

Name create-node-ssh – creates a node that is enabled for communication over SSH

Synopsis create-node-ssh [--help]
 --nodehost *node-host*
 [--installdir *install-dir*] [--nodedir *node-dir*]
 [--sshport *ssh-port*] [--sshuser *ssh-user*]
 [--sshkeyfile *ssh-keyfile*]
 [--force={false|true}]
node-name

Description The create-node-ssh subcommand creates a node that is enabled for communication over secure shell (SSH).

A node represents a host on which the GlassFish Server software is installed. A node must be present on every host on which GlassFish Server instances reside.

The domain administration server (DAS) contacts a node through the SSH connector to manage GlassFish Server instances that reside on the node. However, the DAS does not use the SSH connector to contact a node on the host where the DAS is running because the DAS can run all `asadmin` subcommands locally.

By default, the subcommand fails and the node is not created if the DAS cannot contact the node's host through SSH. To force the node to be created even if the host cannot be contacted through SSH, set the `--force` option to `true`.

This subcommand is supported in remote mode only.

Options --help
 -?

Displays the help text for the subcommand.

--nodehost

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

--installdir

The full path to the parent of the base installation directory of the GlassFish Server software on the host, for example, `/export/glassfish3/`. The default is the parent of the default base installation directory of the GlassFish Server software. This default is useful only if GlassFish Server is installed in the same location on all hosts.

--nodedir

The full path to the directory that is to contain GlassFish Server instances that are created on the node. The default is `as-install/nodes`, where `as-install` is the base installation directory of the GlassFish Server software on the host.

--sshport

The port to use for SSH connections to this node. The default is 22. If the `--nodehost` is set to `localhost`, the `--sshport` option is ignored.

--sshuser

The SSH user that is to run the process for connecting to this node. The default is the user that is running the DAS process. To ensure that the DAS can read this user's SSH private key file, specify the user that is running the DAS process. If the `--nodehost` option is set to `localhost`, the `--sshuser` option is ignored.

--sshkeyfile

The absolute path to the SSH private key file for user that the `--sshuser` option specifies. This file is used for authentication to the `sshd` daemon on the node.

The path to the key file must be reachable by the DAS and the key file must be readable by the DAS. The path may contain Java properties of the form `${prop.name}`.

The default is the a platform-dependent key file in the user's `.ssh` directory, for example:

- `id_rsa`
- `id_dsa`
- `identitylocation`

--force

Specifies whether the node is created even if validation of the node's parameters fails. To validate a node's parameters, the DAS must be able to contact the node's host through SSH. Possible values are as follows:

`false`

The node is not created if validation of the node's parameters fails (default).

`true`

The node is created even if validation of the node's parameters fails.

Operands *node-name*

The name of the node. The name of the node must be unique in the domain and must not be the name of a cluster, a named configuration, a GlassFish Server instance, or another node.

Examples This example creates the node `adc` on the host `adc.example.com`. The parent of the base installation directory of the GlassFish Server software is `/export/glassfish3`.

EXAMPLE 1 Creating a Node

```
asadmin> create-node-ssh
--nodehost adc.example.com
--installdir /export/glassfish3 adc
```

Command `create-node-ssh` executed successfully.

EXAMPLE 2 Forcing the Creation of a Node

This example forces the creation of node `eg1` on the host `eghost.example.com`. The node is created despite the failure of the DAS to contact the host `eghost.example.com` to validate the node's parameters.

```
asadmin> create-node-ssh --force --nodehost eghost.example.com eg1
Warning: some parameters appear to be invalid.
Could not connect to host eghost.example.com using SSH.
There was a problem while connecting to eghost.example.com:22
eghost.example.com
Continuing with node creation due to use of --force.
```

Command `create-node-ssh` executed successfully.

Exit Status	0	command executed successfully
	1	error in executing the command

See Also [delete-node-ssh\(1\)](#), [install-node\(1\)](#), [list-nodes\(1\)](#), [ping-node-ssh\(1\)](#), [setup-ssh\(1\)](#), [uninstall-node\(1\)](#), [update-node-ssh\(1\)](#)

[asadmin\(1M\)](#)