MikroTik CHR (Cloud Hosted Router) MANUAL Setup Guide

Purpose: MikroTik CHR is a cloud-based virtual router designed to provide network routing functionalities in virtualized environments. It allows you to leverage MikroTik's RouterOS features in cloud infrastructures, making it ideal for network management, VPN services, firewall protection, and bandwidth management in a virtualized or cloud-based setup.

Use Cases:

1. **Virtual Private Network (VPN):** CHR can be used to manage and route VPN traffic, ensuring secure and efficient connectivity between remote locations.
2. **Network Management:** Ideal for managing complex network environments, including routing, switching, and traffic shaping.
3. **Firewall and Security:** Provides robust firewall capabilities to secure network traffic and protect against unauthorized access.
4. **Bandwidth Management:** Useful for monitoring and controlling bandwidth usage to optimize network performance.

Installation Guide:

1. Prepare Your Environment:

- Ensure you have a cloud environment or virtualization platform where you can deploy the CHR. Supported platforms include AWS, Azure, Google Cloud, VMware, Hyper-V, and others.

2. Download MikroTik CHR Image:

- Visit the MikroTik official website or MikroTik.com to download the appropriate CHR image. Choose between different versions based on your needs (e.g., stable or testing).

3. Deploy CHR in Your Cloud Environment:

- **AWS:** Create a new instance and upload the CHR image. Configure the instance with appropriate resources (CPU, RAM, storage).
- **Azure:** Use the Azure Marketplace to deploy a MikroTik CHR virtual machine.
- **VMware/Hyper-V:** Create a new virtual machine and attach the CHR image to it.

4. Initial Configuration:

- **Access CHR:** Connect to the CHR instance using SSH or a console connection.
- **Basic Configuration:** Set up network interfaces, IP addresses, and routing protocols as needed. Refer to MikroTik documentation for specific commands and configurations.

5. Advanced Configuration (Optional):

- **VPN Setup:** Configure VPN tunnels for secure remote access.
- **Firewall Rules:** Set up firewall rules to protect your network.
- **Bandwidth Management:** Implement traffic shaping and bandwidth control policies.

6. Management and Monitoring:

- Use MikroTik's WinBox or WebFig to manage and monitor the CHR instance. These tools provide a graphical interface for configuration and monitoring.

7. Regular Maintenance:

- Keep your CHR instance updated with the latest software releases and patches to ensure security and performance.

Considerations:

- **Licensing:** MikroTik CHR operates under different license levels. Choose a license based on your performance and feature requirements.
- **Resource Allocation:** Ensure your virtual environment provides adequate resources to handle your network traffic and routing needs.

Resources:

MikroTik Documentation: MikroTik CHR Documentation

Community Forums: Engage with the MikroTik community for support and additional tips.

Standart (Long) Script for automated installation

```
# Determine the package manager
if command -v yum &> /dev/null; then
    pkg_manager="yum";
elif command -v apt &> /dev/null; then
    pkg_manager="apt";
else
    echo "Neither yum nor apt found. This script is not supported.";
    exit 1;
fi

# Update packages and install unzip, pwgen, and coreutils
if [ "$pkg_manager" == "yum" ]; then
    sudo yum -y update && sudo yum -y install unzip pwgen coreutils;
elif [ "$pkg_manager" == "apt" ]; then
    sudo apt-get -y update && sudo apt-get -y install unzip pwgen coreutils;
fi
```

```
echo "The system is updated and required packages are installed."

# Determine the root file system device
root_device=$(df / | awk 'NR==2 {print $1}')
root_device_base=$(echo $root_device | sed 's/[0-9]\+$//')

echo "Root filesystem is on device: $root_device"
echo "Device path: $root_device_base"

# Create and mount a temporary directory
mkdir /mt_ros_tmp && mount -t tmpfs tmpfs /mt_ros_tmp/ && cd /mt_ros_tmp

# Get IP address and gateway
INTERFACE=$(ip route | grep default | awk '{print $5}')
ADDRESS=$(ip addr show "$INTERFACE" | grep global | cut -d' ' -f 6 | head -n 1)
GATEWAY=$(ip route list | grep default | cut -d' ' -f 3)

echo "Please enter the channel (default='stable', or='testing'): "
read channel

# Default to 'stable' if no input is provided
if [ -z "$channel" ]; then
    channel="stable"
fi

echo "Installing RouterOS CHR from the '$channel' channel..."

# Download URL based on selected channel
if [ "$channel" == "testing" ]; then
    rss_feed="https://download.mikrotik.com/routeros/latest-testing.rss"
else
    rss_feed="https://download.mikrotik.com/routeros/latest-stable.rss"
fi

# Download the latest version of MikroTik RouterOS
rss_content=$(curl -s $rss_feed)
latest_version=$(echo "$rss_content" | grep -oP '(?<=<title>RouterOS )[\d\.]
+rc\d+' | head -1)

if [ -z "$latest_version" ]; then
    echo "Could not retrieve the latest version number."
    exit 1
fi

echo "Latest version: $latest_version"
download_url="https://download.mikrotik.com/routeros/$latest_version/chr-
$latest_version.img.zip"

echo "Downloading from $download_url..."
wget --no-check-certificate -O "chr-$latest_version.img.zip" "$download_url"

if [ $? -eq 0 ]; then
```

```
    echo "File successfully downloaded: chr-$latest_version.img.zip"
else
    echo "File download failed."
    exit 1
fi

# Unzip and prepare the image
gunzip -c "chr-$latest_version.img.zip" > "chr-$latest_version.img"

# Mount the image
mount -o loop "chr-$latest_version.img" /mnt

# Generate a random password
PASSWORD=$(pwgen 12 1)

# Write autorun script to configure the RouterOS instance
echo "Username (Kullanıcı adı): admin"
echo "Password (Şifre): $PASSWORD"

echo "/ip address add address=$ADDRESS interface=[/interface ethernet find
where name=ether1]" > /mnt/rw/autorun.scr
echo "/ip route add gateway=$GATEWAY" >> /mnt/rw/autorun.scr
echo "/ip service disable telnet" >> /mnt/rw/autorun.scr
echo "/user set 0 name=admin password=$PASSWORD" >>
/mnt/rw/autorun.scr
echo "/ip dns set server=8.8.8.8,1.1.1.1" >> /mnt/rw/autorun.scr

# Remount all mounted filesystems to read-only mode
sync && echo u > /proc/sysrq-trigger

# Flash the image to the disk
dd if="chr-$latest_version.img" of=$root_device_base bs=4M oflag=sync

# Force system reboot
echo 1 > /proc/sys/kernel/sysrq
echo b > /proc/sysrq-trigger
```

ONE-LINER (Short) SCRiPT for Automated Installations

```
if command -v yum &> /dev/null; then pkg_manager="yum"; elif command -v
apt &> /dev/null; then pkg_manager="apt"; else echo "Neither yum nor apt
found. This script is not supported."; exit 1; fi && \
[ "$pkg_manager" == "yum" ] && sudo yum -y update && sudo yum -y install
unzip pwgen coreutils || [ "$pkg_manager" == "apt" ] && sudo apt-get -y
update && sudo apt-get -y install unzip pwgen coreutils && \
root_device=$(df / | awk 'NR==2 {print $1}') && root_device_base=$(echo
$root_device | sed 's/[0-9]\+$//') && \
echo "Root filesystem is on device: $root_device" && echo "Device path:
$root_device_base" && \
mkdir /mt_ros_tmp && mount -t tmpfs tmpfs /mt_ros_tmp/ && cd /mt_ros_tmp
&& \
```

```
INTERFACE=$(ip route | grep default | awk '{print $5}') && ADDRESS=$(ip addr show "$INTERFACE" | grep global | awk '{print $2}' | head -n 1) && \
GATEWAY=$(ip route list | grep default | awk '{print $3}') && \
read -p "Enter channel (default='stable', or='testing'): " channel; [ -z "$channel" ] &&
channel="stable";rss_feed="https://download.mikrotik.com/routeros/latest-$channel.rss" && rss_content=$(curl -s $rss_feed) && \
latest_version=$(echo "$rss_content" | grep -oP '(?<=<title>RouterOS )[\d\.]+rc\d+' | head -1) && \
[ -z "$latest_version" ] && echo "Could not retrieve the latest version number." && exit 1 || \
echo "Latest version: $latest_version" &&
download_url="https://download.mikrotik.com/routeros/$latest_version/chr-$latest_version.img.zip" && \
echo "Downloading from $download_url..." && wget --no-check-certificate -O "chr-$latest_version.img.zip" "$download_url" && \
[ $? -eq 0 ] && echo "File successfully downloaded: chr-$latest_version.img.zip" || echo "File download failed." && \
gunzip -c "chr-$latest_version.img.zip" > "chr-$latest_version.img" && mount -o loop "chr-$latest_version.img" /mnt && \
PASSWORD=$(pwgen 12 1) && echo "Username: admin" && echo "Password: $PASSWORD" && \
echo "/ip address add address=$ADDRESS interface=[/interface ethernet find where name=ether1]" > /mnt/rw/autorun.scr && \
echo "/ip route add gateway=$GATEWAY" >> /mnt/rw/autorun.scr && echo "/ip service disable telnet" >> /mnt/rw/autorun.scr && \
echo "/user set 0 name=admin password=$PASSWORD" >> /mnt/rw/autorun.scr && echo "/ip dns set server=8.8.8.8,1.1.1.1" >> /mnt/rw/autorun.scr && \
sync && echo u > /proc/sysrq-trigger && dd if="chr-$latest_version.img" of=$root_device_base bs=4M oflag=sync && \
echo 1 > /proc/sys/kernel/sysrq && echo b > /proc/sysrq-trigger
```

Automation Scripts' Key Updates and Explanations:

1. **Installing Additional Packages:**
   - Added installation commands for pwgen and coreutils in both yum and apt package managers.
2. **IP Address and Gateway Retrieval:**
   - The script captures the system's IP address and gateway using ip addr and ip route.
3. **Unzipping and Mounting:**
   - The image is unzipped and mounted using gunzip and mount commands with appropriate options.
4. **Generating and Setting Password:**

- A random 12-character password is generated using pwgen and then set in the autorun script for RouterOS.

5. **Autorun Script:**
   - The autorun script includes commands to configure the RouterOS instance, including adding the IP address, setting the gateway, disabling telnet, setting the admin password, and configuring DNS servers.
6. **System Reboot:**
   - Filesystem sync is performed before forcing a system reboot using the SysRq trigger, ensuring that all data is written to disk.
7. **Automatic Network Interface Detection:**
   - INTERFACE=$(ip route | grep default | awk '{print $5}'): Automatically detects the active network interface by finding the default route's interface.
   - The ADDRESS variable is then set using this detected interface.