Application

Fix login at the middle of cloud init

Fix login at the middle of cloud init

Modifications to `userdata.yml`

1. Late Commands (Executed After Installation Completes)

Modify `appliance/image/ubuntu/build/userdata.yml` under `late-commands` to ensure the `getty@tty1.service` is disabled during post-installation:

```
late-commands:
  - curtin in-target --target=/target -- systemctl disable getty@tty1.service
  - curtin in-target --target=/target -- systemctl daemon-reload
```

2. User Data (Post-Installation Commands)

Modify `appliance/image/ubuntu/build/userdata.yml` under `user-data/runcmd` to re-enable `getty@tty1.service` after installation and reboot the system:

```
user-data:
  runcmd:
    - systemctl enable getty@tty1.service
    - reboot
```

Building the ISO Using `appliance_ubuntu_branch` Pipeline

Steps to Trigger the Build:

1. **Ensure Correct Branch Selection:**
   - `appliance_branch`: As specified in the JIRA.
   - `sky_job`: Use the current master branch.
2. **Mark `upgrade_packages` as enabled**, as there is a known bug.
3. **Start the Build Pipeline.**
4. **Wait for the Build to Complete.**
5. **Download the Generated `.iso` File** from the archived artifacts of the job.

Deploying the ISO on ESXi

1. Log in to ESXi:

- Use the following credentials:
  - **Username:** `tomer`
  - **Password:** (local password)

2. Configure the Virtual Machine:

a) Force BIOS Setup

1. Navigate to the VM.

2. Click **Edit → VM Options → Boot Options**.
3. Enable **Force BIOS Setup on Boot**.

b) Upload the ISO File

1. Navigate to **Edit → Virtual Hardware**.
2. Upload the `.iso` file to the correct directory.
3. **Unmark the "Connect" checkbox** to prevent immediate boot.
4. Restart the VM and enter the BIOS.

c) Start Installation

1. Select the uploaded ISO.
2. Choose **"Server Installation"** when prompted.

Troubleshooting

1. Network Issues (Connecting to the Internet)

If the VM does not connect to the internet, modify the Netplan configuration:

1. Open the network configuration file:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

2. Ensure the following configuration is present:

```
network:
  ethernets:
    ens160:
      dhcp4: true
      dhcp-identifier: mac
      dhcp6: false
```

3. Apply the changes:

```
sudo netplan apply
```

4. Verify the new IP address:

```
ifconfig
```

2. Checking Installation Logs

To review the terminal output of the installation process:

```
cat /root/appliance_installation.log
```

This structured guide ensures clarity and smooth execution of the entire process.

Configure Syslog-ng for SSL/TLS Encryption

Configuring Syslog-ng for SSL/TLS Encryption

Step 1: Create Necessary Directories

Ensure the required directories exist and set the correct permissions:

```
sudo mkdir -p /etc/syslog-ng/key.d
```

```
sudo mkdir -p /etc/syslog-ng/ca.d
```

```
sudo chmod 700 /etc/syslog-ng/key.d /etc/syslog-ng/ca.d
```

Step 2: Generate a Self-Signed Certificate

This will serve as both the **server certificate** and its **own Certificate Authority (CA)**.

```
sudo openssl req -x509 -newkey rsa:2048 -nodes \ -keyout /etc/syslog-ng/key.d/syslog-ng.key \ -out /etc/syslog-ng/ca.d/syslog-ng.crt \ -days 365 \ -subj "/CN=$(hostname)/O=Skyboxsecurity/C=IL"
```

```
sudo chmod 400 /etc/syslog-ng/key.d/syslog-ng.key
```

```
sudo chmod 444 /etc/syslog-ng/ca.d/syslog-ng.crt
```

```
sudo ln -sf /etc/syslog-ng/ca.d/syslog-ng.crt \ /etc/syslog-ng/ca.d/$(openssl x509 -noout -hash -in /etc/syslog-ng/ca.d/syslog-ng.crt).0
```

Step 3: Configure Syslog-ng for TLS

Create a configuration file for **TLS-enabled syslog**:

```
sudo tee /etc/syslog-ng/conf.d/skybox-syslog-ng.conf << 'EOF' source s_tls { network( ip("0.0.0.0") port(6514) transport("tls") tls( key-file("/etc/syslog-ng/key.d/syslog-ng.key") cert-file("/etc/syslog-ng/ca.d/syslog-ng.crt") ca-dir("/etc/syslog-ng/ca.d") peer-verify(optional-untrusted) ) ); }; destination d_local { file("/var/log/tls-messages"); }; log { source(s_tls); destination(d_local); }; EOF
```

Verify the configuration:

```
sudo syslog-ng -s
```

Step 4: Configure Systemd Service

Create an **override configuration** for the Syslog-ng service:

```
sudo mkdir -p /etc/systemd/system/syslog-ng.service.d
```

```
sudo tee /etc/systemd/system/syslog-ng.service.d/override.conf << 'EOF' [Service] CapabilityBoundingSet=CAP_NET_BIND_SERVICE CAP_SYSLOG CAP_DAC_READ_SEARCH AmbientCapabilities=CAP_NET_BIND_SERVICE CAP_SYSLOG CAP_DAC_READ_SEARCH EOF
```

Step 5: Configure SELinux (If Enabled - was not)

Ensure the correct SELinux contexts are applied:

```
if command -v semanage >/dev/null; then sudo semanage fcontext -a -t syslog_conf_t "/etc/syslog-ng/ca.d(/.*)?" sudo semanage fcontext -a -t syslog_conf_t "/etc/syslog-ng/key.d(/.*)?" sudo restorecon -Rv /etc/syslog-ng/ca.d/ sudo restorecon -Rv /etc/syslog-ng/key.d/ fi
```

Step 6: Configure Firewall

Allow incoming connections on port **6514** for TLS syslog:

```
if command -v firewall-cmd >/dev/null; then sudo firewall-cmd --permanent --add-port=6514/tcp sudo firewall-cmd --reload elif command -v ufw >/dev/null; then sudo ufw allow 6514/tcp fi
```

Step 7: Restart and Test Syslog-ng

Restart the Service

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart syslog-ng
```

Verify the Service is Running

```
sudo systemctl status syslog-ng
```

Test the TLS Connection

```
openssl s_client -connect localhost:6514 -CAfile /etc/syslog-ng/ca.d/syslog-ng.crt
```

Final Directory Structure

After completing all steps, the **directory structure** should look like this:

```
/etc/syslog-ng/
```

```
├── ca.d/
```

```
│ ├── syslog-ng.crt
```

```
│ └── [hash].0 -> syslog-ng.crt
```

```
├── key.d/
```

```
│ └── syslog-ng.key
```

```
└── conf.d/
```

```
└── skybox-syslog-ng.conf
```

Creating a Self-Signed Certificate With OpenSSL | Baeldung