Ixia APS-M1010/APS-ONE-100

KCOS Command Line Interface (CLI)

Release 9.30

CLI Reference Guide

KEYSIGHT

Document version: 202212151411-08:00

Notices

Copyright Notice

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data. 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contacting Us

Keysight headquarters

1400 Fountaingrove Parkway
Santa Rosa, CA 95403-1738
www.ixiacom.com/contact/info

Support

Global Support	+1 818 595 2599	support@ixiacom.com
Regional and local support contacts:		
APAC Support	+91 80 4939 6410	support@ixiacom.com
Australia	+61-742434942	support@ixiacom.com
EMEA Support	+40 21 301 5699	support-emea@ixiacom.com
Greater China Region	+400 898 0598	support-china@ixiacom.com
Hong Kong	+852-30084465	support@ixiacom.com
India Office	+91 80 4939 6410	support-india@ixiacom.com
Japan Head Office	+81 3 5326 1980	support-japan@ixiacom.com
Korea Office	+82 2 3461 0095	support-korea@ixiacom.com
Singapore Office	+65-6215-7700	support@ixiacom.com
Taiwan (local toll-free number)	00801856991	support@ixiacom.com

Table of Contents

KCOS CLI Overview

Keysight Cluster Operating System (KCOS) is the base operating system for a number of Keysight hardware platforms and web-based applications. Based on Linux, it is a purpose-built platform that provides essential services, including host management, user access control, software upgrades, cluster orchestration, among others. It provides a command line interface (CLI) and a command shell to enable user control of the various services.

This document provides reference information for the KCOS CLI commands.

The KCOS CLI Shell

- Upon initial log in, you are working in the KCOS framework, which defaults to the bash shell. While in this shell, you need to start each of the KCOS CLI commands with the "kcos" keyword; for example, *kcos networking ip set*
- You can use the optional KCOS shell to limit your command set to only the KCOS commands. While working in the KCOS shell, you do not use the "kcos" keyword. For example, you would simply enter *networking ip set*.

Shell auto-completion

1. Type a partial command (*snapshot* for example).
2. Press the **Tab** key twice to display the full list of available matching commands.
3. Use the **Tab** key to advance to the specific command that you wish to execute.
4. Press **Enter** to select the command.
5. Type any additional flags and arguments for the command.
6. Press **Enter** to issue the command.

Root access prohibited

While working in the KCOS framework, you can use any Linux command except for *sudo*: KCOS does not support root access on any Keysight hardware on which it is running.

KCOS syntax description

The KCOS CLI commands are described using the following command syntax:

- The words that comprise the command name are shown without any brackets or braces.
- Required arguments are shown in <angle brackets>.
- Optional arguments are shown in [square brackets].
- Mutually-exclusive items are shown in {braces}, with the items separated by vertical bars (pipe symbols).
- Arguments that can repeat are shown with a trailing ellipses (…).

KCOS shell authentication

To start a KCOS CLI session, you must first authenticate yourself using your user credentials. KCOS supports a single sign-on (SSO) methodology, wherein your application administrator configures the user IDs and passwords that you use for accessing the application and the KCOS CLI shell: the same user credentials are used for both. To start a KCOS CLI session:

1. Use the `ssh` command to open an SSH connection to the remote device. For example:
   ```
   ssh username@kcos_system_host
   ```
2. When prompted, enter your password.

KCOS verifies your sign-on credentials, starts the CLI session, and presents the CLI prompt. For example:

```
(kcos)-APS-M1-TW21160109:~$
```

kcos date-time commands

The `date-time` CLI commands enable management of date-time controls and settings on your system. With these commands you can show and set the system date and time, manage the NTP servers that your system will use, and set the time zone for your system.

kcos date-time date set

Use the **kcos date-time date set** command to set the UTC (Coordinated Universal Time) system date and—optionally—the time for the node on which KCOS is running.

Syntax

```
kcos date-time date set <YYYY-MM-DD> [<hh:mm:ss>] [flags]
```

Flags

-h, --help	Display help for the command

Arguments

YYYY-MM-DD	The calendar date, using ISO 8601 notation.
hh:mm:ss	The time, using 24-hour notation.

Examples

```
kcos date-time date set 2020-09-24
kcos date-time date set 2020-11-04 22:01:30
```

See also

```
kcos date-time date show
kcos date-time time-zone set
```

Notes

Use "kcos date-time time set" to set the time only.

kcos date-time date show

Use the **kcos date-time date show** command to display the date, time, and timezone for the node on which KCOS is running.

Syntax

```
kcos date-time date show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time date show
kcos date-time date show --help
```

See also

```
kcos date-time date set
kcos date-time time-zone set
```

kcos date-time ntp-servers delete

Use the **kcos date-time ntp-servers delete** command to delete the complete list of NTP servers from the node.

Syntax

```
kcos date-time ntp-servers delete [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time ntp-servers delete
kcos date-time ntp-servers delete --help
```

See also

```
kcos date-time ntp-servers set
kcos date-time ntp-servers show
```

kcos date-time ntp-servers set

Use the **kcos date-time ntp-servers set** command to set the list of NTP servers for the node.

Syntax

```
kcos date-time ntp-servers set <NTP server addresses> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

NTP server addresses	A space-separated list of one or more NTP server IP addresses.

Examples

```
kcos date-time ntp-servers set 192.168.7.12
kcos date-time ntp-servers set 10.38.140.12 192.168.22.10
```

See also

```
kcos date-time ntp-servers delete
kcos date-time ntp-servers show
```

kcos date-time ntp-servers show

Use the **kcos date-time ntp-servers show** command to display the list of known NTP servers and their status (enabled or disabled).

Syntax

```
kcos date-time ntp-servers show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time ntp-servers show
kcos date-time ntp-servers show --help
```

See also

```
kcos date-time ntp-servers set
kcos date-time ntp-servers delete
```

kcos date-time ntp disable

Use the **kcos date-time ntp disable** command to disable the NTP settings.

Syntax

```
kcos date-time ntp disable [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time ntp disable
kcos date-time ntp disable --help
```

See also

```
kcos date-time ntp enable
kcos date-time ntp-servers
```

kcos date-time ntp enable

Use the **kcos date-time ntp enable** command to enable the NTP settings on the node.

Syntax

```
kcos date-time ntp enable [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time ntp enable
kcos date-time ntp enable --help
```

See also

```
kcos date-time ntp disable
kcos date-time ntp-servers
```

kcos date-time time-zone set

Use the **kcos date-time time-zone set** command to set the UTC timezone for the node.

Syntax

```
kcos date-time time-zone set <time-zone> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

time-zone	A text string designating the time zone.

Examples

```
kcos date-time time-zone set UTC
kcos date-time time-zone set Asia/Seoul
kcos date-time time-zone set Europe/Bucharest
```

See also

```
kcos date-time time-zone show
kcos date-time date set
kcos date-time date show
```

kcos date-time time-zone show

Use the **kcos date-time time-zone show** command to display the UTC timezone configured for the node.

Syntax

```
kcos date-time time-zone show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time time-zone show
```

See also

```
kcos date-time time-zone set
kcos date-time time show
kcos date-time date show
```

kcos date-time time set

Use the **kcos date-time time set** command to set system time for the node on which KCOS is running.

Syntax

```
kcos date-time time set <hh:mm:ss> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

hh:mm:ss	The system time (using 24-hour notation).

Examples

```
kcos date-time time set 14:30
kcos date-time time set 01:30
```

See also

```
kcos date-time date set
kcos date-time date show
kcos date-time time-zone set
```

Notes

Use "kcos date-time date set" to set both the date and the time.

kcos date-time time show

Use the **kcos date-time time show** command to display the current date, time, and timezone.

Syntax

```
kcos date-time time show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos date-time time show
```

See also

```
kcos date-time time-zone show
kcos date-time date set
kcos date-time date show
```

kcos deployment commands

The `deployment` CLI commands enable control over the installation, upgrading, and downgrading of software components on your system, as well as listing the software packages that are installed on your system and packages that are available for installation.

kcos deployment available-updates

Use the **kcos deployment available-updates** command to list the packages that are available for online installation.

Syntax

```
kcos deployment available-updates [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos deployment available-updates
```

See also

```
kcos deployment online-install
```

kcos deployment offline-install

Use the **kcos deployment offline-install** command to install/upgrade/downgrade software components from package(s) that you have obtained from the support portal. Offline installation is required for products that may operate in an isolated environment with no Internet access.

Syntax

```
kcos deployment offline-install <offline package … > [flags]
```

Flags

-h, --help	Display help for the command.
-r --reboot	If any software component requires it, automatically trigger a system reboot once the deploy operation is complete.

Arguments

offline package	A space-separated list of offline packages (one or more) to install.

Examples

```
kcos deployment offline-install aps-kcos/9.17.39 --reboot
kcos deployment offline-install aps-kcos/9.17.39 aps-bps/9.17.13 -r
kcos deployment offline-install aps-bps/9.17.13
kcos deployment offline-install "abc xyz"
```

See also

```
kcos deployment packages show
kcos deployment online-install
```

Notes

- The offline-packages (tar files) are available for download from your product's support portal.
- Once downloaded, transfer the package to the system where the offline deploy operation will be executed. You will then use the package name and path as the argument in the command.
- KCOS will ignore the command if it specifies a package that is already installed.
- The **kcos deployment offline-install** command always creates a snapshot of the system before installing any *aps-kcos* packages (such as aps-kcos/9.17.39 shown in the examples above). The snapshot name will be:

  <timestamp>-aps-bps-<version>-aps-kcos-<version>-before-upgrade

kcos deployment online-install

Use the **kcos deployment online-install** command to install/upgrade/downgrade software components from the Keysight online repository. The command requires a space-separated list of online packages (one or more) to install.

Syntax

```
kcos deployment online-install <package-name[/version] … > [flags]
```

Flags

-h, --help	Display help for the command.
-r, --reboot	If any software component requires it, automatically trigger a system reboot once the deploy operation is complete.

Arguments

package-name	The name of the package to install.
/version	An optional version number for the package.

Examples

```
kcos deployment online-install aps-kcos/9.17.39 aps-bps/9.17.13
kcos deployment online-install aps-kcos/9.17.39 --reboot
kcos deployment online-install "abc xyz"
```

See also

```
kcos deployment available-updates
kcos deployment packages show
kcos deployment offline-install
```

Notes

- Use the `kcos deployment available-updates` command to get a list of online packages that are available for deployment.
- If the package name and version number string includes spaces, then enclose it in double-quotes.
- KCOS will ignore the command if it specifies a package that is already installed.
- The **kcos deployment online-install** command always creates a snapshot of the system before installing any *aps-kcos* packages (such as aps-kcos/9.17.39 shown in the examples above). The snapshot name will be:

  <timestamp>-aps-bps-<version>-aps-kcos-<version>-before-upgrade

kcos deployment packages show

Use the **kcos deployment packages show** command to display the list of installed packages (package name and version number).

Syntax

```
kcos deployment packages show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos deployment packages show
```

See also

```
kcos deployment available-updates
kcos deployment offline-install
kcos deployment online-install
```

kcos deployment progress show

Use the **kcos deployment progress show** command to show progress information for a pending install operation. The command continually displays the status until the operation completes, unless you use the `--once` flag.

Syntax

```
kcos deployment progress show [flags]
```

Flags

-h, --help	Display help for the command
-o, --once	Display the status once, and then exit (as opposed to continually displaying it until the operation completes).

Arguments

None.

Examples

```
kcos deployment progress show
kcos deployment progress show --once
```

See also

```
kcos deployment packages show
kcos deployment offline-install
```

kcos exit

Use the **kcos exit** command to exit from the current KCOS Shell, returning to the default Linux shell.

Syntax

```
exit [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
exit --help
exit
```

See also

```
kcos shell
```

kcos firewall commands

Blocking TCP and UDP ports that are not required is always recommended.

The `firewall` CLI commands enable control over the KCOS Firewall Service. When turned on, the KCOS Firewall opens only the ports specified by the application, and blocks incoming traffic on any other port. The `kcos firewall show` command displays a list of the open ports; this list is defined by the application at installation time.

kcos firewall disable

Use the **kcos firewall disable** command to turn off the KCOS Firewall Service. When disabled, all TCP and UDP ports are open to incoming traffic.

Syntax

```
kcos firewall disable [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos firewall disable --help
```

See also

```
kcos firewall enable
kcos firewall show
```

Notes

You should never need to disable firewall unless requested to do so in a debug session with the Keysight support team.

kcos firewall enable

Use the **kcos firewall enable** command to enable the entire firewall, in which case KCOS opens only the TCP and UDP ports that are defined by the application's firewall rules: incoming traffic on any other port is blocked. The KCOS Firewall Service is enabled by default.

Syntax

```
kcos firewall enable [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos firewall enable --help
```

See also

```
kcos firewall disable
kcos firewall show
```

Notes

When firewall is enabled, ports 80, 443, and 22 are open (for HTTP, HTTPS, and SSH, respectively).

kcos firewall show

Use the **kcos firewall show** command to display the firewall status (enabled or disabled) and the list of open ports.

Syntax

```
kcos firewall show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos firewall show --help
```

See also

```
kcos firewall disable
kcos firewall enable
```

kcos logs commands

The `logs` CLI commands provide the tools for collecting diagnostics logs for a system and also for displaying the log for an application that is running on the system.

kcos logs application

Use the **kcos logs application** command to display the last 100 lines of the log for an application or a BMC.

Syntax

```
kcos logs application [flags]
```

Flags

-a, --application app-name	The name of the application for which you are retrieving the log. The *app-name* can be either a software application name (such as *bps*) or the name of a BMC (such as *kcos-sol-bmc-00-18-a0-ae-4a-45*).
-c, --container container-name	The name of the container in which the application is running. The container name is required only if the pod has more than one container. Note that BMC pods have only one container.
-h, --help	Display help for the command.

Arguments

None.

Examples

```
kcos logs application -a BPS
kcos logs application -a BPS -c container4
kcos logs application -a kcos-sol-bmc-00-18-a0-ae-4a-45
kcos logs application -h
```

See also

```
kcos logs diagnostics collect
kcos system introspection pods show
```

Notes

If you enter `kcos logs application` without a flag, KCOS will list the available applications and BMCs.

kcos logs diagnostics collect

Use the **kcos logs diagnostics collect** command to collect chassis logs for all available components or for selected components and/or subcomponents.

Syntax

```
kcos logs diagnostics collect <<flags> [string array]>
```

Flags

-a, --all	Collect logs for all available components.
-c, --components stringArray	Collect logs for the components specified by [stringArray], where [stringArray] is a single component name.
-g, --get-components	Get the list of available components.
-h, --help	Display help for the command.
-o, --option stringArray	Option for the selected subcomponent. The -o flag always requires the -v flag.
-s, --subcomponents stringArray	Collect logs for the subcomponents specified by [stringArray], where [stringArray] is a single subcomponent name.
-v, --value stringArray	Value for the selection option.

Arguments

string array	A list of one or more component names (if the **-c** flag is used) or subcomponent names (if the **-s** flag is used).

Examples

```
kcos logs diagnostics collect --all
kcos logs diagnostics collect --get-components
kcos logs diagnostics collect -c kcos-system-diagnostics -s clusterinfo -s elasticdump
kcos logs diagnostics collect -c kcos-system-diagnostics -s clusterinfo -s elasticdump
-o index-name -v <index value>
```

See also

```
kcos logs application
kcos system welcome-screen show
kcos snapshot show
```

Notes

- Each component must be specified with `-c` and each subcomponent with `-s`
- Using component options works for only one subcomponent at a time.
- Once the log collection is complete, KCOS will show the path to the zip file.

- You can use the SFTP client from any machine to connect to the KCOS framework shell and download the zip file.
- Once you download the archive:
  a. Unzip the *logs-yyyy-mm-dd-hh-min-sec.zip*.file.
  b. Unzip the kcos-system-diagnostics and the agent-diagnostics zip files.
  c. Access the log files from the folders that were contained in the zip files.

kcos netif diagnostics commands

The `netif diagnostics` CLI commands provide the tools for quickly troubleshooting front-panel port issues. You can retrieve the state of a front-panel port on any node in the cluster, display driver and transceiver details, and retrieve statistics from a front-panel port on any node in the cluster.

kcos netif diagnostics config show

Use the **kcos netif diagnostics config show** command to get the link status for node interfaces. The default command output lists the available slots on the system, the node associated with each slot, the interfaces on each node, and—for each interface—the link status, speed, FEC value, and media type.

Syntax

```
kcos netif diagnostics config show [nodeName] [netIfName] [--regex] [--raw] [--help]
```

Arguments

nodeName	The name of the node for which you are getting the link status.
netIfName	The node interface name for which you are getting the link status.

Flags

--regex	The `--regex` flag enables the use of regular expressions to match multiple nodeNames or netIfNames.
	• Example 1: to display all "eaglefp0" interfaces for every node:
	`kcos netif diagnostics config show ".*" eaglefp0 --regex`
	• Example 2: to display all interfaces for a given node:
	`kcos netif diagnostics config show eagle-aps-o1-tw21160132 ".*" --regex`
	The Golang regexp syntax is used for the regular expressions. Its specification is available here: https://pkg.go.dev/regexp/syntax.

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Examples

```
kcos netif diagnostics config show
kcos netif diagnostics config show eagle-aps-o1-tw21160136 eaglefp2
kcos netif diagnostics config show ".*" eaglefp0 --regex
kcos netif diagnostics config show eagle-aps-o1-tw21160132 ".*" --regex
kcos netif diagnostics config show --raw
```

See also

```
kcos system introspection node show
kcos logs diagnostics collect
```

kcos netif diagnostics driver show

Use the **kcos netif diagnostics driver show** command to get driver information for node interfaces. The default command output lists the available slots on the system, the node associated with each slot, the interfaces on each node, and—for each interface—the driver type, driver version, firmware version, and PCIe (Peripheral Component Interconnect Express) address.

Syntax

```
kcos netif diagnostics driver show [nodeName] [netIfName] [--regex] [--raw] [--help]
```

Arguments

nodeName	The name of the node for which you are getting the driver information.
netIfName	The node interface name for which you are getting the driver information.

Flags

--regex	The `--regex` flag enables the use of regular expressions to match multiple nodeNames or netIfNames.
	• Example 1: to display the driver information for all "eaglefp0" interfaces for every node:
	`kcos netif diagnostics driver show ".*" eaglefp0 --regex`
	• Example 2: to display the driver information for all interfaces for a given node:
	`kcos netif diagnostics driver show eagle-aps-o1-tw21160132 ".*" --regex`
	The Golang regexp syntax is used for the regular expressions. Its specification is available here: https://pkg.go.dev/regexp/syntax.

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Examples

```
kcos netif diagnostics driver show
kcos netif diagnostics driver show eagle-aps-o1-tw21160136 eaglefp2
kcos netif diagnostics driver show ".*" eaglefp0 --regex
kcos netif diagnostics driver show eagle-aps-o1-tw21160132 ".*" --regex
kcos netif diagnostics driver show --raw
```

See also

```
kcos system introspection node show
kcos logs diagnostics collect
```

kcos netif diagnostics interfaces show

Use the **kcos netif diagnostics interfaces show** command to obtain a list of node interfaces from the system. The default command output lists the available slots on the system, the node associated with each slot, and all of the interfaces on that node.

Syntax

```
kcos netif diagnostics interfaces show [nodeName] [netIfName] [--regex] [--raw] [--help]
```

Arguments

nodeName	The name of the node for which you are getting the list of interfaces.
netIfName	The node interface name for which you are getting the list.

Flags

--regex	The `--regex` flag enables the use of regular expressions to match multiple nodeNames or netIfNames.
	• Example 1: to display all "eaglefp0" interfaces for every node:
	`kcos netif diagnostics interfaces show ".*" eaglefp0 --regex`
	• Example 2: to display all interfaces for a given node:
	`kcos netif diagnostics interfaces show eagle-aps-o1-tw21160132 ".*" --regex`
	The Golang regexp syntax is used for the regular expressions. Its specification is available here: https://pkg.go.dev/regexp/syntax.

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Examples

```
kcos netif diagnostics interfaces show
kcos netif diagnostics interfaces show eagle-aps-o1-tw21160136 eaglefp2
kcos netif diagnostics interfaces show ".*" eaglefp0 --regex
kcos netif diagnostics interfaces show eagle-aps-o1-tw21160132 ".*" --regex
kcos netif diagnostics interfaces show --raw
```

See also

```
kcos system introspection node show
kcos logs diagnostics collect
```

kcos netif diagnostics nodes show

Use the **kcos netif diagnostics nodes show** command to obtain a list of available slots and the node associated with each of the slots.

Syntax

```
kcos netif diagnostics nodes show [--raw] [--help]
```

Flags

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Arguments

None.

Examples

```
kcos netif diagnostics nodes show
kcos netif diagnostics nodes show --raw
```

See also

```
kcos system introspection nodes show
kcos logs diagnostics collect
```

kcos netif diagnostics stats show

Use the **kcos netif diagnostics stats show** command to display statistics for a specific interface on a node. Command switches allow for statistic names to be excluded or included based on Golang regular expression matching, and also allow for the output to be presented in raw JSON format.

The default command output lists the available slots on the system, the node associated with each slot, the interfaces on each node, and—for each interface—the stat name and the value of that statistic.

NOTE The statistics always include the UTC timestamp of the time on the compute node when the statistics were acquired for each interface. This statistic cannot be excluded using the `--exclude` switch.

Syntax

```
kcos netif diagnostics stats show [nodeName] [netIfName] [<--include> <include-regex> |
   <--exclude> <exclude-regex>] [--regex] [--raw] [--help]
```

Arguments

nodeName	The name of the node for which you are displaying statistics.
netIfName	The node interface name for which you are displaying statistics.

Flags

--include	Use this flag to include specific statistics based on a regular expression.
--exclude	Use this flag to exclude specific statistics based on a regular expression.

--regex	The `--regex` flag enables the use of regular expressions to match multiple nodeNames or netIfNames.
	• Example 1: to display statistics for all "eaglefp0" interfaces for every node:
	`kcos netif diagnostics stats show ".*" eaglefp0 --regex`
	• Example 2: to display statistics for all interfaces for a given node:
	`kcos netif diagnostics stats show eagle-aps-o1-tw21160132 ".*" --regex`
	The Golang regexp syntax is used for the regular expressions. Its specification is available here: https://pkg.go.dev/regexp/syntax.

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Examples

```
kcos netif diagnostics stats show
kcos netif diagnostics stats show eagle-aps-o1-tw21160136 eaglefp2
kcos netif diagnostics stats show ".*" eaglefp0 --regex
kcos netif diagnostics stats show eagle-aps-o1-tw21160132 ".*" --regex
kcos netif diagnostics stats show --include ".*faults.nic"
kcos netif diagnostics stats show --raw
```

Output example

SLOT	NODE	INTERFACE	STAT NAME	STAT VALUE
2	eagle-aps-o1-tw21160132	eaglefp0	_utc_timestamp_ns	1628875776903100446
		eaglefp0	chnl_inline_fd_match	0
		eaglefp0	fdir_sb_match.nic	0
		eaglefp0	fdir_sb_status.nic	1
		eaglefp0	illegal_bytes.nic	0
		eaglefp0	mac_local_faults.nic	0
		eaglefp0	mac_remote_faults.nic	0
3	eagle-aps-o1-tw21160116	eaglefp0	_utc_timestamp_ns	1628875776977348475

See also

```
kcos logs diagnostics collect
```

kcos netif diagnostics transceiver show

Use the **kcos netif diagnostics transceiver show** command to get transceiver information for node interfaces. The default command output lists the available slots on the system, the node associated with each slot, the interfaces on each node, and—for each interface—the transceiver type, transceiver vendor, vendor part number, vendor serial number, and the transceiver date code.

Syntax

```
kcos netif diagnostics transceiver show [nodeName] [netIfName] [--regex] [--raw]
   [--help]
```

Arguments

nodeName	The name of the node for which you are getting the transceiver information.
netIfName	The node interface name for which you are getting the transceiver information.

Flags

--regex	The `--regex` flag enables the use of regular expressions to match multiple nodeNames or netIfNames.
	• Example 1: to display the transceiver information for all "eaglefp0" interfaces for every node:
	`kcos netif diagnostics transceiver show ".*" eaglefp0 --regex`
	• Example 2: to display the transceiver information for all interfaces for a given node:
	`kcos netif diagnostics transceiver show eagle-aps-o1-tw21160132 ".*" --regex`
	The Golang regexp syntax is used for the regular expressions. Its specification is available here: https://pkg.go.dev/regexp/syntax.

--raw, -r	When you specify the `--raw` flag, the command displays the raw JSON output rather than presenting the information in a text table. The JSON files often have more information than that displayed in the text table output.
--help, -h	Display help for the command.

Examples

```
kcos netif diagnostics transceiver show
kcos netif diagnostics transceiver show eagle-aps-o1-tw21160136 eaglefp2
kcos netif diagnostics transceiver show ".*" eaglefp0 --regex
kcos netif diagnostics transceiver show eagle-aps-o1-tw21160132 ".*" --regex
kcos netif diagnostics transceiver show --raw
```

See also

```
kcos system introspection node show
kcos logs diagnostics collect
```

kcos networking commands

Using the `networking` CLI commands, you can add, delete, and show the DNS servers for the node on which KCOS is running; and configure the hostname and domain name for the node.

kcos networking dns-servers add

Use the **kcos networking dns-servers add** command to add one or more DNS servers for the node on which KCOS is running.

Syntax

```
kcos networking dns-servers add <DNS server address … > [flags]
```

Flags

-h, --help	Display help for the command

Arguments

DNS server address	A space-separated list of one or more DNS server IP addresses.

Examples

```
kcos networking dns-servers add 192.168.12.2
kcos networking dns-servers add 10.38.140.2 192.168.12.10
```

See also

```
kcos networking dns-servers delete
kcos networking dns-servers show
```

Notes

The order in which you enter the DNS server addresses determines their priority. The first server listed has highest priority, with each additional server having a lesser priority.

kcos networking dns-servers delete

Use the **kcos networking dns-servers delete** command to delete the complete list of DNS servers from the node.

Syntax

```
kcos networking dns-servers delete [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos networking dns-servers delete
```

See also

```
kcos networking dns-servers add
kcos networking dns-servers show
```

kcos networking dns-servers show

Use the **kcos networking dns-servers show** command to display the list of DNS servers configured on the node. The output displays the DNS servers in order of priority, with the highest priority server listed first .

Syntax

```
kcos networking dns-servers show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos networking dns-servers show
```

See also

```
kcos networking dns-servers add
kcos networking dns-servers delete
```

kcos networking hostname set

Use the **kcos networking hostname set** command to configure the hostname—and optionally the domain name—for the node on which KCOS is running.

Syntax

```
kcos networking hostname set <hostname[.<domain name>]> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

hostname	The hostname to assign to the node. The hostname may be a simple name or an FQDN. The semantics are identical to /etc/hostname.
domain name	The domain name to assign to the node. This argument is optional.

Examples

```
kcos networking hostname set kcosMaster27
kcos networking hostname set north14.xyz.is.keysight.com
```

See also

```
kcos networking hostname show
```

Notes

The hostname must observe the following requirements:

- may contain letters and digits
- may contain only the following special characters: hyphen and dot ("-", ".")
- an alpha character must follow each dot
- cannot start with a digit, hyphen, or dot
- cannot end with hyphen (if it ends with a dot, the dot will be removed)
- cannot have more than 63 characters

kcos networking hostname show

Use the **kcos networking hostname show** command to display the node's hostname.

Syntax

```
kcos networking hostname show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos show hostname
```

See also

```
kcos networking hostname set
```

kcos networking ip set

Use the **kcos networking ip set** command to set the IPv4 or IPv6 address for an interface on the node on which KCOS is running. You can designate the address as either DHCP-assigned or static IP.

Syntax

```
kcos networking ip set <<interface>
   <{dhcp|dhcpv4|dhcpv6} | IP-address [subnet]>> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

interface	The name of the interface on which the IP address is being set.
dhcp|dhcpv4|dhcpv6	Designates the address as DHCP-assigned: • Specify "dhcpv4" or "dhcpv6" to enable DHCP only on IPv4 or IPv6. • Specify "dhcp" to enable DHCP on both IPv4 and IPv6.
IP/subnet	The static IPv4 or IPv6 address and (optionally) the subnet.
gateway	The address of the gateway.

Examples

```
kcos networking ip set mgmt0 dhcp
kcos networking ip set mgmt0 dhcpv6
kcos networking ip set mgmt1 10.200.1.20/24
kcos networking ip set mgmt1 10.200.1.20/24 10.200.1.1
```

See also

```
kcos networking ip show
```

kcos networking ip show

Use the **kcos networking ip show** command to show the following information for each network interface: interface name, IP addresses (IPv4 and/or IPv6), MAC address, state, and gateways (if any).

Syntax

```
kcos networking ip show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos networking ip show
kcos networking ip show --help
```

See also

```
kcos networking ip set
```

kcos shell

Use the **kcos shell** command to enter the KCOS shell in the command window. While working in the shell, the prompt changes to **kcos>** and you directly enter KCOS commands without preceding them with "kcos". For example, you would enter a command such as *kcos networking ip set* if you are using the default shell (bash); in contrast, if you are using the KCOS shell, you would simply enter *networking ip set*.

The KCOS shell recognizes only KCOS commands.

Syntax

```
kcos shell [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos shell --help
kcos shell
```

See also

```
kcos exit
```

Notes

Exiting the KCOS Shell returns you to the default Bash shell.

kcos snapshot commands

The `snapshot` CLI commands enable the creation and management of system snapshots on the node in which KCOS is running.

kcos snapshot create

Use the **kcos snapshot create** command to create a new system snapshot of the node on which KCOS is running. The snapshot-create operation requires a reboot.

Syntax

```
kcos snapshot create <snapshot-label> [flags]
```

Flags

-h, --help	Display help for the command.
-y, --yes	Confirm reboot. The reboot is needed to quiesce the system, and thereby ensure that it is in a consistent state before the snapshot it taken.

Arguments

snapshot-label	A label to assign to this snapshot.

Examples

```
kcos snapshot create label20210928 --yes
kcos snapshot create 20211204 -y
```

See also

```
kcos snapshot delete
kcos snapshot restore
kcos snapshot show
```

kcos snapshot delete

Use the **kcos snapshot delete** command to delete an existing system snapshot.

Syntax

```
kcos snapshot delete <snapshot-name> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

snapshot-name	The name of the snapshot to be deleted.

Examples

```
kcos snapshot delete snapshotDec19
kcos snapshot delete shapshotJan20
```

See also

```
kcos snapshot create
kcos snapshot restore
kcos snapshot show
```

kcos snapshot restore

Use the **kcos snapshot restore** command to restore the system from a specific system snapshot. The snapshot-restore operation requires a reboot.

Syntax

```
kcos snapshot restore <snapshot-name> [flags]
```

Flags

-h, --help	Display help for the command.
-y, --yes	Confirm reboot. The reboot is needed to restart the system using the restored snapshot.

Arguments

snapshot-name	The name of the snapshot to restore.

Examples

```
kcos snapshot restore snapshotDec19 --yes
kcos snapshot restore snapshotFeb20 -y
```

See also

```
kcos snapshot delete
kcos snapshot create
kcos snapshot show
```

Notes

- APS systems come with a factory image (named *factory*). If necessary, use the `kcos snapshot restore factory --yes` command to revert to this image.
- In case the KCOS operating system becomes corrupted and inaccessible on the kcos console, the system can be restored to *factory* by directly connecting to it (serial or VGA connecton) and using the grub factory restore option to restore the factory image.

kcos snapshot show

Use the **kcos snapshot show** command to display a list of the existing system snapshots.

Syntax

```
kcos snapshot show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos snapshot show
kcos snapshot show -h
```

See also

```
kcos snapshot delete
kcos snapshot create
kcos snapshot restore
```

kcos system commands

The `system` CLI commands provide the following system-level functionality:

- Basic system operations, including power-down and reboot.
- Firmware control, including informational display of installed firmware and firmware upgrade
- Introspection service, which is used for Kubernetes cluster debugging and operations and also for controlling power on a node using the Baseboard Management Controller (BMC).

System commands:

kcos system firmware show

Use the **kcos system firmware show** command to display information about the firmware that is installed on the system. The information includes the node name, firmware type, firmware name, firmware version, and whether or not an update is available.

Syntax

```
kcos system firmware show [flags]
```

Flags

-h, --help	Display help for the command
-r, --raw	Display the raw JSON for firmware.

Arguments

None.

Examples

```
kcos system firmware show
kcos system firmware show --raw
```

See also

```
kcos system firmware upgrade
```

kcos system firmware upgrade

Use the **kcos system firmware upgrade** command to upgrade firmware for a specific hardware component.

Syntax

```
kcos system firmware upgrade
   <{--all |
   <--FirmwareType <type> --NodeName <node>>
   }>
   [--help]
```

Flags

-h, --help	Display help for the command

Arguments

-a, --all	Upgrade all available firmware on all nodes.
-f, --FirmwareType type	The type of firmware that is being upgraded. Examples: BIOS, BMC, SSD-0.
-n, --NodeName node	Name of the node for which the firmware upgrade will be applied.

Examples

```
kcos system firmware upgrade -f SSD-1 -n master.kcos
kcos system firmware upgrade --all
kcos system firmware upgrade --help
```

See also

```
kcos system firmware show
```

Notes

If you do not use the `--all` option, you must specify the firmware and the node that is to be upgraded.

kcos system introspection bmc power-cycle

Use the **kcos system introspection bmc power-cycle** command to direct a BMC to power-cycle the node on which that BMC resides.

Syntax

```
kcos system introspection bmc power-cycle <flag>
```

Flags

The command requires one (and only one) of the following flags:

-b, --bmcName string	The BMC that will execute the power-cycle operation.
-h, --help	Display help for the command.
-n, --nodeName string	The specific node that the BMC will power-cycle.

Arguments

None.

Examples

```
kcos system introspection bmc power-cycle -b bmc-00-15-b2-ae-4a-45
kcos system introspection bmc power-cycle -n eagle-aps-o1-tw21160132
kcos system introspection bmc power-cycle --help
```

See also

```
kcos system introspection bmcs show
kcos system introspection bmc power-off
kcos system introspection bmc power-on
kcos system introspection bmc power-status
kcos system introspection bmc restart
kcos system introspection node restart
```

kcos system introspection bmc power-off

Use the **kcos system introspection bmc power-off** command to direct a BMC to power-down the node on which that BMC resides.

Syntax

```
kcos system introspection bmc power-off <flag>
```

Flags

The command requires one (and only one) of the following flags:

-b, --bmcName string	The BMC that will execute the power-off operation.
-h, --help	Display help for the command.
-n, --nodeName string	The specific node that the BMC will power down.

Arguments

None.

Examples

```
kcos system introspection bmc power-off -b bmc-00-15-b2-ae-4a-45
kcos system introspection bmc power-off -n eagle-aps-o1-tw21160132
kcos system introspection bmc power-off --help
```

See also

```
kcos system introspection bmcs show
kcos system introspection bmc power-cycle
kcos system introspection bmc power-on
kcos system introspection bmc power-status
kcos system introspection bmc restart
kcos system introspection system poweroff
```

kcos system introspection bmc power-on

Use the **kcos system introspection bmc power-on** command to direct a BMC to power-up the node on which that BMC resides.

Syntax

```
kcos system introspection bmc power-on <flag>
```

Flags

The command requires one (and only one) of the following flags:

-b, --bmcName string	The BMC that will execute the power-on operation.
-h, --help	Display help for the command.
-n, --nodeName string	The specific node that the BMC will power up.

Arguments

None.

Examples

```
kcos system introspection bmc power-on -b bmc-00-15-b2-ae-4a-45
kcos system introspection bmc power-on -n eagle-aps-o1-tw21160132
kcos system introspection bmc power-on --help
```

See also

```
kcos system introspection bmcs show
kcos system introspection bmc power-cycle
kcos system introspection bmc power-off
kcos system introspection bmc power-status
kcos system introspection bmc restart
```

kcos system introspection bmc power-status

Use the **kcos system introspection bmc power-status** command to direct a BMC to obtain and show the power-status of the node on which that BMC resides.

Syntax

```
kcos system introspection bmc power-status <flag>
```

Flags

The command requires one (and only one) of the following flags:

-b, --bmcName string	The BMC that will generate the power status report.
-h, --help	Display help for the command.
-n, --nodeName string	The node for which the BMC is reporting the power status.

Arguments

None.

Command output

The command displays a status message indicating whether the power is ON for OFF.

Examples

```
kcos system introspection bmc power-status -b bmc-00-15-b2-ae-4a-45
kcos system introspection bmc power-status -n eagle-aps-o1-tw21160132
kcos system introspection bmc power-status --help
```

See also

```
kcos system introspection bmcs show
kcos system introspection bmc power-cycle
kcos system introspection bmc power-off
kcos system introspection bmc power-on
kcos system introspection bmc restart
```

kcos system introspection bmc restart

Use the **kcos system introspection bmc restart** command to direct a BMC to restart the node on which that BMC resides.

Syntax

```
kcos system introspection bmc restart <flag>
```

Flags

The command requires one (and only one) of the following flags:

-b, --bmcName string	The BMC that will restart the node.
-h, --help	Display help for the command.
-n, --nodeName string	The node that the BMC will restart.

Arguments

None.

Examples

```
kcos system introspection bmc restart -b bmc-00-15-b2-ae-4a-45
kcos system introspection bmc restart -n eagle-aps-o1-tw21160132
kcos system introspection bmc restart --help
```

See also

```
kcos system introspection bmcs show
kcos system introspection bmc power-cycle
kcos system introspection bmc power-off
kcos system introspection bmc power-on
kcos system introspection bmc power-status
kcos system introspection node restart
```

kcos system introspection bmcs show

Use the **kcos system introspection bmcs show** command to display information about all available BMCs in the KCOS system.

Syntax

```
kcos system introspection bmcs show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Command output

The information returned includes the following values:

Name:	The name that uniquely identifies the BMC on *NodeName*.
Hostname:	The name of the assigned host.
SerialNumber:	The node's product serial number.
NodeName:	The name that uniquely identifies the node on which the BMC resides.

Examples

```
kcos system introspection bmcs show
kcos system introspection bmcs show --help
```

See also

```
kcos logs application
kcos system introspection hosts show
kcos system introspection nodes show
kcos system introspection pods show
```

kcos system introspection hosts show

Use the **kcos system introspection hosts show** command to display the Custom Resource Definition (CRD) information for the system's hosts. The CRD records DHCP leases served by the internal DHCP server on the backplane control network.

Syntax

```
kcos system introspection hosts show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Command output

For each host in the system, the returned information includes the following values:

Name	The name as reported by the node's kernel.
Host name	The Host name option as specified in the DHCP request. For compute nodes, the Node name is set (by default) to the host name of the machine.
IP Address	The IP address assigned by the DHCP server.
Vendor Class	An identifier that uniquely identifies the type of vendor device.

Examples

```
kcos system introspection hosts show
kcos system introspection hosts show --help
```

See also

```
kcos system introspection node restart
kcos system introspection node show
kcos system introspection nodes show
kcos system introspection pod delete
kcos system introspection pod show
kcos system introspection pods show
```

Notes

Each compute node will have two entries in the hosts list: one is for the Baseboard Management Controller (BMC), and the other is for the server.

kcos system introspection node restart

Use the **kcos system introspection node restart** command to restart the specified Kubernetes compute node.

Syntax

```
kcos system introspection node restart <node name> [flags]
```

Flags

-h, --help	Display help for the command
-p, ---power-cycle	Initiate a restart using the node's baseboard management controller (BMC) to power-cycle the node.

Arguments

node name	The name of the Kubernetes node that you need to restart.

Examples

```
kcos system introspection node restart cn-apa-c1-tw30140922
kcos system introspection node restart cn-apa-c1-tw30140922 --power-cycle
kcos system introspection node restart --help
```

See also

```
kcos system introspection bmc power-cycle
kcos system introspection bmc restart
kcos system introspection node show
kcos system introspection nodes show
```

Notes

You cannot restart a non-compute node.

kcos system introspection node show

Use the **kcos system introspection node show** command to display information about a specific Kubernetes node.

Syntax

```
kcos system introspection node show <node name> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

node name	The name of the Kubernetes node for which you are requesting information.

Command output

The information returned includes the following values:

Name:	The name that uniquely identifies the node.
Status:	The operational state of the node. A Ready state indicates that the node is healthy and able to accept pods.
Role:	The node's functional role within the cluster. Nodes are either compute nodes or non-compute nodes.
Version:	The Kubernetes version.
Internal IP:	The internally-routable IP address of the node (routable only within the cluster).
OS Image:	Name of the OS image.
Kernel Version:	Version of the kernel running on this node.
Container Runtime:	The Docker container runtime installed on the node.

Examples

```
kcos system introspection node show eagle-c0ffee010100
```

See also

```
kcos system introspection hosts show
kcos system introspection node restart
kcos system introspection nodes show
kcos system introspection pod delete
kcos system introspection pod show
kcos system introspection pods show
```

kcos system introspection nodes show

Use the **kcos system introspection nodes show** command to display information about all system nodes.

Syntax

```
kcos system introspection nodes show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Command output

The information returned includes the following values:

Name:	The name that uniquely identifies the node.
Status:	The operational state of the node. A Ready state indicates that the node is healthy and able to accept pods.
Role:	The node's functional role within the cluster. Nodes are either compute nodes or non-compute nodes.
Version:	The Kubernetes version.
Internal IP:	The internally-routable IP address of the node (routable only within the cluster).
OS Image:	Name of the OS image.
Kernel Version:	Version of the kernel running on this node.
Container Runtime:	The Docker container runtime installed on the node.

Examples

```
kcos system introspection nodes show
kcos system introspection nodes show --help
```

See also

```
kcos system introspection hosts show
kcos system introspection node restart
kcos system introspection node show
kcos system introspection pod delete
kcos system introspection pod show
kcos system introspection pods show
```

Notes

If a blade does not show up properly in the BPS user interface, you can start troubleshooting by using this command to check the nodes. All nodes should be in a *Ready* state.

kcos system introspection pod delete

Use the **kcos system introspection pod delete** command to delete the specified pod from the KCOS system. When you issue this command, KCOS terminates the pod and starts up a new pod.

Syntax

```
kcos system introspection pod delete <pod namespace> <pod name> [flags]
```

Flags

-h, --help	Display help for the command

Arguments

pod namespace	The namespace associated with the pod that you are deleting.
pod name	The name of the pod that you are deleting.

Examples

```
kcos system introspection pod delete kcos-framework mypod-fru-2
kcos system introspection pod delete --help
```

See also

```
kcos system introspection hosts show
kcos system introspection node restart
kcos system introspection node show
kcos system introspection nodes show
kcos system introspection pod show
kcos system introspection pods show
```

Notes

If the *kcos system introspection pods show* command lists a pod whose Status field shows a state other than Running or Completed, you can use the *kcos system introspection pod delete* command to restart the service.

kcos system introspection pod show

Use the **kcos system introspection pod show** command to display information about a specific pod.

Syntax

```
kcos system introspection pod show <pod namespace> <pod name> [flags]
```

Flags

-e, --events	When present, this flag displays an additional table listing the events from the pod (if there are any such events). The table contains the following columns: Event Type: The type of event. Reason: Reason that the event was initiated. From: The component from which the event was initiated. Message: The message issued by the event.
-h, --help	Display help for the command

Arguments

pod namespace	The namespace in which the Pod Name is defined.
pod name	The name of the pod for which you are requesting information.

Command output

The returned information includes the following values:

Namespace:	The namespace in which the Pod Name is defined.
Name:	The name that uniquely identifies the pod within a given namespace.
Ready:	Displays the number of containers that are currently running, out of the total desired number of containers. For example, "2/3" indicates that the desired number of containers for the pod is three, but only two are running.
Status:	The operational state of the pod.
Pod IP:	The IP address that the pod's containers use to communicate with each other.
Host IP:	The IP address of the host to which the pod is assigned.
Node:	The name of the node in which the pod is scheduled.

Examples

```
kcos system introspection pod show kcos-sso keycloak-postgresql-d48bc000a9
kcos system introspection pod show bps bps-web-1234abcde-zyxwv --events
kcos system introspection pod show --help
```

See also

```
kcos system introspection hosts show
kcos system introspection node restart
kcos system introspection node show
kcos system introspection nodes show
kcos system introspection pod delete
kcos system introspection pods show
```

Notes

If a pod's Status field shows a state other than Running or Completed, you can use the *kcos system introspection pod delete* command to restart the service.

kcos system introspection pods show

Use the **kcos system introspection pods show** command to display information about all pods running on the system.

Syntax

```
kcos system introspection pods show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Command output

The returned information includes—for each pod listed—the following values:

Namespace:	The namespace in which the Pod Name is defined.
Name:	The name that uniquely identifies the pod within a given namespace.
Ready:	Displays the number of containers that are currently running, out of the total desired number of containers. For example, "2/3" indicates that the desired number of containers for the pod is three, but only two are running.
Status:	The operational state of the pod.
Pod-IP:	The IP address that the pod's containers use to communicate with each other.
Host-IP:	The IP address of the host to which the pod is assigned.
Node:	The name of the node in which the pod is scheduled.

Examples

```
kcos system introspection pods show
kcos system introspection pods show --help
```

See also

```
kcos system introspection hosts show
kcos system introspection node restart
kcos system introspection node show
kcos system introspection nodes show
kcos system introspection pod delete
kcos system introspection pod show
```

Notes

If a slot is not working even though the node is up, you can use this command to verify the state of all the pods in the system. If a pod's Status field shows a state other than Running or Completed, you can use the *kcos system introspection pod delete* command to restart the service.

kcos system poweroff

Use the **kcos system poweroff** command to start a shutdown and power-off the system.

Syntax

```
kcos system poweroff [flags]
```

Flags

-h, --help	Display help for the command
-y, --yes	Confirm the power-down operation

Arguments

None.

Examples

```
kcos system poweroff --yes
```

See also

```
kcos system reboot
```

kcos system reboot

Use the **kcos system reboot** command to reboot the system.

Syntax

```
kcos system reboot [flags]
```

Flags

-h, --help	Display help for the command
-y, --yes	Confirm the reboot operation

Arguments

None.

Examples

```
kcos system reboot --yes
```

See also

```
kcos system poweroff
```

kcos system welcome-screen show

Use the **kcos system welcome-screen show** command to display the welcome message/banner for the admin user. The displayed information includes the node's IP address and the names of the installed packages.

Syntax

```
kcos system welcome-screen show [flags]
```

Flags

-h, --help	Display help for the command

Arguments

None.

Examples

```
kcos system welcome-screen show
```

See also

```
kcos networking ip show
```

Index

www.keysight.com