Configuration Management

This module describes the enhancements done to Configuration Management Commands. Configuration management commands are specific instructions used within configuration management tools to automate the setup, modification, and maintenance of system and software configurations.

Auto-Save Configuration

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
Auto-Save with Secure File-Transfer and Additional Configurable Parameters	Release 7.9.1	Apart from automatically backing up the running configuration after every commit, you can also do the following with Auto-Save: • Save running configurations to remote systems using Secure Copy Protocol (SCP) and Secure File Transfer Protocol (SFTP). • Configure wait-time between two subsequent auto-saves. • Append time-stamp to the file name of the saved configuration. • Save the encrypted password. • Specify the maximum number of files that you can auto-save. The feature introduces these changes: CLI: Modified the **configuration commit auto-save**command Yang Data Model: • New XPaths for Cisco-IOS-XR-config-autosave-cfg • New XPaths for Cisco-IOS-XR-um-config-commit-cfg

You can configure the router to automatically take the backup of the running configuration by using **configuration commit auto-save** command. This auto-save feature saves the configuration to the specified location on the router after every **commit** is made. These auto-save files are stored in the form of Linux files.

Starting Cisco IOS XR Software Release 7.9.1, the auto-save feature is enhanced to provide a set of functionalities. This feature adds the following functionailites:

- You can save the running configuration backup files to remote location using **scp** and **sftp** file transfer protocols. SCP is a secure copy protocol to transfer files between servers. Whereas SFTP is a secure file transfer protocol for transfering large files.

- You can can save encrypted passwords for the remote and non-remote URLs.
- You can mention maximum number of files that can be saved automatically. Once the maximum number of auto-saved file is reached, the newer auto-save files starts replacing the older auto-save files. The default value of **maximum** is 1. You can save upto 4294967295 files.
- You can append timestamp to the auto-saved configuration file name. The **timestamp** uses the time and timezone configured on the router. The saved file displays timestamp in <day> <month> <date> <hours> <minutes> <seconds> <milliseconds> format. Here is an example of auto-saved file with time-stamp - *: test_123.autosave.1.ts.Tue_Jan_31_15-15-51_805_IST*
- You can specify how long to wait before next auto-save happens in terms of days, months or hours after the commit is made. The default value of **wait-time** is zero.

You can also configure options such as **password**, **timestamp**, **maximum**, and **wait-time** with the **configuration commit auto-save** command. The location to save the file-name must be specified in <protocol>://<user>@<host>:<port>/<url-path>/<file-name> format. When filename is accessed through VRF, you can specify filename in **filename** <protocol>://<user>@<host>:<port>;<vrf name>/<url-path>/<file-name> format.

Restriction for Auto-Save Configuration

The auto-save configuration is only available on the local paths, scp, and sftp paths.

Configure Auto-Save

Step 1 Use the **configuration commit auto-save** command to auto-save the configuration.

```
Router#configure
Router(config)#configuration commit auto-save
Router(config-cfg-autosave)#commit
```

Step 2 You can also use options such as **password**, **timestamp**, **maximum**, and **wait-time** with the **configuration commit auto-save** command.

```
Router(config-cfg-autosave)#configuration commit auto-save filename
sftp://user1@server1://test-folder/test_123
Router(config-cfg-autosave)#password clear encryption-default cisco
Router(config-cfg-autosave)#timestamp
Router(config-cfg-autosave)#maximum 10
Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
Router(config-cfg-autosave)#commit
```

Step 3 If you're using public-key authentication, you don't need to mention **password.**

Auto-Save and Copy Router Configuration Using Public Key Authentication

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Auto-Save and Copy Router Configuration Using Public Key Authentication	Release 7.10.1	You can now experience passwordless authentication while automatically saving running configurations and securely copying them on the router. The feature uses public key-based authentication, a secure logging method using a secure shell (SSH), which provides increased data security. This feature offers automatic authentication and single sign-on benefits, which also aids in a secure automation process. This feature modifies **configuration commit auto-save** and **copy** command to support password-less authentication.

From Cisco IOS XR Software Release 7.10.1, you don't need to remember and enter the **password** as you can use public key-based authentication while doing the following:

- Automatically saving your running configuration
- Copying the configuration from a source (such as a network server) to a destination (such as a flash disk)

Password is automatically verified when you have enabled SSH connection using public key-based authentication. Using public key-based authentication avoids several problems such as password disclosure and password leakage.

Public key is mathematically related to private key. The private key is secret, whereas the public key is available on the servers. You can copy the public key to the SSH server from the SSH client. Then, when you try to secure the running configuration, the SSH server tries to authenticate by generating a challenge using the public key. Only the private key can answer this challenge. As the keys are related, log-in is successful.

Use Public-Key Authentication with Auto-Save and Copy Commands

Step 1 Ensure you have enabled public key-based authentication of SSH clients, using the following steps:

- Generate RSA key pair on the router configured as the SSH client. Use the **cyrpto key generate authentication-ssh rsa** command to generate the RSA key pair.

- Use the **show crypto key mypubkey authentication-ssh rsa** command to view the details of the RSA key. The key value starts with *ssh-rsa* in this output.
- Copy the RSA public key from the SSH client to the SSH server.
  - You can do this either by logging in to the remote SSH server with your established user credentials, or have a system administrator on the remote system add the key on the SSH server.
  - If the SSH server is a Cisco IOS XR router, then you can use the **crypto key import authentication rsa** command on the router prompt of the server to import the key from the SSH client. You will then be prompted to enter the public key.
  - If the SSH server is a Linux server, then you must add the public key to the `~/.ssh/authorized_keys` file of the respective user account in that server. This file contains a list of all authorized public keys on that server.

For more detailed information on how to enable SSH connection using public-key based authentication, see *Public Key Based Authentication of SSH Clients* in System Security Configuration Guide.

Step 2 When you are using public key authentication for auto-save, you don't need to mention **password**.

```
Router(config-cfg-autosave)#configuration commit auto-save filename
sftp://user1@server1://test-folder/test_123
Router(config-cfg-autosave)#timestamp
Router(config-cfg-autosave)#maximum 10
Router(config-cfg-autosave)#wait-time days 0 hours 0 minutes 0 seconds 5
Router(config-cfg-autosave)#commit
```

Step 3 While you're using public-key authentication for copying running configuration from the router to a remote server, you don't need to mention **password** in the command.

```
Router#copy running-config scp://root@192.0.4.2//var/opt/run_conf_scp.txt
Router#copy running-config sftpp://root@192.0.4.2//var/opt/run_conf_sftp.txt
```