Technical Bulletin CS-17-09

Enabling SSH on CMS 2.x Server

This technical bulletin explains how to enable SSH access on your CMS 2.x Server.

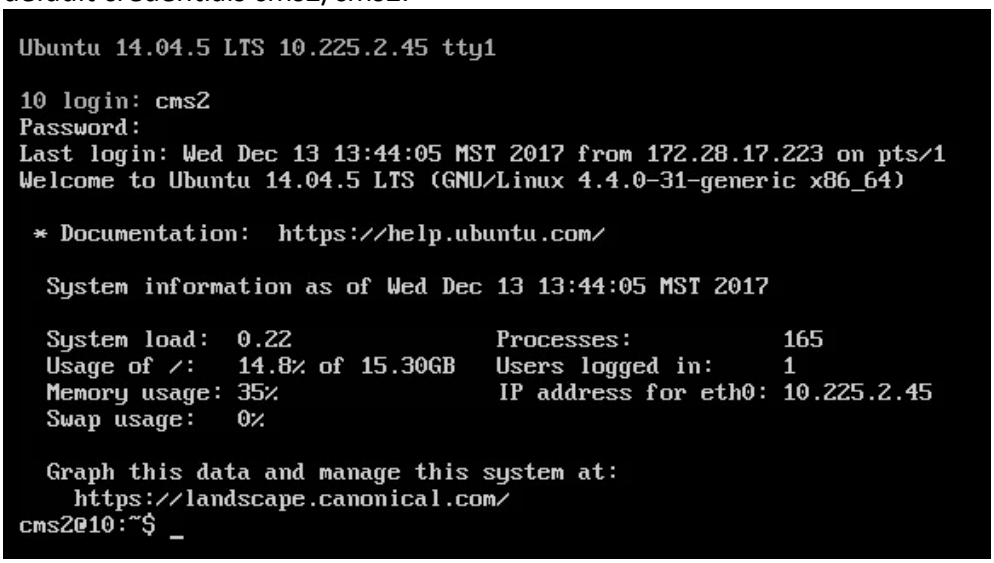
System Affected

CMS Release 2.1 and above

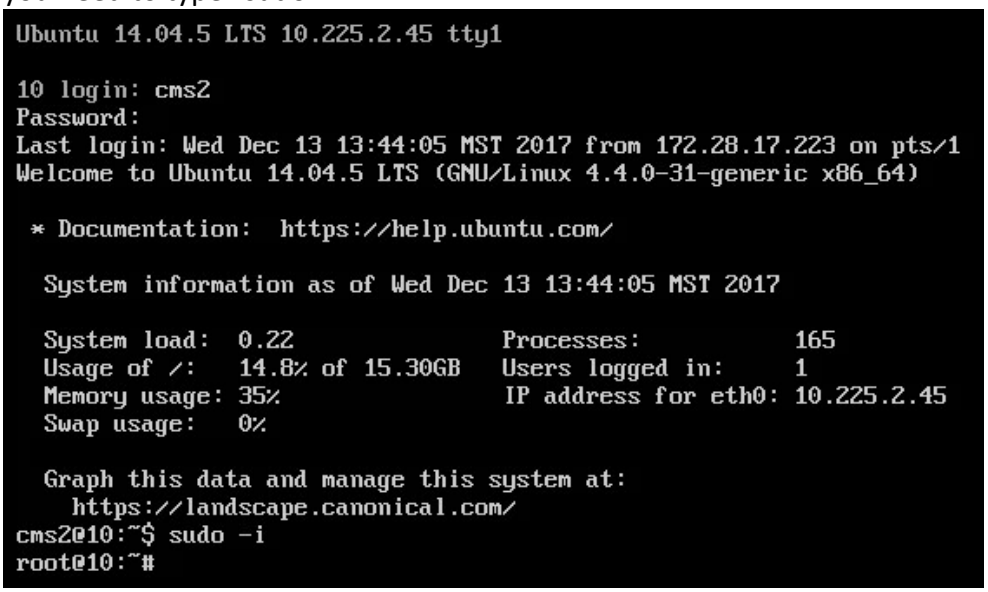
Description

For this document, we will explain how to modify your CMS 2.x Server to enable access via SSH. In order to accomplish this process you must already have access to the CMS console via your virtual machine management platform. Please also note that you should take a snapshot of your virtual machine before starting and save that in case something goes wrong. It would also be helpful for you to have some basic understanding of Linux commands and editors. We should also point out that this procedure is not officially supported by Spectralink and while it won't void any warranties or prevent your CMS from functioning normally, it is not recommended.

Once you've taken a snapshot of your CMS you will need to log into the console using the default credentials cms2/cms2.



Next we need to get into root in order to make changes to the configuration files. To do this, you need to type “sudo -i”



You will know you’re in root as the prompt will change to **root@10:~#**. Now we can proceed to the first set of configuration changes that need to occur.

We’ll navigate to /etc/pam.d/ to make the first set of changes. To do that you will enter the following:

```
cd /etc/pam.d
```

Now we need to edit the configuration file at this location. That file is called `sshd` and we’ll use the vi editor to make changes to it. I want to note right now that if at any point you think you made a mistake in the file you’re editing, you can use the vi command “:q!” to exit the current file without saving any changes. And that’s entered without the quotes. Okay, so on to editing this file. To do that you will enter the following on the screen:

```
vi sshd
```

Once you do that you will see the following appear on your screen:

```
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required     pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
#account  required     pam_access.so

# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close

# Set the loginuid process attribute.
session    required     pam_loginuid.so

# Create a new session keyring.
session    optional     pam_keyinit.so force revoke

# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional     pam_motd.so  motd=/run/motd.dynamic noupdate
session    optional     pam_motd.so # [1]

# Print the status of the user's mailbox upon successful login.
                                                                    1,1           Top
```

From here we will be making two minor changes. First is to add a comment “#” to the beginning of line 7 and the second will be to remove a comment from the beginning of line 11.

Line 7: account required pam_nologin.so
Line 11: #account required pam_access.so

To do this, you will use vi commands. Using your arrow keys, move the cursor down to the start of Line 7 so that your cursor is under the “a” of account. Then press the letter “i” for insert and then type the “#” key. Then you can press escape “ESC” to stop inserting.

Next you’ll move the cursor down to the start of Line 11 so that your cursor is under the “#”. Now press the letters “d” followed by “l”. This will perform a delete letter function to remove the “#” from the from of the line. When you’re done, it should look like this:

```
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
#account    required     pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
account  required     pam_access.so

# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close

# Set the loginuid process attribute.
session    required     pam_loginuid.so

# Create a new session keyring.
session    optional     pam_keyinit.so force revoke

# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional     pam_motd.so  motd=/run/motd.dynamic noupdate
session    optional     pam_motd.so # [1]

# Print the status of the user's mailbox upon successful login.
                                                                    11,1          Top
```

Now we can save this file and move on to the next one. To do that you'll type the following:

:wq!

That performs a write quit and the exclamation point at the end is a "bang" that means to force the operation.

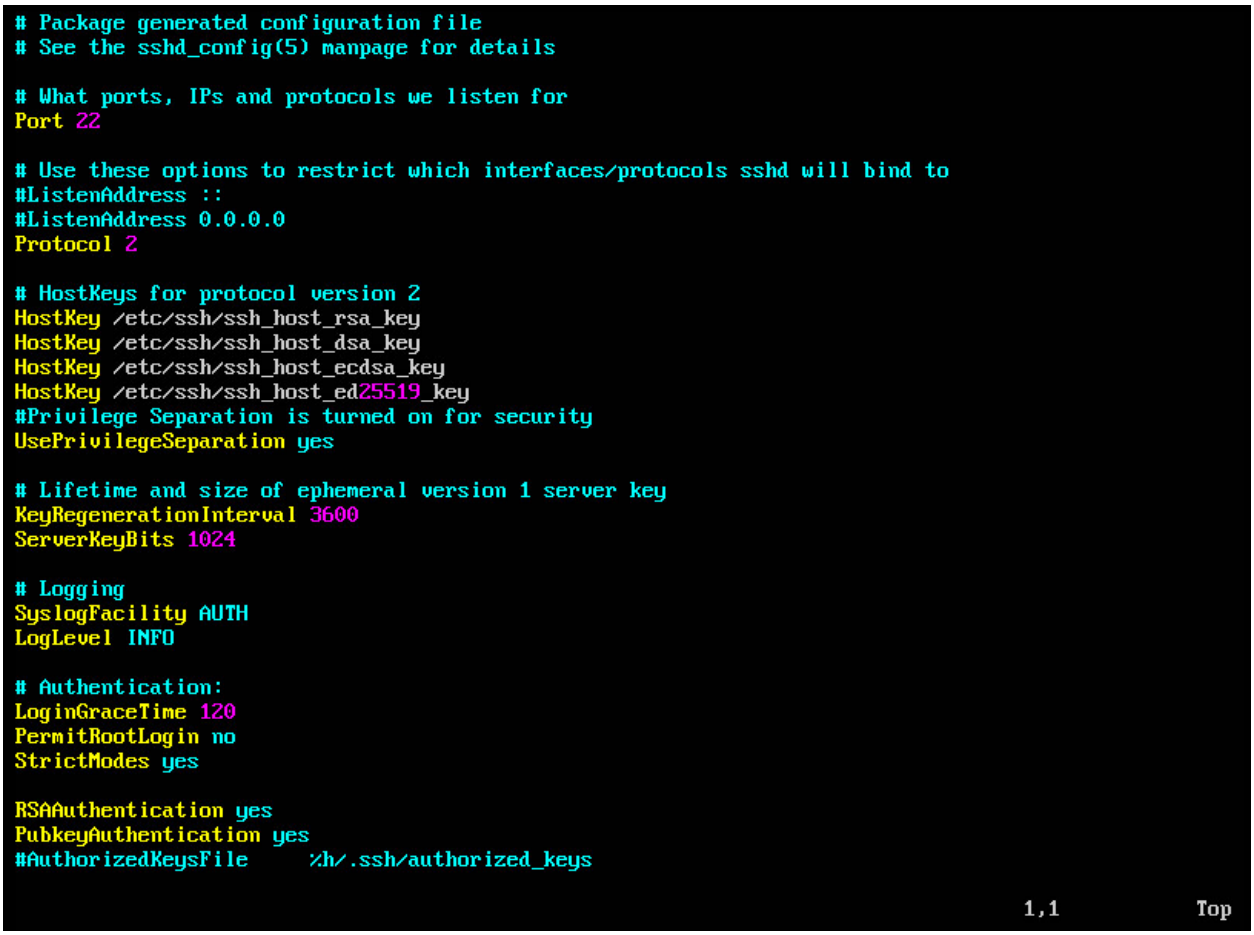
The next file we need to edit is in another directory. After we've exited the previous editing session you should now be at the command prompt again at the bottom of the screen. Here you will type:

cd ./etc/ssh

This will take you to the folder where we will edit the last file. Now you will type:

vi sshd_config

This will display another vi window where we can edit the final file. Your screen should now look like this:



In this file we will be making three changes. We'll using the same commands we did before and you'll still using the arrow keys to move around the file. We'll start by using the arrow keys to move down to the line at the bottom of the screen that says:

#AuthorizedKeysFile %h/.ssh/authorized_keys

When you get down to this line, put your cursor under the "#" key and enter the "dl" command to delete the "#" from the from of the line. Now you can keep using the arrow keys to scroll down further until you get to a line that says:

ChallengeResponseAuthentication no

Here you will use the arrow keys to move the cursor so that it is under the "n" of the word "no". Now enter "dw" that's delete word in vi. That will remove the word "no" from the line.

Now press “i” to start to insert and enter the word “yes”, without the quotes. When you’re done, press Escape to exit insert mode.

Next, we’ll move down just a couple lines to the line that looks like this:

PasswordAuthentication no

Much like we just did, use the arrow keys to move to the end of the line and put the cursor under the “n” of the word “no”. Enter “dw” again to the delete the word. Now press “i” to enter insert mode, and type “yes”, without the quotes. Now press Escape to exit insert mode. What you have now should look something like this:

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication yes

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
```

From here you can do the same save command we did before by entering:

:wq!

Which will then take you back to the command prompt. At this point, all you need to do is restart the CMS which can be done from this prompt using:

init 6

That's a Linux command to re-initialize the operating system through a restart.

Once the system comes back up and you are able to see the login prompt again you should now be able to connect to the CMS via SSH.

Please remember that we don't recommend doing this to your system unless you are comfortable doing these steps. You will not be able to receive support from Spectralink to complete this process. And if you irrepairably damage your CMS you will have to redeploy it, which is why we highly recommend taking a snapshot of the VM before starting. Also, if you're comfortable or familiar with other text editors in Linux, such as nano, then feel free to use those editors instead of VI.

Document Status Sheet

Document Control Number: CS-17-09

Document Title: Enabling SSH on CMS 2.x Server

Revision History: I01 – Released *December 15, 2017*
I02 – Released
I03 – Released

Date: *Current Date*

Status: ☐Draft ☒Issued ☐Closed

Distribution Status: ☐Author Only ☐Internal ☒Partner ☒Public

Copyright Notice

Notice

Warranty

The *Product Warranty and Software License and Warranty* and other support documents are available at http://support.spectralink.com.

Contact Information

US Location
+1 800-775-5330

Spectralink Corporation
2560 55th Street
Boulder, CO 80301
USA

info@spectralink.com

Denmark Location
+45 7560 2850

Spectralink Europe ApS
Bygholm Soepark 21 E Stuen
8700 Horsens
Denmark

infoemea@spectralink.com

UK Location
+44 (0) 20 3284 1536

Spectralink Europe UK
329 Bracknell, Doncastle Road
Bracknell, Berkshire, RG12 8PE
United Kingdom

infoemea@spectralink.com