

◆ ◆ ◆ **2**
CHAPTER 2

Administering GlassFish Server Nodes

A *node* represents a host on which the GlassFish Server software is installed. A node must exist for every host on which GlassFish Server instances reside.

The following topics are addressed here:

- “Types of GlassFish Server Nodes” on page 27
- “Creating, Listing, Testing, and Deleting SSH Nodes” on page 28
- “Creating, Listing, and Deleting CONFIG Nodes” on page 32
- “Updating and Upgrading a Node” on page 35

Types of GlassFish Server Nodes

Each GlassFish Server node is one of the following types of node:

SSH

An SSH node supports communication over secure shell (SSH). If SSH is set up and you plan to administer your GlassFish Server instances centrally, the instances *must* reside on SSH nodes.

CONFIG

A CONFIG node does not support remote communication. If SSH is not set up and you plan to administer your instances locally, the instances can reside on CONFIG nodes. You *cannot* use CONFIG nodes for instances that you plan to administer centrally.

Each domain contains a predefined CONFIG node that is named `localhost - domain`, where *domain* is the name of the domain. On the host where the domain administration server (DAS) is running, this node represents the local host.

Creating, Listing, Testing, and Deleting SSH Nodes

An SSH node supports communication over SSH. If SSH is set up and you plan to administer your GlassFish Server instances centrally, the instances *must* reside on SSH nodes.

GlassFish Server enables you to create SSH nodes for use by instances, obtain information about SSH nodes, test if SSH nodes are usable, and delete SSH nodes that are no longer required.

The following topics are addressed here:

- “To Create an SSH Node” on page 28
- “To List SSH Nodes in a Domain” on page 29
- “To Test if an SSH Node is Usable” on page 30
- “To Delete an SSH Node” on page 31

▼ To Create an SSH Node

Use the `create-node-ssh` command in remote mode to create an SSH node.

Before You Begin Ensure that the user of the DAS can use SSH to log in to the host that the node will represent.

- 1 **Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 **Run the `create-node-ssh` subcommand.**

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for configuring the node, see the `create-node-ssh(1)` help page.

```
asadmin> create-node-ssh --nodehost node-host [--installdir install-dir ]
node-name
```

node-host

The name of the host that the node represents. The name of the host must be specified. Otherwise, an error occurs.

install-dir

The full path to the parent of the base installation directory of the GlassFish Server software on the host, for example, `/export/glassfish3/`. If the GlassFish Server software is installed in the same directory on the node's host and the DAS host, you can omit this option.

node-name

Your choice of name for the node that you are creating.

Example 2-1 Creating an SSH Node

This example creates the SSH node `sj01` to represent the host `sj01.example.com`. The GlassFish Server software is installed in the same directory on the DAS host and on the host `sj01.example.com`.

```
asadmin> create-node-ssh --nodehost sj01.example.com sj01
Command create-node-ssh executed successfully.
```

See Also [create-node-ssh\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help create-node-ssh` at the command line.

Next Steps After creating a node, you can create instances on the node as explained in the following sections:

- “To Create an Instance Centrally” on page 56
- “To Create an Instance Locally” on page 66

▼ To List SSH Nodes in a Domain

Use the `list-nodes-ssh` subcommand in remote mode to obtain information about existing SSH nodes in a domain.

Note – To obtain information about all existing nodes in a domain, use the [list-nodes\(1\)](#) subcommand.

1 **Ensure that the DAS is running.**

Remote subcommands require a running server.

2 **Run the [list-nodes-ssh\(1\)](#) subcommand.**

```
asadmin> list-nodes-ssh
```

Example 2-2 Listing Basic Information About All SSH Nodes in a Domain

This example lists the name, type, and host of all SSH nodes in the current domain.

```
asadmin> list-nodes-ssh
sj01  SSH  sj01.example.com
sj02  SSH  sj02.example.com
Command list-nodes-ssh executed successfully.
```

Example 2-3 Listing Detailed Information About All SSH Nodes in a Domain

This example lists detailed information about all SSH nodes in the current domain.

```
asadmin> list-nodes-ssh --long=true
NODE NAME   TYPE   NODE HOST           INSTALL DIRECTORY   REFERENCED BY
sj01        SSH    sj01.example.com    /export/glassfish3   pmd-i1
sj02        SSH    sj02.example.com    /export/glassfish3   pmd-i2
Command list-nodes-ssh executed successfully.
```

- See Also**
- [list-nodes\(1\)](#)
 - [list-nodes-ssh\(1\)](#)

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help list-nodes`
- `asadmin help list-nodes-ssh`

▼ To Test if an SSH Node is Usable

Use the `ping-node-ssh` subcommand in remote mode to test if an SSH node is usable.

Before You Begin Ensure that SSH is configured on the host where the DAS is running and on the host that the node represents.

- 1 **Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 **Run the `ping-node-ssh` subcommand.**

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for testing the node, see the [ping-node-ssh\(1\)](#) help page.

```
asadmin> ping-node-ssh node-name
node-name
The name of the node to test.
```

Example 2-4 Testing if an SSH Node Is Usable

This example tests if the SSH node `sj01` is usable.

```
asadmin> ping-node-ssh sj01
Successfully made SSH connection to node sj01 (sj01.example.com)
Command ping-node-ssh executed successfully.
```

See Also [ping-node-ssh\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help ping-node-ssh` at the command line.

▼ To Delete an SSH Node

Use the `delete-node-ssh` subcommand in remote mode to delete an SSH node.

Deleting a node removes the node from the configuration of the DAS. The node's directories and files are deleted when the last GlassFish Server instance that resides on the node is deleted.

Before You Begin Ensure that no GlassFish Server instances reside on the node that you are deleting. For information about how to delete an instance, see the following sections.

- [“To Delete an Instance Centrally” on page 60](#)
- [“To Delete an Instance Locally” on page 69](#)

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Confirm that no instances reside on the node that you are deleting.

```
asadmin> list-nodes-ssh --long=true
```

3 Run the `delete-node-ssh(1)` subcommand.

```
asadmin> delete-node-ssh node-name
```

node-name

The name of the node that you are deleting.

Example 2-5 Deleting an SSH Node

This example confirms that no instances reside on the SSH node `sj01` and deletes the node `sj01`.

```
asadmin> list-nodes-ssh --long=true
NODE NAME   TYPE   NODE HOST           INSTALL DIRECTORY   REFERENCED BY
sj01        SSH    sj01.example.com    /export/glassfish3
sj02        SSH    sj02.example.com    /export/glassfish3   pmd-i2
Command list-nodes-ssh executed successfully.
asadmin> delete-node-ssh sj01
Command delete-node-ssh executed successfully.
```

See Also

- [“To Delete an Instance Centrally” on page 60](#)
- [“To Delete an Instance Locally” on page 69](#)
- `delete-node-ssh(1)`

- [list-nodes-ssh\(1\)](#)

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help delete-node-ssh`
- `asadmin help list-nodes-ssh`

Creating, Listing, and Deleting CONFIG Nodes

A CONFIG node does not support remote communication. If SSH is not set up and you plan to administer your instances locally, the instances can reside on CONFIG nodes. You *cannot* use CONFIG nodes for instances that you plan to administer centrally.

GlassFish Server enables you to create CONFIG nodes for use by instances, obtain information about CONFIG nodes, and delete CONFIG nodes that are no longer required.

The following topics are addressed here:

- “To Create a CONFIG Node” on page 32
- “To List CONFIG Nodes in a Domain” on page 33
- “To Delete a CONFIG Node” on page 34

▼ To Create a CONFIG Node

Use the `create-node-config` command in remote mode to create a CONFIG node.

Note – If you create an instance locally on a host for which no nodes are defined, you can create the instance without creating a node beforehand. In this situation, GlassFish Server creates a CONFIG node for you. The name of the node is the unqualified name of the host. For more information, see “To Create an Instance Locally” on page 66.

- 1 Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 Run the `create-node-config` subcommand.**

Note – Only the options that are required to complete this task are provided in this step. For information about all the options for configuring the node, see the [create-node-config\(1\)](#) help page.

```
asadmin> create-node-config [--nodehost node-host] [--installdir install-dir ]
node-name
```

node-host

The name of the host that the node represents. You may omit this option. The name of the host can be determined when instances that reside on the node are created.

install-dir

The full path to the parent of the base installation directory of the GlassFish Server software on the host, for example, `/export/glassfish3/`. You may omit this option. The installation directory can be determined when instances that reside on the node are created.

node-name

Your choice of name for the node that you are creating.

Example 2-6 Creating a CONFIG Node

This example creates the CONFIG node `cfg01`. The host that the node represents and the installation directory of the GlassFish Server software on the host are to be determined when instances are added to the node.

```
asadmin> create-node-config cfg01
Command create-node-config executed successfully.
```

See Also [create-node-config\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help create-node-config` at the command line.

Next Steps After creating a node, you can create instances on the node as explained in “[To Create an Instance Locally](#)” on page 66.

▼ To List CONFIG Nodes in a Domain

Use the `list-nodes-config` subcommand in remote mode to obtain information about existing CONFIG nodes in a domain.

Note – To obtain information about all existing nodes in a domain, use the `list-nodes(1)` subcommand.

- 1 Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 Run the `list-nodes-config(1)` subcommand.**

```
asadmin> list-nodes-config
```

Example 2-7 Listing Basic Information About All CONFIG Nodes in a Domain

This example lists the name, type, and host of all CONFIG nodes in the current domain.

```
asadmin> list-nodes-config
localhost-domain1 CONFIG localhost
cfg01 CONFIG cfg01.example.com
cfg02 CONFIG cfg02.example.com
Command list-nodes-config executed successfully.
```

Example 2-8 Listing Detailed Information About All CONFIG Nodes in a Domain

This example lists detailed information about all CONFIG nodes in the current domain.

```
asadmin> list-nodes-config --long=true
NODE NAME          TYPE      NODE HOST          INSTALL DIRECTORY  REFERENCED BY
localhost-domain1 CONFIG    localhost          /export/glassfish3
cfg01              CONFIG    cfg01.example.com /export/glassfish3 yml-i1
cfg02              CONFIG    cfg02.example.com /export/glassfish3 yml-i2
Command list-nodes-config executed successfully.
```

- See Also**
- [list-nodes\(1\)](#)
 - [list-nodes-config\(1\)](#)

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help list-nodes`
- `asadmin help list-nodes-config`

▼ To Delete a CONFIG Node

Use the `delete-node-config` subcommand in remote mode to delete a CONFIG node.

Deleting a node removes the node from the configuration of the DAS. The node's directories and files are deleted when the last GlassFish Server instance that resides on the node is deleted.

Before You Begin Ensure that no GlassFish Server instances reside on the node that you are deleting. For information about how to delete an instance that resides on a CONFIG node, see [“To Delete an Instance Locally”](#) on page 69.

- 1 Ensure that the DAS is running.**
Remote subcommands require a running server.
- 2 Confirm that no instances reside on the node that you are deleting.**

```
asadmin> list-nodes-config --long=true
```


3 Run the `delete-node-config(1)` subcommand.

```
asadmin> delete-node-config node-name
```

node-name

The name of the node that you are deleting.

Example 2-9 Deleting a CONFIG Node

This example confirms that no instances reside on the CONFIG node `cfg01` and deletes the node `cfg01`.

```
asadmin> list-nodes-config --long=true
NODE NAME          TYPE      NODE HOST          INSTALL DIRECTORY  REFERENCED BY
localhost-domain1 CONFIG    localhost          /export/glassfish3
cfg01              CONFIG    cfg01.example.com /export/glassfish3
cfg02              CONFIG    cfg02.example.com /export/glassfish3  yml-i2
Command list-nodes-config executed successfully.
asadmin> delete-node-config cfg01
Command delete-node-config executed successfully.
```

- See Also**
- [“To Delete an Instance Locally” on page 69](#)
 - [`delete-node-config\(1\)`](#)
 - [`list-nodes-config\(1\)`](#)

You can also view the full syntax and options of the subcommands by typing the following commands at the command line:

- `asadmin help delete-node-config`
- `asadmin help list-nodes-config`

Updating and Upgrading a Node

GlassFish Server enables you to update the configuration data of any node and to upgrade a CONFIG node to an SSH node.

The following topics are addressed here:

- [“To Update an SSH Node” on page 35](#)
- [“To Update a CONFIG Node” on page 36](#)
- [“To Upgrade a CONFIG Node to an SSH Node” on page 37](#)

▼ To Update an SSH Node

Use the `update-node-ssh` subcommand in remote mode to update an SSH node.

Options of this subcommand specify the new values of the node's configuration data. If you omit an option, the existing value is unchanged.

Before You Begin Ensure that the following prerequisites are met:

- SSH is configured on the host where the DAS is running and on the host that the node represents.
- The node that you are updating exists.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `update-node-ssh` subcommand.

```
asadmin> update-node-ssh options node-name
```

options

Options of the `update-node-ssh` subcommand for changing the node's configuration data. For information about these options, see the [update-node-ssh\(1\)](#) help page.

node-name

The name of the SSH node to update.

Example 2-10 Updating an SSH Node

This example updates the host that the node `sj01` represents to `adc01.example.com`.

```
asadmin> update-node-ssh --nodehost adc01.example.com sj01
Command update-node-ssh executed successfully.
```

See Also [update-node-ssh\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help update-node-ssh` at the command line.

▼ To Update a CONFIG Node

Use the `update-node-config` subcommand in remote mode to update a CONFIG node.

Options of this subcommand specify the new values of the node's configuration data. If you omit an option, the existing value is unchanged.

Before You Begin Ensure that the node that you are updating exists.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `update-node-config` subcommand.

```
asadmin> update-node-config options node-name
```

options

Options of the `update-node-config` subcommand for changing the node's configuration data. For information about these options, see the [update-node-config\(1\)](#) help page.

node-name

The name of the CONFIG node to update.

Example 2–11 Updating a CONFIG Node

This example updates the host that the node `cfg02` represents to `adc02.example.com`.

```
asadmin> update-node-config --nodehost adc02.example.com cfg02
Command update-node-config executed successfully.
```

See Also [update-node-config\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help update-node-config` at the command line.

▼ To Upgrade a CONFIG Node to an SSH Node

Use the `update-node-ssh` subcommand in remote mode to upgrade a CONFIG node to an SSH node.

Upgrading a CONFIG node to an SSH node enables SSH communication for the node. As part of the upgrade process, you can also change other configuration data for the node.

Options of the `update-node-ssh` subcommand specify the new values of the node's configuration data. For most options, if you omit the option, the existing value is unchanged. However, default values are applied if any of the following options is omitted:

- `--sshport`
- `--sshuser`
- `--sshkeyfile`

Before You Begin Ensure that the following prerequisites are met:

- SSH is configured on the host where the DAS is running and on the host that the node represents.
- The node that you are updating exists.

1 Ensure that the DAS is running.

Remote subcommands require a running server.

2 Run the `update-node-ssh` subcommand on the CONFIG node that you are updating.

```
asadmin> update-node-ssh [options] node-name
```

options

Options of the `update-node-ssh` subcommand for changing the node's configuration data. For information about these options, see the [update-node-ssh\(1\)](#) help page.

node-name

The name of the CONFIG node to upgrade.

Example 2–12 **Upgrading a CONFIG Node to an SSH Node**

This example upgrades the CONFIG node `cfg02` to an SSH node .

```
asadmin> update-node-ssh cfg02  
Command update-node-ssh executed successfully.
```

See Also [update-node-ssh\(1\)](#)

You can also view the full syntax and options of the subcommand by typing `asadmin help update-node-ssh` at the command line.