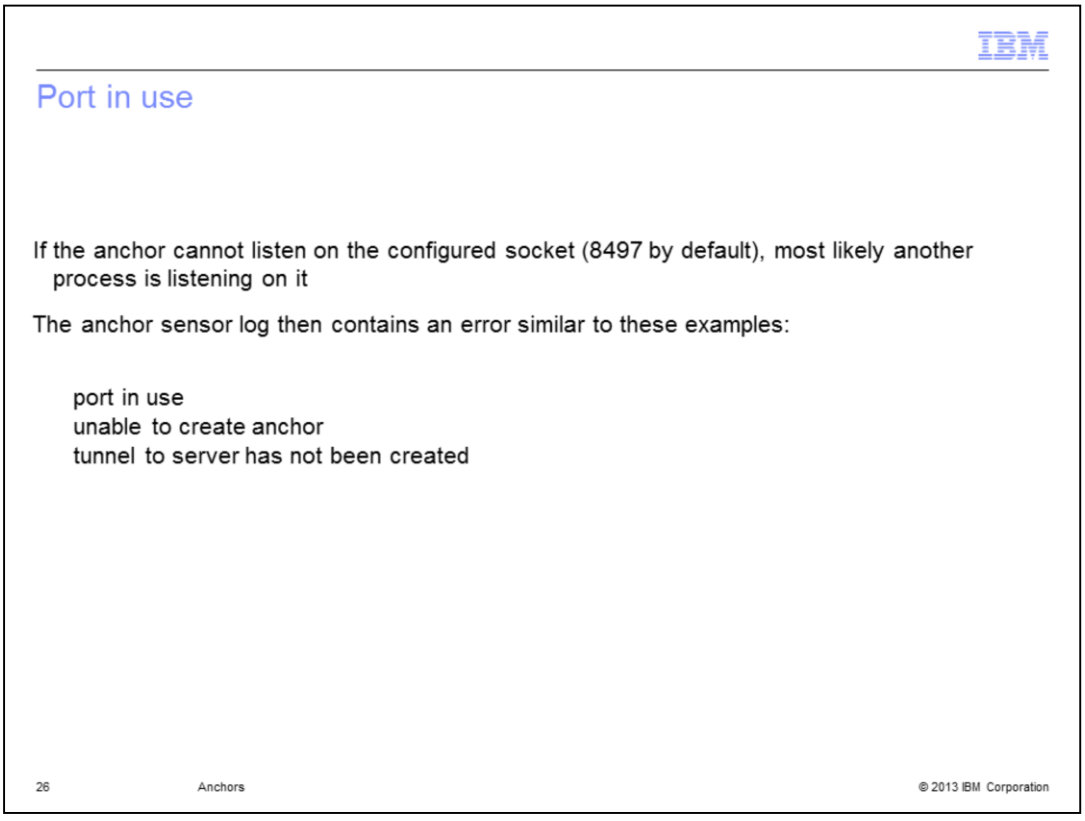


Having the SSH port open from the root server to all targets beyond a firewall, that is, not just the anchor, causes a problem if the root server then attempts to discover applications, such as WebSphere, that needs extra ports to communicate.

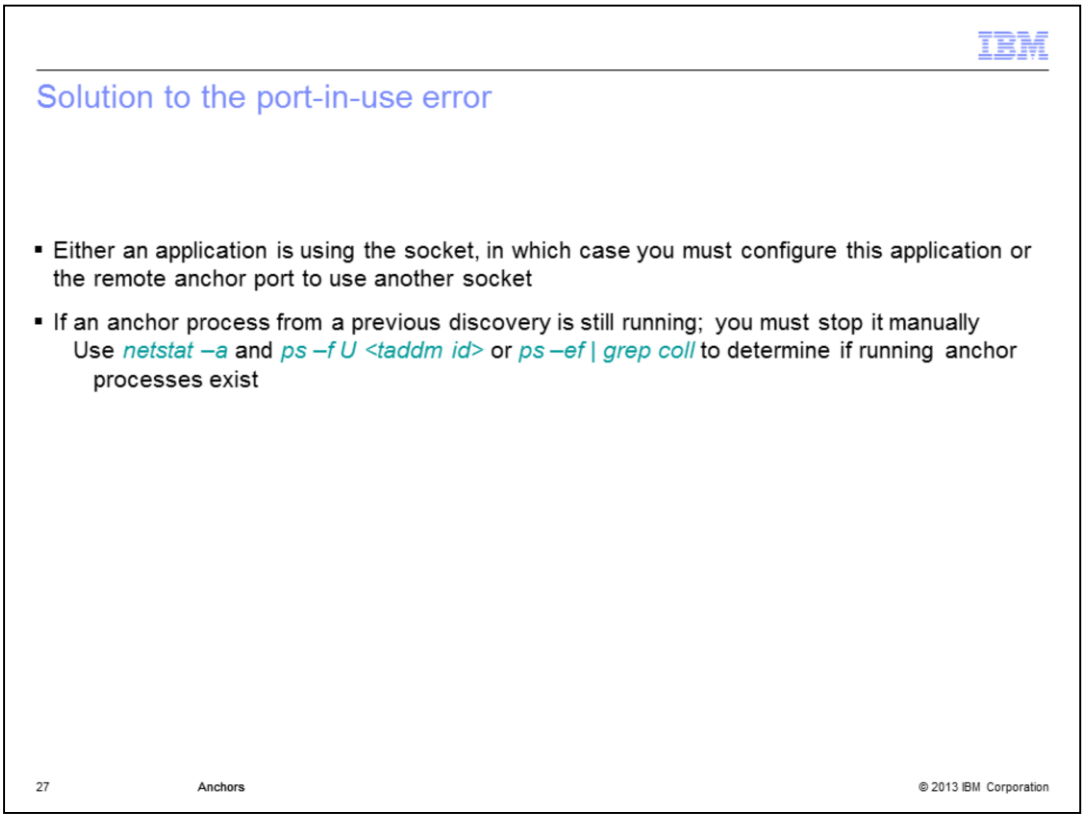
Telnet is useful in troubleshooting connectivity.



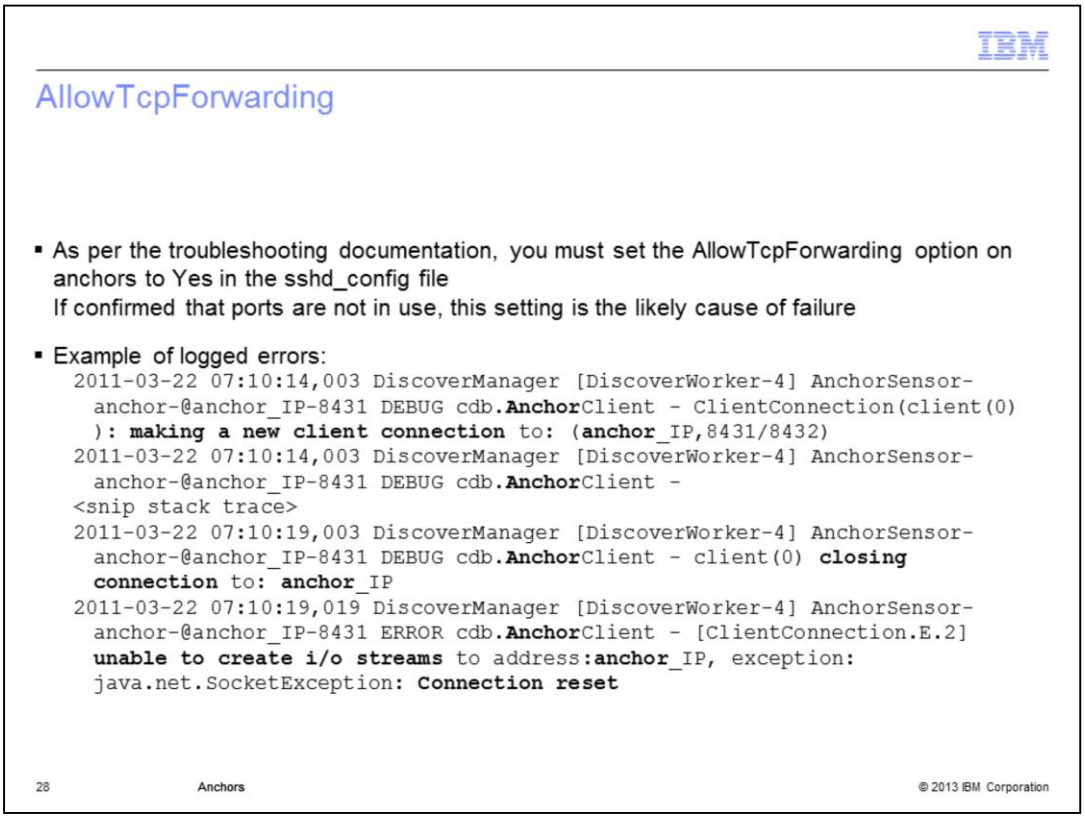
The PortScan sensor on the root server contains the message “found [X]” when port X is found available by the root server. When port X is found available by an anchor, the same log on the root server instead contains the message “result: [X]”.



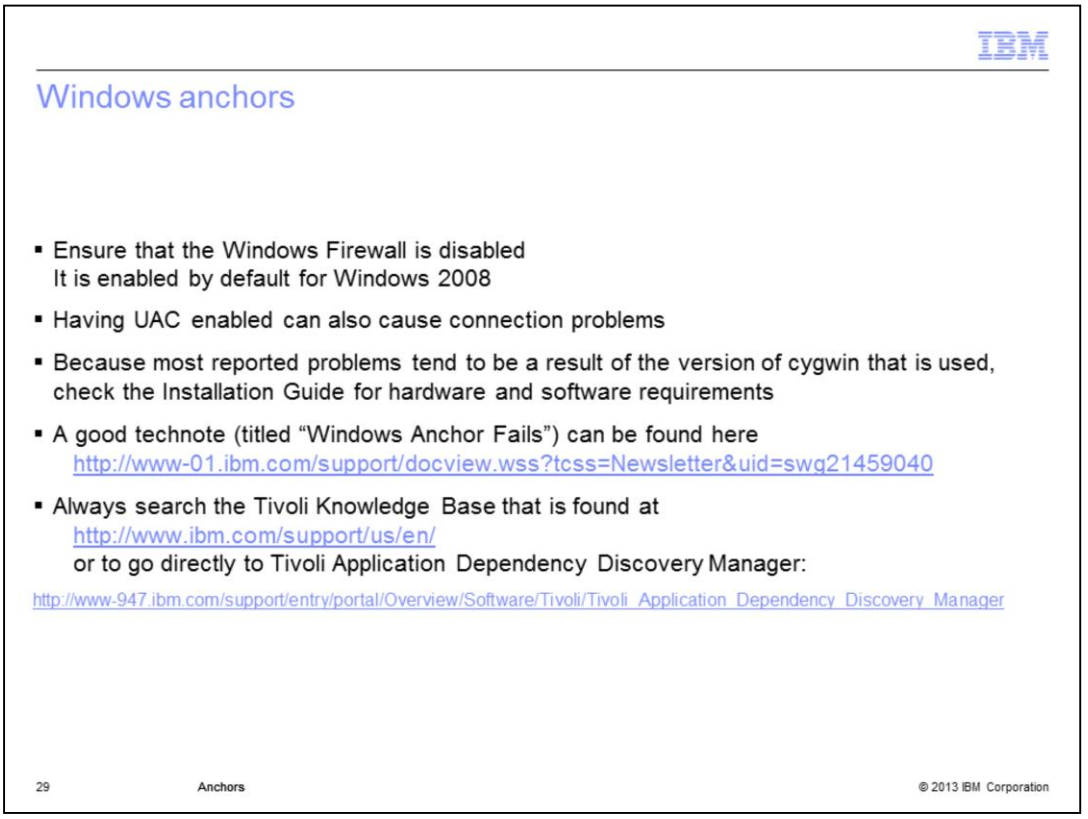
If the anchor cannot listen on its configured socket, the anchor sensor log on the root server contains a error message similar to “port in use,” “unable to create anchor,” or “tunnel to server has not been created.” To fix this issue, you must log on to the anchor and determine what process is using that port and why. Then take necessary actions to free the port.



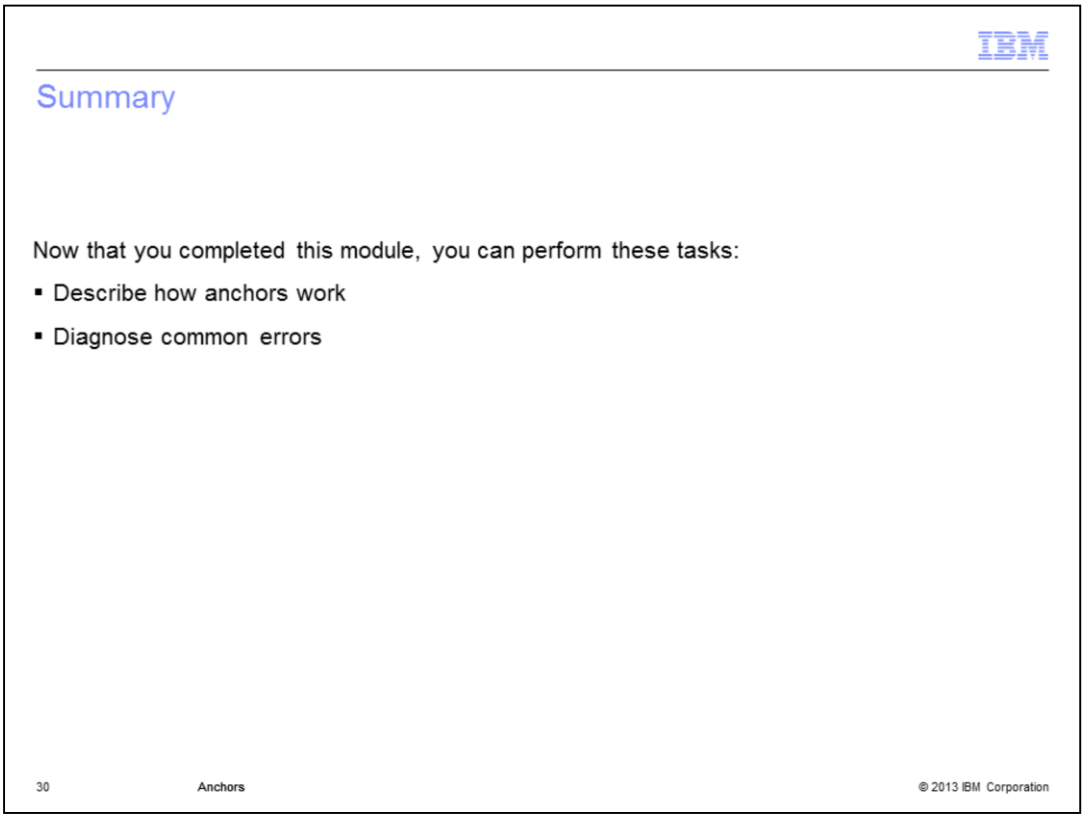
If another application is using the socket, either change that application or change the remote anchor port from the Data Management Console, as shown earlier in this presentation. If an anchor process is incorrectly still running from a previous discovery, stop the anchor manually. If this problem continues to happen, call IBM Support for further analysis.



As stated previously, and in the documentation, you must set the AllowTcpForwarding parameter to TRUE in the sshd_config file.



For problems with anchors on a Windows platform, if possible, test using a Linux box as an anchor to rule out if the issue is on the firewall or in the network configuration. Check the Installation Guide to confirm that proper hardware and software is being used.



Now that you completed this module, you can describe how Tivoli Application Dependency Discovery Manager anchors work, and you can diagnose common errors.

Trademarks, disclaimer, and copyright information