ip COMMAND CHEAT SHEET

for Red Hat Enterprise Linux

IP QUERIES

SUBCOMMAND	DESCRIPTIONS AND TASKS
`addr`	Display IP Addresses and property information (abbreviation of address) **`ip addr`** Show information for all addresses **`ip addr show dev em1`** Display information only for device em1
`link`	Manage and display the state of all network interfaces **`ip link`** Show information for all interfaces **`ip link show dev em1`** Display information only for device em1 **`ip -s link`** Display interface statistics
`route`	Display and alter the routing table **`ip route`** List all of the route entries in the kernel
`maddr`	Manage and display multicast IP addresses **`ip maddr`** Display multicast information for all devices **`ip maddr show dev em1`** Display multicast information for device em1
`neigh`	Show neighbour objects; also known as the ARP table for IPv4 **`ip neigh`** Display neighbour objects **`ip neigh show dev em1`** Show the ARP cache for device em1
`help`	Display a list of commands and arguments for each subcommand **`ip help`** Display ip commands and arguments **`ip addr help`** Display address commands and arguments **`ip link help`** Display link commands and arguments **`ip neigh help`** Display neighbour commands and arguments

MULTICAST ADDRESSING

SUBCOMMAND	DESCRIPTIONS AND TASKS
`maddr add`	Add a static link-layer multicast address **`ip maddr add 33:33:00:00:00:01 dev em1`** Add mutlicast address 33:33:00:00:00:01 to em1
`maddr del`	Delete a multicast address **`ip maddr del 33:33:00:00:00:01 dev em1`** Delete address 33:33:00:00:00:01 from em1

MODIFYING ADDRESS AND LINK PROPERTIES

SUBCOMMAND	DESCRIPTIONS AND TASKS
`addr add`	Add an address **`ip addr add 192.168.1.1/24 dev em1`** Add address 192.168.1.1 with netmask 24 to device em1
`addr del`	Delete an address **`ip addr del 192.168.1.1/24 dev em1`** Remove address 192.168.1.1/24 from device em1
`link set`	Alter the status of the interface **`ip link set em1 up`** Bring em1 online **`ip link set em1 down`** Bring em1 offline **`ip link set em1 mtu 9000`** Set the MTU on em1 to 9000 **`ip link set em1 promisc on`** Enable promiscuous mode for em1

ADJUSTING AND VIEWING ROUTES

SUBCOMMAND	DESCRIPTIONS AND TASKS
`route add`	Add an entry to the routing table **`ip route add default via 192.168.1.1 dev em1`** Add a default route (for all addresses) via the local gateway 192.168.1.1 that can be reached on device em1 **`ip route add 192.168.1.0/24 via 192.168.1.1`** Add a route to 192.168.1.0/24 via the gateway at 192.168.1.1 **`ip route add 192.168.1.0/24 dev em1`** Add a route to 192.168.1.0/24 that can be reached on device em1
`route delete`	Delete a routing table entry **`ip route delete 192.168.1.0/24 via 192.168.1.1`** Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.1
`route replace`	Replace, or add if not defined, a route **`ip route replace 192.168.1.0/24 dev em1`** Replace the defined route for 192.168.1.0/24 to use device em1
`route get`	Display the route an address will take **`ip route get 192.168.1.5`** Display the route taken for IP 192.168.1.5

MANAGING THE ARP TABLE

SUBCOMMAND	DESCRIPTIONS AND TASKS
`neigh add`	Add an entry to the ARP Table **`ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1`** Add address 192.168.1.1 with MAC 1:2:3:4:5:6 to em1
`neigh del`	Invalidate an entry **`ip neigh del 192.168.1.1 dev em1`** Invalidate the entry for 192.168.1.1 on em1
`neigh replace`	Replace, or adds if not defined, an entry to the ARP table **`ip neigh replace 192.168.1.1 lladdr 1:2:3:4:5:6 dev em1`** Replace the entry for address 192.168.1.1 to use MAC 1:2:3:4:5:6 on em1

USEFUL NETWORKING COMMANDS (NOT NECESSARILY PROVIDED FROM IPROUTE)

SUBCOMMAND	DESCRIPTIONS AND TASKS
`arping`	Send ARP request to a neighbour host **`arping -I eth0 192.168.1.1`** Send ARP request to 192.168.1.1 via interface eth0 **`arping -D -I eth0 192.168.1.1`** Check for duplicate MAC addresses at 192.168.1.1 on eth0
`ethtool`	Query or control network driver and hardware settings **`ethtool -g eth0`** Display ring buffer for eth0 **`ethtool -i eth0`** Display driver information for eth0 **`ethtool -p eth0`** Identify eth0 by sight, typically by causing LEDs to blink on the network port **`ethtool -S eth0`** Display network and driver statistics for eth0
`ss`	Display socket statistics. The below options can be combined **`ss -a`** Show all sockets (listening and non-listening) **`ss -e`** Show detailed socket information **`ss -o`** Show timer information **`ss -n`** Do not resolve addresses **`ss -p`** Show process using the socket

COMPARING NET-TOOLS VS. IPROUTE PACKAGE COMMANDS

NET-TOOLS COMMANDS	IPROUTE COMMANDS
`arp -a`	`ip neigh`
`arp -v`	`ip -s neigh`
`arp -s 192.168.1.1 1:2:3:4:5:6`	`ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1`
`arp -i eth1 -d 192.168.1.1`	`ip neigh del 192.168.1.1 dev eth1`
`ifconfig -a`	`ip addr`
`ifconfig eth0 down`	`ip link set eth0 down`
`ifconfig eth0 up`	`ip link set eth0 up`
`ifconfig eth0 192.168.1.1`	`ip addr add 192.168.1.1/24 dev eth0`
`ifconfig eth0 netmask 255.255.255.0`	`ip addr add 192.168.1.1/24 dev eth0`
`ifconfig eth0 mtu 9000`	`ip link set eth0 mtu 9000`
`ifconfig eth0:0 192.168.1.2`	`ip addr add 192.168.1.2/24 dev eth0`
`netstat`	`ss`
`netstat -neopa`	`ss -neopa`
`netstat -g`	`ip maddr`
`route`	`ip route`
`route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0`	`ip route add 192.168.1.0/24 dev eth0`
`route add default gw 192.168.1.1`	`ip route add default via 192.168.1.1`

redhat.