VMS Software

OpenSSH Version 9.9-2C for VSI OpenVMS

Release Notes and Installation Guide

Publication Date: May 2026

Operating Systems: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-2L3
VSI OpenVMS x86-64 Version 9.2-3 + Update V1 or higher

Kit Names: VSI-AXPVMS-OPENSSH-V0909-2C-1.PCSI
VSI-I64VMS-OPENSSH-V0909-2C-1.PCSI
VSI-X86VMS-OPENSSH-V0909-2C-1.PCSI

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

Table of Contents

1. Introduction

Thank you for your interest in this port of OpenSSH to VSI OpenVMS Alpha, IA-64, and x86-64. The current release of OpenSSH for OpenVMS is based on the OpenSSH 9.9p2 distribution.

OpenSSH (https://www.openssh.com/) is an Open Source (BSD licensed) suite of secure networking utilities based on the Secure Shell (SSH) protocol, which provides a secure channel over a potentially unsecured network. OpenSSH is a complete implementation of the SSH protocol (version 2) for secure remote login, command execution, and file transfer. It includes SSH client and server components, file transfer utilities SCP and SFTP, as well as the tools for key generation, run-time key storage, and a number of other supporting programs.

This port of OpenSSH to VSI OpenVMS Alpha, IA-64, and x86-64 is based on the Portable OpenSSH distribution (see https://github.com/openssh/openssh-portable), which is a port of OpenBSD's OpenSSH implementation commonly used on Linux, OS X, and Cygwin.

Acknowledgement

VMS Software Inc. would like to acknowledge the work of the Portable OpenSSH development team for their ongoing efforts in developing and supporting this software.

2. Release Notes

The OpenSSH Version 9.9-2C release fixes the issue encountered when using password authentication to execute a remote command via SSH.

3. Known Problems and Restrictions

- Using wildcard characters with OpenVMS-style path names may cause various issues.
- Restarting TCP/IP Services breaks the OpenSSH Server (OpenSSH Client is not affected). To fix that, you need to restart OpenSSH Server manually.
- Only the password, public-key, and host-based authentication methods are currently supported. Additional methods (such as Kerberos) may be added in the future.
- There must be no spaces used in the OpenSSH configuration parameters, otherwise the connection will fail.
- During installation, OpenSSH creates two user accounts – SSH$SSH and SSH$SSHD. These accounts are required for OpenSSH to function correctly and must never be removed.
- Some of the existing third-party SFTP clients may treat filenames as case-sensitive (and as such, **`LS A.*`** may give a different result than **`LS a.*`**).
- Transferring files between OpenVMS systems using SCP or SFTP between OpenSSH and any other SSH implementation on OpenVMS may result in incorrect transfer of OpenVMS specific file attributes.

4. Installation and Configuration

This section walks you through the tasks that you need to perform to be able to use OpenSSH V9.9-2C for VSI OpenVMS.

4.1. Before You Install

Before installing and using OpenSSH, it is recommended that users read the documentation available at https://www.openssh.com/manual.html in order to better understand how to configure and use the software.

4.2. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. Note that while you probably will not have any problems installing and using this kit on systems running higher versions of the operating system and/or products listed below, running older versions may cause problems.

- One of these operating systems:
  - VSI OpenVMS Alpha V8.4-2L1 with the VMS842L1I_RTL-V1000 (or higher) patch installed.
  - VSI OpenVMS IA-64 V8.4-2L3 with the VMS842L3I_RTL-V1000 (or higher) patch installed.

    Notes:

    The RTL ECO kit mentioned above requires its respective DPML V0200 ECO to be installed first.

    If you are using the Multinet TCP/IP stack, know that RTL V1000 is incompatible with Samba.

  - VSI OpenVMS x86-64 V9.2-3 with the VMS923X_UPDATE-V0100 patch installed
- VSI OpenVMS RTL V10 patch kit
- VSI SSL3 V3.0-18 or later
- If your system has OpenSSH V8.9-1F for VSI OpenVMS or earlier installed, it *must* be uninstalled before installing OpenSSH V9.9-2C for VSI OpenVMS. To uninstall the previous version of VSI OpenSSH, perform the following procedure:

  1. Enter the command **`PRODUCT REMOVE OPENSSH`**
  2. You will get the following error:

     ```
     %PCSI-I-SPAWNEXE, error executing:
     @PCSI$DESTINATION:[OPENSSH.BIN]SSH$RUN_CLEANUP_PROCEDURE.COM
     %PCSI-E-EXERMVFAIL, product supplied EXECUTE REMOVE
     procedure failed
     -RMS-E-FNF, file not found
     %PCSI-E-OPFAILED, operation failed
     Terminating is strongly recommended. Do you want to
     terminate? [YES]
     ```

     Answer **`NO`** to the `Do you want to terminate?` question.
  3. Once OpenSSH has been removed, you will get the following message:

     ```
     %PCSIUI-I-COMPWERR, operation completed after explicit continuation
      from errors
     ```

4.3. Installing the Kit

Note

Do *not* use the **/DESTINATION** qualifier with the **PRODUCT INSTALL** command when installing OpenSSH for VSI OpenVMS to specify an alternative (non-default) installation location. VSI OpenVMS includes OpenSSH components bundled with the operating system, which imposes specific requirements in terms of location of these components and associated configuration files.

This kit is provided as an OpenVMS PCSI kit that can be installed by a suitably privileged user running the following command:

```
$ PRODUCT INSTALL OPENSSH
```

The installation will then proceed as follows. Note that the output may differ slightly from that shown below depending on the platform and other factors.

```
The following product has been selected:
    VSI I64VMS OPENSSH V9.9-2C             Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product and for
any products that may be installed to satisfy software dependency requirements.

Configuring VSI I64VMS OPENSSH V9.9-2C: VSI OpenVMS OpenSSH

    The VSI OpenVMS OpenSSH is a complete implementation of the SSH
    protocol (version 2) for secure remote login, command execution,
    and file transfer. It includes client and server parts (ssh
    and sshd), file transfer utilities (scp and sftp), as well as
    run-time key storage (ssh_agent), tools for generating
    authentication keys (ssh_keygen), and a number of supporting
    programs.


    Copyright 2026 VMS Software, Inc.

    OpenVMS OpenSSH is released under a BSD licence, or a licence more free than that.

    OpenVMS OpenSSH contains no GPL code.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI I64VMS OPENSSH V9.9-2C             DISK$IV842L3SYS:[VMS$COMMON.]

Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%

   User Accounts and User Identification Codes (UICs)
   --------------------------------------------------

    The OpenVMS OpenSSH installation creates two OpenVMS accounts:
    SSH$SSH, SSH$SSHD. The default UIC group number for these new
    accounts depends on the following:

    o If you are installing the server for the first time, the
      default is the first unused UIC group number, starting
      with 3655.

    o If any of these accounts already exists, then the default
```

```
      UIC group number will not be used to change the UIC of
      any existing accounts.

    o If old account TCPIP$SSH already exists, then the default
      UIC group number will be used from TCPIP$SSH account.

    For more information about UIC group numbers, see the
    OpenVMS System Manager's Manual.

Enter default UIC group number for OpenSSH account
Group: [3655]
Creating OpenVMS account required by OpenSSH
SSH$SSH account already exists
SSH$SSHD account already exists
SSH$ROOT is defined as "BALIN$DKA0:[SYS0.SYSCOMMON.OPENSSH.]"
Setting file protections...
  File protections are set

The OpenSSH configuration files were saved in the directory:
  SYS$COMMON:[SYSUPD.SSH$SAFETY]

Should it restore OpenSSH configuration files? [y/n]: [y]
Creating OpenSSH for OpenVMS root definition file SYS$COMMON:[SYS$STARTUP]SSH
$DEFINE_ROOT.COM...
  File created
Save startup files
Setup OpenSSH logical environment

Generating public/private keys:
  ssh_host_dsa_key. and ssh_host_dsa_key.pub are already present.
  ssh_host_ecdsa_key. and ssh_host_ecdsa_key.pub are already present.
  ssh_host_rsa_key. and ssh_host_rsa_key.pub are already present.
  ssh_host_ed25519_key. and ssh_host_ed25519_key.pub are already present.


Do you want to migrate your old SSH settings? [y/n]: [n]
  User canceled this operation!
  BY DEFAULT, THE CONNECTION PORT WILL SET TO 222!
  You can change this value now.
  Port: [222]
  You can use migration tool manually, use
  $ @ssh$root:[bin]ssh$migration.com
  Work log see in ssh$root:[var]ssh$migration_29jan2025_093238.log

Successfully finished

   In a cluster, on all the nodes that are going to use common
   ssh$root installation directory as the current node, copy
   the following files to SYS$STARTUP directory of each node:

        SYS$STARTUP:SSH$STARTUP.COM
        SYS$STARTUP:SSH$SHUTDOWN.COM
        SYS$STARTUP:SSH$DEFINE_ROOT.COM

   To automatically start OpenVMS OpenSSH during system startup
   add the following line to the file SYS$MANAGER:SYSTARTUP_VMS.COM
   after the TCPIP startup command procedure:

        $ @SYS$STARTUP:SSH$STARTUP.COM

   Define symbols for all OpenSSH utilities:

        $ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM

   To have all symbols defined by the time of login,
   add a caller line of SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM file in either:
        1. in SYS$MANAGER:SYLOGIN.COM (note that you need system priv's for this)
        2. in SYS$LOGIN:LOGIN.COM
...100%
```

```
The following product has been installed:
    VSI I64VMS OPENSSH V9.9-2C            Layered Product
```

4.4. Post-Installation Steps

After the installation has successfully completed, follow these steps:

1. Start OpenSSH SSH server by executing the following command:

   ```
   $ @SYS$STARTUP:SSH$STARTUP
   ```

   The name of the TCP/IP service for the OpenSSH SSH server is `SSHD`*`port`*, where *`port`* is the TCP port number the OpenSSH server is configured to listen on. For example, if OpenSSH is configured to listen on TCP port 222, the service name is `SSHD222`.

2. Include the commands displayed at the end of the installation procedure into the SYSTARTUP_VMS.COM file to ensure that OpenSSH components are correctly started when OpenVMS is booted. Details regarding the migration procedure initiated at the end of the installation are provided below.

3. To have all necessary symbols defined by the time of login, add the following command to either the system-wide login procedure (SYS$MANAGER:SYLOGIN.COM) or a user's LOGIN.COM procedure:

   ```
   $ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM
   ```

Note

If you plan to allow users with SSH connected sessions to shutdown the system, do not invoke the SSH$SHUTDOWN.COM from within the site-specific SYS$MANAGER:SYSHUTDWN.COM procedure. The SSH shutdown will delete the process executing the shutdown and leave the system in an inconsistent state with logins and SSH disabled, along with some other parts of the system shutdown, but nothing left running to complete the shutdown or reboot.

Note

The command procedure SSH$ROOT:[BIN]SSH$DEASSIGN_COMMANDS.COM can be used to un-define these command symbols.

If **`SSH$DEFINE_COMMANDS.COM`** is run with the parameter ALL the following additional commands will be defined. These commands are intended primarily for administrative purposes:

- **`SSHSTART`**

  Starts and creates (if necessary) the OpenSSH services. Before running this command, check the file SSH$ROOT:[ETC]SSHD_CONFIG. to ensure that the SSH server configuration details are correct. If you would like to modify the client configuration, edit the SSH$ROOT:[ETC]SSH_CONFIG. file *before* starting the services.

- **`SSHSTO*P`**

  Stops OpenSSH services. If the parameter ALL is specified, the service definitions will also be deleted from the TCP/IP configuration.

- **`SSHSH*OW`**

Show details of running OpenSSH processes including SSH connections, number of connected clients, etc. Note that each client connection consists of two processes, namely a process named `SSHD_BG`*`xxxxx`* (where *`xxxxx`* is the number of the associated BG device) and a user process with a name that either matches the username or begins with the string `FTA`*`xxx`*`_` followed by the username (for example, `FTA110_SMITH`). The name of the user process may of course be changed by the user.

- **`SSHVER*SION`**

  Displays the information about the various OpenSSH programs, including version details and other related data.

Note

When the TCP/IP service is restarted, the OpenSSH server may become unresponsive and stop accepting new connections. To restore the OpenSSH server after a TCP/IP restart, manually restart OpenSSH by executing the following command:

```
$ @SYS$STARTUP:SSH$STARTUP
```

4.5. Migration

Note

This section is applicable to Alpha and IA-64 only. No actions will be performed if the migration tool is run on OpenVMS x86-64, and the tool will exit with a message indicating that no existing old TCP/IP Services SSH configuration was found. Similarly, the migration tool does not need to be run on Alpha or IA-64 if you have run it previously and are upgrading to a new version of OpenSSH for VSI OpenVMS.

As noted above, this release of OpenSSH for VSI OpenVMS includes a migration script (SSH$ROOT:[BIN]SSH$MIGRATION.COM) that can be used to convert configuration files and user public/private keys from the format used by VSI TCP/IP Services to the format expected by OpenSSH. After installing OpenSSH for VSI OpenVMS, this tool can be run to establish an initial OpenSSH system configuration that is comparable to that provided by the existing VSI TCP/IP Services.

The migration tool is run at the end of the OpenSSH kit installation. The user is prompted as to whether they wish to perform the migration at this time, the default being NO. In general, it is recommended that the migration be performed manually as a post-installation activity. Note that if OpenSSH is already installed or the VSI TCP/IP SSH service does not exist, the migration tool will *not* run.

Specific features of the migration facility are summarized as follows:

- The migration tool does not modify your old VSI TCP/IP Services files, making it possible to revert if necessary (migration is non-destructive).
- The tool creates a log file in SSH$ROOT:[VAR] containing the details of all migration activities. The name of the log file is SSH$MIGRATION_*xxxx*.LOG, where *xxxx* is replaced by the date and time at which the migration was performed.
- The following VSI TCP/IP Services configuration files will be examined and converted. As noted previously, the existing TCP/IP Services configuration files will not be changed.

  - TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSHD2_CONFIG.
  - TCPIP$SSH_DEVICE:[TCPIP$SSH.SSH2]SSH2_CONFIG.

- Users' public/private keys (RSA, DSA) can be converted optionally. The migration tool assumes that for a given username these files will reside in [.SSH2] under the user's login directory, and the converted keys will be written to [.SSH]. The first key defined in a user's [.SSH2] identification file is renamed to ID_RSA as the ID_RSA is one of the filenames used for public keys by default, per an IdentityFile entry in SSH_CONFIG. If there is an existing [.SSH]AUTHORIZED_KEYS. file, no conversion will be performed.

- When users run the migration script, they can choose a port that will be used for the OpenSSH server. For the client, the default port is 22. If the specified port is used during the migration, the script will prompt the user for another port.

The following brief notes illustrate how to run the migration tool to perform migration tasks or to revert to your old VSI TCP/IP Services configuration. Note that when running the migration tool, it is recommended that you not be logged into the OpenVMS system via VSI TCP/IP Services SSH.

- The migration tool can be run as follows to create an OpenSSH configuration from existing VSI TCP/IP Services configuration files:

```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM
```

- Running the following command will revert the system to using the old VSI TCP/IP Services configuration (assuming the old configuration has not otherwise been removed). This command will delete the OpenSSH SSH service and revert to the VSI TCP/IP Services SSH service:

```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM REVERT
```

- Public/private keys can be converted to OpenSSH format for a specified username using the following command:

```
$ @SSH$ROOT:[BIN]SSH$MIGRATION.COM "" <USERNAME>
```

- Conversion of a single key file can be performed as follows:

```
$ @SSH$ROOT:[BIN]SSH$DEFINE_COMMANDS.COM
$ PIPE SSH_KEYGEN "-I" "-F" [.SSH2]FILENAME > [.SSH]FILENAME
$ SET FILE/OWNER=<USER UIC> /PROTECTION=(G:"",W:"") [.SSH]FILENAME
```

The following table summarizes the parameter conversions that are performed by the migration tool for the SSHD2_CONFIG. and SSH2_CONFIG. configuration files (for additional details regarding the configuration parameters, see the next section).

VSI TCP/IP Services	**OpenSSH**
AccountingAuthentications	VmsAccountingAuthentications
IntrusionAuthentications	VmsIntrusionAuthentications
IntrusionIdentMethod	VmsIntrusionIdentMethods
IntrusionIdentSsh	VmsIntrusionIdentSsh
LogFailAuthentications	VmsLogFailAuthentications

VSI TCP/IP Services	OpenSSH
UserLoginLimit	VmsUserLoginLimit
AllowVmsLoginWithExpiredPw no AllowNonVmsLoginWithExpiredPw no	VmsAllowLoginWithExpiredPw no
NumberOfPasswordVerificationPrompts	VmsNumberOfPasswordVerificationPrompts
PrintSysAnnounce	VmsPrintSysAnnounce
PrintSysWelcome	VmsPrintSysWelcome
DisallowSftpServer	VmsDisallowSftpServer
SftpDenyUsers	VmsSftpDenyUsers
MaxConnections	MaxSessions
KeepAlive	TcpKeepAlive
BannerMessageFile	Banner
VerboseMode yes	LogLevel VERBOSE
UserKnownHosts no	IgnoreUserKnownHosts yes
AllowedAuthentications publickey,hostbased,password	HostBasedAuthentication yes PubkeyAuthentication yes PasswordAuthentication yes
Ciphers AnyStdCipher	Ciphers none
MACs AnyStdMAC	MACs none

4.6. Configuration Parameters

This section describes the configuration parameters unique to OpenVMS that can be used in OpenSSH client and server configuration files.

4.6.1. Server Configuration Parameters

The following parameters may be defined in SSH$ROOT:[ETC]SSHD_CONFIG. to control various aspects of SSH server operation with regard to maximum sessions, authentication, audit logging, and intrusions.

`VmsUserLoginLimit`

This parameter can be used to specify the maximum number of SSH clients that can be logged into the OpenVMS system. The default value is -1 (not limited); the maximum permitted value is 8192.

`VmsNumberOfPasswordVerificationPrompts`

This parameter can be used to specify the maximum number of password change attempts (the number of times that the user will be prompted to verify their new password). The default value is 3.

`VmsAllowLoginWithExpiredPw`

Setting this parameter to `yes` (the default) allows users to change their password if the password has expired and the user is connecting from an OpenVMS system. Permitted values for this parameter are `yes` and `no`.

VmsPrintSysAnnounce

Setting this parameter to `yes` (the default) causes the OpenVMS welcome banner associated with the logical name SYS$ANNOUNCE to be displayed when logging in. Permitted values for this parameter are `yes` and `no`.

VmsPrintSysWelcome

Setting this parameter to `yes` (the default) causes the welcome banner associated with the logical name SYS$WELCOME to be displayed when logging in. The permitted values for this parameter are `yes` and `no`.

VmsAccountingAuthentications

Generates an accounting record for all authentications via the specified authentication method(s) (`publickey`, `password`, and `hostbased`). The default value for this parameter is `publickey,password,hostbased`.

VmsIntrusionAuthentications

Reports users as intruders if they attempt and fail to connect using any one of the specified authentication method or methods. The default value for this parameter is `publickey,password,hostbased`, such that all authentication failures will be reported as intrusions.

VmsIntrusionAddServerAddress

Adds address details to the audit message. For example, `SSH_`*`authentication method`*`:`*`client ip-address`*`:`*`server ip-address`*. The default value is `no`.

VmsIntrusionIdentMethods

Specifying this parameter with a value comprising one or more authentication methods causes intrusion records pertaining to those authentication methods to specify the authentication method in addition to the IP address. Specifically, intrusion records will contain strings of the form `SSH_`*`authentication method`*`:`*`ip-address`*. The default value for this parameter is `publickey,password,hostbased`.

VmsIntrusionIdentSsh

If this parameter is specified, only the IP address will be reported in intrusion records; the authentication method will not be included in the record. The default value for this parameter is `publickey,password,hostbased`. If the same values are specified for both `VmsIntrusionIdentMethods` and `VmsIntrusionIdentSsh`, then `VmsIntrusionIdentMethods` takes precedence.

VmsLogFailAuthentications

This parameter can be used to control the reporting of login failures. Default value for this parameter is `publickey,password,hostbased`.

VmsDisallowSftpServer

This parameter can be used to control access to the SFTP server for all users. The default value of this parameter is `no`. Setting the value to `yes` will deny access to the SFTP server for all users.

VmsSftpDenyUsers

This parameter can be used to specify a list of users to be denied access to the SFTP server. The list of users must be specified as a list of username patterns separated by spaces. By default, no users will be denied access to the SFTP server.

VmsSftpDenyGroups

This parameter can be used to specify a list of user groups to be denied access to the sftp server. The list of groups must be specified as a list of group name patterns separated by spaces. By default, no OpenVMS user groups will be denied access to the SFTP server.

XAuthLocation

This parameter is used to specify the location of the XAuth tool.The default value is SYS$SYSTEM:DECW$XAUTH.EXE.

X11Forwarding

The `X11Forwarding` parameter controls whether the SSH Server permits X11 forwarding for connected clients. The value of this parameter can be specified as either `yes` or `no`:

- Specifying `yes` enables X11 forwarding. This allows SSH Clients to request X11 forwarding on connection.
- Specifying `no` disables X11 forwarding. This is the default.

4.6.2. Client Configuration Parameters

Client configuration parameters are defined in SSH$ROOT:[ETC]SSH_CONFIG. to control various aspects of OpenSSH client utilities (SSH, SFTP, and SCP). Additionally, the client configuration parameters may be specified when executing a client utility command using the **-o** command line option.

VSI has created a new client configuration parameter which may be used to disable the SFTP client extension known as the “VMSPlus mode”.

Note

At this time, this new parameter is only effective when specified on the SFTP command line, as shown below. Adding this parameter to the SSH_CONFIG. file will have no effect.

VMS Plus mode is enabled by default in the SFTP client to facilitate transfer of files with various OpenVMS file formats, including OpenVMS BACKUP savesets, between two OpenVMS hosts running OpenSSH v8.9-1G or newer.

Should VMS Plus mode cause issues when communicating with other SSH server implementations, it may be disabled by setting the `NoVmsPlus` parameter to any positive integer value. For example:

```
$ SFTP “-o NoVmsPlus 1” USER@HOST
```

4.7. Logical Names

The following logical names may be defined (at any level) to control the exit status of SFTP and other OpenSSH utilities, and to control the behaviour of the SFTP client when errors are encountered during file transfer operations.

TCPIP$SSH_SFTP_ALWAYS_EXIT_NORMAL

If this logical name is defined to TRUE or 1, the OpenSSH SFTP client will exit with an OpenVMS status of SS$_NORMAL in all cases. It should be noted that this logical name is applicable to the sftp client only (it is not applicable to SCP or any other OpenSSH utilities).

TCPIP$SSH_SFTP_BATCH_ABORT_ON_ERROR

This logical name can be used to prevent the SFTP client aborting during batch operations involving the transfer of multiple files. If this logical name is defined to FALSE or 0, the SFTP client will continue batch file transfer operations if an error occurs. Details of any errors will be logged.

OPENSSH$POSIX_EXIT_STATUS

Defining this logical name to TRUE or 1 will cause OpenSSH utilities to exit with a POSIX exit status (as would be the case on Linux).

OPENSSH$SFTP_UNIX_STYLE

The logical name OPENSSH$SFTP_UNIX_STYLE can be defined to control how VMS file versions are managed in the SFTP tool on the server.

When this logical name is not defined, the OpenSSH server will use file versions for any file-related operations (listing, downloading, and so on).

When this logical name is set (regardless of its value), the OpenSSH SFTP server will ignore file versions for any file related operations.

Note

By default, OPENSSH$SFTP_UNIX_STYLE is not defined.

4.8. Rights Identifiers

The rights identifier `TCPIP$SSH_FILECOPY_DISALLOWED` can be used to prevent users from connecting to the SFTP server.

4.9. X11 Port Forwarding

Note

X11 port forwarding requires DECwindows Motif for OpenVMS Systems to be installed. Ensure DECwindows Motif V1.8 is installed for VSI OpenVMS x86-64 systems.

X11 Port Forwarding via OpenSSH enables users to connect to an SSH server on the VSI OpenVMS host and run X11 client programs, which will appear on their local display.

X11 Port Forwarding must be enabled on both the SSH client and server. On the client side, this is typically done with the `-X` or `-Y` option when initiating an SSH connection.

The server must also have X11 Port Forwarding enabled in its SSH configuration file, ssh$root:[etc]sshd_config, by setting the option `X11Forwarding` to `Yes`. For further details see *Section 4.6.1, "Server Configuration Parameters"*.

Note

This version of OpenSSH for VSI OpenVMS does not support X11 Port Forwarding on the client side, which means you cannot use this feature to run graphical applications from a remote server on your local machine.